University of Southern Queensland

Faculty of Engineering and Surveying

Can High-Fidelity Photorealism Provided by Unreal Engine 5 Bridge the Synthetic-to-Reality Gap for Improved Multi-Rotor Drone Detection?

A research proposal submitted by

Aneesh Vasudevan

in fulfillment of requirements of

ENP4111 Professional Engineer Research Project

towards the degree of:

Bachelor of Engineering (Honours) Mechatronic Engineering

Submitted: January 19, 2025

Abstract

Rising challenges to international privacy, safety, security and peace posed by multi-rotor drone operators has accelerated research and development of counter small Unmanned Aircraft Systems (C-sUAS) systems. As part of this, computer vision (CV) approaches using deep learning (DL) have been typically utilised as an effective method for the detection of multi-rotor drones, by way of electro-optical (EO) camera detection systems. Since large datasets are often required to train and validate CV models, synthetically generated image datasets are a promising method that have been explored to meet this demand. Unreal Engine 5 (UE5) is a rendering engine that has grown in prominence due to achievable high-fidelity photorealism not only for gaming experiences, but also digital content creation such as imagery. This is made possible with *Lumen*, a new lighting system by UE5, which is a global illumination (GI) and reflections system, that simulates accurate lighting, soft shadows and physics-based material properties.

Training drone direction models exclusively with synthetic data, even with high-levels of photorealism, has long caused an issue called synthetic-to-reality gap, due to inherent differences between real-world and synthetic data. Strategies such as domain randomisation are commonly practiced to minimise this issue, however, the question still remains if high-fidelity photorealism in generated training imagery by a state-of-the-art rendering platform such as UE5, in tandem with other strategies can effectively bridge the synthetic-to-reality gap for improved drone detection. Preliminary results of this research have indicated that UE5 can in fact can assist in bridging this gap, however, future research is still required.

Key Words: Unreal Engine 5 * Drone Detection * Synthetic Data * Photorealism * Lighting * Fidelity

University of Southern Queensland

Faculty of Engineering and Surveying

ENP4111 Professional Engineering Research Project

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, and the staff of the University of Southern Queensland do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

University of Southern Queensland

Faculty of Engineering and Surveying

ENP4111 Professional Engineering Research Project

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses, and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Student Name: Aneesh Vasudevan

Supervisor: Dr. Tobias Low

Student Identity (ID): [redacted]

Student Signature: [redacted]

Signed Date: Sunday, 19 January 2025

Acknowledgements

I would like to express my deep thanks to my project supervisor Dr. Tobias Lowe for his valuable contribution to the evolution of this research project, with his patient support and expertise having been instrumental in the refinement of this project.

A sincere thank you must also be given to my industry mentor Mr. Andrew Tuxford for his encouragement and interest in the project, as well as the insights provided in discussion to navigate the final challenges of this project.

Finally, no words can define my gratitude to my father and mother for their fundamental role in my learning journey. I hope to make them proud now and always.

Table of Contents

List of Figures

List of Tables

Abbreviations / Acronyms / Initialisms

Term	Definition
AI	Artificial Intelligence
AoA	Angle of Attack
ANN	Artificial Neural Network
AP	Average Precision
API	Application Programming Interface
ARF	Almost Ready to Fly
A-sUAS	Armed-small Unmanned Aircraft System
ATC	Air Traffic Control
BNF	Bind-and-Fly
BVLOS	Beyond Visual Line of Sight
CAD	Computer Aided Design
CC	Creative Commons
CNN	Convolutional Neural Network
CPU	Central Processing Unit
C-sUAS	Counter small Unmanned Aircraft Systems
CV	Computer Vision
DCC	Digital Content Creation
DCNN	Deep Convolutional Neural Network
DIY	Do-it-Yourself
DL	Deep Learning
DNN	Deep Neural Network
DoD	Department of Defense
DPU	Data Processing Unit
DRL	Deep Reinforcement Learning
DSLR	Digital Single-Lens Reflex Camera
EO	Electro-Optical

EO/IR	Electro-Optical/Infra-Red
FN	False Negative
FNN	Feedforward Neural Network
FP	False Positive
GPU	Graphics Processing Unit
HDR	High Dynamic Range
IBL	Image-Based Lighting
IR	Infra-Red
LOD	Level of Detail
ML	Machine Learning
mAP	Mean Average Precision
MOA	Military Operations Area
MV	Machine Vision
OS	Operating System
OTS	Off-the-Shelf
PC	Personal Computer
PNP	Plug-and-Play
PSU	Power Supply Unit
RAM	Random Access Memory
RGB	Red, Green, Blue (Colour Model)
RV	Robotic Vision
RTF	Ready-to-Fly
SALW	Small Arms and Light Weapons
SDR	Standard Dynamic Range
SL	Supervised Learning
SPP	Samples per Pixel
SSGI	Screen Space Global Illumination
SSR	Screen Space Reflections
sUAS	Small Unmanned Aircraft System
TN	True Negative
TP	True Positive
TPU	Tensor Processing Unit
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UE4	Unreal Engine 4
UE5	Unreal Engine 5
UI	User Interface
VFX	Visual Effects
VGG	Visual Geometry Group
VLOS	Visual Line of Sight
VNSA	Violent Non-State Actor
VRAM	Video Random-Access Memory
VTOL	Vertical Take-Off and Landing
3D	Three Dimensional

Chapter 1 – Introduction

1.1 Background Context

1.1.1 Unmanned Aircraft System (UAS) Categorisation

The term unmanned aircraft system (UAS) is composed of three elements, a flight capable robotic system known as an unmanned aerial vehicle (UAV) (also referred to as a drone), the human operator (end-user) who is able to manually control (remotely pilot) the UAV, or delegate a level of autonomy to complete assigned task(s), and the datalink technologies, which enables remote control by an end user within their visual line of sight (VLOS) or beyond visual line of sight (BVLOS). However, the term UAS is commonly used to refer to the UAV itself, hence the following paper will keep with this convention.

Types of UAS include: Single-rotor, fixed-wing, multi-rotor, fixed-wing hybrid vertical take-off and landing (VTOL), as seen in Figure 1.1, which are defined by primary control surfaces, and or propulsion configuration (Garg 2022; Heiets et al. 2023).

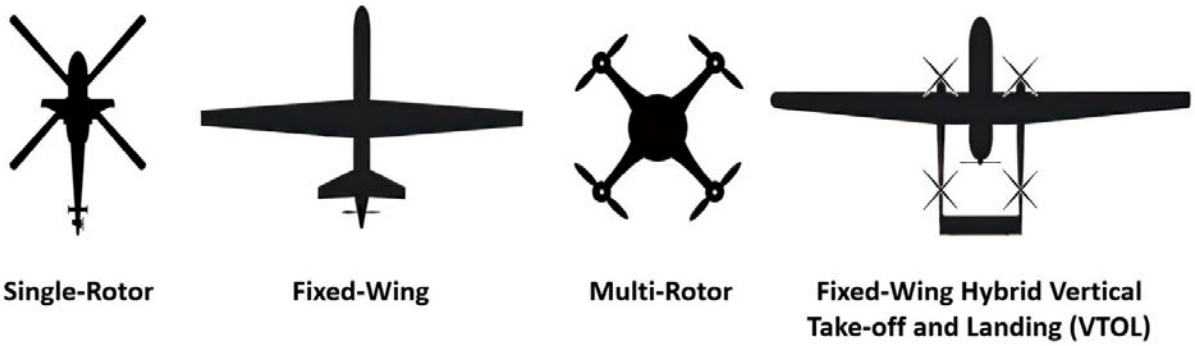


Figure 1.1: Types of Unmanned Aerial Systems (UAS) (Adapted from Alghamdi, Munir & La 2021)

The reader should note that a globally standardised classification scheme for UAS is non-existent, with variations amongst civilian and military sectors based on operational capabilities. For clarity the following paper considers the UAS categorisation framework by the United States Department of Defense (DoD) as seen in Table 1.1. It must be made clear that the selection of this scheme is not an indication of author bias or sponsored influence in this undergraduate research project; the author rather considers it to be an unambiguous classification framework for smaller types of drones, which is a focus point in this research.

Unmanned Aircraft System (UAS) Categorisation According to U.S. Department of Defense (DoD)						
UAS Category	Maximum Take of Mass		Normal Operating Altitude		Indicated Airspeed	
	Kilograms (kg)	Pounds (lbs)	Meters (m)	Feet (ft)	Kilometers per Hour (km/h)	Knots (Kn)
Group 1	0 - 9.07	0 - 20	< 365.76 AGL	< 1200 AGL	< 185.20	<100
Group 2	9.53 - 24.95	21 - 55	< 1066.80 AGL	< 3500 AGL	< 463.00	< 250
Group 3	< 598.74	< 1320	< 5486.40 MSL	< 18000 MSL		
Group 4	> 598.74	> 1320			Any Airspeed	Any Airspeed
Group 5			> 5486.40 MSL	> 18000 MSL		
Source: U.S. Department of Defense (2009)						
Notes:	1. If UAS shares characteristics in multiple categories, the UAS will be classified by the higher category, where Group 1 represents the lowest and Group 5 represents the highest (U.S. Joint Unmanned Aircraft Systems Center of Excellence 2010).					
	2. Small Unmanned Aircraft Systems (sUAS) includes Group 1, 2 and 3 categories (U.S. Department of Defense Joint C-sUAS Office 2020).					
Legend:	AGL: Above Ground Level		MSL: Mean Sea Level			

Table 1.1: Unmanned Aircraft System (UAS) Categorisation According to U.S. Department of Defense (DoD)

1.1.2 Rise of Multi-Rotor small Unmanned Aircraft Systems (sUAS)

Multi-rotor small unmanned aircraft systems (sUAS), which comprise the Group 1 to Group 3 UAS, has observed expeditious proliferation over the past decade in the civilian sector. The appeal of these systems is founded on its rotary-wing design, where propellers are used to facilitate vertical take-off and landing (VTOL); enabling omnidirectional movement, hovering for stationary flight, and the flexibility to manoeuvre unhindered within low airspace compared to other UAS types. At a basic level, on-board camera sensors are typically equipped as payloads to provide real-time visual feedback to enable piloting, as well as serve secondary aims such as imaging, data acquisition and monitoring. The capacity to lift payloads further allows other sensor packages to be employed, or for tasks beyond remote sensing with attached equipment, which grants multipurpose applications that address to an array of needs encompassing consumer (i.e. personal use and hobbyist) and commercial (i.e. professional, industrial and research) spheres (Camilli 2015; Kovalev, Voroshilova & Karaseva 2019; Heiets et al. 2023) – refer to Figure 1.2.

Considering these capabilities, multi-rotor sUAS holds a dominant market share compared to other UAS types, which is further driven by primarily off-the-shelf (OTS), ready-to-fly (RTF) systems offered by manufacturers, which are affordable, user/beginner-friendly, and readily available from stores and online vendors (Vargas-Ramírez & Paneque-Gálvez 2019; Garg 2022). Additionally, consistent engineering innovations in terms of hardware (i.e. polymer and composite materials with high-strength-to-weight ratios, integrated electronics, battery characteristics), firmware (i.e. flight controller performance) as well as software, all combine to foster adoption (Floreano & Wood 2015; Yang et al. 2017; Sala 2021), with improvements to flight characteristics (i.e. speed, range, altitude, endurance, payload capacity), as well as durability, on-board sensors, navigation and control, communication technologies, autonomous and user features, operability in adverse weather and low-light conditions etc.

The secondary root of popularity has stemmed from 'do-it-yourself' (DIY) multi-rotor sUAS, which are platforms modified or configured from a software, firmware and hardware level for various degrees of customisation, made by enthusiasts, researchers and secondary manufacturers (Camilli 2015; Rao, Gopi & Maione 2016; Florence et al. 2018) – refer to Figure 1.2. Systems range from altered OTS units, partially prebuilt systems requiring the purchase and or assembly of additional components (i.e. Almost Ready to Fly (ARF), Bind-and-Fly (BNF), Plug-and-Play (PNP)), kits requiring complete assembly of included components, or units built end-to-end with individually selected components (Verup & Olin 2016; Pereira et al. 2020; Motion RC 2021; Liang 2023). This has been achieved with the large availability of cheap consumer electronic components on the market, made possible by the culmination of electronic miniaturisation, manufacturing efficiencies, with high-volume production, leading to manufacturing cost reductions (a trend that also translated to OTS multi-rotor sUAS production) (Qin et al. 2010; Floreano & Wood 2015; Maghazei, Lewis & Netland 2022). The making of DIY multi-rotor sUAS can be quite involved, however abundant support is present for all stages of the development cycle to create units ranging from lower sophistication to higher performance and technical complexity – although these are overall less advanced than their OTS counterparts. Makers are able to actualise these systems with the assistance of books and manuals, as well as online resources, including tutorials, forums and social media (Radharamanan et al. 2016; Lobo et al. 2021). Furthermore, open-source, computer aided design (CAD) models and code shared by the DIY community online, contribute to a collaborative design process, which can accelerate development times (Rao, Gopi & Maione 2016; Cummings et al. 2017; Lobo et al. 2021). Moreover, this process coupled with 3D printing (additive manufacturing) permits rapid prototyping, assisting in time and fabrication cost savings for complex and custom parts, not achievable by conventional fabrication means. Lastly, the diverse possibilities of 3D prints can easily facilitate the attachment of payload equipment, similar to OTS systems, or even accommodate more unconventional additions to extend system capabilities (Lacher et al. 2019).

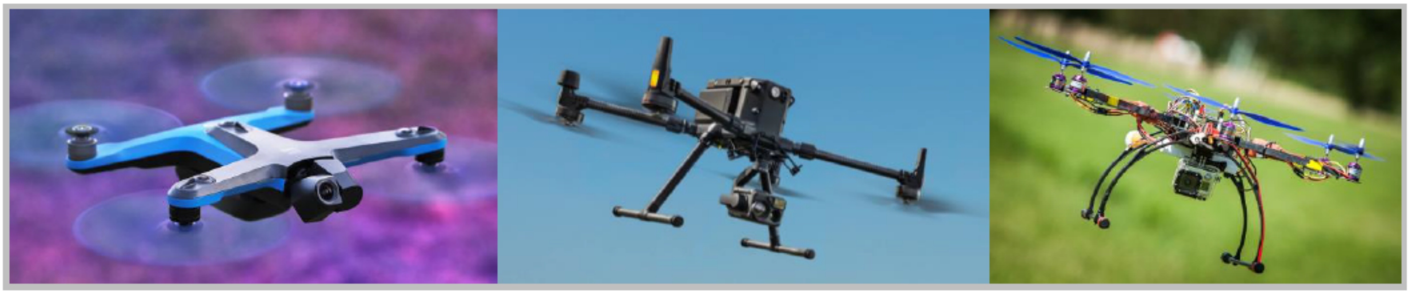


Figure 1.2: Skydio *Skydio 2* Consumer Grade Multi-Rotor sUAS (Pavic 2019) [Left Image] | DJI (Shenzen Dà-Jiāng Innovations Science and Technology Co., Ltd.) *Matrice 300 RTK* Commercial Grade Multi-Rotor sUAS (DJI 2024) [Middle Image] | DIY Multi-Rotor sUAS (Moss 2015) [Right Image]

1.2 Background Information

1.2.1 Multi-Rotor sUAS Threat

The rapid uptake of multi-rotor sUAS technology within the civil domain has prompted various countries to similarly establish or expand existing legislations (laws), regulations (rules) and guidelines concerning all types of sUAS. Considering the underpinning capabilities for aerial mobility, imaging and other remote sensing, these serve to direct operator actions and responsibilities in the interest of minimising the associated risks and hazards from sUAS activities, chiefly addressing important issues of safety, privacy, and acts of trespass that can impact the public and aviation community.

Regulations relating to safety, have predominantly been implemented at national levels, generally stipulating acceptable usage and requirements for sUAS operation including, registration and or use by approved persons, mandatory control by VLOS, limitations on using systems in proximity to people and populous areas, flights only within day-light hours and ideal weather conditions, flight altitude constraints, as well as operability within or near managed airspaces that have been designated under a national airspace architecture, such as (AirSight 2019; Australian Government 2024; U.K. Department of Transport 2024):

- **Controlled Airspace:** Airspace classes of prescribed dimensions, where aircraft movement (commercial, private and or military) is actively supervised by air traffic controllers (ATC) (Federal Aviation Administration 2016; U.K. Department of Transport 2017). Processes are used to ensure the orderly flow of aircraft in/out of these classes to mitigate the potential of mid-air collisions. Controlled airspaces are located above airports and surrounding areas, aircraft approach/departure paths and enroute airspace (Airservices Australia 2023; Australian Government 2024).

- **Restricted / Prohibited Airspace / No-Fly Zone:** Designated areas prohibiting sUAS activities; reserved for military bases, military operations areas (MOA), large public events, areas of emergency response and public safety operations, correctional facilities and critical infrastructure (Federal Aviation Administration 2016; AirSight 2019; Airservices Australia 2023; U.K. Civil Aviation Authority 2024).

Laws and guidelines drawing from existing privacy, surveillance and trespass legislations and regulations introduced at both the national and or subnational levels, have delineated unacceptable conduct, in particular, the infringement of privacy by using sUAS to view / listen / record / transmit, visual and or audio of an individual(s) without their consent, as well as offences of trespass, committed by the unauthorised flight of sUAS over private property (Burger & Scuderi 2018; AirSight 2019; Government of Canada 2021; Department of Infrastructure 2024).

These legislations, regulations and guidelines stand to harmonise sUAS operator conduct, allowing these systems to provide continued benefit and enrichment to society. However, intentional or unintentional breaches and unsafe use have the substantive power to endanger, disrupt and harm lives. Multi-rotor sUAS further exacerbate these issues with versatile capabilities and has resultantly opened 'Pandora's Box' by the onset of a worldwide threat environment, the magnitude of which has not been fully grasped by the wider public. Adapted work by Lacher et al. (2019) from *The MITRE Corporation* clearly characterises the threat posed by the intent of multi-rotor sUAS operators by way of a spectrum, with levels scaling in severity from inadvertent to malicious, illustrated in Figure 1.3.

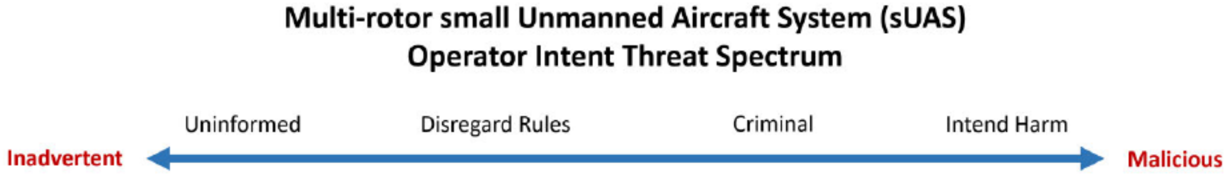

Multi-rotor small Unmanned Aircraft System (sUAS)
Operator Intent Threat Spectrum
Uninformed
Disregard Rules
Criminal
Intend Harm
Inadvertent
Malicious

Figure 1.3: Multi-rotor small Unmanned Aircraft System (sUAS) Operator Intent Threat Spectrum (Adapted from Lacher et al. 2019 (*The MITRE Corporation*))

The details provided below give further insight and context into the operator intent threat spectrum.

Operator Intent Threat: **Uninformed**

Any violation of national/subnational, legislations, regulations and or guidelines is accidental, resultant of end-user ignorance and or incompetence in flying a system (Lacher et al. 2019).

Operator Intent Threat: **Disregard Rules**

Operator is intentionally uncooperative with national/subnational legislations, regulations and or guidelines, or controls a system in a reckless (careless) manner - however, actions do not carry criminal intent.

Operator Intent Threat: **Criminal**

Operator wilfully chooses not to comply with national/subnational, legislations, regulations and or guidelines, with full understanding of consequences - however, actions do not intend to cause harm. Other threats include, negligent operation, harassment, intimidation, engaging in criminal operations (i.e. drug trafficking, smuggling, etc.), causing nuisance, endangering safety, etc (Department of Infrastructure 2024).

Notable Case: In 2018, between the 19th and 21st of December, repeated incursions into London Gatwick Airport controlled airspace by two multi-rotor sUAS (of unknown model) caused a major safety and security risk, with the possibility of midair collision. This led to suspension of air traffic during the incident, diverting or cancelling approximately 1000 flights and impacting 140,000 passengers (reported figures vary) (Robinson, Boyle & Murphy-Bates 2018; PA Media 2019; Calder 2023).

Operator Intent Threat: **Intended Harm**

This represents the most severe threat to international peace and security, where an end-user(s) is motivated to use multi-rotor sUAS maliciously (Chavez & Swed 2020; Hunter-Perkins, Shiotani & Rogers 2023). Systems can be used as recoverable or expendable attack platforms, either flown in a manner that is dangerous, or equipped with armament payloads, to enable offensive operational capabilities for lethal attacks on people, vehicles and or positions (i.e. open-spaces, infrastructure, buildings, etc.); non-lethal attacks can also be achieved for physical, physiological, psychological harm and or destructive ends - termed as armed-small unmanned aircraft systems A-sUAS (Bone & Bolkcom 2003; Friese, Jenzen-Jones & Smallwood 2016; Narang 2019; Nichols et al. 2022; Rogers & Kunertova 2022; Danczuk 2023; Hunter-Perkins, Shiotani & Rogers 2023). Attacks can be achieved individually or collaboratively conducted by a lone actor (LA) or violent non-state actor (VNSA) defined as an individual or organisation operating independently of a nation-state, that are willing or actively use violence to forward an objective, including terrorist, insurgent, criminal groups, etc. (Hofmann & Schneckener 2011; Lacher et al. 2019; Narang 2019; Chavez & Swed 2020). VNSA and LA fundamentally develop A-sUAS through DIY means. Affordability and access to multi-rotor systems, armament payloads and or constituent components through a variety of possible acquisition and development channels grants a low barrier to entry to achieving air power (a capability once exclusive to nation states with conventional military aircraft), which has compounded the challenge for national authorities to minimise and prevent the substantial risks and threats associated with VNSA and LA use of A-sUAS (Pomerleau 2018; Chavez & Swed 2020; Hunter-Perkins, Shiotani & Rogers 2023). Although, documented attacks with multi-rotor systems have largely been limited to using explosive payloads (examples given below) (Bunker, Sullivan & Kuhn 2021), other types of armaments have also been demonstrated to be possible, including chemical, biological, radiological, nuclear, and small arms and light weapons (SALW), which can greatly extend the potency of a potential attack, and has been cause for heightened concerns from an ill-intentioned operator (Friese, Jenzen-Jones & Smallwood 2016; Kneen 2020; Nichols et al. 2022; Rogers & Kunertova 2022; Tin et al. 2023).

Notable Case: The first uses of multi-rotor A-sUAS by VNSA can be attributed to the Islamic State of Iraq and the Levant (ISIL) in 2016 (Rassler 2018; Wright & Jenzen-Jones 2018) with consumer grade DJI *Phantom* series multi-rotor drones, which underwent minor modifications with controllable actuators to drop improvised and conventional, low-yield explosive munitions (De Cubber 2019; Nichols et al. 2022).

Notable Case: On the 4th of August 2018, an assassination attempt was made on President Nicolás Maduro of the Bolivarian Republic of Venezuela during a military parade in Caracas. Two DJI *Matrice 600* multi-rotor sUAS each armed with approximately one kilogram of C-4 explosive (Plastic Explosive PE-4) were remotely detonated at the location, injuring seven national guard members (McLaughlin, Sterling & Pozzebon 2018; Li 2019).

1.2.2 Visual sUAS Detection

As a direct response to the serious and escalating risks to international privacy, safety, security and peace manifested by the threats of criminal and intended harm by a multi-rotor sUAS operator (and other sUAS types), academia, industry, law enforcement and militaries worldwide are contributing to research and development, or are actively deploying counter-small unmanned aircraft system (C-sUAS) systems. These include several hierarchical countermeasures for detection, tracking and neutralisation of sUAS threat(s) in an environment (Matthew Henderson 2020; Şen & Akarslan 2020; Gradient 2021; Apratim Sharma 2022).

SUAS detection systems serve as the foundation for C-sUAS operations, which use sensors for environmental perception and data capture (input data), to monitor for incoming sUAS in a section of airspace in real-time, as detailed in Figure 1.4:

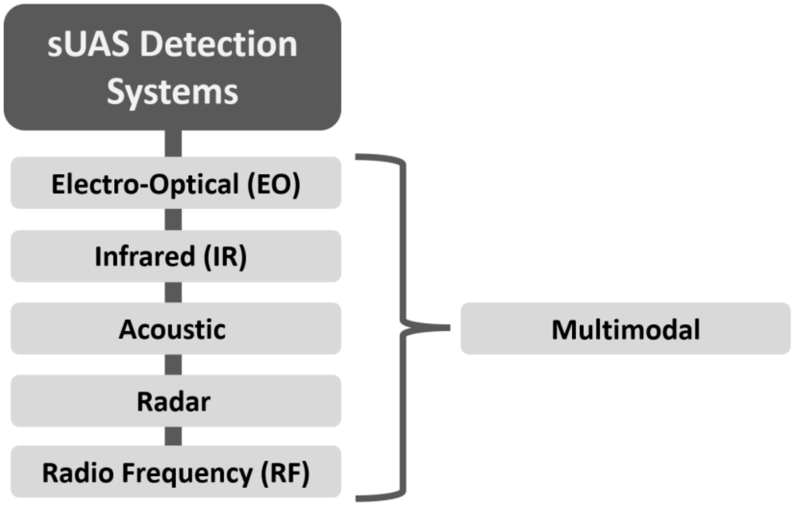


Figure 1.4: Small Unmanned Aircraft System (sUAS) Detection Systems (Taha & Shoufan 2019; Şen & Akarslan 2020)

Since the efficacy of each sUAS detection system is contingent on the respective advantages and disadvantages of its underlying medium of detection, no definitive detection solution exists, hence multimodal detection approaches area best (Apratim Sharma 2022; Zitar et al. 2023). However, visual sUAS detection systems, specifically electro-optical (EO) and associated technologies, have some of the strongest interest for research, development and deployment in field (aside from infrared (IR) detection systems). Reasoning for this is grounded in the fact that these systems utilise camera sensors for data acquisition of an environment within the visible spectrum (Elsayed et al. 2021; Park et al. 2021; AeroExpo 2024; Echodyne 2024), which holds numerous benefits, including lower associated costs, reduced regulatory limitations, large availability of sensor hardware, as well as ease of technology implementation and data analysis (Şen & Akarslan 2020; Park et al. 2021; Dieter, Weinmann & Brucherseifer 2023).

Computer vision (CV) models enable computer systems to interpret visual input data and extract key data features to execute tasks based on this information. Among these tasks is object detection, which involves the identification, location and classification of instance(s) of objects that are of interest within visual data (i.e. image, or a frame-by-frame within pre-recorded or real-time video sequences). In the case of EO detection systems, these have inbuilt computer systems to detect sUAS targets in real-time – information which is then further utilised by subsequent C-sUAS systems (Ziemba 2019). CV techniques have widely adopted deep learning (DL) considering enhanced capabilities in data feature extraction and pattern recognition that can improve performance (Kamilaris, van den Brink & Karatsiolis 2019; Xiao et al. 2020). Convolutional neural network (CNN), a type of DL architecture, is specifically utilised in CV applications due to suitability to analyse data with spatial relationships, such as visual data (Janiesch, Zschech & Heinrich 2021). Further to note is that CV models (and thereby CNNs) used for sUAS detection, are traditionally trained and validated using supervised learning (SL), which require annotated visual training datasets.

SUAS detection by EO systems typically uses a combination of binary classification and multi-class classification (Mrabet, Sliti & Ammar 2024):

- **SUAS Binary Classification:** Categorisation of sUAS and non-sUAS objects within visual input data into two mutually exclusive classes, typically by distinction with an expected and unexpected class, e.g., “Drone” and “No Drone” (Taha & Shoufan 2019; Brownlee 2020; Mandal & Bhattacharya 2020).

- **SUAS Multi-Class Classification:** Categorisation of sUAS and non-sUAS objects within visual input data into more than two mutually exclusive classes, typically to differentiate different types of sUAS and or specific sUAS characteristics, e.g., “Drone Type 1” vs “Drone Type 2” or “Drone with Payload” vs “Plane” vs “Bird” etc. (Brownlee 2020; Del Moral, Nowaczyk & Pashami 2022; Mrabet, Sliti & Ammar 2024).

1.3 Research Problem Definition

1.3.1 Synthetically Generated Data

***Note:** In CV, the domain is the environment and the elements that constitute the visual data.

- Source domain is the data distribution on which an algorithm is trained to perform a task.
- Target domain is the data distribution that a trained model is applied on, to perform a similar task.

Despite the successes of CNN for CV tasks, the current architectural limitations of DL impose inefficiencies that demand large volumes of annotated visual training datasets, where parameters of the source domain are varied to create a diverse image distribution, as would be expected in the real-world (Bengio, Goodfellow & Courville 2017). This facilitates the ability of a CV model to effectively generalise, which lends to increased robustness and accuracy when the model is applied to a testing dataset or when deployed for operations in real-world. In the case of training a multi-rotor sUAS detection model, according datasets require the following source domain parameters to be varied (Dieter, Weinmann & Brucherseifer 2023):

- **Environmental Conditions:** Weather (e.g. fog, rain, types of cloud cover, etc.), lighting conditions (e.g. natural light according to time of day and artificial light), occlusion, clutter within the background or foreground of an image with multi-rotor sUAS (Hsiao 2013; Gupta et al. 2024).

- **Target Object Spatial Attributes:** Position, pose, orientation, size of multi-rotor sUAS (Kashiyama, Sobue & Sekimoto 2020; Qi et al. 2024; Zheng et al. 2024).

- **Target Object Features:** Colour, configuration, type of multi-rotor sUAS (Dieter, Weinmann & Brucherseifer 2023).

- **Object Similarities:** Inclusion of objects similar in appearance to multi-rotor sUAS which may be difficult to visually discriminate, e.g. “Bird” vs “Drone” or “Drone Type 1” vs “Drone Type 2”.

- **Camera Sensor Properties:** Camera sensors are electro-mechanical units which capture imagery from the real-world. When light interacts with these units during image capture, visible artifacts and effects are introduced, e.g. Chromatic aberration, blur, image noise, lens flare, exposure levels, etc (Carlson et al. 2018; Hu et al. 2020).

- **Background Locations:** The background location where the multi-rotor sUAS is imaged.

Typically, CV image datasets for training require the manual collection of images in the real-world, which holds several disadvantages, that are particularly exacerbated for the collection of training imagery for multi-rotor sUAS detection considering the plethora of source domain parameters that should be varied during data acquisition. These disadvantages are outlined below:

Real-World Multi-Rotor sUAS Image Data Disadvantages

- Procurement of image data requires extensive manpower in terms of time and effort (Dieter, Weinmann & Brucherseifer 2023).

- Obtaining a sufficiently diverse distribution of visual data that captures multi-rotor sUAS, can be difficult or impossible, considering the aforementioned national and subnational, legislations, regulations and guidelines which impose limitations on sUAS operation, and can subsequently introduce inherent biases within the training datasets (Gastelum, Shead & Higgins 2020; Dieter et al. 2023)

- Publicly available datasets and purchasable datasets offered for multi-rotor sUAS detection are largely problem specific, insufficiently diverse, with some lacking the required quality (El Emam 2020).

- Outlier parameters are difficult to acquire in the real-world or reproduce frequently (Dieter, Weinmann & Brucherseifer 2023)

- Acquiring and gaining access to a wide selection of multi-rotor sUAS for dataset diversity can incur difficulties and costs.

The maturation of 3D computer graphics software tools, driven by continuous progress in supporting computer hardware such as central processing units (CPU) and graphics processing units (GPU) has enabled digital content creation (DCC) with the ability to create high-quality and realistic visuals for numerous fields including film, television, video games, architecture, advertising, digital art, etc. This has resultantly posited research interest in harnessing these tools to synthetically generate data (images) as a substitute for real-world data required for CV, offering a promising solution for the challenges of real-world data collection.

Synthetic Image Data Advantages

- Capability to rapidly and efficiently generate synthetic imagery with reduced time, effort and cost.

- Capability to design and develop virtual environments where parameters of the source domain can be sufficiently varied to generate a diverse image distribution. Furthermore, this can circumvent real-world restrictions regarding image acquisition and mitigate incurred biases within the training datasets.

- Can control source domain parameters to curate generated imagery for a specific task and ensure quality control.

- Outlier parameters which may be difficult to reproduce frequently in the real-world can be generated with ease for synthetic imagery.

1.3.2 Synthetic-to-Reality Gap

Despite being a logical idea to train CV models with only synthetic data, the literature has overwhelmingly demonstrated severe performance losses when deployed in the real-world (Nikolenko 2021; de Melo et al. 2022). This is fundamentally attributed to the domain gap or synthetic-to-reality gap (Tobin et al. 2017; Reway et al. 2020), which is the discrepancy between the source domain (the data distribution on which an algorithm is trained to perform a task) and the target domain (the data distribution that a trained model is applied on, to perform a similar task) (Kundu 2022; Doan et al. 2024).

The works of Sadeghi and Levine (2016), Tobin et al. (2017), and Tremblay et al. (2018), have been heavily influential in prevailing research involving the training of CV models using synthetically generated data, establishing that that the randomisation of source domain parameters (domain randomisation) within rendered synthetic training images is a major contributor to CV model generalisation and robustness which translates to improved performance when applied for real-world tasks. Therefore, subsequent works in this field have overlooked photorealism of synthetic training data, with the assumption that it is unnecessary. However, conclusions from the work of Nikolenko (2021) which provides the most comprehensive survey of works surrounding the application of synthetic data for CV, states that the findings of additional literature works concerning the level of photorealism required for synthetic training datasets and its impact on CV model performance is contradictory within and therefore remains an open question.

Since a domain randomisation strategy applied to synthetic training data are not a complete solution to improving CV model performance in the real-world, it is indicative of a remaining domain gap and brings into question if this can be further closed with additional strategies to achieve highly accurate CV model performance. An in-depth literature review involving the works of Sadeghi and Levine (2016), Tobin et al. (2017), and Tremblay et al. (2018) in this research paper reveals that these previous studies have not completely explored the efficacy of photorealistic synthetic training data, with discovered experimental biases. Furthermore, it is important to note that since the writing of these papers, the quality of photorealism that is achievable with rendering engines has drastically improved. Therefore this aligns with the understanding of work by Nikolenko (2021), that the impact of photorealism for synthetic training data is yet to be resolved, and furthermore and its potential to bridge the synthetic-to-reality gap.

1.3.2 The Case for Unreal Engine 5

Unreal Engine is a software platform (engine) by *Epic Games* for 3D game development, originally intended to accelerate the game development process. The main draw of this platform was its Level Editor, which is the Unreal Engine development environment which has a suite of interfaceable settings, tools, plugins and programmable features that permit creative control over a design of a virtual world environment in real-time, as well as engine customisation. Since the initial release of Unreal Engine 1 in 1998, generational improvements have been made to the engine, with each making strides to push the bounds of the platform for (1) wider creative control by developers, providing an anecdotally user-friendly platform with the capability to rapidly develop 3D virtual environments manually, and (2) taking advantage of improvements to computer systems to achieve state-of-the-art rendering of computer graphics, in order to deliver highly immersive gaming experiences. However, utility of Unreal Engine has expanded beyond gaming with the introduction of settings and tools suited for DCC (Buecheler 2001; Lightbown 2018; Jensen 2023; Epic Games 2024au).

The release of the latest generation of the platform, Unreal Engine 5 (UE5) in 2021 made significant innovations in terms of rendering capability, with high-fidelity photorealism achievable for DCC. This is made possible with *Lumen*, a new lighting system by UE5, which is a global illumination (GI) and reflections system, that simulates accurate lighting, soft shadows and physics-based material properties. Although GI is available for many rendering engines, *Lumen* distinguishes itself as the first production-ready, real-time, dynamic GI and reflection system. Dynamic GI and reflections compute the intensity of light reflected from a point on object surfaces within a virtual environment in real-time, which facilitates instant and adaptive lighting changes during rendering, for adjustments made to a scene, for out-of-the-box' results during development of a virtual environment. Complementing the Lumen system is the capability of ray traced shadow high-fidelity simulation of hard shadows. The results of UE5 rendering capabilities are demonstrable in Figure 1.5.

Additionally *Epic Games* provides support for Unreal Engine Marketplace, which is an online store with free and purchasable content made by third-party developers, which can be downloaded and imported into the engine to supplement the design and development of the virtual world, as well as reduce project development times. Consequently, these considerations have made UE5 an ideal choice for synthetic image data generation over other existing rendering engines and robotic simulation environments for CV-based problems, including multi-rotor sUAS detection (Epic Games 2023, 2024av).

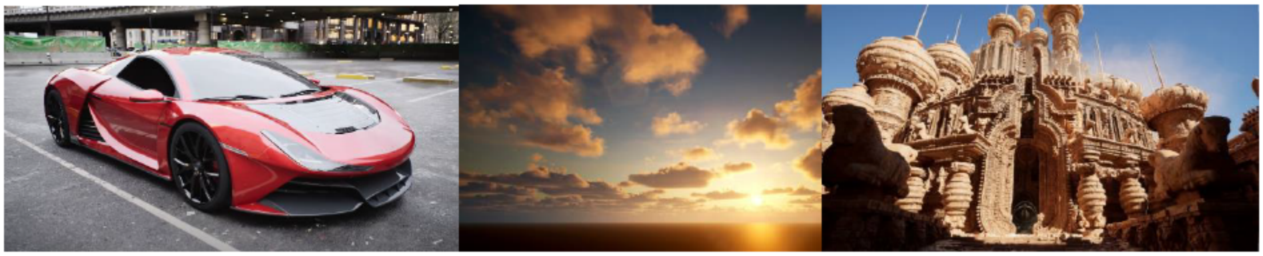


Figure 1.5: Unreal Engine 5 Rendering Results: Car (JSFILMZ 2023) [Left Image] | Cloud Cover (Gunther 2024) [Middle Image] | Fantasy Building (Poryagin 2024) [Right Image]

1.5 Research Questions

Considering the level of photorealism capable of being provided by UE5 for synthetically generated data and the lack of available research concerning its potential impact on performance for real-world CV applications, this warrants a research investigation, particularly for a critical application such as multi-rotor sUAS detection. The following core question will guide the direction for this research paper:

- Can high-fidelity photorealism provided by Unreal Engine 5 bridge the synthetic-to-reality gap for improved multi-rotor drone detection?

1.5 Expected Outcomes

The contribution of this experimental research intends to gain a preliminary understanding of the impact of high-fidelity photorealism in synthetically generated training data using UE5, which is used to train multi-rotor sUAS detection CV models. These results will contribute to ongoing C-sUAS research and serve to inform future research initiatives of the potential advantages that renders using UE5 can offer, not only for visual multi-rotor sUAS detection, but also other CV problems.

1.6 Thesis Outline

This research paper is structured into the following chapters outlined as follows:

Chapter 1: Introduction	Provides to the reader the project background context, background information, problem definition and stated research questions.
Chapter 2: Literature Review	Relevant background theory, information and past research will be explored in a literature review. This understanding will be used to guide the research methodology of this report.
Chapter 3: Methodology	Clear explanation of the methodology undertaken when performing the experimental investigation, in service of stated research aim and objectives.
Chapter 4: Results and Discussion	Experimental investigation results are presented with critical assessment of experimental investigation results in the context of the reviewed literature, which will be used to provide new insights and a solution to the defined problem as well as research questions. Experimental limitations and sources of error will be discussed.
Chapter 6: Conclusion	Summarisation of key findings and contributions of research project to answer the stated research questions, as well as suggests future directions for research.

Chapter 2 – Literature Review

This chapter will conduct a literature review and intends to complete a comprehensive overview of relevant and credible research available as of writing this paper in 2024, to provide insight into established works in terms of their contributions to theory, achievements and limitations. This will ultimately facilitate the identification of gaps in knowledge, which will develop objectives and methodology of this project, in service of the stated aim.

2.1 Background Information

2.1.1 Object Detection Theory

Considering the object detection involves highly technical concepts and processes that will be explored in this literature review, a brief explanation of the theory will initially be provided to offer context to the novice reader, on concepts related to the research.

A) Artificial Intelligence (AI)

AI is a technology that grants computer systems the ability to perform a range of cognitive tasks exhibited by human intelligence, such as learning, problem-solving, pattern recognition, creativity, etc, in both familiar and novel domains (Chen, Chen & Lin 2020; Du-Harpur et al. 2020). By drawing upon increases in computational power, AI has the capacity to process high dimensional data, at high volumes and frequencies, which includes structured data, categorised as quantitative data (i.e. time-series, tabular etc.) and unstructured, categorised as qualitative data (i.e. image, video, audio, text, etc.) (Peng et al. 2021; Leveau 2023; International Business Machines Corporation 2024; SAS Institute 2024). This has empowered AI to reliably and accurately exceed human capabilities in applied tasks, which has resultantly spawned various fields of AI (i.e. computer vision (CV), natural language processing (NLP), generative AI, etc.) that have been to serve numerous applications.

B) Machine Learning (ML)

ML is the subset and manifestation of AI – refer to Figure 2.1. It utilises a set of mathematical and software-based computational processes termed as an algorithm, to experientially learn from data, which it has been exposed to (training), by recognising meaningful features, such as patterns, as well as linear and non-linear relationships (Janiesch, Zschech & Heinrich 2021). Various types approaches of learning (paradigms) for ML have been developed, which utilise different types of algorithms that are specialised in performing certain tasks, generally selected based on the characteristics of the data and its application (Bengio, Goodfellow & Courville 2017; Mahesh 2020; Liang et al. 2022).

Figure 2.2 uses SL, which is a ML paradigm, as an example to present a hierarchical overview of the relationship between a ML paradigm, tasks, and algorithms. It should be noted that details of the algorithms/models displayed are beyond the scope of this report.

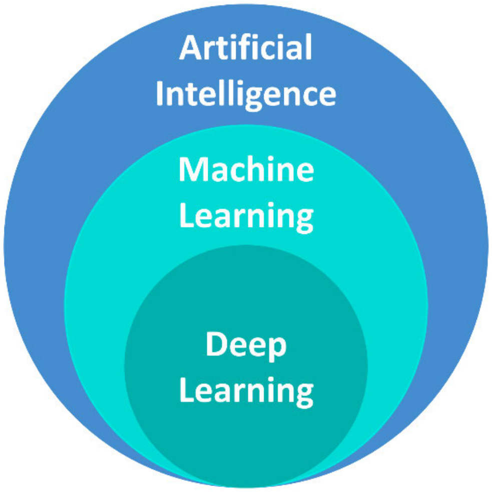


Figure 2.1: Artificial Intelligence (AI), Machine Learning (ML) & Deep Learning (DL) Venn Diagram

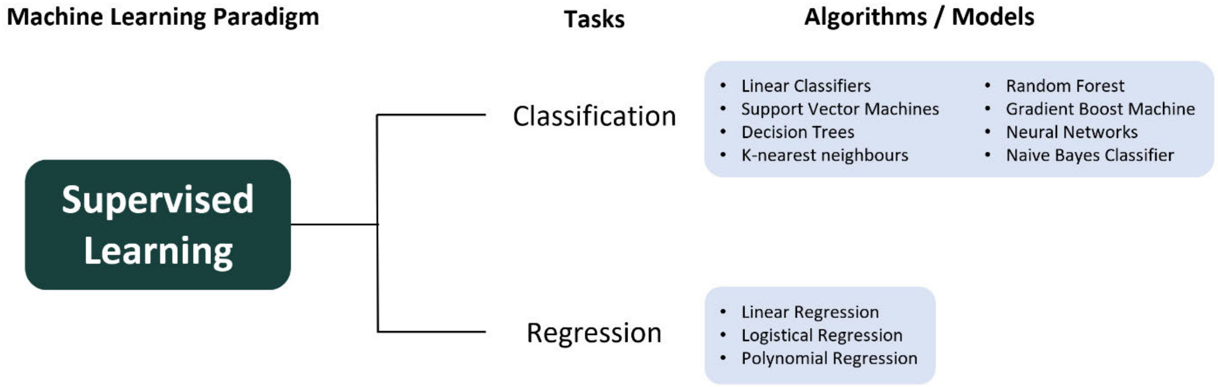


Figure 2.2: Supervised Learning Hierarchy Overview (Adapted from Castañé et al. 2021)

After the training process, the ML algorithm is termed as a ML model, which leverages the information it has learnt, to carry out an intended operation, which can range from visual data processing, mechanical design and optimisation, language translation to data analytics.

B.1) Supervised Learning (SL)

SL is a type of ML paradigm, where algorithms learn from human-driven guidance, and perform tasks of classification with categorical data, and regression with continuous data (refer to Figure 2.2), either separately or simultaneously, for predictions in data categorisation, and outcome estimation, respectively (Bengio, Goodfellow & Courville 2017; Tatsat, Puri & Lookabaugh 2020; Peng et al. 2021).

A training dataset is initially provided to an algorithm, consisting of an input-output pair that an algorithm is intended to learn from (Jung & Pedram 2010; Bernard 2021). This pair consists of *training input data*, X_{train}, is a set of raw data that provides the base information, as well as *target output data*, Y_{target}, is a labelled version of the corresponding raw data (ground truth), where critical aspects of data pertaining to the task are identified with an informative descriptor (Jung & Pedram 2010; Sarker 2021a; MathWorks 2024a). Data labelling is completed in a procedure known as data annotation, facilitated by software tools done either manually by a human supervisor or by an automated process that is reviewed by a human for accuracy.

During a SL training phase, a training dataset allows an algorithm to learn to map (associate) *training input data*, X_{train}, with the correct label of the *target output data*, Y_{target}, in terms of key data features (Sarker 2021b; MathWorks 2024a). This is generally achieved by the iterative optimisation of *parameters*, which are internal variables within ML algorithms, that are adjusted during training to influence the accuracy of the algorithm output. In the case of SL, outputs are classification and regression predictions (Hossain & Timmer 2021; Petru Potrimba 2023). Ultimately this training/learning process aims to determine a sound *mapping function*, $Y = f(X)$, which results in a trained model, that approximates the underlying relationship between input data and output data labels of the training data (Cunningham, Cord & Delany 2008; Mohamed 2017; Brownlee 2019). The *mapping function*, $Y = f(X)$, essential facilitates generalisation, which is the ability of the model to accurately predict an *output*, Y, when deployed on new and unseen *input data*, X, similar in distribution to the training dataset. Model generalisation can be evaluated using types of metrics that quantify model performance, which generally differ for ML tasks.

B.2) Supervised Classification

Supervised classification are tasks, where a ML model attempts to correctly categorise unseen input data into distinct types/classes that correspond to predefined data labels from the training dataset. There are four major types of classification tasks, binary classification, multi-class classification, multi-label classification and imbalanced classification (Alnuaimi & Albaldawi 2024). The former two are the most relevant in understanding the work of this report, defined as follows:

- **Binary Classification:** Task of categorising input data into two mutually exclusive classes. This can be deployed for applications seeking to distinguish input data with two specific classes, e.g. “Dog Bark” and “Cat Meow”; or are used in applications seeking to differentiate input data with an expected and unexpected class, e.g., “Drone” and “No Drone” (Brownlee 2020; Mandal & Bhattacharya 2020).

- **Multi-Class Classification:** Task of categorising input data into more than two mutually exclusive classes, e.g., “Drone”, “Bird”, “Plane” etc. (Brownlee 2020; Del Moral, Nowaczyk & Pashami 2022).

C) Deep Learning (DL)

DL is a subset and evolution of ML – refer to Figure 2.1. It utilises different computational algorithms (computer programs that simulate complex systems based on mathematics) based on artificial neural networks (ANN) to experientially learn from data, each suited to a certain tasks and applications. ANN are roughly inspired by the structure of a biological brain, comprised of interconnected artificial neurons (nodes), which are individual data processing units (mathematical functions), where connections are made node-to-node, alike synaptic links between neurons in the brain (Alaloul & Qureshi 2020; Montesinos López, Montesinos López & Crossa 2022), This facilitates data transfer/information flow throughout an

ANN (Janiesch, Zschech & Heinrich 2021), which is a critical enabler for the DL learning process and trained model function.

Compared to ML algorithms, DL algorithms exhibit improved learning performance with greater volumes of information/input data, and thereby are more computationally intensive, necessitating specialised hardware accelerators such as Graphics Processing Units (GPU), Tensor Processing Units (TPU) or Data Processing Units (DPU) to improve the efficiency of learning and execution of tasks and applications (Jeon et al. 2021; Mathew, Amudha & Sivakumari 2021; Yagawa & Oishi 2021); however, DL algorithms are more adept in determining patterns, non-linear relationships of higher complexity – particularly excelling in unstructured data (Najafabadi et al. 2015; Sarker 2021a). This is in part attributed to ANN using representation (feature) learning, which is a set of data processing methods that supports the extraction of low-level (simple/local) features from raw data (e.g. image pixel) and hierarchical feature abstraction, over multiple levels, to high-level (complex/global) representations/features (e.g. object within image); ultimately empowering an algorithm to better understand complex data during the learning phase, and become a capable DL model (LeCun, Bengio & Hinton 2015; Bengio, Goodfellow & Courville 2017).

It should be noted that similarly to ML, various paradigms for DL have been developed, including SL (a focus point within this research), which employ different types of algorithms. Algorithms that employ SL are popular, with leading results being currently achieved in classification and regression tasks (Karhunen, Raiko & Cho 2015; Wani et al. 2020). These algorithms can be further categorised according to the utilised architecture, which refers to various ANN designs, characterised by the topology, activation functions and paradigm (Kalogirou 2001; Srihari n.d.). For reference the taxonomy of DL can is visualised in Figure 2.3.

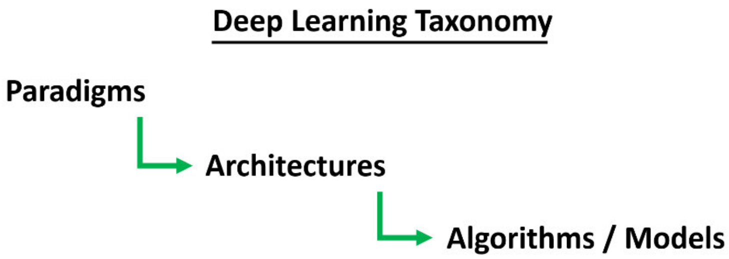


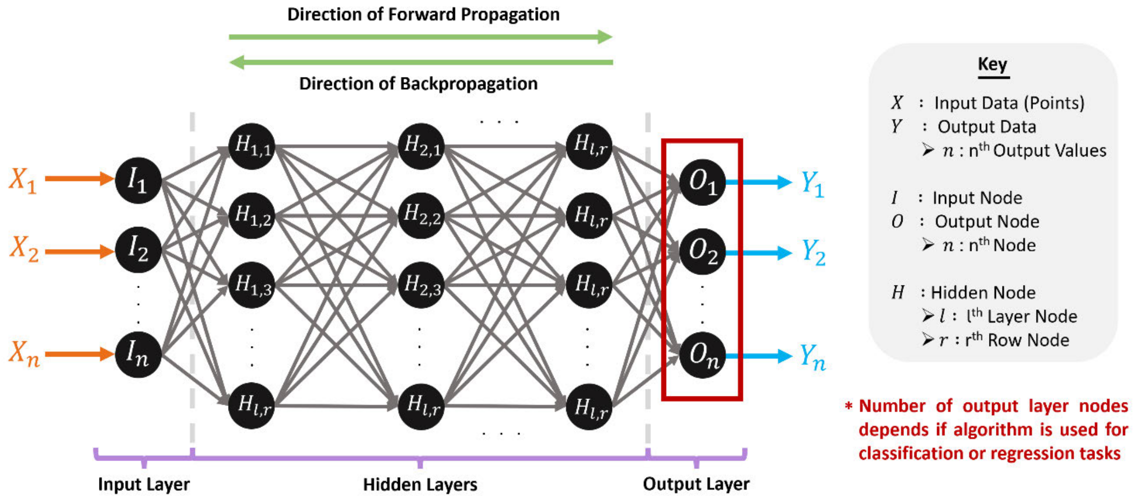
Figure 2.3: Deep Learning (DL) Taxonomy

D) Artificial Neural Network (ANN) Topology

ANN topologies for DL architectures differ based on neural framework, defined as the number of node layers and number of nodes per layer, in addition to the interconnection structure (network connectivity), defined as the way which nodes are connected (i.e. intralayer, interlayer, self-connection or supralayer). Additionally, each node-to-node connection can be unidirectional or multidirectional, which dictates the possible direction of information flow within an ANN. These hold a strong influence on respective DL algorithm capabilities, learning performance, suited tasks and applications (Fiesler & Beale 1996; Sammut & Webb 2011).

E) Feedforward Neural Network (FNN)

A FNN topology is one the simplest and widely used ANN topologies (a focus point within this research), which are configured in multiple interconnected layers, consisting sequentially of an input layer, hidden layers and output layer (Sammut & Webb 2011; Wang et al. 2015). These are typically labelled as a fully connected topology, however, can more specifically be termed as a fully interlayer connected topology, meaning all possible node-to-node connections are made between the layers (Fiesler & Beale 1996; Csáji 2001). All connections are unidirectional, to facilitate forward propagation during algorithm training and deployment as a model. This is where input information to the ANN is processed and passed through each layer sequentially (forward), with each layer hierarchically abstracting features of increasing complexity, by building on the features from the previous layer; the outcome of which generates a learnt output result (prediction), relevant to the task (Sazli 2006; Bengio, Goodfellow & Courville 2017; Badillo et al. 2020). For reference, a basic FNN is illustrated in Figure 2.4.



* Number of output layer nodes depends if algorithm is used for classification or regression tasks

Figure 2.4: Basic Feedforward Neural Network (FNN)

***Note:** ANN are classified as deep neural networks (DNN) when it is comprised of more than two hidden layers, otherwise it is known as a shallow neural network (Shrivastava 2022).

FNN operation algorithm training and model deployment will be summarised as follows:

E.1) FNN Algorithm under Training Operation

The training of FNN is primarily undertaken by a SL, which is exacted in the following stages (Vora & Yagnik 2014; Alauddin et al. 2022; Logunova 2023b):

- **Forward Propagation:** In this stage a sample of the *training input data*, X_{train}, makes a forward pass, by being fed to the input layer and propagated layer-by-layer through the hidden layers of the network, with the objective of abstracting data in this process to predict an *output*, Y, at the output layer – refer to Figure 2.4. Drivers for the abstraction process are made by the modifiable network *parameters* – *weights*, w, and *biases*, b, assigned to individual hidden nodes (values range between 0 and 1) (Sazli 2006; Bengio, Goodfellow & Courville 2017; Fetaya, Lucas & Andrews n.d.). Forward propagation facilitates the algorithm learning and predictions.

- **Backpropagation:** After forward propagation, the stage of backpropagation optimises the network *parameters* to minimise the difference (error) between the *predicted output*, Y and the *target output data*, Y_{target} (Pedrycz & Chen 2020; Sekhar & Meghana 2020). This *error*, $E\ (w, b)$, is initially computed using an *error function* (specific to the task or application) that takes into account all network *weights*, w, and *biases*, b (Alzubaidi et al. 2021; Nan, Du & Ibrahimli 2024). A backward pass is completed, where the *error*, $E\ (w, b)$, is fed to the output layer and propagated layer-by-layer through the network to the input layer (refer to Figure 2.4), during which the partial derivatives of the *error*, $E\ (w, b)$, with respect to each node *weight*, w, and *bias*, b is computed – refer to the according mathematical expressions below (Haykin 2009; Davila 2020; Pearson & Ginsburg 2022; Yen 2023; Nan, Du & Ibrahimli 2024):

  Partial derivative of the error with respect to node weight : $\frac{\partial E(w,b)}{\partial w_{kj}}$

  Partial derivative of the error with respect to node bias : $\frac{\partial E(w,b)}{\partial b_k}$

  Where: $E(w, b) = Error\ ,\ w_{kj}\ =\ Node\ Weight\ \ ,\ b_k =\ Node\ Bias$

  ***Note:** Subscript variables are to be explained further.

  Gradients are vectors of partial derivatives, oriented in the direction of steepest ascent, which yields the greatest rate of change at a *point*, p and indicate the direction in which the error will increase most quickly from that *point*, p. By this definition, the partial derivatives of the error, with respect to the node weight, $\frac{\partial E(w,b)}{\partial w_{kj}}$, and bias, $\frac{\partial E(w,b)}{\partial b_k}$, all together determine the *error gradient*, $\nabla E(w, b)$; which serves to quantify the incurred *error*, $E(w, b)$ from the current network *parameters,* as well as determined the direction along which current error will be further exacerbated (Joyce 2014; Antonio 2022).

  ***Note:** The unidirectionality of connections in a FNN is specific to the forward propagation of information; this does not apply to the backward propagation of errors.

- **Gradient Descent:** After backpropagation has been completed, a gradient descent algorithm is employed, which is responsible for minimising/optimising the *error*, $E(w, b)$, and enables the learning process. Several types of gradient descent algorithms are available; the basic implementation of this process is where updates are made iteratively to the node *weights*, w, and *biases*, b, to drive the *error gradient*, $\nabla E(w, b)$, in the opposite/negative direction, known as the direction of steepest descent. It should be noted that in usual training practice these *parameters* are initialised randomly (Wythoff 1993; Kvaal & McEwan 1996; Demuth, Beale & Hagan 2010; Ruder 2016). The general formula to update current network *parameters* with gradient descent, involves subtracting the product of the *learning rate*, α_k, and the corresponding *partial derivative* from each *weight*, w, and *bias*, b, as presented below (Demuth, Beale & Hagan 2010; Tapkir 2023; Yen 2023):

<u>General Weight Update Formula</u>

$$w_{kj+1} = w_{kj} - \alpha_k \left(\frac{\partial E(w,b)}{\partial w_{kj}}\right)$$

Where:

- $w_{kj+1} = Updated\ Weight$
- $w_{kj}\ = Current\ Weight$
- $b_{k+1} = Updated\ \ Bias$

<u>General Bias Update Formula</u>

$$b_{k+1} = b_k - \alpha_k \left(\frac{\partial E(w,b)}{\partial b_k} \right)$$

- $b_k = Current\ Bias$
- $E(w,b) = Error$
- $\alpha_k = Learning\ Rate$
- $\frac{\partial E(w,b)}{\partial w_{kj}} = Partial\ derivative\ of\ the\ error\ with\ respect\ to\ node\ weight$
- $\frac{\partial E(w,b)}{\partial b_k} = Partial\ derivative\ of\ the\ error\ with\ respect\ to\ node\ bias$

***Note:** Subscript variables are to be explained further.

Central to the gradient descent process is the *learning rate*, α_k, which defines the step size to be taken per an update of *parameters*, resulting in a new and *error gradient*, $\nabla E(w,b)$.

As part of the FNN training process, the forward propagation, backpropagation and gradient descent process is cyclically repeated to update the *weights*, *w*, and *biases*, *b*, of all nodes, as seen in Figure 2.5.

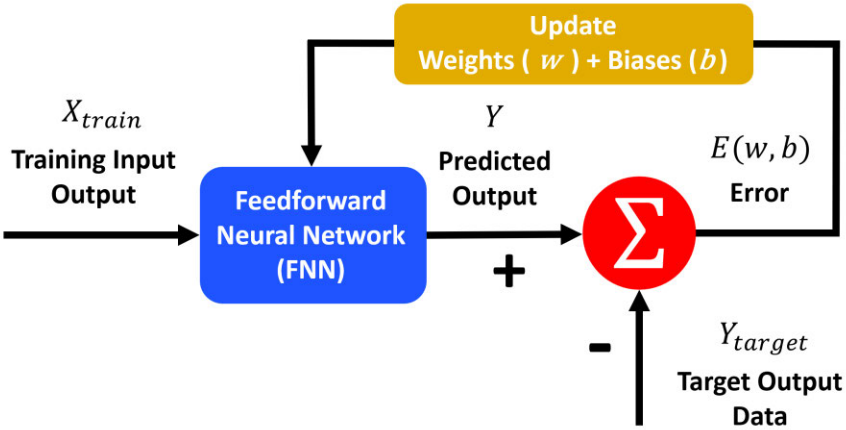


Figure 2.5: Feedforward Neural Network Training Process (Adapted from Werbos (1990))

The goal of the training process is to minimise the *error*, $E(w,b)$, the which is intended to guide the *error gradient*, $\nabla E(w,b)$, to eventually approximate a convergence point, ideally the global minimum, however a local minimum may be practically acceptable, as seen in Figure 2.6 (Wythoff 1993).

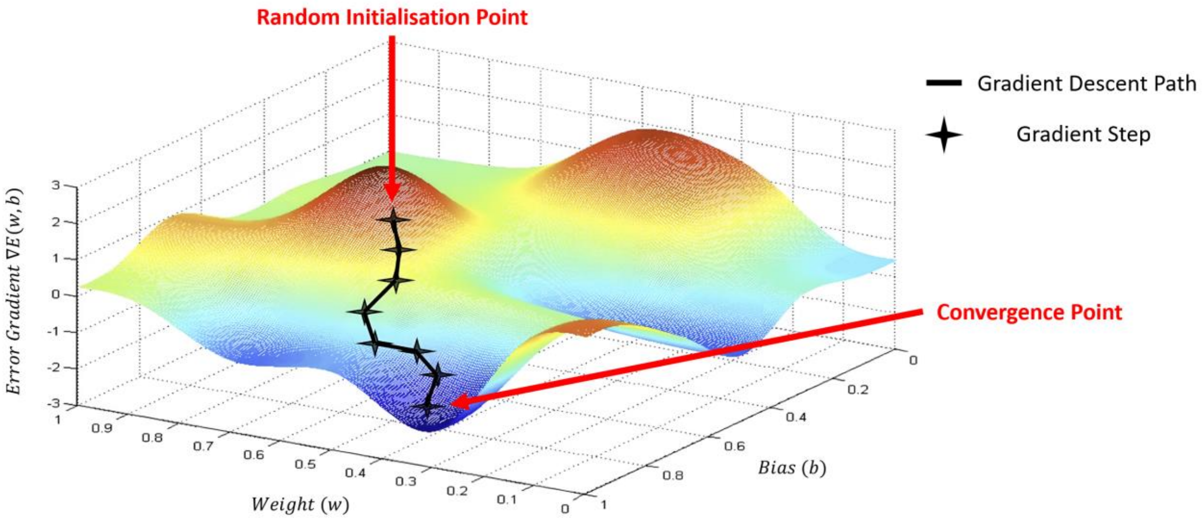


Figure 2.6: Deep Learning Gradient Descent Process (Adapted from Adejumo (2023))

E.2) Hyperparameters

Hyperparameters are modifiable external algorithm variables that are manually set by a human supervisor before the training process. These influence the response of internal algorithm parameters to provide additional degrees of freedom to indirectly assist in *error*, $E(w,b)$, minimisation and thereby the output performance of the final trained model. Hyperparameters are tuned and optimised by manual or automated processes (Hossain & Timmer 2021; Logunova 2023a; Arnold et al. 2024). Three key hyperparameters are of most concern and include:

- **Epoch:** Defines the number of times that the ANN processes the complete set of samples within a training dataset (Sazli 2006; Afaq & Rao 2020).

- **Batch:** Defines the number of training dataset samples processed by the ANN before the algorithm *parameters* are updated (Brownlee 2022).

- **Learning Rate ($\boldsymbol{\alpha_k}$):** Defines the step size to be taken per an update of *parameters*, and in effect how fast the algorithm learns. An excessively small *learning rate*, α_k, results in a slow and time-consuming *error*, $E(w,b)$ minimisation process – refer to Figure 2.7 left image. Alternatively an excessively large *learning rate*, α_k, can lead to an unsuccessful approximation of the convergence point, and or incur divergent behaviour to the detriment of the *error*, $E(w,b)$ minimisation process – refer to Figure 2.7 right image (Bhatnagar 2022; Sellat, Bisoy & Priyadarshini 2022; Zhao et al. 2024).

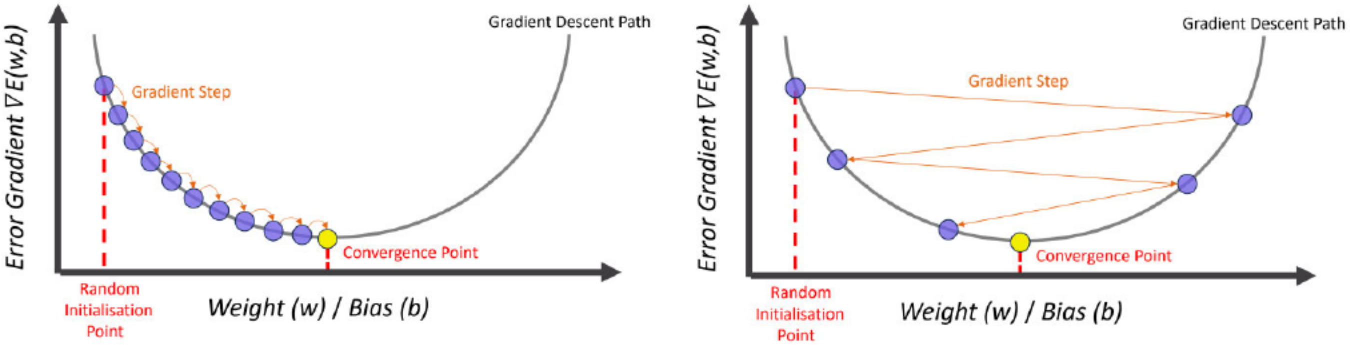


Figure 2.7: Gradient Descent Excessively Small Learning Rate [Left Image] vs Gradient Descent Excessively Large Learning Rate [Right Image] (Adapted from Zhao et al. (2024))

An appropriate and optimum *learning rate*, α_k, is critical to ensuring an ideal rate of convergence. Different *learning rate schedules* can be used by a supervisor to adjust the *learning rate*, α_k, during the training process – typically decreasing between epochs as seen in Figure 2.8 (Paperswithcode ; Zhang et al. 2023).

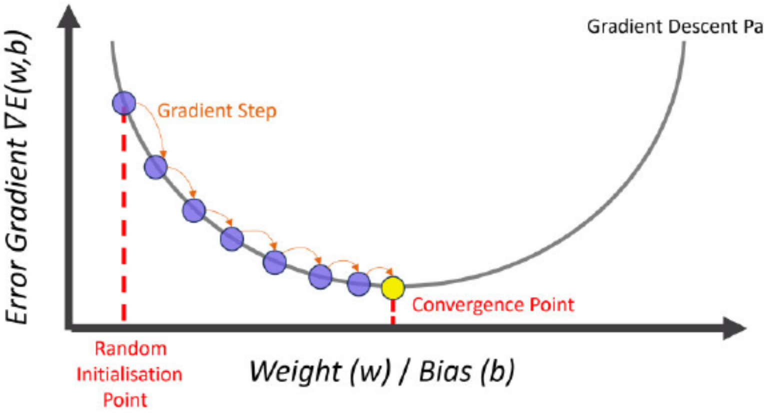


Figure 2.8: Gradient Descent Based-On Learning Rate Scheduling (Adapted from Zhao et al. (2024))

F) Training, Validation & Testing Datasets

As part of the development process for a SL based DL model, all input data samples are split into distinct datasets, utilized in service of the following purposes:

***Note:** Datasets are comprised of data samples.

Training Dataset

An DNN algorithm is repeatedly exposed to a training dataset over multiple epochs to learn meaningful features by recognition of the underlying patterns and relationships of the data; which is achieved by the procedural adjustment of internal *parameters* that directly guide the minimisation of the *error*, $E(w, b)$, between the *predicted output*, Y and the *target output data*, Y_{target} for the training dataset.

Validation Dataset

Commonly after each epoch of training, a validation dataset applied to the DNN algorithm for evaluation of its generalisation on unseen data (Deshpande 2024). This generalisability is quantified using a selected performance metric, which is then used to tune hyperparameters via manual or automated methods to indirectly assist in minimisation of the prediction *error*, $E(w, b)$, between the *output*, Y and the *target output data*, Y_{target} for the validation dataset (Logunova 2023a; Amazon Web Services 2024).

Testing Dataset

After the completion of training and validation processes, the now model is exposed to a testing dataset, which acts as a benchmark to evaluate its final performance, in terms of its generalisation, by the use of one or multiple performance metrics (Encord 2023a).

F.1) Recommended Data Splits

For model development, the following percentages of the total input data samples are recommended to be allocated for each dataset (data split):

- 70% Training Dataset, 10% Validation Dataset, 20% Testing Dataset
- 80% Training Dataset, 10% Validation Dataset, 10% Testing Dataset
- 60% Training Dataset, 20% Validation Dataset, 20% Testing Dataset

G) Convolutional Neural Networks (CNN)

CNN is a type of DL architecture that uses a FNN topology, and is currently the primary choice for computer vision related tasks, due to superior performance comparative to other DL architectures (Wani et al. 2020; Sarker 2021b).

H) Computer Vision (CV)

CV is a field of AI, which capacitates computer systems to interpret visual information, for the analysis and extraction of important features. Trained models can be used on a variety of input data, spanning different formats, including digital images and videos within or outside the visual spectrum, 3D data (Ioannidou et al. 2017; Guo et al. 2020; MathWorks 2024b). The section below outlines the range of tasks that CV models can be utilised for (Szeliski 2022):

Computer Vision Tasks		
Image Formation	Image Processing	Feature Detection and Matching
Segmentation	Feature-Based Alignment	Structure from Motion
Dense Motion Estimation	Image Stitching	Computational Photography
Stereo Correspondence	3D Reconstruction	Image-Based Rendering
	Object Recognition	

Although, traditional methods of CV have been in existence and continue to have relevance with certain CV field, the introduction of DL has significantly impacted the predictive performance of CV tasks and operations, with advantages including but not limited to: Greater accuracy with unstructured visual data, increased computational efficiency with processing of larger datasets, automated feature learning during training, ease of generalising for trained models, untrained algorithms can readily be adapted for completely new training data, outperformance in real-time applications, etc. (Voulodimos et al. 2018; O'Mahony et al. 2020).

I) Object Recognition

Object recognition is primarily a supervised CV task, that involves the identification of target object(s) within visual data, such as an image, or a frame-by-frame within pre-recorded or real-time video sequences (Zhu et al. 2020; Lazebnik 2021; Zou 2022; Tran, Kanaujia & Parameswaran 2023). Object recognition can be grouped into two types, outlined as follows:

***Note:** Instance refers to an occurrence of an object.

- **Generic Object Recognition:** Aim is to identify and categorise of instances of objects according to single or multiple predefined class labels (Liu et al. 2020; Khanday & Sofi 2021; Lazebnik 2021; Szeliski 2022).

- **Instance-Level Recognition:** Aim is to identify single instances of an object with unique features – e.g. human faces, car models, landmarks, etc (Askew & Araujo 2020; Liu et al. 2020; Khanday & Sofi 2021; Szeliski 2022).

J) Generic Object Detection

Generic object detection is a subtask of generic object recognition, where the *output*, Y, predicts the presence of target object(s) within unseen visual *input data*, X.

For the object detection, the input-output pairs for the datasets consist of raw image data of object(s) that provide the base information, as well as the corresponding target output image data (ground truth) which is labelled with a *ground-truth bounding box*, A. This is a rectangular spatial area that surrounds the object(s) of interest to be predicted within *input data*, X and are assigned with class label to indicate the object type(s).

During testing, validation and testing the *output*, Y, of the algorithm / model provides a *predicted bounding box*, B, to objects of interest.

Object detection is achieved by the following set of processes (Jain, Kasturi & Schunck 1995; Cao & Gu 2019; Zhao et al. 2019; Liu et al. 2020; Zou 2022):

- **Object Classification:** Aim is to identify the class(es) of instance(s) of target object(s) within a visual data according to predefined class labels that the model has been trained to recognise – refer to Figure 2.9.

- **Object Localisation:** Aim is to determine the coordinate location of instance(s) of target object(s) within visual data with a *predicted bounding box*, B – refer to Figure 2.9.

***Note:** Object detection can be used for tasks of binary classification and multi-class classification.

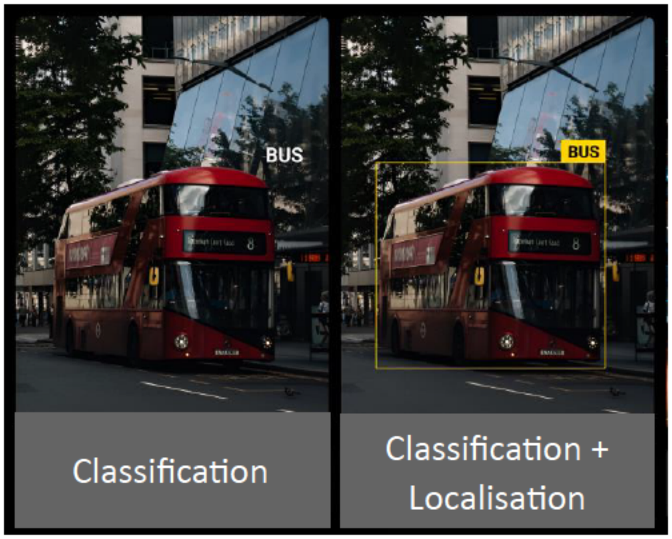


Figure 2.9: Illustration of Object Detection Processes: Object Classification and Object Localisation (Adapted from Kniazieva (2023))

K) Object Detection Model Performance Metrics

Generalisation is the ability of a trained model to perform well on unseen data. Performance metrics are quantitative measurements of this ability that appraise DL model performance, in terms of the how effectively a model can approximate the desired output (correctness) (Terven et al. 2023) and robustness, defined as the capacity of the model to maintain stable predictive performance despite variations in the input data (Braiek & Khomh 2024). Different metrics are specific to certain DL tasks, however, do share some in common as with the case of case of object detection, and its sub-tasks of binary classification and multi-class classification models (Terven et al. 2023).

This section will provide an overview of all performance metrics used to evaluate object detection models after training with a test dataset, as these will be relevant for the upcoming literature review.

***Note:** Conceptually some of these performance metrics can be used for validation datasets.

K.1) Intersection over Union (IoU)

This serves as the foundational metric upon which other performance metrics are built upon and measures the localisation accuracy, by how closely a *predicted bounding box*, B, made on the input data of the test dataset, matches the *ground-truth bounding box*, A, from the labelled target output data of the test dataset (Padilla et al. 2021) – refer to Figure 2.10.

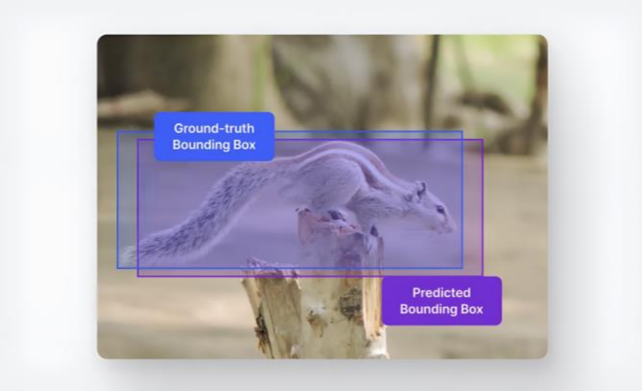


Figure 2.10: Illustration of Intersection over Union (IoU) (Shah 2024)

The match of these bounding boxes is quantified by its IoU, a value based on the Jaccard Index, which measures the similarity between the two datasets. IoU is calculated as the ratio of the overlap of the predicted bounding box and the ground truth box (intersection), to their combined areas (union) (Boesch 2024) – refer to Figure 2.11.

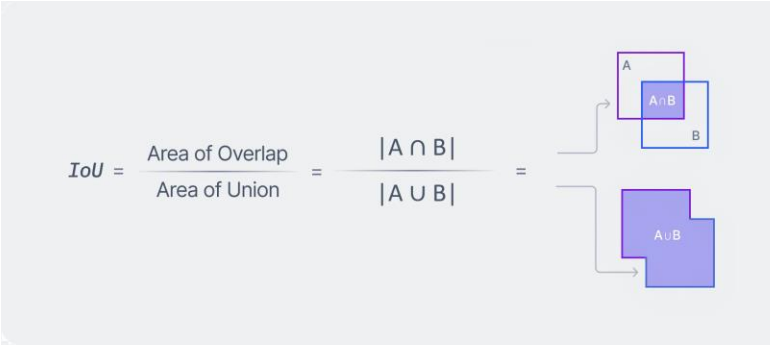


Figure 2.11: IoU Calculation (Adapted from Shah (2024))

IoU score limits range from 0 to 1, and indicate following (Gustafsson 2023):

- IoU = 1: Indicates that both bounding boxes completely overlap for a perfect object detection.
- IoU = 0: Indicates that both bounding boxes share no overlap for no object detection.

The qualification for detection is governed by a supervisor selected, overlap threshold, t, given as follows (Padilla, Netto & Da Silva 2020):

- IoU $\geq t$: When the threshold, t, is surpassed this is an indication of a correct detection.
- IoU $< t$: When the threshold, t, is not surpassed this is an indication of an incorrect / missed detection.

In practice the values of this threshold, t, can range from 0.5, 0.6, 0.75 or 0.95 (50%, 60%, 75% or 95%) (Riveros 2018).

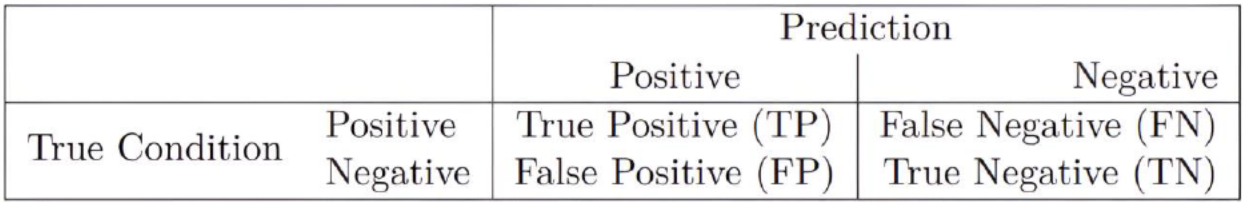
The level of correctness for detection is categorised as follows, and is used to determine other performance metrics (Padilla et al. 2021; Gustafsson 2023; Terven et al. 2023; Zhao 2024):

True Positive (TP) :	Number of correctly placed detections of a target object
False Positive (FP) :	Number of incorrectly placed detections of a target object with another / Incorrect detection of a non-existent target object
False Negative (FN) :	Number of incorrectly placed detections where existent target objects are not detected
True Negative (TN) :	Non-Applicable – since it means accounting the number of correctly placed detection of non-existent target object

***Note:** Positive indicates that the model has made a detection prediction, and negative means that the model has not made a detection prediction.

K.1) Confusion Matrix

A confusion matrix is a table used to illustrate the performance of an object detection model, by displaying the four (Breton 2019). An example confusion matrix can be observed in Figure 2.12.



		Prediction	
		Positive	Negative
True Condition	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 2.12: Example Confusion Matrix (Breton 2019)

K.2) Precision and Recall

Precision and recall are paired metrics presented in combination as they provide a biased view of the data separately (Gustafsson 2023).

Precision measures how many positive classifications made by the model that are actually correct, out of all the positive predictions made by the model, as given by the formula below (Google 2024; Jurafsky & Martin 2024):

$$Precision = \frac{TP}{TP + FP}$$

***Note:** Precision can also be given as a percentage

Precision is used when it is important for positive predictions to be accurate (Google 2024).

Recall measures how many positive classifications made by the model that are actually correct out of all the actual positive samples in the dataset, as given in the formula below (Google 2024; Jurafsky & Martin 2024):

$$Recall = \frac{TP}{TP + FN}$$

***Note:** Recall can also be given as a percentage

K.3) Precision-Recall Curve

The precision-recall curve is a plot of model precision and recall values. The precision-recall curve displays the trade-off between precision and recall for a selected threshold. The following conditions provide a better understanding to interpret a precision-recall curve (Scikit-Learn 2024):

- A model with low precision and high recall returns most of the relevant items, but the proportion of returned results that are incorrect labelled is high.

- A model with high precision and low recall returns few of the relevant items, but most predicted labels are correct.

- A model with high precision and high recall will return most of the relevant items, with most results labelled correctly (perfect classifier – refer to Figure 2.13).

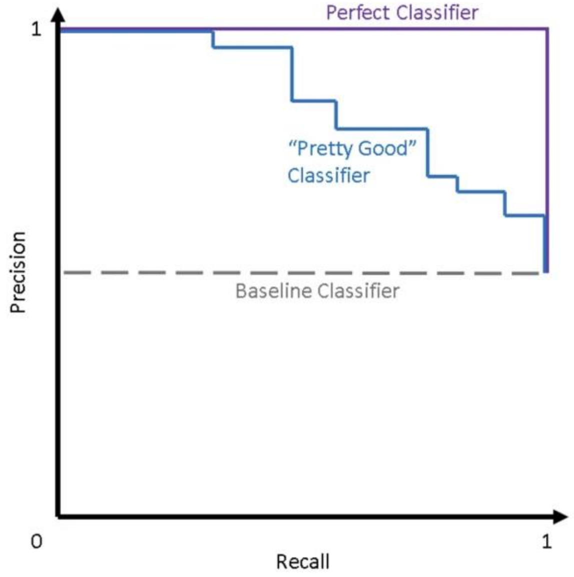


Figure 2.13: Theoretical Precision Recall Curves (Steen 2020)

K.4) Average Precision (AP)

AP is the value of the enclosed area under precision-recall curve and summarises it as the weighted mean of precisions achieved at each threshold (Shah 2022; Scikit-Learn 2024).

***Note:** AP can be given as a percentage

K.5) Mean Average Precision (mAP)

MAP is the premier metric used to evaluate object detection models and is the mean of the AP over all established classes, calculated as (Padilla et al. 2021; Shah 2022):

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$$

Where: $k = class, AP_k = the\ AP\ value\ \ for\ the\ k^{th}\ class, n = number\ of\ classes\ evaluated$

***Note:** mAP can be given as a percentage

K.6) F1 Score

F1 Score is a measure of balance between precision and recall, as the harmonic mean of precision and recall, to give a more stable performance evaluation (Encord 2023b). It is calculated as:

$$F1\ Score = 2\ \times \left(\frac{Precision\ \times Recall}{Precision + Recall}\right)$$

***Note:** F1 Score can also be given as a percentage

An F1 Score - Confidence curve is used to visualise the F1 score of a model over a range of confidence thresholds.

A high F1 score is indicative of well-balanced performance, demonstrating that the model can simultaneously attain high precision and high recall; whereas a low F1 score is shows the model has low recall and or precision (Encord 2023b).

An F1 score can also be paired with a confidence threshold to form a graph, as dependent and independent variable respectively). A confidence threshold is the minimum score that the model will consider the prediction to be a true positive. Predictions with a confidence score below a confidence threshold are ignored. A higher confidence threshold leads to fewer, but more reliable predictions, while a lower confidence threshold yields more predictions but with a higher likelihood of false positives (Edge AI and Vision Alliance 2024).

***Note:** Performance metrics are model-agnostic and therefore can be used for comparison of different models, or for comparison of the same model using different configurations (Terven et al. 2023), and therefore will be applied in the following literature review.

2.2 Literature Review

The following section presents the literature review that was conducted for this research report. Peer-reviewed research papers were examined for major concepts concerning synthetic data generation. Any assessment of non-peer reviewed research papers will be made known to the reader.

2.2.1 Synthetic Data Types

The generation of synthetic data does not entail a definitive method, but rather a set of approaches that have been developed via experimental practice (Akyon et al. 2021). The following list provides an overview of the primary synthetic data types and approaches for image dataset development, that have been demonstrated for CV applications as well as other vision-based DL fields including robotic vision (RV) and machine vision (MV) (Natarajan & Madden 2023):

- **Cut-and-Paste Synthetic Data:** Images are generated utilising methods of compositing.
    - Real-world images of target objects are cut and superimposed onto real-world images of a background (Dwibedi, Misra & Hebert 2017).
- **Hybrid Synthetic Data:** Datasets are developed / formed using a combination of synthetic data generated within selected rendering engine and real-world data.

  - **Type 1:** Image capture 3D model target objects placed in front of a 2D standard dynamic range (SDR) image or skybox, that acts as a background scene (Tremblay et al. 2018; Natarajan & Madden 2023).

  - **Type 2:** Datasets are formed using a mixed percentage of synthetic data and real-world data (Nowruzi et al. 2019).

    Type 3: Images capture 3D model of target objects under or in front of a real-world 360° (omnidirectional) image or panoramic image, which are projected as a background within the virtual environment (scene), as a dome, sphere, or skybox. The background images are high dynamic range (HDR) images that store real-world lighting information incident to the camera sensor, with a wider range of light, colour and subtle nuances of ambient light than a SDR image. In this consideration, these HDR images are used to provide image-based lighting (IBL), to illuminate a virtual environment with precise lighting detail from the real-world (Debevec 2004, 2006, 2008; Mantiuk, Myszkowski & Seidel 2015; Calian et al. 2018; Zhang, Jia & Ivrissimtzis 2020; Barisic, Petric & Bogdan 2022; Pennington 2024).

- **Pure Synthetic Data:** Images are generated only using synthetic data generated within selected rendering engine.

  - Images capture 3D model target objects under default environmental lighting provided by rendering engine.

Despite being a logical idea to train CV models with just photorealistic Pure Synthetic Data, the literature has overwhelmingly demonstrated severe performance losses when the deployed in the real-world (Nikolenko 2021; de Melo et al. 2022). This is fundamentally attributed to the domain gap or synthetic-to-reality gap (Tobin et al. 2017; Reway et al. 2020), which is the discrepancy between the source domain (the data distribution on which an algorithm is trained to perform a task) and the target domain (the data distribution that a trained model is applied on, to perform a similar task), widely accepted as the result of the low-level feature differences at the pixel level (Kundu 2022; Doan et al. 2024). Considering this issue, a plethora of studies looking into synthetic data generation strategies that can increase model generalisability and robustness of models using synthetic training images have been investigated, with two of the most common including domain randomisation and alterations to synthetic image realism.

- **Domain Randomisation:** An effective strategy used to create diverse synthetic image datasets that form the source domain. This is achieved by the randomisation of a variety of visual characteristics in the virtual environment where images are generated / captured. Randomised parameters often set to be within plausible ranges that will be found within the real-world visual data, however, randomisations may not precisely reflect the operating conditions and environment of the target domain (Collins et al. 2020). Although, it is recommended from well-established work by Prakash et al. (2019) that randomisations should account for the surrounding context of the synthetic image to also be representative of the potential target domain.

- **Synthetic Image Realism:** This strategy considers two major aspects of synthetic image fidelity.

  - **Photorealism:** This entails accurately reproducing the physical traits of light, texture, materials and shadows, within imagery (Kluge & Staadt 2024). In a published review paper by Nikolenko (2021) which provides the most comprehensive survey of various directions in the development and application of synthetic data for CV, concludes that findings of works concerning the level of photorealism required for synthetic training datasets and its impact on CV model performance is contradictory within literature and therefore remains an open question.

- **Sensor-Realistic Properties:** Camera sensors are electro-mechanical units which capture imagery from the real-world. Sensor characteristics and how light interacts with these units during image capture, are known to have a major impact on digital image quality, with introduced visible artifacts and effects, e.g. Chromatic aberration, blur, image noise, lens flare, exposure levels, etc. Synthetic training images inclusive and or simulating these properties are known to positively influence the performance of vision algorithms (Carlson et al. 2018; Hu et al. 2020)

  Nikolenko (2021), suggests that there is a plausibility that for low-level CV tasks, concerning low-level feature extraction, such as edge detection, segmentation, etc., sensor-realistic synthetic data is more important, compared to high-level CV tasks concerning high-level feature extraction, such as object detection.

Notable works that have investigated the strategies of domain randomisation and image realism and image augmentation are elucidated below:

2.2.2 Domain Randomisation

Sadeghi and Levine (2016) pioneered the application of domain randomisation, for synthetic data generation and set the foundation for subsequently related research. As part of their work, they proposed a method to successfully teach vision algorithms to process raw monocular camera imagery from a multi-rotor sUAS (and other robotic systems) and visually manoeuvre within real-world indoor environments, by being trained to operate in simulated (synthetic) environments. As part of this training, various synthetic hallway environments were created in *Blender* open-source rendering engine, involving 3D models of indoor settings, with varying fixtures and populated with obstructive items, alike environments to be encountered in the real-world.

To mimic the perspective of the multi-rotor sUAS in-flight and obtain visual data, a virtual camera / simulated camera was implemented in the simulation (details of which are omitted from the report). The movement of this camera was controllable in simulation by a deep convolutional neural network (DCNN) algorithm, which processed imagery it acquired from the virtual camera and extracted information to detect free space / detect the probable collision obstacles, before then being provided a command to alter the directional velocity of the multi-rotor sUAS to avoid a potential obstacle. A deep reinforcement learning (DRL) algorithm was used to optimise this model for collision-free flight through an indoor environment, after which experimental testing in the real environment was conducted using an actual multi-rotor system with an onboard monocular camera.

The main innovation of their work was that it presents the first demonstration of the concept of domain randomisation for synthetic data generation and shows that it can contribute to improved generalisation of vision-based models when applied to real-world scenarios, observed with reasonably good performance displayed during testing as evidenced by low collision rates. This is explained in the words of Sadeghi and Levine (2016), who state that domain randomisation "…forces the network to handle a variety of obstacle appearances and lighting conditions, which makes the learned representations invariant to surface appearance. As a result, the network learns geometric features and can robustly detect…". Additionally, since the synthetic data intentionally utilised non-photorealistic renders, this work is also significant in positing the idea that realism is not a required element for training vision-based AI learning with synthetic data.

Of particular note from the research method, were the domain parameters that were randomised in the synthetic training environments, which included:

- **Indoor Environment (Obstacle):** Virtual 3D environment was varied from a selection of 24 unique hallway configurations based on 12 floorplan layouts, along with fixtures, e.g. open or closed doors, windows, stairs, corridor width, etc.
- **Indoor Items (Obstacle):** A combination of different 3D models of common furniture items typically found within hallway environments were used and varied from a selection of 1 to 21, e.g. bench, chair, etc., to populate the virtual environment.
- **Indoor Environment Textures:** Textures of walls and floors were varied from a selection of 200 possible textures, e.g., wood, metal, textile, carpet, stone, glass, etc.
- **Virtual Camera Parameters:** Height and yaw angle of the virtual camera of the multi-rotor sUAS in-flight were varied to provide a diversity of viewpoints.
- **Lighting Parameters:** Directional light sources were placed and oriented within environments to vary illumination / lighting conditions.

Although the research focus is a robotic vision (RV) problem, these findings provide insight into the learning behaviour of vision-based algorithms, and thereby present relevancy to CV tasks, which is further provided in the conducted ablation study as part of their methodology.

To gain an understanding of how statistically significant domain randomisation is for synthetic training data compared to photorealistic training data, an ablation study was performed, which are a series of experiments conducted on trained AI systems by the removal / alteration of contributing factors to examine respective effects on the performance (Sheikholeslami 2019).

This was conducted on a task similar to CV-based objection detection, where an algorithm is trained on synthetic image data and tested on real-world data – in this case prediction of free-space of image pixels.

Synthetic training image data was captured randomly using the virtual camera as it was moved throughout the synthetic hallway environment, and essentially formed Pure Synthetic Data. For the ablation study, four types of synthetic datasets were captured as listed below, with each having specific domain parameters altered:

- **Randomised Simulation:** Multiple domain parameters were randomised within synthetic hallway environment during image data capture, used for nine hallway environments.
- **Realistic Textures and Geometry:** Synthetic hallway environment and items were only textured with real image textures obtained from a similar environment to simulate photorealism.
- **Fixed Textures Version 1:** Fixed textures and lighting were used for three hallway environments.

- **Fixed Textures Version 2:** Fixed textures and lighting were used for nine hallway environments.

***Note:** Synthetic images were rendered at 4K resolution rendered (assumed to be 3840 × 2160 pixels) to ensure the highest quality.

These datasets were then used to train four algorithms respectively, after which the models were then tested on real-world hallway datasets to ascertain performance metrics, in terms of a precision-recall curve, as seen in Figure 2.14.

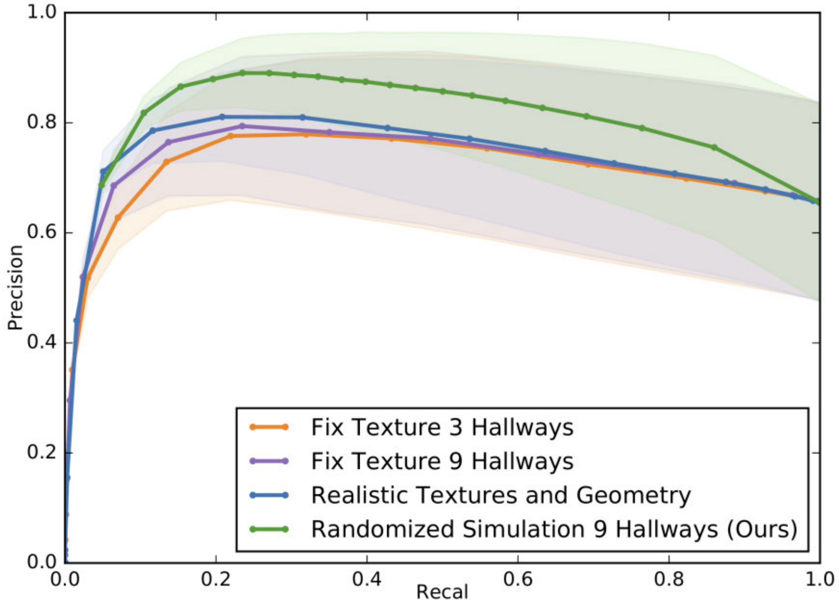


Figure 2.14: Sadeghi and Levine (2016) Image Pixel Free-Space Prediction Precision-Recall Curve

Despite not being explicitly related to CV, these results give an important and general understanding of the behaviour of CV algorithms, trained to achieve a task using synthetic training data. From the results it shows that included randomisation of the synthetic image environment (green), produces the best outcome, with maintained high precision over a wide range of recall values, which is indicative of a robust model. Photorealistic textures and thereby photorealism for synthetic training data does not surpass a randomised approach (blue) and shows less effectiveness, with results being more skewed towards lower precision and recall, indicating that it predicts fewer of the relevant items, but most predicted labels are correct. Fixed textures (orange and purple) indicate similar performance photorealistic textures, albeit slightly lower. A critique of this ablation study is that the amount of training data used for the models were not made known.

Work by Tobin et al. (2017) is the most distinguished investigation into the use of domain randomisation to bridge the gap between Pure Synthetic Data and real-world data. The premise of their work is also a RV problem, where a manipulator robot is taught to use monocular camera imagery to grasp simple geometric objects in a real-world environment using a CV-based object detection algorithm trained on Pure Synthetic Data. The direction of their work is heavily inspired by the final findings of the ablation study made by Sadeghi and Levine (2016) and its criticisms. They argue that synthetic training images generated using a virtual camera will always fall short, as they are unable to replicate the sensor-realistic effects inherent of real-world camera imagery; which contributes to the domain gap, despite the capability create photorealism. Resultantly, they hypothesised that non-photorealistic synthetic training imagery with enough variation provided by domain randomisation allows CV-based object detection models to generalise when deployed in the real world.

***Note:** A modified version of the VGG-16 algorithm was used for this investigation, which is a CNN developed by the Visual Geometry Group (VGG) and was selected due to its high-performance on CV tasks.

For the acquisition of synthetic data, Tobin et al. (2017) used an approach that mirrors Sadeghi and Levine (2016), albeit with notable adjustments. This was achieved with *MuJoCo* open-source physics simulation engine, using low-quality / low-rendering settings. The synthetic environment was setup similarly to the real-world test environment to be expected. In this case, simple 3D geometric models of target objects to be detected were positioned on a 3D table model, with a controllable virtual camera in *MuJoCo* used to capture synthetic imagery in the virtual environment. The randomised domain parameters are listed as follows with noteworthy expansions compared to Sadeghi and Levine (2016):

- **Target Objects:** Position and texture of objects of interest on the table.

- **Distractor Objects:** Number and shape of distractor objects.

    - Tobin et al. (2017) is the first to implement the use of distractors for vision algorithm training, used to clutter the environment and partially occludes the target objects. These are objects intentionally introduced in the environment of synthetic training images / source domain to confuse and draw attention of detector away from the target object, but in effect force the object algorithm model to focus on target objects and ignore the various distractions during training (Li 2023). In this instance, these were similar geometric objects, positioned around the target object on the table.

- **Environment Texture:** Textures of table, floor, and skybox were randomised to be either a standard RGB colour value, a transitional RGB value between two main RGB colour values, and or a checker pattern with two standard RGB values.

- **Virtual Camera Parameters:** Position, orientation and field of view of camera, with respect to target object.

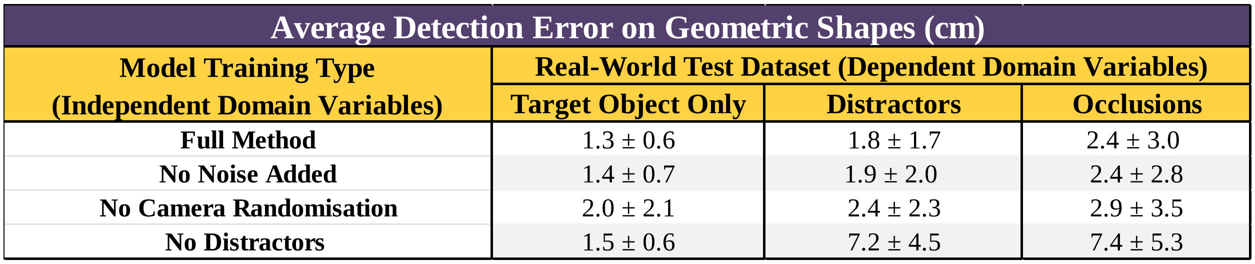
- **Lighting Parameters:** Number and location of directional light sources placed within environment.

- **Noise:** Different types and amounts of image noise introduced into synthetic images to simulate sensor-realistic properties.

    - Noise are undesirable artifacts and effects always present in digital images, such as grains, unrealistic edges, lines, corners, blurred objects, disturbances to background scenes, etc., which can be modelled mathematically and added to an image (Boyat & Joshi 2015). Tobin et al. (2017) is the first to implement noise within synthetic data, although does not specify the exact types and amount of implemented noise.

The novelty of this work is that findings strongly support the hypothesis and highlights that the domain gap can be bridged with sufficiently domain randomised synthetic training data, despite being non-photorealistic, and that this training strategy has credible real-world transfer to achieve CV-based tasks requiring accuracy.

This was made clear from experimental results obtained from the ablation study conducted Tobin et al. (2017), which tested synthetically trained CV-based object detection models on real-world images of the target object within the test environment. These models were trained with different randomised domain parameters (independent domain variables) that were either maintained or removed in the synthetic training data, which provides insight into their respective impacts on detection. Testing was done on three real-world datasets, which were inclusive of other specific domain randomised parameters within these images (dependent domain variables, i.e. target objects only, distractors and occlusion) to provide a deeper understanding of the impact of the independent variables. Refer to Figure 2.15.



Average Detection Error on Geometric Shapes (cm)			
Model Training Type (Independent Domain Variables)	**Real-World Test Dataset (Dependent Domain Variables)**		
	Target Object Only	**Distractors**	**Occlusions**
Full Method	1.3 ± 0.6	1.8 ± 1.7	2.4 ± 3.0
No Noise Added	1.4 ± 0.7	1.9 ± 2.0	2.4 ± 2.8
No Camera Randomisation	2.0 ± 2.1	2.4 ± 2.3	2.9 ± 3.5
No Distractors	1.5 ± 0.6	7.2 ± 4.5	7.4 ± 5.3

Figure 2.15: Tobin et al. (2017) Ablation Study: Average Detection Error of Geometric Shapes with Varied Domain Parameters [Edit]

As can be seen, it was found that the randomisation of all listed domain parameters in the training data incurred the lowest average detection error (1.3 cm - 2.4 cm) for all datasets.

Additional observations from the ablation study, which were not noted by Tobin et al. (2017) are identified below:

- The removal of randomised noise for model training produced a minor increase in average detection error (1.4 cm - 2.4 cm) for all datasets, which indicates that noise has a small statistical significance.
- Models applied to test datasets with target objects only had the lowest average detection error (1.3 cm - 2.0 cm), which shows that distractors and occlusions in test data have an impact on detection accuracy.
- Having no randomised distractor objects in the synthetic training data created the highest average detection error (1.5 cm - 7.4 cm), followed by no camera randomisation with an average detection error of (2.0 cm – 2.9 cm).
- Models trained with synthetic training data without distractors, were significantly impacted by distractors and occlusions within the test datasets, evidenced by large detection errors comparative to other models.

Factors that detract from these results are as follows:

- The results were not quantified using typical CV performance metrics, which makes it difficult to compare with other literature.

- The number of training data samples for the ablation study were not made clear, which further provides issues for a general comparison.

- These results focus on simple geometric objects in a simple synthetic environment, which holds limited scope for real-world detection applications. This was a missed opportunity to expand this research, to compare a system trained to detect more complex objects, within a more complex environment using a domain randomisation strategy.

However, this report does provide valuable insight into CV model pretraining to improve performance. Pretrained models are algorithms that have been trained on transfer learning datasets. These are large-scale, curated datasets, consisting of labelled images from a broad distribution relevant to the CV task and or operation. The pretraining process facilitates the acquisition an initial set of *weights*, w, and *biases*, b, at a random initialisation point (refer to Figure 2.6). This enables transfer learning, that does not necessarily improve model accuracy, but assists in the fine tuning of *parameters*, which accelerates convergence, particularly in the earlier phase of training with the actual training data. Therefore, the use of pretrained models is a standard practice for training CV algorithms (He, Girshick & Dollár 2019; El-Nouby et al. 2021; Encord 2023c).

This understanding is supported by the evaluation study run by Tobin et al. (2017), for the detection of target objects within the real-world test environment images. As part of this, a VGG-16 model with a pretrained backbone (feature extracting neural network part of a greater CNN algorithm) using ImageNet was compared with a VGG-16 algorithm trained from scratch, in terms of its average detection error of the target object, after being trained with 5000 to 50,000 samples of randomised synthetic data. It was found that the pretrained model can achieve lower average error on fewer samples, compared to a model trained from scratch. However, as the number of training samples increased towards 50,000, the average error of the algorithm and model came to be similar, as seen in Figure 2.16.

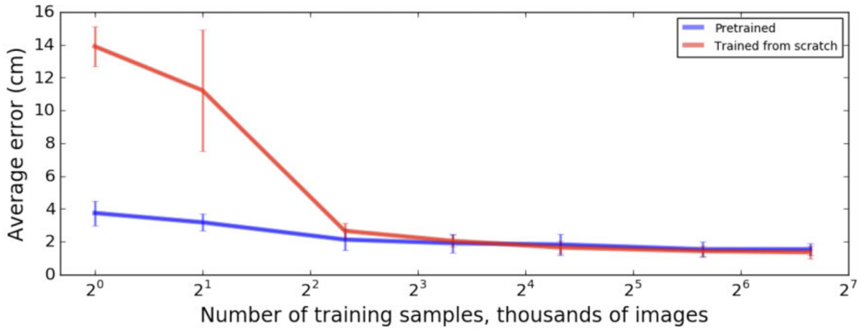


Figure 2.16: Tobin et al. (2017) Evaluation of Model Pretrained vs Trained from Scratch

Research by Tremblay et al. (2018) builds upon the research knowledge established by Tobin et al. (2017) and Sadeghi and Levine (2016), however their work is specifically focused on CV-based object detection and investigates if domain randomisation of synthetic data can be used to bridge the domain gap for detection of more complex real-world target objects amongst complex environments – in this case cars. Tremblay et al. (2018) asserts that the expense required to generate photorealistic and high-quality synthetic data, i.e. the use of high-end computer system equipment, undermines the argument that Pure Synthetic Data is a cost-free solution to gather large amounts of data. Therefore, as part of looking into a low-cost approach to synthetic data generation that does not prioritise photorealism, Tremblay et al. (2018) pursued the use of Type 1 Hybrid Synthetic Data, which are simpler to render than photorealistic Pure Synthetic

Data, and thereby are more achievable by low-end computer systems. UE4 game engine software was used, where low-quality / low-rendering settings were applied. Synthetic data was acquired in software by use of a virtual camera to capture imagery of 3D car models (imported) positioned in front of the 2D background images (imported). They interestingly state the creation and use of a plugin (extension software to Unreal Engine) that collected 1200 × 400 resolution images via the virtual camera, which then automatically annotated the car models, to form the synthetic images to be used for training. This was done to reduce the need for time-consuming manual labelling, however the methodology for this was not clarified.

The following parameters were randomised for the capture of these synthetic images and deserves attention:

- **Target Objects:** Number and types of target objects (3D car models) were varied from a set of 36 downloaded models. The texture of these target objects was also randomised.

- **Background Images:** Background images were randomised using Flikr8K dataset (publicly available large dataset consisting of 8000 random images (Kaggle 2024)).

- **Distractor Objects:** Number, type, colour and size of distractors were varied as 3D geometric shapes. These distractor objects created clutter and partially occluded the target objects similar to those used by Tobin et al. (2017), however instead of being only placed on the same level as the target object, these were also located in the airspace surrounding the target object within the virtual environment – known as flying distractors. Distractors on or near the same level as the target object will from now on be known as standard distractors.

- **Virtual Camera Parameters:** Randomisation of the orientation of the virtual camera with respect to target object, with a pan, tilt and roll ranging between -30° to 30°. Position of virtual camera with respect to target object, with azimuth ranging from 0° to 360° and elevation ranging from 5° to 30°.

- **Lighting Parameters:** Number and location of directional light sources placed within environment were varied from 1 to 12. Planar lighting (distributed light source from a large, uniform plane) provided for ambient illumination.

- **Visibility of Ground:** Ground plane was ensured to be visible to capture the complete domain environment.

In contrast to the experimental process by Tobin et al. (2017), it can be objectively stated that Tremblay et al. (2018) uses a more rigorous methodology to assess the effectiveness of non-photorealistic, domain randomised synthetic training data for CV-based object detection. The initial part of the methodology involved a comparison of three different types of object models trained using domain randomisation methods proposed by Tremblay et al. (2018), versus object detection models trained with a benchmark dataset of photorealistic Pure Synthetic images.

As part of this comparison, it is important to acknowledge the three CNN architectures selected by Tremblay et al. (2018), which used common high-performing backbone architectures, listed as follows:

- **Faster R-CNN:** Used Inception-Resnet V2 backbone as the feature extractor, pretrained on ImageNet.

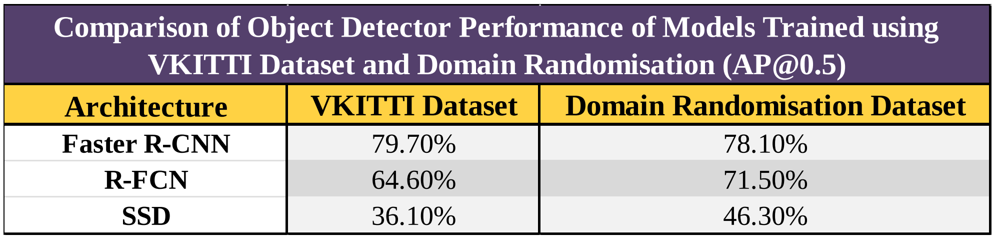
- **R-FCN:** Used Inception-Resnet V2 backbone as the feature extractor, pretrained on ImageNet.

- **SSD:** Used Resnet101 backbone as the feature extractor, pretrained on ImageNet.

While the exact rational for the selection of specifically three architectures was not explained, it can be assumed that this was to ensure a fair test, considering differences CNN architecture can impact values of accuracy of a model (as documented by Sanchez, Romero and Morales (2020)).

Additionally, to facilitate a proper comparison of the trained models, a benchmark dataset was utilised. In AI development, these datasets, are large-scale, curated datasets, consisting of labelled images from a broad distribution relevant to the CV task and or operation. These are standardised datasets (publicly available or accessible on request) and allow the meaningful comparison and valuation of trained models. This is completed by a model being trained with a benchmark dataset, and a separate model being trained with its respective data, after which both are evaluated on the same test dataset from the benchmark dataset (Blagec et al. 2023). Since the target object for detection were cars, the Virtual Karlsruhe Institute of Technology and Toyota Technological Institute (VKITTI) benchmark dataset was utilised, which is a photorealistic synthetic dataset of urban environments under different weather conditions and daylight hours, generated using the *Unity* game engine.

For this experiment, Tremblay et al. (2018) used 100,000 domain randomised synthetic images, generated by way of their method (Type 1 Hybrid Synthetic Data), and used 2,500 images from the VKITTI dataset (Pure Synthetic Data) for both training and testing. Also, the performance metric used was AP using a 50% IoU threshold (AP@0.5), which produced the following results, as seen in Figure 2.17.



Comparison of Object Detector Performance of Models Trained using VKITTI Dataset and Domain Randomisation (AP@0.5)		
Architecture	**VKITTI Dataset**	**Domain Randomisation Dataset**
Faster R-CNN	79.70%	78.10%
R-FCN	64.60%	71.50%
SSD	36.10%	46.30%

Figure 2.17: Tremblay et al. (2018) Comparison of Object Detector Performance of Models Trained using VKITTI Dataset and Domain Randomisation (AP@0.5) [Edit]

From these results it is evident that training an object detection model with synthetic data that uses domain randomisation produces better performance than photorealistic Pure Synthetic Data. This is evident with the difference in average precision for the R-FCN and SSD models trained with the domain randomised dataset and the VKITTI dataset, calculated to be 6.9% and 10.2% respectively. Since the average precision values are considerably high, these provide strong metrics that support the hypothesis of Tobin et al. (2017), that the domain gap can be sufficiently bridged with domain randomisation and ostensibly indicates that photorealism for training models with synthetic data is not necessary.

However, a noticeable issue with these results is the fact of the imbalance between the amount of training data used for the domain randomised dataset and VKITTI dataset, which may be a significant contributor to a biased result. It was also seen that model training with the photorealistic Pure Synthetic Data produced marginally better performance with the Faster R-CNN model with a 1.6% advantage. This outlier result may be due to superior precision exhibited by the specific Faster R-CNN model architecture used. Alternative reasoning could be attributed to the fact that a moderate level of domain randomisation is induced when the object detection model is trained on the VKITTI dataset considering variations in object textures, backgrounds, distractor objects, and occlusions as an uncontrolled environment. This may be suggestive of the underlying potential of photorealism for synthetic training data to be competitive, provided that the source domain is sufficiently varied with enough training data volume. Additionally, this indicates that domain randomisation can be of parameters that would be expected to change within the real-world (realistic domain randomisation) and does not necessarily have to be of a high-level as seen in previous works.

Tremblay et al. (2018) provides additional value from their investigation, by performing a comprehensive ablation study to understand how statistically significant the individual randomised parameters of synthetic training data are for detection in the real-world, as seen in Figure 2.18.

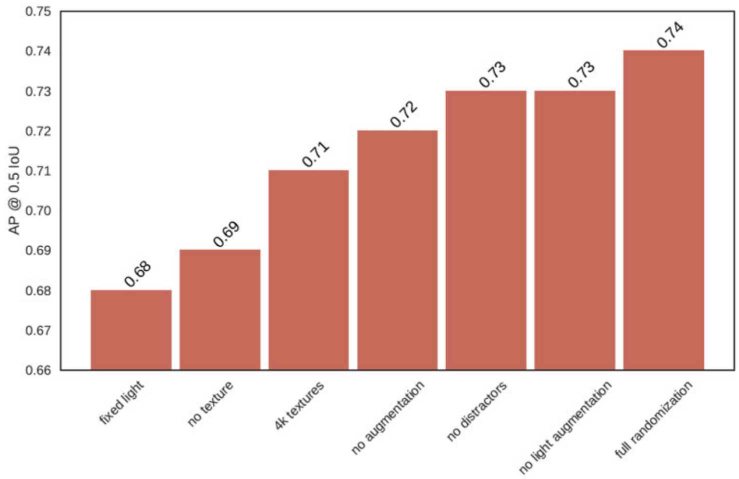


Figure 2.18: Tremblay et al. (2018) Ablation Study Results (AP@0.5)

For this ablation study, a Faster R-CNN architecture using Resnet V1 backbone as a feature extractor, pretrained on ImageNet, was trained with 50,000 domain randomised synthetic images, generated by way of their method (Type 1 Hybrid Synthetic Data), and was tested on a real-world benchmark dataset – Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI), which consists of real-world images of urban environments in traffic scenarios with cars, under clear weather and daylight hours (Paperswithcode 2024). Performance was measured using AP@0.5. Results from this study are important to take into account:

- Using the complete set of domain randomised parameters the AP was measured at a base line of 74%

- When the lights positions were randomised, but brightness and contrast turned off (no light augmentation) the AP was measured at 73%, which is a minor 1% decrease from the baseline. However, when the detection algorithm was trained on fixed lights, this had the greatest impact with the lowest recorded AP at 68%, which is a 6% decrease from the baseline. These findings indicate that that randomised lighting for synthetic datasets significantly contributes to improved generalisation and model robustness, compared to static lighting. Additionally, it shows that lighting augmentations can slightly degrade detection, and thereby lighting properties hold some significance, albeit minor.

- The use of no flying distractors for the synthetic training dataset had a minimal effect with a 1% reduction of performance. Tobin et al. (2017) similarly performed an ablation study with a model trained on synthetic training data using no standard distractors; when applied on real-world data with distractors and occlusions, it instead showed poor detection (refer to Figure 2.15) and therefore shows a contradiction in results with Tremblay et al. (2018).

- Texture properties contribute low-level feature information which is extracted by a CV model during the hierarchical feature abstraction process of training, and thereby contributes to CV model generalisation when deployed.

  4K textures (assumed to be 3840 × 2160 pixels) saw a 3% reduction in performance with an AP of 74%, which shows that high-fidelity textures have a minor impact on performance and demonstrates that the photorealism brought about with 4K textures is not as impactful as domain randomisation.

  The use of no textures for synthetic training data saw a performance decrease of 5% with an AP of 69%. With the second lowest performance this demonstrates that textures are statistically significant parameter for synthetic training data.

***Note:** Ablation study hyperparameter values: Epoch: Not stated, Batch Size: 4, Learning Rate: 0.0003

In a similar, but different vein, research by Damian et al. (2023) investigated the comparative impact of using primarily photorealistic Pure Synthetic Data (using a minor level of domain randomisation), versus using photorealistic Pure Synthetic Data using a high level of domain randomisation, for CV-based object detection. For context this work involved the detection of barrel drums. Damian et al. (2023) made innovations of interest with the development of a pipeline that provides a streamlined and progressive methodology to generate synthetic data and train CV-based object detection models. Results of their work are pertinent to the focus of this research report, since the rendering engine used was UE5.

The following pipeline was implemented by Damian et al. (2023) for their investigation:

1. Photogrammetry was used to scan the target object to be detected, in this case barrel drums, to obtain high-fidelity 3D model scans. The process was enabled by the *Scaniverse* application using an iPad tablet. The target object scans were then imported into the UE5 virtual environment. Damian et al. (2023) does not justify reasoning for this approach in modelling barrel drums, however it can be assumed that this was to ensure a high similarity between the training and testing object data, in order to limit the impact of discrepancies in object form perceived by the model feature extractor, contributing to unrepresentatively lower performance than what could be.

2. A background environmental setup within UE5 was designed, for synthetic imagery to be acquired using a virtual camera.

3. This camera was initially positioned near ground-level of the virtual environment with a view of the target object and was rotated in 5° intervals along the azimuth angle to capture imagery. After each rotation, domain parameters were randomised. This process was repeated until the camera completed a 360° revolution, thereafter the camera was raised by 5° to repeat this process again, until a 90° elevation was reached. Damian et al. (2023) justified the logic behind this process as being intended to gather a comprehensive training dataset from different viewpoints, however, Prakash et al. (2019) forwards the idea that it may not be necessary to exhaustively iterate over entire ranges of modifiable domain parameters.

4. *Roboflow*, an online platform which allows the end-to-end development and deployment of CV models with relative ease, including object detection was used. Users can select algorithms to be trained 'out-of-the-box' by uploading according image data for training, validation and testing. Upon completion of training, performance metrics are obtainable for model review. Damian et al. (2023) opted to use CNN algorithm, YOLOv8, which employs a CSPDarknet53 backbone for feature extraction (Torres 2024). A key choice that Damian et al. (2023) makes is that they purposefully avoid the use of a pretrained backbone, to eliminate biases that would be introduced to the model from preconfigured algorithm parameters, that would ultimately influence test results.

As part of the study, synthetic image data of barrels were manually collected by Damian et al. (2023) and used as training and validation data for the YOLOv8 algorithm. To clarify, 260 real-life images were captured using both a multi-rotor sUAS and smartphone camera sensors. The reasoning for the use of two camera sources was not specified by the authors, however it can be assumed that this was for the purpose of diversifying image data considering inherent camera lens characteristics and camera sensor effects, which may induce dataset bias.

Three experiments were conducted using datasets that progressively increased to higher levels of domain randomisation, based on the outcome of the previous, after which changes to the synthetic environment were made. Performance results for the experiments were in terms of an F1 score. The key strength of the results was the use of three confidence levels, 20%, 50% and 80%, to effectively quantify the balance between precision and recall.

For the first experiment Damian et al. (2023) created Dataset Version 1, by designing a synthetic forest background environment with the barrels placed on the ground amongst trees, fallen branches and grass. These acted as distractor objects that created clutter around the target objects, although these were not randomised, nor were they modified in number, shape and or colour as with standard or flying distractors. It is not known if these created occlusions, since Damian et al. (2023) does not make clear the distance(s) between the virtual camera and the target object that was utilised. The only domain parameter randomised was the lighting conditions, by modifying the virtual environment lighting, by changing the time of day, over a day-night cycle to mitigate fixed lighting condition biases, which would impact the object detector, as found by Tremblay et al. (2018). The results of the model, trained with captured synthetic imagery from this environment, showed suboptimal F1 scores when deployed on the real-world test dataset, with results ranging between 15% - 25% amongst the three confidence levels – refer to Figure 2.19.

Considering the poor model performance, Damian et al. (2023) generated Dataset Version 2, which increased the number of training images to 6,737, with the introduction of standard distractors, as well as randomisation of colours for each material within the virtual environment. Testing showed an improvement in the F1 score with results ranging from 25% - 43% amongst the three confidence levels – refer to Figure 2.19. With the still suboptimal performance, Dataset Version 3 was formed, with the number of training images expanded to 13,978, as well as distractors varied in size and shape, which displayed a demonstrable boost for generalisation, with an F1 score ranging from 80% - 85% – refer to Figure 2.19.

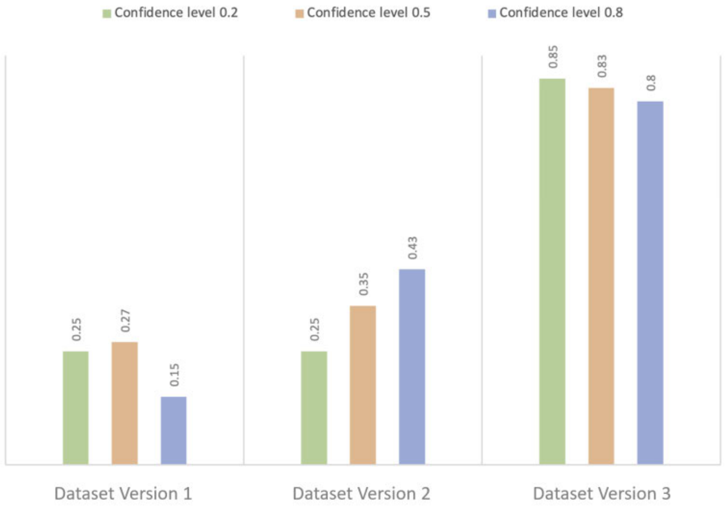


Figure 2.19: Damian et al. (2023) Testing Results Dataset Version 1, 2 & 3 [Edit]

The investigation by Damian et al. (2023) holds a large significance, as it distinctly demonstrates that high-quality photorealistic Pure Synthetic training data achieved by a state-of-the-art rendering engine with high-fidelity lighting, shadow, texture and material simulation provided by this software, such as UE5, cannot independently improve model generalisation by virtue of enhancing image details for extraction by the CV algorithm – this was clearly demonstrated with the underperformance of the model trained on Dataset Version 1. This result can also be attributed to the minor level of randomisation that Dataset Version 1 utilised.

Improvements in the generalisation of models trained using higher levels of domain randomised synthetic training data provided by Dataset Version 2 and Dataset Version 3 respectively, reinforces findings of Tremblay et al. (2018), Tobin et al. (2017) and Sadeghi and Levine (2016). However, it is important to note that these improvements were also made possible by large, stepped increases of training data for Dataset Version 2 and Dataset Version 3 respectively, which expands the volume of information the models can learn from. Resultantly, this skewed the findings of Damian et al. (2023) and presents biased results in favour of domain randomisation. Therefore, it can be stated that the question of the efficacy of photorealistic synthetic training data for CV-based object detection is still open.

Other drawbacks and points for improvement of this research include the following:

- Lack of clarification of which YOLOv8 model and hyperparameters were used for training.
- UE5 settings used in investigation to achieve photorealism were not made clear.
- Details are not provided of the target domain from the real-world test dataset; therefore, it is difficult to contextualise the precise reasoning for the poor model performance.
- Additional models should have been tested for this investigation, to ensure a fair test.

2.2.3 Photorealism

Most works in literature involving the use of synthetic training data for applications on real-world visual data are heavily influenced by the apparent conclusions of Sadeghi and Levine (2016), Tobin et al. (2017), and Tremblay et al. (2018), and thus assume that photorealism unnecessary. An in-depth literature review of these works reveals otherwise, and that these previous studies have not completely explored the efficacy of photorealistic synthetic training data and its potential to bridge the synthetic-to-reality gap, considering experimental biases.

Only a limited number of works have made further investigations into this literature gap. The study by Movshovitz-Attias, Kanade and Sheikh (2016) is one of these main works. They pioneer the use of using synthetically generated training data for the CV task of viewpoint estimation of real-world objects, in this case cars. The relevancy of this study is that it is one of the only works to properly demonstrate the importance of photorealism for synthetic training data and what photorealistic features can be used to maximise its potential for a CV task, which has transfer for object detection.

***Note:** A CNN architecture called AlexNet was used with several modifications made for the task.

For this investigation, they comprehensively compare the performance of models for 3D viewpoint estimation, which are trained with both synthetic and real-world training datasets, both individually and mixed. These datasets are detailed below:

- **Real-World Datasets:**
  - ➢ **CMUCar:** Real-world dataset of over 3,500 (SDR) images consisting of cars, augmented by Movshovitz-Attias, Kanade and Sheikh (2016) camera matrices for images, using an augmented dataset by Naresh Boddeti, Kanade and Vijaya Kumar (2013) containing landmark annotations of cars, previously based on the MIT Street Dataset. This dataset was used to assess the performance of a model trained on data only relevant target objects in their according domain.
  - ➢ **PASCAL3D+ (PASCAL)**: Benchmark dataset contains over 13,900 (SDR) images using 12 categories of random objects in the real-world, i.e. airplanes, boats, bottles, cars, chairs, etc., with over 3,000 instances of each category. Images are captured under uncontrolled settings, in cluttered scenes and under many different poses. These objects are annotated with pose information, i.e. azimuth, elevation and distance to camera (Elhoseiny et al. 2015; Yu Xiang 2017). This dataset was used to assess the performance of a model trained on a larger variety of data with relevant target objects and non-relevant objects in varied domains.
  - ➢ **Mixed Real-World Dataset (PASCAL + CMUCar):** Dataset combined PASCAL3D+ (P) and CMUCar (C) datasets. This dataset was used to assess the performance of a model trained with a larger dataset of relevant target objects and non-relevant objects.
- **Synthetic Datasets:**

  Type 1 Hybrid Synthetic Dataset (Render): Although this research intends to assess the performance of a model trained with synthetic data prioritising photorealism, Movshovitz-Attias, Kanade and Sheikh (2016) considers that Pure Synthetic Data is both time consuming and computationally intensive to render (similar to Tremblay et al. (2018) stated), therefore photorealistic Type 1 Hybrid Synthetic Data was used as training data instead. For the generation of this data, Movshovitz-Attias, Kanade and Sheikh (2016) used Autodesk *3Ds Max* rendering engine with the VRAY plugin, a third-party extension to *3Ds Max* software that can achieve high-quality renders with high-fidelity lighting and shadows

– details of software settings were not stated. In order to capture this imagery, 3D CAD models of cars were imported from online into the software and positioned in front of background images derived from the PASCAL3D+ dataset. However, in the case of this training dataset, high-fidelity lighting and shadows were omitted to provide a moderate level of photorealism. This dataset was known as RenderCar and cars consisted of 819,000 images. A unique approach was taken to acquire imagery with the use of multiple virtual cameras placed around the randomised target object to form a sphere. These cameras were spaced in 1° increments within rings at five elevations: -5°, 0°,10°, 20°, 30°

- **Type 2 Hybrid Synthetic Dataset (Render + PASCAL):** This dataset was used to assess the impact of combining Type 1 Hybrid Synthetic Data as well as real-world data.

***Note:** Both training and testing images were captured close to the car as the target object.

The following parameters were randomised for the capture of these synthetic images. They make highly innovative and in-depth considerations for randomised domain parameters, with variations being mostly limited to bounds of what would be possible in the real-world, as detailed below:

- **Lighting Parameters:**

  - **Light Position:** Directional lighting source was positioned randomly around sphere at elevations of 10° and 80°.

  - **Light Temperature:** One of nine light temperature profiles in terms of Kelvin are randomly selected within the virtual environment to simulate real world lighting scenarios, i.e. midday sun, overcast sky, tungsten lightbulb, etc, to vary the colour of the light source.

  - **Light Intensity:** Luminous power which is the total visible light power, was randomly modulated between 1,400 to 10,000 to also randomly selected to simulate real world lighting scenario variation.

- **Virtual Camera Parameters:** Sensor-realistic properties were introduced to data to enhance synthetic image realism.

  - **Camera F-Stop:** Virtual camera aperture, which controls the amount of light entering a camera lens and controls how much of the image is in focus (depth of field) was randomly varied.

  - **Camera Shutter Speed:** Shutter speed of virtual camera was randomised between $\frac{1}{25}$ and $\frac{1}{200}$ of a second.

  - **Lens Vignetting:** Varied simulation of optical vignetting effect, which is the gradual reduction of image brightness or saturation from the image centre to the periphery.

  - **Background Images:** Background images for Type 1 Hybrid Synthetic Data were randomised using PASCAL3D+ dataset.

- **Compression Effects:** Movshovitz-Attias, Kanade and Sheikh (2016) mentions that it would have been ideal to have synthetic images captured from the virtual camera saved in a PNG image format due to its lossless compression, which preserves image quality. This is a critical considerations since current DNN architectures are significantly affected by low image resolution (Koziarski & Cyganek 2018). However, as the model was to be exposed to benchmark images that use JPEG images during training, which have image quality losses when compressed in its respective format, Movshovitz-Attias, Kanade and Sheikh (2016) made a compromise to save rendered synthetic images in JPEG format (.jpg).

- **Occlusions:** To improve the model robustness to potential occlusions, randomised rectangular patches of uniform colour or images from PASCAL3D+ dataset were overlayed on the image. These patches took up between 20% to 60% of the synthetic image. This can be stated to be an effective method simulate real-world scenarios where a target object may be partially obscured by another object.

As part of a comparison of the different models trained on these datasets in terms of car viewpoint estimation performance, all models were tested on car class images from the PASCAL3D+ dataset, with results quantified in terms of median azimuth angular error (degrees), as observed in Figure 2.20.

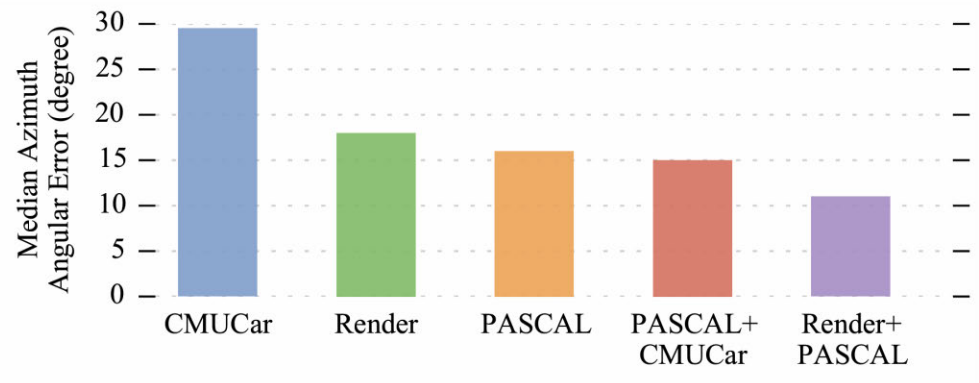


Figure 2.20: Movshovitz-Attias, Kanade and Sheikh (2016) Results of Car Viewpoint Estimation by Models Trained Varied Datasets

These results are important to understand, as they clearly display a model trained with synthetic data with a moderate level of photorealism (Render training dataset – green), outperforming a model trained with real-world data (CMUCar training dataset – blue) by a considerable margin (approximately 12° from visual inspection of Figure 2.20) for a CV task and challenges the established notion that synthetic training data prioritising photorealism will not enable a model to effectively generalise with real-world data. Since the Render training dataset consists of a larger volume of training data compared to the CMUCar training dataset, and also is comprised of imagery that is more randomised in terms of domain, this confirms observations from Damian et al. (2023), that these factors improve model generalisation. Since the Render training dataset also uses realistic domain randomisation, that is randomised to a high-level and has a moderate level of photorealism, this supports the idea that synthetic data employing photorealism, which is sufficiently varied with enough training data can effectively generalise on real-world data and potentially bridge the synthetic-to-reality gap, as was surmised with the high performance of the Faster R-CNN model trained with the VKITTI dataset in the work of Tremblay et al. (2018).

Another valuable revelation from these results is that Type 2 Hybrid Synthetic Data demonstrates the best results with the lowest error (Render + PASCAL dataset). Movshovitz-Attias, Kanade and Sheikh (2016) suggests that this is due to the training algorithm being forced to learn from two types of data. Further expanding on their idea, this can be stated as logical since the algorithm must learn over a wider range of data, essentially two different domains and thereby the gap is more effectively bridged as the algorithm must abstract higher-level features that share commonality between both domains, rather than low-level features, which in effect improves robustness when encountering real-world data which is highly variable and nuanced.

Movshovitz-Attias, Kanade and Sheikh (2016) also investigates the impact of photorealism for synthetic training data and questions if it is worth the cost of computational resources. Photorealism in rendering considers the 3D object relationship within the virtual environment and thereby places an emphasis of the simulation of lighting, shadows, material and texture properties (Joon 2010). Rendering engines have the ability to control levels of photorealism via adjustable rendering quality. For this evaluation three AlexNet models were trained to perform car viewpoint estimation using Type 1 Hybrid Synthetic datasets, that combined a PASCAL3D+ dataset with synthetic car image data. All models were tested on car class images from the PASCAL3D+ dataset, with results quantified in terms of median azimuth angular error (degrees). Each of the three models were respectively trained using synthetic car image data consisting of three adjusted levels of photorealism (taken at random angles), as seen below:

- **Simple Material, Ambient Lighting (1):** Simple render of car body material property and uniform ambient lighting (non-randomised) – low-level photorealism
- **Complex Material, Ambient Lighting (2):** Complex render of car body material property and uniform ambient lighting (non-randomised).
- **Complex Material, Directional Lighting (3):** Complex render of car body material property, and directional lighting where the location, colour, and strength were randomised – high-level photorealism.

Considering the established importance of dataset quantity, model performances were evaluated over a range of synthetic dataset sizes, synthetic images ranging from 5,000, 22,000, 58,000 and 94,000 images. Results are presented in Figure 2.21:

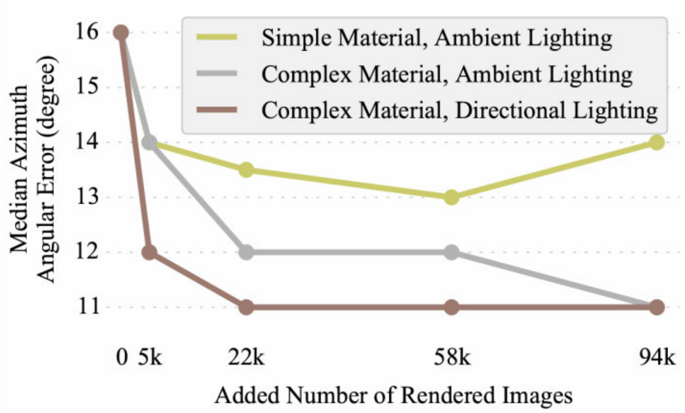


Figure 2.21: Movshovitz-Attias, Kanade and Sheikh (2016) Results of Car Viewpoint Estimation by Models Trained with Three Levels of Photorealistic Synthetic Data

Referring to Figure 2.21, the following observations from this experiment are valuable to take note, which were not highlighted by Movshovitz-Attias, Kanade and Sheikh (2016):

- For all models it was noted that increasing the size of the training dataset does improve model performance. Training volumes are ideally in the tens of thousands, as shown in the work of Damian et al. (2023) and to a lesser degree Tremblay et al. (2018), but only to a certain extent, depending on the model, before diminishing returns are met.

- Over the use of 0 to 58,000 training images, the viewpoint estimation error of a model trained with synthetic training data using a simple material property and uniform ambient lighting (1) decreased by approximately 3°; where as the model trained with synthetic training data using a complex car material property and uniform ambient lighting (2) decreased by approximately 4°, which demonstrates the minor impact of material property photorealism for rendered objects on CV model performance, since ambient lighting is not varied for these two models. Material properties contribute low-level feature information which is extracted by a CV model during the hierarchical feature abstraction process of training, and thereby contributes to CV model generalisation when deployed.

  In comparison to Model (2) the model trained with synthetic training data using a complex material property and randomised directional lighting (3) decreased by approximately 5°, over the use of 0 to 58,000 training images. This is indicative of the greater significance of randomised directional lighting compared to complex materials. The ablation study by Tremblay et al. (2018) is also in agreement with this finding, that randomised lighting for synthetic datasets substantially contributes to improved generalisation. Although Tremblay et al. (2018) does miss an opportunity to explore the impact of randomised ambient lighting to quantify its impact comparative to randomised directional lighting. These clear improvements in model performance over increased levels of photorealism / fidelities of photorealism, shows evidence that CV models trained with synthetic data benefit in terms of accuracy, provided that the synthetically generated data has a large volume and employs sufficient domain randomisation, which supports observations from the work of Tremblay et al. (2018).

- Model (3) appears to achieve better performance, reaching a low viewpoint estimation error, with a smaller volume of training images at 22,000 images, whereas Model (2) and Model (1), with lower levels of photorealism, achieve lower viewpoint estimation errors at 55,000 images and 94,000 respectively. This is an indication that both lighting and material properties support each other and plausibly indicates that synthetic training data with a high-level of photorealism encompassing factors of lighting and complex material properties, can provide improved results over smaller training data volumes, whilst employing domain randomisation. Although this observation contradicts the results of Damian et al. (2023), which showed significant CV model underperformance despite employing high-level photorealism with the UE5 engine; their CV model results significantly improved with larger training data volumes – further, exploration of this observation is required.

The latter two points supports the need for high-fidelity photorealism in terms of lighting and material properties within synthetically generated data and that it is worth additional computational effort.

2.2.4 Rendering Engine Lighting Theory

2.2.4-1 Introduction to Rendering Engine Lighting Theory

From previous works it is apparent that high-level domain randomisation and large training volumes for synthetically generated training data plays a primary role in the reduction in the synthetic-to-reality gap, providing the most substantial contribution to the enhancement of CV model performance. In addition, high quality photorealism in rendered imagery with simulated lighting and material properties can provide complementary support to further improve performance. Expanding on the latter observation, rendering engines use the interaction of simulated light with object surfaces within the virtual environment (light transport) as a proxy to achieve photorealism, which facilitates the representation of shadows and physical properties such as material and texture. At a fundamental level this is enabled by the combined support of both illumination models (also known as lighting models or shading models) and physically-based rendering (PBR), which are methods to assist rendering techniques accurately replicate real-world lighting (lighting fidelity), including direct and indirect lighting, generated from light source models and interactions within materials. Other concepts important to achieving photorealism will be outlined in this section, which form part of general rendering engine capabilities.

2.2.4-1.1 Light Source Models

Light source models are simulated sources of light in the virtual environment (Yadav 2020).

- **Point Source:** Light rays emanate from a point radially in all directions, i.e. spotlight (Foley & Dam n.d.; Tel Aviv University n.d.).

- **Parallel Source:** Light rays emanate and travel from point source in parallel to each other in a single direction. The point source is modelled at an infinite point in the virtual environment, i.e. sun (Foley & Dam n.d.; Tel Aviv University n.d.).

- **Distributed Source:** Light rays emanate from a finite area. The uniformity of a light source can be modified to achieve different lighting effects, i.e. IBL, planar light, etc (Foley & Dam n.d.; Tel Aviv University n.d.).

2.2.4-1.2 Illumination Models

Illumination models (group of algorithms) simulate lighting within a virtual environment by computing the intensity of light reflected from a point on an 3D object surface by use of illumination and shading functions. Calculations made by illumination models are contingent on the following factors (Komura 2011; Yadav 2020; Bar-Ilan University n.d.; Wüthrich n.d.):

- **Light Source Parameters:** Position of lighting source(s), colour and strength of the emitted light, emitted light direction, shape of light source(s), etc (Middle East Technical University 2007).

- **Object Surface Parameters:** Object position with respect to light source(s), object reflectance properties, object position with respect to other objects (Middle East Technical University 2007; Foley & Dam n.d.; Tel Aviv University n.d.).

- **Virtual Camera Parameters:** Position of virtual camera(s), sensor-realistic properties, orientation of viewing plane, etc. (Foley & Dam n.d.; Tel Aviv University n.d.).

The primary illumination model which is used by rendering engines is global illumination (GI), which simulates the complexities of both direct lighting and indirect lighting (i.e. ambient light, diffuse reflection and specular reflection) within the virtual environment, which encompasses light emitted from direct light source(s), light reflected and refracted by the other surfaces, as well as soft shadow effects (Unity 2016c; Whitted 2020; Adobe 2024h; Wüthrich n.d.).

2.2.4-1.3 Physics-Based Rendering

PBR is an approach to simulate the physical behaviour of light when it interacts with object surface materials within a virtual environment. This is achieved using shading models and surface scattering models, which determine the distribution of light that is scattered from points on different types of object surfaces due to incoming light – primarily reflective and refractive light (Jakob 2013; McGuire et al. 2020; Adobe 2024a; SGVR Lab n.d.). Material properties are assigned to object meshes of a 3D model within software that match the real-world; surface scattering models utilise these properties to calculate light scatter and more accurately achieve photorealistic visual fidelity within the virtual environment (Moor Insights & Strategy 2017; Adobe 2024f; Iontcheva 2024).

***Note:**

- Prior to further explanation, it is worth mentioning that for DCC, 3D models utilise data rich file formats that store information encompassing not only the geometric data that defines the structure of a model as an object, but also a range of other model attributes such as assigned material and texture properties (Martínez-Díaz 2022; Alpha3D 2023; Adobe 2024d; Schechter 2024). Generally, a 3D model of an object is defined as a polygon mesh (object mesh), where each polygon consists of vertices, edges and faces are used to virtually represent the surface of an object/component and thereby the overall geometry in detail. Models rendered within software are localised within a 3D cartesian coordinate system (x,y,z) of the rendering engine. Vertices of a model store geometry and position data of a model within the 3D space of the virtual environment, in terms of the local coordinates of the rendering engine (Adobe 2024c; Tiigimägi 2024; UNSW Sydney n.d.).

- Surface reflection and refraction are important visual characteristics that contribute to photorealism, which are dependent on both illumination and viewing orientation within a virtual environment (Nayar, Ramamoorthi & Hanrahan n.d.). Since surface reflections are also offered by PBR and by extension GI, this further emphasises the need for the orientation of a virtual camera to be varied as part of domain randomisation.

2.2.4-1.4 Shadow Mapping

To achieve hard shadows in renders, shadow mapping is used, which at a basic level is completed in two stages (passes) by a shadow mapping algorithm. For the first pass the shadow map is created, this is completed when a scene is rendered from a light source perspective, where fragments are sampled from the light source to the closest visible object surface points in the scene, as viewed by the light source. A fragment represents a point on an object mesh surface, which is an interpolation of the vertices of a mesh, and holds respective values such as texture, material, coordinates. The distance of the fragments from the light source (depth values) are derived using the 3D coordinate data, which is then transformed (mapped) into the light source coordinate space using view and projection matrices specific to the light source. After transformation, the depth information of the fragments is processed and stored as pixels in a shadow map (2D texture) – refer to Figure 2.22. The pixel resolution of the shadow map determines how detailed rendered shadows will be, with larger shadow map resolutions providing more precise the shadows (vice versa) (Gerasimov 2004; Hock-Chuan 2012; Unity 2016a; De Vries 2017; Hocevar 2017; Unity 2024b).

For the second pass, a standard render of a scene is completed using backwards ray tracing from the camera perspective. As part of this process a secondary set of fragments are sampled, where the 3D coordinates of each fragment are similarly transformed into the light source coordinate space, by use of the same view and projection matrices that are employed in the first pass. To explain this concept further, when a fragment of an object mesh is sampled from the camera perspective at point *P*, its coordinates are transformed to *P'* within the light source coordinate space, which essentially aligns *P* with the shadow map and provides a depth value relative to the light source. Using the coordinate *P'*, the corresponding 2D location in the shadow map is indexed to retrieve the stored fragment depth value *C* (sampled from the light source) – refer to Figure 2.22. A comparison is then done between the depth values of the transformed fragments from the camera perspective (second pass) and the corresponding depth values stored in the shadow map from the light source perspective (first pass), to check if fragments in the standard render are lit or shadowed from the employed light source, which considers the following:

- If depth values of a transformed fragment from the camera perspective is greater than the depth values stored in the shadow map, this indicates that the fragment is occluded by an object that is closer to the light source in the scene and therefore is in shadow.

- If depth values of a transformed fragment from the camera perspective is less than or equal to the depth values stored in the shadow map, the fragment is considered lit and not in shadow.

Calculations for this check are made by a fragment shader, which then renders shadows accordingly within a scene (Gerasimov 2004; De Vries 2017; Hocevar 2017).

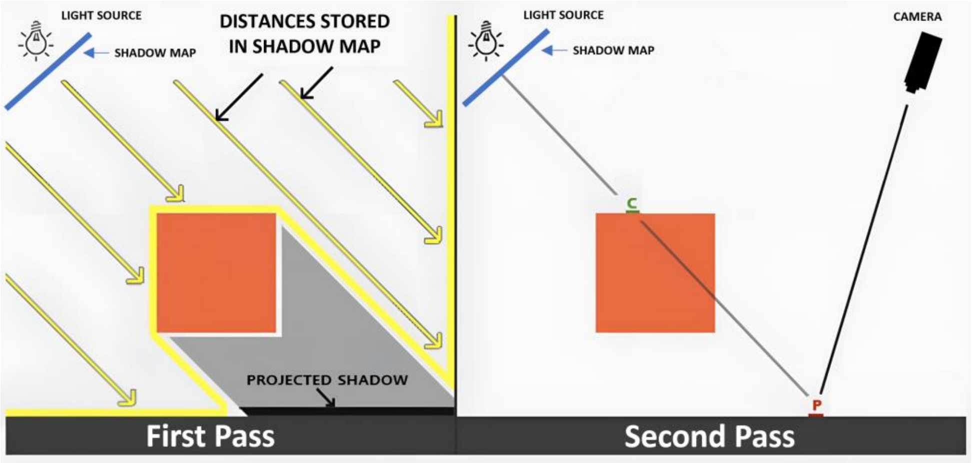


Figure 2.22: Illustration of Shadow Mapping Process (Adapted from De Vries (2017))

2.2.4-2 Ray Tracing and Path Tracing

It is important to understand that GI and PBR are methods of photorealistic lighting and material simulation, employed for rendering. Rendering is the process where a rendering engine computes and converts 3D environments captured by a virtual camera into a 2D image frame. As part of rendering, different techniques are used to model light transport, which utilise GI and PBR. Currently, there are two primary rendering methods – ray tracing and path tracing. These techniques are implemented in various rendering engines using differing algorithms (Keller et al. 2019).

2.2.4-2.1 Ray Tracing

For the sake of process efficiency, backwards ray tracing algorithms are chiefly used for image rendering, where initially multiple rays (computational constructs) are cast from the view of the virtual camera, through each respective pixel of the image frame (pixel plane which makes up the rendered image), which is then output into the virtual environment (Stanford University 1998a; Eclat Digital 2024; NVIDIA 2024a). These are initially known as primary rays, depending on material properties assigned to the object mesh encountered, these intersect object surfaces and at the point of intersection may be absorbed, while other rays may bounce (reflect or refract) to generate multiple secondary rays. These rays may continue to bounce (forming a tree of rays) until the rays are absorbed by an object, or they eventually reach a light source model – refer to Figure 2.23 (Whitted 2005; Eck 2021; Eclat Digital 2024; Epic Games 2024f). Rays serve the purpose of sampling, which calculate the surface colour and intensity at the coordinate point of intersection on an object by the ray. This point is determined with interpolation calculations using coordinate values from the mesh vertices (Washington University 2007; Hobart and William Smith Colleges n.d.; University of Texas at Austin n.d.). Sampling is achieved by computing the sum of the individual contributions of light simulated to be at the points by GI and PBR (Carnegie Mellon University 1998; Stanford University 1998b; Rupard 2003; Eck 2021; Peer Eric Molvar 2021; SGVR Lab n.d.). In the case of ray tracing, deterministic sampling is generally utilised, where primary and secondary rays are uniformly spaced and distributed, at angles that align with the direction of known light source models (Cook 1989; Whitted 2005; Subr 2008; Mardaljevic 2011). The determined surface colours are then assigned to the corresponding pixels to form the finalised image render (Carnegie Mellon University 1998; Peer Eric Molvar 2021; NVIDIA 2024a). The traced paths made by these rays mimic the physical behaviour and interaction of light that would be made with real-world object surfaces (Eclat Digital 2024; Epic Games 2024f); therefore, rays accurately gather relevant light contributions from the virtual environment that assist in the determination of an image render with accurate lighting. However, it is important to note that ray tracing is not physically precise, but is rather an approximation, with limitations such as:

- Unable to simulate diffuse inter-reflection between surfaces (Bouatouch n.d.).
- Rendered images are have extreme and unrealistic sharpness (Ragnemalm 2024).

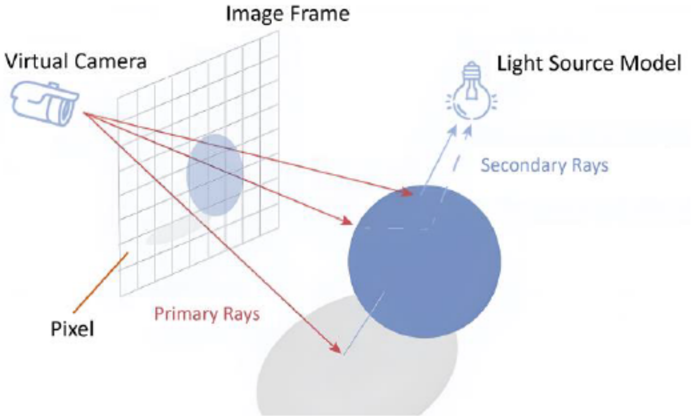


Figure 2.23: Backwards Ray Tracing Illustration (Adapted from Eclat Digital (2024))

2.2.4-2.2 Path Tracing

This is also known as Monte Carlo path tracing, which is considered a subset of ray tracing that relies on the Monte Carlo method for rendering. Path tracing initially utilises stochastic sampling where primary rays are randomly output (generated via probability distribution) in different directions (non-uniformly spaced) from each respective pixel (Cook 1989; Subr 2008; Mardaljevic 2011). Considering the use random generation of rays, a significantly larger number of samples per pixel (SPP) are required comparative to ray tracing, to ensure proper coverage of the virtual environment, as facilitated by the use of hundreds to thousands of rays (AMD 2024). For the generation of secondary ray from an intersection point, path tracing again employs stochastic sampling, where only a single successive ray is randomly selected from possible alternatives, as opposed to multiple rays generated in ray tracing (Whitted 2020), as illustrated in Figure 2.24.

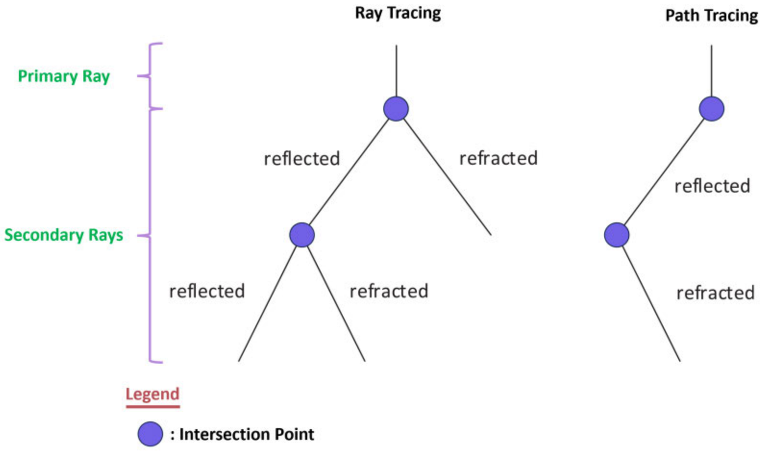


Figure 2.24: Ray Tracing vs Path Tracing Secondary Ray Secondary Ray Paths (Adapted from Whitted (2020))

After samples for each pixel have been collected, a Monte Carlo estimator is used to average the contributions that will be assigned to the corresponding pixels (Lambers n.d.; SGVR Lab n.d.). The benefit of the Monte Carlo method for the tracing of rays is improved accuracy and more faithfully accounts for the randomness of physical lighting behaviour, which allows for higher lighting fidelity compared to ray tracing, as observed in Figure 2.25 (Eclat Digital 2024; Modelo 2024). However, it is also not a perfect solution with the following disadvantages:

- Rendering of images is computationally intensive and time consuming, compared to ray tracing, with large quantity of SPP required for sampling and convergence to an ideal image quality (to be clarified further) (Rotenberg 2017; SGVR Lab n.d.)

- Difficult to simulate diffuse inter-reflection between surfaces, caustics (specular reflections from a surface that then strikes a diffuse surface resulting an a concentrated amount of light in an area), and high specular reflection (Rotenberg 2017; Nichols 2024; Imperial Collage n.d.).

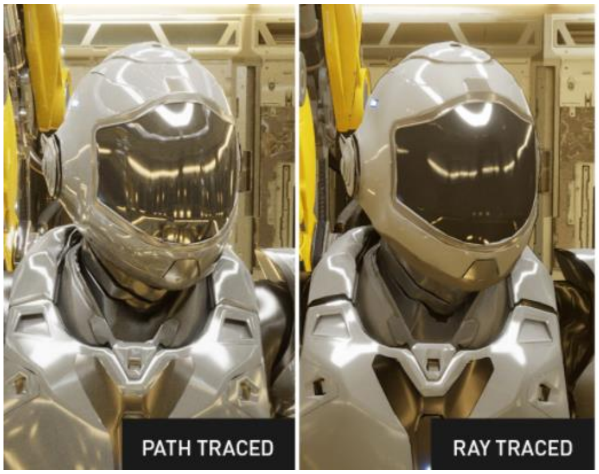


Figure 2.25: Path Tracing vs Ray Tracing (Adapted from Caulfield (2019a))

2.2.4-3 Render Quality Factors

The following controlling factors dictate rendering image quality achieved by both ray tracing and path tracing. The first four listed factors are user-adjustable within rendering engine settings.

- **Samples per Pixel:** This determines the number of rays that are cast from an image frame pixel. Increased SPP in effect increases the amount of lighting information collected to enhance pixel quality and thereby the photorealism of the rendered image, as well as reduces noise artifacts, which are a result of insufficient ray coverage within the virtual environment (AMD 2024; Hain n.d.). Park and Baek (2021) who use their own custom-built real-time ray tracing rendering engine, clearly demonstrate the impact of increased SPP, with a subsequent reduction of noise artifacts and improved render quality, as observed from visual inspection of their rendered images of a suit case in Figure 2.26, using SPP values ranging from 1 SPP to 100 SPP. Considering the results of the rendered images, Park and Baek (2021) makes an observation that noise is exacerbated for simulation of diffuse reflection scatter (omni-directional light scattering due to light interaction with rough or uneven surfaces) for low SPP counts. Additionally, from these results it was also demonstrated that increases in SPP resulted in a marked increase in the rendering time, as observed with a rise from 3ms to 336ms over 1 SPP to 100 SPP. Therefore, a trade-off exists between the increased computational demands from more SPP and the quality of the final image render.

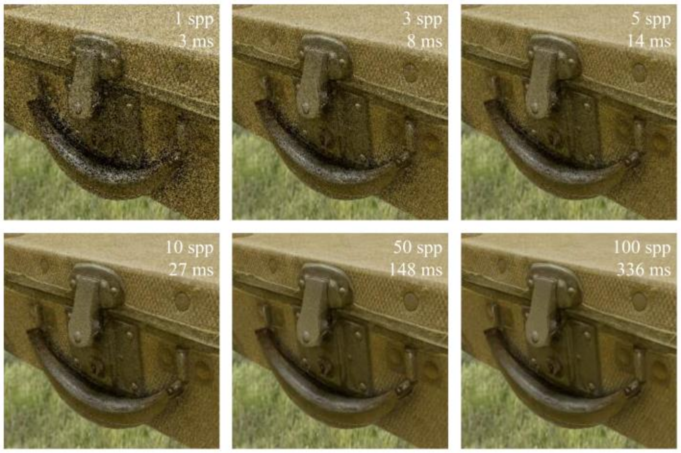


Figure 2.26: Relationship Between Increased Samples per Pixel (SPP) and Render Quality (Park & Baek 2021)

- **Number of Ray Bounces:** This determines the number of times cast rays are permitted to reflect or refract before termination, and is critical for the generation of accurate indirect lighting and by extension GI (Martin 2017; Blender 2024). Research by Vasiou et al. (2018) found that the majority of contributions to render quality by ray bounces are most effectual in the initial few bounces. In their investigation, they used the TRaX real-time ray tracing engine and tested up to nine ray bounces for rendering small and large city environments. It was found that four ray bounces were sufficient to achieve photorealistic renders, before diminishing returns were observed in terms of expended computational energy, which was measured using compute execution time, CPU stall time, memory stall time and other metrics.

- **Virtual Environment Complexity:** Larger amounts of detail within a virtual environment including but not limited to: Number and type of lighting, number and size of objects in association with assigned type of materials and textures, etc. (3S Cloud Render Farm 2023).

- **Resolution:** Increased pixel dimensions of images obtained from the virtual camera require more time to render, since an increased quantity of pixels will constitute an image frame for rendering (vice versa) (iRender 2020; 3S Cloud Render Farm 2023).

- **Rendering Engine/Software:** Differences in rendering engine including but not limited to available settings, parameters, implemented rendering algorithms, which contribute to its capabilities, and can lead to some providing comparatively better or poorer results in terms of render quality (iRender 2020).

- **Hardware:** Computer hardware is generally demanded by rendering engines to ascertain the best rendering results, including the latest high-end CPUs and GPUs. These perform the computational backend to support and accelerate ray tracing calculations for faster and accurate renders (iRender 2020; 3S Cloud Render Farm 2023).

2.2.4-4 Offline and Real-Time Rendering

It was noticed that most of the reviewed literature does not make explicitly clear the type of rendering approach that the rendering engine employed – either offline and real-time rendering, and furthermore if rendering techniques of ray tracing or path tracing was utilised. These are critical pieces of information that have been omitted, since these considerations offer different render qualities.

2.2.4-4.1 Offline Rendering

Offline rendering consists of offline path tracing and offline ray tracing. Offline path tracing is more heavily adopted, typically used for DCC applications in film, television and visualisation (i.e. advertising, architecture, etc.). Achieving offline renders to a high image quality generally comes at the expense of longer rendering times, ranging from minutes to hours (depending on desired results), while shorter rendering times generally sacrifice image quality (Caulfield 2019a; Keller et al. 2019; Chaos 2023; VNTANA 2023; Epic Games 2024f; NVIDIA 2024a). Traditionally this was done by the support of CPUs with the aid of high-capacity random-access memory (RAM). CPUs perform necessary rendering calculations, while RAM enables the storage and management of larger amounts of graphical data (i.e. materials, textures, lighting, object polygon meshes, etc.) relevant to the render of the environment. Resultantly, offline path tracing, provides superior results for rendering physically precise, photorealistic image frames, with high environmental complexity, resolution and lighting fidelity (Archxstudio 2024; Autodesk 2024; Epic Games 2024f; Lenovo 2024a). It should be noted that with current improvements to VRAM, GPUs can also perform offline rendering with engines, however do not perform as well as CPUs for rendering highly detailed scenes as well as simulations requiring a lot of memory (Autodesk 2024). Attention should be drawn to the fact that offline rendering requires more user technical skill and adjustment to produce desired lighting results (Epic Games 2023; LinkedIn 2024).

2.2.4-4.2 Real-Time Rendering

Currently, real-time rendering only includes ray tracing, with real-time path tracing being an emerging field at the time of writing. This rendering method accelerates ray tracing calculations with the use of GPUs (typically NVIDIA *RTX* series or AMD *Radeon* series), which employ dedicated video random-access memory (VRAM), which is a type of high-speed, low-capacity memory for storage and management of graphical data. Additionally, GPUs employ multi-core parallel processing to perform faster and more efficient rendering operations (graphics processing); thereby in combination with VRAM, GPUs are optimised to undertake the computational demands of high-quality renders. Although, due to the lower memory capacity of GPUs using VRAM, as opposed to CPUs using RAM, it provides comparatively lower accuracy for rendering photorealistic image frames. However, real-time rendering should not be easily dismissed, as rendering engines can achieve photorealistic results 'out-of-the-box', and even obtain competitive results with offline rendering by optimisation of settings and parameters, e.g. lighting, shadows, post processing effects, particle effects, anti-aliasing, etc., though it will compromise real-time performance, and often be to the detriment of other settings and parameters (Archxstudio 2024).

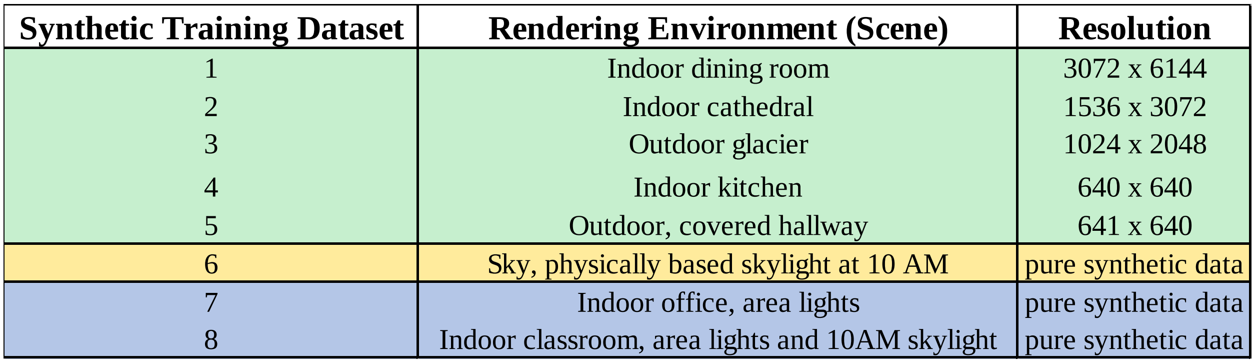
It should be clarified that for real-time rendering, imagery is generated instantaneously with respect to the view of the virtual camera, instead of long render times provided by offline rendering. Hence, the utilisation for real-time ray tracing has primarily been for video games, which prioritise fast rendering times of images approximating 30 to 60 frames per second, to interact seamlessly within the virtual environments, and require varying levels of photorealism, detail and resolution (Caulfield 2019b; VNTANA 2023; Autodesk 2024; Lenovo 2024a). These advantages have also been capitalised by rendering engines, with real-time rendering capabilities offered for time-effective DCC.

2.2.5 Photorealism: Lighting Fidelity

Considering this underlying lighting background theory, this prompts enquiry into whether lighting fidelity of synthetically generated training data can make a difference in further closing a remaining domain gap for highly accurate CV model performance, since lighting fidelity is a part of achieving photorealism. Presently, there is an outstanding literature gap of works that have directly explored this question, with only research by Zhang, Jia and Ivrissimtzis (2020), which makes these works highly valuable in this regard.

The investigation by Zhang, Jia and Ivrissimtzis (2020) involved a demonstrative problem requiring the CV model to detect and extract watermarks consisting of a 20 × 20-bit array (matrix) of semi-spherical bumps on the surface of 3D printed square plates within real-world images. This was achieved using a generated confidence map that encoded the probability that an image pixel was part of a bump or background, which was then used for a thresholding operation to retrieve the watermark matrix. Synthetic data consisting of 3D models of these printed square plates within various virtual environments was specifically used to train algorithms to detect the bumps via a confidence map. The relevancy of their work is derived from the research focus being on the impact of how different lighting sources employed within the virtual environment, which offer different lighting fidelities, can affect the quality of synthetic training imagery and thereby performance of a trained CV model.

Zhang, Jia and Ivrissimtzis (2020) diverges from most other research with the use of the *Mitsuba* rendering system used to generate synthetic training data. *Mitsuba* is a rendering engine, oriented towards research purposes, which is modularised to support a variety of customisation, more than standard rendering engines, with control over implemented algorithms for ray tracing, path tracing GI and PBR, render quality factors, as well as control over the use of CPUs of GPUs for backend computation and much more (Jakob 2013; Mitsuba 2024). The synthetic training data consisted of eight different datasets, which were distinguished by three types of lighting sources deployed within the virtual environment (not to be confused with light source models) that were used for illumination, as listed below – refer to Figure 2.27.



Synthetic Training Dataset	Rendering Environment (Scene)	Resolution
1	Indoor dining room	3072 x 6144
2	Indoor cathedral	1536 x 3072
3	Outdoor glacier	1024 x 2048
4	Indoor kitchen	640 x 640
5	Outdoor, covered hallway	641 x 640
6	Sky, physically based skylight at 10 AM	pure synthetic data
7	Indoor office, area lights	pure synthetic data
8	Indoor classroom, area lights and 10AM skylight	pure synthetic data

Figure 2.27: Zhang, Jia and Ivrissimtzis (2020) Synthetic Datasets [Edit]

- **Synthetic Training Datasets 1 - 5 (Green):** The first type of lighting source was provided by light probes, which are a type of real-world 360° HDR image that offer IBL (Calian et al. 2018). For individual datasets, the light probes captured natural lighting within various indoor and outdoor environments at different resolutions and were imported into the *Mitsuba* virtual environment to create high-fidelity synthetic background lighting. This in effect formed Type 3 Hybrid Synthetic Data. Refer to Figure 2.27.

- **Synthetic Training Dataset 6 (Yellow):** The second type of lighting source simulated natural lighting from the virtual environment itself, and in effect formed pure synthetic data. The scene was set in an outdoor environment with clear sky, and was devoid of any objects. Additionally, this dataset was rendered using path tracing, using an algorithm proposed by Preetham, Shirley and Smits (1999). Refer to Figure 2.27.

- **Synthetic Training Datasets 7 - 8 (Blue):** These datasets focused on simulations with artificial lighting sources within indoor environments. The first dataset utilised only ceiling point lights within the scene, whereas the second used ceiling point lights, as well as natural lighting from windows. Refer to Figure 2.27.

Since synthetic data generated for Synthetic Training Dataset 6 used path tracing, it can be assumed that it utilised an offline rendering approach, however for Synthetic Training Datasets 1 - 5 and Synthetic Training Datasets 7 - 8, Zhang, Jia and Ivrissimtzis (2020) does not clarify the exact rendering method (ray tracing or path tracing) or approach (offline or online) used for these virtual environments. Given the large customisability of the *Mitsuba* rendering engine, this is difficult to contextualise, other than stating that a non-path tracing method was utilised for generated data using Synthetic Training Datasets 1 - 5 and Synthetic Training Datasets 7 - 8. However, this gives further insight into the impact of lighting fidelity for synthetically generated data, by assessing influence of offline path tracing compared to an alternative rendering method and approach.

Zhang, Jia and Ivrissimtzis (2020) generated each dataset using the following strategies which are worth highlighting:

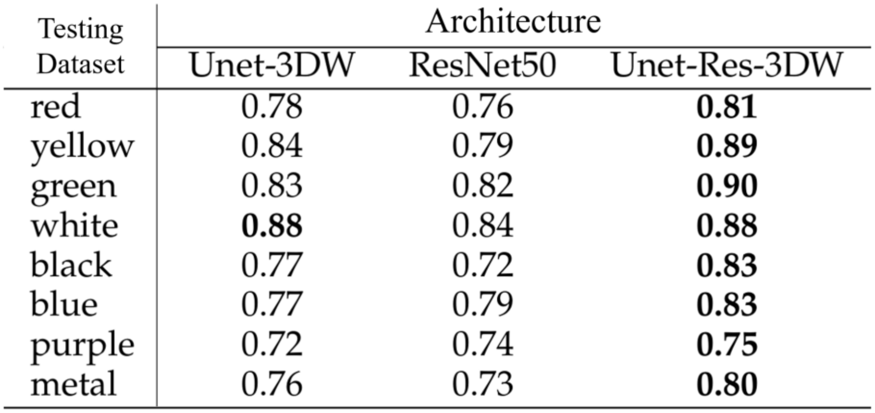
- **3D Model Usage:** It was expected that differences in the geometry between the training and test datasets would contribute significantly to poor detection performance and compromise validity of the results. Since Damian et al. (2023) takes a similar approach in using 3D identical models for training and test datasets, this is a strategy worth taking note.

- **Virtual Camera Parameters:** Orientation of the virtual camera view of the target object was randomised in terms of distance, d, elevation, θ, and azimuth ϕ for set domain values, e.g. $d \in [50\ cm,\ 100\ cm]$, $\theta \in [45°, 90°]$ and $\phi \in [0°, 360°]$. This strategy was also similarly followed for the generation of the test dataset to ensure domain consistency.

- **3D Model Textures and Materials:** Texture with associated material and colour for the 3D models of square plates were varied from a selection of 100 from online sources.

For fair testing of these datasets in the real-world, Zhang, Jia and Ivrissimtzis (2020) takes an interesting approach that merits acknowledgement. Preliminary experimental training and testing of three types of CNN detection architectures was initially completed to determine an ideal detection algorithm to be used for actual training and testing, as opposed to training and testing these three architectures separately and then comparing the results, which would have been a more standard experimental approach that also ensures fairness of results. The following three architectures were included, which hold respective strengths and weaknesses: ResNet50 – provides low noise predictions, Unet-3DW – a custom architecture by Zhang, Jia and Ivrissimtzis (2020) based on the U-Net architecture, which has good localisation capability (therefore is more effective at extracting local features), however provides predictions with more noise, and Unet-Res-3DW – a custom architecture by Zhang, Jia and Ivrissimtzis (2020) that is a combination of the former two algorithms, where the early layers utilised U-Net for low-level feature extraction, whereas later layers utilised ResNet50 for high-level feature extraction. This in effect combined the best aspects of ResNet50 and Unet-3DW. These architectures were trained on each synthetic dataset, which were comprised of 2,000 generated images, to create eight models for each architecture.

Eight real-world testing datasets were then applied to these eight trained models of each architecture – seven of the testing datasets included 3D printed plastic plates varying in dark and light colours, and one with a 3D printed metal plate. Each test dataset consisted of 20 real-world images captured under strong ambient light within indoor and covered outdoor environments with the camera being at distances of 1m. This was done to understand if real-world reflective surface properties, including colour and roughness of a target object, can significantly impact detector generalisation after being trained on synthetically generated data. For each of the three architectures, the eight models were tested on each of the eight real-world testing datasets, with the watermark detection results obtained in terms of F1 scores. The F1 scores from the eight models for each real-world dataset were averaged separately (therefore a combined total of 16,000 training images contributed to an averaged F1 score), which provided the following preliminary results presented in Figure 2.28, and gives several key insights as listed below:



Testing Dataset	Architecture		
	Unet-3DW	ResNet50	Unet-Res-3DW
red	0.78	0.76	**0.81**
yellow	0.84	0.79	**0.89**
green	0.83	0.82	**0.90**
white	**0.88**	0.84	**0.88**
black	0.77	0.72	**0.83**
blue	0.77	0.79	**0.83**
purple	0.72	0.74	**0.75**
metal	0.76	0.73	**0.80**

Figure 2.28: Zhang, Jia and Ivrissimtzis (2020) Average F1 Scores for Watermark Detection on 3D Printed Square Plates (Preliminary Experimental Results) [Edit]

- The most prominent observation from these results is that the Unet-Res-3DW architecture consistently performs better than the other architectures with results on average ranging between 75% - 90%, compared to Unet-3DW, ranging from 72% - 88%, as well as ResNet50 ranging between 72% - 79%. Zhang, Jia and Ivrissimtzis (2020) does not elaborate on this further but considering that the images consisted of bumps that provided subtle low-level image features (the detection focus), as well as prominent high-level features provided by the overall 3D printed plate; these results are indicative of the potential of hybrid neural network architectures to improve the performance of CV models (Lin, Guo & Aberer 2017). This is because the respective strengths and weaknesses of each architecture can be leveraged and strategically implemented within the greater architecture, at certain layers, to enhance the hierarchical feature abstraction process for both high-level and low-level features – seen with better performance from the Unet-Res-3DW architecture. In comparison, single architectures are limited in this flexibility – therefore lower performance was seen in ResNet50 and more specifically Unet-3DW, since it was more specialised towards low-level feature extraction.

- Zhang, Jia and Ivrissimtzis (2020) does not provide a comprehensive interpretation of these preliminary results other than making a brief note that the three architectures unexpectedly demonstrated good detection results for white and black 3D printed plastic plates as well as the 3D printed metal plates. It can be assumed that their concern was based on the idea that these represent the extremes of reflective surface properties that may impact feature extraction performance for the bumps. Taking a deeper review of their observation, it should be clarified that all real-world object surfaces have various reflective properties, and is dependent on two factors, the roughness and colour:

- **Surface Roughness:** This factor impacts the irregularities on an object surface, which affects reflection type, either being diffuse and specular, which impact the scatter characteristics. For surfaces with more microscopic irregularities/roughness, diffuse reflections occur when incident light interacts with a surface, which causes light to scatter in all directions; whereas for a surface that is smooth, where microscopic surface elements are typically oriented in the same direction as the surface, specular reflections occur, which causes light to scatter in a more concentrated range of directions (i.e. metals, mirrors, polished surfaces, etc.) (Dynamic Graphics Project n.d.; Lehtinen & Durand n.d.) – refer to Figure 2.29.

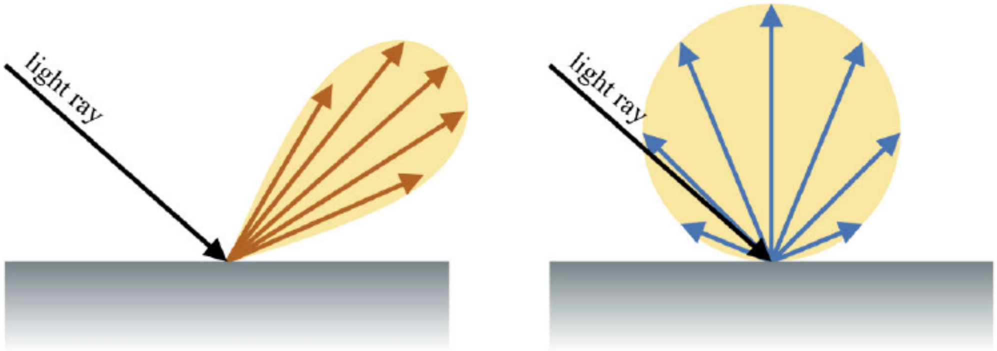


Figure 2.29: Specular Reflection vs Diffuse Reflection (Park & Baek 2021)

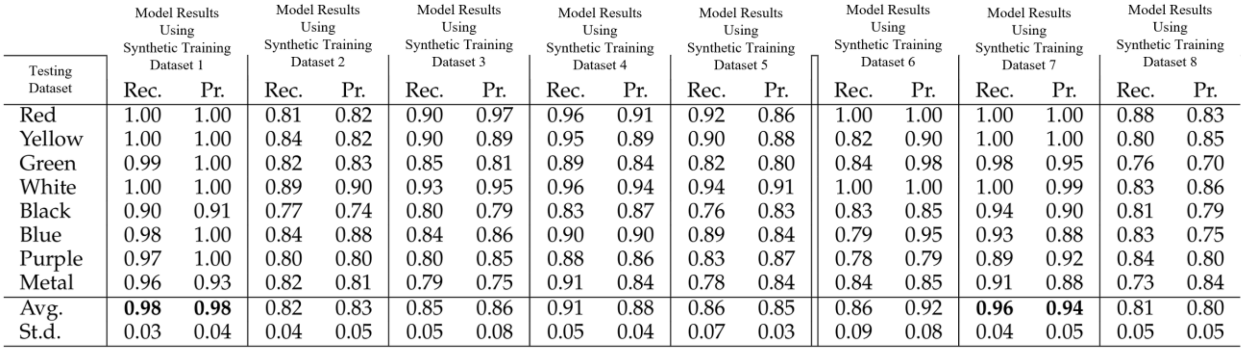
All object surfaces exhibit a combination of both types of reflection types, with some displaying one more dominant than the other. In the case of this experiment, the 3D printed plastic plates likely primarily generated diffuse reflections, whereas the 3D printed metal plates primarily generated specular reflections.

- **Surface Colour:** The perception of colour is dependent on the wavelength of light reflected from an object surface (Evident Scientific 2024). Dark surface colours absorb more incident light and reflect less, while light surface colours reflect more incident light and absorb less.

***Note:** Surface colours and surface roughness are considered PBR properties.

Referring to Figure 2.28 preliminary results, since the average F1 detection scores for the test datasets are both reasonably high and within relatively similar ranges amongst each architecture, the results are a critical indication that on average, reflective surface properties of real-world target objects generally do not have a major impact on detection performance of CV models trained with synthetically generated data. Following from this observation, it demonstrates that in general, the simulated fidelity of material properties facilitated by PBR algorithms for synthetically generated data (inclusive of both path tracing and non-path tracing rendering methods), is essential for robust CV model feature extraction in the real-world data with varied reflective surface properties. This agrees with the findings of Movshovitz-Attias, Kanade and Sheikh (2016). However, it should be noted that the average F1 score results observed are likely to have been influenced positively by a combination of domain randomisation and the averaging of the F1 scores.

Zhang, Jia and Ivrissimtzis (2020) provides additional value with the main experiment, which takes a similar approach to the preliminary experiment. Unet-Res-3DW, as the best performing architecture from the preliminary experiment was trained on each synthetic dataset, consisting of 2,000 generated images, to create eight corresponding models. A test dataset comprised of 20 real-world images was applied to these models respectively, with results quantified in terms of precision and recall. The average of the according results is presented in Figure 2.30, and provides important findings involving synthetic training data lighting fidelity, as listed below, which are not explored by Zhang, Jia and Ivrissimtzis (2020):



Testing Dataset	Model Results Using Synthetic Training Dataset 1		Model Results Using Synthetic Training Dataset 2		Model Results Using Synthetic Training Dataset 3		Model Results Using Synthetic Training Dataset 4		Model Results Using Synthetic Training Dataset 5		Model Results Using Synthetic Training Dataset 6		Model Results Using Synthetic Training Dataset 7		Model Results Using Synthetic Training Dataset 8	
	Rec.	Pr.	Rec.	Pr.	Rec.	Pr.	Rec.	Pr.	Rec.	Pr.	Rec.	Pr.	Rec.	Pr.	Rec.	Pr.
Red	1.00	1.00	0.81	0.82	0.90	0.97	0.96	0.91	0.92	0.86	1.00	1.00	1.00	1.00	0.88	0.83
Yellow	1.00	1.00	0.84	0.82	0.90	0.89	0.95	0.89	0.90	0.88	0.82	0.90	1.00	1.00	0.80	0.85
Green	0.99	1.00	0.82	0.83	0.85	0.81	0.89	0.84	0.82	0.80	0.84	0.98	0.98	0.95	0.76	0.70
White	1.00	1.00	0.89	0.90	0.93	0.95	0.96	0.94	0.94	0.91	1.00	1.00	1.00	0.99	0.83	0.86
Black	0.90	0.91	0.77	0.74	0.80	0.79	0.83	0.87	0.76	0.83	0.83	0.85	0.94	0.90	0.81	0.79
Blue	0.98	1.00	0.84	0.88	0.84	0.86	0.90	0.90	0.89	0.84	0.79	0.95	0.93	0.88	0.83	0.75
Purple	0.97	1.00	0.80	0.80	0.80	0.85	0.88	0.86	0.83	0.87	0.78	0.79	0.89	0.92	0.84	0.80
Metal	0.96	0.93	0.82	0.81	0.79	0.75	0.91	0.84	0.78	0.84	0.84	0.85	0.91	0.88	0.73	0.84
Avg.	**0.98**	**0.98**	0.82	0.83	0.85	0.86	0.91	0.88	0.86	0.85	0.86	0.92	**0.96**	**0.94**	0.81	0.80
St.d.	0.03	0.04	0.04	0.05	0.05	0.08	0.05	0.04	0.07	0.03	0.09	0.08	0.04	0.05	0.05	0.05

Figure 2.30: Zhang, Jia and Ivrissimtzis (2020) Precision and Recall for Watermark Detection on 3D Printed Square Plates (Main Experimental Results) [Edit]

- Average results of the model trained using Synthetic Training Dataset 1 (Indoor dining room) which employed a light probe as the lighting source, indicated the best performance at 98% precision and 98% recall, in comparison to the results of models using Synthetic Training Dataset 6 - 8, which employed lighting sources derived from the virtual environment itself. This was expected since simulated lighting is directly based on measurements from a real-world scene, as opposed to relying solely on approximated rendered lighting behaviour by path tracing and or ray tracing methods.

  Further to note is that the light probe used for Synthetic Training Dataset 1 had the highest resolution at 3072 × 6144 pixels, in comparison to Synthetic Training Datasets 2 - 5, and models trained with Synthetic Training Dataset 1 exhibited superior results. This shows that IBL resolution is a crucial factor for improved model performance, which is a logical point, since a higher resolution would capture a larger amount of lighting information with direct and indirect lighting. When this is implemented in a virtual environment in tandem with object material properties, it translates to a higher simulated fidelity of lighting interactions with improved illumination of low-level environmental details, such as the plate bumps for the synthetic training data. With the use of Unet-Res-3DW architecture, the model was better able to learn to extract lower-level features, which were made available with this improved lighting in the training data, which thus led to improved hierarchical feature abstraction and thereby model generalisation performance with real-life data.

  This again agrees with the findings of Movshovitz-Attias, Kanade and Sheikh (2016) and demonstrates how lighting and material properties support each other and the importance of the simulated fidelity of these properties (high-fidelity photorealism) as facilitated by GI and PBR; however further underscores the critical impact of lighting fidelity as the proxy for photorealism.

- Attention should also be focused on the average results of models using lighting sources derived from the virtual environment itself (Pure Synthetic Data). It is worth noting that on average the model using Synthetic Training Dataset 6 (Sky, physically based skylight at 10 AM), which was rendered using path tracing, had a 92% Precision and 86% Recall, whereas on average the model using Synthetic Training Dataset 7 (Indoor office, area lights), which was rendered using non-path tracing methods, indicated a higher performance at 94% Precision and 96% Recall. This is despite the capability of path tracing to produce comparatively more photorealistic training data with a higher fidelity of lighting, which clearly demonstrates that synthetically generated training data, rendered with non-path tracing methods, as well as Pure Synthetic Data, can achieve competitive results for real-world CV model applications, as supported by the fact that these results are second highest observed in this experiment.

- The virtual environments provided by Synthetic Training Dataset 1 (Indoor dining room) and Synthetic Training Dataset 7 (Indoor office, area lights), which had the better performing models, were well-lit scenes with ambient light elements, highly similar to the test dataset environment, which enabled improved generalisation results. Whereas some of the other synthetic training datasets provided lighting environments that had considerable differences, which contributed to lower performance. A prime example is the average results of the model trained with Synthetic Training Dataset 8 (Indoor classroom, area lights and 10AM skylight), where major differences in the amount and type of lighting sources, compared to the test data environment likely resulted in a large domain shift. This is because CV detection models learn to extract critical features in the source distribution of the training data, and when these are not present in the target domain, this causes a domain shift, which can exacerbate poor feature extraction and thereby performance (Mukherjee et al. 2021). Considering that low-level features in images are sensitive to the adjustments in illumination, this significant domain shift led to the average results of models trained with Synthetic Training Dataset 8 (Indoor classroom, area lights and 10AM skylight) being the lowest at an 80% Precision and 81% Recall. This is a strong indication that the source domain of the training dataset and the target domain of the testing dataset should ideally have a high similarity in terms of lighting conditions to minimise a domain shift and that the fidelity lighting (encompassing direct and indirect lighting) is an important aspect of CV feature extraction that can assist in the minimisation of a domain shift.

- Considering the extremely high average performance values for models trained using Synthetic Training Dataset 1 (Indoor dining room) and to a lesser extent Synthetic Training Dataset 7 (Indoor office, area lights) had the better simulated lighting, compared to the average results of models using other training datasets, this is a strong indication that high fidelity lighting for synthetic training data can also contribute to closing the synthetic-to-real gap.

This research by Zhang, Jia and Ivrissimtzis (2020) serves as a good entry point to understanding the impact of lighting fidelity and by extension material properties for synthetically generated data, although there are limitations to the investigative approach which detract from a more detailed understanding, as listed below:

- Since a consistent virtual environment was not used for the synthetically generated training datasets, a fair comparison cannot be made on the impact of factors such as, path tracing versus non-path tracing methods, differences in IBL resolutions and the effectiveness of different types of lighting sources.

- Considering the use of various indoor and outdoor virtual environments, this investigation could have been extended to provide a comparative understanding of the impact of lighting sources for each of these scenes respectively.

- Although a low volume of training data with 2,000 images was used for the synthetic training datasets, both training and testing images were captured close to the target object, which likely led to improved detection conditions, therefore the results are likely not representative of model performance at varied distances under lighting conditions presented by the types of lighting sources. However, the improved results at lower training volumes, do ostensibly support the observation made with the work of Movshovitz-Attias, Kanade and Sheikh (2016), that a high-level of photorealism for synthetic training data can reduce the amount of training data required, provided that sufficient domain randomisation is utilised for the generated training data. However, a noted caveat to this observation is that both training and test images were captured close to the target object in the work of Zhang, Jia and Ivrissimtzis (2020) and Movshovitz-Attias, Kanade and Sheikh (2016). This suggests a revision to this observation that a high-level of photorealism for synthetic training data can reduce the amount of training data required, provided that sufficient domain randomisation is utilised for the generated training data and that test images are captured close to the target object.

2.3 Unreal Engine 5 Capabilities

Considering that to the best of current knowledge, relevant details documenting major UE5 capabilities for its implementation in CV related applications is unavailable in literature, this section will provide a high-level overview of UE5 capabilities that are deemed important to the research questions. Both real-time ray tracing and offline path tracing rendering methods can be deployed in the UE5 rendering engine. Real-time ray tracing is the most utilised rendering method for DCC using this engine, considering its cutting-edge technologies that enable the rapid development of high-quality and photorealistic digital content. Therefore, the focus of reviewed details for UE5 will be with respect to real-time ray tracing.

2.3.1 Unreal Engine 5 Important Terms

Project: A folder structure that holds all necessary files for the development of an Unreal Engine project.

Level: A level is established within a project which is the virtual world (environment) that is developed by a user (Epic Games 2024at).

Level Editor: The UE5 development environment which has interfaceable settings and tools that permit creative control over a level design and its virtual world environment in real-time (Epic Games 2024au).

Content: This is a broad term for resources known as assets, that are implemented in a level and contribute to the design and development of the virtual world, as listed below (Epic Games 2024as, 2024ac):

Unreal Engine Content		
2D Assets	3D Models	Animations
Atlases	Audio	Brushes
Decals	3D Virtual Environments	Game Systems
Game Templates	HDRI Backdrops	Material & Textures
Smart Assets	Tools & Plugins	Visual Effects (VFX)
	User Interface (UI)	

Actors: Objects that are placeable within a level. By default these support 3D transformation properties such as translation, rotation and scaling (Epic Games 2024aq). These also have adjustable functions, properties and settings as per its assigned component (Epic Games 2024ap).

Components: These are functions, properties and settings assigned to actors, which can grant them complex capabilities, including AI, lighting, physics, camera perspective view, etc (Epic Games 2024ap, 2024ao). Components are also assigned to actors via the UE5 *Blueprint* system (Epic Games 2024an).

Viewport: This is a window within the engine user interface (UI), which allows users to visualise and interact with the virtual environment during level development, which is made possible with real-time rendering. A viewport can be configured to view a level by different types, layouts and modes to assist with the level development process, with Perspective View being the default 3D view used to operations (Epic Games 2024am).

Blueprint: Blueprint is a programmable feature and system which allows the creation and modification of actor components using a visual scripting system or directly using C++, by either two methods (Epic Games 2024al, 2024an):

- **Class Blueprint:** Used to define a single actor.
- **Level Blueprint:** Used to define multiple actors, as well as control and coordinate their respective interactions with other actors, and or other blueprints within the level.

Material Editor: Allows user to edit material and texture and material properties for actors (Epic Games 2024ak).

Cine Camera Actor: An actor with real-world camera capabilities that allows for the capture of visual data within the virtual environment for offline rendering of imagery (virtual camera). Renders can be completed using ray tracing or path tracing (Epic Games 2024a).

Plugin: Extension software to Unreal Engine that can be enabled or disabled locally within the *Editor* for individual projects, to supplement project development. Some major uses for plugins include, but are not limited to, modifications to existing engine features or the addition of new ones, extending the capabilities of the *Editor* with new menus, tools, and sub-modes, the creation of new file types, etc. Plugins can be officially developed by Epic Games or developed by third-party developers, which can be downloaded from the Unreal Engine Marketplace (Epic Games 2024aj).

Movie Render Queue: Plugin tool that generates high quality offline renders of visual data captured by the Cine Camera Actor (Epic Games 2024ai).

High Dynamic Range Image (HDRI) Backdrop: This is a 360°, HDR image which can be projected as a background to provide IBL to an UE5 virtual environment (Epic Games 2024ah).

Unreal Engine Marketplace: The Unreal Engine Marketplace, renamed as *Fab* as of writing this paper, is an online store (by Epic Games) with free or paid content made by third-party developers that can be downloaded and imported into UE5 to supplement project development (Epic Games 2024ag, 2024ac).

2.3.2 Unreal Engine 5 Lumen

Lumen is a new lighting system by UE5, which is stated as a GI and reflections system (note that the reflections component of the system is PBR) that uses ray tracing methods. Although GI is available for many rendering engines, *Lumen* distinguishes itself with the first production-ready, real-time, dynamic GI and reflection system. Dynamic GI and reflections compute the intensity of light reflected from a point on object surfaces within a virtual environment in real-time, which facilitates instant and adaptive lighting changes during rendering, for adjustments made to a scene, e.g. modified light sources and objects – thereby is more user-friendly with 'out-of-the-box' results during development of a virtual environment for DCC. *Lumen* is a highly accurate GI and reflection system, attributed to the fact that calculations for lighting interactions encompass the entire virtual environment, with the computation of an infinite amount of ray bounces during ray tracing operations, which replicates natural lighting interactions. The alternative is baked GI, which pre-computes surface lighting within a virtual environment to be stored as a light map, that is then overlayed on object textures and materials for incorporation during rendering. This is applicable only for offline rendering and requires that the scene be baked again for every scene change, which can be time consuming for complex virtual environments. Furthermore, working with baked GI is less user-friendly for beginners, requiring in-depth experience to adjust lighting (Unity 2017; Luksch, Wimmer & Schwärzler 2019; Unity 2019a; Epic Games 2023; Adobe 2024h; Epic Games 2024av).

2.3.3 Hardware Ray Tracing vs Software Ray Tracing

In UE5, the *Lumen* system can be set to utilise hardware ray tracing or software ray tracing as per the specifications of the user computer system performing rendering operations:

2.3.3-1 Hardware Ray Tracing

Relies on computational acceleration by GPUs, which supports higher fidelity of lighting simulation by *Lumen* and ultimately a higher visual quality for ray traced renders (Epic Games 2023). For this type of ray tracing, the process is as described in Section 2.2.5-2.1 (Epic Games 2024af; NVIDIA 2024b).

For UE5 hardware ray tracing, the recommended GPU for a computer system is an NVIDIA RTX-2000 series and higher, or AMD RX 6000 series and higher, with DirectX 12 (DX12) support. DX12 is a communication Application Programming Interface (API), that enables applications (apps) to take advantage of the graphics processing capabilities of computer systems equipped with one or more GPUs. However, these GPUs must be DX12-compatible, which are typically newer generation GPUs (Microsoft 2021; Epic Games 2024ae).

Note:** This is enabled when the ***Use Hardware Ray Tracing when available and ***Support Hardware Ray Tracing*** settings are checked within UE5 settings (Epic Games 2024h).

2.3.3-2 Software Ray Tracing

This ray tracing process does not rely on acceleration from ray tracing capable GPUs (facilitated by DX12 support), instead it is a less computationally intensive ray tracing process that allows ray tracing operations to be completed with older GPUs that do not provide ray tracing capabilities – primarily GPUs with DirectX 11 (DX11) support. This ultimately allows for software ray tracing to be used with a wider range of computer systems (Epic Games 2023, 2024af). Software ray tracing is made possible with the use of a *Mesh Distance Field*, a 3D grid generated by voxelisation, which converts polygonal meshes (continuous geometric structures), that form 3D objects, into discrete volumetric elements known as voxels, which in effect forms a discretised spatial representation of the scene to be rendered. This is achieved by voxels measuring and storing the minimum distance from any point in the rendered scene, to the closest point on any present object mesh surface. This distance is determined with transformation calculations using coordinate values from the mesh vertices (Jones 1995; Jones & Satherley 2001; Bærentzen & Aanæs 2002; Ridzuan et al. 2022; NVIDIA 2024b; PixCap 2024). A Mesh Distance Field ultimately allows for a more computationally efficient ray tracing process that can skip through empty space, since the distance to the nearest surface is known in prior, which reduces the number of steps required to find the ray intersection points (Epic Games 2024ad).

Moreover, for software ray tracing, *Lumen* generates a Surface Cache, which is a lower resolution data structure that is a parameterisation of all surfaces within a scene, that precomputes and stores material properties of each object mesh surface determined by PBR, as well as direct and indirect lighting incoming and outgoing to the surface, as simulated by GI and PBR. The Surface Cache is layered over surfaces in a scene, which allows simulated rays during software ray tracing to rapidly reference the precomputed lighting interactions with objects and simulate appropriate light bounce from the ray intersection points during ray tracing operations. In combination, these aspects of software ray tracing significantly reduces the computational overhead for rendering, compared to hardware ray tracing, however, includes some technical limitations, with the most relevant being listed as follows (Epic Games 2024af, 2024ab; NVIDIA 2024b; Tatzgern et al. 2024):

Note:** This is enabled when the ***Use Hardware Ray Tracing when available and ***Support Hardware Ray Tracing*** settings are unchecked within UE5 settings (Epic Games 2024h).

2.3.3-3 Software Ray Tracing Technical Limitations

- In general, hardware ray tracing produces superior accuracy for rendered lighting and visual fidelity compared to software ray tracing. This difference is clearly displayed in Figure 2.31, with a noticeable difference surface reflection and to a lesser extent refraction. This is because during the sampling process for hardware ray tracing, rays pass through a virtual environment to directly intersect with the object mesh, which offers a higher resolution of data to sample GI and PBR lighting interactions. In contrast, for software ray tracing, the traced rays refer to the Surface Cache to perform sampling, which is a data structure that stores the GI and PBR lighting interactions of an object mesh at a lower resolution to improve rendering efficiency and thereby produces a lower accuracy (Epic Games 2023, 2024h, 2024af; NVIDIA 2024b).

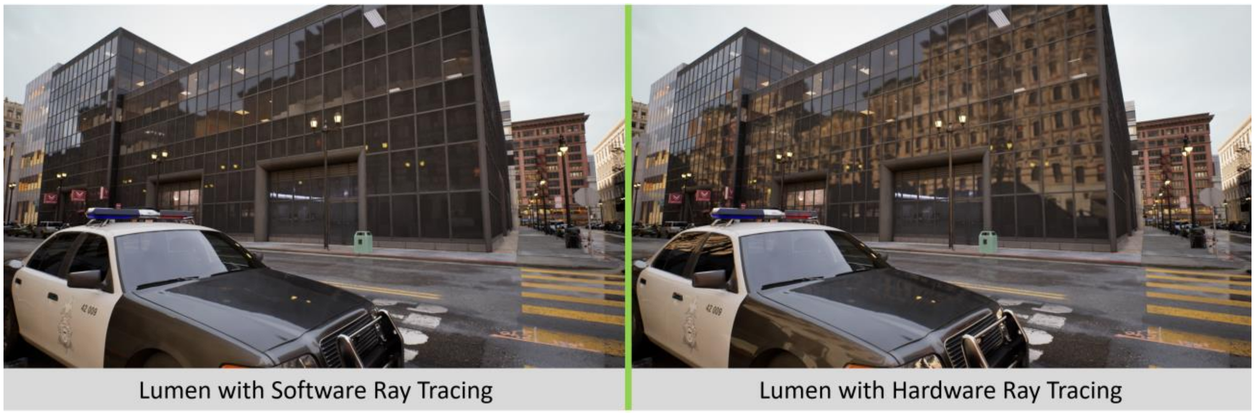

Lumen with Software Ray Tracing
Lumen with Hardware Ray Tracing

Figure 2.31: Unreal Engine 5 Lumen Global Illumination System using Hardware Ray Tracing vs Software Ray Tracing (Adapted from Epic Games (2024h))

- Software ray tracing has difficulty representing Mesh Distance Fields of extremely thin and large objects (Epic Games 2024af).

- A 3D model that is imported in the rendering engine in a very small size and then scaled up, will result in an insufficient resolution for the Mesh Distance Field to accurately represent the object surface properties (Epic Games 2024af).

2.3.4 Unreal Engine 5 Shadowing Methods

Shadows are a critical component that compliment GI, adding photorealistic attributes to a lit virtual scene by providing a sense of depth and space to an illuminated virtual environment. Typically shadows are calculated with GI, however, UE5 offers five types of separate shadowing methods, two in particular provide the highest quality for rendered shadows (Epic Games 2024aa):

- **Virtual Shadow Maps:** This is the default shadowing method for UE5. Virtual shadow maps are conceptually ultra high-resolution shadow maps used by UE5 as part of a shadow mapping process, which are 16,384 × 16,384 pixels used for an entire scene. However, a virtual shadow maps are sectioned into tiles of 128 × 128 pixels known as pages, which are deployed as required during scene rendering of shadows and thus imposes reduced computation and memory costs compared to ray traced shadows (discussed below). One of the major technical limitations of virtual shadow maps is that they exhibit better performance on object meshes with high polygon counts, and display undesirable artifacts for object meshes with low polygon counts (Epic Games 2023, 2024h, 2024z).

- **Ray Traced Shadows:** This standalone method of shadowing uses hardware ray tracing which produces shadow renders with more accurate visual fidelity than virtual shadow maps. The distinction is particularly observed with ray traced shadows being able to render sharper/harder contact shadows the closer a shadow casting object is to a receiving surface, and softer shadows the farther a shadow casting object is from the receiving surface. An example of this difference in rendering capability between ray traced shadows and virtual shadow maps is demonstrated in Figure 2.32 with a rendering of a light pole (Epic Games 2024aa, 2024h).

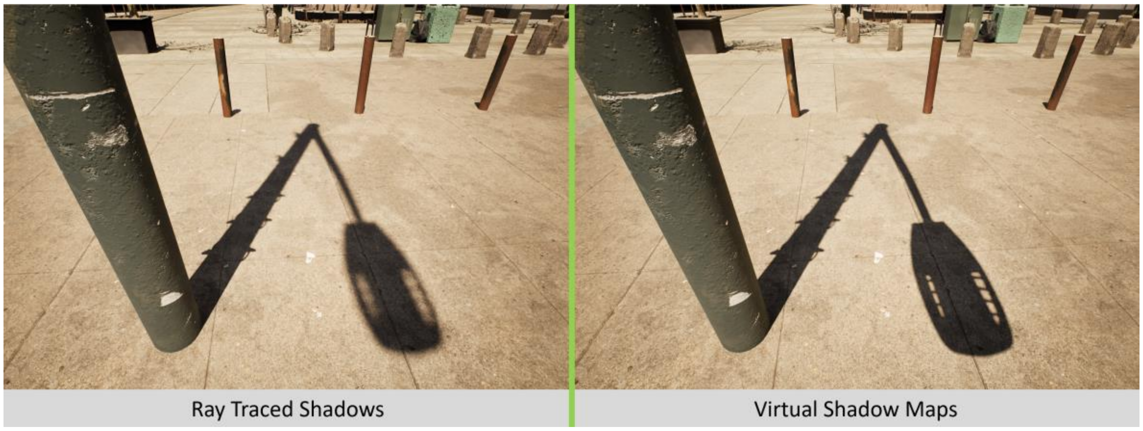


Figure 2.32: Ray Traced Shadows vs Virtual Shadow Maps (Adapted from Epic Games (2024h))

Another example of distinctions in shadow rendering accuracy between ray traced shadows and virtual shadow maps is observed in terms of the sharpness and softness of shadows with variations in light source size and angles. Figure 2.33 demonstrates these differences for generated shadows using different directional lighting source angles, where it can be observed that ray traced shadows is better able to render complex interactions of shadowing phenomena with hard and soft shadows. This is noticeable with better hard shadow definition from the model tree trunk and branches and soft shadow definition from the leaves (Epic Games 2024h).

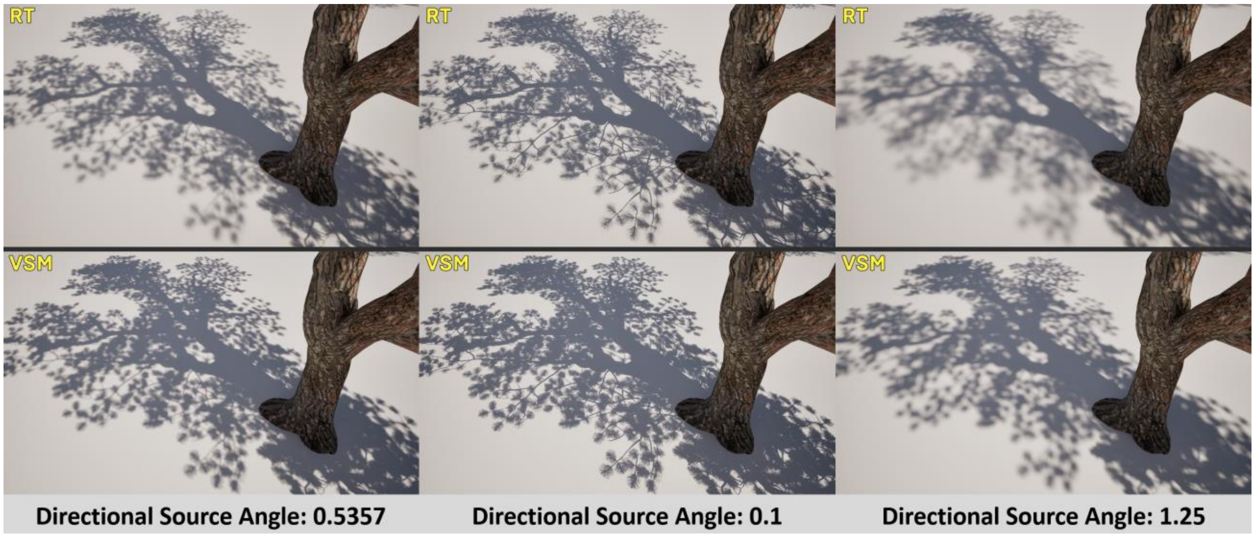
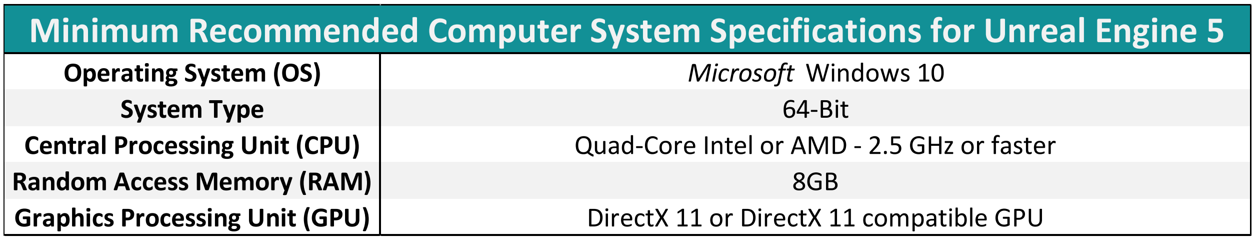

RT
RT
RT
VSM
VSM
VSM
Directional Source Angle: 0.5357
Directional Source Angle: 0.1
Directional Source Angle: 1.25

Figure 2.33: Generated Shadows for a Tree Model using Ray Traced Shadows (RT) vs Virtual Shadow Maps (VSM) Over Various Directional Lighting Source Angles (Adapted from Epic Games (2024h))

2.3.5 Unreal Engine 5 Scalability Settings

The following computer system specifications are the minimum recommended for both hardware and software ray tracing – refer to Figure 2.34 (Epic Games 2024y).



Minimum Recommended Computer System Specifications for Unreal Engine 5	
Operating System (OS)	*Microsoft* Windows 10
System Type	64-Bit
Central Processing Unit (CPU)	Quad-Core Intel or AMD - 2.5 GHz or faster
Random Access Memory (RAM)	8GB
Graphics Processing Unit (GPU)	DirectX 11 or DirectX 11 compatible GPU

Figure 2.34: Minimum Recommended Computer System Specifications for Unreal Engine 5

Renders of higher image quality produce higher computational loads with increased graphics processing requirements (vice versa). Even if the computer system of a user meets the minimum requirements, it may not be able to meet demands of higher quality renders due to the inadequacies of available hardware.

Therefore UE5 scalability settings are provided, ranging from low to cinematic, in order to adjust the quality level of various rendered properties within the UE5 engine, in order to accommodate UE5 use by various computer systems (Epic Games 2024b). Figure 2.35 displays UE5 scalability settings (Epic Games 2024ab):

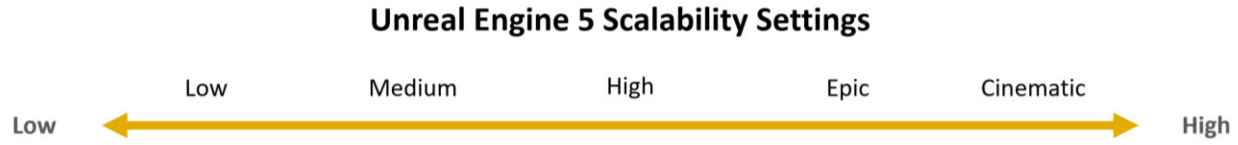


Figure 2.35: Unreal Engine 5 Scalability Settings (Epic Games 2024b)

The rendered properties that are adjusted by UE5 scalability settings are listed below – eleven in total. Note that adjustments to the scalability settings correspondingly impact the level of photorealism provided by UE5.

- **View Distance:** Objects can be removed based on their distance to the viewer, including Cine Camera Actor and Viewport (Epic Games 2024b).

- **Anti-Aliasing:** Adjusts the quality of Anti-Aliasing selected method (Epic Games 2024b). Aliasing is a rendering artifact produced by the limited resolution of digital displays, commonly appearing as jagged edges for objects with smooth curves and diagonal edges relative to an image. This effect is a result of digital displays which use a grid of discrete pixels to represent details of an image. Thus digital displays have finite pixel densities, which produce visible pixelation and decreased sharpness for rendered objects with smooth curves and diagonal edges. Anti-aliasing is a rendering technique that can be employed, to minimise the aliasing effects by sampling multiple points around the edges of objects in an image to determine the average colour values, which are then used to effectively blur the edges of objects, to reduce visible pixelation and thus create appearance of smoother edges (Cabading 2024; Lenovo 2024b). For reference Figure 2.36 illustrates the effects of aliasing and anti-aliasing for rendered geometric objects.

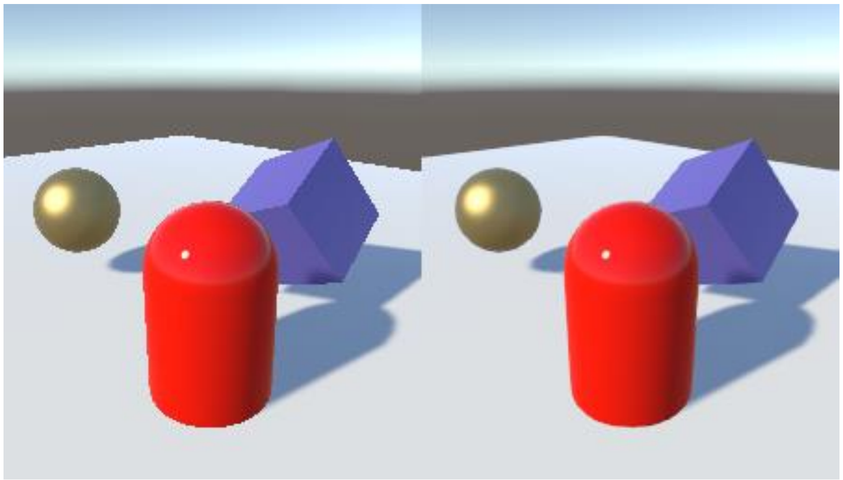


Figure 2.36: Effects of Aliasing (Left) vs Anti-Aliasing (Right) for Rendered Geometric Objects (Unity 2016b)

- **Post Processing:** This encompasses a wide range of effects mostly tied to camera properties, that enhance the visual realism of a rendered scene and are applied prior to a complete render of a scene (Epic Games 2024x). A list of the most noteworthy effects that are changed with adjustments to the scalability settings are provided below:

  - **Motion Blur:** Reproduces the visual effect where a fast-moving object appears blurry for imagery captured by real-world cameras (sensor-realistic property) (Houdini n.d.).

  - **Ambient Occlusion:** This is a shadowing method additional to the primary methods described in Section 2.2.6-3, which reproduces soft shadows generated from ambient lighting (sensor-realistic property) (Hoberock 2007; Plural Sight 2014).

  - **Depth of Field:** Reproduces the effect where certain parts of an image appear blurred when not in the focus of a real-world camera, which provides a sense of distance between near and far objects (sensor-realistic property) (Epic Games 2024w).

  - **Lens Flare:** Reproduces the real-world phenomenon when a strong source of light interacts with a camera lens, causing light to scatter with visible artifacts to appearing in captured imagery (sensor-realistic property) (Epic Games 2024v).

  - **Colour Fringe:** Reproduces the real-world phenomenon of chromatic aberration/colour fringing, which is a distortion of image colours when a camera lens fails to converge all light wavelengths of incoming light to the camera sensor focal point, which is explainable as follows. Light is refracted as it passes between two transparent mediums of differing densities, which alters the direction of light. During this process dispersion occurs, where variation in the refractive index of the incoming medium for different wavelengths, causes wavelengths to refract at different angles, and hence exhibit a noticeable separation of visible light per constituent colours. In the case of light passing between the mediums of air to a camera lens, dispersion contributes to the inability of cameras to precisely focus light on the camera sensor focal point, which results in the camera sensor capturing these wavelength separations, to ultimately produce colour artifacts outlining object edges, as seen in Figure 2.37 (sensor-realistic property) (Tunney 2023; Adobe 2024g; Image Engineering 2024; City University of Hong Kong n.d.).

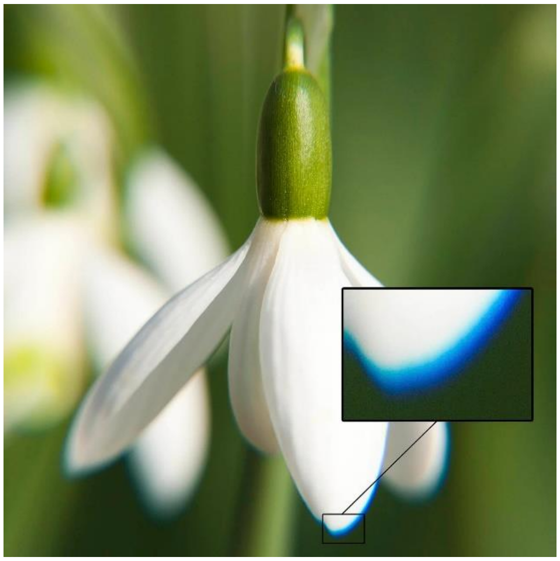


Figure 2.37: Chromatic Aberration Artifact (Jkk 2013)

- **Eye Adaptation:** Reproduces automatic exposure features of modern real-world cameras, where the exposure of a scene is automatically adjust when transitioning from darker environment to a lighter environment (vice versa) (sensor-realistic property) (Epic Games 2024u).

- **Bloom Quality:** Reproduces imaging artifact of real-world cameras of glow around lights and reflective surfaces (sensor-realistic property) (Epic Games 2024ar).

- **Shadows:** Both virtual shadow maps and ray traced shadows can be adjusted. A list of the most noteworthy sub-properties that are changed with adjustments to the scalability settings are provided below (Epic Games 2024b):

  - **Light Function Quality:** Adjusts the quality of light functions used to simulate lighting.

  - **Shadow Quality:** Adjusts the quality of rendered shadows.

  - **Shadow Maximum Resolution:** Adjusts the maximum resolution that can be used.

  - **Shadow Distance Scale:** Adjusts the rendered distance of shadows.

- **Global Illumination & Reflections:** Associated scalability settings impact the quality provided by *Lumen* system, as listed below (Epic Games 2024ab):

  - **Cinematic:** Scalability setting is intended for high quality offline renders with Movie Render Queue.

  - **Epic:** Scalability setting targets a 30 frames per second rendering speed.

  - **High:** Scalability setting targets a 60 frames per second rendering speed.

- **Medium and Low:** Scalability settings disable the *Lumen* GI and reflections, instead resorting to Screen Space Global Illumination (SSGI) and Screen Space Reflections (SSR), which performs GI and reflection calculations for what is visible on screen in a scene, which means it does not account for objects and lighting outside the Viewport or Cine Camera Actor vision. To clarify, SSGI and SSR are less accurate compared to the *Lumen* GI and reflections, since *Lumen* performs calculations accounting for lighting interactions within the whole scene, even if not visible to the Viewport or Cine Camera Actor (Shinsoj 2023; Epic Games 2024t, 2024ab).

Textures: This adjusts three sub-properties associated with the render of textures, listed as follows (Epic Games 2024b):

To understand the effects of the first two properties it should be understood that UE5 uses a texture streaming system (streamer) that dynamically loads object mesh textures into a scene as required during a render and dynamically regulates the level of detail (LOD) for each object mesh texture (resolution), using mipmaps. These are cached and down sampled version of the original high-definition textures; a higher mipmap LOD is used for objects closer to the View Port and the Cine Camera Actor, and lower mipmap LOD are used for more distant objects. Texture streaming in effect allows for efficient GPU memory management, which is particularly important since textures contribute a large portion of memory usage during rendering operations and can reduce computational and rendering performance if unchecked. The texture streaming pool is a designated amount of memory used to store textures used by the texture streaming system (Unity 2019b; Epic Games 2024s; Mower 2024; Unity 2024a).

- **Streaming Mip Bias:** A mipmap bias establishes the threshold for selection of mipmap LOD by a GPU. The Streaming Mip Bias is a global property that is applied to all mipmaps and is important in ensuring that the texture streaming system loads appropriately sized mipmaps that do not place an excessive GPU memory load (Epic Games 2024r).

- **Streaming Pool Size:** Adjusts the available pool size for textures in the engine (Epic Games 2024r).

- **Maximum Anisotropy:** Uses Anisotropic filtering, which improves the appearance of textures viewed at oblique angles (Intel 2024).

- **Effects:** Adjusts the quality of many different types of effects. A list of the most noteworthy effects is provided below (Epic Games 2024b):

  - **Translucency Lighting Volume Dimensions:** Adjusts the resolution of 3D textures (volume textures) that provide translucent material properties for Actors.

  - **Refraction Quality:** Adjusts the quality of rendered refractions.

  - **Screen Space Reflections:** Activates the use of SSR per the UE5 scalability setting.

  - **Detail Mode:** Adjusts the minimum level of detail that an Actor should be rendered at.

- **Material Quality:** Adjusts the use of high-quality or low-quality material properties.

- **Foliage:** Adjusts how many foliage meshes are rendered at one time for 3D vegetation objects (Epic Games 2024b).

- **Shading:** Shading Models are used as part of PBR to determine how a mesh material reflects incoming light (Epic Games 2024b, 2024q).

- **Landscape:** Adjusts texture and material resolution applied to landscape object meshes, as well as the LOD for the number of polygons used to represent the mesh surface based on distance to the Cine Camera Actor or Viewport (3D Studio 2024; Epic Games 2024p, 2024b).

2.3.6 Unreal Engine 5 Environmental Lighting

Environmental lighting is one of the main draw features of UE5, which provides the ability to create realistic virtual worlds, with high-fidelity natural lighting conditions. This is made possible with lighting components, which can model natural phenomena such as fog, clouds, sky lighting and atmospheric features and how light interacts with them. These components work in unison to determine light interactions globally within a virtual environment. The *Lumen* system in tandem with associated tools allow for dynamic and real-time modification of these components for rapid scene development, and the ability to set a variety of lighting conditions as desired by the user. A brief overview of some of the main lighting components provided by UE5 is presented below (Epic Games 2024o):

- **Exponential Height Fog:** A physically-based fog rendering system (model), which creates a diffuse lighting effect, which is a soft light that is scattered uniformly within an environment that contributes to the ambient light of a scene (Wronski 2014; Adobe 2024e). In a virtual environment, Exponential Height Fog component generates multi-layered fog with less density in high areas, increased density in low-lying areas and at distances further from the Viewport and Cine Camera Actor, as well as volumetric effects for light shafts (Epic Games 2024o, 2024d). Figure 2.38 compares an effect of Exponential Height Fog component when enabled and disabled in UE5 render.

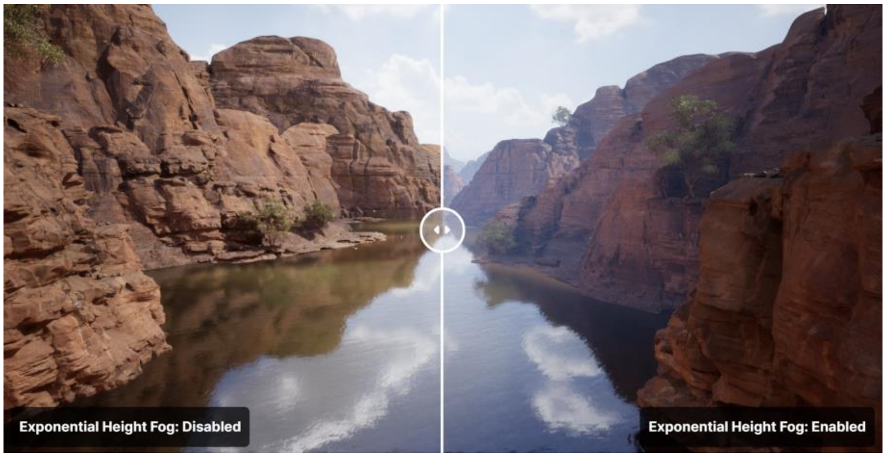

Exponential Height Fog: Disabled
Exponential Height Fog: Enabled

Figure 2.38: Exponential Height Fog Disabled vs Enabled (Epic Games 2024d)

- **Volumetric Clouds:** A physically-based cloud rendering system (model), which can create various cloud types, shapes, and effects within a scene, ranging from mostly clear skies to storm clouds.

Figure 2.39 provides examples of the range of possible cloud formations that can be rendered using Volumetric Clouds (Epic Games 2024e, 2024o).

- **Directional Light:** This is a parallel light source model used to simulate the sun and emanated sunlight within a scene. The angle of a directional light can be adjusted within a scene environment in relation to a simulated sun path, which allows for the simulation of time-of-day (Epic Games 2024n, 2024o).

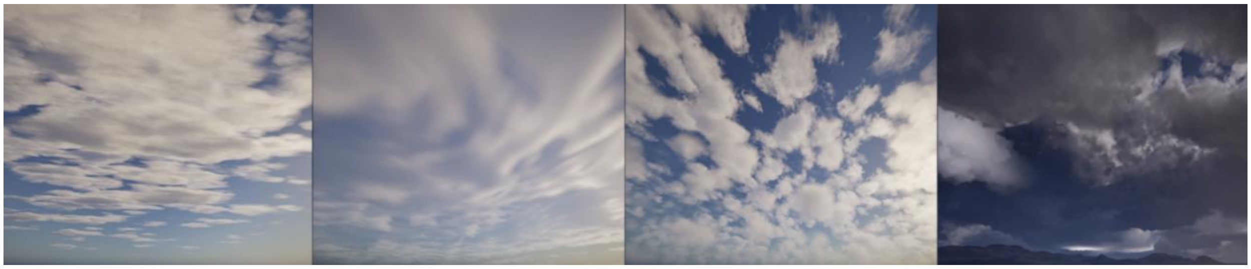


Figure 2.39: Examples of Volumetric Cloud Renders (Epic Games 2024e)

- **Sky Atmosphere:** A physically-based sky and atmosphere-rendering system (model), which simulates light scatter interactions with particles and molecules within the Earth atmosphere and considers the impact of other components such as Exponential Height Fog, Volumetric Clouds and Directional Light. Since the angle of the sun (Direction Light) is adjustable, the modelled scattering effects by the Sky Atmosphere component, enable the sky to dynamically change colours depending on the altitude of the sun to replicate times during day and night. Figure 2.40 illustrates renders using Sky Atmosphere. Furthermore, the lighting interactions facilitated by the Sky Atmosphere, Volumetric Clouds and Exponential Height Fog facilitates the accurate representation of the Directional Light as a sun disk, as opposed to an artificial lighting source (Epic Games 2024c, 2024n, 2024m, 2024o).

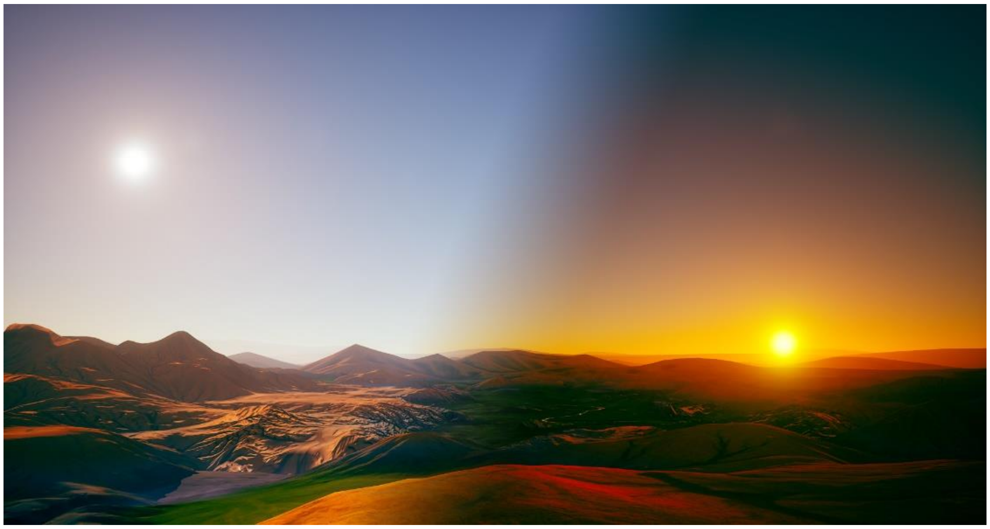


Figure 2.40: Example of Sky Atmosphere Renders (Epic Games 2024c)

- **Sky Light:** Models lighting in distant parts of virtual environment sky, and matches this lighting with the Sky Atmosphere model (Epic Games 2024l, 2024k, 2024o).

2.3.7 Unreal Engine 5 Light Source Models

UE5 provides additional types of lights that can be implemented in a virtual environment, typically used for artificial, local environmental lighting (Epic Games 2024k):

- **Point Light:** A point light source, which acts a light bulb to cast light in all directions from a single point.

- **Spot Light:** A point light source, which emits in a single direction.

- **Rectangular Area Light:** Provides planar lighting emitted from a rectangular plane, to provide a distributed light source for ambient illumination.

2.3.8 Unreal Engine 5 Physics-Based Materials

Physics-based materials are assigned to object meshes, replicating the lighting interaction response of a physical material as would be in the real-world, within the virtual environment. Physics-based materials incorporate the following components (Epic Games 2024aw, 2024j):

- **Base Colour:** Controls the colour and texture of an object mesh.
- **Roughness:** Controls how rough or smooth a material appearance is, defined by how sharp or blurry object mesh surface reflections appear.
- **Metallic:** Controls object mesh surface reflections ranging from non-metallic to metallic appearance.
- **Specular:** Controls how much specular light the object mesh surface reflects.

2.4 Literature Review Main Findings Summary and Conclusion

2.4.1 Literature Review Summary

From this literature review it is realised that closing the synthetic-to-reality gap is multifaceted and is dependent on a variety of strategies that ultimately contribute to higher model generalisability and robustness of a CV model, with some being more significant than others. From the assessed works, an overview of the main findings concerning the effectiveness of strategies used in generating synthetic training data and the resulting impact on model generalisation and robustness when employed to train a CV mode are presented below:

- **Domain Randomisation:** This is the primary contributor to model generalisation. The randomisation of image parameters to a level that would be expected in the real-world was found to be acceptable. All other strategies listed appear to complement domain randomisation and can further assist in the minimisation of the synthetic-to-reality gap. Domain randomised lighting (and by default including shadows) plays a major role for model performance compared to other randomised parameters.

- **Large Volume Training Datasets:** Training dataset volumes should optimally be in the tens of thousands and domain randomised. However, over a certain quantity, depending on the model, before diminishing returns are observed.

- **Synthetic Image Realism:** Associated properties should be randomised with the use of large volume training datasets; these should not be use independently as a strategy:
  - **Photorealism:** Higher fidelities of lighting (and by default including shadows) achievable with high resolution IBL can significantly boost generalisation performance with real-world data, which indicates great potential to majorly contribute to closing the synthetic-to-reality gap. This is resultant of improved lighting fidelity in terms of direct and indirect lighting, with the support of material properties (facilitated by GI and PBR), enabling the CV model to learn to extract lower-level features made available with improved lighting, which leads to improvement in hierarchical feature abstraction and CV model generalisation. Separately, photorealistic elements including material properties and textures contribute minor performance gains for model generalisation as these contribute low-level feature information that can be extracted by a CV model. Increased levels of photorealism plausibly appear to reduce the amount of training data required, when a sufficient amount of domain randomisation is used for the generated training data and test images are captured close to the target object.
  - **Sensor-Realistic Properties:** Noise added to synthetic imagery was found incur minor performance improvements, although further investigation into the impact of other sensor-realistic effect is necessary.

Additional strategies which optimise CV model enhance model generalisation and robustness are presented below:

- **Use of Pretrained CV Models:** Model pretraining facilitates transfer learning which accelerates learning efficiency and model convergence with actual training data.
- **Use of Hybrid Neural Network Architecture:** Combining the strengths of separate CNN architectures as part of a unified architecture, improves the hierarchical feature abstraction capability of a CV model.
- **Use of Type 2 Hybrid Synthetic Datasets:** The use of a mixed percentage of synthetic data and real-world data for training datasets forces the training algorithm over two different domains and abstract higher-level features that share commonality between both domains.

2.4.2 Literature Review Conclusion

In the context of this research report, the broad insight into these synthetic data generation strategies from this literature review is vital, since any advantage afforded to CV model detection performance of multi-rotor sUAS, no matter how minor, can serve to provide edge for mission-critical C-sUAS operations.

The most outstanding results indicated amongst these strategies is a combination of high-fidelity photorealism and a domain randomisation strategy, as presented in the work by Zhang, Jia and Ivrissimtzis (2020). As part of their work, CV models were trained with synthetically generated data, which employed high-fidelity lighting source derived from high resolution IBL (Type 3 Hybrid Synthetic Data), as well as randomised the target object surface texture, with associated material and colour, as well as the orientation of the virtual camera view which captured the synthetic imagery.

However, the primary issue with the use of Type 3 Hybrid Synthetic Data for synthetically generated training data, for multi-rotor sUAS detection applications, is that fact that the projected background images need real-world scenery to be captured. This contradicts the motivation for the use of synthetically generated data, which is to circumvent real-world restrictions regarding image acquisition – therefore, an alternative synthetic data type needs to be explored.

Zhang, Jia and Ivrissimtzis (2020) shows that Pure Synthetic Data, where the lighting source is derived from the virtual environment itself, is a suitable alternative that can achieve close results to Type 3 Hybrid Synthetic Data using high resolution IBL, however, the generation of Pure Synthetic training data is criticised for being time consuming, computationally intensive and a costly solution in terms of required hardware to achieve renders of a high fidelity photorealism. Although these are valid arguments, there are two important points supporting its use case:

1. Adequate interest by interested parties, i.e. academia, industry, law enforcement and militaries to invest in the potential effective solutions for C-sUAS.

2. Excluding Type 3 Hybrid Synthetic Data, the advantage of Pure Synthetic Data over other synthetic data types is that generated images are theoretically capable of maximising the potential of a rendering engine, with the capacity to accurately simulate lighting interactions to a high fidelity within the virtual environment. This is an important consideration since the interaction of simulated light with object surfaces within the virtual environment is the proxy by which photorealism is achieved in renders, by facilitating the representation of shadows and physical properties such as material and texture. Therefore, it is possible to achieve high levels of photorealism with Pure Synthetic Data, which can translate to improved CV model generalisation performance, when coupled with a domain randomisation strategy.

It is also important to underscore that the works of Zhang, Jia and Ivrissimtzis (2020) with the support of Movshovitz-Attias, Kanade and Sheikh (2016) both strongly suggest that high-fidelity photorealism of synthetically generated data in terms of rendered lighting and material properties can further close the synthetic-to-reality gap, when employed in tandem with a domain randomisation strategy.

With this consideration, the following general capabilities pertaining to the rendering engine are primarily responsible for the rendering of lighting and material properties and thereby the achieved fidelity of photorealism in synthetically generated data:

General Rendering Engine Capabilities	**Literature Review Reference**
Global Illumination (GI)	Section 2.2.4-1.2
Physics-Based Rendering (PBR)	Section 2.2.4-1.3
Shadowing	Section 2.2.4-1.4
Ray Tracing	Section 2.2.4-2.1
Path Tracing	Section 2.2.4-2.2
Render Quality Factors	Section 2.2.4-3
Offline Rendering	Section 2.2.4-4.1
Real-Time Rendering	Section 2.2.4-4.2

However, it was found that despite the capability of path tracing to produce photorealistic synthetic training data with the highest fidelity of rendered lighting and material properties, synthetic training data generated via alternative, non-path tracing methods can achieve competitive results for developing CV-models to be applied in the real-world. Therefore this leaves an opening for UE5 which primarily employs ray tracing for renders.

2.5 Knowledge Gap

High-fidelity photorealism of synthetically generated data, in terms of rendered lighting and material properties, has been indicated to make a significant difference in closing a remaining domain gap of a CV model trained with synthetically generated data using a domain randomisation strategy and can achieve highly accurate CV model performance. Movshovitz-Attias, Kanade and Sheikh (2016) and to a greater extent Zhang, Jia and Ivrissimtzis (2020) are among the only major works that support this idea, however, do not provide a direct investigation into the matter; therefore, this encourages further exploration into the advantages of high-fidelity photorealism of synthetically generate data – specifically Pure Synthetic Data.

UE5 has state-of-the-art general rendering engine capabilities that can generate high-fidelity photorealistic digital content, by ray traced GI, PBR and shadowing. However, to date no known research has assessed the performance impact of high-fidelity photorealism (enabled by these rendering engine capabilities) in synthetically generated training data using UE5, when employed in training a CV model and if it can further close the synthetic-to-reality gap, when employed with a domain randomisation strategy. In order to understand the performance impact of high-fidelity photorealism in synthetically generated training data using UE5, when it is employed in training a CV model, the performance impact of *Lumen* GI and reflections, as well as ray traced shadows will need to be investigated. This research project intends to address this literature knowledge gap in terms of a real-world application – multi-rotor sUAS detection, which will gain the most benefit from determined results for C-sUAS research and development using EO detection systems.

To guide the direction of this research in line with the question, the following research aim, and objectives have been determined:

2.6 Research Aim

Project Aim: To investigate the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data (Pure Synthetic Data), on CV models for the application of multi-rotor sUAS detection in the real-world.

2.7 Research Objectives

Objective 1: Collect real-world testing image datasets of multi-rotor sUAS under different lighting conditions with a sufficient distribution, that can be used in the assessment of CV models trained with synthetically generated data.

Objective 2: Generate and collect synthetic training images of multi-rotor sUAS under similar lighting conditions in UE5. For each of these lighting conditions, separate datasets will be generated with varying levels of *Lumen* global illumination and reflections, as well as ray traced shadow support.

Objective 3: Evaluate and compare the performance of CV models trained with synthetic datasets employing varying levels of *Lumen* global illumination and reflections, as well as ray traced shadow support, respectively for each lighting condition.

Chapter 3 – Methodology

3.1 Background Information

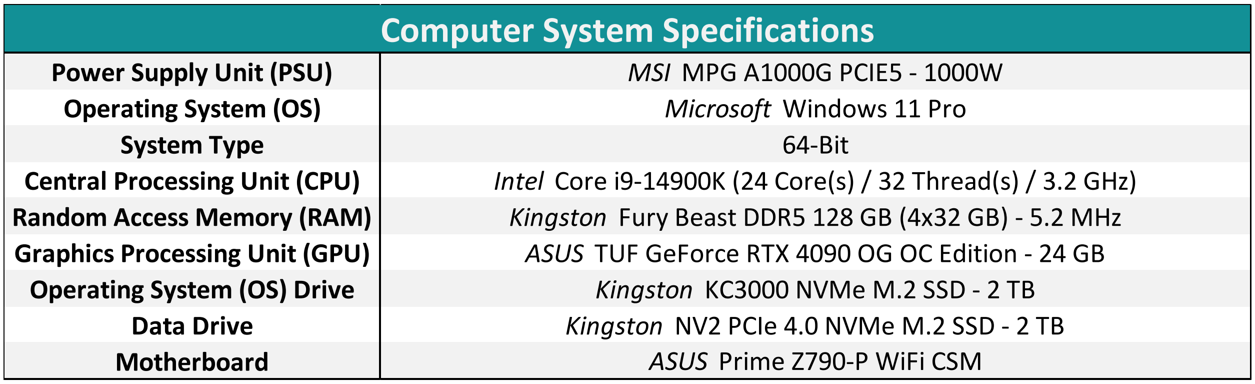
To address the stated aim and relevant literature knowledge gap, the following methodology was devised and undertaken for this experimental research project. The reader should acknowledge that there is no specific method to synthetic data generation and is highly depended on the problem at hand, as seen with previous works. Therefore, this methodology will be a novel approach, although, it should be clearly stated that it draws inspiration from prior research.

***Note:**

- All experimental work was conducted on a personal computer (PC) with access to the internet.
- Actions completed in this guide used Windows OS and Unreal Engine 5.4.4.

3.1.1 Computer System Specifications

Referring to the minimum recommended computer system specifications for UE5 outlined in Figure 2.34, the computer system utilised for this research report had the following specifications as presented in Figure 3.1. This can be stated as a high-end computer system, which was necessary with the use of hardware ray tracing in UE5. This was to ensure the best possible results that could be acquired in UE5 in terms of render quality and ensure reasonable render times. However, the incurred financial costing required to support this project can be considered a critical limitation for independent validation of results.



Computer System Specifications	
Power Supply Unit (PSU)	*MSI* MPG A1000G PCIE5 - 1000W
Operating System (OS)	*Microsoft* Windows 11 Pro
System Type	64-Bit
Central Processing Unit (CPU)	*Intel* Core i9-14900K (24 Core(s) / 32 Thread(s) / 3.2 GHz)
Random Access Memory (RAM)	*Kingston* Fury Beast DDR5 128 GB (4x32 GB) - 5.2 MHz
Graphics Processing Unit (GPU)	*ASUS* TUF GeForce RTX 4090 OG OC Edition - 24 GB
Operating System (OS) Drive	*Kingston* KC3000 NVMe M.2 SSD - 2 TB
Data Drive	*Kingston* NV2 PCIe 4.0 NVMe M.2 SSD - 2 TB
Motherboard	*ASUS* Prime Z790-P WiFi CSM

Figure 3.1: Computer System Specifications

***Note:** Refer to Appendix C for according settings that were set for the use of hardware ray tracing as part of this experimental research project.

3.2 Methodology

Considering that this research project intended to be a preliminary investigation, this experimental research project, was divided into two stages – Stage 1 and Stage 2. For each stage, multiple experiments were carried out, where for each experiment, respective tests were then completed. Experiments devised for Stage 2 were based on the collective findings of tests from Stage 1 experiments.

3.2.1 Stage 1: Experimental Design Overview and Context

In service of the aim of this research paper, to effectively assess the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data (Pure Synthetic Data), on CV models for the application of multi-rotor sUAS detection in the real-world, the following experimental design was developed for Stage 1. This section will provide an overview of the experimental design for the reader to contextualise concepts prior to detailed explanation of the approach during implementation in Section 3.2.2.

3.2.1-1 Source Domain and Target Domain Considerations

In this experimental research, the developed CV-based multi-rotor sUAS detection models used binary classification; therefore, all dataset samples (images) were to have a single airborne multi-rotor sUAS. This was decided as the best approach for this work, to simplify the findings derived from results, considering that this research was a preliminary investigation. The development of each CV-based multi-rotor sUAS detection model involved the use of training and validation datasets comprised of synthetically generated imagery of airborne multi-rotor sUAS obtained from the UE5 virtual environment. It is important to note is that this generated imagery was Pure Synthetic Data, with the lighting source derived from the UE5 virtual environment itself and additionally. The corresponding testing dataset of real-world images was collected from online sources.

3.2.1-1.1 Outdoor Environment

Since EO detection systems are primarily deployed in outdoor settings to visually monitor an airspace for multi-rotor sUAS threats. To enhance the practical value of findings in this investigation for C-sUAS research, it was considered ideal that for developing the CV-based multi-rotor sUAS detection models, both the source domain of generated synthetic training and validation imagery, as well as the target domain of training imagery will consist of multi-rotor sUAS within an outdoor environment.

Additional justification for the use an outdoor environment for synthetically generated data is provided below:

- Since a critical component and enabling factor of high-fidelity photorealism by rendering engines has been demonstrated to be high-fidelity lighting, it was determined as necessary for the investigation that a virtual environment can provide lighting conditions conducive to delivering the best expression of *Lumen* GI and reflections, as well as ray traced shadows, which can facilitate high-fidelity photorealism within UE5 renders. An outdoor virtual environment can be deemed a suitable choice, being able to provide natural lighting conditions, which can take advantage of UE5 environmental lighting, and simulate light interactions globally within a virtual environment. UE5 environmental lighting can provide well-lit conditions (high-level environmental illumination), as well as nuanced direct and indirect lighting, to facilitate high-fidelity lighting for rendered synthetic training and validation data. Therefore, potential CV-based multi-rotor sUAS detection models in training are capable of learning to extract lower-level features, which are made available with this enhanced lighting and with the support of UE5 physics-based materials within generated training image data. This in turn can improve hierarchical feature abstraction and thereby the performance of multi-rotor sUAS detection models.

- UE5 environmental lighting with its Exponential Height Fog, Volumetric Clouds, Directional Light, Sky Atmosphere and Sky Light features can simulate sky conditions and corresponding natural lighting conditions within a virtual environment. This allows the source domain of generated synthetic imagery of multi-rotor sUAS to closely replicate the target domain of real-world test images, which can minimise domain shift and enables more effective feature detection performance, and thus a reduced contribution to the synthetic-to-reality gap.

- Comparatively UE5 light source models, which provides artificial, local environmental lighting within a virtual environment, is unable to provide high-level environmental illumination, as well as nuanced direct and indirect lighting. Resultantly it is unable to match the fidelity of UE5 environmental lighting, which can potentially contribute a domain shift and lower model performance and therefore was discounted for use.

3.2.1-1.2 Image Viewing Angles

Given that EO detection systems are ground-based, the viewing angle of airborne multi-rotor sUAS generally range from the sides to the bottom, as seen in Figure 3.2. Therefore, in keeping with the desire to make findings of this work to have a practical value for C-sUAS research, all real-world and synthetically generated imagery of multi-rotor sUAS used for training, validation and testing were ensured to have similar viewing angles when acquired.

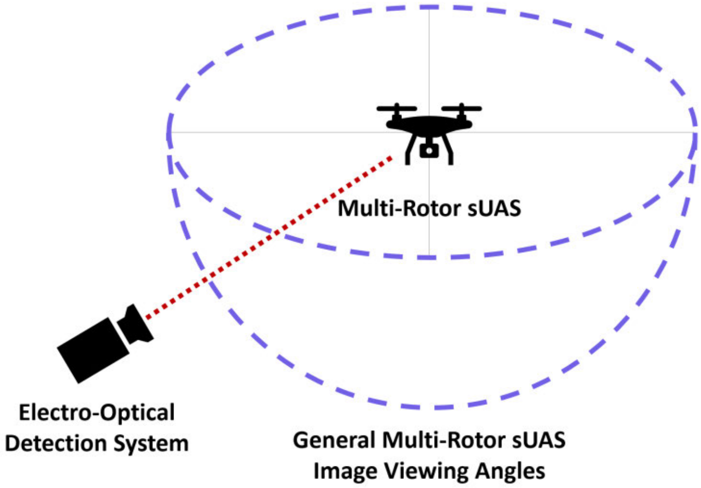

Multi-Rotor sUAS
Electro-Optical Detection System
General Multi-Rotor sUAS Image Viewing Angles

Figure 3.2: General Multi-Rotor sUAS Image Viewing Angles

3.2.1-2 Experiment Independent and Dependent Variables

A proper method of quantifying the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data, on CV-based multi-rotor sUAS detection models, can be stated to be challenging considering that there are numerous high-level controlling factors outlined in the literature review, as well as low-level controlling factors that were excluded from the literature review, as these were excessive in number to be practically covered and further accounted for within experiments of this research paper.

Drawing from the high-level controlling factors recognised from the literature review, the UE5 scalability settings was determined as the primary controller for UE5 *Lumen* GI and reflections, as well as ray traced shadows, which can adjust the degree of photorealism provided in UE5 image renders. Therefore, it was decided that the best way to gauge the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data, on CV-based multi-rotor sUAS detection models, was by comparison of model performances, when developed with synthetic imagery rendered using *Low* and *Cinematic* scalability settings. To reiterate these settings are defined as follows:

- **Cinematic:** Maximum scalability setting, which is intended for high quality offline renders

- **Low:** Minimum scalability setting, which disables *Lumen* GI and reflections, instead resorting to Screen Space Global Illumination (SSGI) and Screen Space Reflections (SSR), which performs GI and reflection calculations for what is visible on screen in a scene, which means it does not account for objects and lighting outside the Viewport or Cine Camera Actor vision. SSGI and SSR are less accurate compared to the *Lumen* GI and reflections, since *Lumen* performs calculations accounting for lighting interactions within the whole scene, even if not visible to the Viewport or Cine Camera Actor.

These two UE5 scalability settings (independent variables) were to be used for separate experiments that allowed comparison of model performance (dependent variable). These experiments are listed below with the intended results to be obtained:

- **Experiment 1:** Results of developed CV detection models using UE5 synthetically generated image data of multi-rotor sUAS under Cinematic settings.

- **Experiment 2:** Results of developed CV detection models using UE5 synthetically generated image data of multi-rotor sUAS under Low settings.

3.2.1-3 CV Model Types

For each experiment, different CV model types were planned for use since these provide different accuracies and speeds, due to a variety of factors such as the model size (which generally encompasses the number of hidden layers, number of parameters, used backbone architecture), the use of pretraining, datasets used for pretraining, set hyperparameter values, etc., which impact the learned values of *weights*, *w*, and *biases*, *b*, and ultimately the output performance of the final trained model. To ensure a fair test and eliminate the possibility of model bias impacting the detection performance and results, multiple CV model types were developed for Experiment 1 and Experiment 2.

3.2.1-4 Additional Source Domain and Target Domain Considerations

3.2.1-4.1 Lighting Conditions

Lighting conditions are a domain feature that can influence the level of illumination within an environment and thereby the features available to be learned and extracted from synthetic training and validation data, as well as real-world testing data. A high-level of environmental illumination allows improved CV model feature extraction, where as a low-level of environmental illumination challenges CV model feature extraction (Morawski et al. 2021; Mukherjee et al. 2021).

Therefore, to prevent the potential introduction of bias by utilising one type of lighting condition for image data used to develop CV model types for Experiment 1 and Experiment 2, an additional set of CV model types for Experiment 1 and Experiment 2 was developed with image data using alternative lighting conditions and background features (domain conditions).

As part of this consideration, day light and low light conditions were utilised as two extremes of lighting, which can be considered optimal lighting conditions with a high-level environmental illumination, as well as challenging lighting conditions with a low-level environmental illumination respectively. This further allowed an understanding into the performance impact of photorealism provided by UE5 synthetic image data on detection models operating under different ends of lighting conditions.

3.2.1-4.2 Background Features

Since lighting is the proxy for photorealism in renders, to ensure that experimental findings pertain to the impact of UE5 render photorealism, it was decided that the domains of generated synthetic training and collected real-world testing image datasets, were to include only natural lighting features in the background e.g. sun and clouds. No other features were included as these could act as potential distractors, which may negatively skew results.

With this context, to ensure consistency in the domain for training, validation and testing datasets that were used to develop CV model types under day light and low light conditions, a classification scheme for background features in an image was developed, as explained below and listed in Figure 3.3:

- **Level 0 Background Features:** Sun is the only background feature present with multi-rotor sUAS in image. Sun is positioned anywhere within image, including directly behind. No clouds are present.
- **Level 1 Background Features:** Clouds and sun are background features behind multi-rotor sUAS in image. Sun is positioned anywhere within image, including directly behind.

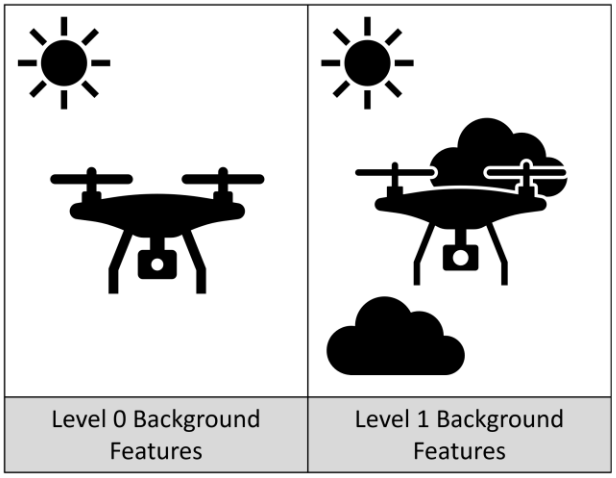


Figure 3.3: Level 0 and Level 1 Background Features

For clarification, it was decided that the following combination of lighting conditions and background features (domain conditions) for images with multi-rotor sUAS will be used for the development of CV model types for Experiment 1 and Experiment 2:

- Day Light Conditions / No Clouds / Level 0 Background Features
- Low-Light Conditions / Clouds / Level 1 Background Features

For reference, real-world images of multi-rotor sUAS with domain conditions of (Low-Light Conditions / Clouds / Level 1 Background Features) and (Day Light Conditions / No Clouds / Level 0 Background Features) are presented below in Figure 3.4 and Figure 3.5 respectively. It should be noted that these combinations were specifically chosen due to their prevalence amongst real-world images.

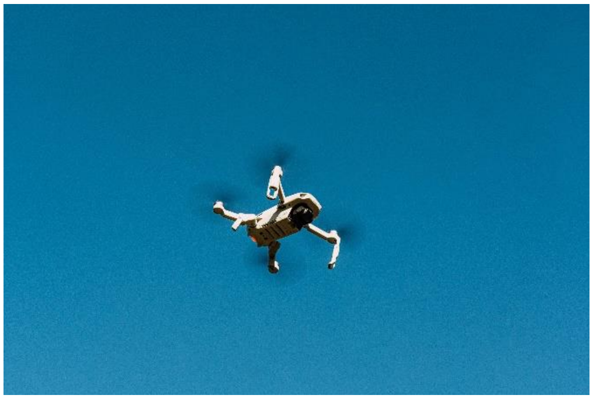


Figure 3.4: Multi-Rotor sUAS with (Day Light Conditions / No Clouds / Level 0 Background Features) (Geometric Photography 2022)

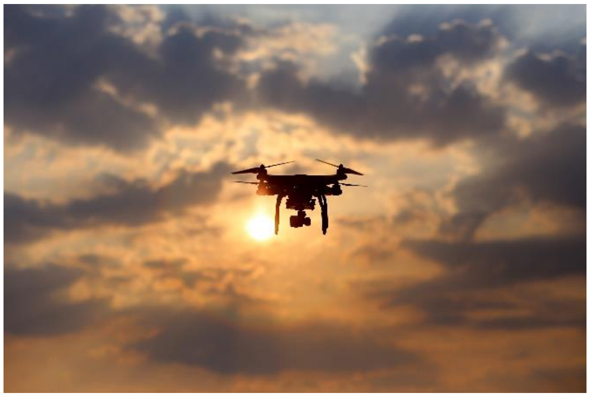


Figure 3.5: Multi-Rotor sUAS with (Low-Light Conditions / Clouds / Level 1 Background Features) (EyeEm n.d.)

3.2.1-4.3 Domain Randomisation

For the generation of the synthetic training and validation data, domain randomisation strategies were used with the variation of the following domain parameters, which is similarly observed in real-world testing data, in order to minimise the domain gap.

- **Multi-Rotor sUAS Texture Properties:** Variation of multi-rotor sUAS model texture properties.

- **Multi-Rotor sUAS Orientation & Position**: Adjustment of multi-rotor sUAS model orientation within UE5 virtual environment in pitch, roll, yaw and position, to simulate airborne flight.

- **Virtual Environment Time of Day:** Variation of simulated time in UE5 virtual environment.

- **Virtual Environment Cloud Formations:** UE5 Volumetric Clouds were varied in terms the amount of cloud coverage and type of cloud cover.

***Note:**

- Other domain randomisation strategies were not included in synthetic data generation, to limit the impact of other strategies on CV model performance, which may interfere with observations of the impact of UE5 photorealism.

3.2.1-5 Offline Rendering

It was decided to use the Cine Camera Actor feature to generate offline rendered synthetic training images of the virtual environment, because it can offer the best quality results for renders for complex scenes requiring high-fidelity photorealism and at a high resolution. The latter point is important since peer reviewed work by Hao et al. (2023) demonstrates that the resolution of imagery enables a greater detection range of objects in CV, which serves to benefit the detection of multi-rotor sUAS that are in-flight at various ranges and altitudes.

3.2.2 Stage 1: Implemented Experimental Method

This section will provide an in-depth explanation of the method undertaken for the Stage 1 Experiments. Figure 3.6 provides a brief overview of the method employed.

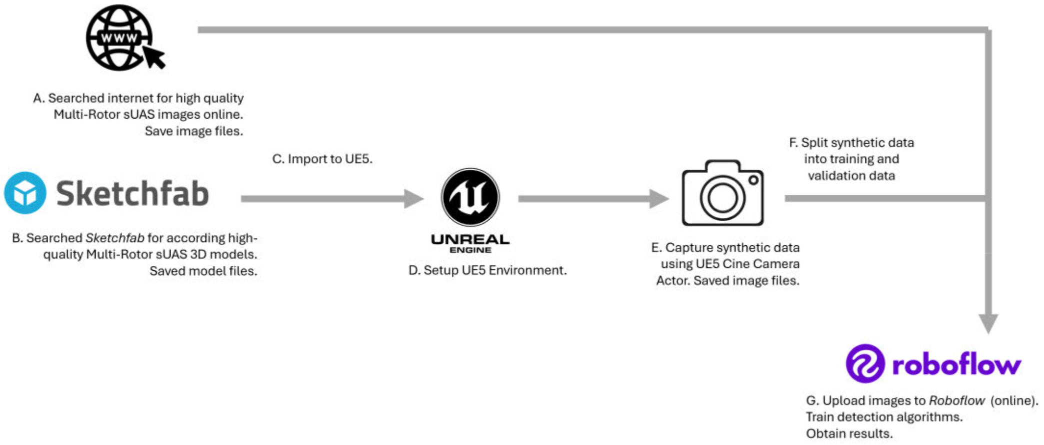
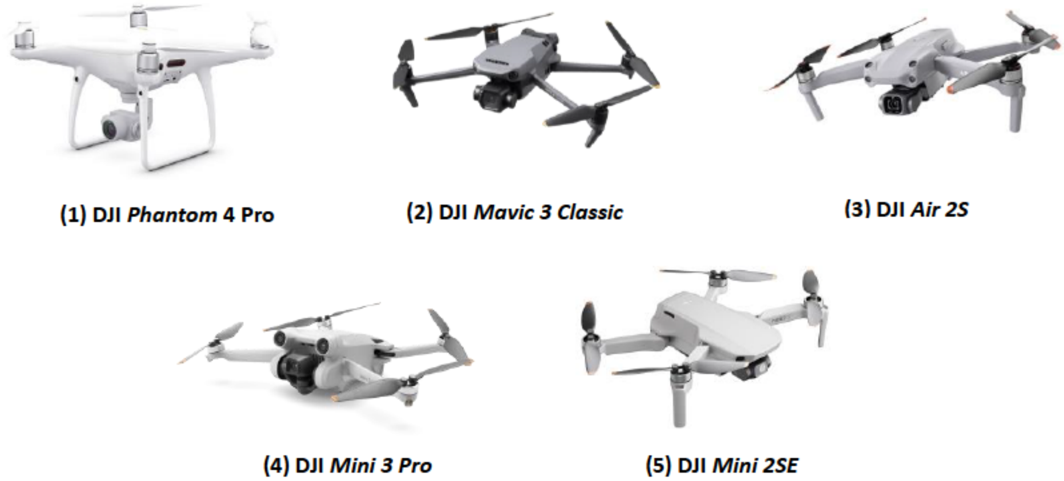

A. Searched internet for high quality Multi-Rotor sUAS images online. Save image files.
Sketchfab
B. Searched *Sketchfab* for according high-quality Multi-Rotor sUAS 3D models. Saved model files.
C. Import to UE5.
UNREAL ENGINE
D. Setup UE5 Environment.
E. Capture synthetic data using UE5 Cine Camera Actor. Saved image files.
F. Split synthetic data into training and validation data
roboflow
G. Upload images to *Roboflow* (online). Train detection algorithms. Obtain results.

Figure 3.6: High-Level Overview of Research Method

Step 1: Testing Dataset Acquisition

Testing dataset images of multi-rotor sUAS with domain conditions of (Day Light Conditions / No Clouds / Level 0 Background Features) and (Low-Light Conditions / Clouds / Level 1 Background Features) were acquired from online sources, due to limited availability of systems on-hand and research investigation time restrictions. The best possible effort was made to obtain images incorporated under various Creative Commons (CC) licences or similar, which permits the public use of these images free of charge, subject to certain terms of use, with online websites including *Wikimedia Commons*, *Pixabay*, *Pexels*, *Unsplash* and *Freepik*. As part of this dataset collection, it was ensured images captured multi-rotor sUAS at different imaging angles and distances with reasonable visibility – approximately 1m - 100m.

However, finding the range of images that catered to the specific domain conditions of the datasets proved difficult with only free images, and consequently there was a need to resort to non-free image content available through subscription-based services, to supplement the training datasets. Resultantly, a *Freepik* premium subscription was used, as well as an individual subscription with Institute of Electrical and Electronics Engineers (IEEE) *DataPort* repository, to access the *Visiodect Dataset: An Aerial Dataset for Scenario-Based Multi-Drone Detection And Identification*. With the use of these online sources, the most prevalent images of multi-rotor sUAS consisted of consumer grade multi-rotor sUAS, including DJI *Mavic* series, DJI *Phantom* series, DJI *Mini* series and DJI *Air* series – refer to Figure 3.7. Therefore, for ease of data collection, images of these types of multi-rotor sUAS were obtained for the testing datasets.



(1) DJI *Phantom* 4 Pro

(2) DJI *Mavic 3 Classic*

(3) DJI *Air 2S*

(4) DJI *Mini 3 Pro*

(5) DJI *Mini 2SE*

Figure 3.7: (1) DJI *Phantom* 4 Pro (DJI 2024f) | (2) DJI *Mavic* 3 Classic (DJI 2024b) | (3) DJI *Air* 2S (PC Market 2024) | (4) DJI Mini 3 Pro (DJI 2024c) | (5) DJI *Mini* 2SE (DJI 2024d)

A decision was made that for acquired images for the test datasets, the multi-rotor sUAS systems within the images were to primarily consist of colours ranging from white, shades of grey and black, and that these systems would have no apparent customisations. This was to ensure that the testing dataset domain was homogenous and did not consist of outlier domain conditions that were not possible or difficult to replicate in synthetic training and validation data, which would undermine the generalisation performance models.

These images were downloaded at the highest resolution, when possible, often at 4K (3840 × 2160 pixels) with some ranging below or even above, with the file format being JPEG. As a point of quality control, some images consisted of undesirable features such as landscapes, people or objects, which may have acted as potential distractors, therefore these images were modified with cropping actions completed using image editing software *Paint* by *Microsoft*.

Further to note is that only 40 images were obtained for each training dataset, considering the significant manual effort required to collect and curate these datasets, with time limitations imposed on this research project. Therefore, it was decided that for the development of each CV model type, a standard data split of 70% training dataset, 10% validation dataset, 20% testing dataset was to be used. From back calculations it was determined that 105 images and 15 images were required for the training and validation datasets respectively.

***Note:** Refer to Appendix B for a list of images utilised for these datasets.

Step 1: Testing Dataset Acquisition Limitations

The following limitations can be stated for the Step 1 testing dataset acquisition process:

- The limitation of the number of testing samples that were able to be obtained, carried over to the number of training and validation samples that will be used to develop each CV model type. This is a low training and validation dataset volume, which ideally be in the tens of thousands for optimal CV model performance.

- Available images online often had image modifications, such as sharpness, exposure, contrast, etc., for aesthetic purposes. The best possible effort was taken to assess images for modifications and not incorporate these images within the dataset, as these can lead to a domain shift between the synthetic data and testing data. However, as this relied on human judgment, it cannot be guaranteed that tests datasets are without these modified images.

- Images collected from online sources were saved as a JPEG, which is an image format that produces image quality losses during the compression process when it is saved. Resultant degradation in image resolution may contribute to minor performance losses in CV performance.

Step 2: Training and Validation Dataset Acquisition

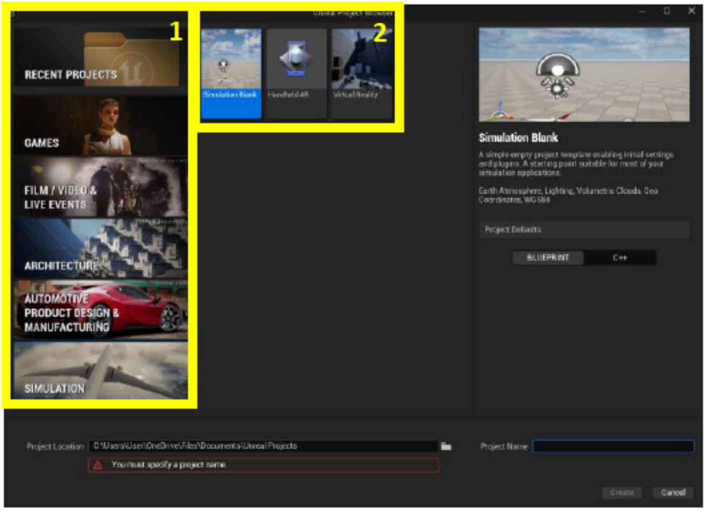
This section will outline the steps taken for generation of training and validation dataset images of multi-rotor sUAS with domain conditions of (Day Light Conditions / No Clouds / Level 0 Background Features) and (Low-Light Conditions / Clouds / Level 1 Background Features) in UE5.

1. Unreal Engine 5 Simulation Environment

For the setup of an UE5 project for this research investigation a *Simulation* environment offered in UE5 was utilised.

Section 1: *Simulation* development category was selected as it provides project templates with preset assets and content packs suitable for simulation-based projects – refer to Figure 3.8 (yellow section) (Epic Games 2024as).

Section 2: Selected *Simulation Blank* which provides a level with simulation-specific settings and functionality suitable for this research project – refer to Figure 3.8 (yellow section) (Epic Games 2024i).



Simulation Blank

A simple empty project template enabling initial settings and plugins. A starting point suitable for most of your simulation applications.

Earth Atmosphere, Lighting, Volumetric Clouds, Geo Coordinates, WGS84

Project Location C:\Users\User\OneDrive\Files\Documents\Unreal Projects

⚠ You must specify a project name.

Project Name

Figure 3.8: Simulation Blank Setup for UE5 Project

2. Setup Multi-Rotor s-UAS 3D Models

For the generation of synthetic datasets, similar multi-rotor sUAS systems to those present within the testing dataset were sourced from an online 3D model repository, *Sketchfab*. Five models were selected, including a DJI *Phantom* 4 Pro, DJI *Mavic 3 Classic*, DJI *Air 2S,* DJI *Mini 3 Pro*, DJI *Mini 2SE*.

Data formats are important for the transfer of 3D content between software, whilst retaining critical model data. GLB and FBX are among many file formats specifically used in the field of DCC, however FBX represents the standard for the VFX and gaming industry, as it can hold more information about model attributes, as well as provide wider software support (Martínez-Díaz 2022; Alpha3D 2023; Adobe 2024d; Schechter 2024). Resultantly, provided GLB files (.glb) for the models were initially downloaded and then converted to a FBX file (.fbx), using online website, *Convert 3D*, which allowed for use within the UE5 virtual environment.

However, the following issues were found when attempting to import the directly converted FBX file of the multi-rotor s-UAS into UE5.

Issue 1:

In UE5 a 3D model can be positioned, oriented and scaled within its local coordinates, which is relative to the origin of global coordinates of an UE5 level. The original 3D models downloaded from *Sketchfab* were typically positioned by the file author in an arbitrary location relative to the origin of the global coordinates of the software used for model creation. When these models were imported into UE5, the arbitrary location of the 3D model is maintained within the UE5 global coordinate system, however, alterations of a model position, orientation and scale within its local coordinates, are then relative to the arbitrary starting location. This in effect produced an offset, which created difficulty when trying to accurately control the orientation and position of the multi-rotor s-UAS model relative to the UE5 global coordinate system, for operations such as domain randomisation (discussed further). To ameliorate this problem, the FBX file was opened and edited using Autodesk *Fusion360* (under Education Licence) to align the approximate centroid of the multi-rotor s-UAS model to the origin of the coordinate system in the Autodesk *Fusion360* development environment.

Issue 2:

Complex 3D models are typically composed of multiple components, and thereby are defined using several meshes for independent control of component model properties (Adobe 2024b; Tiigimägi 2024). The original multi-rotor sUAS models downloaded from *Sketchfab* constituted of tens of components / elements / meshes, that were assigned with respective material and texture properties. This provided an unnecessary amount of control over the individual component model attributes, which needed to be changed individually for each component for the randomisation of the appearance of a multi-rotor sUAS models as part of the used domain randomisation strategy.

To remove this requirement, the edited FBX file, following from Issue 1, was saved and exported as an OBJ file (.obj), as it preserves geometric data, and separates material and texture data with a reference file in MTL (.mtl) format (Library of Congress 2020). After this the OBJ file was opened in Autodesk *Fusion360* and exported again as an FBX file, which combines the separate components into a single mesh (element), which was then imported into UE5. This allowed for easier assignation of different material and texture properties for multi-rotor sUAS models in UE5 as part of domain randomisation when collecting training image datasets.

3. Setup Ultra Dynamic Sky Plugin

Ultra Dynamic Sky is a downloadable plugin from the Unreal Engine Marketplace (*Fab*), which can be setup within an UE5 rendering environment. It is a system which expands the standard capabilities of UE5 environmental lighting, by offering a large set of tools and features, capable of generating high fidelity sky and weather renders in an integrated manner. This allows for the generation of more nuanced lighting conditions and interactions with respect to natural lighting features, which is otherwise a much more involved process using standard UE5 tools, with the consideration of constituent factors separately (Gunther 2024).

The development of accurate cloud formations is one the natural lighting features that significantly benefits from this plugin. This is the major appeal of this plugin for this research, along with its user-friendliness, which can enable the rapid development of the multi-rotor sUAS synthetic data under the required domain conditions (Day Light Conditions / No Clouds / Level 0 Background Features) and (Low-Light Conditions / Clouds / Level 1 Background Features), especially considering the use of domain randomisation strategies for synthetic data generation. Resultantly, *Ultra Dynamic Sky* was purchased and installed in UE5.

For reference, examples of generated results of synthetic images with *Ultra Dynamic Sky* of multi-rotor sUAS with domain conditions of (Low-Light Conditions / Clouds / Level 1 Background Features) and (Day Light Conditions / No Clouds / Level 0 Background Features) are presented below in Figure 3.9 and Figure 3.10 respectively.

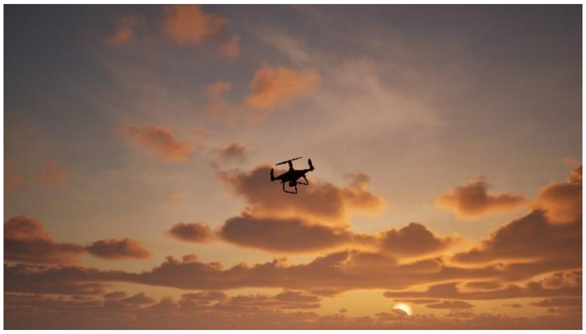


Figure 3.10: Synthetically Generated Image of Multi-Rotor sUAS with (Low-Light Conditions / Clouds / Level 1 Background Features) using *Ultra Dynamic Sky*

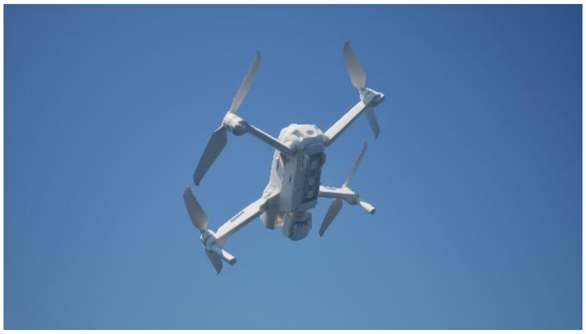


Figure 3.9: Synthetically Generated Image of Multi-Rotor sUAS with (Day Light Conditions / No Clouds / Level 0 Background Features) using *Ultra Dynamic Sky*

4. Implementation of Domain Randomisation Strategies in Unreal Engine

The implementation of domain randomisation strategies outlined in Section 3.2.1-4.3, within the UE5 virtual environment for the generation of synthetic imagery is detailed further below:

4.1 Implementation of Randomisation of Multi-Rotor sUAS Texture Properties

As part of domain randomisation of multi-rotor sUAS texture properties, five copies (arbitrarily chosen value) of each of the multi-rotor sUAS 3D models were imported into the virtual environment (level), as seen in Figure 3.11.

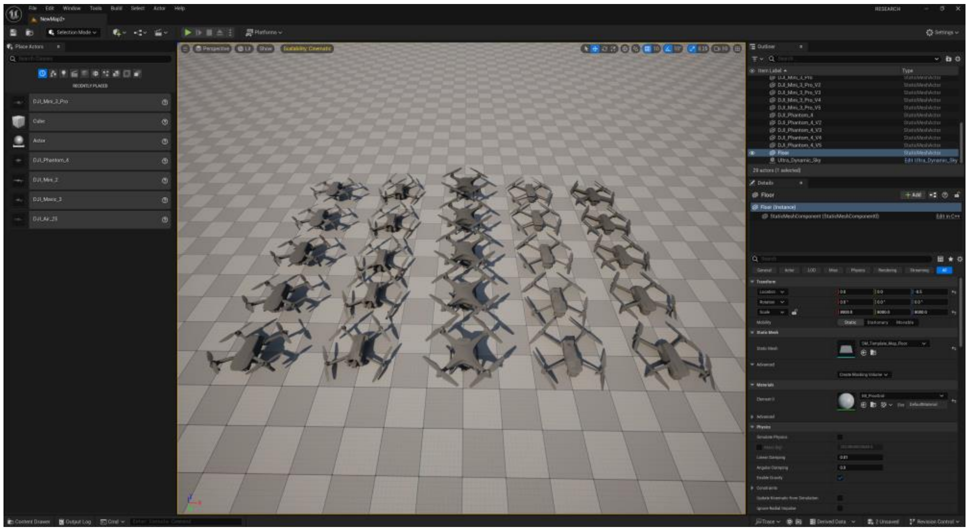


Figure 3.11: Import Multiple Multi-Rotor sUAS Models into UE5 Level

The material properties of each model were then selected and assigned as plastic, as shown in Figure 3.12 (yellow section), to generally simulate the real-world material properties of consumer grade drones which use a combination of polycarbonate carbon fibre composite materials (DJI 2016, 2024a).

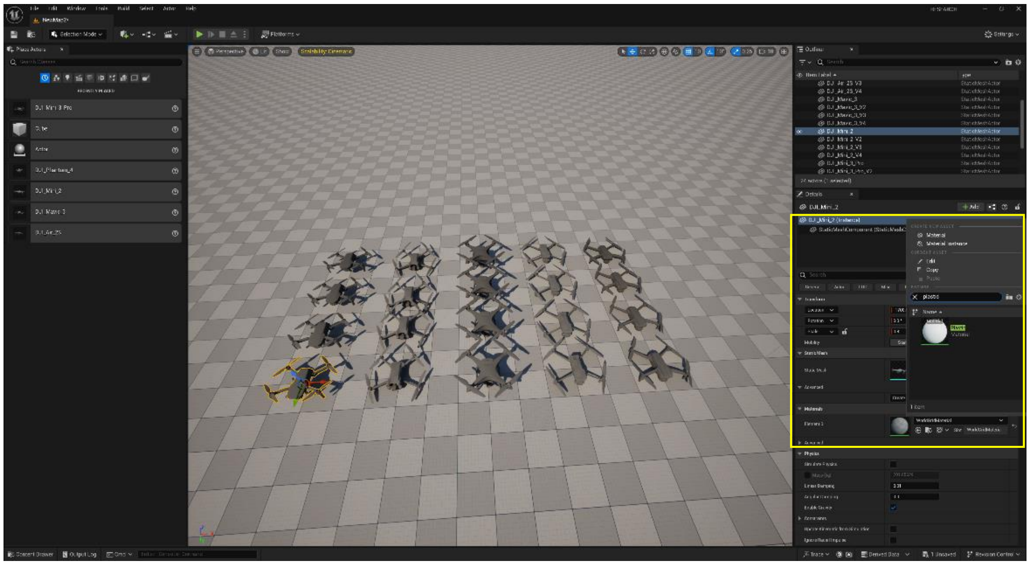
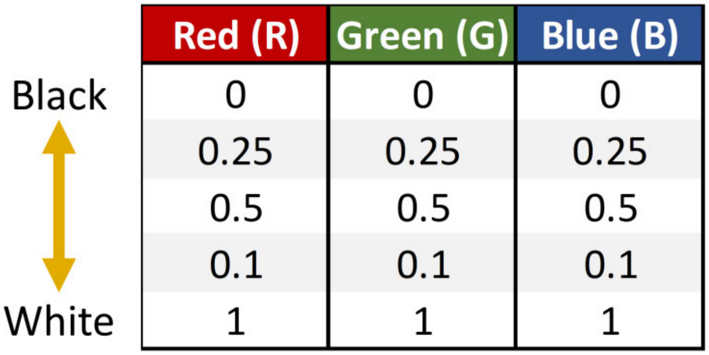


Figure 3.12: Assign Material Properties to Multi-Rotor sUAS Models

Since real-world appearances of drones in the test data varied from white, shades of grey and black, each of the five copies of the multi-rotor sUAS models were similarly altered in terms of texture colour by editing the texture properties of the assigned plastic material, with five RGB values.

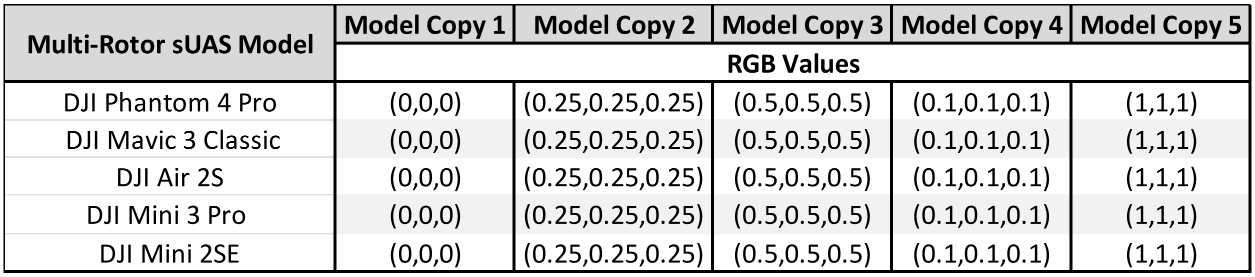
Table 3.1 presents the assigned RGB values used for each of the five copies of each multi-rotor sUAS model.



	Red (R)	Green (G)	Blue (B)
Black	0	0	0
↕	0.25	0.25	0.25
	0.5	0.5	0.5
	0.1	0.1	0.1
White	1	1	1

Table 3.1: RGB Values of Texture Colour for Multi-Rotor sUAS Model Copies

For clarification Table 3.2 presents the assigned RGB values for each of the five copies of each multi-rotor sUAS model.



Multi-Rotor sUAS Model	Model Copy 1	Model Copy 2	Model Copy 3	Model Copy 4	Model Copy 5
	RGB Values				
DJI Phantom 4 Pro	(0,0,0)	(0.25,0.25,0.25)	(0.5,0.5,0.5)	(0.1,0.1,0.1)	(1,1,1)
DJI Mavic 3 Classic	(0,0,0)	(0.25,0.25,0.25)	(0.5,0.5,0.5)	(0.1,0.1,0.1)	(1,1,1)
DJI Air 2S	(0,0,0)	(0.25,0.25,0.25)	(0.5,0.5,0.5)	(0.1,0.1,0.1)	(1,1,1)
DJI Mini 3 Pro	(0,0,0)	(0.25,0.25,0.25)	(0.5,0.5,0.5)	(0.1,0.1,0.1)	(1,1,1)
DJI Mini 2SE	(0,0,0)	(0.25,0.25,0.25)	(0.5,0.5,0.5)	(0.1,0.1,0.1)	(1,1,1)

Table 3.2: Assigned RGB values of Texture Colour for Copy of Multi-Rotor sUAS Models

The visual results of modifying the RGB values of each of the five copies of each multi-rotor sUAS model is presented visually presented in Figure 3.13.

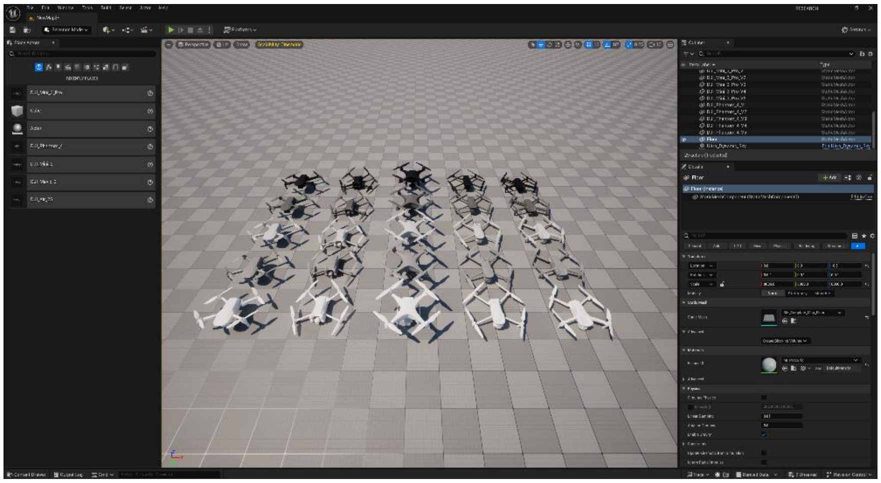


Figure 3.13: Results of Assigned RGB values of Texture Colour for Copy of Multi-Rotor sUAS Models

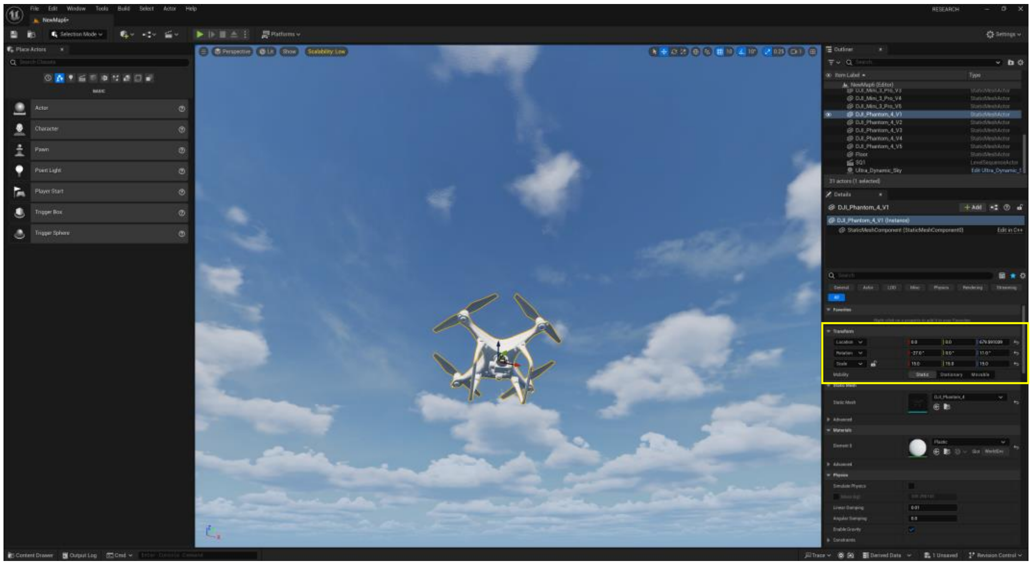
For the generation of a synthetic image, one of the five multi-rotor sUAS models were randomly selected, as well as an according model copy. This process was completed by use of an online random number generator, which generated two numbers 1 - 5 (whole numbers), to index the multi-rotor sUAS model and the model copy, from Table 3.3, to be used in the randomisation process.

Randomisation Index Values for Selection of Multi-Rotor sUAS Model and Model Copy					
Multi-Rotor sUAS Model	Model Copy 1	Model Copy 2	Model Copy 3	Model Copy 4	Model Copy 5
DJI Phantom 4 Pro	(1,1)	(2,1)	(3,1)	(4,1)	(5,1)
DJI Mavic 3 Classic	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
DJI Air 2S	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
DJI Mini 3 Pro	(1,4)	(2,4)	(3,4)	(4,4)	(5,4)
DJI Mini 2SE	(1,5)	(2,5)	(3,5)	(4,5)	(5,5)

Table 3.3: Randomised Index Values for Selection of Multi-Rotor sUAS Model and Model Copy

4.2 Implementation of Randomisation of Multi-Rotor sUAS Orientation & Position

Using the selected 3D model copy, parameters constituting its orientation and position within its local coordinates were randomised. This was achieved with the modification of *Location* and *Rotation* settings (yellow section), as shown in Figure 3.14. The *Location* settings effectively controlled the height (z) and distance (x,y) of the multi-rotor sUAS within local coordinates, relative to the UE5 global coordinate origin. The *Rotation* settings controlled the rotation of the model relative to its local (x,y,z) axe, which effectively controlled the perceived pitch, roll and yaw of a multi-rotor sUAS model within the UE5 virtual environment.



DJI_Phantom_4_V1 (Instance)

Transform

Location 0.0 0.0 679.991009

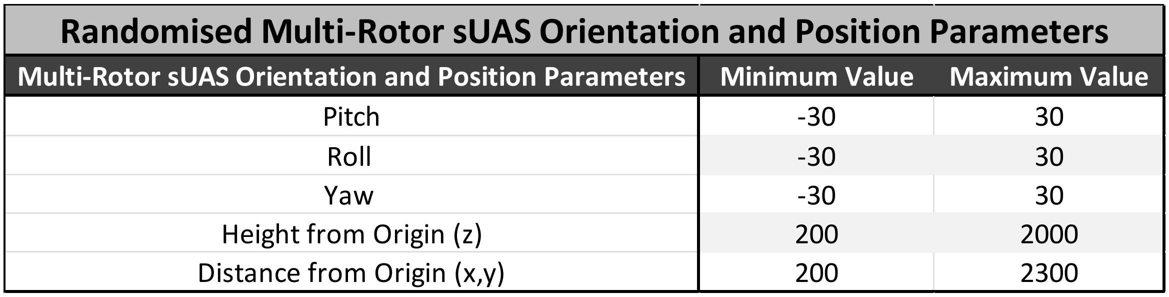
Rotation -27.0° 0.0° 11.0°

Scale 15.0 15.0 15.0

Mobility Static Stationary Movable

Figure 3.14: Unreal Engine 5 *Location* and *Rotation* Settings

The randomisation of the orientation and position parameters of a selected multi-rotor sUAS for a generated synthetic image was completed with the assistance of an online random number generator, which generated values (whole numbers) between the established minimum and maximum ranges, as presented in Table 3.4. The determined values were then applied to the multi-rotor sUAS model in the virtual environment. Note that the ranges for the pitch, roll and yaw, as well as the distance and height from the origin, were established through trial and error as the best ranges that produced similar results in appearance to the real-world datasets, with the use of the Cine Camera Actor for imaging.



Randomised Multi-Rotor sUAS Orientation and Position Parameters		
Multi-Rotor sUAS Orientation and Position Parameters	**Minimum Value**	**Maximum Value**
Pitch	-30	30
Roll	-30	30
Yaw	-30	30
Height from Origin (z)	200	2000
Distance from Origin (x,y)	200	2300

Table 3.4: Randomised Multi-Rotor sUAS Orientation and Position Parameters

***Note:** The minimum height from origin (z) of 200, was specifically determined as value at which the Cine Camera Actor (with preset *Filmback* (to be discussed further)), when also set at the same height, does not view the floor of the virtual environment with the imaged multi-rotor sUAS, which may act as a potential distractor in synthetic training imagery.

4.3 Implementation of Randomisation of Virtual Environment Time of Day and Virtual Environment Cloud Formations

For the randomisation of the virtual environment time of day and virtual environment cloud formation domain parameters, corresponding settings provided by the *Ultra Dynamic Sky* plugin were varied within the framework of the synthetic image domain conditions to be generated, i.e. (Low-Light Conditions / Clouds / Level 1 Background Features) or (Day Light Conditions / No Clouds / Level 0 Background Features). The settings included the following:

4.3.1 Virtual Environment Time of Day

Using the *Time of Day* setting of the *Ultra Dynamic Sky* plugin (yellow section), as shown in Figure 3.15, it was possible to vary the time of day along with the according environmental lighting conditions, between 0000 to 2400 hours. Figure 3.15 demonstrates this controllability, with an example of simulated time at 1800 hours.

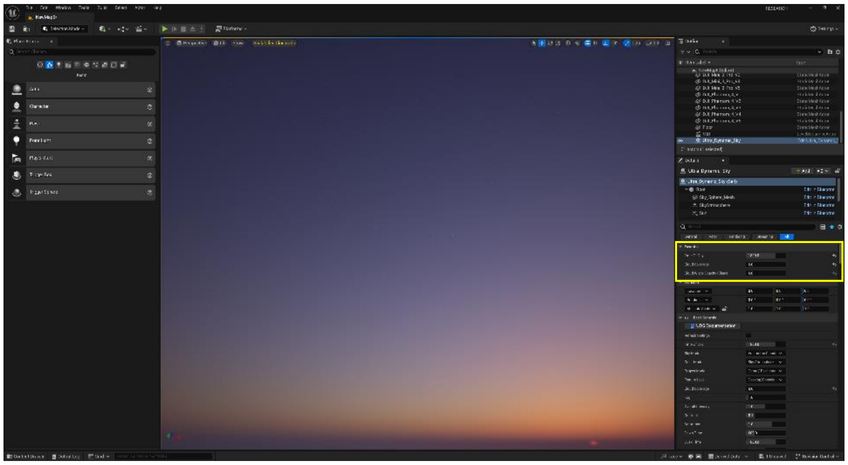


Figure 3.15: ***Ultra Dynamic Sky*** **Time of Day Setting Example**

***Note:** *Ultra Dynamic Sky* can simulate seasons and the respective time-of-day lighting conditions, however for simplicity, the default season setting was used for experimentation.

4.3.2 Virtual Environment Cloud Formations

For the simulation of virtual environment cloud formations, two factors were used which varied the amount of cloud coverage and type of cloud cover, to create levels of nuance to the synthetic image sky background, alike real-world conditions:

- **Cloud Coverage:** Using the *Cloud Coverage* setting (yellow section) as shown in Figure 3.16, it was possible to generate various levels of cloud coverage, ranging from clear sky to overcast. These can be considered low-level to middle-level clouds which appear 'puffy' (Ceranic 2020). Figure 3.16 demonstrates this controllability with an example of simulated cloud coverage at a setting value of 4.0.

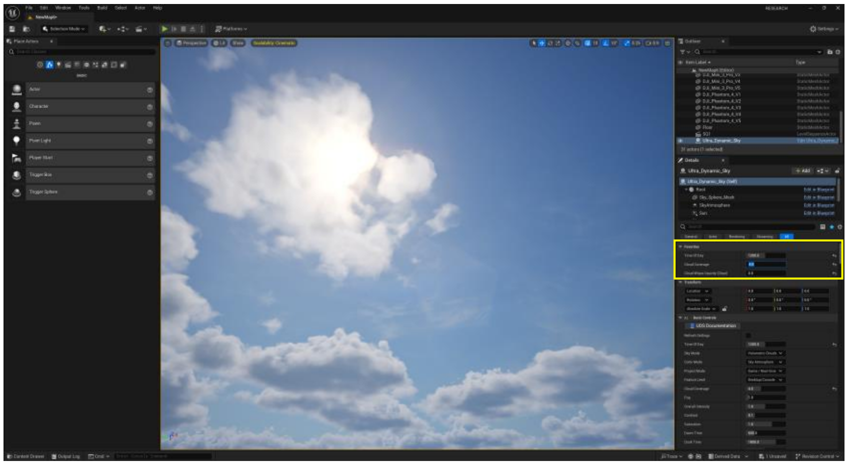


Figure 3.16: ***Ultra Dynamic Sky*** **Time of Day Setting Example**

- **Cloud Wisps:** Using the *Cloud Wisps Opacity (Clear)* setting (yellow section), as shown in Figure 3.17, it was possible to generate high-level clouds, which appear as wisps – strands or tufts of thin clouds (Ceranic 2020). The setting adjusts the level of opacity, however in effect can be considered a form of cloud coverage. Figure 3.17 demonstrates this controllability with an example of simulated cloud coverage at a setting value of 3.0.

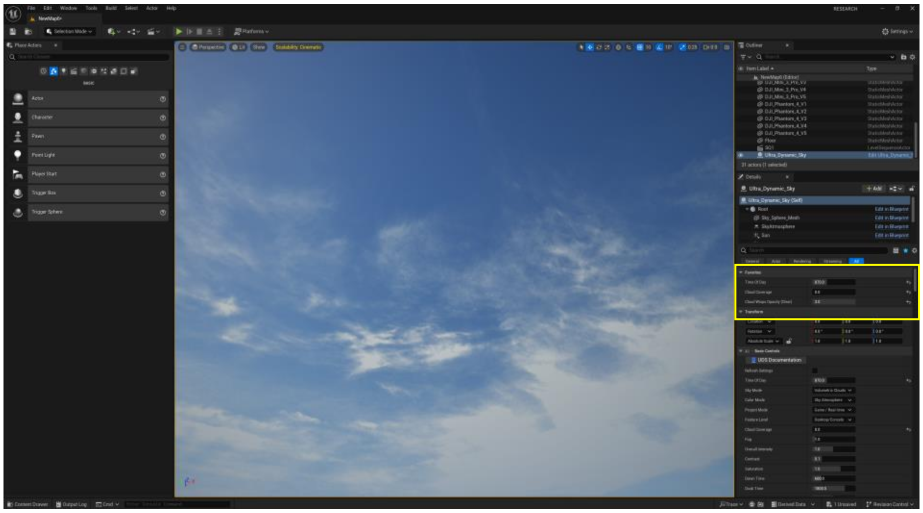


Figure 3.17: *Ultra Dynamic Sky* Cloud Wisps Setting Example

Taking these *Ultra Dynamic Sky* plugin settings into account, for the generation of synthetic data consisting of multi-rotor sUAS with (Low-Light Conditions / Clouds / Level 1 Background Features) and (Day Light Conditions / No Clouds / Level 0 Background Features), the following virtual environment time of day and virtual environment cloud formation domain parameters were randomised, as outlined in Table 3.4 and Table 3.5 respectively. The minimum and maximum ranges used in the randomisation process were also established through trial and error as the best ranges that produced similar results in appearance to the real-world datasets. Furthermore, the randomisation process of the was also completed by use of an online random number generator, which generated values between the established minimum and maximum ranges, to be used in the randomisation process. The determined values of the parameters were then implemented in the UE5 virtual environment with the according *Ultra Dynamic Sky* settings.

Randomised *Ultra Dynamic Sky* Parameters for Domain Condition: Low-Light Conditions / No Clouds / Level 1 Background Features⍰		
Domain Parameters (*Ultra Dynamic Sky Settings*)	**Minimum Value**	**Maximum Value**
Time of Day – Low Light Range 1	600	700
Time of Day – Low Light Range 2	1780	1800
Cloud Wisps	0	5
Cloud Coverage	0	6.5

Table 3.5: Randomised Ultra Dynamic Sky Parameters for Domain Condition: Low-Light Conditions / Clouds / Level 1 Background Features

Details regarding the ranges of *Ultra Dynamic Sky* settings for the generation of images of multi-rotor sUAS with (Low-Light Conditions / Clouds / Level 1 Background Features) are outlined below:

- **Time of Day – Low Light Range 1:** This range simulated low light in early hours of day.

- **Time of Day – Low Light Range 2:** This range simulated low light in later hours of day.

  - During the randomisation process, the use of ranges of Time of Day – Low Light Range 1 or Time of Day – Low Light Range 2 were alternated. Furthermore, determined values by the online random number generator were whole numbers.

  - **Cloud Wisps and Cloud Coverage:** Determined values by the online random number generator were decimal numbers, accurate to one decimal place.

Randomised *Ultra Dynamic Sky* Parameters for Domain Condition: Day Light / No Clouds / Level 0 Background Features		
Domain Parameters (*Ultra Dynamic Sky Settings*)	**Minimum Value**	**Maximum Value**
Time of Day	800	1500
Cloud Wisps	Set to a Value of 0	Set to a Value of 0
Cloud Coverage	Set to a Value of 0	Set to a Value of 0

Table 3.6: Randomised Ultra Dynamic Sky Parameters for Domain Condition: Day Light Conditions / No Clouds / Level 0 Background Features

Details regarding the ranges of domain parameters for the generation of images of multi-rotor sUAS with (Day Light Conditions / No Clouds / Level 0 Background Features) are outlined below:

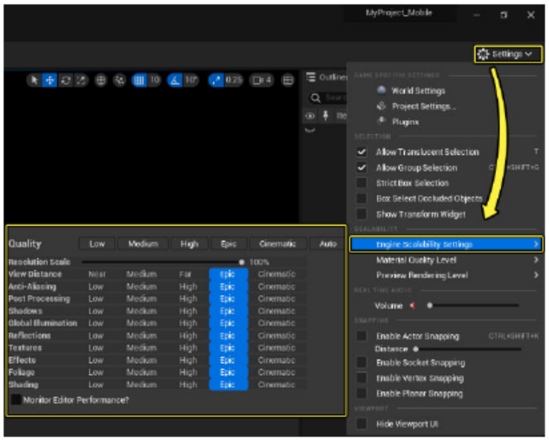
- **Time of Day:** This range simulated full day light hours with no dimming or low light.

- **Cloud Wisps and Cloud Coverage:** Values were set to zero considering the use of Level 0 Background Features.

5. Implementation of Scalability Settings for Experiment 1 and Experiment 2

5.1 Scalability Settings Inspection

Before the acquisition of synthetically generated image data, an initial inspection of imagery using *Low* scalability settings versus *Cinematic* scalability settings was initially completed to gain an understand of their comparative impact on rendering quality. For a quick and effective comparison, separate screenshots were taken of real-time renders made within the level environment of UE5, of a multi-rotor sUAS under (Day Light Conditions / No Clouds / Level 0 Background Features), using *Low* scalability settings and *Cinematic* scalability settings for all properties. These adjustments were made by selecting: ***UE5 Settings > Engine Scalability Settings*** as seen in Figure 3.18.



Quality	Low	Medium	High	Epic	Cinematic	Auto
Resolution Scale					100%	
View Distance	Near	Medium	Far	Epic	Cinematic	
Anti-Aliasing	Low	Medium	High	Epic	Cinematic	
Post Processing	Low	Medium	High	Epic	Cinematic	
Shadows	Low	Medium	High	Epic	Cinematic	
Global Illumination	Low	Medium	High	Epic	Cinematic	
Reflections	Low	Medium	High	Epic	Cinematic	
Textures	Low	Medium	High	Epic	Cinematic	
Effects	Low	Medium	High	Epic	Cinematic	
Foliage	Low	Medium	High	Epic	Cinematic	
Shading	Low	Medium	High	Epic	Cinematic	

□ Monitor Editor Performance?

Figure 3.18: UE5 Scalability Settings (Epic Games 2024g)

The results of these screenshots are presented in Figure 3.19, and clearly confirm that these settings in fact have an effect on photorealism and hence supported the approach to investigation – to gauge the performance impact of high-fidelity photorealism on CV-based multi-rotor sUAS detection models, by comparison of model performances in Experiment 1 and Experiment 2, developed with synthetic imagery rendered using *Low* and *Cinematic* scalability settings respectively.

Considering that the focus of investigation is to understand the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows, another visual inspection was completed to understand the impact of each of the eleven rendered properties pertaining to the scalability settings. This was completed with a comparative investigation of eleven screenshots of real-time renders made within the level environment of UE5, of a multi-rotor sUAS under (Day Light Conditions / No Clouds / Level 0 Background Features), using primarily *Cinematic* scalability settings, however, for each of the screenshots one rendered property had its scalability dropped to *Low*. Each image was then compared with the screenshot of real-time renders made within the level environment of UE5, of a multi-rotor sUAS under (Day Light Conditions / No Clouds / Level 0 Background Features), using *Cinematic* scalability settings for all rendered properties (refer to Figure 3.19 Right), which was used as a benchmark image to evaluate the visual significance of each property.

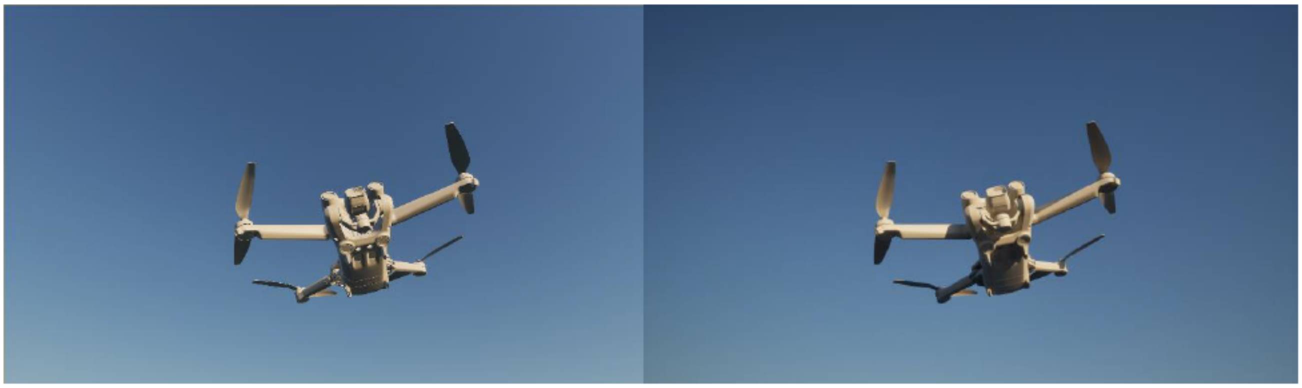
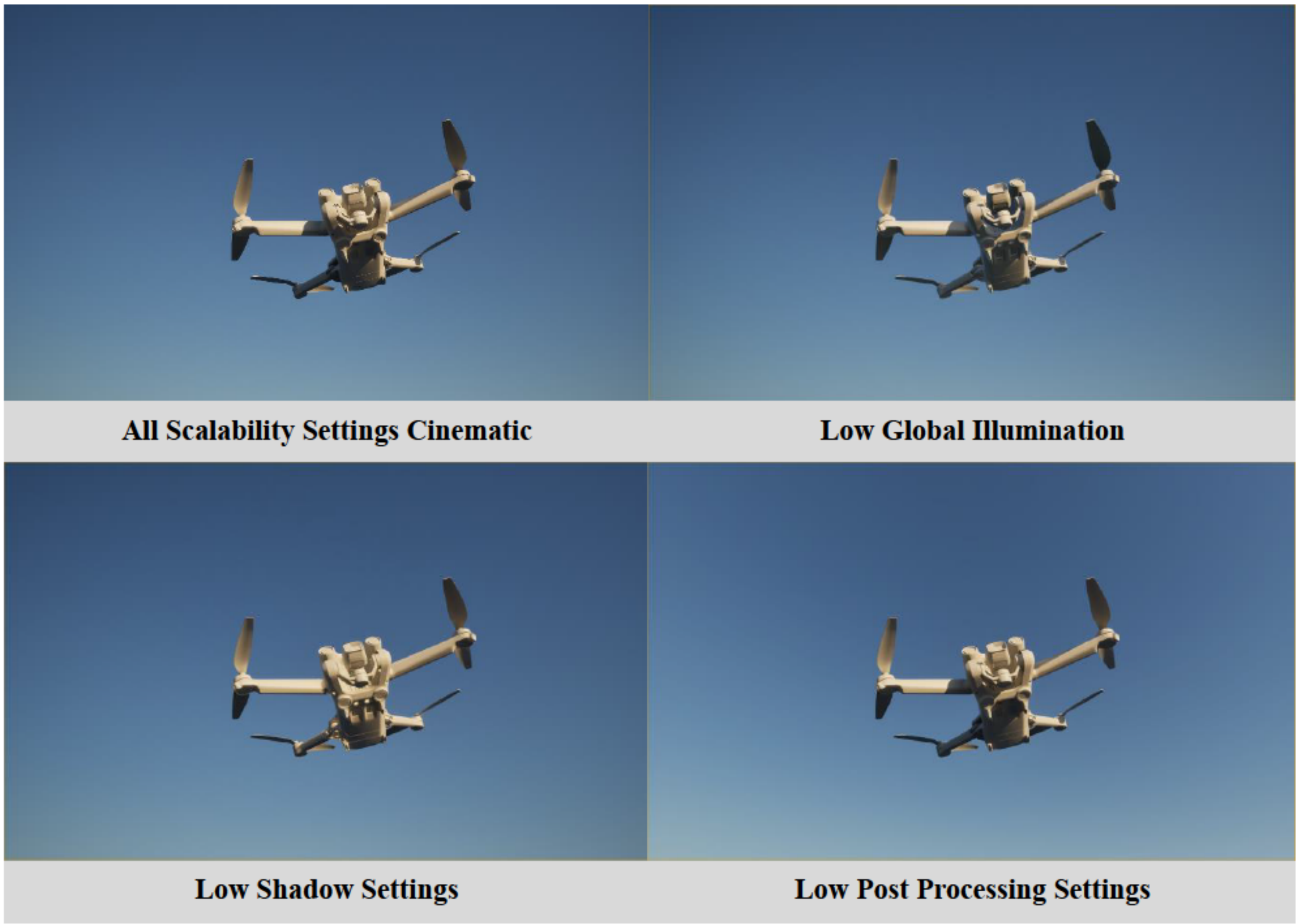


Figure 3.19: Render using *Low* Scalability Settings [Left] vs Render using *Cinematic* Scalability Settings [Right] in Unreal Engine 5 (Real-Time Render)

From this comparison the following rendered properties were noted to be visually significant – Post Processing, Shadows, Global Illumination, Reflections and Textures, as seen in Figure 3.20. This further supported the approach taken for this investigation – to gauge the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data.



All Scalability Settings Cinematic

Low Global Illumination

Low Shadow Settings

Low Post Processing Settings

Figure 3.20: Impact of Visually Significant Render Properties

It should be noted that with the use of *Low* and *Cinematic* scalability settings with synthetic imagery rendered for Experiment 1 and Experiment 2 respectively, technical differences in the methods of global illumination and reflections likely contributed to the differences in photorealism observed. Below an overview is provided of rendering engine capabilities employed for Experiment 1 and Experiment 2:

Technical Differences in Rendering Engine Capabilities Used for *Low* and *Cinematic* Scalability Settings	
Experiment 1 (*Low Scalability Settings*)	**Experiment 2 (*Cinematic* Scalability Settings)**
Screen Space Global Illumination	*Lumen* Global Illumination
Screen Space Reflections	*Lumen* Reflections
Ray Traced Shadows	Ray Traced Shadows

Hardware Ray Tracing	Hardware Ray Tracing

6. Implementation of Synthetic Data Capture – Cine Camera Actor

For the capture of synthetic training and validation images, the Cine Camera Actor was employed by selecting ***Quick Add > Cinematic > Cine Camera Actor***, as seen in Figure 3.21, which then allowed placement within the UE5 virtual environment.

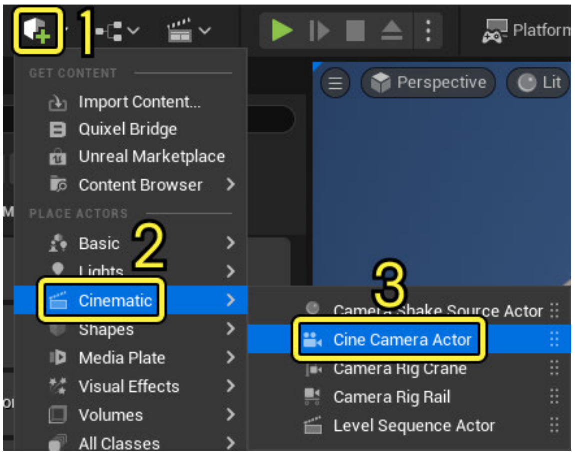
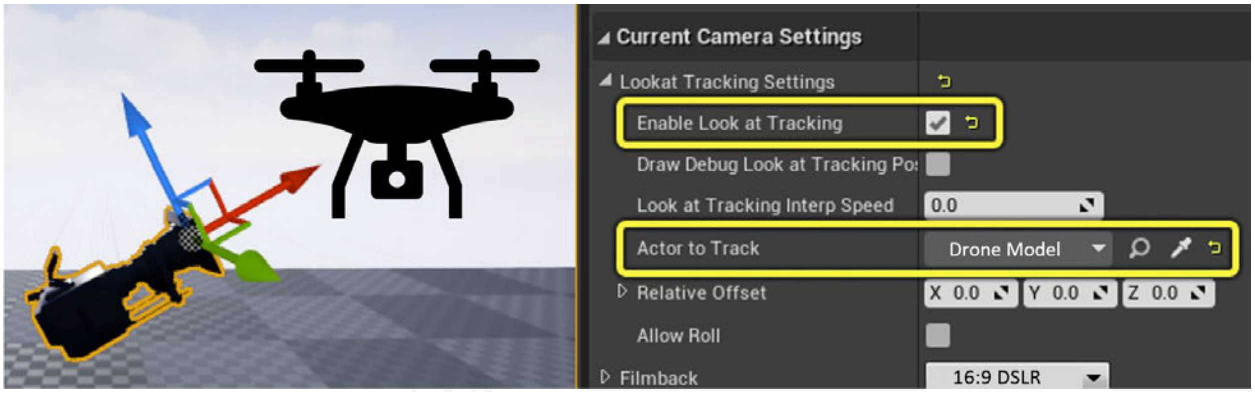


Figure 3.21: Setup of Cine Camera Actor (Epic Games 2024a)

The Cine Camera Actor was initially placed on the ground level of the virtual environment over the global coordinate origin of the Level floor. For the capture of each synthetic image of a selected model copy, under the *Current Camera Settings* the *Enable Look at Tracking* setting was activated with the *Actor to Track* set as the requisite multi-rotor sUAS model copy – refer to Figure 3.22. This allowed for a more effective method of manipulating the camera viewing angles to capture a multi-rotor sUAS, especially during the randomisation of the orientation and the position of the model, relative to the global coordinate origin within the virtual environment, as opposed to manual control.



Current Camera Settings

Lookat Tracking Settings

Enable Look at Tracking	☑
Draw Debug Look at Tracking Po:	□
Look at Tracking Interp Speed	0.0
Actor to Track	Drone Model
Relative Offset	X 0.0 Y 0.0 Z 0.0
Allow Roll	□
Filmback	16:9 DSLR

Figure 3.22: Cine Camera Actor Settings

Additionally, the *Filmback* setting under the *Current Camera Settings* was used, which provides selectable preset camera types to be simulated by the Cine Camera Actor, in terms of a captured image field of view and aspect ratio. Considering that most images sourced from online sources stated the use of a Digital Single-Lens Reflex Camera (DSLR) camera, in order to replicate this in captured synthetic imagery the 16:9 DSLR preset was utilised.

***Note:** The height position of the Cine Camera Actor (z) was ensured so that captured imagery does not view the floor of the virtual environment with the multi-rotor sUAS, which may act as a potential distractor in synthetic training imagery.

To provide visual context, Figure 3.23 illustrates all implemented strategies for synthetic data capture within the UE5 virtual environment.

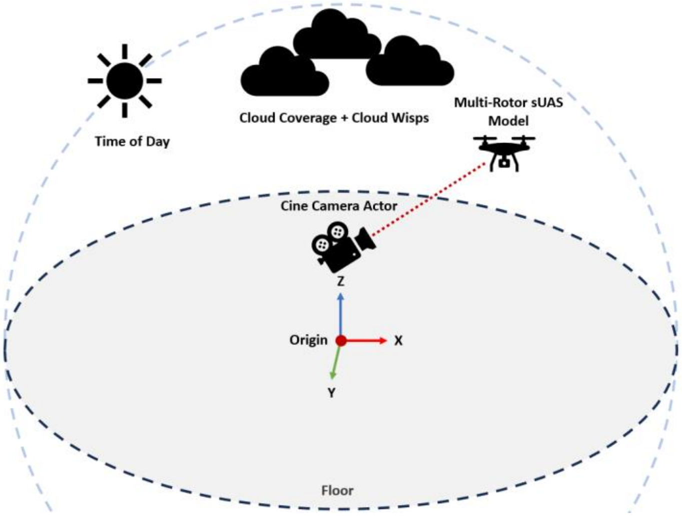


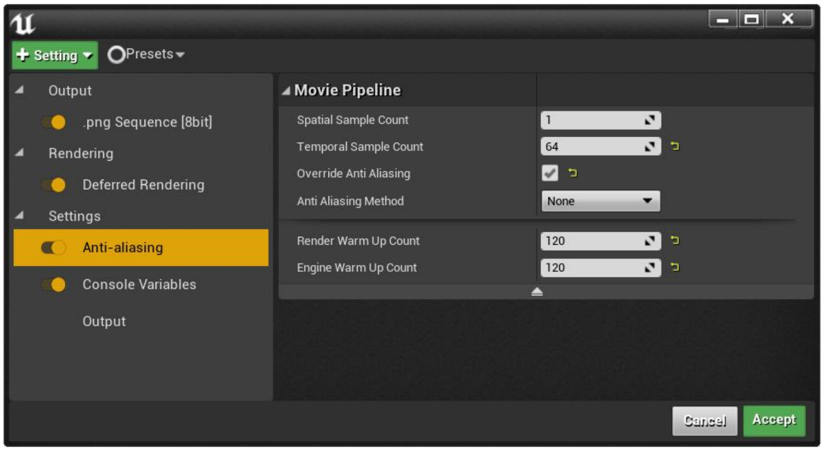
Figure 3.23: Implemented Strategies for Synthetic Data Capture

7. Implementation of Synthetic Data Capture – Movie Render Queue

In order to capture an image using the Cine Camera Actor, the Movie Render Queue plugin was installed and to generate offline rendered images of the virtual environment. Considering that this process is highly involved, this will not be detailed in this methodology, however key points are listed below.

- The *Sequencer Editor* was set to output a single frame to capture an image.

- Recommended anti-aliasing settings were configured as seen in Figure 3.24. This was found to be necessary to remove ‘ghosting’, which is an aliasing artifact present on rendered object edges, showing as a slightly blurred outline. This was particularly noticeable upon closer inspection of initial offline renders output by Movie Render Queue, as observed in Figure 3.25.



Movie Pipeline

Spatial Sample Count	1
Temporal Sample Count	64
Override Anti Aliasing	☑
Anti Aliasing Method	None
Render Warm Up Count	120
Engine Warm Up Count	120

Figure 3.24: Recommended Anti-Aliasing Settings

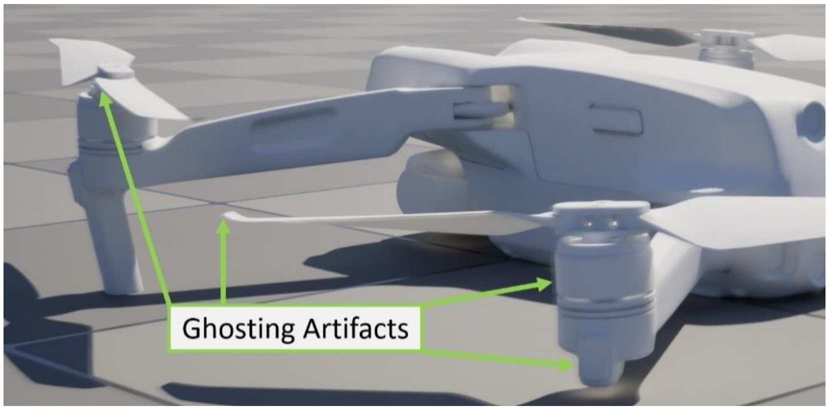


Figure 3.25: Offline Render Ghosting Artifacts

- Images were output at 4K (3840 × 2160 pixels) with a JPEG file format to match average resolution and file compression to mitigate the influence of potential low-level domain inconsistencies which may impact detection.

For more information regarding the installation and generation of image output via Movie Render Queue, refer to the following website:

- https://dev.epicgames.com/documentation/en-us/unreal-engine/rendering-high-quality-frames-with-movie-render-queue-in-unreal-engine

Step 2: Training and Validation Dataset Acquisition Limitations

The following limitations can be stated for the Step 2 training and validation dataset acquisition process:

- Since the *Look at Tracking* process automatically centres the Actor / target object to be tracked within the image captured by the Cine Camera Actor, this may bias CV model learning to search for multi-rotor sUAS in the middle of an image, and thereby impact generalisation when applied to real-world images, where the location of multi-rotor sUAS are varied within images of the target distribution.

- The randomisation of the *Filmback* presented a further option for domain randomisation that could have been used to increase CV model robustness. However, this can be explored in a further investigation.

- Synthetic data acquisition is a time consuming process, alike real-world data acquisition.

Step 3: CV Model Development Process with *Roboflow*

To build and develop multiple CV model types for Experiment 1 and Experiment 2, inspiration was drawn from the work of Damian et al. (2023) to use an online platform that removes the need for model development from scratch. Upon investigation it was found that two platforms offer such a capability, including *Roboflow* and *V7 Labs*, with features such as hosted model training features, dataset management, dataset annotation tools, deployment options, etc. From comparison it was found that *Roboflow* presented a more affordable and user-friendly option for CV projects for the consumer-level and small teams and therefore was the platform of choice, compared to *V7 Labs*, which is suited for enterprise-level CV projects (OpticIntellect 2024).

Step 3 Technical Challenge

Originally it was opted to use *Roboflow* Notebooks for carious CV model development, although not explicitly part of the *Roboflow* online platform itself, these are a collection of open-source Jupyter Notebooks available on GitHub that enable users to learn and develop CV models for differing tasks with step-by-step guides and outlined sections of code. One major advantage of this method is the customisability of CV models with tuneable hyperparameters and attainable performance metrics. However unforeseen difficulty was encountered during usage, which can either be attributed to human error or systematic error, therefore the *Roboflow* online platform was employed.

Step 3 Solution

In comparison, CV model development with the *Roboflow* online platform is a more streamlined process, which provides an API that offers assistive tools, to systematically develop CV models end-to-end for differing tasks. Furthermore, hyperparameters are optimised during training by the platform itself, which can be considered both and advantage and a disadvantage with unavailable control of model performance by the user (Witt 2024).

The following steps were taken for the development of all CV model types developed for Experiment 1 (*Low* scalability settings) and Experiment 2 (*Cinematic* scalability settings), per each domain condition (Low-Light Conditions / Clouds / Level 1 Background Features) and (Day Light Conditions / No Clouds / Level 0 Background Features).

1. Created a *Workspace* for CV project.

2. Selected a *Roboflow* user plan – considering that this platform was to be extensively used for CV model training operations, a subscription plan was selected to gain access to required training credits.

3. Established *Project Name* and annotation (label) class for multi-rotor sUAS.

4. Selected *Project Type* as object detection.

5. Uploaded 105 synthetic training images (70%), 15 synthetic validation images (10%) and 40 real-world testing images (20%) for each domain condition.

6. Images samples were initially labelled with the automatic labelling feature by *Roboflow* to reduce annotation time, before manual annotation was applied to missed or incorrectly labelled samples.

7. After these steps, selectable options of preprocessing (application of image transformations to datasets) and augmentation (creation of new image data by modification of images from existing dataset to increase number of dataset samples) to improve model performance were provided by *Roboflow*, however were avoided to limit the impact of other strategies on CV model performance, other than photorealism.

   ***Note:** By default, all images are preprocessed by being resized to a standard 640 × 640 pixels for usage as an input to hosted CV models by *Roboflow*.

8. Hosted CV models by *Roboflow* refer to machine learning models that are hosted on the cloud that are readily usable through the API. You Only Look Once (YOLO) is a popular architecture used for real-time object detection and tracking, with multiple model types having been iteratively produced by various researchers and organisations. YOLO models are commonly available in different model network sizes, which typically trade speed for accuracy with larger sizes (vice versa). *Roboflow* primarily offers YOLO hosted models – as of writing this report YOLOv11 Small, YOLOv11 Nano, YOLO NAS Medium and YOLO NAS Small.

   Therefore, Steps 3 -7 were repeated to develop these four types of YOLO models using datasets for each domain condition pertaining to Experiment 1 and Experiment 2. For clarity this is outlined below:

	Experiment 1 Low Scalability Settings	Experiment 1 Low Scalability Settings	Experiment 2 Cinematic Scalability Settings	Experiment 2 Cinematic Scalability Settings
	Day Light / No Clouds / Level 0 Background Features	Day Light Conditions / No Clouds / Level 0 Background Features	Day Light / No Clouds / Level 0 Background Features	Day Light Conditions / No Clouds / Level 0 Background Features
CV Model Type	YOLOv11 Small	YOLOv11 Small	YOLOv11 Small	YOLOv11 Small
	YOLOv11 Nano	YOLOv11 Nano	YOLOv11 Nano	YOLOv11 Nano
	YOLO NAS Medium	YOLO NAS Medium	YOLO NAS Medium	YOLO NAS Medium
	YOLO NAS Small	YOLO NAS Small	YOLO NAS Small	YOLO NAS Small

9. *Roboflow* provides the option to use pretrained versions each CV model type using an industry standard dataset – *Microsoft* Common Objects in Context (COCO), or versions each CV model type that have not been pretrained. For the sake of minimising training times and improving performance, pretrained versions of these models were utilised.

10. Training process was then initiated.

After training completion, the performance metrics of each model were provided by *Roboflow*. It should be noted that *Roboflow* online platform provides a limited set of performance metrics, including the following:

- MAP with a 50% IOU threshold (mAP@50)
- Precision
- Recall
- Training graphs:

- ➢ MAP with a 50% IOU threshold (mAP@50) versus epoch range optimised by *Roboflow*.
- ➢ MAP with a 50% - 95% IOU threshold (mAP@50:95) versus epoch range optimised by *Roboflow.*
- ➢ Box loss versus epoch range optimised by *Roboflow.*
- ➢ Class loss versus epoch range optimised by *Roboflow.*
- ➢ Object loss versus epoch range optimised by *Roboflow.*

Step 3: CV Model Development Process with *Roboflow* Limitations

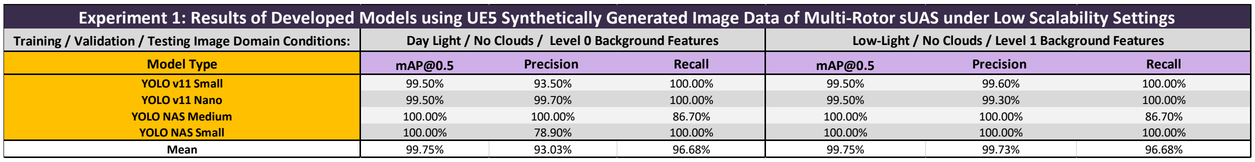
The following limitation can be stated for the Step 3: CV Model Development Process with *Roboflow*:

- ▪ *Roboflow* online platform offers a limited amount of usable performance metrics for comparison. Training graphs in particular provide an ideal data for comparison; however data cannot be exported, hence metrics of mAP@50, Precision and Recall were recorded for performance comparison between Experiment 1 and Experiment 2.
- ▪ The use of pretrained models, preconfigures algorithm parameters, which will positively influence test results of models. However, it will not be a true representation of the impact that photorealism from UE5 synthetic data has on CV model performance and may detract from findings for this investigation.

Chapter 4 – Results and Discussion

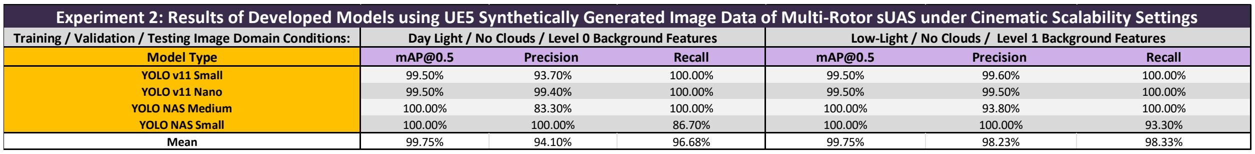
4.1 Stage 1 Results

The performance metrics (mAP@50, Precision, Recall) acquired from each trained model of Experiment 1 and Experiment 2 were accordingly tabled in *Microsoft* Excel spreadsheeting software, as seen in Figure 4.1 and Figure 4.2. Considering the volume of raw data, in order to effectively interpret the overall performance of CV models for each experiment, the overall mean for each type of performance metric, for each domain experimental domain condition was calculated.



Experiment 1: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Low Scalability Settings						
Training / Validation / Testing Image Domain Conditions:	Day Light / No Clouds / Level 0 Background Features			Low-Light / No Clouds / Level 1 Background Features		
Model Type	mAP@0.5	Precision	Recall	mAP@0.5	Precision	Recall
YOLO v11 Small	99.50%	93.50%	100.00%	99.50%	99.60%	100.00%
YOLO v11 Nano	99.50%	99.70%	100.00%	99.50%	99.30%	100.00%
YOLO NAS Medium	100.00%	100.00%	86.70%	100.00%	100.00%	86.70%
YOLO NAS Small	100.00%	78.90%	100.00%	100.00%	100.00%	100.00%
Mean	99.75%	93.03%	96.68%	99.75%	99.73%	96.68%

Figure 4.1: Experiment 1: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Low Scalability Settings



Experiment 2: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Cinematic Scalability Settings						
Training / Validation / Testing Image Domain Conditions:	Day Light / No Clouds / Level 0 Background Features			Low-Light / No Clouds / Level 1 Background Features		
Model Type	mAP@0.5	Precision	Recall	mAP@0.5	Precision	Recall
YOLO v11 Small	99.50%	93.70%	100.00%	99.50%	99.60%	100.00%
YOLO v11 Nano	99.50%	99.40%	100.00%	99.50%	99.50%	100.00%
YOLO NAS Medium	100.00%	83.30%	100.00%	100.00%	93.80%	100.00%
YOLO NAS Small	100.00%	100.00%	86.70%	100.00%	100.00%	93.30%
Mean	99.75%	94.10%	96.68%	99.75%	98.23%	98.33%

Figure 4.2: Experiment 2: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Cinematic Scalability Settings

4.2 Stage 1 Performance Analysis

Inspection of mean performance results from Figure 4.1 and Figure 4.2, indicates almost identical results for multi-rotor sUAS detection for Experiment 1, which used *Low* scalability settings and Experiment 2 which used *Cinematic* scalability settings. Noted differences were observed with the values of Precision, with a minor advantage of 1.08% for *Low* scalability settings over *Cinematic* scalability settings for a (Day Light / No Clouds / Level 0 Background Features) domain condition and a minor advantage of 1.50% for *Cinematic* scalability settings over *Low* scalability settings for a (Low-Light / No Clouds / Level 1 Background Features) domain condition.

***Note:** The observed percentage differences in performance for Precision do not provide a meaningful indication of the impact of photorealism provided by UE5 synthetic image data on detection models operating under different ends of lighting conditions and therefore can also be considered insignificant at this time.

From Stage 1 findings, it would be easy to disagree with the research question – *Can high-fidelity photorealism provided by Unreal Engine 5 bridge the synthetic-to-reality gap for improved multi-rotor drone detection?*, since the method of gauging the performance impact of high-fidelity photorealism provided by UE5 in synthetically generated training and validation data, on CV-based multi-rotor sUAS detection models, was by comparison of model performances, when developed with synthetic imagery rendered using *Low* and *Cinematic* scalability settings. Especially considering that these scalability settings impact UE5 *Lumen* global illumination and reflections, as well as ray traced shadows, which both theoretically and practically demonstrated a considerable impact on render photorealism (refer to Section 2.2.4 and Section 3.2.2 Inspection respectively). However, it can be stated that there is more to these results.

It should be noted that developed CV models for Experiment 1 and Experiment 2, using *Low* and *Cinematic* scalability settings respectively, both perform to a high degree of correctness with average results greater than 97% in terms of mAP@50, Precision and Recall. This exceeds or matches results in comparison to previous literature, such as major findings by Zhang, Jia and Ivrissimtzis (2020), who used high-fidelity photorealism derived from high resolution IBL (Type 3 Hybrid Synthetic Data) in tandem with similar domain randomisation strategies employed Experiment 1 and Experiment 2 for synthetic data capture, such as the randomisation of target object surface texture, material and colour, as well as the orientation of the virtual camera view, which resulted in Precision and Recall values of 98%. This unexpected performance could be attributed to the use of the *Microsoft* COCO model, which may have significantly supported model generalisation.

It is further interesting to note that a limited volume of training and validation samples were used for the development of models in Experiment 1 and Experiment 2, totalling 105 and 15 synthetic images. This contradicts that majority of works reviewed such as Damian et al. (2023), Movshovitz-Attias, Kanade and Sheikh (2016), and to a lesser extent Tremblay et al. (2018), who indicate in their work that training volumes are ideally in the tens of thousands. However, these works did not use pretrained models, which facilitates transfer learning that accelerates learning efficiency and model convergence with actual training data. Therefore, it can be stated that using pretrained models for this research experiment was likely a cause of bias, which positively skewed performance results, and are not representative of the performance impact of UE5 photorealism. However, the alternative interpretation of the results of the Stage 1 experiments is that the source domain and target domain of the Stage 1 experiments were ensured to be highly similar in the methodology to enhance detection performance and may have also contributed to the high model performance metrics, therefore providing a strong indication that the photorealism of UE5 training data can be used for the purposes of domain gap minimisation.

4.4 Stage 2 Methodology Expansion

Considering the uncertainty of the impact of model pretraining and or the domain similarity across datasets for Stage 1 experimentation. Stage 2 expanded the methodology of Stage 1. For Stage 2 it was maintained that the best way to gauge the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data, on CV-based multi-rotor sUAS detection models, was by comparison of model performances, when developed with synthetic imagery rendered using *Low* and *Cinematic* scalability settings. However, testing data would be more complex allowing for the demonstration of model generalisation capability based on the photorealism provided by UE5 synthetic data. As part of Stage 2, Experiments 3 – 6 were conducted.

For each experiment, training and validation datasets used synthetically generated data with Level 0 Background Features with *Low* and *Cinematic* scalability settings. Testing datasets used real-world images with classified Level 2 and Level 3 Background Features, as explained below and listed in Figure 4.3:

- **Level 2 Background Features:** Clouds and sun are background features behind multi-rotor sUAS in image. Sun is positioned anywhere within image, including directly behind. Objects, people, landscapes, and other elements are located below multi-rotor sUAS.

- **Level 3 Background Features:** Objects, people, landscapes and other elements are background features behind multi-rotor sUAS in image.

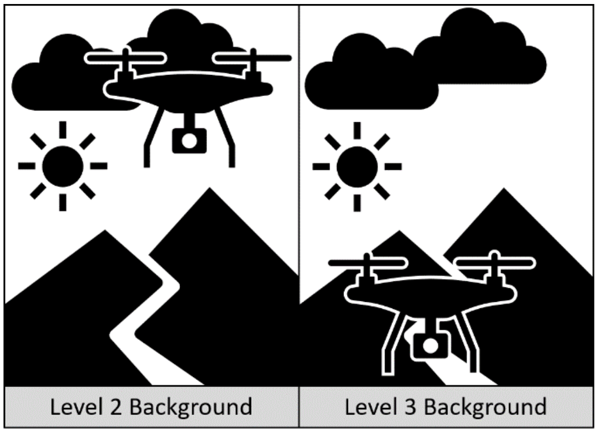


Figure 4.3: Level 2 and Level 3 Background Features

For clarification, it was decided that the following combination of lighting conditions and background features (domain conditions) for testing images with multi-rotor sUAS will be used for the development of CV model types for Experiments 3 - 6:

- (Day Light / No Clouds / Level 2 Background Features)
- (Day Light / Clouds / Level 3 Background Features)
- (Low-Light / Clouds / Level 2 Background Features)
- (Low-Light Sky / Clouds / Level 3 Background Features)

For reference, real-world images of multi-rotor sUAS with these domain conditions are presented below in Figure 3.4 and Figure 3.5 respectively. It should be noted that these combinations were specifically chosen due to their prevalence amongst real-world images.

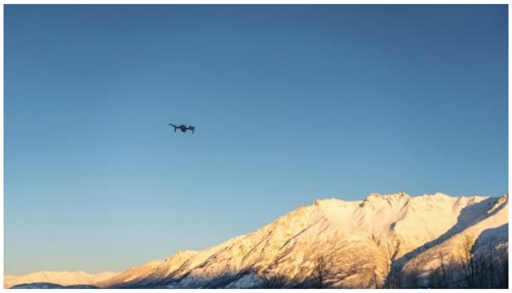


Figure 4.4: Multi-Rotor sUAS with Day Light / No Clouds / Level 2 Background Features)

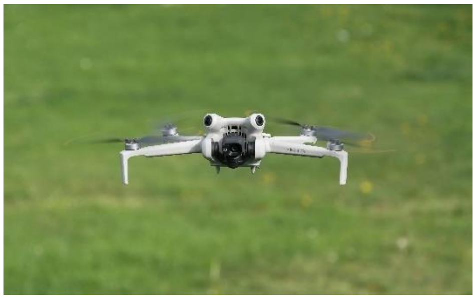


Figure 4.5: Multi-Rotor sUAS with Day Light / Clouds / Level 3 Background Features

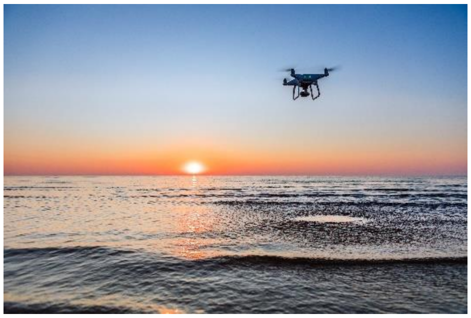


Figure 4.7: Multi-Rotor sUAS with Low-Light / Clouds / Level 2 Background Features

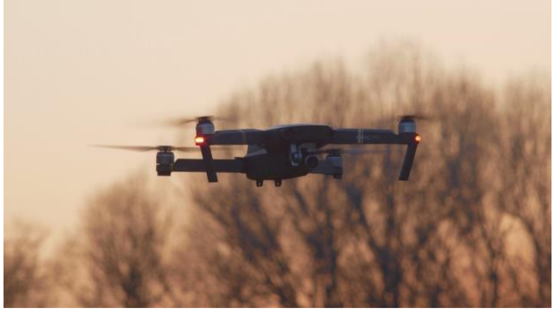


Figure 4.6: Multi-Rotor sUAS with Low-Light Sky / Clouds / Level 3 Background Features

Similar to Stage 1, multiple pretrained models were developed using *Roboflow*. For clarification the results of each experiment are presented below:

4.5 Stage 2 Results

Experiment 3: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Cinematic vs Low Scalability Settings						
Training / Validation Image Conditions: (Day Light / No Cloud / Level 0 Background Features) Testing Image Conditions: (Day Light / No Clouds / Level 2 Background Features)						
Testing Image Conditions:	Cinematic Settings: (Day Light / No Clouds / Level 2 Background Features)			Low Settings: (Day Light / No Clouds / Level 2 Background Features)		
Model	mAP	Precision	Recall	mAP	Precision	Recall
YOLO v11 Small	99.10%	92.90%	100%	99.50%	93.30%	100%
YOLO v11 Nano	99.50%	97.10%	100%	99.50%	99.60%	100%
YOLO NAS Medium	100%	100.00%	86.7%	100%	75.00%	100%
YOLO NAS Small	100%	88.2%	100.00%	100%	100%	100.00%
Mean	99.65%	94.55%	96.68%	99.75%	91.98%	100.00%

Experiment 4: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Cinematic vs Low Scalability Settings						
Training / Validation Image Conditions: (Day Light / No Clouds / Level 0 Background Features)						
Testing Image Conditions: (Day Light / Clouds / Level 3 Background Features)						
Testing Image Conditions:	Cinematic Settings: (Day Light / Clouds / Level 3 Background Features)			Low Settings: (Day Light / Clouds / Level 3 Background Features)		
Model	mAP	Precision	Recall	mAP	Precision	Recall
YOLO v11 Small	97.40%	91.90%	100.00%	99.50%	100.00%	100.00%
YOLO v11 Nano	99.50%	98.60%	100.00%	99.50%	99.70%	100.00%
YOLO NAS Medium	100.00%	100.00%	100.00%	100.00%	93.80%	100.00%
YOLO NAS Small	100.00%	88.20%	100.00%	100.00%	100.00%	93.30%
Mean	99.23%	94.68%	100.00%	99.75%	98.38%	98.33%

Experiment 5: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Cinematic vs Low Scalability Settings						
Training / Validation Image Conditions: (Low-Light Sky / Clouds / Level 1 Background Features)						
Testing Image Conditions: (Low-Light / Clouds / Level 2 Background Features)						
Testing Image Conditions:	Cinematic Settings: (Low-Light / Clouds / Level 2 Background Features)			Low Settings: (Low-Light / Clouds / Level 2 Background Features)		
Model	mAP	Precision	Recall	mAP	Precision	Recall
YOLO v11 Small	99.50%	99.70%	100.00%	99.50%	99.60%	100.00%
YOLO v11 Nano	99.50%	99.70%	100.00%	99.50%	99.50%	100.00%
YOLO NAS Medium	100.00%	100.00%	80.00%	100.00%	60.00%	100.00%
YOLO NAS Small	100.00%	100.00%	93.30%	100.00%	100.00%	100.00%
Mean	99.75%	99.85%	93.33%	99.75%	89.78%	100.00%

Experiment 6: Results of Developed Models using UE5 Synthetically Generated Image Data of Multi-Rotor sUAS under Cinematic vs Low Scalability Settings						
Training / Validation Image Conditions: (Low-Light Sky / Clouds / Level 1 Background Features)						
Testing Image Conditions: (Low-Light Sky / Clouds / Level 3 Background Features)						
Testing Image Conditions:	Cinematic Settings: (Low-Light Sky / Clouds / Level 3 Background Features)			Low Settings: (Low-Light Sky / Clouds / Level 3 Background Features)		
Model	mAP	Precision	Recall	mAP	Precision	Recall
YOLO v11 Small	99.50%	99.60%	100.00%	99.50%	100.00%	99.10%
YOLO v11 Nano	99.50%	99.70%	100.00%	99.50%	96.60%	100.00%
YOLO NAS Medium	100.00%	100.00%	100.00%	100.00%	88.20%	100.00%
YOLO NAS Small	100.00%	32.60%	100.00%	100.00%	100.00%	100.00%
Mean	99.75%	82.98%	100.00%	99.75%	96.20%	99.78%

4.6 Stage 2 Performance Analysis

The results of Stage 2 are similar to Stage 1, with multi-rotor detection models exhibiting high-level of correctness on average in terms of mAP@50, Precision and Recall values. Although, a distinguishable difference was also not observed between *Low* scalability settings and *High* scalability settings to gauge the performance impact of high-fidelity photorealism provided by UE5 in synthetically generated training and validation data, on CV-based multi-rotor sUAS detection models, this work is able to successfully demonstrate that UE5 can bridge the synthetic-to-reality gap for improved multi-rotor drone detection under various domain conditions with domain randomisation and the use of pretrained models.

4.7 Benefits of Research

This research will serve to benefit future C-sUAS research based on CV

4.8 Future Work / Limitations

To extend this work further, the following future work is recommended considering limitations:

- It is recommended that experimental methodologies in this research paper be repeated without the use of pretrained models, in order to truly understand the impact that high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data (Pure Synthetic Data), on CV models for the application of multi-rotor sUAS detection in the real-world.

- Taking the approach outlined in this research report, synthetic data acquisition of multi-rotor sUAS in an UE5 virtual environment is an extremely time-consuming process, considering the use of domain randomisation of parameters including the multi-rotor sUAS model, Cine Camera Actor, and environmental parameters. Future work should explore methods UE5 that can automate this process, which will serve to benefit the multi-rotor sUAS detection research community.

- This work relied entirely on *Roboflow*. While an effective platform future work should compare performance with other model training methods.

- Comparisons should be made between the high-fidelity photorealism provided by UE5 and other rendering engines to extend the research of this paper specifically.

Chapter 5 – Conclusion

Unreal Engine 5 (UE5) is a state-of-the-art rendering engine, which can not only be utilised for immersive game development, but also for the generation of high-fidelity photorealistic imagery. This is enabled by rendering capabilities such as *Lumen* global illumination and reflections, as well as ray traced shadows. Furthermore, this experimental research project provides preliminary investigation into the impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data (Pure Synthetic Data), on CV models for the application of multi-rotor sUAS detection in the real-world. Although findings were not able to explicitly quantify the exact impact that UE5 brings to improving drone detection, it has clearly demonstrated the ability to bridge the synthetic-to-reality gap with a domain randomisation strategy and model pretraining.

Report References

3D Studio 2024, *What is LOD: Level of Detail*, 3D Studio, viewed 14 October 2024, <https://3dstudio.co/3d-lod-level-of-detail/#:~:text=LOD%20or%20the%20level%20of,increase%20the%20efficiency%20of%20rendering.&text=Correspondingly%2C%20there%20are%20various%20levels,detailed%20version%20of%20the%20model.&text=If%20you're%20wondering%20whether,mesh%20optimization%20is%20handled%20differently.>.

3S Cloud Render Farm 2023, *Render Times: Understanding the Duration of Rendering at a Render Farm*, 3S Cloud Render Farm, https://3sfarm.com/render-times-understanding-the-duration-of-rendering-at-a-render-farm/#:~:text=Render%20Times%20and%20Tools,idea%20of%20the%20time%20needed.>.

Adejumo, J 2023, *Gradient Descent From Scratch- Batch Gradient Descent, Stochastic Gradient Descent, and Mini-Batch Gradient Descent.*, Medium, viewed 1 October 2024, <https://medium.com/@jaleeladejumo/gradient-descent-from-scratch-batch-gradient-descent-stochastic-gradient-descent-and-mini-batch-def681187473>.

Adobe 2024a, *The PBR Guide - Part 1*, Adobe, https://www.adobe.com/learn/substance-3d-designer/web/the-pbr-guide-part-1?locale=en&learnIn=1>.

Adobe 2024b, *Visual dictionary of 3D terms*, Adobe, viewed 8 October 2024, <https://helpx.adobe.com/beta/fuse/help/3d-visual-dictionary.html>.

Adobe 2024c, *What is a polygon in 3D modeling?*, Adobe, viewed 8 October 2024, <https://www.adobe.com/products/substance3d/discover/3d-polygon-modeling.html>.

Adobe 2024d, *A guide to 3D file types*, Adobe, viewed 8 October 2024, <https://www.adobe.com/au/products/substance3d/discover/3d-files-formats.html>.

Adobe 2024e, *Diffused light photography*, Adobe, viewed 27 October 2024, <https://www.adobe.com/creativecloud/photography/discover/diffused-light-photography.html>.

Adobe 2024f, *What is PBR (physically based rendering)?*, Adobe, viewed 2 October 2024, <https://www.adobe.com/au/products/substance3d/discover/pbr.html>.

Adobe 2024g, *What is chromatic aberration?*, Adobe, https://www.adobe.com/creativecloud/photography/discover/chromatic-aberration.html#:~:text=Chromatic%20aberration%2C%20also%20known%20as,of%20a%20bright%20blue%20sky.>.

Adobe 2024h, *A guide to global illumination*, Adobe, viewed 1 October 2024, <https://www.adobe.com/products/substance3d/discover/what-is-global-illumination.html>.

AeroExpo 2024, *Optical drone detection systems*, AeroExpo, viewed 11 October 2024, <https://www.aeroexpo.online/prod/controp-precision-technologies-ltd/product-169577-70988.html>.

Afaq, S & Rao, S 2020, 'Significance of epochs on training a neural network', *Int. J. Sci. Technol. Res*, vol. 9, no. 06, pp. 485-8.

Airservices Australia 2023, *Guide to our operations*, Airservices Australia, Canberra, https://www.airservicesaustralia.com/wp-content/uploads/12-058BKT_Guide-to-our-operations_WEB-1.pdf>.

AirSight 2019, *U.S. Drone Laws - Overview of US Drone Rules and Regulations In USA by State*, AirSight, Richardson, viewed 16 July 2024, <https://www.utsystem.edu/sites/default/files/offices/police/policies/USDroneLaws.pdf>.

Akyon, FC, Eryuksel, O, Ozfuttu, KA & Altinuc, SO 2021, 'Track boosting and synthetic data aided drone detection', *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, pp. 1-5.

Alaloul, WS & Qureshi, AH 2020, 'Data Processing Using Artificial Neural Networks', in D Harkut (ed.), *Dynamic Data Assimilation - Beating the Uncertainties*, https://www.intechopen.com/chapters/71673>.

Alauddin, M, Arunthavanathan, R, Amin, MT & Khan, F 2022, 'Methods in Chemical Process Safety', in F Khan, et al. (eds), Elsevier, vol. 6, ch 6, pp. 179-226.

Alghamdi, Y, Munir, A & La, HM 2021, 'Architecture, classification, and applications of contemporary unmanned aerial vehicles', *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 9-20.

Alnuaimi, AF & Albaldawi, TH 2024, 'An overview of machine learning classification techniques', *BIO Web of Conferences*, EDP Sciences, p. 00133.

Alpha3D 2023, *The Hitchhiker's guide to understanding digital 3D model file formats*, Alpha3D, viewed 8 October 2024, <https://www.alpha3d.io/3d-file-formats/>.

Alzubaidi, L, Zhang, J, Humaidi, AJ, Al-Dujaili, A, Duan, Y, Al-Shamma, O, Santamaría, J, Fadhel, MA, Al-Amidie, M & Farhan, L 2021, 'Review of deep learning: concepts, CNN architectures, challenges, applications, future directions', *Journal of big data*, vol. 8, pp. 1-74.

Amazon Web Services 2024, *Understand the hyperparameter tuning strategies available in Amazon SageMaker*, Amazon Web Services, viewed 30 October 2024, <https://docs.aws.amazon.com/sagemaker/latest/dg/automatic-model-tuning-how-it-works.html>.

AMD 2024, *Neural Supersampling and Denoising for Real-time Path Tracing*, AMD, viewed 27 September 2024, <https://gpuopen.com/learn/neural_supersampling_and_denoising_for_real-time_path_tracing/#:~:text=Noise%20in%20path%20tracing%20images,and%20shows%20visually%20annoying%20noise.>.

Antonio, UoTaS 2022, *Gradients*, University of Texas at San Antonio, viewed 24 September 2024, <https://mathresearch.utsa.edu/wiki/index.php?title=Gradients#:~:text=If%20the%20gradient%20of%20a,the%20greatest%20absolute%20directional%20derivative.>.

Apratim Sharma 2022, 'Counter-Unmanned Aircraft Systems (C-UAS): Future of Warfare', *Journal of Defence Studies*, vol. 16, no. 4, pp. 221–41.

Archxstudio 2024, *Comparing Offline and Real-Time Rendering: Pros and Cons*, Archxstudio, viewed 2 September 2024, <https://archxstudio.com/comparing-offline-and-real-time-rendering-pros-and-cons/#:~:text=Limited%20Photorealism:%20While%20real%2Dtime,compromises%20in%20real%2Dtime%20performance.>.

Arnold, C, Biedebach, L, Küpfer, A & Neunhoeffer, M 2024, 'The role of hyperparameters in machine learning models and how to tune them', *Political Science Research and Methods*, vol. 12, no. 4, pp. 841-8.

Askew, C & Araujo, A 2020, *Advancing Instance-Level Recognition Research*, Google LLC., viewed 31 September 2024, <https://research.google/blog/advancing-instance-level-recognition-research/>.

Australian Government 2024, *Civil Aviation Safety Regulations 1998*, Volume 3: Regulations 99.005–135.465, F2024C00377, Australian Government, Canberra.

Autodesk 2024, *For real-time rendering, GPUs deliver the most efficient results*, Autodesk, viewed 1 October 2024, <https://www.autodesk.com/solutions/gpu-rendering>.

Badillo, S, Banfai, B, Birzele, F, Davydov, II, Hutchinson, L, Kam-Thong, T, Siebourg-Polster, J, Steiert, B & Zhang, JD 2020, 'An introduction to machine learning', *Clinical pharmacology & therapeutics*, vol. 107, no. 4, pp. 871-85.

Bærentzen, JA & Aanæs, H 2002, *Generating Signed Distance Fields From Triangle Meshes*, Technical University of Denmark, viewed 13 October 2024, <https://www2.imm.dtu.dk/pubdb/edoc/imm1289.pdf>.

Bar-Ilan University n.d., *Illumination Models and Shading*, Bar-Ilan University, Ramat Gan, <https://u.cs.biu.ac.il/~kapaho/CG/09_shading.pdf>.

Barisic, A, Petric, F & Bogdan, S 2022, 'Sim2air-synthetic aerial dataset for uav monitoring', *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3757-64.

Bengio, Y, Goodfellow, I & Courville, A 2017, *Deep learning*, vol. 1, MIT press Cambridge, MA, USA.

Bernard, E 2021, *Introduction to Machine Learning*, 1 edn, Wolfram Media, Incorporated.

Bhatnagar, D 2022, *How Neural Networks Learn using Gradient Descent*, Medium, https://bhatnagar91.medium.com/how-neural-networks-learn-using-gradient-descent-f48c2e4079a6>.

Blagec, K, Kraiger, J, Frühwirt, W & Samwald, M 2023, 'Benchmark datasets driving artificial intelligence development fail to capture the needs of medical professionals', *Journal of Biomedical Informatics*, vol. 137, p. 104274.

Blender 2024, *Light Paths*, Blender, viewed 14 October 2024, <https://docs.blender.org/manual/en/latest/render/cycles/render_settings/light_paths.html>.

Boesch, G 2024, *What is Intersection over Union (IoU)?*, Viso.ai, viewed 27 September 2024, <https://viso.ai/computer-vision/intersection-over-union-iou/>.

Bone, E & Bolkcom, C 2003, *Unmanned Aerial Vehicles: Background and Issues for Congress*, ADA467807, Library of Congress, Congressional Research Service, Washington, D.C.

Bouatouch, K n.d., *Introduction to Computer Graphics*, IRISA, <https://www.irisa.fr/prive/kadi/Cours_LR2V/Cours/RayTracing.pdf>.

Boyat, AK & Joshi, BK 2015, 'A review paper: noise models in digital image processing', *arXiv preprint arXiv:1505.03489*.

Braiek, HB & Khomh, F 2024, 'Machine Learning Robustness: A Primer', *ArXiv*, vol. abs/2404.00897.

Breton, M 2019, 'Overview of two performance metrics for object detection algorithms'.

Brownlee, J 2019, *How Machine Learning Algorithms Work (they learn a mapping of input to output)*, Machine Learning Mastery, viewed 30 August 2024, <https://machinelearningmastery.com/how-machine-learning-algorithms-work/>.

Brownlee, J 2020, *4 Types of Classification Tasks in Machine Learning*, viewed 14 September 2024, <https://machinelearningmastery.com/types-of-classification-in-machine-learning/>.

Brownlee, J 2022, *Difference Between a Batch and an Epoch in a Neural Network*, Machine Learning Mastery, viewed 29 October 2024, <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>.

Buecheler, C 2001, *Tim Sweeney discusses the Unreal Engine*, GameSpy, viewed 10 October 2024, <https://web.archive.org/web/20010621195145/http://www.gamespy.com/legacy/articles/engineweek2_a.shtm>.

Bunker, RJ, Sullivan, JP & Kuhn, DA 2021, 'Use of weaponized consumer drones in Mexican crime war', *Small Wars Journal*, pp. 70-1.

Burger, J & Scuderi, E 2018, 'Unmanned & unregulated: Where are the privacy protections from drones?', *Law Society Journal*, no. 46, pp. 88 - 9.

Cabading, Z 2024, *What is Anti-Aliasing? A Comprehensive Guide to Smoother Gaming Graphics*, Hewlett-Packard, viewed 16 October 2024, <https://www.hp.com/us-en/shop/tech-takes/what-is-anti-aliasing#:~:text=A%3A%20Anti%2Daliasing%20works%20by,lines%20and%20reducing%20visible%20pixelation.>.

Calder, S 2023, *Five years on from Gatwick drone disruption, what happened – and could it happen again?*, Independent, viewed 13 August 2024, <https://www.independent.co.uk/travel/news-and-advice/gatwick-drone-flights-cancelled-passengers-b2466653.html>.

Calian, DA, Lalonde, JF, Gotardo, P, Simon, T, Matthews, I & Mitchell, K 2018, 'From faces to outdoor light probes', *Computer Graphics Forum*, Wiley Online Library, pp. 51-61.

Camilli, L 2015, *Emerging technologies, applications, regulations, and market challenges in the consumer aerial drone industry: A strategic analysis of the 3D Robotics Business Model*, San Francisco State University College of Business, San Francisco.

Cao, L & Gu, H 2019, *Introduction to Object Detection*, Lehigh University, Bethlehem, Pennsylvania.

Carlson, A, Skinner, KA, Vasudevan, R & Johnson-Roberson, M 2018, 'Modeling Camera Effects to Improve Deep

Vision for Real and Synthetic Data', *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0-.

Carnegie Mellon University 1998, *Ray Tracer Road Map*, Carnegie Mellon University, viewed 27 September 2024, <http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15864-s04/www/assignment4/roadmap/>.

Caulfield, B 2019a, *What Is Path Tracing?*, NVIDIA, viewed 22 June 2024, <https://blogs.nvidia.com/blog/what-is-path-tracing/>.

Caulfield, B 2019b, *What's the Difference Between Hardware- and Software-Accelerated Ray Tracing?*, NVIDIA, viewed 17 October 2024, <https://blogs.nvidia.com/blog/whats-the-difference-between-hardware-and-software-accelerated-ray-tracing/>.

Ceranic, I 2020, *What's in a cloud? A guide to what cloud formations can tell us about the weather*, Australian Broadcasting Corporation, viewed 1 November 2024, <https://www.abc.net.au/news/2020-04-20/a-guide-to-what-each-cloud-formation-means-for-weather/12157826>.

Chaos 2023, *Render Settings Explained - Quality vs. Render Time*, Chaos, viewed 14 September, <https://support.chaos.com/hc/en-us/articles/4409180217361-Render-Settings-Explained-Quality-vs-Render-Time>.

Chavez, K & Swed, O 2020, 'Off the shelf: the violent nonstate actor drone threat', *Air & Space Power Journal*, vol. 34, no. 3, pp. 29-43.

Chen, L, Chen, P & Lin, Z 2020, 'Artificial intelligence in education: A review', *IEEE Access*, vol. 8, pp. 75264-78.

City University of Hong Kong n.d., *Refraction of Light*, City University of Hong Kong, Hong Kong, <https://www.cityu.edu.hk/phy/appkchu/ap5301/Refraction%20of%20light.pdf>.

Collins, J, Brown, R, Leitner, J & Howard, D 2020, 'Traversing the reality gap via simulator tuning', *ArXiv*.

Cook, RL 1989, 'Stochastic Sampling in Computer Graphics', in AS Glassner (ed.), *An introduction to ray tracing*, 1.4 edn, Morgan Kaufmann.

Csáji, BC 2001, 'Approximation with artificial neural networks', *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, no. 48, p. 7.

Cummings, AR, McKee, A, Kulkarni, K & Markandey, N 2017, 'The rise of UAVs', *Photogrammetric Engineering & Remote Sensing*, vol. 83, no. 4, pp. 317-25.

Cunningham, P, Cord, M & Delany, SJ 2008, 'Supervised Learning', in M Cord & P Cunningham (eds), *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 21-49.

Damian, A, Filip, C, Nistor, A, Petrariu, I, Mariuc, C & Stratan, V 2023, 'Experimental Results on Synthetic Data Generation in Unreal Engine 5 for Real-World Object Detection', *2023 17th International Conference on Engineering of Modern Electric Systems (EMES)*, IEEE, pp. 1-4.

Danczuk, J 2023, 'Bayraktars and Grenade-Dropping Quadcopters: How Ukraine and Nagorno-Karabakh Highlight Present Air and Missile Defense Shortcomings and the Necessity of Unmanned Aircraft Systems', *Military Review*, no. July-August 2023, pp. 21 - 33.

Davila, AL 2020, 'Neural Networks: The Backpropagation Algorithm', William & Mary

De Cubber, G 2019, 'Explosive drones: How to deal with this new threat?', *Proceedings of the 9th International Workshop on Measurement, Prevention, Protection and Management of CBRN Risks*, International CBRNE Institute, Belgium, pp. 1-8.

de Melo, CM, Torralba, A, Guibas, L, DiCarlo, J, Chellappa, R & Hodgins, J 2022, 'Next-generation deep learning based on simulators and synthetic data', *Trends in cognitive sciences*, vol. 26, no. 2, pp. 174-87.

De Vries, J 2017, *Learn OpenGL - Graphics Programming*, 3 edn, Kendal Welling, viewed 19 October 2024, <https://learnopengl.com/Introduction>.

Debevec, P 2004, *Light Probe Image Gallery*, Paul Debevec, viewed 20 September 2024, <https://pauldebevec.com/Probes/>.

Debevec, P 2006, 'Image-based lighting', in *ACM SIGGRAPH 2006 Courses*, pp. 4-es.

Debevec, P 2008, 'Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography', in *Acm siggraph 2008 classes*, pp. 1-10.

Del Moral, P, Nowaczyk, S & Pashami, S 2022, 'Why is multiclass classification hard?', *IEEE Access*, vol. 10, pp. 80448-62.

Demuth, H, Beale, M & Hagan, M 2010, *Backpropagation*, 6.04 edn, Neural Network Toolbox™ 6 User's Guide, MathWorks, Massachusetts, viewed 24 September 2024, <https://kashanu.ac.ir/Files/Content/neural_network_toolbox_6.pdf>.

Department of Infrastructure, Transport, Regional Development, Communications and the Arts, 2024, *Drone Privacy Guidelines*, Department of Infrastructure, Transport, Regional Development, Communications and the Arts, Canberra, viewed 18 July 2024, <https://www.drones.gov.au/sites/default/files/documents/Drone%20Privacy%20Guidelines.pdf>.

Deshpande, S 2024, *Epoch, Iteration and Batch in Deep Learning*, OpenGenus IQ, viewed 30 October 2024, <https://iq.opengenus.org/epoch-iteration-and-batch-in-deep-learning/>.

Dieter, T, Weinmann, A & Brucherseifer, E 2023, 'Generating synthetic data for deep learning-based drone detection', *AIP Conference Proceedings*, AIP Publishing.

Dieter, TR, Weinmann, A, Jäger, S & Brucherseifer, E 2023, 'Quantifying the Simulation–Reality Gap for Deep Learning-Based Drone Detection', *Electronics*, vol. 12, no. 10, p. 2197.

DJI 2016, *Inside a Drone - Propellers*, DJI, viewed 5 July 2024, <https://www.dji.com/au/newsroom/news/inside-a-drone-propellers#:~:text=For%20instance%2C%20the%20props%20of,a%20high%20performance%20base%20material.>.

DJI 2024a, *Support for Mavic Air*, DJI, viewed 12 July 2024, <https://www.dji.com/au/support/product/mavic-air>.

DJI 2024b, *DJI Mavic 3 Pro Fly More Combo w. DJI RC*, DJI, viewed 7 October 2024, <https://www.d1store.com.au/products/dji-mavic-3-pro-fmc-dji-rc>.

DJI 2024c, *DJI Mini 3 Pro w. DJI RC*, DJI, viewed 20 Sepember 2024, <https://www.d1store.com.au/products/dji-mavic-mini-3-pro-w-dji-rc>.

DJI 2024d, *DJI Mini 2 SE*, DJI, viewed 7 October 2024, <https://store.dji.com/au/product/dji-mini-2-se?vid=132621>.

DJI 2024e, *DJI Matrice 300 RTK*, Digital Image of DJI Matrice 300 RTK DJI, <https://enterprise.dji.com/matrice-300>.

DJI 2024f, *Phantom 4 Pro V2.0*, DJI, viewed 7 October 2024, <https://www.dji.com/au/phantom-4-pro-v2>.

Doan, A-D, Nguyen, BL, Gupta, S, Reid, I, Wagner, M & Chin, T-J 2024, 'Assessing domain gap for continual domain adaptation in object detection', *Computer Vision and Image Understanding*, vol. 238, p. 103885.

Du-Harpur, X, Watt, FM, Luscombe, NM & Lynch, MD 2020, 'What is AI? Applications of artificial intelligence to dermatology', *British Journal of Dermatology*, vol. 183, no. 3, pp. 423-30.

Dwibedi, D, Misra, I & Hebert, M 2017, 'Cut, paste and learn: Surprisingly easy synthesis for instance detection', *Proceedings of the IEEE international conference on computer vision*, pp. 1301-10.

Dynamic Graphics Project n.d., *Lighting & Reflection models*, Dynamic Graphics Project, Toronto, <https://www.dgp.toronto.edu/~karan/courses/418/slides/lecture7.compressed.pdf>.

Echodyne 2024, *What Drone Detection Solution Does Your Organization Need?*, Echodyne, viewed 11 October 2024, <https://www.echodyne.com/resources/news-events/what-drone-detection-solution-do-you-need/>.

Eck, DJ 2021, *Introduction to Computer Graphics*, 1.4 edn, Hobart and William Smith College, https://math.hws.edu/graphicsbook/c8/s1.html>.

Eclat Digital 2024, *Path Tracing vs Ray Tracing: What's the difference for accurate rendering?*, Eclat Digital, viewed 27 September 2024, <https://eclat-digital.com/path-tracing-vs-ray-tracing-whats-the-difference-for-accurate-rendering/#:~:text=While%20rasterization%20techniques%20can%20be,visuals%20in%20the%20digital%20world.>.

Edge AI and Vision Alliance 2024, *Mean Average Precision (mAP): Common Definition, Myths and Misconceptions* Edge AI and Vision Alliance viewed 31 September 2024, <https://www.edge-ai-vision.com/2023/08/mean-average-precision-map-common-definition-myths-and-misconceptions/>.

El-Nouby, A, Izacard, G, Touvron, H, Laptev, I, Jegou, H & Grave, E 2021, 'Are large-scale datasets necessary for self-supervised pre-training?', *arXiv preprint arXiv:2112.10740*.

El Emam, K 2020, *Accelerating AI with Synthetic Data: Generating Data for AI Projects*, O'Reilly Media, Incorporated.

Elhoseiny, M, El-Gaaly, T, Bakry, A & Elgammal, A 2015, 'Convolutional models for joint object categorization and pose estimation', *arXiv preprint arXiv:1511.05175*.

Elsayed, M, Reda, M, Mashaly, AS & Amein, AS 2021, 'Review on real-time drone detection based on visual band electro-optical (EO) sensor', *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, IEEE, pp. 57-65.

Encord 2023a, *Training, Validation, Test Split for Machine Learning Datasets*, Encord, viewed 29 October 2024, <https://encord.com/blog/train-val-test-split/>.

Encord 2023b, *F1 Score in Machine Learning*, Ecord, viewed 1 October, <https://encord.com/blog/f1-score-in-machine-learning/>.

Encord 2023c, *Pre Trained Model*, Encord, viewed 24 September 2024, <https://encord.com/glossary/pre-trained-model-definition/#:~:text=A%20pre%2Dtrained%20model%20is,tuned%20for%20a%20specific%20task.>.

Epic Games 2023, *Lighting In Unreal Engine 5 For Educators and Students*, Cary, North Carolina, viewed 21 July 2024, <https://cdn2.unrealengine.com/lighting-for-educators-and-students-20mb-6b793664ebe8.pdf>.

Epic Games 2024a, *Cine Camera Actor*, Epic Games, viewed 4 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/cinematic-cameras-in-unreal-engine>.

Epic Games 2024b, *Scalability Reference*, Epic Games, viewed 2 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/scalability-reference-for-unreal-engine>.

Epic Games 2024c, *Sky Atmosphere Component*, Epic Games,, viewed 25 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/sky-atmosphere-component-in-unreal-engine>.

Epic Games 2024d, *Exponential Height Fog*, Epic Games, viewed 27 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/exponential-height-fog-in-unreal-engine>.

Epic Games 2024e, *Volumetric Cloud Material*, Epic Games, viewed 26 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/volumetric-cloud-material-in-unreal-engine>.

Epic Games 2024f, *What is real-time ray tracing, and why should you care?*, Epic Games, viewed 27 September 2024, <https://www.unrealengine.com/en-US/explainers/ray-tracing/what-is-real-time-ray-tracing>.

Epic Games 2024g, *Customizing Device Profiles and Scalability for Android*, Epic Games, viewed 3 November 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/customizing-device-profiles-and-scalability-in-unreal-engine-projects-for-android>.

Epic Games 2024h, *Hardware Ray Tracing*, Epic Games, viewed 20 September 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/hardware-ray-tracing-in-unreal-engine>.

Epic Games 2024i, *Simulation Blank Template*, Epic Games, viewed 2 November 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/simulation-blank-template-in-unreal-engine>.

Epic Games 2024j, *Physical Materials Reference*, Epic Games, viewed 20 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/physical-materials-reference-for-unreal-engine>.

Epic Games 2024k, *Light Types and Their Mobility*, Epic Games, viewed 15 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/light-types-and-their-mobility-in-unreal-engine>.

Epic Games 2024l, *Sky Lights*, Epic Games, viewed 31 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/sky-lights-in-unreal-engine>.

Epic Games 2024m, *Volumetric Cloud Component*, Epic Games,, viewed 26 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/volumetric-cloud-component-in-unreal-engine>.

Epic Games 2024n, *Directional Lights*, Epic Games, viewed 26 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/directional-lights-in-unreal-engine>.

Epic Games 2024o, *Environmental Light with Fog, Clouds, Sky and Atmosphere*, Epic Games, viewed 3 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/environmental-light-with-fog-clouds-sky-and-atmosphere-in-unreal-engine>.

Epic Games 2024p, *Landscape Technical Guide*, Epic Games, viewed 15 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/landscape-technical-guide-in-unreal-engine>.

Epic Games 2024q, *Shading Models*, Epic Games, viewed 25 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/shading-models-in-unreal-engine>.

Epic Games 2024r, *Texture Streaming Configuration*, Epic Games, viewed 26 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/texture-streaming-configuration-in-unreal-engine>.

Epic Games 2024s, *Texture Streaming Overview*, Epic Games, viewed 26 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/texture-streaming-overview-for-unreal-engine>.

Epic Games 2024t, *Screen Space Global Illumination*, Epic Games, viewed 26 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/screen-space-global-illumination-in-unreal-engine>.

Epic Games 2024u, *Auto Exposure (Eye Adaptation)*, Epic Games, viewed 24 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/auto-exposure-eye-adaptation?application_version=4.27>.

Epic Games 2024v, *Lens Flare*, Epic Games, viewed 19 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/lens-flare?application_version=4.27>.

Epic Games 2024w, *Depth of Field*, Epic Games, viewed 22 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/depth-of-field-in-unreal-engine>.

Epic Games 2024x, *Post Process Effects*, Epic Games, viewed 14 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/post-process-effects?application_version=4.27>.

Epic Games 2024y, *Hardware and Software Specifications*, Epic Games, viewed 20 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/hardware-and-software-specifications-for-unreal-engine>.

Epic Games 2024z, *Virtual Shadow Maps*, Epic Games, viewed 7 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/virtual-shadow-maps-in-unreal-engine#limitationsofshadowmapraytracing>.

Epic Games 2024aa, *Shadowing*, Epic Games, viewed 30 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/shadowing-in-unreal-engine>.

Epic Games 2024ab, *Lumen Performance Guide*, Epic Games, https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-performance-guide-for-unreal-engine>.

Epic Games 2024ac, *Unreal Engine*, Epic Games, viewed 20 October 2024, <https://www.fab.com/channels/unreal-engine>.

Epic Games 2024ad, *Mesh Distance Fields*, Epic Games, viewed 12 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/mesh-distance-fields-in-unreal-engine>.

Epic Games 2024ae, *Hardware and Software Specifications*, Epic Games, viewed 2 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/hardware-and-software-specifications-for-unreal-engine>.

Epic Games 2024af, *Lumen Technical Details*, Epic Games, viewed 10 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-technical-details-in-unreal-engine>.

Epic Games 2024ag, *Unreal Engine Marketplace is now Fab*, Epic Games, viewed 20 October 2024, <https://www.unrealengine.com/marketplace/en-US/>.

Epic Games 2024ah, *HDRI Backdrop*, Epic Games, viewed 30 September 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/hdri-backdrop?application_version=4.27>.

Epic Games 2024ai, *Rendering High Quality Frames with Movie Render Queue*, Epic Games, viewed 3 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/rendering-high-quality-frames-with-movie-render-queue-in-unreal-engine>.

Epic Games 2024aj, *Plugins*, Epic Games, viewed 20 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/plugins-in-unreal-engine>.

Epic Games 2024ak, *Material Editor Guide*, Epic Games, viewed 2 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-material-editor-user-guide>.

Epic Games 2024al, *Introduction to Blueprints*, Epic Games, viewed 3 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/introduction-to-blueprints?application_version=4.27>.

Epic Games 2024am, *Viewport Basics*, Epic Games, viewed 19 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/viewport-basics?application_version=4.27>.

Epic Games 2024an, *Introduction to Blueprints*, Epic Games, viewed 2 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/introduction-to-blueprints-visual-scripting-in-unreal-engine>.

Epic Games 2024ao, *Components*, Epic Games, viewed 3 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/components-in-unreal-engine>.

Epic Games 2024ap, *Components*, Epic Games, viewed 2 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/components?application_version=4.27>.

Epic Games 2024aq, *Actors and Geometry*, Epic Games, viewed 2 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/actors-and-geometry-in-unreal-engine>.

Epic Games 2024ar, *Post Process Effects*, Epic Games, viewed 23 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/post-process-effects-in-unreal-engine>.

Epic Games 2024as, *Assets and Content Packs*, Epic Games, viewed 7 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/assets-and-content-packs-in-unreal-engine>.

Epic Games 2024at, *Levels*, Epic Games, viewed 7 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/levels-in-unreal-engine>.

Epic Games 2024au, *Level Editor*, Epic Games, https://dev.epicgames.com/documentation/en-us/unreal-engine/level-editor-in-unreal-engine>.

Epic Games 2024av, *Lumen Global Illumination and Reflections*, Epic Games, viewed 25 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-global-illumination-and-reflections-in-unreal-engine#lumenreflections>.

Epic Games 2024aw, *Physically Based Materials*, Epic Games, viewed 25 October 2024, <https://dev.epicgames.com/documentation/en-us/unreal-engine/physically-based-materials-in-unreal-engine>.

Evident Scientific 2024, *Introduction to the Reflection of Light*, Evident Scientific, viewed 1 October 2024, <https://www.olympus-lifescience.com/en/microscope-resource/primer/lightandcolor/reflectionintro/>.

EyeEm n.d., *Low angle view of silhouette helicopter against sky*, Freepik, viewed 5 July 2024, <https://www.freepik.com/premium-photo/low-angle-view-silhouette-helicopter-against-sky_101475371.htm#fromView=search&page=45&position=36&uuid=70aeea5a-79ba-4816-b631-caf88cb4cfa9>.

Federal Aviation Administration 2016, *Remote Pilot – Small Unmanned Aircraft Systems Study Guide*, FAA-G-8082-22, United States Department of Transportation, Washington, D.C., https://www.faa.gov/sites/faa.gov/files/regulations_policies/handbooks_manuals/aviation/remote_pilot_study_guide.pdf>.

Fetaya, E, Lucas, J & Andrews, E n.d., *CSC 411 Lecture 10: Neural Networks I*, University of Toronto, Toronto.

Fiesler, E & Beale, R 1996, 'Neural network topologies', *The Handbook of Neural Computation, E. Fiesler and R. Beale (Editors-in-Chief), Oxford University Press and IOP Publishing*.

Floreano, D & Wood, RJ 2015, 'Science, technology and the future of small autonomous drones', *nature*, vol. 521, no. 7553, pp. 460-6.

Florence, PR, Carter, J, Ware, J & Tedrake, R 2018, 'Nanomap: Fast, uncertainty-aware proximity queries with lazy search over local 3d data', *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 7631-8.

Foley, JD & Dam, AV n.d., *Illumination Models and Shading*, Brandeis University, Waltham, Massachusetts, <https://www.cs.brandeis.edu/~cs155/Lecture_16.pdf>.

Friese, L, Jenzen-Jones, NR & Smallwood, M 2016, *Emerging Unmanned Threats: The use of commercially-available UAVs by armed non-state actors*, Perth, Australia.

Garg, P 2022, 'Characterisation of Fixed-Wing Versus Multirotors UAVs/Drones', *Journal of Geomatics*, vol. 16, no. 2, pp. 152-9.

Gastelum, ZN, Shead, T & Higgins, M 2020, *Synthetic Training Images for Real-World Object Detection*, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Geometric Photography 2022, *Low-Angle Shot of a White Drone Camera Flying in the Blue Sky*, Pexels,, viewed 1 July 2024, <https://www.pexels.com/photo/low-angle-shot-of-a-white-drone-camera-flying-in-the-blue-sky-12975479/>.

Gerasimov, PS 2004, 'Omnidirectional Shadow Mapping', in R Fernando (ed.), *GPU Gems: Programming techniques, tips, and tricks for real-time graphics*, Addison-Wesley Reading, Boston.

Google 2024, *Classification: Accuracy, recall, precision, and related metrics*, Google, viewed 5 October 2024, <https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall>.

Government of Canada 2021, *Privacy guidelines for drone users*, Government of Canada, viewed 18 July 2024, <https://tc.canada.ca/en/aviation/drone-safety/learn-rules-you-fly-your-drone/privacy-guidelines-drone-users#toc2>.

Gradient 2021, *Counter UAS*, Gradient, viewed 11 August 2024, <https://www.gradiant.org/wp-content/uploads/2021/07/Counter-UAS-EN.pdf>.

Gunther, E 2024, *Ultra Dynamic Sky*, Epic Games,, viewed 1 September 2024, <https://www.fab.com/listings/84fda27a-c79f-49c9-8458-82401fb37cfb>.
Guo, Y, Wang, H, Hu, Q, Liu, H, Liu, L & Bennamoun, M 2020, 'Deep learning for 3d point clouds: A survey', *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4338-64.
Gupta, H, Kotlyar, O, Andreasson, H & Lilienthal, AJ 2024, 'Robust Object Detection in Challenging Weather Conditions', *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7523-32.
Gustafsson, H 2023, 'Exploration of performance evaluation metrics with deep-learning-based generic object detection for robot guidance systems', Bachhelor thesis, Linköping University, Linköping, Sweden.
Hain, E n.d., *Ray and Path Tracing Today*, Rea-Time Rendering, <https://www.realtimerendering.com/erich/Ray_and_Path_Tracing_Today.pdf>.
Hao, Y, Pei, H, Lyu, Y, Yuan, Z, Rizzo, J-R, Wang, Y & Fang, Y 2023, 'Understanding the impact of image quality and distance of objects to object detection performance', *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 11436-42.
Haugstvedt, H 2024, 'Still Aiming at the Harder Targets: An Update on Violent Non State Actors' Use of Armed UAVs', *Perspectives on terrorism*, vol. 18, no. 1, pp. 132 - 43.
Haykin, SS 2009, *Neural Networks and Learning Machines*, 3 edn, Pearson Education.
He, K, Girshick, R & Dollár, P 2019, 'Rethinking imagenet pre-training', *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4918-27.
Heiets, I, Kuo, Y-W, La, J, Yeun, RC & Verhagen, W 2023, 'Future Trends in UAV Applications in the Australian Market', *Aerospace*, vol. 10, no. 6, p. 555.
Hobart and William Smith Colleges n.d., *CS424 Ray Tracing*, Hobart and William Smith Colleges, <https://math.hws.edu/eck/cs424/graphicsbook-1.1/c8/s1.html>.
Hoberock, J 2007, 'High-Quality Ambient Occlusion', in H Nguyen (ed.), *GPU Gems 3*, Addison-Wesley Professional.
Hocevar, S 2017, *Tutorial 16 : Shadow mapping*, Sam Hocevar, viewed 16 October 2024, <https://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>.
Hock-Chuan, C 2012, *3D Graphics with OpenGL Basic Theory*, Nanyang Technological University, viewed 15 October 2024, <https://personal.ntu.edu.sg/ehchua/programming/opengl/CG_BasicsTheory.html#zz-3.4>.
Hofmann, C & Schneckener, U 2011, 'Engaging non-state armed actors in state-and peace-building: options and strategies', *International review of the Red Cross*, vol. 93, no. 883, pp. 603-21.
Hossain, R & Timmer, D 2021, 'Machine learning model optimization with hyper parameter tuning approach', *Glob. J. Comput. Sci. Technol. D Neural Artif. Intell*, vol. 21, no. 2, p. 31.
Houdini n.d., *Motion blur*, Houdini, viewed 19 October 2024, <https://www.sidefx.com/docs/houdini/render/blur.html>.
Hsiao, E 2013, 'Addressing Ambiguity In Object Instance Detection', Doctor of Philosophy thesis, Carnegie Mellon University, Pittsburgh.
Hu, J, Choe, G, Nadir, Z, Nabil, O, Lee, S-J, Sheikh, H, Yoo, Y & Polley, M 2020, 'Sensor-realistic synthetic data engine for multi-frame high dynamic range photography', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 516-7.
Hunter-Perkins, R, Shiotani, H & Rogers, J 2023, *Global Report on the Acquisition, Weaponization and Deployment of Unmanned Aircraft Systems by Non-State Armed Groups for Terrorism related Purpose*, United Nations Office of Counter-Terrorism, Conflict Armament Research, Global Counter-Terrorism Programme on Autonomous and Remotely Operated Systems (AROS Programme)

Image Engineering 2024, *Chromatic Aberration*, Image Engineering, viewed 16 October 2024, <https://www.image-engineering.de/library/image-quality/factors/1074-chromatic-aberration#:~:text=Chromatic%20aberration%20is%20caused%20by,or%20bent%20at%20the%20boundaries.>.

Imperial Collage n.d., *Ray Tracing*, Imperial Collage, London, <https://www.doc.ic.ac.uk/~bkainz/graphics/notes/GraphicsNotes1011.pdf>.

Intel 2024, *What Is Anisotropic Filtering?*, Intel, viewed 25 October 2024, <https://www.intel.com/content/www/us/en/gaming/resources/what-is-anisotropic-filtering.html#:~:text=Anisotropic%20filtering%20makes%20distant%20objects,visual%20impact%20in%20different%20games.>.

International Business Machines Corporation 2024, *Structured vs unstructured data*, International Business Machines Corporation, viewed 8 September 2024, <https://www.ibm.com/think/topics/structured-vs-unstructured-data>.

Ioannidou, A, Chatzilari, E, Nikolopoulos, S & Kompatsiaris, I 2017, 'Deep learning advances in computer vision with 3d data: A survey', *ACM computing surveys (CSUR)*, vol. 50, no. 2, pp. 1-38.

Iontcheva, I 2024, *What Is PBR (Physically-Based Rendering)? A complete guide*, https://www.chaos.com/blog/what-is-pbr-physically-based-rendering-a-complete-guide?srsltid=AfmBOopcvMYPPZG-40_7YxGIUqGxza6POwejOmBu7STdVKA7kaeBJwwO>.

iRender 2020, *Factors that affects the rendering performance in 3D software*, iRender viewed 30 July 2024, <https://irendering.net/factors-that-affects-the-rendering-performance-in-3d-software/>.

Jain, R, Kasturi, R & Schunck, BG 1995, *Machine Vision*, vol. 1, McGraw-Hill, Inc.

Jakob, W 2013, *Mitsuba Documentation*, Mitsuba, viewed 2 September 2024, <https://www.mitsuba-renderer.org/releases/0.4.4/documentation_lowres.pdf>.

Janiesch, C, Zschech, P & Heinrich, K 2021, 'Machine learning and deep learning', *Electronic Markets*, vol. 31, no. 3, pp. 685-95.

Jensen, KT 2023, *25 Years Later: The History of Unreal and an Epic Dynasty*, PC Magazine, viewed 10 October 2024, <https://www.pcmag.com/news/25-years-later-the-history-of-unreal-and-an-epic-dynasty>.

Jeon, W, Ko, G, Lee, J, Lee, H, Ha, D & Ro, WW 2021, 'Deep learning with GPUs', in *Advances in Computers*, Elsevier, vol. 122, pp. 167-215.

Jkk 2013, *Chromatic aberration with detail*.

Jones, MW 1995, '3D distance from a point to a triangle', *Department of Computer Science, University of Wales Swansea Technical Report CSR-5*, p. 5.

Jones, MW & Satherley, R 2001, 'Using distance fields for object representation and rendering', *Proc. 19th Ann. Conf. of Eurographics (UK Chapter)*, London, pp. 37-44.

Joon, JS 2010, 'Principles of Photorealism to develop photorealistic visualisation for Interface Design: A review', *2010 Seventh International Conference on Computer Graphics, Imaging and Visualization*, IEEE, pp. 17-25.

Joyce, D 2014, *Math 131 Directional derivatives, steepest ascent, tangent planes*, Clark University, Worcester, Massachusetts.

JSFILMZ 2023, *Unreal Engine 5.3 Matrix Demo with High Quality Reflections*, online video, https://www.youtube.com/watch?v=biVmlI6pAw8>.

Jung, H & Pedram, M 2010, 'Supervised learning based power management for multicore processors', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 9, pp. 1395-408.

Jurafsky, D & Martin, J 2024, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3 edn, Prentice Hall, Hoboken, New Jersey.

Kaggle 2024, *Flickr 8k Dataset*, Kaggle, viewed 1 October 2024, <https://www.kaggle.com/datasets/adityajn105/flickr8k>.

Kalogirou, SA 2001, 'Artificial neural networks in renewable energy systems applications: a review', *Renewable and sustainable energy reviews*, vol. 5, no. 4, pp. 373-401.

Kamilaris, A, van den Brink, C & Karatsiolis, S 2019, 'Training deep learning models via synthetic data: Application in unmanned aerial vehicles', *Computer Analysis of Images and Patterns: CAIP 2019 International Workshops, ViMaBi and DL-UAV, Salerno, Italy, September 6, 2019, Proceedings 18*, Springer, pp. 81-90.

Karhunen, J, Raiko, T & Cho, K 2015, 'Unsupervised deep learning: A short review', *Advances in independent component analysis and learning machines*, pp. 125-42.

Kashiyama, T, Sobue, H & Sekimoto, Y 2020, 'Sky monitoring system for flying object detection using 4K resolution camera', *Sensors*, vol. 20, no. 24, p. 7071.

Keller, A, Viitanen, T, Barré-Brisebois, C, Schied, C & McGuire, M 2019, 'Are we done with ray tracing?', *SIGGRAPH Courses*, pp. 3:1-3:381.

Khanday, NY & Sofi, SA 2021, 'Taxonomy, state-of-the-art, challenges and applications of visual understanding: A review', *Computer Science Review*, vol. 40, p. 100374.

Kluge, S & Staadt, O 2024, 'Assessing Photorealism of Rendered Objects in Real-World Images: A Transparent and Reproducible User Study', *arXiv preprint arXiv:2407.01767*.

Kneen, T 2020, *Weaponizing Unmanned Aircraft Systems*, DSIAC-2020-1319, Defense Systems Information Analysis Center, Belvoir, Virginia.

Kniazieva, Y 2023, *Image Classification vs. Object Detection: Key Differences*, Label Your Data, viewed 31 October 2024, <https://labelyourdata.com/articles/object-detection-vs-image-classification>.

Komura, T 2011, *Inf4/MSc Lecture 8 Hidden Surface Removal*, University of Edinburgh, <https://www.inf.ed.ac.uk/teaching/courses/cg/lectures/lect8cg2011handouts.pdf>.

Kovalev, I, Voroshilova, A & Karaseva, M 2019, 'Analysis of the current situation and development trend of the international cargo UAVs market', *Journal of Physics: Conference Series*, IOP Publishing, p. 055095.

Koziarski, M & Cyganek, B 2018, 'Impact of low resolution on image recognition with deep neural networks: An experimental study', *International Journal of Applied Mathematics and Computer Science*, vol. 28, no. 4, pp. 735-44.

Kundu, R 2022, *What is Domain Adaptation in Computer Vision and why is it important? Explore Domain Adaptation techniques and improve your models' performance to build more accurate AI faster.*, V7 Labs, viewed 5 October 2024, <https://www.v7labs.com/blog/domain-adaptation-guide>.

Kvaal, K & McEwan, JA 1996, 'Analysing complex sensory data by non-linear artificial neural networks', in *Data Handling in Science and Technology*, Elsevier, vol. 16, pp. 103-33.

Lacher, A, Baron, J, Rotner, J & Balazs, M 2019, *Small Unmanned Aircraft: Characterizing the Threat*, Defense Technical Information Center.

Lambers, M n.d., *Path Tracing*, Martin Lambers, <https://marlam.de/path-tracing/path-tracing-02.pdf>.

Lazebnik, S 2021, 'Object Class Recognition (Categorization)', in K Ikeuchi (ed.), *Computer vision: A reference guide*, Springer, pp. 533–6.

LeCun, Y, Bengio, Y & Hinton, G 2015, 'Deep learning', *nature*, vol. 521, no. 7553, pp. 436-44.

Lehtinen, J & Durand, F n.d., *Material Appearance and Shading*, Aalto University, Espoo, <https://users.aalto.fi/~lehtinj7/CS-C3100/2020/slides/16.2.diffuse.and.specular.reflectance.pdf>.

Lenovo 2024a, *What is video memory?*, Lenovo, viewed 12 September 2024, <https://www.lenovo.com/au/en/glossary/what-is-video-memory/>.

Lenovo 2024b, *What is Anti-Aliasing?*, Lenovo, viewed 17 October 2024, <https://www.lenovo.com/au/en/glossary/what-is-anti-aliasing/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsltid=AfmBOoqwD_m9Sn7G0udprehZeFtQAXeiZS_xl_O1RQU6fzFp6SxEp6kd>.

Leveau, P 2023, *Unstructured Data vs. Structured Data: differences and Impact on Your ML Pipeline*, Kili Technology, viewed 8 September 2024, <https://kili-technology.com/training-data/unstructured-data-vs-structured-data-know-the-difference-and-its-impact-on-your-ml-pipeline#unstructured-data--definition>.

Li, D 2023, *Sim2Real: Generating synthetic images from industry CAD models with domain randomization*.

Li, Y 2019, *Identification of Commercial UAV Used in Bomb Attacks Using Airframe Material Analysis and Electronic Component Restoration*, University of California, Davis.

Liang, D, Frederick, DA, Lledo, EE, Rosenfield, N, Berardi, V, Linstead, E & Maoz, U 2022, 'Examining the utility of nonlinear machine learning approaches versus linear regression for predicting body image outcomes: The US Body Project I', *Body Image*, vol. 41, pp. 32-45.

Liang, O 2023, *What's BNF, RTF and PNP? What're included in FPV drone kit and should I build my own drone?*, Oscar Liang, viewed 2 March 2024, <https://oscarliang.com/what-is-bnf/>.

Library of Congress 2020, *Wavefront OBJ File Format*, Library of Congress, viewed 8 October 2024, <https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml>.

Lightbown, D 2018, *Classic Tools Retrospective: Tim Sweeney on the first version of the Unreal Editor*, Game Developer, viewed 10 October 2024, <https://www.gamedeveloper.com/design/classic-tools-retrospective-tim-sweeney-on-the-first-version-of-the-unreal-editor>.

Lin, T, Guo, T & Aberer, K 2017, 'Hybrid neural networks for learning the trend in time series', *Proceedings of the twenty-sixth international joint conference on artificial intelligence*, pp. 2273-9.

LinkedIn 2024, *What are the benefits and drawbacks of using real-time rendering engines?*, LinkedIn, viewed 30 September 2024, <https://www.linkedin.com/advice/3/what-benefits-drawbacks-using-real-time-rendering-engines-9dygc>.

Liu, L, Ouyang, W, Wang, X, Fieguth, P, Chen, J, Liu, X & Pietikäinen, M 2020, 'Deep learning for generic object detection: A survey', *International journal of computer vision*, vol. 128, pp. 261-318.

Lobo, D, Patel, D, Morainvile, J, Shekhar, P & Abichandani, P 2021, 'Preparing students for drone careers using active learning instruction', *IEEE Access*, vol. 9, pp. 126216-30.

Logunova, I 2023a, *Hyperparameter Tuning and Evaluation of ML Models*, Serokell, viewed 30 October 2024, <https://serokell.io/blog/hyperparameter-tuning-in-ml>.

Logunova, I 2023b, *Backpropagation in Neural Networks*, Serokell, https://serokell.io/blog/understanding-backpropagation>.

Luksch, C, Wimmer, M & Schwärzler, M 2019, 'Incrementally baked global illumination', *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*, pp. 1-10.

Maghazei, O, Lewis, MA & Netland, TH 2022, 'Emerging technologies and the use case: A multi-year study of drone adoption', *Journal of Operations Management*, vol. 68, no. 6-7, pp. 560-91.

Mahesh, B 2020, 'Machine learning algorithms-a review', *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, no. 1, pp. 381-6.

Mandal, JK & Bhattacharya, D 2020, *Emerging technology in modelling and graphics*, vol. 937, Springer.

Mantiuk, R, Myszkowski, K & Seidel, H-P 2015, 'High Dynamic Range Imaging'.

Mardaljevic, J 2011, *Ambient Calculation: Crash Course*, De Montfort University, Leicester.

Martin, J 2017, *7 mistakes Blender users make when trying to render faster..* Blendergrid, https://blendergrid.com/learn/articles/7-mistakes-when-trying-to-render-faster#:~:text=One%20of%20the%20more%20well,direct%20view%20of%20the%20light.>.

Martínez-Díaz, V 2022, *Everything You Need to Know About FBX Files: A Comprehensive Guide*, Vection Technologies, viewed 8 October 2024, <https://vection-technologies.com/blog/Everything-You-Need-to-Know-About-FBX-Files-A-Comprehensive-Guide/>.

Mathew, A, Amudha, P & Sivakumari, S 2021, 'Deep learning techniques: an overview', *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*, pp. 599-608.

MathWorks 2024a, *Supervised Learning*, MathWorks, viewed 26 August 2024, <https://au.mathworks.com/discovery/supervised-learning.html>.

MathWorks 2024b, *What Is Computer Vision?*, MathWorks, https://au.mathworks.com/discovery/computer-vision.html#:~:text=Computer%20vision%20is%20a%20set,video%20tracking%2C%20and%20motion%20estimation.>.

Matthew Henderson 2020, 'Detection and Classification of Small UAS for Threat Neutralization', *Journal of the Defense Systems Information Analysis Center*, vol. 7, no. 3.

McGuire, M, Dorsey, J, Haines, E, Hughes, JF, Marschner, S, Pharr, M & Shirley, P 2020, 'A Taxonomy of Bidirectional Scattering Distribution Function Lobes for Rendering Engineers', *MAM@ EGSR*, pp. 25-8.

McLaughlin, EC, Sterling, J & Pozzebon, S 2018, *Venezuela says it has ID'd mastermind, accomplices in apparent Maduro assassination try*, Cable News Network, viewed 18 August 2024, <https://edition.cnn.com/2018/08/06/americas/venezuela-maduro-apparent-assassination-attempt/index.html>.

Microsoft 2021, *Direct3D 12 programming guide*, Microsoft, viewed 19 October 2024, <https://learn.microsoft.com/en-us/windows/win32/direct3d12/directx-12-programming-guide>.

Middle East Technical University 2007, *CEng 477 Illumination Models and Illumination Models and Surface-Rendering Methods*, Middle East Technical University, Ankara, <https://ocw.metu.edu.tr/pluginfile.php/1020/mod_resource/content/0/documents/lecturenotes_2007/week11_Illumination.pdf>.

Mitsuba 2024, *Choosing variants*, Mitsuba, viewed 1 October 2024, <https://mitsuba.readthedocs.io/en/latest/src/key_topics/variants.html>.

Modelo 2024, *Ray Tracing vs Path Tracing: Understanding the Differences*, Modelo, viewed 27 September 2024, <https://www.modelo.io/damf/article/2024/07/24/0132/ray-tracing-vs-path-tracing--understanding-the-differences?hl=en>.

Mohamed, AE 2017, 'Comparative study of four supervised machine learning techniques for classification', *International Journal of Applied*, vol. 7, no. 2, pp. 1-15.

Montesinos López, OA, Montesinos López, A & Crossa, J 2022, *Multivariate statistical machine learning methods for genomic prediction*, Springer Nature.

Moor Insights & Strategy 2017, *Physically-Based Rendering Revolutionizes Product Development*, Moor Insights & Strategy, Austin, Texas, https://images.nvidia.com/content/pdf/grid/whitepaper/PBR-Publish-2017.pdf>.

Morawski, I, Chen, Y-A, Lin, Y-S & Hsu, WH 2021, 'Nod: Taking a closer look at detection under extreme low-light conditions with night object detection dataset', *arXiv preprint arXiv:2110.10364*.

Moss, M 2015, *Stop "droning" on about power supply issues: find a solution*, A drone in action, Texas Instruments, <https://e2e.ti.com/blogs_/archives/b/fullycharged/posts/stop-droning-on-about-power-supply-issues-find-a-solution>.

Motion RC 2021, *ARF, PNP, RTF, RTR - What Does It All Mean?*, Motion RC, viewed 2 March 2024, <https://www.motionrc.com/blogs/motion-rc-blog/arf-bnf-pnp-rtf-what-does-it-all-mean>.

Movshovitz-Attias, Y, Kanade, T & Sheikh, Y 2016, 'How useful is photo-realistic rendering for visual learning?', *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, Springer, pp. 202-17.

Mower, N 2024, *Fixing 'Texture Streaming Pool Over Budget' in Unreal*, techarthub, viewed 26 October 2024, <https://techarthub.com/fixing-texture-streaming-pool-over-budget-in-unreal/#:~:text=Moreover%2C%20it%20is%20highly%20customizable,the%20quick%20and%20easy%20solution.>.

Mrabet, M, Sliti, M & Ammar, LB 2024, 'Machine Learning Algorithms Applied for Drone Detection and Classification: Benefits and Challenges', *Frontiers in Communications and Networks*, vol. 5, p. 1440727.

Mukherjee, R, Bessa, M, Melo-Pinto, P & Chalmers, A 2021, 'Object detection under challenging lighting conditions using high dynamic range imagery', *IEEE Access*, vol. 9, pp. 77771-83.

Najafabadi, MM, Villanustre, F, Khoshgoftaar, TM, Seliya, N, Wald, R & Muharemagic, E 2015, 'Deep learning applications and challenges in big data analytics', *Journal of big data*, vol. 2, pp. 1-21.

Nan, L, Du, S & Ibrahimli, N 2024, *GEO5017 Backpropagation*, Delft University of Technology, Delft, Netherlands.

Narang, R 2019, 'Armed suAs swarm: Big Threat of small uAs–C-suAs Development and Threat Mitigation by India', *Asian Defence Review*, pp. 75-100.

Naresh Boddeti, V, Kanade, T & Vijaya Kumar, BVK 2013, 'Correlation filters for object alignment', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2291-8.

Natarajan, SA & Madden, MG 2023, 'Hybrid synthetic data generation pipeline that outperforms real data', *Journal of Electronic Imaging*, vol. 32, no. 2, pp. 023011-.

Nayar, S, Ramamoorthi, R & Hanrahan, P n.d., *Basic Principles of Surface Reflectance*, Carnegie Mellon University, Pittsburgh, Pennsylvania, <https://www.cs.cmu.edu/afs/cs/academic/class/15462-f09/www/lec/lec8.pdf>.

Nichols, C 2024, *What Are Caustics and How to Render Them the Right Way*, Chaos, viewed 30 August 2024, <https://www.chaos.com/blog/what-are-caustics-and-how-to-render-them-the-right-way>.

Nichols, RK, Sincavage, S, Hans Mumm, WL, Candice Carter, JPH, Randall Mai, MJ, Mike Monnik, RM, Slofer, W & Harding, T 2022, 'DRONE DELIVERY OF CBNRECy – DEW WEAPONS Emerging Threats of Mini-Weapons of Mass Destruction and Disruption (WMDD)', in JP Hood (ed.), *Explosives Delivered by Drone*, NPP eBooks, Manhattan, Kansas, pp. 143 - 67.

Nikolenko, SI 2021, *Synthetic data for deep learning*, vol. 174, Springer.

Nowruzi, FE, Kapoor, P, Kolhatkar, D, Hassanat, FA, Laganiere, R & Rebut, J 2019, 'How much real data do we actually need: Analyzing object detection performance using synthetic and real data', *arXiv preprint arXiv:1907.07061*.

NVIDIA 2024a, *Ray Tracing*, NVIDIA, viewed 27 September 2024, <https://developer.nvidia.com/discover/ray-tracing#:~:text=Ray%20tracing%20is%20a%20rendering,%2C%20shadows%2C%20and%20indirect%20lighting.>.

NVIDIA 2024b, *Unreal Engine 5.4 Raytracing Guide*, NVIDIA, viewed 1 October 2024, <https://dlss.download.nvidia.com/uebinarypackages/Documentation/UE5+Raytracing+Guideline+v5.4.pdf>.

O'Mahony, N, Campbell, S, Carvalho, A, Harapanahalli, S, Hernandez, GV, Krpalkova, L, Riordan, D & Walsh, J 2020, 'Deep learning vs. traditional computer vision', *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1*, Springer, pp. 128-44.

OpticIntellect 2024, *Roboflow vs V7 Labs*, OpticIntellect, viewed 1 November 2024, <https://www.opticintellect.com/compare/roboflow-vs-v7-labs>.

PA Media 2019, *People behind drone chaos had 'detailed knowledge' of Gatwick*, The Guardian, viewed 13 August 2024, <https://www.theguardian.com/uk-news/2019/sep/27/gatwick-drone-disruption-perpetrators-detailed-knowledge-airport-police-report>.

Padilla, R, Netto, SL & Da Silva, EA 2020, 'A survey on performance metrics for object-detection algorithms', *2020 international conference on systems, signals and image processing (IWSSIP)*, IEEE, pp. 237-42.

Padilla, R, Passos, WL, Dias, TLB, Netto, SL & Silva, EABd 2021, 'A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit', *Electronics*.

Paperswithcode *Learning Rate Schedules*, Paperswithcode, viewed 29 October 2024, <https://paperswithcode.com/methods/category/learning-rate-schedules>.

Paperswithcode 2024, *KITTI*, Paperswithcode, viewed 27 August 2024, <https://paperswithcode.com/dataset/kitti>.

Park, S & Baek, N 2021, 'A shader-based ray tracing engine', *Applied Sciences*, vol. 11, no. 7, p. 3264.

Park, S, Kim, HT, Lee, S, Joo, H & Kim, H 2021, 'Survey on anti-drone systems: Components, designs, and challenges', *IEEE Access*, vol. 9, pp. 42635-59.

Pavic, V 2019, *Skydio 2: the self-flying future of drones starts at $999*, Image of Skydio 2 Flying, The Verge, <https://www.theverge.com/2019/10/1/20892377/skydio-2-drone-autonomous-self-flying-camera-controller-price-release-date-announcement>.

PC Market 2024, *DJI Air 2S*, PC Market, viewed 2 November, <https://www.pcmarket.com.au/dji-air-2s>.

Pearson, C & Ginsburg, B 2022, 'Deep learning', in WH Wen-Mei, et al. (eds), *Programming Massively Parallel Processors (Fourth Edition)*, 4 edn, Morgan Kaufmann, pp. 355-89.

Pedrycz, W & Chen, S-M 2020, 'Deep learning: Concepts and architectures'.

Peer Eric Molvar 2021, *Ray Tracing - Concepts*, AGi32, viewed 27 September 2024, <https://docs.agi32.com/AGi32/Content/ray_tracing/Ray_Tracing-Concepts.htm>.

Peng, J, Jury, EC, Dönnes, P & Ciurtin, C 2021, 'Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases: applications and challenges', *Frontiers in pharmacology*, vol. 12, p. 720694.

Pennington, A 2024, *The power of image-based lighting* Quasar Science, viewed 6 July 2024, <https://britishcinematographer.co.uk/supplement_posts/the-power-of-image-based-lighting/#:~:text=IBL%20is%20a%20VFX%20technique,its%20subtle%20nuances%2C%20is%20achieved.>.

Pereira, L, Carvalho, JRH, Krus, P, Klofsten, M & De Negri, VJ 2020, *Proceedings of IDEAS 2019: The Interdisciplinary Conference on Innovation, Design, Entrepreneurship, And Sustainable Systems*, vol. 198, Springer Nature.

Petru Potrimba 2023, *What is Hyperparameter Tuning? A Deep Dive.*, Roboflow, viewed 30 August 2024, <https://blog.roboflow.com/what-is-hyperparameter-tuning/>.

PixCap 2024, *Voxelization*, PixCap, viewed 12 September 2024, <https://pixcap.com/glossary/voxelization#:~:text=Voxelization%20is%20the%20process%20of%20converting%20a,of%20discrete%20volumetric%20elements%20known%20as%20voxels.>.

Plural Sight 2014, *Ambient Occlusion: What You Need to Know*, Plural Sight, viewed 21 October 2024, <https://www.pluralsight.com/resources/blog/software-development/understanding-ambient-occlusion>.

Pomerleau, M 2018, 'How $650 drones are creating problems in Iraq and Syria', *C4ISRNET-Media for the Intelligence Age Military*.

Poryagin, V 2024, *Temple of Sun*, Epic Games,, viewed 1 September 2024, <https://www.fab.com/listings/62690cf0-5e85-47bc-8a67-e977cf6ae108>.

Prakash, A, Boochoon, S, Brophy, M, Acuna, D, Cameracci, E, State, G, Shapira, O & Birchfield, S 2019, 'Structured domain randomization: Bridging the reality gap by context-aware synthetic data', *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 7249-55.

Preetham, AJ, Shirley, P & Smits, B 1999, 'A practical analytic model for daylight', *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 91-100.

Qi, PS, Tan Jia Hui, J, Sheng, TJ & Tamilchelvamani, J 2024, *Evaluation of Drone Detection Technologies*, Defence Science and Technology Agency, viewed 28 October 2024, <https://www.dsta.gov.sg/staticfile/ydsp/projects/files/reports/Evaluation_of_Drone_Detection_Technologies.pdf>.

Qin, Y, Brockett, A, Ma, Y, Razali, A, Zhao, J, Harrison, C, Pan, W, Dai, X & Loziak, D 2010, 'Micro-manufacturing: research, technology outcomes and development issues', *The International Journal of Advanced Manufacturing Technology*, vol. 47, pp. 821-37.

Radharamanan, R, Rosa, ACM, Neto, BD & Santos, VFB 2016, 'Use of 3D Printers to Design, Build, Test and Fly a Quadcopter Drone', *Journal of Management & Engineering Integration*, vol. 9, no. 1, pp. 24-33.

Ragnemalm, I 2024, *TSBK07 Ray-tracing*, Ingemar Ragnemalm, <https://computer-graphics.se/TSBK07/files/pdf20/_13f%20Ray-tracing.pdf>.

Rao, B, Gopi, AG & Maione, R 2016, 'The societal impact of commercial drones', *Technology in society*, vol. 45, pp. 83-90.

Rassler, D 2018, *The Islamic State and Drones: Supply, Scale, and Future Threats*, Combating Terrorism Center at West Point, United States Military Academy.

Reway, F, Hoffmann, A, Wachtel, D, Huber, W, Knoll, A & Ribeiro, E 2020, 'Test method for measuring the simulation-to-reality gap of camera-based object detection algorithms for autonomous driving', *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, pp. 1249-56.

Ridzuan, N, Ujang, U, Azri, S, Mohamad Yusoff, I & Choon, TL 2022, 'Voxelization techniques: data segmentation and data modelling for 3d building models', *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, pp. 149-55.

Riveros, JMR 2018, 'A metric evaluation framework for object detection', Masters thesis, Politecnico di Milano, Milano.

Robinson, M, Boyle, D & Murphy-Bates, S 2018, *FIRST VIDEO of the drone holding hundreds of thousands hostage: Gatwick saboteur plays cat-and-mouse with snipers as drone is spotted AGAIN, minutes before airport was due to re-open - with chaos set to last until Xmas Eve*, Daily Mail, viewed 13 August 2024, <https://www.dailymail.co.uk/news/article-6513923/Chaos-Gatwick-drone-spotted-near-airport-SHUTS-runway.html>.

Rogers, J & Kunertova, D 2022, *The Vulnerabilities of the Drone Age: Established Threats and Emerging Issues out to 2035*, NATO, Science for Peace and Security, https://css.ethz.ch/content/dam/ethz/special-interest/gess/cis/center-for-securities-studies/pdfs/NATO_VDA_Policy_Report.pdf>.

Rotenberg, S 2017, *CSE168 Rendering Algorithms*, University of California San Diego, San Diego, <https://cseweb.ucsd.edu/classes/sp17/cse168-a/CSE168_08_PathTracing.pdf>.

Ruder, S 2016, 'An overview of gradient descent optimization algorithms', *arXiv preprint arXiv:1609.04747*.

Rupard, J 2003, *Ray Tracing And Global Illumination*, University of North Florida Florida.

Sadeghi, F & Levine, S 2016, 'Cad2rl: Real single-image flight without a single real image', *arXiv preprint arXiv:1611.04201*.

Sala, Y 2021, 'Firmware Design and Implementation for a Quadrotor UAV', Politecnico di Torino.

Sammut, C & Webb, GI 2011, *Encyclopedia of machine learning*, Springer Science & Business Media.

Sanchez, S, Romero, H & Morales, A 2020, 'A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework', *IOP conference series: materials science and engineering*, IOP Publishing, p. 012024.

Sarker, IH 2021a, 'Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions', *SN computer science*, vol. 2, no. 6, p. 420.

Sarker, IH 2021b, 'Machine learning: Algorithms, real-world applications and research directions', *SN computer science*, vol. 2, no. 3, p. 160.

SAS Institute 2024, *Artificial Intelligence – What is AI & Why It Matters?*, SAS Institute, viewed 26 August 2024, <https://www.sas.com/en_au/insights/analytics/what-is-artificial-intelligence.html>.

Sazli, MH 2006, 'A brief review of feed-forward neural networks', *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01.

Schechter, S 2024, *Essential Guide to 3D File Formats*, 3D Cloud, viewed 8 October 2024, <https://3dcloud.com/3d-file-formats/>.

Scikit-Learn 2024, *Precision-Recall*, Scikit-Learn, viewed 14 September 2024, <https://scikit-learn.org/1.5/auto_examples/model_selection/plot_precision_recall.html>.

Sekhar, C & Meghana, PS 2020, 'A study on backpropagation in artificial neural networks', *Asia-Pacific Journal of Neural Networks and Its Applications*, vol. 4, no. 1, pp. 21-8.

Sellat, Q, Bisoy, SK & Priyadarshini, R 2022, 'Semantic segmentation for self-driving cars using deep learning: a survey', in S Mishra, et al. (eds), *Cognitive Big Data Intelligence with a Metaheuristic Approach*, 1 edn, Elsevier, Amsterdam, Netherlands.

Şen, O & Akarslan, H 2020, 'Terrorist Use of Unmanned Aerial Vehicles: Turkey's Example', *Defence Against Terrorism Review*, vol. 13.

SGVR Lab n.d., *Part II Physically-based Rendering*, https://sgvr.kaist.ac.kr/~sungeui/render/pbr/part2.pdf>.

Shah, D 2022, *Mean Average Precision (mAP) Explained: Everything You Need to Know*, V7 Labs, viewed 11 November 2024, <https://www.v7labs.com/blog/mean-average-precision>.

Shah, D 2024, *Intersection over Union (IoU): Definition, Calculation, Code*, V7 Labs, viewed 15 August 2024, <https://www.v7labs.com/blog/intersection-over-union-guide>.

Sheikholeslami, S 2019, *Ablation programming for machine learning*.

Shinsoj, I 2023, *Lighting: Unreal 5 Features Reference*, Medium, viewed 24 October 2024, <https://medium.com/@shinsoj/lighting-features-cheat-sheet-5b81b63b3ab7#:~:text=SSGI%20can%20be%20used%20in,%2Din%2Dunreal%2Dengine/>.

Shrivastava, A 2022, *COMP 642 Lecture 6: Deep Learning*, Rice University, Houston, Texas, <https://www.cs.rice.edu/~as143/COMP642_Spring22/Scribes/Lect6#:~:text=A%20hidden%20layer%20is%20just,more%20than%20two%20hidden%20layers.>.

Srihari, SN n.d., *Architecture Design for Deep Learning*, University at Buffalo, viewed 11 September 2024, <https://cedar.buffalo.edu/~srihari/CSE676/6.4%20ArchitectureDesign.pdf>.

Stanford University 1998a, *Types of Ray Tracing*, Stanford University, viewed 27 September 2024, <https://cs.stanford.edu/people/eroberts/courses/soco/projects/1997-98/ray-tracing/types.html#:~:text=To%20make%20ray%20tracing%20more,that%20point%20of%20the%20view plane.>.

Stanford University 1998b, *Ray Tracing Implementation*, Stanford University, viewed 27 September 2024, <https://cs.stanford.edu/people/eroberts/courses/soco/projects/1997-98/ray-tracing/implementation.html>.

Steen, D 2020, *Precision-Recall Curves*, Medium, viewed 14 September 2024, <https://medium.com/@douglaspsteen/precision-recall-curves-d32e5b290248>.

Subr, K 2008, *Sampling Strategies for Efficient Image Synthesis*, ProQuest.

Szeliski, R 2022, *Computer vision: algorithms and applications*, Springer Nature.

Taha, B & Shoufan, A 2019, 'Machine learning-based drone detection and classification: State-of-the-art in research', *IEEE Access*, vol. 7, pp. 138669-82.

Tapkir, A 2023, 'A Comprehensive Overview of Gradient Descent and its Optimization Algorithms', *International Advanced Research Journal in Science, Engineering and Technology*, vol. 10, no. 11, pp. 37 - 45.

Tatsat, H, Puri, S & Lookabaugh, B 2020, *Machine Learning and Data Science Blueprints for Finance*, O'Reilly media.

Tatzgern, W, Weinrauch, A, Stadlbauer, P, Mueller, JH, Winter, M & Steinberger, M 2024, *Radiance Caching with On-Surface Caches for Real-Time Global Illumination*, High-Performance Graphics, <https://www.highperformancegraphics.org/2025/index.html>.

Tel Aviv University n.d., *Illumination Models and Shading*, Tel Aviv University Tel Aviv, Israel <https://www.cs.tau.ac.il/~dcor/Graphics/cg-slides/shading04.pdf>.

Terven, J, Cordova-Esparza, DM, Ramirez-Pedraza, A & Chavez-Urbiola, EA 2023, 'Loss functions and metrics in deep learning. A review', *arXiv preprint arXiv:2307.02694*.

Tiigimägi, S 2024, *What is a Polygon Mesh and How to Edit It?*, 3D Studio, viewed 8 October 2024, <https://3dstudio.co/polygon-mesh/>.
Tin, D, Cheng, L, Shin, H, Hata, R, Granholm, F, Braitberg, G & Ciottone, G 2023, 'A descriptive analysis of the use of chemical, biological, radiological, and nuclear weapons by violent non-state actors and the modern-day environment of threat', *Prehospital and disaster medicine*, vol. 38, no. 3, pp. 395-400.
Tobin, J, Fong, R, Ray, A, Schneider, J, Zaremba, W & Abbeel, P 2017, 'Domain randomization for transferring deep neural networks from simulation to the real world', *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, pp. 23-30.
Torres, J 2024, *YOLOv8 Architecture: A Deep Dive into its Architecture*, Ultralytics, viewed 1 September 2024, <https://yolov8.org/yolov8-architecture/>.
Tran, R, Kanaujia, A & Parameswaran, V 2023, 'Fast Object Detection in High-Resolution Videos', *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1469-78.
Tremblay, J, Prakash, A, Acuna, D, Brophy, M, Jampani, V, Anil, C, To, T, Cameracci, E, Boochoon, S & Birchfield, S 2018, 'Training deep networks with synthetic data: Bridging the reality gap by domain randomization', *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 969-77.
Tunney, J 2023, *What is Chromatic Aberration? How to Deal with Color Fringing*, PetaPixel, viewed 25 October 2024, <https://petapixel.com/what-is-chromatic-aberration/#:~:text=Chromatic%20aberration%20is%20a%20color,on%20objects%20in%20your%20picture.>.
U.K. Civil Aviation Authority 2024, *Airspace restrictions for remotely piloted aircraft and drones*, U.K. Civil Aviation Authority, viewed 16 July 2024, <https://www.caa.co.uk/drones/airspace-and-restrictions/airspace-restrictions-for-remotely-piloted-aircraft-and-drones/>.
U.K. Department of Transport 2017, *Upgrading UK Airspace Strategic Rationale*, U.K. Department of Transport, London, https://assets.publishing.service.gov.uk/media/5a80e5bfed915d74e623111b/upgrading-uk-airspace-strategic-rationale.pdf>.
U.K. Department of Transport 2024, *The Drone and Model Aircraft Code*, CAP2320, U.K. Department of Transport, London, viewed 16 July 2024, <https://register-drones.caa.co.uk/drone-code/the_drone_code.pdf>.
U.S. Department of Defense 2009, *FY2009-2034 Unmanned Systems Integrated Roadmap*, ADA522247, Washington, D.C.
U.S. Department of Defense Joint C-sUAS Office 2020, *Counter-Small Unmanned Aircraft Systems Strategy*, AD1120305.
U.S. Joint Unmanned Aircraft Systems Center of Excellence 2010, *U.S. Army Unmanned Aircraft Systems Roadmap 2010-2035: Eyes of the Army*, ADA518437, Fort Novosel, Alabama.
Unity 2016a, *Shadows*, Unity, viewed 31 September 2024, <https://docs.unity3d.com/540/Documentation/Manual/Shadows.html>.
Unity 2016b, *Antialiasing*, Unity, viewed 15 October 2024, <https://docs.unity3d.com/540/Documentation/Manual/script-Antialiasing.html>.
Unity 2016c, *Global Illumination*, Unity, viewed 30 September, <https://docs.unity3d.com/540/Documentation/Manual/GIIntro.html>.
Unity 2017, *Baked lighting*, Unity, viewed 30 October 2024, <https://docs.unity3d.com/560/Documentation/Manual/LightMode-Baked.html>.
Unity 2019a, *Using precomputed lighting*, Unity, viewed 30 October 2024, <https://docs.unity3d.com/2018.3/Documentation/Manual/UsingPrecomputedLighting.html#:~:text=Baked%20GI%20is%20all%20precomputed,Mode%20setting%20to%20do%20this).>.

Unity 2019b, *Texture Streaming*, Unity, viewed 26 October 2024, <https://docs.unity3d.com/2019.1/Documentation/Manual/TextureStreaming.html>.

Unity 2024a, *Mipmaps*, Unity, viewed 25 October 2024, <https://docs.unity3d.com/6000.0/Documentation/Manual/texture-mipmaps-introduction.html>.

Unity 2024b, *Shadow mapping*, Unity, viewed 12 October 2024, <https://docs.unity3d.com/Manual/shadow-mapping.html>.

University of Texas at Austin n.d., *CS354 Basic Ray Tracing*, University of Texas at Austin, <https://www.cs.utexas.edu/~theshark/courses/cs354/lectures/cs354-4.pdf>.

UNSW Sydney n.d., *COMP3421 Introduction to 3D Graphics*, UNSW Sydney, <https://cgi.cse.unsw.edu.au/~cs3421/15s2/lectures/04_3DIntro.pdf>.

Vargas-Ramírez, N & Paneque-Gálvez, J 2019, 'The global emergence of community drones (2012–2017)', *Drones*, vol. 3, no. 4, p. 76.

Vasiou, E, Shkurko, K, Mallett, I, Brunvand, E & Yuksel, C 2018, 'A detailed study of ray tracing performance: render time and energy cost', *The Visual Computer*, vol. 34, pp. 875-85.

Verup, M & Olin, M 2016, 'Security models and exploitations in theory and practice for unmanned aerial vehicles', Master’s thesis, Technical University of Denmark.

VNTANA 2023, *Offline vs Real-Time Rendering*, VNTANA, viewed 23 June 2024, <https://www.vntana.com/blog/offline-vs-real-time-rendering/#:~:text=The%20benefits%20of%20real%2Dtime%20vs.&text=If%20all%20you%20need%20to,rendering%20would%20work%20perfectly%20well.>.

Vora, K & Yagnik, S 2014, 'A survey on backpropagation algorithms for feedforward neural networks', *International Journal of Engineering Development and Research*, vol. 1, no. 3, pp. 193-7--7.

Voulodimos, A, Doulamis, N, Doulamis, A & Protopapadakis, E 2018, 'Deep learning for computer vision: A brief review', *Computational intelligence and neuroscience*, vol. 2018, no. 1, p. 7068349.

Wang, S, Chung, F-L, Wang, J & Wu, J 2015, 'A fast learning method for feedforward neural networks', *Neurocomputing*, vol. 149, pp. 295-307.

Wani, MA, Bhat, FA, Afzal, S, Khan, AI, Wani, MA, Bhat, FA, Afzal, S & Khan, AI 2020, 'Basics of supervised deep learning', *Advances in Deep Learning*, pp. 13-29.

Washington University 2007, *Barycentric Coordinates with Applications in Mesh Deformation* Washington University, viewed 20 October 2024, <https://www.cs.wustl.edu/~taoju/research/bary.htm>.

Werbos, PJ 1990, 'Backpropagation through time: what it does and how to do it', *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-60.

Whitted, T 2005, 'An improved illumination model for shaded display', in *ACM Siggraph 2005 Courses*, pp. 4-es.

Whitted, T 2020, 'Origins of global illumination', *IEEE computer graphics and applications*, vol. 40, no. 1, pp. 20-7.

Witt, J 2024, *pers. comm.*, 23 October.

Wright, G & Jenzen-Jones, N 2018, 'Improvised, Craft-produced and Repurposed Munitions Deployed from UAVs in Recent Years', *Counter-IED Report*, vol. Spring/Summer 2018, pp. 53 - 64.

Wronski, B 2014, *Volumetric Fog: Unified compute shader based solution to atmospheric scattering*, 41st International Conference and Exhibition on Computer Graphics and Interactive Techniques, <https://bartwronski.com/wp-content/uploads/2014/08/bwronski_volumetric_fog_siggraph2014.pdf>.

Wüthrich, CA n.d., *Computer Graphics: 3-Local Illumination Models* Bauhaus University, Weimar, Weimar, <https://www.uni-weimar.de/fileadmin/user/fak/medien/professuren/Computer_Graphics/3-local-illu-16.pdf>.

Wythoff, BJ 1993, 'Backpropagation neural networks: a tutorial', *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 2, pp. 115-55.

Xiao, Y, Tian, Z, Yu, J, Zhang, Y, Liu, S, Du, S & Lan, X 2020, 'A review of object detection based on deep learning', *Multimedia Tools and Applications*, vol. 79, pp. 23729-91.

Yadav, A 2020, *Illumination Model*, University of Lucknow, Lucknow, <https://www.lkouniv.ac.in/site/writereaddata/siteContent/202003242118235253akanksha_yadav_Illuminationmodels.pdf>.

Yagawa, G & Oishi, A 2021, *Computational Mechanics with Neural Networks*, Springer.

Yang, H, Lee, Y, Jeon, S-Y & Lee, D 2017, 'Multi-rotor drone tutorial: systems, mechanics, control and state estimation', *Intelligent Service Robotics*, vol. 10, pp. 79-93.

Yen, N 2023, *#4. A Beginner's Guide to Gradient Descent in Machine Learning*, Medium, viewed 27 September 2024, <https://medium.com/@yennhi95zz/4-a-beginners-guide-to-gradient-descent-in-machine-learning-773ba7cd3dfe>.

Yu Xiang 2017, *Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild*, Stanford University Computational Vision and Geometry Lab, viewed 1 October 2024, <https://cvgl.stanford.edu/projects/pascal3d.html>.

Zhang, A, Smola, AJ, Lipton, Z & Li, M 2023, *Dive into Deep Learning*, 1 edn, Cambridge University Press, Cambridge, United Kingdom.

Zhang, X, Jia, N & Ivrissimtzis, I 2020, 'A study of the effect of the illumination model on the generation of synthetic training datasets', *arXiv preprint arXiv:2006.08819*.

Zhao, X, Wang, L, Zhang, Y, Han, X, Deveci, M & Parmar, M 2024, 'A review of convolutional neural networks in computer vision', *Artificial Intelligence Review*, vol. 57, no. 4, p. 99.

Zhao, Y 2024, 'Exploring Object Detection: Datasets, Metrics and Algorithms', *2nd International Conference on Artificial Intelligence, Database and Machine Learning (AIDML 2024)*, vol. 5, no. 1, pp. 556 - 61.

Zhao, Z-Q, Zheng, P, Xu, S-t & Wu, X 2019, 'Object detection with deep learning: A review', *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212-32.

Zheng, Y, Zheng, C, Shen, J, Liu, P & Zhao, S 2024, 'Keypoint-Guided Efficient Pose Estimation and Domain Adaptation for Micro Aerial Vehicles', *IEEE Transactions on Robotics*.

Zhu, H, Wei, H, Li, B, Yuan, X & Kehtarnavaz, N 2020, 'A review of video object detection: Datasets, metrics and methods', *Applied Sciences*, vol. 10, no. 21, p. 7834.

Ziemba, L 2019, *Observe and Report: Considerations for Evaluating Drone Detection Systems*, Security Industry Association, viewed 27 October 2024, <https://www.securityindustry.org/2019/09/06/observe-and-report-considerations-for-evaluating-drone-detection-systems/>.

Zitar, RA, Al-Betar, M, Ryalat, M & Kassaymeh, S 2023, 'A review of UAV visual detection and tracking methods', *arXiv preprint arXiv:2306.05089*.

Zou, L 2022, *Meta-learning: Theory, algorithms and applications*, Elsevier.

Appendix A – Project Specification

ENP4111 Professional Engineer Research Project

For:	Aneesh Vasudevan
Degree:	Bachelor of Engineering (Honours)
Major:	Mechatronic & Robotic Engineering

Title:	Can High-Fidelity Photorealism Provided by Unreal Engine 5 Bridge the Synthetic-to-Reality Gap for Improved Multi-Rotor Drone Detection?
Supervisor:	Dr. Tobias Lowe
Project Aim:	To investigate the performance impact of high-fidelity photorealism provided by UE5 *Lumen* global illumination and reflections, as well as ray traced shadows in synthetically generated training and validation data (Pure Synthetic Data), on CV models for the application of multi-rotor sUAS detection in the real-world.
Subject:	ENP4111 – Professional Engineer Research Project

1. Collect real-world testing image datasets of multi-rotor sUAS under different lighting conditions with a sufficient distribution, that can be used in the assessment of CV models trained with synthetically generated data.

2. Generate and collect synthetic training images of multi-rotor sUAS under similar lighting conditions in UE5. For each of these lighting conditions, separate datasets will be generated with varying levels of *Lumen* global illumination and reflections, as well as ray traced shadow support.

3. Evaluate and compare the performance of CV models trained with synthetic datasets employing varying levels of *Lumen* global illumination and reflections, as well as ray traced shadow support, respectively for each lighting condition.

Appendix B – Image Attribution for Images Used for Testing and Applied Modifications

This section provides an image attribution of all images sourced from the internet and used for the testing datasets and details the following:

- Modifications applied to these images and respective copyright licenses that these images are used under.

- Images acquired using a paid subscription are indicated with the following colours:

  - ➢ Image files highlighted in light green 🟩 identify images sourced from the dataset *Visiodect Dataset: An Aerial Dataset for Scenario-Based Multi-Drone Detection And Identification*, which was accessible through an individual subscription (monthly) with Institute of Electrical and Electronics Engineers (IEEE) DataPort, which is a repository for research data.

    Citation:

    Simeon Okechukwu Ajakwe, Vivian Ukamaka Ihekoronye, Golam Mohtasin, Rubina Akter, Ali Aouto, Dong Seong Kim, Jae Min Lee, 2022. VisioDECT Dataset: An Aerial Dataset for Scenario-Based Multi-Drone Detection and Identification. Available at: https://dx.doi.org/10.21227/n27q-7e06.

    Image files highlighted in light orange 🟧 identify images sourced using a *Freepik* premium subscription (monthly).

- All other unhighlighted image files identify images sourced for free.

Image Attribution of Multi-Rotor sUAS Images with Day Light / No Clouds / Level 0 Background Features

Day Light / No Clouds / Level 0 Background Features			
File Name	**Original Image**	**Modification Details**	**Copyright Licence**
1_0.jpg	Bradford, C 2019, *flying drone during day*, digital image of a multi-rotor sUAS flying in clear blue sky , Unsplash, viewed 2 July 2024, < https://unsplash.com/photos/flying-drone-during-day-Y0LsduX0aTY>		Unsplash License
2_0.jpg	Denunzio, A 2023, *A photo of a DJI small drone flying in the blue skies of Tampa Bay*, digital image of a multi-rotor sUAS in clear blue sky, Unsplash, viewed 1 July 2024,< https://unsplash.com/photos/a-black-and-white-photo-of-a-black-and-white-plane-flying-in-the-sky-nqNqkiKQeGM>		Unsplash License
3_0.jpg	Elsemargriet 2020, *Drone, Fly, Technology image*, digital image of a multi-rotor sUAS flying in clear blue sky, Pixabay, viewed 30 June 2024, < https://pixabay.com/photos/drone-fly-technology-propeller-5099737/>		Pixabay Content License

4_0.jpg	Müller, C 2017, *MavicProPlatinum-2*, digital image of a multi-rotor sUAS flying in clear blue sky, Wikimedia Commons, viewed 30 June 2024, < https://commons.wikimedia.org/wiki/File:MavicProPlatinum-2.jpg>		Creative Commons Attribution - ShareAlike 4.0 International (CC BY-SA 4.0)
5_0.jpg	Iswoar 2021, *A drone DJI Mavic 2 Pro flying*, digital image of a multi-rotor sUAS flying in clear blue sky, Wikimedia Commons, viewed 30 June 2024, < https://commons.wikimedia.org/wiki/File:A_drone_DJI_Mavic_2_Pro_flying.jpg>		Creative Commons Attribution - ShareAlike 4.0 International (CC BY-SA 4.0)
6_0.jpg	Holobar M 2018, *Flying Drone*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 1 July 2024 < https://www.pexels.com/photo/flying-drone-1757697/>		Pexels License
7_0.jpg	Screen Post 2021, *A Drone Camera Across the Blue Sky*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 2 July 2024, < https://www.pexels.com/photo/a-drone-camera-across-the-blue-sky-7582145/>		Pexels License
8_0.jpg	McCullough, D 2013, *Drone and Moon*, digital image of a multi-rotor sUAS flying in clear blue sky with the moon in the background, Flickr, viewed 30 June 2024, < https://www.flickr.com/photos/69214385@N04/8725078749>	Image edited to remove moon in background.	Creative Commons Attribution 2.0 Generic (CC BY 2.0)
9_0.jpg	Ajairapara 2018, *Drone of Bangladesh police*, digital image of a multi-rotor sUAS flying in clear blue sky, Wikimedia Commons, viewed 30 June 2024, < https://commons.wikimedia.org/wiki/File:Drone_of_Bangladesh_police_.jpg>		Creative Commons Attribution - ShareAlike 4.0

			International (CC BY-SA 4.0)
10_0.jpg	Suraj, A 2022, *No Image Title*, digital image of a multi-rotor sUAS flying in overexposed clear blue sky, Unsplash, viewed 1 July 2024 < https://unsplash.com/photos/a-small-propeller-plane-flying-through-a-blue-sky-jUQC9GR-hwQ>		Unsplash License
11_0.jpg	Gnos, I 2017, *Mavic pro on YuLong mountain*, digital image of a multi-rotor sUAS flying in overexposed clear blue sky, Unsplash, viewed 30 June 2024, < https://unsplash.com/photos/white-quadcopter-drone-flying-near-snow-mountain-during-daytime-ZlkRrzJl20Q>	Image cropped to remove mountains.	Unsplash License
12_0.jpg	Cebeci, Z 2013, *Quadcopter 03*, digital image of a multi-rotor sUAS flying in clear blue sky, Wikimedia Commons, viewed 30 June 2024, < https://commons.wikimedia.org/wiki/File:Quadcopter_03.JPG>	Image cropped to remove trees.	Creative Commons Attribution-Share Alike 3.0 Unported (CC BY-SA 3.0)
13_0.jpg	Kandark, W, *Close-up of a Drone Flying under Clear, Blue Sky*, Pexels, viewed 6 July 2024, < https://www.pexels.com/photo/close-up-of-a-drone-flying-under-clear-blue-sky-19806135/>		Pexels License
14_0.jpg	LN_Photoart 2018, *Drone, Flying drone, Quadrocopter image*, digital image of a multi-rotor sUAS flying in clear blue sky, Pixabay , viewed 2 July 2024, <https://pixabay.com/photos/drone-flying-drone-quadrocopter-3198323/		Pixabay Content License
15_0.jpg	RGY23 2018, *Drone, Fly, Camera image*, digital image of a multi-rotor sUAS flying in clear blue sky, Pixabay , viewed 2 July 2024, < https://pixabay.com/photos/drone-fly-camera-game-sky-3845299/>		Pixabay Content License
16_0.jpg	The Lazy Artist Gallery, *Man Holding Drone Remote Standing on Gray Concrete Pavement*, Pexels, viewed 6 July 2024,< https://www.pexels.com/photo/man-holding-drone-remote-standing-on-gray-concrete-pavement-1451283/>	Image cropped to remove person and desert.	Pexels License
17_0.jpg	The Lazy Artist Gallery 2018, *Woman Standing Near Basketball Hoop*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 5 July 2024, < https://www.pexels.com/photo/woman-standing-near-basketball-hoop-1406510/>	Image cropped to remove person and	Pexels License

		basketball court.	
18_0.jpg	Morales, JC 2017, *Drone - DJI Maverick*, digital image of a multi-rotor sUAS flying in clear blue sky, Flickr, , viewed 2 July 2024, < https://www.flickr.com/photos/e-lexia/37650657696/in/album-72157637358097006/>		Creative Commons Attribution - Noncommercial 2.0 Generic (CC BY-NC 2.0)
19_0.jpg	Gerber, K 2020, *Drone Flying in the Sky*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 30 June 2024, < https://www.pexels.com/photo/drone-flying-in-the-sky-4058939/>		Pexels License
20_0.jpg	Carrozzo, C 2020, *DJI Mavic Air 2 hovering*, digital image of a multi-rotor sUAS flying in clear blue sky, Unsplash, viewed 1 July 2024, < https://unsplash.com/photos/black-and-gray-drone-in-mid-air-ZnKnQgmm0Eg>		Unsplash License
21_0.jpg	Morales, JC 2018, *Drone DJI*, digital image of a multi-rotor sUAS flying in clear blue sky, Flickr, , viewed 2 July 2024, < https://www.flickr.com/photos/e-lexia/41234478262/in/album-72157637358097006/>		Creative Commons Attribution - Noncommercial-Noderivs 2.0 Generic (CC BY-NC-ND 2.0)
22_0.jpg	Freitas, P 2023, *Drone Flying under Clear Sky*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, , viewed 1 July 2024, < https://www.pexels.com/photo/drone-flying-under-clear-sky-17146818/>		Pexels License
23_0.jpg	Gowtham Poppy 2021, *An Airborne Drone Machine*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 1 July 2024, < https://www.pexels.com/photo/an-airborne-drone-machine-7890868/>		Pexels License

24_0.jpg	Sayles, B 2019, *Black Drone Flying Above Body of Water*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 5 July 2024, <https://www.pexels.com/photo/black-drone-flying-above-body-of-water-2314657/>	Image cropped to remove mountains.	Pexels License
25_0.jpg	Ajakwe, SO 2022, *Dji pantom Sunny (35)*, digital image of a multi-rotor sUAS flying in clear blue sky, IEEE DataPort, viewed 30 June 2024, <https://dx.doi.org/10.21227/n27q-7e06>	Image cropped to remove buildings and trees.	Creative Commons Attribution 4.0 International (CC BY 4.0)
26_0.jpg	Geometric Photography 2022, *Low-Angle Shot of a White Drone Camera Flying in the Blue Sky*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 1 July 2024, < https://www.pexels.com/photo/low-angle-shot-of-a-white-drone-camera-flying-in-the-blue-sky-12975479/>		Pexels License
27_0.jpg	Soklič, N 2021, *A Drone Flying Under a Clear Blue Sky*, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 1 July 2024,< https://www.pexels.com/photo/a-drone-flying-under-a-clear-blue-sky-6655162/>		Pexels License
28_0.jpg	Freitas, P 2023, No Image Title, digital image of a multi-rotor sUAS flying in clear blue sky, Pexels, viewed 1 July 2024, < https://www.pexels.com/photo/drone-drone-flying-drone-footage-16923757/>	Image cropped to remove trees.	Pexels License
29_0.jpg	Massaro, M 2017, *"Drone" ad Avezzano*, digital image of a multi-rotor sUAS flying in clear blue sky, Wikimedia Commons, viewed 3 July 2024,< https://commons.wikimedia.org/wiki/File:%22Drone%22_ad_Avezzano.jpg>	Image cropped to remove buildings.	Creative Commons Attribution - ShareAlike 4.0 International (CC BY-SA 4.0)
30_0.jpg	Alexman89 2023, *Drone, Technology, Equipment image*, digital image of a multi-rotor sUAS flying in clear blue sky, Pixabay, viewed 2 July 2024, < https://pixabay.com/photos/drone-technology-equipment-8188144/>		Pixabay Content License

Image Attribution of Multi-Rotor sUAS Images with Low-Light Sky / No Clouds / Level 1 Background Features

Low-Light Sky / No Clouds / Level 1 Background Features			
File Name	Original Image	Modification Details	Copyright Licence
1_LL.jpg	Royal, T 2024, *a small plane flying through a cloudy sky*, digital image of a multi-rotor sUAS flying in low light sky, Unsplash, viewed 6 July 2024, < https://unsplash.com/photos/a-small-plane-flying-through-a-cloudy-sky-VsMyIYWfgAk>		Unsplash License
2_LL.jpg	Pressl, J 2020, *black drone flying over green trees during daytime*, digital image of a multi-rotor sUAS flying in low light sky, Unsplash, viewed 6 July 2024, <https://unsplash.com/photos/black-drone-flying-over-green-trees-during-daytime-1m5qX-gTINk>	Image cropped to remove trees and water body.	Unsplash License
3_LL.jpg	Mavrommatis, J 2017, *white quad-copter drone during golden hour*, digital image of a multi-rotor sUAS flying in low light sky, Unsplash, viewed 6 July 2024, < https://unsplash.com/photos/white-quad-copter-drone-during-golden-hour-IBLo7-N4z94>	Image cropped to remove landscape.	Unsplash License
4_LL.jpg	Ericearles 2019, *Drone, Sunset, Sky image*, digital image of a multi-rotor sUAS flying in low light sky, Pixabay, viewed 6 July 2024, < https://pixabay.com/photos/drone-sunset-sky-quadrocopter-3975568/	Image cropped to remove houses, trees and water body.	Pixabay Content License
5_LL.jpg	StockSnap 2017, *Drone, Camera, Ice image*, digital image of a multi-rotor sUAS flying in low light sky, Pixabay, viewed 6 July 2024, <https://pixabay.com/photos/drone-camera-ice-iceberg-snow-2591566/>	Image cropped to remove landscape.	Pixabay Content License
6_LL.jpg	Kgbo 2020, *Drone over Lake Cooroibah, Queensland, 2020*, digital image of a multi-rotor sUAS flying in low light sky, Wikimedia Commons, viewed 6 July 2024, < https://commons.wikimedia.org/wiki/File:Drone_over_Lake_Cooroibah,_Queensland,_2020.jpg>	Image cropped to remove vegetation and distance black artifact.	Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)
7_LL.jpg	Yaroslavsky, M 2021, *Black Drone Flying Under the Cloudy Sky*, digital image of a multi-rotor sUAS flying in low light sky, Flickr , Pexels, viewed 30 June 2024, < https://www.pexels.com/photo/black-drone-flying-under-the-cloudy-sky-8637484/>		Pexels License

8_ LL.jpg	Vieriu, C 2021, No Image Title, digital image of a multi-rotor sUAS flying in low light sky, Unsplash, viewed 2 July 2024, < https://unsplash.com/photos/black-and-brown-drone-flying-in-the-sky-e8XCRTkHT_k>		Unsplash License
9_ LL.jpg	Torres, I 2023, *Flying Drone over City at Sunrise*, digital image of a multi-rotor sUAS flying in low light sky, Pexels, viewed 30 June 2024, < https://www.pexels.com/photo/flying-drone-over-city-at-sunrise-16110311/>	Image cropped to remove buildings.	Pexels License
10_ LL.jpg	Cormack, J 2019, *black quadcopter in sky*, digital image of a multi-rotor sUAS flying in low light sky , Pexels, viewed 30 June 2024, < https://unsplash.com/photos/black-quadcopter-in-sky-DwMuY7PFPg0>	Image cropped to remove water body.	Pexels License
11_ LL.jpg	Sawant, S 2020, *A Drone Flying during Sunset*, digital image of a multi-rotor sUAS flying in low light sky, Pexels, viewed 30 June 2024,< https://www.pexels.com/photo/a-drone-flying-during-sunset-7307966/>	Image cropped to remove ground in background.	Pexels License
12_ LL.jpg	Ali, M 2018, *White Quad-copter Drone Flying over Body of Water during Golden Hour*, digital image of a multi-rotor sUAS flying in low light sky, Pexels, viewed 30 June 2024, < https://www.pexels.com/photo/white-quad-copter-drone-flying-over-body-of-water-during-golden-hour-1590242/>	Image cropped to remove water body and boats.	Pexels License
13_ LL.jpg	Dunnok_D 2015, *Toy drone flying above Castle Sands*, digital image of a multi-rotor sUAS flying in low light sky, Flickr, viewed 3 July 2024, < https://www.flickr.com/photos/dunnock_d/21897816175/>		Creative Commons Attribution-Noncommercial 2.0 Generic (CC BY-NC 2.0)
14_ LL.jpg	Sorasak 2017, silhouette camera drone flying midair, digital image of a multi-rotor sUAS flying in low light sky, Unsplash, viewed 6 July 2024, <https://unsplash.com/photos/silhouette-camera-drone-flying-midair-jlLC2hAQ7uQ>	Image cropped to remove landscape.	Unsplash License
15_ LL.jpg	Colah, Z 2024, *a small plane flying over a lush green hillside*, digital image of a multi-rotor sUAS flying in low light sky, Unsplash, viewed 2 July 2024 < https://unsplash.com/photos/a-small-plane-flying-over-a-lush-green-hillside-T8nGldduE9g>	Image cropped to remove landscape.	Unsplash License

16_LL.jpg	The Lazy Artist Gallery 2018, *Woman Holding Remote of Drone*, digital image of a multi-rotor sUAS flying in low light sky, Pexels, viewed 30 June 2024, < https://www.pexels.com/photo/woman-holding-remote-of-drone-1170064/>	Image cropped to remove person and trees.	Pexels License
17_LL.jpg	Pexels 2016, *Dawn, Drone, Dusk image*, digital image of a multi-rotor sUAS flying in low light sky, Pixabay, viewed 5 July 2024, <https://pixabay.com/photos/dawn-drone-dusk-mountains-outdoors-1868870/>	Image cropped to remove landscape.	Pixabay Content License
18_LL.jpg	Clarke, G 2015, *Sunset Drone*, digital image of a multi-rotor sUAS flying in low light sky, Wikimedia Commons, viewed 5 July 2024, < https://commons.wikimedia.org/wiki/File:Sunset_Drone.png>		Creative Commons Attribution 2.0 Generic (CC BY 2.0)
19_LL.jpg	Pressl, J 2021, No Image Title, digital image of a multi-rotor sUAS flying in low light sky, Unsplash, viewed 2 July 2024, <https://unsplash.com/photos/black-drone-flying-in-the-sky-gs_essafm9c>		Unsplash License
20_LL.jpg	Xray40000 2015, *DJI Phantom Quadrokopter im Sauerland - Flickr - Xray40000*, digital image of a multi-rotor sUAS flying in low light sky, Wikimedia Commons, viewed 5 July 2024, < https://commons.wikimedia.org/wiki/File:DJI_Phantom_Quadrokopter_im_Sauerland_-_Flickr_-_Xray40000.jpg>	Image cropped to remove landscape.	Creative Commons Attribution 2.0 Generic (CC BY 2.0)
21 LL.jpg	EyeEm n.d, *A beautiful minimalist scenery of mountain sunset in purple*, A beautiful minimalist scenery of mountain sunset in purple tones, Freepik, viewed 5 July 2024, <https://www.freepik.com/premium-photo/beautiful-minimalist-scenery-mountain-sunset-purple-tones_126292672.htm#fromView=search&page=44&position=8&uuid=70aeea5a-79ba-4816-b631-caf88cb4cfa9>	Image cropped to remove landscape.	Freepik License
22_LL.jpg	Baxter, W 2017, *A drone over Maidens Beach - geograph.org.uk – 5526032*, digital image of a multi-rotor sUAS flying in low light sky, Wikimedia Commons, viewed 5 July 2024, < https://commons.wikimedia.org/wiki/File:A_drone_over_Maidens_Beach_-_geograph.org.uk_-_5526032.jpg>		Creative Commons Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0)
23 LL.jpg	EyeEm n.d, *Low angle view of silhouette helicopter against sky*, digital image of a multi-rotor sUAS flying in low light sky, Freepik, viewed 5 July 2024, < https://www.freepik.com/premium-photo/low-angle-view-silhouette-helicopter-against-		Freepik License

	sky_101475371.htm#fromView=search&page=45&position=36&uuid=70aeea5a-79ba-4816-b631-caf88cb4cfa9>		
24 LL.jpg	EyeEm n.d, *Low angle view of airplane flying against sky during sunset*, digital image of a multi-rotor sUAS flying in low light sky, Freepik, viewed 5 July 2024, < https://www.freepik.com/premium-photo/low-angle-view-airplane-flying-against-sky-sunset_125073331.htm#fromView=search&page=43&position=15&uuid=70aeea5a-79ba-4816-b631-caf88cb4cfa9>		Freepik License
25_LL.jpg	EyeEm n.d, *Low angle view of airplane flying against sky during sunset*, digital image of a multi-rotor sUAS flying in low light sky, Freepik, viewed 5 July 2024,< https://www.freepik.com/premium-photo/low-angle-view-airplane-flying-against-sky-sunset_116310890.htm#fromView=search&page=30&position=46&uuid=70aeea5a-79ba-4816-b631-caf88cb4cfa9>	Image cropped to remove landscape.	Freepik License
26_LL.jpg	Bannafarsai n.d, *Drone on sunset time*, digital image of a multi-rotor sUAS flying in low light sky, Freepik, viewed 5 July 2024, <https://www.freepik.com/premium-photo/drone-sunset-time_3171125.htm#fromView=search&page=18&position=38&uuid=4cb9f596-03d8-4a4d-8240-3879272c4550>		Freepik License
27_LL.jpg	StockSnap 2017, *Drone, Plants, Trees image*, digital image of a multi-rotor sUAS flying in low light sky, Pixabay, viewed 8 July 2024, < https://pixabay.com/photos/drone-plants-trees-camera-water-2588156/>	Image cropped to remove city.	Pixabay Content License
28_LL.jpg	Florent P 2021, *Drone, Camera drone, Aircraft image*, digital image of a multi-rotor sUAS flying in low light sky, Pixabay, viewed 8 July 2024,< https://pixabay.com/photos/drone-camera-drone-aircraft-device-6586331/>		Pixabay Content License
29 LL.jpg	k_samurkas n.d, *Flying drone with blue sky*, digital image of a multi-rotor sUAS flying in low light sky, Freepik, viewed 8 July 2024, <https://www.freepik.com/premium-photo/flying-drone-with-blue-sky_10125451.htm#fromView=search&page=43&position=46&uuid=70aeea5a-79ba-4816-b631-caf88cb4cfa9>		Freepik License
30 LL.jpg	hiv_360 n.d, *Silhouette of woman controlling drone on sunset background*, digital image of a multi-rotor sUAS flying in low light sky, Freepik, viewed 8 July 2024, <https://www.freepik.com/premium-photo/silhouette-woman-controlling-drone-sunset-background_65073569.htm#fromView=search&page=2&position=8&uuid=b9a334f2-27c8-4ba4-b8d4-cb71806656ac>	Image cropped to obtain a single multi-rotor sUAS (right) and remove person and vegetation.	Freepik License

Image Attribution of Multi-Rotor sUAS with Day Light / No Clouds / Level 2 Background Features

Day Light / No Clouds / Level 2 Background Features			
File Name	Original Image	Modification Details	Copyright Licence
1_0_sb.jpg	Wirestock n.d., *White drone flying over a parched desert area against a blue sky*, digital image of a multi-rotor sUAS flying in clear sky with ground-based background features, Freepik, viewed 21 October 2024, < https://www.freepik.com/free-photo/white-drone-flying-parched-desert-area-against-blue-sky_17244078.htm#fromView=search&page=2&position=37&uuid=dbaf4764-71c5-4bee-9dea-6736a9f2e5b7>		Freepik License
2_0_sb.jpg	Freepik n.d, *Side view hand and drone outdoors*, digital image of a multi-rotor sUAS flying in clear sky over water body, Freepik, viewed 21 October 2024, < https://www.freepik.com/free-photo/side-view-hand-drone-outdoors_25965502.htm#fromView=search&page=2&position=1&uuid=d6471b9e-5fde-4b7f-97db-72a555b93156>		Freepik License
3_0_sb.jpg	Evelyn Arleth n.d, *A Black Drone Flying*, digital image of a multi-rotor sUAS flying in clear sky with ground-based background features, Freepik, viewed 21 October 2024, < https://www.pexels.com/photo/a-black-drone-flying-7028561/>		Pexels License
4_0_sb.jpg	Yuhan Du 2024, *Shot of DJI Mini 2 drone with sunny snow mountain.*, digital image of a multi-rotor sUAS flying in clear blue sky with ground-based background features, Unsplash, viewed 21 October 2024, < https://unsplash.com/photos/an-airplane-is-flying-over-a-mountain-range-meqyAhrisWQ>		Unsplash License
5_0_sb.jpg	Wirestock n.d., *Closeup shot of drone over a beautiful mountainous landscape covered with snow*, digital image of a multi-rotor sUAS flying in clear sky with ground-based background features, Freepik, viewed 21 October 2024, <https://www.freepik.com/free-photo/closeup-shot-drone-beautiful-mountainous-landscape-covered-with-snow_14376631.htm#fromView=search&page=2&position=16&uuid=406c7306-c1f7-4da4-9252-b9faf6f622ac>		Freepik License
6_0_sb.jpg	Freepik n.d., *Medium shot woman with drone outdoors*, digital image of a multi-rotor sUAS flying in clear sky with person, water body and mountainous terrain in background out of focus, Freepik, viewed 21 October 2024, < https://www.freepik.com/free-photo/medium-shot-woman-with-drone-outdoors_25965526.htm>		Freepik License
7_0_sb.jpg	EyeEm n.d., *Low angle view of airplane flying in sky*, digital image of a multi-rotor sUAS flying in clear sky over clouds and landscape, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/low-angle-view-airplane-flying-		Freepik License

	sky_103047813.htm#fromView=author&page=7&position=44&uuid=3380e13d-e8c9-4cca-b438-75f11e61439e>		
8_0_sb.jpg	Souza, N n.d., *A Drone Camera Flying*, digital image of a multi-rotor sUAS flying in clear sky over water body, Pexels, viewed 21 October 2024, < https://www.pexels.com/photo/a-drone-camera-flying-8459530/>		Pexels License
9_0_sb.jpg	Beard, O 2021, *a large black flying over a lush green field*, digital image of a multi-rotor sUAS in clear sky with urban area in background out of focus, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/a-large-black-flying-over-a-lush-green-field-D1xAHu0ZJ8E>		Unsplash License
10_0_sb.jpg	NeuroSky n.d., *Drone flying over a grassy field*, digital image of a multi-rotor sUAS flying over grass field with trees in background, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/drone-flying-grassy-field_339904995.htm#fromView=search&page=35&position=32&uuid=62cd3870-9c65-4ac9-b1fa-5581be8376fb>		Freepik License
11_0_sb.jpg	The Lazy Artist Gallery n.d, *Woman Holding Remote of Drone*, digital image of a multi-rotor sUAS flying in clear sky with person in foreground and trees in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/woman-holding-remote-of-drone-1170064/>		Pexels License
12_0_sb.jpg	The Lazy Artist Gallery n.d, *Woman Playing Drone*, digital image of a multi-rotor sUAS flying in clear sky over water body with person in foreground and fence structure in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/woman-playing-drone-1170344/>		Pexels License
13_0_sb.jpg	The Lazy Artist Gallery n.d, *Man Controlling Quadcopter Drone*, digital image of a multi-rotor sUAS flying in clear sky with person to the side, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/woman-playing-drone-1170344/>		Pexels License
14_0_sb.jpg	Sayles, B n.d., *Black Drone*, digital image of a multi-rotor sUAS flying in clear sky with mountains and water body in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/black-drone-2505972/>		Pexels License
15_0_sb.jpg	Oleksandr P n.d., *Selective Focus Photograph of White Quadcopter Drone during Blue Hour*, digital image of a multi-rotor sUAS flying in clear sky with structures in background, Pexels, viewed 21 October 2024,<https://www.pexels.com/photo/selective-focus-photograph-of-white-quadcopter-drone-during-blue-hour-319968/>		Pexels License
16_0_sb.jpg	Solargalaxy n.d., *Flying drone in modern city*, digital image of a multi-rotor sUAS flying in clear sky with city background, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/flying-drone-modern-		Freepik License

	city_92564988.htm#fromView=search&page=2&position=18&uuid=3ea0228e-7b77-4719-82b2-ec43b7c9c23f>		
17_0_sb.jpg	EyeEm n.d., *Drone flying over a wheat field*, digital image of a multi-rotor sUAS flying in clear sky with field below, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/drone-flying-wheat-field_125636728.htm#fromView=search&page=8&position=12&uuid=f865d081-8296-48dd-b789-531436b3272a>		Freepik License
18_0_sb.jpg	EyeEm n.d., *Low angle view of camera against clear blue sky*, digital image of a multi-rotor sUAS flying in clear sky with landscape below, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/low-angle-view-camera-against-clear-blue-sky_102568809.htm#fromView=search&page=12&position=14&uuid=f865d081-8296-48dd-b789-531436b3272a>		Freepik License
19_0_sb.jpg	EyeEm n.d., *Low angle view of airplane flying against clear blue sky*, digital image of a multi-rotor sUAS flying in clear sky with landscape below, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/low-angle-view-airplane-flying-against-clear-blue-sky_115059174.htm#fromView=search&page=13&position=14&uuid=f865d081-8296-48dd-b789-531436b3272a>		Freepik License
20_0_sb.jpg	EyeEm n.d., *Drone flying over sea against clear sky*, digital image of a multi-rotor sUAS flying in clear sky with landscape below, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/drone-flying-sea-against-clear-sky_102895963.htm#fromView=author&page=4&position=4&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
21_0_sb.jpg	EyeEm n.d., *Low angle view of airplane flying against clear sky*, digital image of a multi-rotor sUAS flying in clear sky with trees below, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/low-angle-view-airplane-flying-against-clear-sky_97862165.htm#fromView=author&page=7&position=16&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
22_0_sb.jpg	EyeEm n.d., *Quadcopter drone hovering over a wheat field*, digital image of a multi-rotor sUAS flying in clear sky with field below, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/quadcopter-drone-hovering-wheat-field_133064314.htm#fromView=author&page=7&position=40&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
23_0_sb.jpg	EyeEm n.d., *Drone views*, digital image of a multi-rotor sUAS flying in clear sky with properties below, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/drone-	Clouds Cropped	Freepik License

	views_109456360.htm#fromView=author&page=17&position=48&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		
24_0_sb.jpg	EyeEm n.d., *Low angle view of drone airplane flying over lake by mountains against sky*, digital image of a multi-rotor sUAS flying in clear sky with mountainous terrain and water body below, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/low-angle-view-drone-airplane-flying-lake-by-mountains-against-sky_123626608.htm#fromView=author&page=19&position=6&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
25_0_sb.jpg	EyeEm n.d., *Rear view senior man wearing cap flying drone against sky*, digital image of a multi-rotor sUAS flying in clear sky with landscape below, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/rear-view-senior-man-wearing-cap-flying-drone-against-sky_123166989.htm#fromView=author&page=20&position=41&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>	Person cropped	Freepik License
26_0_sb.jpg	EyeEm n.d., *Low angle view of buildings against clear blue sky*, digital image of a multi-rotor sUAS flying in clear sky with building below, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/low-angle-view-buildings-against-clear-blue-sky_103979173.htm#fromView=author&page=23&position=21&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
27_0_sb.jpg	EyeEm n.d., *Dji drone mavic 2 pro with hasselblad camera flying against blue sky copy space*, digital image of a multi-rotor sUAS flying in clear sky with trees and sun, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/dji-drone-mavic-2-pro-with-hasselblad-camera-flying-against-blue-sky-copy-space_114312539.htm#fromView=author&page=26&position=8&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
28_0_sb.jpg	EyeEm n.d., *Flying drone near scenic view of sea against clear sky*, digital image of a multi-rotor sUAS flying in clear sky with water body below with person, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/flying-drone-near-scenic-view-sea-against-clear-sky_119107899.htm#fromView=author&page=28&position=22&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
29_0_sb.jpg	EyeEm n.d., *Low angle view of drone flying against clear blue sky during sunny day*, digital image of a multi-rotor sUAS flying in clear sky with telecommunications tower at the side, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/low-angle-view-drone-flying-against-clear-blue-sky-sunny-		Freepik License

	day_100050220.htm#fromView=author&page=30&position=22&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		
30_0_sb.jpg	EyeEm n.d., *A flighting drone with san peter dome in the background*, digital image of a multi-rotor sUAS flying in clear sky with buildings below, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/flighting-drone-with-san-peter-dome-background_131018614.htm#fromView=author&page=37&position=5&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License

Image Attribution of Multi-Rotor sUAS with Low-Light / Clouds / Level 2 Background Features

Low-Light / Clouds / Level 2 Background Features			
File Name	Original Image	Modification Details	Copyright Licence
1_LL_sb.jpg	Sorasak 2017, *drone flying to the horizon*, digital image of a multi-rotor sUAS flying in low-light sky with ground-based background features, Unsplash, viewed 21 October 2024, <https://unsplash.com/photos/silhouette-camera-drone-flying-midair-jlLC2hAQ7uQ>		Unsplash License
2_LL_sb.jpg	EyeEm n.d., *Drone flying over field against sky during sunset*, digital image of a multi-rotor sUAS flying in low-light sky with trees in background, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/drone-flying-field-against-sky-sunset_109019524.htm#fromView=search&page=18&position=36&uuid=aa2f325a-e463-43a4-9c66-7f3bf13d4674>		Freepik License
3_LL_sb.jpg	Jack Redgate n.d., *A Black Quadcopter Drone Flying outdoors*, digital image of a multi-rotor sUAS flying in low-light sky with ground-based background features, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/silhouette-of-drone-at-golden-sunset-14783271/>		Pexels License
4_LL_sb.jpg	EyeEm n.d., *Airplane flying in city against sky during sunset*, digital image of a multi-rotor sUAS flying in low-light sky with city in background, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/airplane-flying-city-against-sky-sunset_100446818.htm#fromView=search&page=11&position=22&uuid=62cd3870-9c65-4ac9-b1fa-5581be8376fb>		Freepik License
5_LL_sb.jpg	Johannes Strötker n.d., *Photo Of Drone During Dawn*, digital image of a multi-rotor sUAS flying in low-light sky with ground-based background features, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/photo-of-drone-during-dawn-3660617/>		Pexels License
6_LL_sb.jpg	EyeEm n.d., *Maze and drone*, ditigal image of a multi-rotor sUAS flying in low-light sky over corn field, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/maze-		Freepik License

	drone_129611276.htm#fromView=search&page=18&position=38&uuid=1d0b7e2c-09f9-415e-b60a-831f83b51402>		
7_LL_sb.jpg	Pexels n.d., *Drone and Sunset Sky*, digital image of a multi-rotor sUAS flying in low-light sky with ground-based background features, Freerange Stock, viewed 21 October 2024,< https://freerangestock.com/photos/173837/drone-hovering-in-dramatic-dusk-sky.html>		Equalicense
8_LL_sb.jpg	Wirestock n.d., *Drone flying over the hills with the beautiful sunset in Kenya, Nairobi, Samburu*, digital image of a multi-rotor sUAS flying in low-light sky with ground-based background features, Freepik, viewed 21 October 2024,< https://www.freepik.com/free-photo/drone-flying-hills-with-beautiful-sunset-kenya-nairobi-samburu_9282609.htm#from_view=detail_alsolike>		Freepik License
9_LL_sb.jpg	Ali, M n.d., *Silhouette of Drone Over the Ocean under Blue Sky*, digital image of a multi-rotor sUAS flying in low-light sky over water body, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/silhouette-of-drone-over-the-ocean-under-blue-sky-6795618/>		Pexels License
10_LL_sb.jpg	Manu n.d., digital image of a multi-rotor sUAS flying in low-light sky over water body, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/a-small-plane-flying-over-the-ocean-at-sunset-Zg46IPfPjko>		Unsplash License
11_LL_sb.jpg	Kaven, J 2016, *Flying with DJI Mavic Pro over Prague*, digital image of a multi-rotor sUAS flying in low-light sky over city, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/black-quadcopter-drone-flying-over-the-hill-during-daytime-e3hH6_pSk1g>		Unsplash License
12_LL_sb.jpg	Loewen, K 2017, *Photo of the Mavic Pro during takeoff*, digital image of a multi-rotor sUAS flying in low-light sky with distant trees out of focus, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/black-quadcopter-drone-hovering-in-the-air-473CzHmPWIE>		Unsplash License
13_LL_sb.jpg	Sandro, A n.d., *Silhouette Photo of Quadcopter*, digital image of a multi-rotor sUAS flying in low-light near a bridge, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/silhouette-photo-of-quadcopter-2473837/>		Pexels License
14_LL_sb.jpg	Avilés, A n.d., *A Drone Flying above the Shore at Sunset*, digital image of a multi-rotor sUAS flying in low-light sky with land formations out of focus in distance, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-drone-flying-above-the-shore-at-sunset-16541140/>		Pexels License
15_LL_sb.jpg	Avilés, A n.d., *Young Man Flying a Drone on the Shore*, digital image of a multi-rotor sUAS flying in low-light sky over water body with person in foreground and buildings in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/young-man-flying-a-drone-on-the-shore-16541142/>		Pexels License

16_LL_sb.jpg	Eremin, D n.d., *Man with Drone at Dusk*, digital image of a multi-rotor sUAS flying in low-light sky with person, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/man-with-drone-at-dusk-16434457/>		Pexels License
17_LL_sb.jpg	Juhasz n.d., *A Drone Hovering Over the Remote Control*, digital image of a multi-rotor sUAS flying in low-light sky with hay bales and remote in foreground as well as trees out of focus in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-drone-hovering-over-the-remote-control-10017498/>	Overhead Powerline cropped	Pexels License
18_LL_sb.jpg	Januzzi, JR n.d., *Drone Flying above a Shirtless Man in Water*, digital image of a multi-rotor sUAS flying over water body with person in low-light sky, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/drone-flying-above-a-shirtless-man-in-water-9202648/>		Pexels License
19_LL_sb.jpg	Aydemir, B n.d., *A Man Flying a Drone Machine*, digital image of a multi-rotor sUAS flying in low-light sky over grass landscape with person, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-man-flying-a-drone-machine-10356051/>		Pexels License
20_LL_sb.jpg	Wirestock n.d., *Airliner figure flying in the sky under a cloudy sky*, digital image of a multi-rotor sUAS flying under overcast sky with people, trees and buildings out of focus in background, Freepik, viewed 21 October 2024, < https://www.freepik.com/free-photo/airliner-figure-flying-sky-cloudy-sky_12304866.htm#fromView=search&page=1&position=31&uuid=197802ea-d9c5-46f4-8d6e-519e8bbc0f86>		Freepik License
21_LL_sb.jpg	Wirestock n.d., *A drone hovering up above*, digital image of a multi-rotor sUAS flying under overcast sky with bridge out of focus in background, Freepik, viewed 21 October 2024, < https://www.freepik.com/free-photo/drone-hovering-up_7630363.htm#fromView=search&page=1&position=14&uuid=197802ea-d9c5-46f4-8d6e-519e8bbc0f86>		Freepik License
22_LL_sb.jpg	VanderMeer, J n.d., *Drone Flying over a Body of Water*, digital image of a multi-rotor sUAS flying over body of water with overcast sky and mountainous terrain in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/drone-flying-over-a-body-of-water-8821965/>		Pexels License
23_LL_sb.jpg	tawatchai07 n.d., *Drone with a camera is flying on iceberg.*, digital image of a multi-rotor sUAS flying under overcast sky over glacier with mountain formation in background, Freepik, viewed 21 October 2024, <https://www.freepik.com/free-photo/drone-with-camera-is-flying-iceberg_11768944.htm#fromView=search&page=2&position=6&uuid=197802ea-d9c5-46f4-8d6e-519e8bbc0f86>		Freepik License
24_LL_sb.jpg	ruslan ivantsov n.d., *Flying drone on a background of sea sunset*, digital image of a multi-rotor sUAS flying in low-light sky over body of water, Freepik, viewed 21 October 2024, <		Freepik License

	https://www.freepik.com/premium-photo/flying-drone-background-sea-sunset_5393598.htm#from_view=detail_alsolike>		
25_LL_sb.jpg	esolex n.d., *Drone is flying over the field at sunrise. Modern technological background - silhouette of flying machine in glowing red sunset sky.*, digital image of a multi-rotor sUAS flying in low-light sky over sunflower field, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/drone-flying-sunset-with-wind-turbines-background_369835756.htm#fromView=search&page=10&position=26&uuid=aa2f325a-e463-43a4-9c66-7f3bf13d4674>		Freepik License
26_LL_sb.jpg	EyeEm n.d., *Low angle view of dramatic sky over silhouette mountains during sunset*, digital image of a multi-rotor sUAS flying in low-light sky with mountainous landscape and powerlines in background, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/low-angle-view-dramatic-sky-silhouette-mountains-sunset_120785624.htm#fromView=search&page=13&position=34&uuid=aa2f325a-e463-43a4-9c66-7f3bf13d4674>		Freepik License
27_LL_sb.jpg	alexx-wl n.d., *The male controls a quadrocopter on the sunset background*, digital image of a multi-rotor sUAS flying in low-light sky over water body with person, Freepik, viewed 21 October 2024,< https://www.freepik.com/premium-photo/male-controls-quadrocopter-sunset-background_19994730.htm#fromView=search&page=7&position=9&uuid=aa2f325a-e463-43a4-9c66-7f3bf13d4674>		Freepik License
28_LL_sb.jpg	Norollahi, K n.d., *Close-Up Shot of a Drone Camera*, digital image of a multi-rotor sUAS flying in low-light sky over landscape, Pexels, viewed 21 October 2024, < https://www.pexels.com/photo/close-up-shot-of-a-drone-camera-14907877/>		Pexels License
29_LL_sb.jpg	Lovelypeace n.d., *City at sunset with a frying drone in the golden orange sky*, multi-rotor sUAS flying in low-light sky over city, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/city-sunset-with-frying-drone-golden-orange-sky_17770804.htm#from_view=detail_alsolike>		Freepik License
30_LL_sb.jpg	EyeEm n.d., *Airplane flying in sky at sunset*, multi-rotor sUAS flying in low-light sky over landscape with trees, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/airplane-flying-sky-sunset_122867316.htm#fromView=search&page=4&position=47&uuid=c37195e2-fe05-4aaf-b8d3-5b667bf75349>		Freepik License

Image Attribution of Multi-Rotor sUAS with Day Light / Clouds / Level 3 Background Features

Day Light / Clouds / Level 3 Background Features			
File Name	Original Image	Modification Details	Copyright Licence
1_DL_cb.jpg	EyeEm n.d., *View of motorcycle*, digital image of a multi-rotor sUAS flying with a building and trees in background out of focus, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/view-motorcycle_127517732.htm#fromView=author&page=20&position=48&uuid=d4c4bc88-77a8-4102-beb7-13f468068738>		Freepik License
2_DL_cb.jpg	Ocharonata n.d., *A gray drone flies in the sky over the construction*, digital image of a multi-rotor sUAS flying with construction materials in background out of focus, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/gray-drone-flies-sky-construction_371409737.htm#fromView=search&page=6&position=18&uuid=09ba8d4c-5b06-451b-925d-6ca24018dc58>		Freepik License
3_DL_cb.jpg	Henig, I 2021, *white drone flying during daytime,* digital image of a multi-rotor sUAS flying over shrubbery, Unsplash, viewed 21 October 2024, <https://unsplash.com/photos/white-drone-flying-during-daytime-4gWz2nJv0ac >		Unsplash License
4_DL_cb.jpg	Halicki, J 2024, *2024 Dron DJI Mini 4 Pro (17)*, digital image of a multi-rotor sUAS flying over grass, Wikimedia Commons, viewed 21 October 2024, < https://commons.wikimedia.org/wiki/File:2024_Dron_DJI_Mini_4_Pro_(17).jpg>		Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)
5_DL_cb.jpg	Lindner, C 2021, *flying, flight, landscape, and aerial*, digital image of a multi-rotor sUAS flying with shrubbery in background, Unsplash, viewed 21 October 2024, <https://unsplash.com/photos/a-remote-controlled-airplane-flying-over-a-forest-wQyBwNgsYfA>		Unsplash License
6_DL_cb.jpg	HKesteloo 2021, *DJI Mavic 3*, digital image of a multi-rotor sUAS flying with shrubbery in background, Wikimedia Commons, viewed 21 October 2024, < https://commons.wikimedia.org/wiki/File:DJI_Mavic_3.jpg>	Image cropped to remove sky.	Creative Commons Attribution-ShareAlike 4.0

			International (CC BY-SA 4.0)
7_ DL_cb.jpg	Dinu, DR 2021, *black helicopter flying over green mountains during daytime*, digital image of a multi-rotor sUAS flying with forested mountain range in background and clear skies, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/black-helicopter-flying-over-green-mountains-during-daytime-18DnTApZHzU>		Unsplash License
8_ DL_cb.jpg	Stewart, W n.d., digital image of a multi-rotor sUAS flying amongst forest, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/gray-quadcopter-drone-in-the-middle-of-forest-during-day-1QrqstxH-Uw>		Unsplash License
9_ DL_cb.jpg	Bublikhaus n.d, *Flying drone landing on sand at beach*, digital image of a multi-rotor sUAS flying over sand at beach, Freepik, viewed 21 October 2024,< https://www.freepik.com/free-photo/flying-drone-landing-sand-beach_11253671.htm#fromView=search&page=1&position=0&uuid=914c0a12-504f-42d9-bf29-8a2eb1cc9c97>		Freepik License
10_ DL_cb.jpg	Vinh, QN n.d., *Dji Phantom Drone on Air*, digital image of a multi-rotor sUAS flying over flower farm with people in background, Pexels, viewed 21 October 2024, < https://www.pexels.com/photo/dji-phantom-drone-on-air-2157925/>		Pexels License
11_ DL_cb.jpg	Osman, T n.d., *Drone Flying over Road*, digital image of a multi-rotor sUAS flying over road with trees and power pole in background out of focus, Pexels, viewed 21 October 2024, <https://www.pexels.com/photo/drone-flying-over-road-17037083/>		Pexels License
12_ DL_cb.jpg	EyeEm n.d., *Dji mavic 2 pro drone from dji is flown on a green meadow by a female pilot*, digital image of a multi-rotor sUAS flying over grass with person and trees in background out of focus, Freepik, viewed 21 October 2024,<https://www.freepik.com/premium-photo/dji-mavic-2-pro-drone-from-dji-is-flown-green-meadow-by-female-pilot_100518730.htm#fromView=author&page=1&position=3&uuid=5afaab16-a69e-4002-9c4f-51432b476b48>		Freepik License
13_ DL_cb.jpg	Miles, L 2018, *gray quadcopter drone*, digital image of a multi-rotor sUAS flying with wooden fence and trees in background, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/gray-quadcopter-drone-f-OXUlmT4_U>		Unsplash License
14_ DL_cb.jpg	Taufik, A n.d., *Close-up of a Flying Drone*, digital image of a multi-rotor sUAS flying with building and trees in background out of focus, Pexels, viewed 21 October 2024, <https://www.pexels.com/photo/close-up-of-a-flying-drone-10137950/>		Pexels License

15_DL_cb.jpg	Oleksandr P n.d., *White Drone Quadcopter*, digital image of a multi-rotor sUAS flying with building in background, Pexels, viewed 21 October 2024, <https://www.pexels.com/photo/white-drone-quadcopter-1093232/>		Pexels License
16_DL_cb.jpg	AEY n.d., *Drone Flying Close to a Lake*, digital image of a multi-rotor sUAS flying over water body with grass and trees on the banks and overcast skies, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/drone-flying-close-to-a-lake-18677901/>		Pexels License
17_DL_cb.jpg	cottonbro studio n.d., *A Drone Flying over a River*, digital image of a multi-rotor sUAS flying over water body with mountainous terrain in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-drone-flying-over-a-river-9940763/>		Pexels License
18_DL_cb.jpg	Fazekas, P n.d., *Drone on the Rock*, digital image of a multi-rotor sUAS sitting on rock with flowers in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/drone-on-the-rock-8231825/>		Pexels License
19_DL_cb.jpg	Chomka, K 2022, digital image of a multi-rotor sUAS with sea and boat in background, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/a-drone-flying-over-a-beach-oxf6IScAU0Q>		Unsplash License
20_DL_cb.jpg	Mendez, J n.d. *Photo of a Flying Drone*, digital image of a multi-rotor sUAS flying with graffiti background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/photo-of-a-flying-drone-13002780/>		Pexels License
21_DL_cb.jpg	Kandemir, V n.d., *Close-up of a Drone Flying above a Street in City*, digital image of a multi-rotor sUAS flying with city background out of focus, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/close-up-of-a-drone-flying-above-a-street-in-city-20049700/>		Pexels License
22_DL_cb.jpg	Demir, A n.d., *A Drone in Winter*, digital image of a multi-rotor sUAS flying in snow covered area with trees and fence in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-drone-in-winter-10958083/>		Pexels License
23_DL_cb.jpg	SCREEN POST n.d., *Close-Up Shot of a White Drone Flying*, digital image of a multi-rotor sUAS flying with trees in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/close-up-shot-of-a-white-drone-flying-5948309/>		Pexels License
24_DL_cb.jpg	Diemar, D 2018, *DJI Phantom Test Flight*, digital image of a multi-rotor sUAS flying with building in background, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/white-concrete-house-at-daytime-xoyHw4P0gyA>		Unsplash License
25_DL_cb.jpg	Ataş, R n.d., *Close Up Shot of a Drone*, digital image of a multi-rotor sUAS in focus and person behind out of focus, Pexels, viewed 21 October 2024, < https://www.pexels.com/photo/close-up-shot-of-a-drone-5734963/>		Pexels License

26_ DL_cb.jpg	Fernandes, R 2020, *blue and black drone flying during daytime*, digital image of a multi-rotor sUAS flying with concrete structure(s) in background, Unsplash, viewed 21 October 2024, <https://unsplash.com/photos/blue-and-black-drone-flying-during-daytime-YksvK11AFuU>		Unsplash License
27_ DL_cb.jpg	Image Hunter n.d., *Modern Drone Flying over Mountains*, digital image of a multi-rotor sUAS flying with mountain background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/modern-drone-flying-over-mountains-13012582/>		Pexels License
28_ DL_cb.jpg	Pesce, M 2016, *DJI Phantom Drone (24109503259)*, digital image of a multi-rotor sUAS flying indoors with people, lights, a net and display screens out of focus in the background, Wikimedia Commons, viewed 21 October 2024, < https://commons.wikimedia.org/wiki/File:DJI_Phantom_Drone_(24109503259).jpg>		Creative Commons Attribution 2.0 Generic (CC BY 2.0)
29_ DL_cb.jpg	Nester, S n.d., *Drone Pilot on Railway Tracks in the Forest*, digital image of a multi-rotor sUAS flying next to person with railway tracks and forest in background, Pexels, viewed 21 October 2024, <https://www.pexels.com/photo/drone-pilot-on-railway-tracks-in-the-forest-18834775/>		Pexels License
30_ DL_cb.jpg	Halicki, J 2015, *2015 Dron DJI Phantom 3 Advanced*, digital image of a multi-rotor sUAS flying in clear sky with trees, a van, building and lamp post out of focus in the background, Wikimedia Commons, viewed 21 October 2024, < https://commons.wikimedia.org/wiki/File:2015_Dron_DJI_Phantom_3_Advanced.JPG>		Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

Image Attribution of Multi-Rotor sUAS with Low-Light Sky / Clouds / Level 3 Background Features

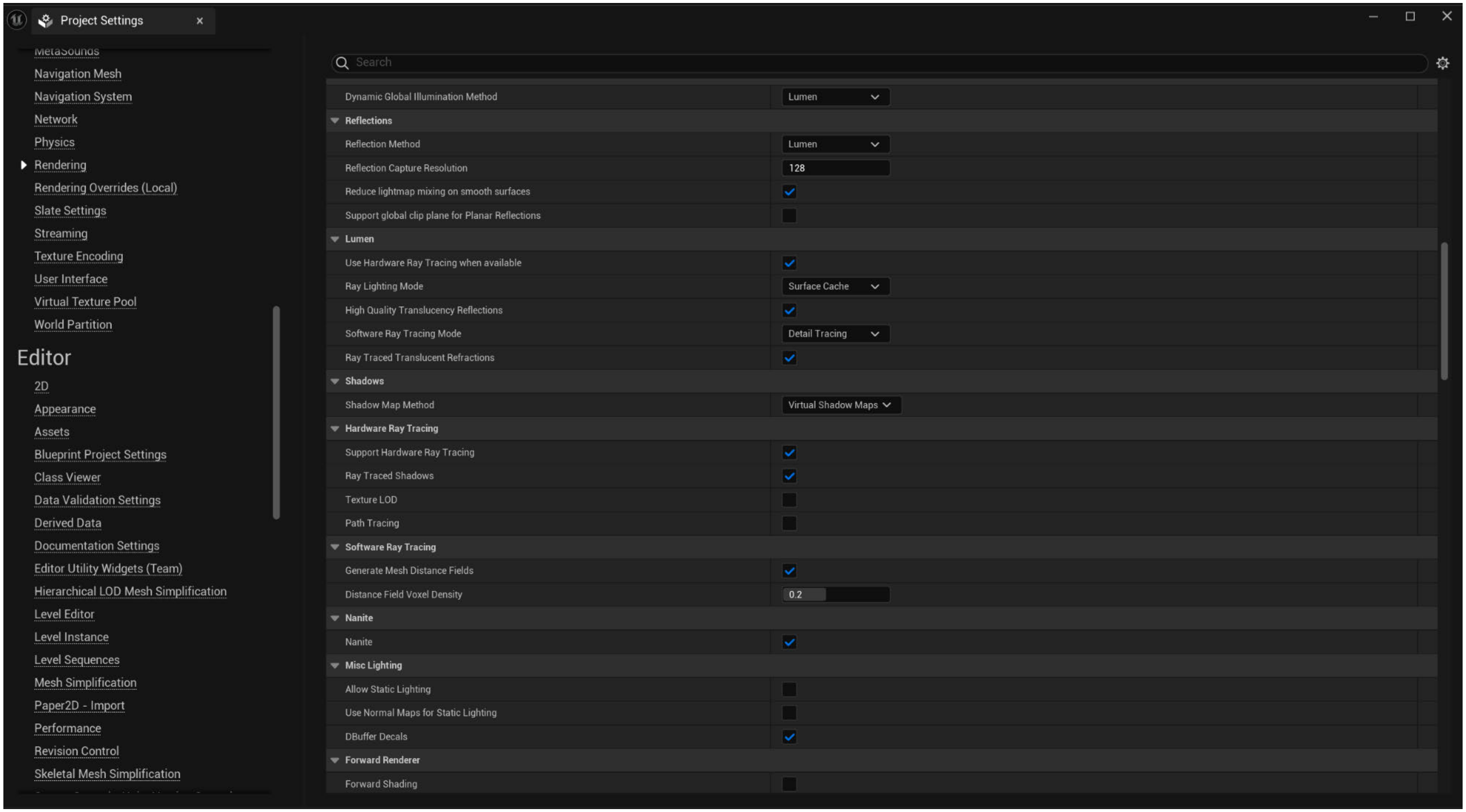
Low-Light Sky / Clouds / Level 3 Background Features			
File Name	Original Image	Modification Details	Copyright Licence
1_LL_cb.jpg	Fazekas, P n.d., *Photo of Drone Flying above the Lake*, digital image of a multi-rotor sUAS flying in low-light sky over water body with buildings and elevated terrain in background, Pexels, viewed 21 October 2024, <https://www.pexels.com/photo/photo-of-drone-flying-above-the-lake-997135/>		Pexels License
2_LL_cb.jpg	Arenas, D n.d., *A White Drone Flying*, digital image of a multi-rotor sUAS flying in low-light skies with city in background out of focus, Pexels, viewed 21 October 2024, < https://www.pexels.com/photo/a-white-drone-flying-11690533/>		Pexels License
3_LL_cb.jpg	Beirne, R n.d., *Close-up of a Flying DJI Mini Drone*, digital image of a multi-rotor sUAS flying in low-light sky, <https://www.pexels.com/photo/close-up-of-a-flying-dji-mini-drone-20339627/>		Pexels License
4_LL_cb.jpg	Rasmussen, A 2019, digital image of a multi-rotor sUAS flying over flowers with low-light sky, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/black-drone-during-daytime-C8UjCsWp3Go>		Unsplash License
5_LL_cb.jpg	Javardh n.d., digital image of a multi-rotor sUAS flying over water body, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/yellow-dji-phantom-pro-above-water-during-golden-hour-F1-mzEiLJOg>		Unsplash License
6_LL_cb.jpg	Banks, C 2019, *Hovering the Mavic Pro around some autumn trees*, digital image of a multi-rotor sUAS flying in low-light sky with trees, cars and overcast sky out of focus in the background, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/gray-and-black-drone-midair-19saRtNCKHY>		Unsplash License
7_LL_cb.jpg	Tomašić, N n.d., *Close Up of Flying Drone*, digital image of a multi-rotor sUAS flying in low-light sky with trees out of focus in background, Pexels, viewed 21 October 2024, <https://www.pexels.com/photo/close-up-of-flying-drone-15499221/>		Pexels License
8_LL_cb.jpg	Avilés, A n.d, *A Drone Flying above the Shore*, digital image of a multi-rotor sUAS flying near shore line of water body with rocks and trees in background out of focus, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-drone-flying-above-the-shore-16541141/>		Pexels License
9_LL_cb.jpg	Lautarosoto n.d., *White drone flying and filming over mountains and lake*, digital image of a multi-rotor sUAS flying in low-light sky with mountain in background, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/white-drone-flying-filming-		Freepik License

	mountains-lake_247878940.htm#fromView=search&page=22&position=45&uuid=0d203cc3-c346-4049-b302-bdb163a1597f>		
10_LL_cb.jpg	Lautarosoto n.d., *White drone flying and filming over mountains and lake*, digital image of a multi-rotor sUAS flying in low-light sky with mountain in background, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/white-drone-flying-filming-mountains-lake_247878442.htm#from_view=detail_alsolike>		Freepik License
11_LL_cb.jpg	Lautarosoto n.d., *White drone flying and filming over mountains and lake*, digital image of a multi-rotor sUAS flying in low-light sky with mountain and lake in background, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/white-drone-flying-filming-mountains-lake_247878984.htm#fromView=search&page=2&position=24&uuid=e49393fe-fbe0-4260-92cf-ee0ac70ccb36>		Freepik License
12_LL_cb.jpg	life._.kor n.d., *A Black Drone Camera Flying*, digital image of a multi-rotor sUAS flying in low-light sky with mountainous terrain in background, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-black-drone-camera-flying-11168594/>		Pexels License
13_LL_cb.jpg	JESHOOTS.com n.d., *Black Quadcopter Drone on Green Grass Field*, digital image of a multi-rotor sUAS flying in low-light sky over field, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/black-quadcopter-drone-on-green-grass-field-442589/>		Pexels License
14_LL_cb.jpg	Wirestock n.d., *Drone flying over the sea and the beach*, digital image of a multi-rotor sUAS flying in low-light sky over beach area, Freepik, viewed 21 October 2024, < https://www.freepik.com/free-photo/drone-flying-sea-beach_9654482.htm#fromView=search&page=30&position=49&uuid=62cd3870-9c65-4ac9-b1fa-5581be8376fb>		Freepik License
15_LL_cb.jpg	Avramoski, P n.d., *Professional drone flying against mountains*, digital image of a multi-rotor sUAS flying near shoreline of water body with mountainous terrain in background out of focus, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/professional-drone-flying-against-mountains-6081353/>		Pexels License
16_LL_cb.jpg	Altansukh E, *Close-Up Photo of Drone*, digital image of a multi-rotor sUAS flying in low-light with mountainous terrain an trees in background out of focus, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/close-up-photo-of-drone-6504158/>		Pexels License
17_LL_cb.jpg	Hashim, A n.d., *A Drone in Flight*, digital image of a multi-rotor sUAS flying in low-light with vegetation background out of focus, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-drone-in-flight-16255721/>		Pexels License
18_LL_cb.jpg	Horst, BV 2020, *DJI Mavic 2 drone flying close to the ground in the Austrian Alps.*, digital image of a multi-rotor sUAS flying in low-light with snow and trees in background,		Unsplash License

	Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/black-helicopter-flying-over-snow-covered-ground-during-daytime-gK7xAE4ZKxw>		
19_LL_cb.jpg	Lloyd, C 2021, digital image of a multi-rotor sUAS flying in low-light, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/black-and-white-drone-flying-over-the-brown-field-during-daytime-kx6VkhGxGdw>		Unsplash License
20_LL_cb.jpg	Khodzinskyi, V 2023, digital image of a multi-rotor sUAS flying in low-light sky, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/a-white-remote-control-flying-over-a-lush-green-field-r_FtE_hTrrg>		Unsplash License
21_LL_cb.jpg	Khodzinskyi, V 2022, *a remote controlled flying device in the air*, digital image of a multi-rotor sUAS flying in low-light sky with building, pathway and trees in background out of focus, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/a-remote-controlled-flying-device-in-the-air-4ViPQJq7S_U>		Unsplash License
22_LL_cb.jpg	Ramadan, O n.d, *A Flying Drone Camera*, digital image of a multi-rotor sUAS flying in low-light sky, Pexels, viewed 21 October 2024,< https://www.pexels.com/photo/a-flying-drone-camera-9977848/>		Pexels License
23_LL_cb.jpg	tourist07, n.d, *Small compact quadcopter flies in the air on sunset drone hovers and stays stable*, digital image of a multi-rotor sUAS flying in low-light sky, Freepik, <https://www.freepik.com/premium-photo/small-compact-quadcopter-flies-air-sunset-drone-hovers-stays-stable_28389719.htm#fromView=search&page=11&position=33&uuid=1d0b7e2c-09f9-415e-b60a-831f83b51402>		Freepik License
24_LL_cb.jpg	EyeEm n.d., *Domestic drone flying in the sunset with railway trestle bridge in soft focus background*, digital image of a multi-rotor sUAS flying in low-light sky with structure in background out of focus, Freepik, viewed 21 October 2024, < https://www.freepik.com/premium-photo/domestic-drone-flying-sunset-with-railway-trestle-bridge-soft-focus-background_110091401.htm#fromView=search&page=12&position=12&uuid=1d0b7e2c-09f9-415e-b60a-831f83b51402>		Freepik License
25_LL_cb.jpg	EyeEm n.d., *Airplane flying over field against sky during sunset*, digital image of a multi-rotor sUAS flying in low-light sky with trees in background out of focus, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/airplane-flying-field-against-sky-sunset_97181928.htm#fromView=search&page=14&position=28&uuid=1d0b7e2c-09f9-415e-b60a-831f83b51402>		Freepik License
26_LL_cb.jpg	Bublikhaus n.d., *Werial drone flying low to ground on beach*, digital image of a multi-rotor sUAS flying in low-light sky over sand, Freepik, viewed 21 October 2024, <		Freepik License

	https://www.freepik.com/free-photo/werial-drone-flying-low-ground-beach_11253673.htm#fromView=search&page=1&position=7&uuid=b55b9694-435f-4ca1-8676-a23b7d358c64>		
27_LL_cb.jpg	Pohle, J 2022, digital image of a multi-rotor sUAS flying in low-light with trees in background out of focus, Unsplash, viewed 21 October 2024,< https://unsplash.com/photos/a-close-up-of-a-remote-controlled-flying-object-XkvV4SS6GY4>		Unsplash License
28_LL_cb.jpg	Benwhitephotography n.d, *Flying black drone during winter*, digital image of a multi-rotor sUAS flying in snow in forest area under low-light, Freepik, viewed 21 October 2024, < https://www.freepik.com/free-photo/flying-black-drone-winter_12647483.htm#fromView=search&page=3&position=15&uuid=dbaf4764-71c5-4bee-9dea-6736a9f2e5b7>		Freepik License
29_LL_cb.jpg	Soto, J 2020, *DJI Mini 2*, digital image of a multi-rotor sUAS sitting on ground in low-light, Unsplash, viewed 21 October 2024, < https://unsplash.com/photos/gray-and-black-drone-on-ground-during-daytime-dxYA0UDWm7g>		Unsplash License
30_LL_cb.jpg	EyeEm n.d., *High angle view of man jumping on rock*, digital image of a multi-rotor sUAS sitting on ground in low-light, Freepik, viewed 21 October 2024, <https://www.freepik.com/premium-photo/high-angle-view-man-jumping-rock_125413606.htm#fromView=search&page=4&position=18&uuid=291a5143-765c-4fe7-8993-0e2c5ffd0195>		Freepik License

Appendix C – Hardware Ray Tracing Settings



Dynamic Global Illumination Method	Lumen
Reflections	
Reflection Method	Lumen
Reflection Capture Resolution	128
Reduce lightmap mixing on smooth surfaces	☑
Support global clip plane for Planar Reflections	□
Lumen	
Use Hardware Ray Tracing when available	☑
Ray Lighting Mode	Surface Cache
High Quality Translucency Reflections	☑
Software Ray Tracing Mode	Detail Tracing
Ray Traced Translucent Refractions	☑
Shadows	
Shadow Map Method	Virtual Shadow Maps
Hardware Ray Tracing	
Support Hardware Ray Tracing	☑
Ray Traced Shadows	☑
Texture LOD	□
Path Tracing	□
Software Ray Tracing	
Generate Mesh Distance Fields	☑
Distance Field Voxel Density	0.2
Nanite	
Nanite	☑
Misc Lighting	
Allow Static Lighting	□
Use Normal Maps for Static Lighting	□
DBuffer Decals	☑
Forward Renderer	
Forward Shading	□

End of Research Paper