

University of Southern Queensland

School of Engineering

**Automated Cattle Counting: Leveraging YOLOv8 for Accurate  
Object Detection in Feedlot Environments**

A dissertation submitted by

Daniel Steven O'Brien

in fulfillment of the requirements of

**ENP4111 Professional Engineer Research Project**

towards the degree of

**Bachelor of Engineering Honours**

2024

**This page intentionally left blank**

# **University of Southern Queensland**

## **School of Engineering**

### **ENP4111 Dissertation Project**

(This is a 2-unit research project in Bachelor of Engineering Honours Program)

#### **Limitations of Use**

The Council of the University of Southern Queensland, its Academic Affairs, 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 and Science 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 this dissertation 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.

## **CERTIFICATION**

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.

**Daniel O'Brien**

**Student Number:** [REDACTED]

[REDACTED]

Signature

03/11/2024

Date

## ABSTRACT

Based on wanting to modernise current livestock production processes, this project investigated whether an artificial intelligence-driven model can accurately detect cattle from remotely captured Unmanned Aerial Vehicle (UAV) images. By combining recent developments in artificial intelligence-driven, real-time object detection with well-established and tested methods for gathering aerial images (i.e., UAVs fitted with high-resolution cameras), we obtained accurate information about cattle numbers in tightly held feedlot pens. Five different pre-trained machine learning models that differed in size and complexity (i.e., YOLOv8-s, a smaller, faster model designed for real-time applications where speed is crucial and YOLOv8-m, a medium-sized model that offered improved accuracy at the cost of slightly slower inference times), were fine-tuned to recognise images of individual cattle from the UAV images captured. Models also compared object identification using both a full image, high resolution dataset (4056x3040 pixels) with a tiled image dataset, where the original full image dataset was divided into smaller 640x640 pixel tiles. Results indicated that the YOLOv8-m\_full model outperformed all other models tested (i.e., YOLOv8-\_tiled; YOLOv8-s\_full; YOLOv8-s\_tiled; RoboFlow\_custom) in metrics related to Precision, Mean Average Precision, and recall. This result highlighted that a Convolutional Neural Networks parameters such as the number of network layers, directly impacts the detection accuracy in complex datasets. The project outcomes also present an applied system for the autonomous monitoring (i.e., detect, count, identify) of cattle in intensive feedlot settings that can reduce the amount of manual labour, and associated costs, currently required to monitor cattle in extensive production environments.

## **ACKNOWLEDGEMENTS**

My thesis project and overall journey would not have been possible without the help and support I received over the past two years. First, my supervisor, Dr Derek Long, was prepared to supervise me for the duration of this project. His experience and interest in applied computer vision and the machine learning field meant that I was strongly guided throughout my thesis. Dr Long's contacts in the livestock industry enabled us to complete the data collection required to conduct the project and he was always available to assist with my project questions and analysis of the computer vision using different machine learning tools. Without this help, I would not have been able to finish this project. I would also like to acknowledge my family. Their unwavering support and encouragement were vital throughout this journey.

# TABLE OF CONTENTS

	PAGE
CERTIFICATION .....	iv
ABSTRACT .....	v
ACKNOWLEDGEMENTS .....	vi
TABLE OF CONTENTS .....	vii
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
NOMENCLATURE .....	xii
GLOSSARY .....	xiii
Chapter 1 INTRODUCTION .....	1
Chapter 2 BACKGROUND AND LITERATURE REVIEW .....	5
2.1    CURRENT INTENSIVE CATTLE FEEDLOT SYSTEMS .....	6
2.2    UNMANNED AERIAL VEHICLES (UAVs) .....	8
2.3    CNN ARCHITECTURE DEVELOPMENT .....	11
2.4    MECHANISMS BEHIND CNN ARCHITECTURES .....	12
2.5    SOFTWARE DEVELOPMENT – AI ALGORITHMS .....	17
2.6    STUDIES RELATED TO REMOTE DATA CAPTURE .....	19
Chapter 3 METHODOLOGY .....	21
3.1    STUDY AREA AND OBJECTS .....	21
3.2    UAV HARDWARE - OVERVIEW .....	22

3.3	IMAGE DETECTION WORKFLOW .....	25
Chapter 4 MODEL DEVELOPMENT .....		30
4.1	PEROFRMANCE METRICS .....	31
4.1.1	OBJECT DETECTION MODEL OUTPUT .....	32
4.1.2	CLASSIFICATION AND EVALUATIONI METRICS .....	32
Chapter 5 RESULTS AND DISCUSSION .....		34
5.1	ROBOFLOW_CUSTOM MODEL .....	34
5.2	YOLOV8-S_FULL MODEL .....	34
5.3	YOLOV8-S_TILED MODEL .....	34
5.4	YOLOV8-M_FULL MODEL .....	35
5.5	YOLOV8-M_TILED MODEL .....	35
5.7	COMPARISION OF DETECTION PROCESSES & MODELS .....	36
5.8	IMPACT OF UAV FLIGHT ALTITUDE ON DETECTION ACCURACY.....	38
Chapter 6 CONCLUSION.....		39
REFERENCES .....		41
APPENDIX A .....		45

## LIST OF TABLES

Table 3.2 Technical Specifications – Mavic 2 P600 UAV	23
Table 3.4 Project Training Models	31
Table 4.1.2A Equation for Precision	33
Table 4.1.2B Equation for Recall	33
Table 4.1.2C Equation for Average Precision and Mean Average Precision	33
Table 5.1 Precision, Recall, and mAP metrics after training	36

## **LIST OF FIGURES**

Figure 2.1 Beef City Feedlot – Aerial View and Production Pen View	<b>7</b>
Figure 2.2 UAV Types (Australian Aerial Imagery, 2023)	<b>9</b>
Figure 2.2.1 Articles on Livestock Detection Using UAVs	<b>10</b>
Figure 2.3 CNN Development (v7labs.com, 2024)	<b>12</b>
Figure 2.4 Fast R-CNN mechanisms (viso.ai, 2024)	<b>13</b>
Figure 2.4.1: Layers in a Fast R-CNN Search Process (v7labs.com, 2024)	<b>14</b>
Figure 2.4.2 YOLOv10 Architecture (Ghosh, 2024)	<b>16</b>
Figure 2.5 Roboflow Vision Model Training Exemplar (Roboflow.com, 2024)	<b>18</b>
Figure 2.5.1 Ultralytics Training Exemplar (ultralytics, 2024)	<b>18</b>
Figure 2.6 Examples DL Methods using UAV Imagery (Yousefi et al., 2022)	<b>20</b>
Figure 3.1 Feedlot Cattle (Andrews, 2015)	<b>22</b>
Figure 3.2.1 Aerial UAV Images Captured from the Dalby Study Site	<b>24</b>
Figure 3.3 Project Workflow Design	<b>25</b>
Figure 3.3.1 UAV Captured Image for Project Dataset	<b>26</b>
Figure 3.2.2 Annotated Dataset Image	<b>27</b>
Figure 3.3.3 YOLO Txt Format: [class, x-coordinate, y-coordinate, width, height]	<b>27</b>

Figure 3.3.4 “Tiled 640x640 Image” - Created from Original 4056x3040 Image **29**

Figure 3.3.5 Trained Image (‘0’ represents a class of object detected i.e., cattle) **30**

## NOMENCLATURE

AP = Average-Precision

i = Index for classes in the dataset

n = Specific index or threshold in recall/precision calculations.

N = Total number of classes in the object detection model

P = Precision

R = Recall

## GLOSSARY

AI	=	Artificial Intelligence
CNN	=	Convolutional Neural Network
FN	=	False Negative
FP	=	False Positive
GPS	=	Global Positioning System
IDE	=	Integrated Development Environment
IoU	=	Intersection over Union
mAP	=	Mean Average Precision
R-CNN	=	Region-Based Convolutional Neural Network
RoIs	=	Regions of Interest
SSD	=	Single Shot Multi-Box Detector
TN	=	True Negative
TP	=	True Positive
UAV	=	Unmanned Aerial Vehicle
YOLO	=	You Only Look Once

# CHAPTER 1 INTRODUCTION

The management of livestock in intensive production environments such as cattle feedlots can be challenging due to production processes not keeping pace with both industry and technological advances (Barbedo et al., 2020, p. 1; Li & Xing, 2019, p. 324). As the importance of livestock production continues to grow, innovative technologies and modern techniques are starting to play increasingly important roles in helping society fulfill its future demands for food supply in more efficient and effective ways (Džermeikaitė et al., 2023, p. 780). For example, the term ‘smart farming’ is now applied to describe the technification of tracking and monitoring livestock in relation to hardware advancements (e.g., automated sensors, actuators, manned and unmanned vehicles), including cutting-edge Artificially Intelligent (AI) systems that automatically track the movement of livestock from video-based information (Ortega et al., 2023, p. 2). Likewise, the development of digital twin systems, where a digital replica of the physical system mirrors the real-world counterpart, allowing for simulation, analysis, monitoring, and optimisation in real-time or near-real-time, also offer an alternative, cutting-edge method for addressing issues related to livestock production typically characterised by high costs and significant levels of human input (Symeonaki et al., 2024, p. 686). However, while many of these advances offer the livestock industry promising tools and technologies, to the best of our knowledge, cost effective, applied innovations that combine hardware advancements with contemporary, cutting-edge AI to overcome the intensive nature of livestock production have been limited to date and require further research (Džermeikaitė et al., 2023, p. 780; Ortega et al., 2023, p. 2; Symeonaki et al., 2024, p. 691).

Notably, known industry and technological advances in livestock management over the past five years include major developments in Unmanned Aerial Vehicle (UAV) technologies

(see Barbedo et al., 2020; Los et al., 2023), which have made the remote monitoring of livestock in both outdoor and intensive production environments much simpler. For example, feeding patterns, stress factors, livestock numbers, optimal breeding times, and body dimensions such as height and weight have all been monitored using basic UAVs (Symeonaki et al., 2024, p. 689). Supported by the development of high-resolution camera technologies, applied researchers have also begun to explore how to count cattle in crowded feedlot settings and dispersed natural environments such as open grassland settings using UAV cameras that view the fine texture of images (Alanezi et al., 2022, p. 45002; De Lima Weber et al., 2023, p. 3). The introduction of automated “biosensors and wearable technology in livestock management is also becoming increasingly widespread, as these modern-day instruments can facilitate the early detection of diseases in animals, decreasing economic losses” (Džermeikaitė et al., 2023, p. 781). Likewise, livestock tracking sensors and cattle collars are becoming useful for finding the location of herds in remote regions, including the combined use of a UAVs fitted with high-resolution cameras that can visually check the livestock found (Los et al., 2023, p. 2). Moreover, according to Mücher et al., (2022), Artificial Intelligence (AI) technologies are also gradually gaining momentum as a modern, cutting-edge method for monitoring cattle and sheep, especially in intensive production environments such as cattle and sheep feedlot settings (p. 2377).

Based on wanting to modernise current livestock production processes, this project plans to investigate whether an AI-driven model can accurately detect cattle numbers from remotely captured UAV images. By combining recent developments in AI-driven, real-time object detection with well-established and tested methods for gathering aerial images (i.e., UAVs fitted with high-resolution cameras), we expect to obtain accurate information about cattle numbers that reduces the amount of manual labour, and associated costs, currently required

to monitor cattle in extensive production environments. The research described in the project therefore presents a system for the autonomous monitoring (i.e., detect, count, identify) of cattle in intensive feedlot settings using a UAV (Rivas et al., 2018, p. 2049). Essentially, a Convolutional Neural Network (CNN) is used to identify the objects captured by the UAV images. CNNs are one of the most widely used image recognition techniques as they form the backbone or hidden architecture in readily available, open source, object detection models such as Roboflow, Single Shot MultiBox Detector (SSD), You Only Look Once (YOLO), and Mask R-CNN (De Lima Weber et al., 2023, p. 100900; Xu et al., 2020, p. 105300). By comparing two different, readily available CNNs (e.g., Roboflow and YOLO), the goal of the research project is to design an applied model for automating livestock detection in closely held, commercially designed, feedlot environments that regularly hold cattle numbers ranging from 50 to 300 head (Watts & Davis, p. 5). This cutting-edge project will contribute to the fields of computer vision and digital object identification. It will also provide real-world information that assists with lowering labour costs, efficiency improvements for cattle growth and development, give greater emphasis to cattle health, safety, and welfare, and contribute to the growth of knowledge in closely related discipline fields such as data analytics, pattern recognition, livestock health technologies, predictive modelling and behaviour analysis.

The practical task of detecting livestock such as cattle in closely confined feedlot pens can be broadly considered as object detection (Alanezi et al., 2022, p. 45004). Object detection accuracy is generally measured by how well a system or model correctly identifies and locates objects (e.g., individual cattle), within an image or video frame (Alanezi et al., 2022, p. 45004; Mùcher et al., 2022, p. 2377). In simple terms this means using Precision, Recall, and Mean Average Precision (mAP) metrics to measure whether the bounding boxes and

class labels are correct in the image or video frames being used (De Lima Weber et al., 2023, p. 100901). In contrast, counting in object detection simply refers to the task of determining the number of instances of a particular object or objects in images (De Lima Weber et al., 2023, p. 100901; Tabernik et al., 2024; p. 110540). For this project, counting is defined as accurately reporting how many times the specific object appears (i.e., cattle in feedlot pens), regardless of whether the exact location (bounding box) is perfect (Tabernik et al., 2024; p. 110540). The novelty of the project design is that it utilises recent advances in AI-based architectures by combining CNN-based algorithms with UAV collected images (captured by the auxiliary UAV camera, positioned perpendicularly to the ground), to create Regions of Interest (RoIs) or “bounding boxes” around individual livestock images (Alanezi et al., 2022, p. 45004). Image analysis in this project will also compare two different CNNs (Roboflow [AutoML] and YOLO) to determine which system provides stronger detection accuracy and counting for the identification of cattle held in closely confined pen spaces (Alanezi et al., 2022, p. 45004). Specific project objectives are:

**Objective 1:**

- Determine which CNN Object Detection Model (YOLOv8.m vs YOLOv8.s vs Roboflow) provides stronger object detection accuracy.

**Objective 2:**

- Determine whether an image tiling-based approach provides stronger object detection accuracy and counting.

**Objective 3:**

- Determine the most appropriate UAV flight level and pattern for data collection.

The remainder of the thesis is structured as follows:

- Chapter 2 reviews current literature in the livestock detection space. This includes how current cattle feedlot environments presently operate, and the opportunities afforded by modern UAVs for remotely collecting data about livestock located in intensive production pens. This chapter also examines the growing use of CNNs for detecting and identifying objects from UAV images (Rivas et al., 2018, p. 2050).
- Chapter 3 outlines the project methodology by defining the property used for data collection, project hardware, image detection protocols and project workflow.
- Chapter 4 provides the model development process and performance metrics used.
- Chapter 5 presents the results of the project by discussing the Object Detection Models, comparing whether full or tiled image datasets provide stronger object detection accuracy and counting across the different CNN models.
- Chapter 6 presents conclusions from the conducted research, highlighting both limitations and potential future project work in the image analysis space.

## **CHAPTER 2 BACKGROUND AND LITERATURE REVIEW**

In what follows, we provide a brief background overview of the design and operations of current cattle feedlots to give context to the overall project design. The types of UAVs used for livestock detection processes are then highlighted, followed by associated developments in CNN architectures and the mechanisms behind CNN architectures and their applications. AI algorithm software developments are then briefly summarised before studies related to remote data capture for livestock in open and intensive settings using UAV imagery and AI algorithms are presented.

## 2.1 CURRENT INTENSIVE CATTLE FEEDLOT SYSTEMS

Cattle feedlots around the globe represent intensive animal production systems that aim to quickly grow and fatten cattle until they reach slaughter weight. In Australia, the close confinement of cattle in open air feedlots using intensive feeding procedures was developed as a method for raising cattle commercially in the 1960s (Grandin, 2016, p. 23; Salvin et al., 2020, p. 1569). Currently, there are approximately 400 accredited cattle feedlots operating across Australia, with an average capacity of 2793 cattle per feedlot (Australian Lot Feeders Association, 2015; Salvin et al., 2020). Notably, facilities for the close handling of various sizes and breeds of cattle need to follow the Australian National Guidelines for Beef Cattle Feedlots (2012); that is, cattle housed in open-air feedlots must be provided with a minimum space allowance of 9m<sup>2</sup>/animal, with a pen slope and drainage system of 2–6% away from feed bunks, and a pen-surface permeability of less than 1 x 10<sup>-9</sup> m/s (Salvin et al., 2020, p. 1572). These legislated requirements highlight how cattle feedlot production processes require strongly regulated processes that result in high running costs due to intensive labour costs, expenses related to feed purchase requirements, extensive infrastructure to move the cattle around safely, and the need to check how many cattle are located within production pen spaces numerous times per year.

Production pens in cattle feedlots represent the main animal housing unit. As shown in Figure 2.1, sound design principles that meet the National Guidelines try to ensure optimum animal performance and growth, good animal welfare and high standards of environmental performance (Watts & Davis, p. 2). For example, many commercial feedlots such as Beef City (Aubigny, Queensland), have a range of pen sizes from 50 head to 300 head (Watts & Davis, p. 5). However, this means that tracking cattle numbers around the feedlot can be

extremely challenging, and this sometimes results in the stopping of all operations on site, in that area, for large parts of a day.



**Figure 2.1 Beef City Feedlot – Aerial View and Production Pen View**

In traditional cattle feedlots, experienced pen riders using their own horses are responsible for receiving, drafting, processing, and moving cattle daily to ensure the safety and welfare of all animals. Pen riders are also responsible for maintaining daily records of health treatments and data entry related to the number of cattle in pens (Stockyard Beef). However, this process is costly and time-consuming as this type of approach requires experienced pen riders. This type of approach to tracking cattle can also cause the cattle extra stress due to being ‘pushed around’ in small pens by the handlers, resulting in the cattle feeding less. In contrast, contemporary feedlots over the past five years have started to switch to drones for collecting aerial images of cattle numbers. While this automated aerial process is still in its

infancy, the counting of cattle numbers with drones has been shown to save time, lower labour costs, and reduces overall production costs associated with cattle growth and development (Alanezi et al., 2022, p. 45002; Barbedo et al., 2020, p. 1; Crop Quest; Rivas et al., 2018, p. 2). While collecting aerial imagery of cattle numbers in feedlots appears to provide a simple solution to lowering overall production costs, issues in counting from aerial images using UAVs and automated AI systems can arise when the cattle are standing close together in pens or moving/changing places while the drone images are being collected.

## **2.2 UNMANNED AERIAL VEHICLES (UAVs)**

There are several prominent UAVs, popularly referred to as drones, that can serve to reduce the repetitive, mundane task of counting cattle numbers (Ortega et al., 2023, p. 2). UAVs in both civil and industrial applications are extremely popular due to their simplicity of deployment, low maintenance costs, high mobility, and ability to hover (Ganesh & Gudipalli, 2024, p. 637). Unmanned aerial vehicles that do not require an onboard pilot are also functional for data collection and reconnaissance in remote areas or environments that may be dangerous for human habitation.

The operation of UAVs can be broadly divided into one of two different categories. The first category, termed 'autonomous' relies on Global Positioning System (GPS) technology to determine the UAVs precise location and navigate to specific waypoints, or a pre-programmed flight path set by the operator. The benefit of this category of drones is their high precision of accuracy (if needed) in the flight path taken by the UAV and the reduction in need for human interaction. However, this 'autonomy' in operation from the UAV presents an increased risk in safety in areas such as collision avoidance. This can occur when the UAV fails to recognise an oncoming obstacle and change its flight path. The second category of drone operation termed 'remotely piloted', as the name suggests, relies on a

human operator to monitor the UAVs status, adjust the flight path when required and control features such as the operation of an onboard camera (Garg, 2021).

The other feature of UAVs is that they exist as either fixed-wing or rotary models. As illustrated in Figure 2.2, the choice between employing a fixed-wing UAV or a rotary-wing UAV hinges on the specific application and intended use, as each type of UAV model possesses distinct advantages and drawbacks (Garg, 2022, p. 1).



**Figure 2.2 UAV Types (Australian Aerial Imagery, 2023)**

Fixed-wing UAVs operate in the same manner to conventional airplanes, generating lift as the forward thrust pushes the wings through the air. This makes them ideal for purposes necessitating long-distance coverage or extended flight durations (Lee et al., 2021, p. 1). Conversely, rotary-wing UAVs generate lift by a propeller's rotation (commonly a rotary-wing UAV will have up to 8 propellers in total), creating a downward air push that lifts the UAV against gravity. By modulating the rotation speed of one or multiple of these propellers, it becomes possible to manipulate the UAVs direction, velocity, and altitude or to hover in one place. This inherent combination of having high agility and being able to

hover in place for prolonged periods is an extremely valuable feature when stable image capture is a project requirement (Garg, 2022, p. 1).

As noted in Figure 2.2.1, Alanezi et al., (2022), highlighted that multirotor UAVs, specifically quadcopters, were the most popular UAV type used for livestock detection.

Year of publication	Livestock Type	UAV Type	Country	Method	Reference
2016	Cattle	3DR IRIS+	USA	Count manually from captured video	[9]
2017	Cattle	Custom-made	UK	Python <code>astropy</code> library to detect thermal image	[11]
2017	Cattle	DJI Inspire M1	UK	Faster-R-CNN with VGG-CNN M-1024	[12]
2018	Cattle	Multirotor	Spain	Apply sliding window on images, use CNN to check whether each window is livestock or not	[10]
2018	Sheep	Information not provided	New Zealand	R-CNN	[10]
2019	Cattle	SenseFly eBee	USA	DisCountNet with SfM	[48]
2019	Cattle	DJI Phantom 4	Japan	YOLOv2 with VisualSfM	[49]
2019	Sheep	Information not provided	UK	Single-Shot Multibox Detector	[26]
2019	Cattle	DJI Matrice 100	UK	YOLOv2	[13]
2019	Cattle	DJI Phantom 4 Pro	Brazil	15 pretrained transfer-learning architectures	[50]
2019	Information not provided	Quadcopter	Information not provided	Segmentation using U-Net and Inception-V4	[51]
2020	Cattle	DJI Mavic Pro	Australia	Segmentation using Mask R-CNN with ResNet-101	[52]
2020	Cattle and Sheep	DJI Mavic Pro	Australia	Segmentation using Mask R-CNN with ResNet-101	[53]
2020	Cattle	DJI Mavic 2 Pro	Brazil	NASNet-Large	[54]
2020	Sheep	Quadcopter	Qatar	Image processing with morphological operators; YOLOv2	[8]
2020	Cattle	DJI Mavic 2 Pro	Brazil	Xception architecture	[55]
2021	Sheep	DJI Phantom 3 Pro	New Zealand	Segmentation using U-Net and custom CNN	[56]
2021	Cattle	DJI Mavic Pro	Brazil	Faster R-CNN with Inception-ResNetV2	[57]

**Figure 2.2.1 Articles on Livestock Detection Using UAVs**

While typically not as strong as UAVs with more rotors, quadcopters can still carry a variety of payloads, including high-resolution cameras, sensors, and payload packages, making them suitable for diverse missions (Ganesh & Gudipalli, 2024, p. 637). Moreover, the four rotors provide excellent stability, making quadcopters easier to control than UAVs with

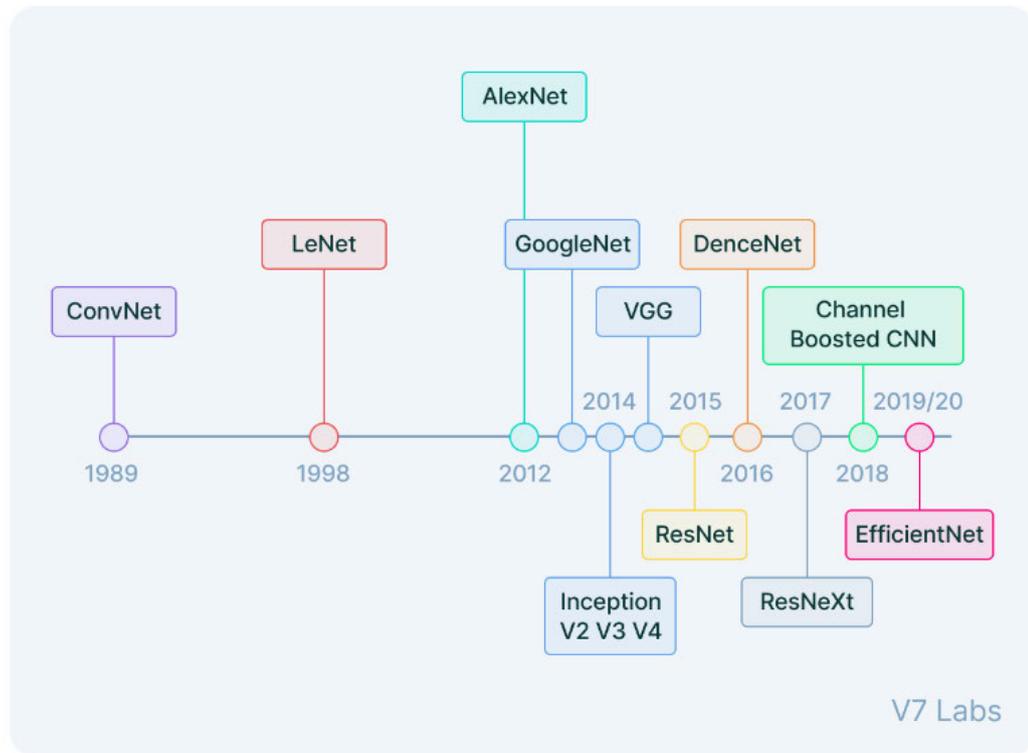
fewer rotors. This means that they can hover in place, perform smooth manoeuvres, and maintain stability even in windy conditions, making them ideal for livestock detection work (Gano et al., 2024, p. 3).

## **2.3 CNN ARCHITECTURE DEVELOPMENT**

While UAVs can complete the manual process of collecting remote imagery, state of the art CNNs have been developed over the past eight years to analyse the aerial information being collected by the high-resolution camera attachments (De Lima Weber et al., 2023, p. 100900; Xu et al., 2020, p. 105300). CNNs represent a subset of machine learning that features feed-forward neural networks that learn features via layers or filters (Xu et al., 2020, p. 2). Real-time mobile systems for cattle tracking, for instance, that combine the use of video captured from a UAV with CNNs have been recently used in investigations by authors such as Ortega et al., 2023 (A Proposal for Decentralized and Secured Data Collection from Unmanned Aerial Vehicles in Livestock Monitoring with Blockchain and IPFS) and Yousefi et al., 2022 (Review on the Use of Deep Learning in Precision Livestock Detection and Localization Using Unmanned Aerial Vehicles). Moreover, faster CNNs that combine computer vision with advances in artificial intelligence algorithms “have shown their great potential in object detection and classification of thousands of global images due to higher accuracy, precision and a quicker processing speed” (Xu et al., 2020, p. 2; Yousefi et al., 2022, p.1).

CNN development was also designed to address specific challenges in image processing and computer vision, making CNNs highly versatile and effective for a wide range of applications. For example, Standard CNN architectures, such as LeNet and AlexNet, shown in Figure 2.3, were one of the original CNN models that consisted of basic convolutional

layers, pooling layers, and fully connected layers used for tasks like image classification, object detection, and image recognition (Suganuma et al., 2020, p. 142).



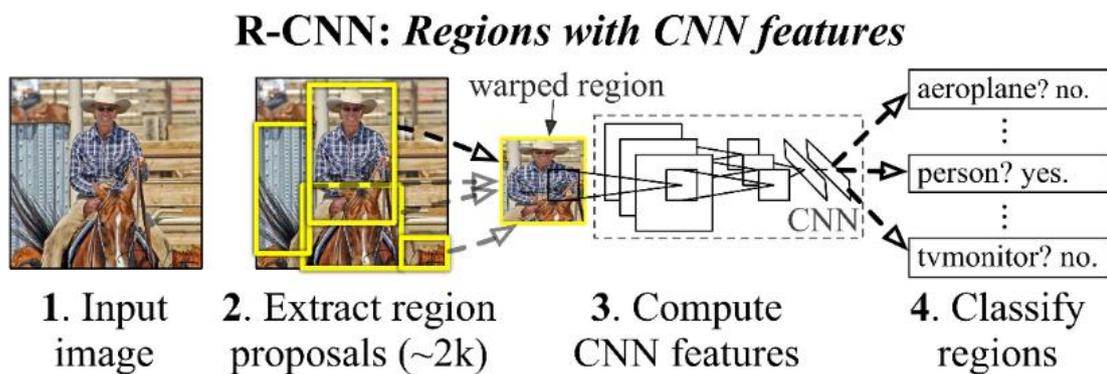
**Figure 2.3 CNN Development (v7labs.com/blog/convolutional-neural-networks-guide)**

Today, CNNs have continued to evolve in terms of their (1) enhanced computational efficiency, making them more viable on smaller devices; (2) advancements in processing 3-Dimensional data and complex time series; and (3) increased integration with other AI domains, such as reinforcement learning and unsupervised learning (Klinger, 2024, np).

## 2.4 MECHANISMS BEHIND CNN ARCHITECTURES

“CNNs unique ability to interpret visual data is pivotal in object detection, segmentation, video analysis, and real-time processing” (Klinger, 2024, np). Being able to pick out and detect patterns from images and text, and make sense of them, means that CNNs are highly applicable and widely used for detecting, identifying, and recognising different

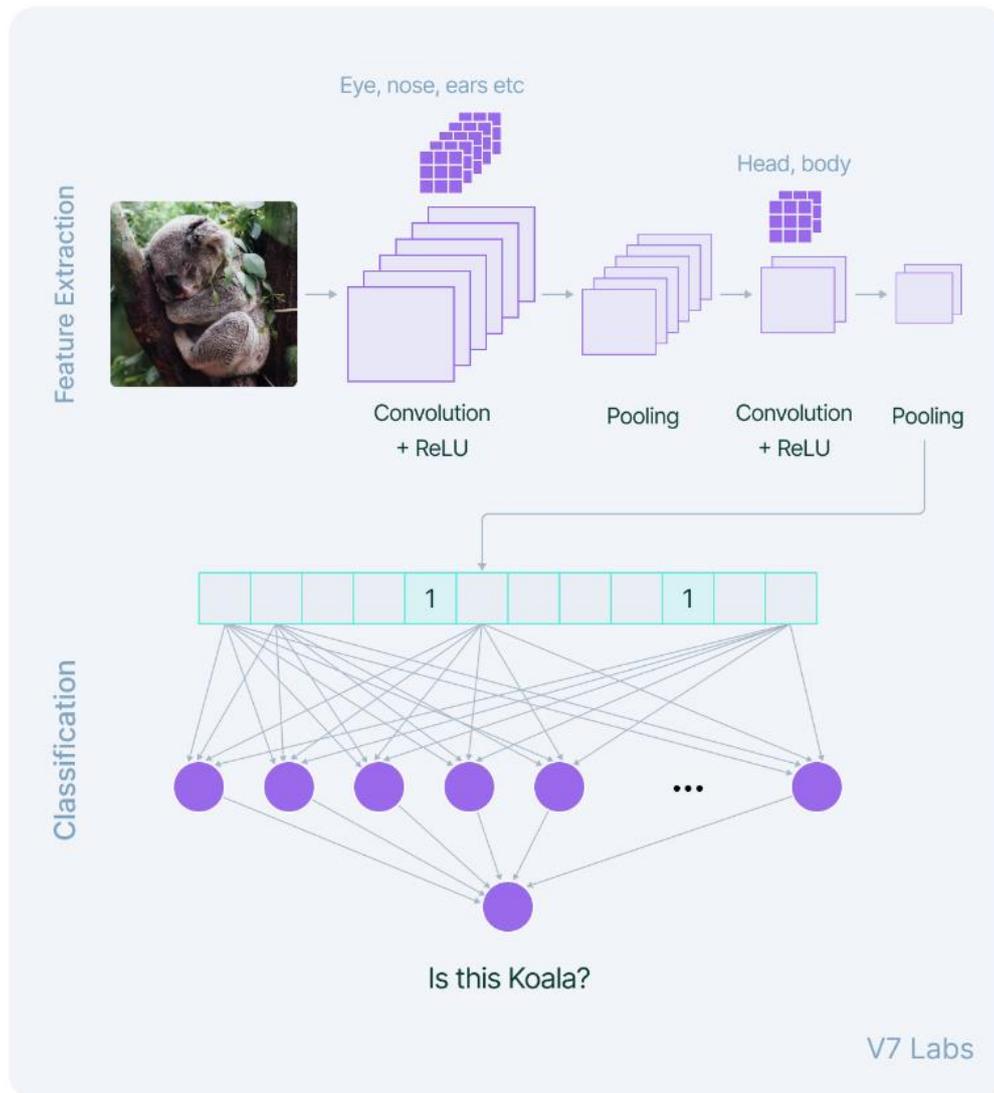
characteristics in remotely collected images of livestock such as cattle (Los et al., 2023, p. 100168; Mücher et al., 2022, p. 2378). However, regardless of what characteristics are required, or the type of AI model used, the mechanisms behind the CNN architecture processes for object detection essentially work the same. For example, the “Region-Based Convolutional Neural Network (R-CNN) architecture and its subsequent iterations, Fast R-CNN and Faster R-CNN, use a combination of selective searches to propose regions for classification, which significantly improves the accuracy and speed of object detection” (Klingler, 2024). Figure 2.4 illustrates a simple model that displays the search mechanisms used in a Fast R-CNN system.



**Figure 2.4 Fast R-CNN mechanisms (viso.ai/deep-learning)**

Essentially, the search task “involves dividing an image into segments to locate and understand objects at the pixel level” (Klinger, 2024, np). The computer, however, unlike human eyesight, views the input image as a matrix of numbers with (rows\*columns\*number of channels) shape. The CNN system works systematically by comparing the input images piece by piece. With the help of multiple filters, image details are initially taken care of by the first few convolutional layers (see Klinger, 2024; v7labs.com/blog/CNN-guide). However, as shown in Figure 2.4.1, the deeper the network goes, the more sophisticated the

pattern searching becomes to ensure that the correct image is being detected (Klingler, 2024).



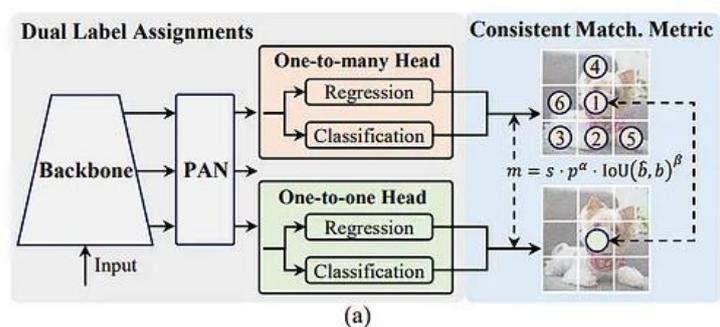
**Figure 2.4.1: Layers in a Fast R-CNN Search Process (v7labs.com/blog/CNN-guide)**

Furthermore, in the context of object detection using a CNN, the terms "one-stage" and "two-stage" detectors refers to different approaches for detecting and classifying objects in images. For example, choosing between a one-stage or two-stage detection model depends on the specific requirements of the image detection task; that is, whether the priority is speed (favouring one-stage models) or accuracy (favouring two-stage models) (Do Monte Lima et

al., 2016, p. 194; Yousefi et al., 2022, p. 80073). One-stage detector models, such as YOLO, SSD (Single Shot Multi-Box Detector) and Roboflow eliminate the need for regional proposals, as they complete a single pass to directly predict class probabilities and bounding box coordinates from the input image (Boesch, 2024). Notably, bounding boxes, described as a rectangular box that surrounds an object in an image, providing a visual and numerical way to identify and localise targeted objects within an image, are used in both "one-stage" and "two-stage" detectors to predict both the class of the object and the coordinates of the bounding box around it (Boesch, 2024). The multi-stage pipeline of two-stage detector models such as R-CNN, Fast R-CNN, and Faster R-CNN, are more complex than one-stage detectors as they divide the task of image detection into a region proposal (i.e., area likely to contain objects) and then a classification and refinement proposal, designed to improve the accuracy of the bounding boxes (Boesch, 2024).

Notably, machine vision technology has seen significant CNN advancements in recent years, particularly in the areas of object counting and tracking, with open-source "one-stage" and "two-stage" detector models such as YOLO, Python, Roboflow, and Mask R-CNN representing popular, readily available, open-source applications (Batchelor, 2012, p. 25; Boesch, 2024). The YOLO algorithm for example is a popular, single-shot object detection model designed for real-time object detection applications. YOLO algorithms, similar to other single-shot detector models, can be generally divided into two main sections: the backbone and head. The backbone of YOLO models is a CNN architecture known as 'Cross Stage Partial Networks Darknet53,' abbreviated as CSPDarknet53 (Ghosh, 2024). This terminology can be further broken down into three key processing components. 'Cross Stage Partial Networks' is a strategy designed to improve information flow and optimise feature extraction within the network. It achieves this by dividing the feature map of the initial layers

(representing an image in the context of object detection) into two distinct segments. These segments are independently processed within the network. When this parallel processing is complete, the features are merged at various points in the network, enhancing its ability to capture and utilise information. 'Darknet' serves as the foundational network and is responsible for capturing and extracting the features of an image. The '53' signifies the number of layers in the Darknet network (i.e., YOLOv4 Speed). The head of the YOLO model, often termed the 'detection layer' refers to the final layers of the neural network. This part is responsible for making object detection predictions. More specifically, the detection layer is responsible for bounding box predictions, class predictions, confidence scores, and non-maximum suppression (a post-processing step that removes redundant or low confidence bounding boxes). Since 2020, YOLO architectures have undergone several iterations with Version 10 (YOLOv10) now being publicly available. For instance, as shown in Figure 2.4.2, YOLOv10 combines the benefits of one-to-one and one-to-many matching strategies used during the training of object detection models, which leads to stronger supervision, resulting in higher accuracy and faster convergence during training (Ghosh, 2024)



**Figure 2.4.2 YOLOv10 Architecture (Ghosh, 2024)**

Machine vision systems such as YOLO, Python, Roboflow, and Mask R-CNN represent a significant leap forward in object detection, suggesting that they are all promising “methods

for farmers and industry to efficiently monitor their cattle herds in both remote and intensive feedlot production systems” (Mücher et al., 2022, p. 2377).

## **2.5 SOFTWARE DEVELOPMENT – AI ALGORITHMS**

In recent years, software development has grown exponentially due to AI tools, libraries and frameworks being freely available online (Suganuma et al., 2024, p. 142). For example, sites such as GitHub, PyTorch, TensorFlow, Roboflow and Ultralytics foster collaboration, transparency, and innovation while making AI technology more accessible, customisable, and affordable for anyone wanting to program machine learning models (Ghosh, 2024). Moreover, for object detection projects, these sites offer researchers the flexibility and efficiency required to tailor different tasks and use-cases, as they cover the entire lifecycle of the model from image data capture and model training to validation, deployment, and real-world tracking (Ultralytics, 2024). As illustrated in Figure 2.5, Roboflow guides novice users through how to build their vision model with what they term Quickstart methods; that is, building and deploying a basic model, uploading, managing, annotating and analysing images and training the model for your specific object detection application. Likewise, Figure 2.5.1, illustrates how Ultralytics adopts a similar approach by providing researchers with freely accessible YOLOv8 training examples and coding.

Notably, open-source sites readily allow access to their source codes and, in most cases, its free redistribution under certain conditions. However, these types of digital collaborative platforms are generally not governed by industry checks; that is, they rely on the trustworthy behaviour of those contributing new and/or revised code structures (Lapena-Manero et al., 2022, p. 918; Stephany et al., 2021, p. 1949). This therefore places emphasis on whomever is completing the research to have some understanding of the code structures involved to ensure that the project aims and objectives are being met. For example, in most cases,

accuracy metrics related to **precision** (i.e., the proportion of correctly identified objects out of all detected objects), **recall** (i.e., proportion of correctly identified objects out of all actual objects present in the image), **intersection over union (IoU)** which is a measure of how accurately the predicted bounding box matches the original “ground truth” bounding box, and **mean average precision**, are used to evaluate most object detection models, taking into account both precision and recall across different performance thresholds (Padilla et al., 2020, p. 238).



```
import roboflow

rf = roboflow.Roboflow(api_key='')

project = rf.project("ID")

# Upload an image to Roboflow

project.upload("example.jpeg")

# Run inference on an image

prediction = model.predict("example.jpeg")

# Convert predictions to JSON

json = prediction.json()
```

```
{
  "predictions": [
    {
      "x": 311.5,
      "y": 154,
      "width": 177,
      "height": 200,
      "confidence": 0.821,
      "class": "dog-beagle"
    }
  ]
}
```

**Figure 2.5 Roboflow Vision Model Coding Exemplar**(<https://docs.roboflow.com/>)



```
Single-GPU and CPU Training Example

Device is determined automatically. If a GPU is available then it will be used, otherwise training will start on CPU.

Python  CLI

from ultralytics import YOLO

# Load a model
model = YOLO("yolov8n.yaml") # build a new model from YAML
model = YOLO("yolov8n.pt") # load a pretrained model (recommended for training)
model = YOLO("yolov8n.yaml").load("yolov8n.pt") # build from YAML and transfer weights

# Train the model
results = model.train(data="coco8.yaml", epochs=100, imgsz=640)
```

**Figure 2.5.1 Ultralytics Training Exemplar** (Ultralytics, 2024)

## 2.6 STUDIES RELATED TO REMOTE DATA CAPTURE

In many countries around the world, the cattle industry has begun to investigate the use of UAVs and AI-driven object detection software systems to collect and analyse data related to cattle health and herd numbers. Indeed, over the past five years, animal detection and counting systems have been gradually growing in popularity, with most designs being based on small number of real-world applied studies. For example, supported by fast-growing developments in CNN systems, Barbedo & Koenigkan's (2018) study provided an early example of how to deal with clustered animal images and images that have similar colours to the environment being captured. Likewise, Rivas et al., (2018) investigation of how to create solutions for overlapping cattle images was also one of the first studies that used UAVs and CNN architectures to try and unravel issues related to the remote counting of cattle numbers. Unmanned aerial vehicle technologies have also been significant in the wildlife monitoring space for many years (Lou et al., 2024, p. 2). Recently, this same technology has been adopted for monitoring livestock such as sheep in rural and remote settings (Lou et al., 2024, p. 2; Rancic et al., 2023, p. 180; Sawar et al., 2021, p. 2). "Supported by high-resolution cameras, UAVs, for instance, now fly over livestock groups in order to capture images, making it possible to create computer vision systems to support the management of these animals" (De Lima Weber et al., 2023, p. 100901).

As shown in Figure 2.6, what we can learn from previous work is that "several state-of-the-art object detection approaches based on CNNs such as DenseNet, ResNet, and NASNet have begun to achieve reliable outcomes for wild animal and livestock recognition using UAV imagery" (Yousefi et al., 2022, p. 80079)

Application	Method	Backbone network	Type of livestock	Type of data	Environment	Platform	Altitude
Segmentation and body contour extraction	Mask R-CNN	FPN based ResNet101	Cattle	Image	Feedlot, Australia	Camera	-
Individual cattle unrecognizable boundary and body shape recognition and segmentation	Enhanced Mask R-CNN	ResNet101+RPN+Feature Pyramid Network (FPN)	Cattle	Image	Ranch, Nigeria	Camera	-
Detection and counting	FCN, CNN	Network-I, Network II, AlexNet, GoogLeNet, VGG16, VGG19, ResNet50, and U-Net	Sheep	Video	Farm, New Zealand	UAV- a DJI Phantom 3 Pro	80-120 m
Detection	CNN	VGG-16/VGG-19/ResNet-50 v2/ResNet-101 v2/ ResNet-152 v2/ MobileNet/MobileNet v2/ MobileNet 121/DenseNet 169/ DenseNet 201/Xception/ Inception v3/Inception ResNet v2/NASNet Mobile/NASNet Large	Cattle	Image	Farm, Brazil	UAV- a DJI Phantom 4 Pro	30 m
Detection	CNN	64x64-18C7-MP4-96C5-MP2-4800L-2	Cattle	Video	Farm, Brazil	Multi-rotor UAV	-
Visual localization and identification	R-CNN	VGG CNN M1024	Holstein Friesian cattle	Image Video	Farm, UK	Camera and DJI Inspire MKL	25 m
Detection	YOLOv2	AlexNet, GoogleLeNet, Inceptionv3	Holstein Friesian cattle	Video	Farm, UK	DJI Matrice 100 quadrotor	10 m

**Figure 2.6 Examples DL methods using UAV imagery (Yousefi et al., p. 80080)**

Moreover, “in the last decade, researchers have focused more on deep CNNs than other machine learning algorithms for object detection, localisation, classification and segmentation” (see Sarwar et al., 2021, p. 1; Yousefi et al., 2022). The reason behind this change of approach is that CNNs adopt the process of using bounding boxes as the “ground truth data”, meaning that object detection accuracy becomes much higher when analysing aerial images. For example, several recent livestock studies (e.g., Lo et al., 2023; Mütcher et al., 2022; Symeonaki et al., 2024) highlight how the availability of open-source coding software, particularly the YOLO algorithm, has driven object detection research in the last few years. Tabernik et al., (2024) further stress how computer vision-based object counting processes using probabilistic techniques, where the prediction of an object’s centre is done by directly learning the probability for every pixel in an image that belongs to the object centre, greatly reduces the intense labour requirements of manual bounding box annotations

(p. 2). However, as highlighted by Yousefi et al., (2022), this type of approach to counting livestock is still not well developed, with the majority of studies involving cattle counting using a UAV (e.g., Barbedo et al., 2020; Li & Xing, 2019; Xu et al., 2020; Yang et al., 2021), still possessing issues related to the identification of small objects, generalizing visual feature ratios, capturing clear, well-established datasets and the time-consuming, labour-intensive nature of precise image annotations.

## **CHAPTER 3 METHODOLOGY**

This section introduces the research hardware, planned tests, study area and image collection protocols. As noted earlier, recent advances in AI-based approaches (see Yousefi et al., 2022), underline how this project plans to use UAV imagery to compare five different custom trained object detection models as the basis for detecting and counting cattle in feedlot settings (Lou et al., 2024, p. 2; Wang et al., 2023, p. 1594).

### **3.1 STUDY AREA AND OBJECTS**

The study area selected was situated near the township of Dalby, 50km west of Toowoomba and 200km west of Brisbane. Currently, the farm location accommodates crossbred steers, termed feedlot steers, which are mostly Angus but also Angus cross and Shorthorn steers. These cattle are maintained on grass or grain-supplied feedlot pens; usually fed for 150-200 days to produce a 300-400kg carcass with some marbling (Andrews, 2015, p. 3.) Consequently, this location serves as an ideal setting for capturing aerial UAV images of cattle to ascertain whether an AI-driven object detection system will work.

On 2<sup>nd</sup> August 2024, the author and research supervisor conducted aerial photography at the Dalby site. A total of 175 aerial photographs were completed over altitudes of 40 meters, 30 meters and 20 meters for sampling over a range of different pens of similar size. The camera

was positioned underneath the UAV. For 150 images, the camera was angled at 90 degrees. For 25 images, the camera was angled at 15 degrees. Feedlot steers were selected as the research subject, as depicted in Fig 3.1. Feedlot steers served as the object for detection as they possess a range of black, white, tan and brown animal top-line colours which will provide a suitable test whether annotated images of different colour can be accurately detected



**Figure 3.1 Feedlot Cattle (Andrews, 2015)**

## **3.2 UAV HARDWARE - OVERVIEW**

To collect data within the selected area, a Mavic 2 P600 type intelligent UAV (manufactured by Chengdu Bobei Technology Co., Ltd., China) was utilised. Specific specifications regarding the UAV used are presented in Table 3.2. The P600 UAV serves as a robust research platform, featuring a substantial payload capacity, extended endurance, and scalability. Equipped with pods; that is, two-dimensional planar Lidar and GPS, the Mavic 2 P600 type intelligent UAV affords operational functions such as pod selection and tracking, LiDAR obstacle avoidance, accurate UAV positioning, and speed-guided flight.

This type of advanced equipment allows for the capture of real-time images through the airborne computer, achieving image resolutions as fine as 5 cm.

**Table 3.2: Technical Specifications - Mavic 2 P600 UAV:**

Sensor	1" CMOS Effective Pixels: 20 million
Lens	FOV: about 77° 35 mm Format Equivalent: 28 mm Aperture: f/2.8–f/11 Shooting Range: 1 m to ∞
ISO Range	Video: - 100-6400; Photo: 100-3200 (auto) - 100-12800 (manual)
Shutter Speed	Electronic Shutter: 8–1/8000s
Still Image Size	5472×3648
Takeoff Weight	Mavic 2 Pro: 907 g Mavic 2 Zoom: 905g
Dimensions	Folded: 214×91×84 mm (length×width×height) Unfolded: 322×242×84 mm (length×width×height)
Max Speed (near sea level, no wind)	72 kph (S-mode)
Maximum Takeoff Altitude	6000 m
Max Flight Time (no wind)	31 minutes (at a consistent 25 kph)
Max Hovering Time (no wind)	29 minutes

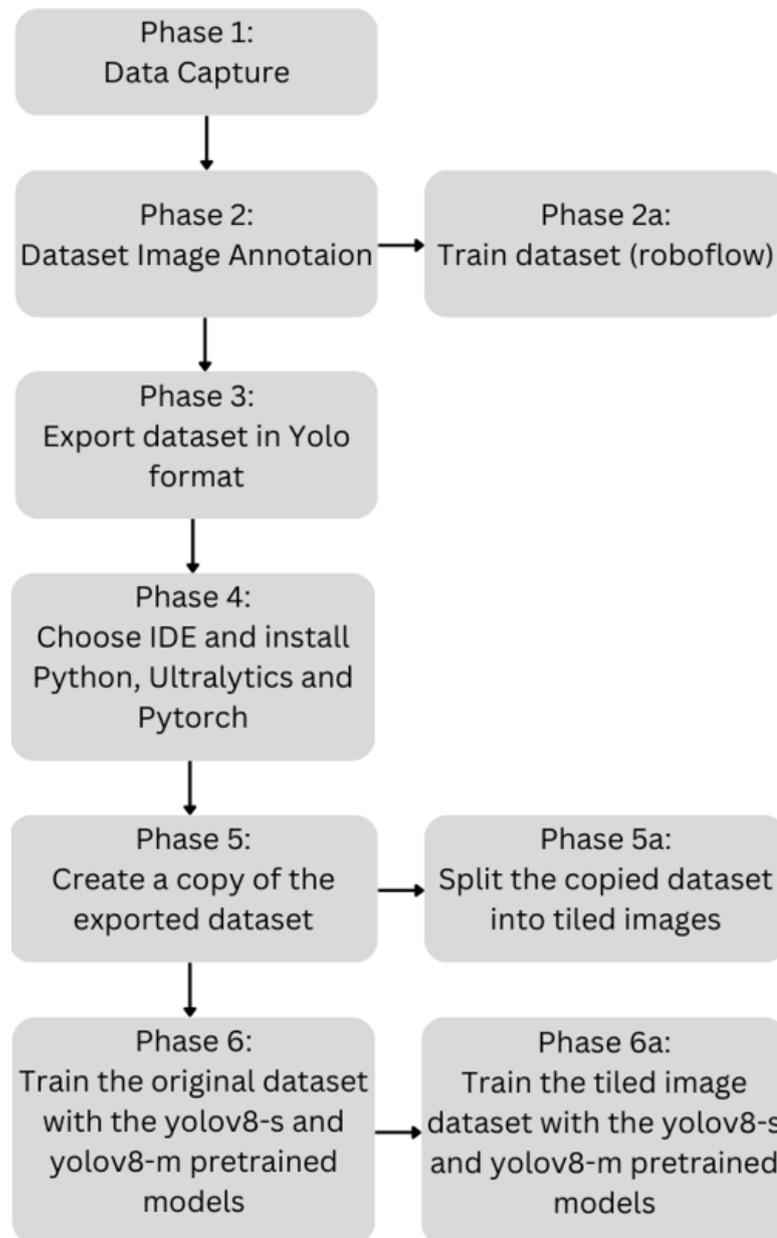
Figure 3.2A and Figure 3.2B illustrate the study site, captured from 40m, at an angle of 15 degrees, using the Mavic 2 P600 type intelligent UAV



**Figures 3.2.1 Aerial UAV Images Captured from the Dalby Study Site**

### 3.3 IMAGE DETECTION WORKFLOW

The overall project workflow is illustrated in Figure 3.3. Following data capture using the Mavic 2 P600 UAV, an image/object detection workflow was established to begin annotating the images captured in order to custom train the models chosen.



**Figure 3.3 Project Workflow Design**

**Phase 1** involves creating a dataset of images for fine-tuning the pre-trained object detection models. In Figure 3.3.1, the goal of this phase is to gather images that accurately represent objects of interest (e.g., cattle) in different environments (feedlot and open fields) and from various heights (40m, 30m, 20m), which will be used to train and test the different models.



**Figure 3.3.1 UAV Captured Image for Project Dataset**

**Phase 2** involves the annotation of the dataset images. This step was completed in the Roboflow browser. Annotation is the process of labelling objects within the images by either drawing bounding boxes around them or using pixel-level segmentation, depending on the detection task. Each object is then assigned a class, such as "cattle." This project phase is crucial because the accuracy of the annotations directly impacts the performance of the object detection model. Figure 3.3.2 illustrates the same sample UAV image after being

annotated in Roboflow. Once the image annotations are complete, the dataset can then be trained using Roboflow, meaning the use of a machine learning model to identify and locate objects within images. The goal is to provide the specific location of each object.



**Figure 3.3.2 Annotated Dataset Image**

**Phase 3** involves exporting the annotated dataset in YOLO format. In YOLO format, each image annotation is saved as a text file where each line contains [class, x-coordinate, y-coordinate, width, height] values for every object. As shown in Figure 3.3.3, these values are normalised to be between 0 and 1, representing the object's location and size relative to the image dimensions.

0	0.3293196786334435	0.13827954876427626	0.04157330961663119	0.0739722934316077
0	0.42041936934140267	0.06688941304506721	0.05081073160946964	0.03797169868665454
0	0.4892578122431507	0.029513888966181567	0.038366868542409616	0.06437880534750887

**Figure 3.3.3 YOLO Txt Format: [class, x-coordinate, y-coordinate, width, height]**

**Phase 4** focuses on setting up the “development environment”. This involves installing an Integrated Development Environment (IDE), along with Python, Ultralytics, and PyTorch. “Python is an interpreted, object-oriented, high-level programming language with dynamic semantics” (Python, 2024) It is an ideal language for both beginners and experienced developers in the object detection space as it provides flexibility in how to design the detection program. Ultralytics, an open-source platform that primarily focuses on creating efficient machine learning models and tools using YOLO for real-time object detection and related tasks (Ultralytics, 2024), into the Python terminal. Pytorch, developed by Facebook's AI Research (FAIR) lab, is an open-source deep learning framework primarily used for developing and training machine learning models, known for its flexibility and ease of use (Pytorch, 2024).

**Phase 5 and Phase 5a** involve making copy of the original dataset to create a new “Tiled image dataset”. The “tiled image dataset”, shown in Figure 3.3.4, is a copy of the original dataset that has divided the high-resolution images (originally 4056x3040 pixels) into smaller 640x640 pixel tiles. According to Zhao & Shi (2022), employing a tiling process offers several benefits, including improved training accuracy, reduced computational load, and better object localisation (p. 7087). For example, by focusing on smaller regions of an image, a tiled model can potentially detect smaller objects more effectively while also reducing the memory and processing power needed during training. When using this method for identifying pests such as the flea beetle in vegetable crops grown in crowded greenhouse environments, Zhao & Shi (2022) noted that an image tiling processes effectively dealt with the problem of small object size and high image resolution (p. 7100). Moreover, “in the case of wide scene and high resolution, image tiling [should be able to] solve the limitation of devices and further tap the potential of deep learning models” (Zhao & Shi, 2022, p. 7100).



**Figure 3.3.4 “Tiled 640x640 Image” - Created from Original 4056x3040 Image**

**Phase 6 and Phase 6a** include training both the original and tiled datasets using pre-trained YOLOv8 models - YOLOv8-s (i.e., small) and YOLOv8-m (i.e., medium). These different models differ in size and complexity: YOLOv8-s is a smaller, faster model designed for real-time applications where speed is crucial, whereas YOLOv8-m is a medium-sized model that offers improved accuracy at the cost of slightly slower inference times. Both models, as shown in Figure 3.3.5, were fine-tuned on the custom dataset to evaluate their performance in terms of object detection accuracy and cattle counting.

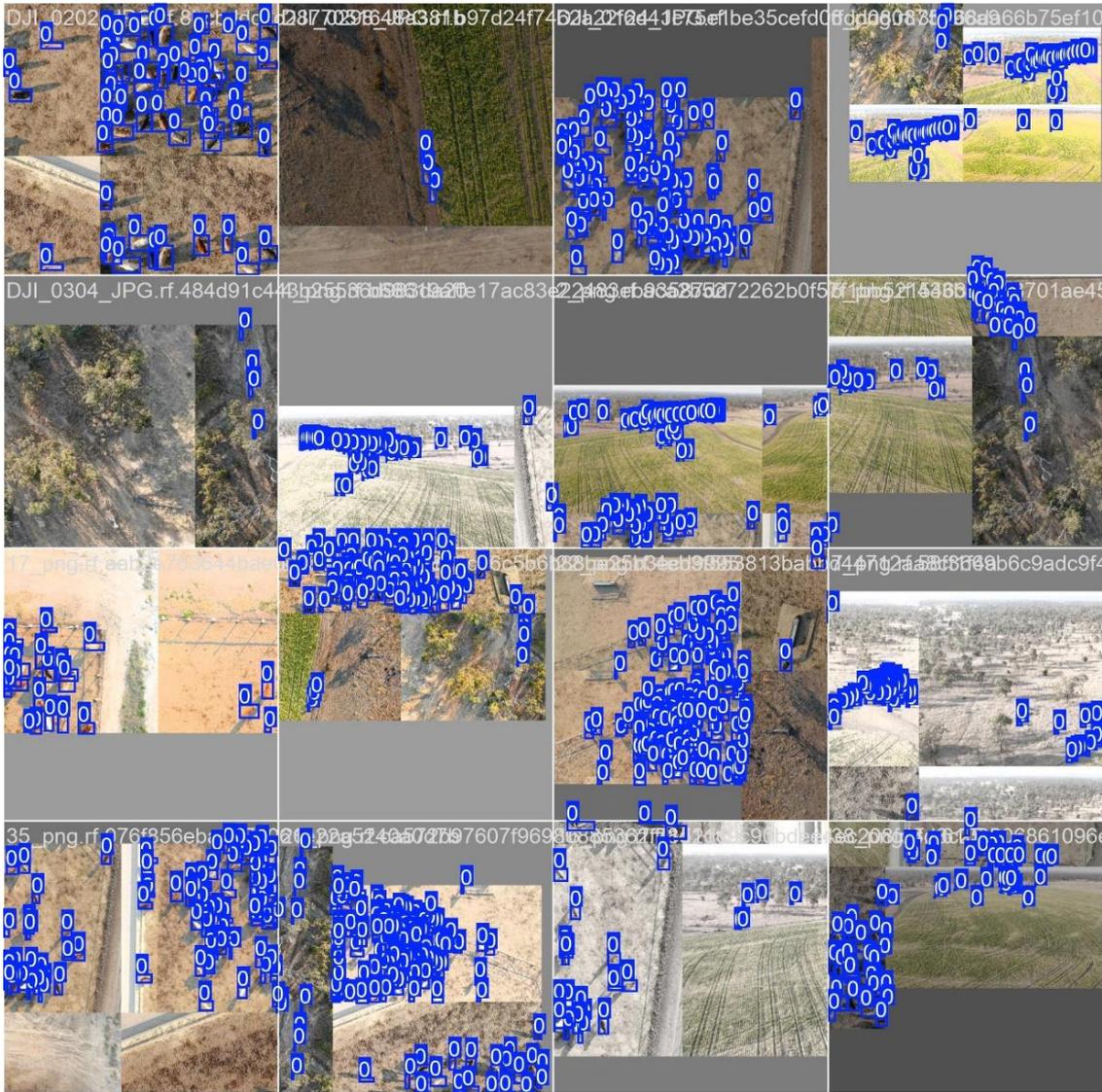


Figure 3.3.5 Trained Image ('0' represents a class of object detected i.e., cattle)

## CHAPTER 4 MODEL DEVELOPMENT

Five different pre-trained machine learning models were fine-tuned to recognise images of individual 'cattle' from the dataset. Table 3.4 displays the name of the different models and which pre-trained machine learning algorithm it was trained on.

**Table 4.1 Project Training Models**

<b>Model Label:</b>	<b>Model Name:</b>	<b>Trained On:</b>
A	RoboFlow_custom	AutoML solution
B	YOLOv8-s_full	YOLOv8-s.pt
C	YOLOv8-s_tiled	YOLOv8-s.pt
D	YOLOv8-m_full	YOLOv8-m.pt
E	YOLOv8-m_tiled	YOLOv8-m.pt

YOLOv8-small (YOLOv8-s) and YOLOv8-medium (YOLOv8-m) were used to compare the accuracy of different pre-trained object detection models. Roboflow was also used as an alternative comparison model. Additionally, the YOLO models used full and a tiled image datasets for comparison. During the training process, precautions were also taken with overfitting to make sure that the model performed well with both the training data and new, unseen data images.

#### **4.1 PERFORMANCE METRICS**

Evaluating the effectiveness of an object detection model relies on performance metrics that quantify its ability to accurately detect, locate, and classify objects within an image. These metrics provide insights into both the object detection model's strengths and areas for further improvement. The following two sub-sections outline the software development processes and performance metrics used to achieve the projects objectives.

### 4.1.1 OBJECT DETECTION MODEL OUTPUT

An object detection model's outputs includes bounding boxes around detected objects, characterised by three main attributes:

1. **Object Class:** The category assigned to the detected object (e.g 'cattle').
2. **Bounding Box:** A rectangular boundary around the detected object, indicating its location.
3. **Confidence Score:** A probability score (0 to 1) reflecting the model's confidence in the prediction.

### 4.1.2 CLASSIFICATION AND EVALUATION METRICS

Evaluating the performance of an object detection model involves comparing bounding boxes outputted from the model against ground-truth bounding boxes, which represent the actual object locations of objects. Each comparison is then categorised into one of the following classification metrics, based on the correctness of the prediction:

1. **True Positives (TP):** The model correctly predicts the location and identifies an object.
2. **True Negative (TN):** The model correctly predicts that there is no object present.
3. **False Positive (FP):** The model predicts an object is present when there is not one present.
4. **False Negative (FN):** The model fails to predict an object that is present.

These classifications enable the calculation of metrics that quantify the model's detection quality. The key metrics used in this project's evaluation include:

- 1. Precision (P):** Precision indicates how many of the object detection model's predictions are accurate, reflecting its reliability in identifying objects.

**Table 4.1.2A Equation for Precision**

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

- 2. Recall (R):** Recall indicates the object detection model's ability to detect all instances of the object.

**Table 4.1.2B Equation for Recall**

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

- 3. Mean Average-Precision:** mAP indicates the object detection model's accuracy in detecting and classifying objects. It is calculated as the average-precision across all classes and a range of recall levels.

**Table 4.1.2C Equation for Average-Precision and Mean Average-Precision**

Metric	Equation	Description
Average Precision (AP)	$= \sum_n (R_n - R_{n-1})P_n$	Measures a model's accuracy in predicting objects, averaged over various recall levels.
Mean Average Precision (mAP)	$= \frac{1}{N} \sum_{i=1}^N AP_i$	

## **CHAPTER 5 RESULTS AND DISCUSSION**

### **5.1 ROBOFLOW\_CUSTOM MODEL**

The RoboFlow\_custom model, trained on the full-image dataset, achieved an impressive mAP of 96.5%, with a precision of 96.0% and recall of 91.0%. These metrics demonstrate the model's robust detection capabilities, reflecting its ability to accurately identify cattle in high-resolution UAV images. The high precision indicates that the model is highly accurate in detecting true positives, minimizing false positives. While the recall is also strong at 91.0%, it lags slightly behind precision. This suggests that while the RoboFlow\_custom model is effective at correctly identifying cattle, it may miss a small proportion of instances, potentially due to factors such as occlusion, variations in cattle appearance, or challenges presented by overlapping objects in the feedlot environment.

### **5.2 YOLOV8-S\_FULL MODEL**

The YOLOv8-s\_full model, trained on the full-image dataset, achieved a mAP of 94.9%, with a precision of 96.0% and a recall of 88.4%. While the precision is comparable to the RoboFlow\_custom and YOLOv8-m\_full models, the lower recall indicates that the YOLOv8-s\_full model struggles to identify all cattle instances. The drop in recall could be attributed to the model's smaller architecture compared to that of the YOLOv8-m.pt models, which may limit its capacity to generalize across the full spectrum of cattle appearances or varying image conditions. This outcome reflects the typical trade off seen with smaller models, which tend to prioritise inference speed over accuracy (Ultralytics Inc, 2024).

### **5.3 YOLOV8-S\_TILED MODEL**

The YOLOv8-s\_tiled model, trained on the tiled dataset, achieved a higher mAP of 96.0%, with a precision of 95.0% and a recall of 89.7%. Compared to its full-image counterpart

(YOLOv8-s\_full), the tiled approach slightly improved recall from 88.4% to 89.7%. This suggests that tiling each image did enhance the model's ability to detect cattle in various positions within an image, likely due to the increased focus on smaller portions of the image, allowing the model to better capture finer details. However, it remains slightly behind the RoboFlow\_custom model in terms of overall performance.

## **5.4 YOLOV8-M\_FULL MODEL**

The YOLOv8-m\_full model, fine-tuned on the full-image dataset, outperformed the RoboFlow\_custom model, achieving a slightly higher mAP of 96.7%, with a precision of 95.9% and a recall of 93.2%. The increase in mAP and recall highlights YOLOv8-m\_full's superior ability to detect individual cattle across the feedlot. The model's larger architecture likely contributes to its enhanced feature extraction capability, allowing it to identify more cattle instances, particularly in complex environments. The marginally lower precision (95.9% compared to 96.0% for RoboFlow\_custom) is outweighed by the significant improvement in recall, indicating that YOLOv8-m\_full is better equipped to minimise missed detections.

## **5.5 YOLOV8-M\_TILED MODEL**

The YOLOv8-m\_tiled model, trained on the tiled dataset, achieved a mAP of 96.3%, with a precision of 94.7% and a recall of 90.5%. While these results remain competitive, they reflect a slight reduction in both mAP and recall compared to the full-image variant, YOLOv8-m\_full. The primary challenge introduced by the tiling approach lies in the handling of image boundaries. When images are split into tiles, objects (in this case, cattle) positioned near the edges of the tiles may become fragmented or partially visible, making them more difficult for the model to detect (Plainsight, 2022). This fragmentation can lead

to missed detections, which is likely why the recall is lower compared to the full-image model.

**Table 5.1 Precision, Recall, and mAP metrics after training**

<b>Model Name</b>	<b>Precision (%)</b>	<b>Recall (%)</b>	<b>mAP (%)</b>
RoboFlow_custom	96.0	91.0	96.5
YOLOv8-s_full	96.0	88.4	94.9
YOLOv8-s_tiled	95.0	89.7	96.0
YOLOv8-m_full	95.9	93.2	96.7
YOLOv8-m_tiled	94.7	90.5	96.3

Training and validation results for all models are included in Appendix A.

## **5.7 COMPARISION OF DETECTION PROCESSES & MODELS**

The YOLOv8-m.pt models consistently outperformed YOLOv8-s.pt model and the RoboFlow Auto ML model in the areas of precision, mAP, and recall metrics. This highlighted that a CNNs parameters, such as the number of network layers, directly impacts the detection accuracy in complex datasets (Ultralytics Inc, 2024). This finding supports earlier AI-driven object detection studies (e.g., Barbedo & Koenigkan’s (2018); Rivas et al., (2018)), which found that larger models, such as YOLOv8-m.pt, are particularly effective for high density object detection. The results of this project also support Terven & Cordova-Esparza (2024) findings that larger model architectures excel in high variability contexts by handling intricate object features and variations with greater ease. While the YOLOv8-m.pt

models were found to be capable of discerning subtle differences between overlapping or partially obscured objects, the project also found that appropriate threshold selection was crucial for the performance evaluation of cattle detection; that is, as highlighted previously by Yousefi et al., (2022), systems will be weak if a threshold is incorrectly selected resulting in overlapping bounding boxes and missing objects. Furthermore, similar to Tabernik et al., (2024) results that highlighted how computer vision-based object counting processes using probabilistic techniques could reduce the labour requirements of manual bounding box annotations, this project found that the YOLOv8-m.pt models had the ability to discern subtle differences between overlapping or partially obscured objects. However, while discerning subtle differences in objects is crucial in environments such as cattle feedlots where livestock is often densely clustered or partially occluded, the amount of time it takes to train an object detection model is still significant. This supports earlier research findings (e.g., Barbedo & Koenigkan, 2018; Youssefi et al., 2022) that stressed how using different classification techniques for object detection, using applied deep object detection methods, required considerable modifications to detect small objects such as livestock.

The impact of dataset tiling on model performance showed both improvements and drawbacks. Both the yolov8-s.pt models and yolov8-m.pt models experienced slight declines in mAP and recall when trained on the tiled dataset, aligning with previous observations from Plainsight (2022), that tiling, while computationally efficient and potentially better at detecting small objects, can disrupt spatial coherence and hinder object recognition across tiled images. Notably, the YOLOv8-s\_tiled model did show a slight improvement in recall when trained on tiled images, suggesting that similar to Zhao & Shi (2022), tiling can offer benefits for smaller models by exposing them to more varied object positions and perspectives.

The precision-recall trade-off observed across all models highlights that every CNN model used possesses architectural strengths and limitations. For example, while Mücher et al., (2022) and Symeonaki et al., (2024) emphasised the availability of open-source coding software for object detection, highlighting how YOLO has strongly driven object detection processes and research in the last few years, this project underlined how using open-source coding for closely confined livestock was not well developed (Barbedo et al., 2020; Li & Xing, 2019). Indeed, similar to Yang et al., 2021, the process of dividing images into segments to locate and understand objects at the pixel level still requires further research work. Moreover, while all models maintained high precision, their recall scores varied, particularly the YOLOv8-s\_full and YOLOv8-s\_tiled models, which achieved strong detection accuracy results but missed a larger proportion of cattle instances compared to the YOLOv8-m\_full, YOLOv8-m\_tiled and RoboFlow\_custom models. Similar to Mücher et al., (2022) findings, this trade-off reflects the compromise between model size and performance consistency.

## **5.8 IMPACT OF UAV FLIGHT ALTITUDE ON DETECTION ACCURACY**

Project Objective 3 sought to identify the most effective UAV flight level and pattern for accurate cattle detection. Images were captured at 40m, 30m, and 20m, providing a range of perspectives for evaluating model performance across various altitudes. However, results indicated no significant differences in mAP, precision, or recall among the flight heights for all models. This suggests that within the 20-40m range, each model's ability to detect and count cattle remained consistent, implying robustness to moderate changes in image resolution and perspective. The stability of detection accuracy across these altitudes suggests

that operators have flexibility when choosing flight levels, making it feasible to optimise for other factors like battery life and flight coverage without compromising model performance.

## **CHAPTER 6 CONCLUSION**

The results of this project provide promising ideas and opportunities for livestock industry applications. First, the project successfully designed and tested an AI-driven, object detection system based on aerial images of livestock captured from a UAV fitted with a high-resolution camera. Second, all five pre-trained machine learning models were capable of recognising ‘cattle’ from the captured aerial images. However, reliable detection metrics were stronger in certain models over others; that is, the YOLOv8-m\_full model achieved the highest mAP among the tested models, indicating its strength in overall detection accuracy, though its precision and recall were slightly lower than other models in certain areas. These findings highlight those larger architectures, like YOLOv8-m.pt, may better capture complex object variations. Third, the project presents a viable, cost-effective system for the autonomous monitoring (i.e., detect, count, identify) of cattle in intensive feedlot settings which can reduce the amount of manual labour, and associated costs, currently required to monitor cattle in extensive production environments.

However, there are several recommendations for further advancing object detection models in livestock management. For example, the computing power required to run advanced object detection CNNs, including the safe storage of the captured video data is large, demanding specialist computer software. Secondly, the open-source CNNs require a significant amount of training process time, meaning that having to take precautions to ensure that overfitting did not take place, also substantially adds time to training, checking, and (re)training the model over several iterations to make sure it can accurately detect,

locate, and classify objects within images. These iterative processes therefore add to the costs of selecting the right model for each different environment.

## REFERENCES

- Alanezi, M.A., Shahriar, M.S., Hasan, M.B., Ahmed, S., Sha'aban, Y.A. & Bouchekara, H.R.E.H. 2022, 'Livestock Management With Unmanned Aerial Vehicles: A Review', *IEEE Access*, vol. 10, pp. 45001-45028.
- Andrews, T. 2015, Market specifications for cattle, NSW Department of Primary Industries, viewed viewed 5 September 2024, <[https://www.dpi.nsw.gov.au/\\_\\_data/assets/pdf\\_file/0005/148415/Market-specifications-for-cattle.pdf](https://www.dpi.nsw.gov.au/__data/assets/pdf_file/0005/148415/Market-specifications-for-cattle.pdf), pp. 1-5.>
- Australian Aerial Imagery, 2023. *Different Types of UAVs* viewed 9 October 2023, <<https://www.aaisydney.com.au/australian-drone>>.
- Australian Lot Feeders' Association 2012, *Australian National Guidelines for Beef Cattle Feedlots*, viewed 7 October 2023, < Australian Lot Feeders' Association | Grain Fed Beef ([feedlots.com.au](http://feedlots.com.au))>.
- Barbedo, J.G.A., Koenigkan, L.V., Santos, P.M. & Ribeiro, A.R.B. 2020, 'Counting cattle in UAV images-dealing with clustered animals and animal/background contrast changes', *Sensors (Basel, Switzerland)*, vol. 20, no. 7, p. 2126.
- Batchelor, B.G. (Ed.) 2012, *Machine Vision Handbook*, 1st ed. 2012, Springer London.
- V7 Labs, *YOLO: Algorithm for object detection explained + examples*, viewed 12 October 2023, <<https://www.v7labs.com/blog/yolo-object-detection>>.
- Boesch, viso.ai, 2024, FCOS: Fully Convolution One-Stage Object Detection, viewed 5 September 2024, < <https://viso.ai/deep-learning/fcos-object-detection/>>
- Built In, 2023, *What Is Python for Machine Learning? (Definition, Uses)*, viewed 12 October 2023, <<https://builtin.com/machine-learning/python-machine-learning>>.
- Chai, Y. 2020, 'In-sensor computing for machine vision', *Nature*, vol. 579, no. 7797, pp. 32-33.
- Crop Quest, *Recent Ag News*, viewed 10 October 2023, <<https://www.cropquest.com/>>.
- De Lima Weber, F., de Moraes Weber, V.A., de Moraes, P.H., Matsubara, E.T., Paiva, D.M.B., de Nadai Bonin Gomes, M., de Oliveira, L.O.F., de Medeiros, S.R. & Cagnin, M.I. 2023, 'Counting cattle in UAV images using convolutional neural network', *Remote Sensing Applications*, vol. 29, p. 100900.
- Do Monte Lima, J.P.S., Simões, F.P.M., Uchiyama, H., Teichrieb, V. & Marchand, E. 2016, 'Depth-assisted rectification for real-time object detection and pose estimation', *Machine Vision and Applications*, vol. 27, no. 2, pp. 193-219.

Džermeikaitė, K., Bačėninaitė, D. & Antanaitis, R. 2023, 'Innovations in cattle farming: Application of innovative technologies and sensors in the diagnosis of diseases', *Animals*, vol. 13, no. 5, pp. 780.

Ganesh Kumar, S.S. & Gudipalli, A. 2024, 'A comprehensive review on payloads of unmanned aerial vehicle', *The Egyptian Journal of Remote Sensing and Space Sciences*, vol. 27, no. 4, pp. 637–644.

Gano, B., Bhadra, S., Vilbig, J.M., Ahmed, N., Sagan, V. & Shakoor, N. 2024, 'Drone-based imaging sensors, techniques, and applications in plant phenotyping for crop breeding: A comprehensive review', *Plant Phenome Journal*, vol. 7, no. 1.

Garg, P. K. 2021, *Unmanned Aerial Vehicles: An Introduction*, Mercury Learning & Information, viewed 10 October 2023, < <https://ebookcentral-proquest-com.ezproxy.usq.edu.au/lib/usq/reader.action?docID=5149434>>.

Garg, P.K. 2022, 'Characterisation of Fixed-Wing Versus Multirotors UAVs/Drones', *Journal of Geomatics*.

Grandin, T. 2016, 'Evaluation of the welfare of cattle housed in outdoor feedlot pens', *Veterinary and Animal Science*, vol. 1-2, no. C, pp. 23-28.

Hao, J. & Ho, T.K., 2019. Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics*, 44(3), pp.348-361.

Klingler, viso.ai, 2024, Convolutional Neural Networks (CNNs): A 2024 Deep Dive, viewed 5 September 2024, < <https://viso.ai/deep-learning/fcos-object-detection/>>.

Lapena-Manero, P., Garcia-Casuso, C., Miguel Montenegro-Cooper, J., King, R.W. & Behrens, E.M. 2022, 'An open-source system for generating and computer grading traditional non-coding assignments', *Electronics*, vol. 11, no. 6, pp. 917.

Lee, C., Kim, S. & Chu, B. 2021, 'A Survey: Flight Mechanism and Mechanical Structure of the UAV', *Int. J. Precis. Eng. Manuf.*, vol. 22, pp. 719-743.

Li, X. & Xing, L. 2019, 'Use of Unmanned Aerial Vehicles for Livestock Monitoring based on Streaming K-Means Clustering', *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 324-329.

Los, S., Mücher, C.A., Kramer, H., Franke, G.J. & Kamphuis, C. 2023, 'Estimating body dimensions and weight of cattle on pasture with 3D models from UAV imagery', *Smart Agricultural Technology*, vol. 4, pp. 100167.

Luo, W., Zhang, G., Yuan, Q., Zhao, Y., Chen, H., Zhou, J. & et al. 2024, 'High-precision tracking and positioning for monitoring Holstein cattle', *PLoS ONE*, vol. 19, no. 5, pp. 101.

Mücher, C.A., Los, S., Franke, G.J. & Kamphuis, C. 2022, 'Detection, identification and posture recognition of cattle with satellites, aerial photography and UAVs using deep learning techniques', *International Journal of Remote Sensing*, vol. 43, no. 7, pp. 2377-2392.

Murphy-Hill, E, Markstrum, S & Anslow, C 2010, 'Evaluation and usability of programming languages and tools (PLATEAU)', *Proceedings of the ACM SIGPLAN Conference on Systems, Programming, Languages, and Applications: Software for Humanity (SPLASH) 2010*, pp. 265-266, doi: 10.1145/1869542.1869605.

Ortega, J.C.Ú., Rodríguez-Molina, J., Martínez-Núñez, M. & Garbajosa, J. 2023, 'A Proposal for Decentralized and Secured Data Collection from Unmanned Aerial Vehicles in Livestock Monitoring with Blockchain and IPFS', *Applied Sciences*, vol. 13, no. 1, p. 471.

Padilla, R., Netto, S.L. & da Silva, E.A.B. 2020, 'A survey on performance metrics for object-detection algorithms', 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 237–242.

Parab, C.U., Mwitwa, C., Hayes, M., Schmidt, J.M., Riley, D., Fue, K., Bhandarkar, S. & Rains, G.C. 2022, 'Comparison of Single-Shot and Two-Shot Deep Neural Network Models for Whitefly Detection in IoT Web Application', *AgriEngineering*, vol. 4, no. 2, pp. 507-522.

Plainsight 2022, *Tiling: The Key to Small Object Detection*, viewed 20 October, <Tiling and Computer Vision AI | Plainsight>.

Rancic, K., Blagojević, B., Bezdán, A., Ivošević, B., Tubić, B., Vranešević, M., Pejak, B., Crnojević, V. & Marko, O. 2023, 'Animal detection and counting from UAV images using convolutional neural networks', *Drones*, vol. 7, no. 3, pp. 179.

Ren, C.Y., Prisacariu, V.A., Kähler, O., Reid, I.D. & Murray, D.W. 2017, 'Real-Time Tracking of Single and Multiple Objects from Depth-Colour Imagery Using 3D Signed Distance Functions', *International Journal of Computer Vision*, vol. 124, no. 1, pp. 80-95.

Rivas, A., Chamoso, P., González-Briones, A. & Corchado, J.M. 2018, 'Detection of cattle using drones and convolutional neural networks', *Sensors (Basel, Switzerland)*, vol. 18, no. 7, p. 2048.

Salvin, H.E., Lees, A.M., Cafe, L.M., Colditz, I.G. & Lee, C. 2020, 'Welfare of beef cattle in Australian feedlots: A review of the risks and measures', *Animal Production Science*, vol. 60, no. 13, pp. 1569-1590.

Sarwar, F, Griffin, A, Rehman, SU & Pasang, T 2021, 'Detecting sheep in UAV images', *Computers and electronics in agriculture*, vol. 187, pp. 106219.

Stephany, F., Braesemann, F. & Graham, M. 2021, 'Coding together - coding alone: The role of trust in collaborative programming', *Information, Communication & Society*, vol. 24, no. 13, pp. 1944–1961.

- Suganuma, M., Kobayashi, M., Shirakawa, S. & Nagao, T. 2020, 'Evolution of deep convolutional neural networks using Cartesian genetic programming', *Evolutionary Computation*, vol. 28, no. 1, pp. 141–163.
- Symeonaki, E., Maraveas, C. & Arvanitis, K.G. 2024, 'Recent advances in digital twins for Agriculture 5.0: Applications and open issues in livestock production systems', *Applied Sciences*, vol. 14, no. 2, pp. 686.
- Tabernik, D., Muhovič, J. & Skočaj, D. 2024, 'Dense center-direction regression for object counting and localization with point supervision', *Pattern Recognition*, vol. 153, pp. 110540.
- Terven, JR & Cordova-Esparza, D 2024, *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*, viewed 21 October 2024, <<https://arxiv.org/abs/2304.00501>>.
- Texas A&M University 2019, *Drones could apply thermal imaging to identify sick livestock in feedlots*, viewed 10 October 2023, <<https://research.tamu.edu/2019/03/07/drones-could-apply-thermal-imaging-to-identify-sick-livestock-in-feedlots/>>.
- Ultralyrics Inc 2024, *Ultralyrics YOLOv8*, viewed 20 October 2024, <YOLOv8 - Ultralyrics YOLO Docs>.
- Watts & Davis, 'Feedlot design & construction', viewed 7 October 2023, <<https://www.feedlots.com.au/>>.
- Xu, B., Wang, W., Falzon, G., Kwan, P., Guo, L., Chen, G., Tait, A. & Schneider, D. 2020, 'Automated cattle counting using Mask R-CNN in quadcopter vision system', *Computers and Electronics in Agriculture*, vol. 171, p. 105300.
- Yang, Fei. Zhu, Ningbo. Pei, Shuonan. & Cheng, Irene., 2021, 'Real-Time Open Field Cattle Monitoring By Drone: A 3D Visualization Approach', *University of Alberta, Computer Science*, pp. 124-128.
- Yousefi, D.B.M., Rafie, A.S.M., Al-Haddad, S.A.R. & Azrad, S. 2022, 'A Systematic Literature Review on the Use of Deep Learning in Precision Livestock Detection and Localization Using Unmanned Aerial Vehicles', *IEEE Access*, vol. 10, pp. 80071-80091.

## APPENDIX A

Training and validation results for each fine-tuned model:

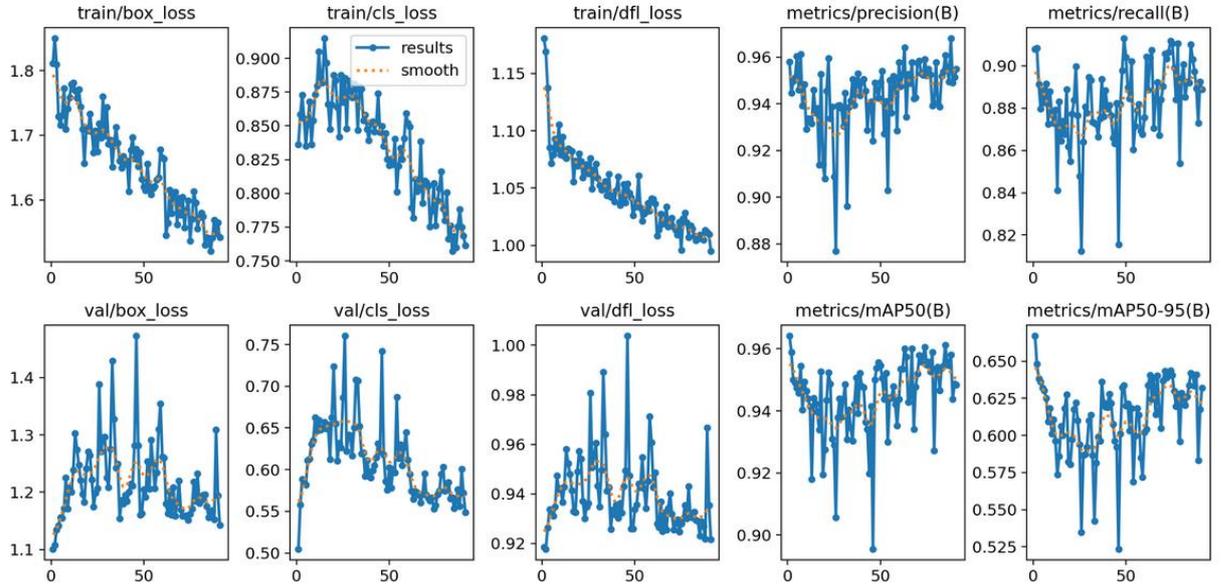


Figure 8.1: Training and validation metrics for the RoboFlow\_custom model

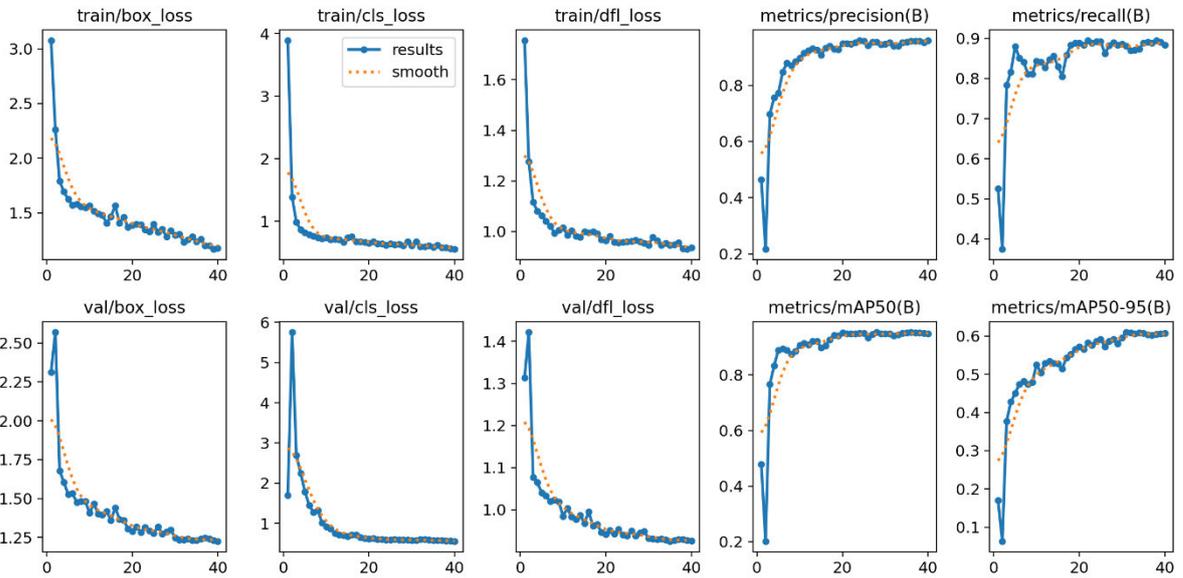


Figure 8.1.1: Training and validation metrics for the YOLOv8-s\_full model

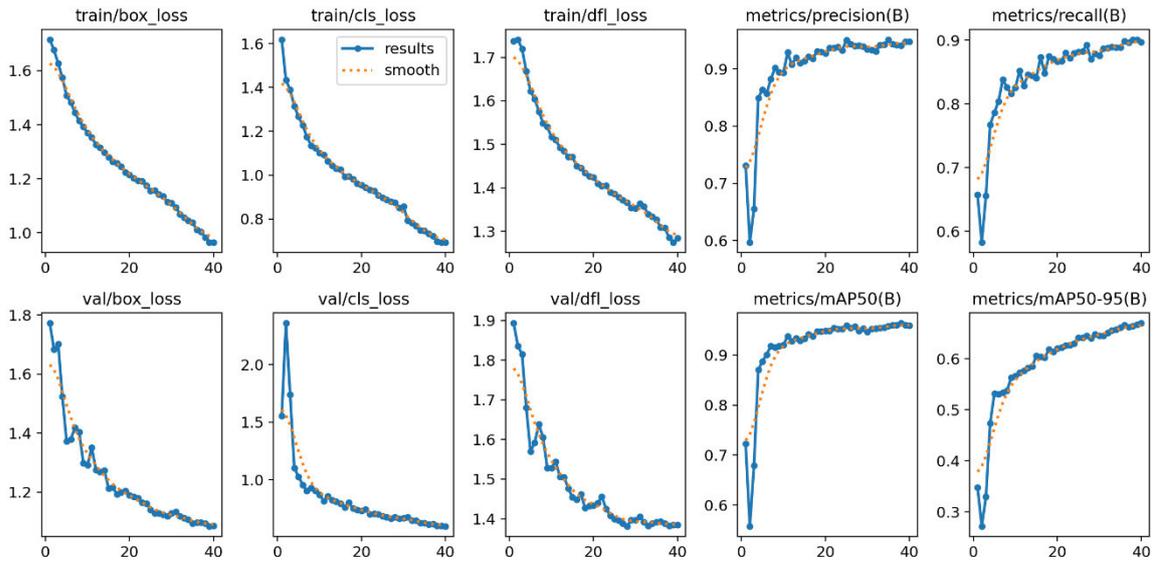


Figure 8.1.2: Training and validation metrics for the YOLOv8-s\_tiled model

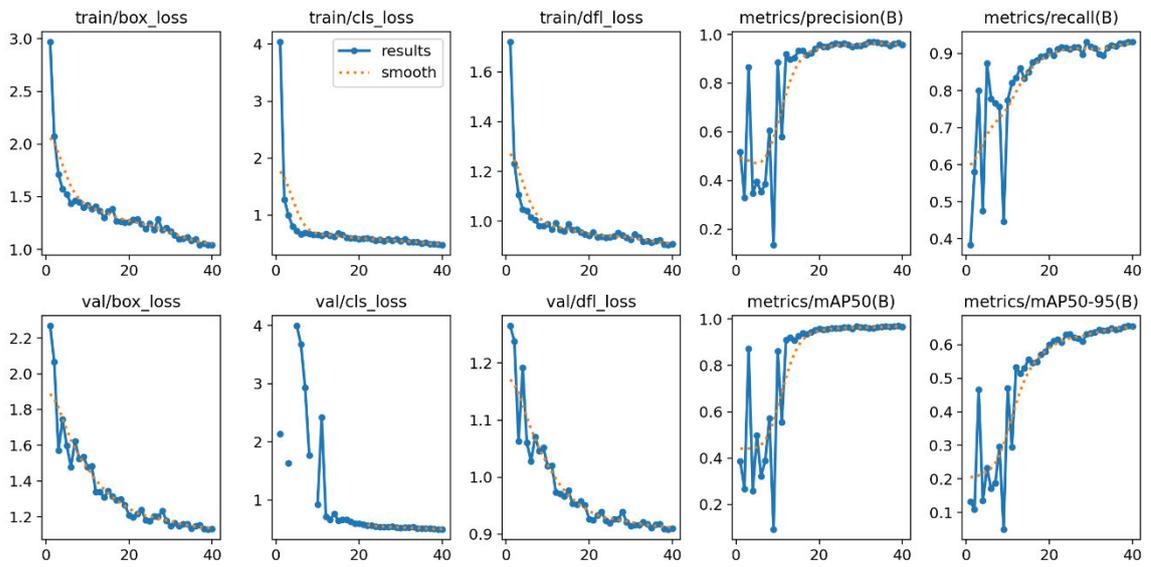


Figure 8.1.3: Training and validation metrics for the YOLOv8-m\_full model

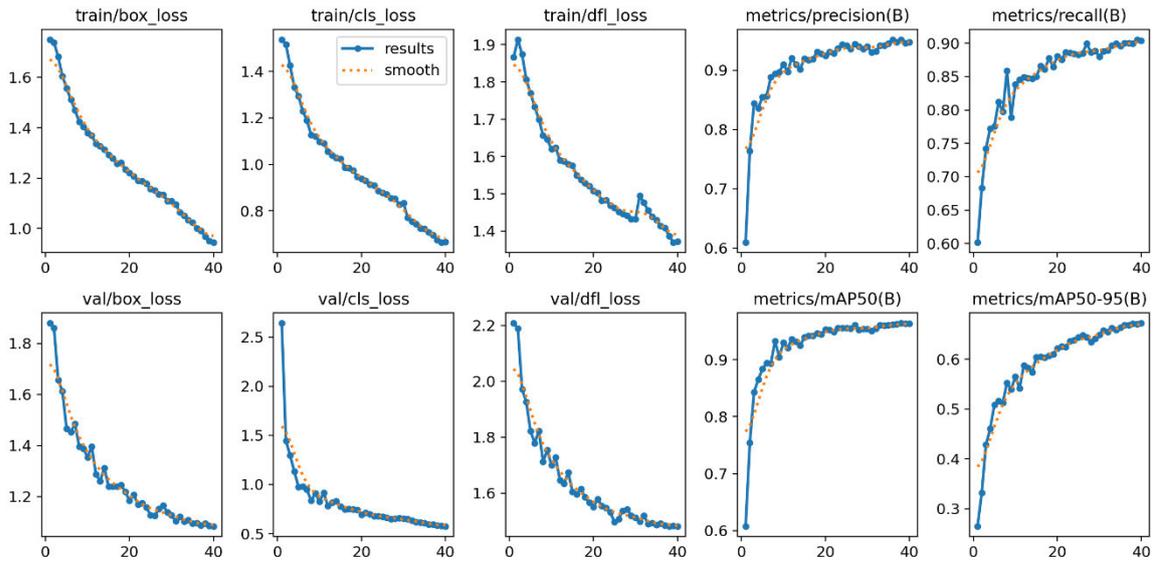


Figure 8.1.4: Training and validation metrics for the YOLOv8-m\_tiled model