University of Southern Queensland

Faculty of Health, Engineering & Sciences

Smart Condition Assessment of Tunnel Structures

A dissertation submitted by

Taylor Meerwald

in fulfilment of the requirements of

ENG4111 and ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Honours) (Civil)

Submitted October 2023

Abstract

Tunnel structures are a major part of our transport infrastructure. These tunnels have a significant financial impact with approximately $9.5 billion spent on their construction and $105 million spent on maintenance in the 2021-2022 financial year(Moore 2019; Transurban 2022). Part of the maintenance processes is the annual inspection of the tunnel structure(Louis 2018). Currently this is done by an experienced engineer who physically examines the tunnel. Recent research has shown that artificial intelligence (AI) technologies can be used to assist with the detection of defects in structures. This paper aims to compare the effect of lighting and resolution quality on the training of a machine learning model for tunnel condition assessment applications.

The project found that there are a number of complicated and interacting variables to navigate in a tunnel environment. It showed that a higher resolution and lighting quality was beneficial in the annotation of images for training of a defect detection model. However, further data would need ot be collected and annotated, ensuring that all classes and quality types have the same number of instances so that the only variable between models is the quality of the image. Whilst this project has not directly contributed to the ultimate industry goal for an integrated infield drone controller and live in-field inspection device, it has helped to gain an understanding of the effects of the challenging variables that are encountered in a tunnel environment.

University of Southern Queensland

Faculty of Health, Engineering and Sciences

ENG4111 & ENG4112 Research Project

Limitations of Use

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

Persons using all or any of this material do so at their own risk, not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering and 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 entitles "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and any 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.

Taylor Meerwald

Student Number: [redacted]

Acknowledgements

Thanks goes to Dr Andy Nguyen, Long Nguyen and Dr Jason Brown for their assistance on numerous aspects of this project. I would also like to acknowledge the support of my friends, family and colleagues throughout this project and my studies; without your encouragement I would not have made it to this stage.

Contents

List of tables

List of Figures

Chapter 1. Introduction

Tunnel structures are a major part of our transport infrastructure. They allow us to better service the population whilst maintain the habitable land. In Brisbane, the Clem7, Airport Link and Legacy way tunnels span over 18km and provide direct routes through the city for motorists which saves them time(Linkt 2023). These tunnels have a significant financial impact with approximately $9.5 billion spent on their construction and $105 million spent on maintenance in the 2021-2022 financial year(Moore 2019; Transurban 2022). Part of the maintenance processes is the annual inspection of the tunnel structure(Louis 2018). Currently this is done by an experienced engineer who physically examines the tunnel structure for any areas of concern(Louis 2018). This is a time-consuming process which causes disruption to motorists as the tunnel or part thereof must be closed to allow for this inspection.

Recent research has shown that artificial intelligence (AI) technologies can be used to assist with the detection of defects in structures. In general, this is less time and resource consuming and more precise(Altabey & Noori 2022). There has been extensive research into defect detection and classification on structures such as buildings, bridges and road pavements, however research for tunnel structures has stopped short at defect detection(Yu et al. 2007; Menendez et al. 2018; Flah et al. 2020; Guo et al. 2020). This paper aims to compare the effect of lighting and resolution quality on the training of a machine learning model for tunnel condition assessment applications.

Chapter 2. Literature Review

2.1. Tunnel construction and deterioration

There are many ways that a tunnel can be constructed including cut and cover, bored, clay kicking, shaft, pipe jacking and box jacking methods(Anupoju 2023). Most modern road tunnels are constructed using the bored tunnel method and a tunnel boring machine (TBM)(Northeast Maglev 2023). A TBM is a large cylindrical boring machine which not only cuts through and disposes of the material where the tunnel will be but also install precast concrete rings in the precise location to line the tunnel as it progresses(CrossRiver Rail 2023).

Despite their unique construction, they are still subject to similar structural defects as other structures. The main difference between a bridge structure and a tunnel structure is where the loading on the structure comes from; for a bridge it is imposed loads by through the use of the bridge whilst tunnel structures are subjected to constant loads from the surrounding earth. Over time, this loading can cause many structural defects of varying severity. The most concerning being horizontal, diagonal and vertical cracks caused by a shearing or compressive failure of the tunnel lining(Centre d'Études des Tunnels 2015). These defects cause localized weakening of the tunnel structure and should be monitored closing and repaired if they exceed a pre-defined width(Centre d'Études des Tunnels 2015). Shrinkage cracks pose a less significant threat to the structural integrity of the tunnel lining depending on the extent and depth of the cracking(Centre d'Études des Tunnels 2015). These cracks should be monitored and repairs undertaken if they are deeper than surface level; being more than 1-2mm deep(Centre d'Études des Tunnels 2015). Some visual defects, however, have no impact on the structural integrity. For example dense crazing, bleeding and blistering do not pose significant risks to the structure but may be displeasing to look at(Centre d'Études des Tunnels 2015).

2.1. Devices and techniques used for condition assessments of tunnels

Some early techniques for condition assessments in tunnels included ground penetrating radar (GPR) and vehicle-mounted detection systems using a range of sensors including high-resolution cameras, LiDAR scanners, infrared lighting and ultrasonic sensors. Ground penetrating radar is a non-destructive assessment tool that uses electromagnetic radiation microwaves to penetrating through a structure(Qin et al. 2016). The device detects the reflected microwaves which can then be used to determine the integrity of the subsurface structure(Qin et al. 2016). Ground penetrating radar was used for assessment of tunnels structures as early as 2014 (Alsharqawi et al. 2022). GPR has been used most often in research applications for the detection of voids behind concrete and in grout lines (Alsharqawi et al. 2022). It has also been used to detect delamination, cracks, leakage and clay lumps in both simulations and in field tunnel condition assessment applications (Alsharqawi et al. 2022). Alsharqawi et al. found only two papers, both published in 2016, which used GPR to detect cracks in tunnel structures; Yu et al. and Qin et al. (Qin et al. 2016; Yu et al. 2016; Alsharqawi et al. 2022). Both of these papers looked at simulated defect detection as opposed to in field testing, using either waveform inversion or image processing techniques (Alsharqawi et al. 2022). Ground penetrating radar requires physical testing of the structure which poses some challenges with transport tunnels as the roadway would need to be closed for a significant amount of time to allow for sufficient coverage in the testing.

2.2. Architecture of machine learning programs

As artificial intelligence (AI) continues to develop, its applications for structural condition monitoring have increased. The most popular method of machine learning with researchers for object detection is a deep convolutional neural network (DCNN or DNN) as it requires a fewer number of artificial neurons than previous convolutional networks (run.ai 2023). In a DNN, images are input and used to train the classifier to detect objects of interest (run.ai 2023) . Objects are identified using a mathematical operation know as a convolution(run.ai 2023). Most DNN's consist of 4 different layers being a convolution, pooling, activation and fully connected.

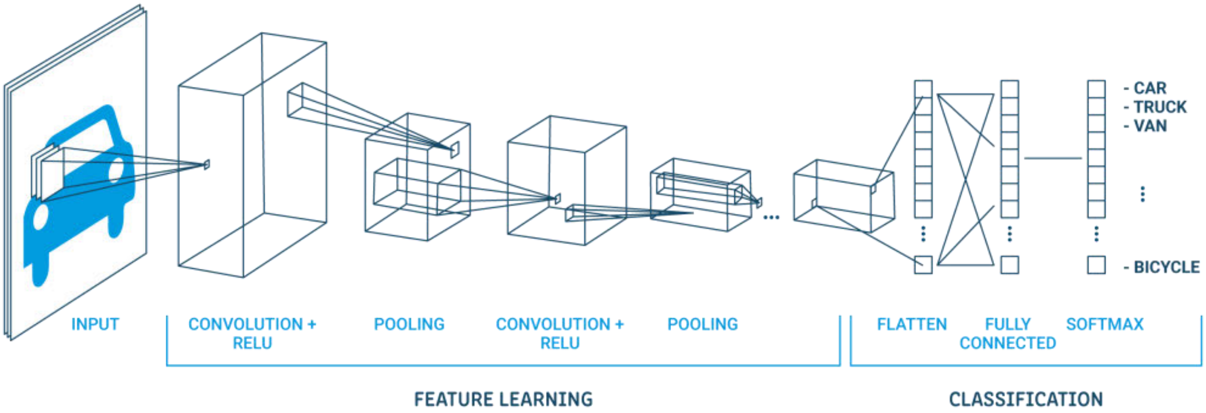


Figure 1. DNN Network Architecture - Source:(run.ai 2023)

In the convolutional layer, a set of weights is applied to an image(run.ai 2023). A filter is then passed over the image multiple times from top right to bottom left, horizontally and multiplies the weight of the filter by the number already associated with the pixel or image area(run.ai 2023).The value of these weights is then summed for each position of the filter before reaching the activation and pooling layers (run.ai 2023). The activation layer replaces all negative values with a zero and the pooling layer reduces the image size by keeping only the pixel group with the highest or average values, depending on the setup of the program(run.ai 2023). The program is finished with a fully connected layer which receives a vector input of the image pixels which were put through the first three layers and allows the program to determine the probability that the image belongs to a certain group(run.ai 2023). There is often multiple convolution and pooling layers in a single DNN(run.ai 2023). This project will aim to utilize a Fast R-CNN which is a type of DNN that is capable of identifying areas of interest in a faster manner than an R-CNN alone areas of interest in a faster manner than an R-CNN alone(run.ai 2023).

To train a DNN, you must first have image or video data that has been labelled with areas of interest that you want the network to identify(Maayan 2022). There are a number of types of image labelling. The first is image classification where the network will learn to classify an entire image into a specific group(Maayan 2022). These classifications can be binary, multiclass or multi-label depending on the required outcome(Maayan 2022). Another type of image labelling is image segmentation where objects are separated from the background image using a pixel map and binary notation to identify areas of interest(Maayan 2022). Object detection is another type of image labelling where a bounding box is placed around the object of interest(Maayan 2022).

2.3. Previous research for condition assessment of tunnels

An early paper on crack detection in tunnels is that by Yu et al. where cracks were identified by light reflectance values(Yu et al. 2007). They then used the Sobel and Laplacian algorithms to measure the crack length and determine its direction(Yu et al. 2007). The detected cracks were then grouped by their pattern of connection(Yu et al. 2007). Despite being an interesting approach, this method did have an error rate of 75-85% but the measurement error of the recognized crack was less than 10% which is a promising result(Yu et al. 2007). The robot used in this paper was able to reach an accuracy of 0.3mm per pixel at a speed of 5km/h(Yu et al. 2007). The accuracy and efficiency of this research could have been assisted by the use of an artificial intelligence neural network.

Structural health monitoring using neural networks is not a new topic. Prasanna et al. developed in 2016 one of the example of a neural network capable of identifying surface cracks(Prasanna et al. 2016). Recent works have shown the ability for an AI network to detect defects in a variety of structures. Guo et al. proposed and tested a deep-width network which was capable of isolating defects and further classifying the type of defect on bridges, buildings and road pavements(Guo et al. 2020). This paper also used the process of batch training to reduce the necessary computing power(Guo et al. 2020). The network architecture consisted of multiple layers or convolutions beginning with an initial binary classification layer which divided the segmented images into two classes depending on the number of nodes and layers; various wide feature nodes and deep nodes with fixed single layer or fixed wide feature nodes and different layers of deep nodes(Guo et al. 2020). This was followed by a multi-label convolution which further divided the structural image segments into one of three categories: non-crack, single-module crack (simple) or multi-module crack (complicated)(Guo et al. 2020). This is a great example of crack identification and classification but the program was not trained specifically for use in tunnels which have slightly different concerns compared to bridges, building and road pavements.

Another noteworthy paper is that by Flah et al. who designed a deep-learning program that used both binary and multi-label classifiers to identify cracks in concrete structures(Flah et al. 2020). This paper also utilised image segmentation to remove noise, connect cracked regions and calculate the width of the identified cracks(Flah et al. 2020). The determined crack widths were then used to predict the failure mode of the structure and advise the severity of the damage(Flah et al. 2020). This program was trained for application on bridges and buildings only as the severity of the damage is dependent on the building type and exposure to the elements(Flah et al. 2020). This paper conducted a field test on a square type pedestrian tunnel(Flah et al. 2020). Although this paper is a perfect example of the type of program we are aiming to develop, it is not directly applicable to larger tunnel structures which leaves a space for further works to be conducted.

In 2016, Protopapadakis et al. created a network that was able to detect cracks in tunnels(Protopapadakis et al. 2016). The cracks were quantified with the use of a laser scanner and placed on a 3D point cloud for easy location by engineers(Protopapadakis et al. 2016). Menendez et al. built on this in 2018 by not only applying the network by Protopapadakis et al. for crack, spalling and efflorescence detection but further constructed an autonomous robot to conduct the assessment(Menendez et al. 2018). The robot was fitted with a camera and an ultrasonic sensor on the end of a boom lift(Menendez et al. 2018). The program was able to detect a crack defect in real-time and then move the ultrasonic sensor over the crack to determine the depth and width of the crack(Menendez et al. 2018). The robot was able to store information on board until it return to the laboratory and the data was downloaded (Menendez et al. 2018). This paper showed great progress towards a fully automated structural health monitoring system however, as it was confined to a large, fixed machine, the part of the roadway still had to be closed to allow for the slow-moving robot to conduct its assessment. An improvement would be using a unmanned aerial vehicle (UAV) such as a drone to capture the images and send the image data to an off-site or mobile laboratory where the processing equipment would give real-time information on the structure’s condition.

Ren et al. developed a deep fully convolutional network named CrackSegNet and introduced a pixel-wise crack segmentation method(Ren et al. 2020). Ren et al. was able to use a small data set of 409 images by cropping the images and using other data augmentation techniques such as rotation, translation, scaling and sheering to train the network(Ren et al. 2020).The paper also utilised four diluted convolutional layers: a diluted convolutional layer utilises defined gaps or holes to increase the image resolution(Ren et al. 2020). This can be used to remove the pooling layer which results in better feature extraction while maintaining the original resolution of the image(Ren et al. 2020). This method showed an increase of 50% and 8% from the conventional method by Li et al. and U-net method by Liu et al., respectively(Ren et al. 2020). The improvements to previous methods using a pixel-wise segmentation are significant and will be implemented in this project.

Chapter 3. Methodology

3.1. Test tunnel

Data was sampled from the Ernest Junction Railway Tunnel in Molendinar, Queensland(Arts Queensland 2020). This tunnel was built between 1885-1889(Arts Queensland 2020). It formed part of the Brisbane – South Coast rail line until it was closed in 1964(Arts Queensland 2020). The tunnel is curved and runs for 114m long at approximately 21m under the crest of the hill(Arts Queensland 2020).

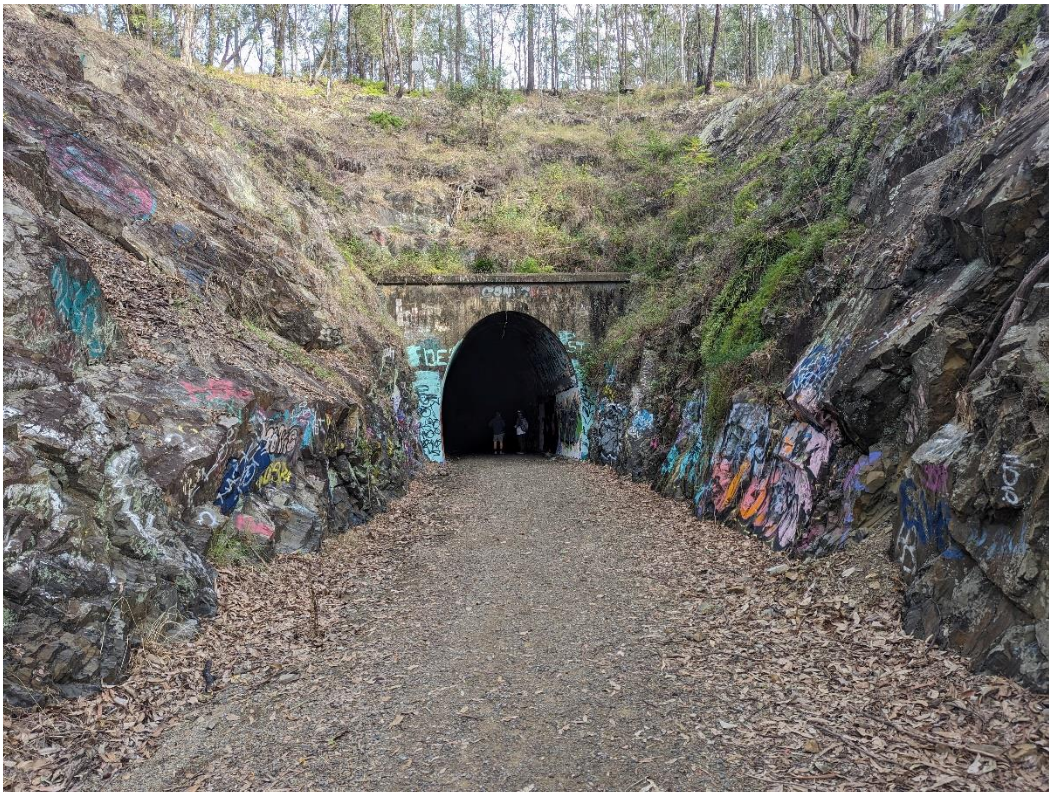


Figure 2. Ernest Junction Railway Tunnel Entrance

3.2. Data collection methods and equipment

Two sets of data were collected from the Ernest Junction Railway Tunnel. The first was done with a Mavic 2 Pro drone with a Milwaukee 12V LED torch (Milwaukee 2021; DJI 2023a). The Mavic 2 Pro has 20 million effective pixels and an ISO maximum of 6400(DJI 2023a). The Milwaukee 12V LED torch has a 6.8 lux rating at 30m(Milwaukee 2021). The torch was held at ground level by the operator kept approximately centered under the drone as it progressed through the tunnel. The drone was manually operated and maintained a relatively consist of height and distance from the wall. Photos of this drone and torch are included in Appendix A. The second set was collected using the Mavic 2 Enterprise Advanced with the M2EA Spotlight attached(DJI 2023b). The Enterprise camera has 48 million effective pixels and the attached spotlight has an 11 lux rating at 30m. A photo of this setup is included in Appendix AFigure 8. Mavic 2 Enterprise Advanced with spotlight attached at tunnel site Figure 8. The Mavic 2 Enterprise was manually controlled with the assistance of the onboard guidance system to prevent collisions. The height and distance from the wall were relatively consistent given the manual operation.

3.3. Data labelling and augmentation

The video files were uploaded to Roboflow and images were extracted at a rate of 1 frame per second for both datasets. The project type was object detection to allow annotation and exporting of the images in the required file type for the model training. Roboflow allows you to change the brightness and contrast of an image and enlarge an image as required to place a square or polygon bounding box. Polygon bounding boxes were used in this project due to the complex nature of concrete cracks and to ensure that each bounding box was placed as close to the defect as possible. Each bounding box is then added to a class. The classes used were concrete crack, surface defect and construction joint. Concrete crack defects included a range of different cracks including horizontal, diagonal and spalling. These cracks were not separated in this study due to the difficulty in maintaining context of the direction of the crack, especially on the roof of the tunnel. Spalling was included with singular cracks due to the limited number of instances in the obtained images. A surface defect included areas or delamination, honeycombing or pitting in the concrete surface. Whilst a construction joint is not a defect, this class was necessary as there were many joints in the surface from the time of construction which could be confused with a concrete crack. Hence this class was included to prevent false prediction of a crack where an intended joint was located. This class will still be referred to as a defect in this report due to ease of reporting. There were two other classes that were added to some images but not included in the analysis due to irrelevance to the project aims and small number of instances; “insufficient” for areas of an image that were over or under exposed and “external” for areas outside of the tunnel. Examples of the image annotation and Roboflow program can be found in Appendix B.

The program also allows the user to apply tags to images. Each image was tagged with a resolution and lighting quality of either low, medium, or high. Although this is a subjective criterion, there was only one annotator so any bias would remain reasonably consistent. Lighting quality was judged based on the percentage of the image that was not visible due to over or under exposure as well as the amount of manipulation required for brightness and contrast. For example, an image with more than 75% not visible or requiring the maximum brightness would be classed as having low lighting quality whereas an image which requires little to no change in the brightness and contrast settings and over 90% of the image is visible would be classed as having high lighting quality. Resolution quality was based on the clearness of defects in the entire image and when the image was enlarged. A low-resolution image was one which was blurry prior to enlarging. Medium resolution was classed as an image which was visible prior to enlargement but became blurry and difficult to read when enlarging. A high-resolution image was one that was still reasonably clear when enlarged. It is important to note that the amount of enlargement also affects how blurry the pixels of an image may become. As with the tagging of image quality, the enlargement was not entirely consistent, however every effort was made to enlarge to similar dimensions when looking for smaller defects. Examples of the image tagging can be found in Appendix B.

Once all of the images were labelled, the final dataset could be generated. Firstly, the labelled images were split into the training, validation and testing sets at a ratio of 70:20:10. A number of pre-processing steps were then completed including “auto-orientate”, “isolate objects”, “resize” and “auto-adjust contrast”. “Isolate objects” was chosen to allow for a larger dataset for training than there would be if an entire annotated image was used. This also allows the model to become familiar with looking for patterns in the defects rather than patterns in the image as a whole. “Resize” was chosen to ensure consistent image sizes and “auto-adjust contrast” was used to optimize the lighting, especially in those images with low lighting quality. The images were then augmented to increase the number of examples and help the model to look for different variations of the same defect. The augmentations steps included “90^o rotate”, “blur” and “noise”. “90^o rotate" rotated the images by 90^o in clockwise and anti-clockwise directions and was chosen to ensure the model could identify different orientations of each defect type. “Blur” created a Gaussian blur of up to 2.5pixel to increase the model’s ability to detect defects in lower quality resolutions. “Noise” introduces a combination of black and white pixels at a 5% ratio to the entire image to also aid in the detection of defects in lower quality resolutions. Once the pre-processing and augmentation was complete, the images could be exported in a YOLOv5 Pytorch format. The zip file for the images was then added to a Google Drive so it could be accessed by the training program. Examples of the program can be seen in Appendix B.

3.4. Model training

This project utilized the YOLOv5 training model in Google Colabs to train the detection model. This training model has previously been implemented in work done by Long Nguyen(Nguyen 2023). This implementation formed the basis of the program for this project, allowing the data set to be uploaded and the model trained. The training input for YOLOv5 required data and weights. The data came from the dataset zip file which was imported and the weights used were the generic ‘YOLOv5.pt’. The epochs were specified as 2, batch number as 1 and the image size as 768pixels. These settings were used as they had previously been optimized by Long Nguyen(Nguyen 2023). Once trained the model uses the validation images to determine the precision and accuracy of the model training by comparing the predicted class with the actual class. The program then plots the results and provides some example outputs from the validation set. These outputs can be found in Appendix C. It is important to note that the Mavic Pro classes were incorrectly labelled and this was not able to be corrected so the model outputs for this model have the “concrete crack” listed as “concrete crack”, “surface defect” as “cavity in concrete” and “construction joint” as “concrete construction separation”.

Chapter 4. Results and Discussion

4.1. Model output

The Model outputs for both the Mavic Pro and Mavic Enterprise can be found in Appendix C. The model outputs a number of graphs including precision-confidence, recall-confidence, F-1-confidence and precision-recall curve, confusion matrix, results graph and label frequency graph. To understand the graphs, we must first understand the possible outcomes; true positive, true negative, false positive and false negative. A positive outcome is one where the model identifies a defect where as a negative is when there is no defect identified('True Positive' 2010). The true and false refer to whether the models prediction is correct; for example, an outcome of true positive is a correct identification of a defect where a false positive is an incorrect prediction of a defect('True Positive' 2010). The precision of a model is the number of true positives divided by the sum of the true positives and the number of false positives(Ting 2010a). The recall value is the number of true positives divided by the sum of the true positives and false negatives (Ting 2010a). An easier way to think of this is the precision is the number of predicted defects which are accurate, and the recall is the number of actual defects that are predicted. The precision-confidence and recall-confidence curves are shown in Figure 19, Figure 21, Figure 29 and Figure 31 of 25Appendix C.

The model also produces the precision-recall curve which shows the relationship between the precision and recall values. As stated above, the precision gives the number of accurate predictions and the recall gives the number of actual defects that are predicted. This gives a good understanding of the overall accuracy of the model as we are comparing the number of accurately predicted defects. It can be seen in Figure 3 and **Error! Reference source not found.** that the Mavic Enterprise model had a higher precision and recall value for all classes. Although the overall precision-recall value for the Mavic Pro model may have been reduced slightly by the inclusion of the "insufficient" class, it can be seen that the individual accuracy of each class is still higher in the Mavic Enterprise model. Another output is the confusion matrix, which shows the performance of the model in tabular form of the assigned class verse the actual class(Ting 2010b). The confusion matrix for both models can be found in Figure 22 and Figure 33.

The final output from the model is the results graph. This shows a number of smaller graphs in a single image. The first three columns show the box loss, object loss and class loss for the training in the top row and the validation in bottom row. The 4th and fifth column on the top row show the metrics/precision and the metrics/recall respectively. The bottom row of the same columns shows the metrics/mAP for the 50% and 50-95% confidence intervals respectively. These plots can be seen in Figure 23Figure 34 in Appendix C. The label frequency graph is also shown in Appendix C and shows the frequency distribution of each class. Aside from the statistical outputs, the model also outputs examples of the validation. Two examples have been included for each model in Figure 24Figure 27 and Figure 35Figure 38 of Appendix C.

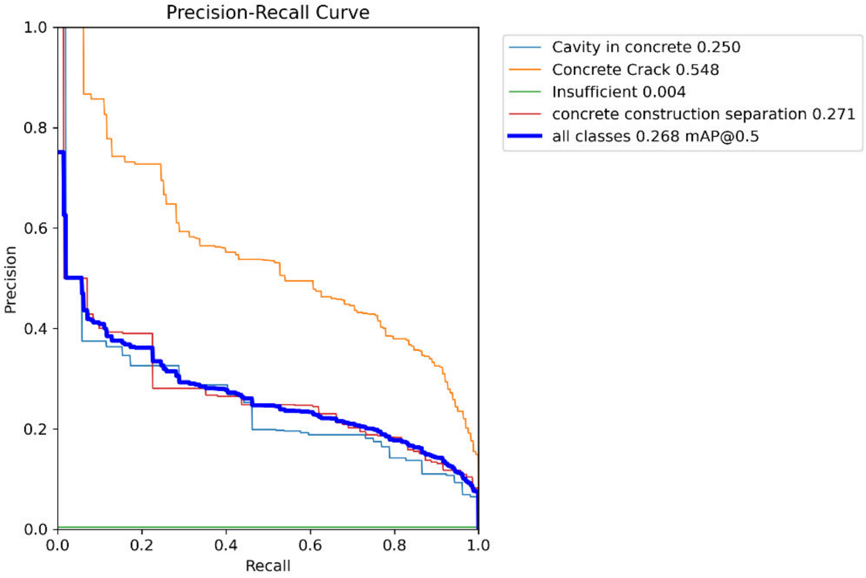


Figure 3. Mavic Pro Precision-Recall Curve

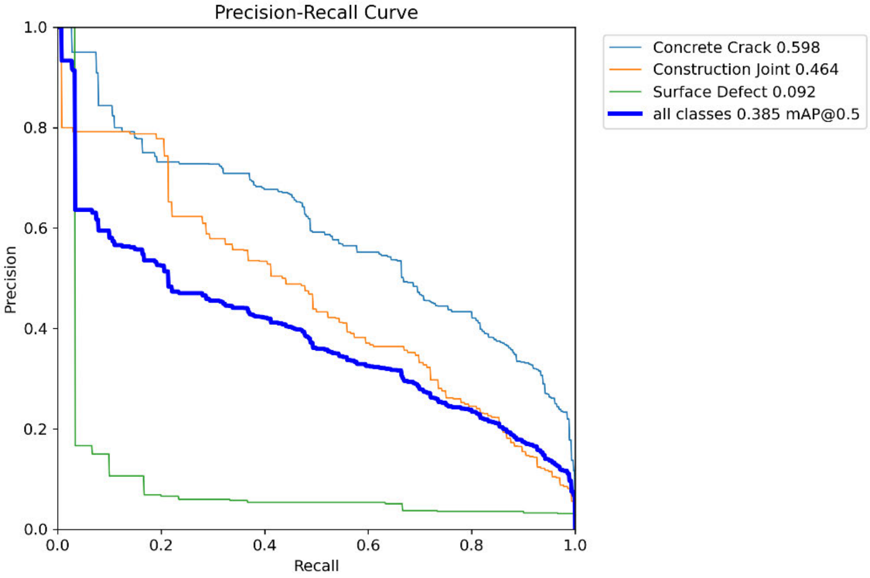


Figure 4. Mavic Enterprise Precision-Recall Curve

4.2. Data distribution

Further analysis of the model results was conducted using excel to compare and graph the distribution of the classes of both models as well as the image lighting and resolution quality. As noted in Section 3.3, the annotated images were split into a training, validation and testing set. The difference in number of training images for the Mavic Pro and Mavic Enterprise datasets was only 24 images as shown in Figure 5. Image Quantity Comparison. In the validation set, the difference between the models is more significant, however the majority of the training has already been conducted and as such this disparity is not a concern for the accuracy of the model.

More information on the distribution between the training and validation sets can be found in Table 1 in Appendix D.

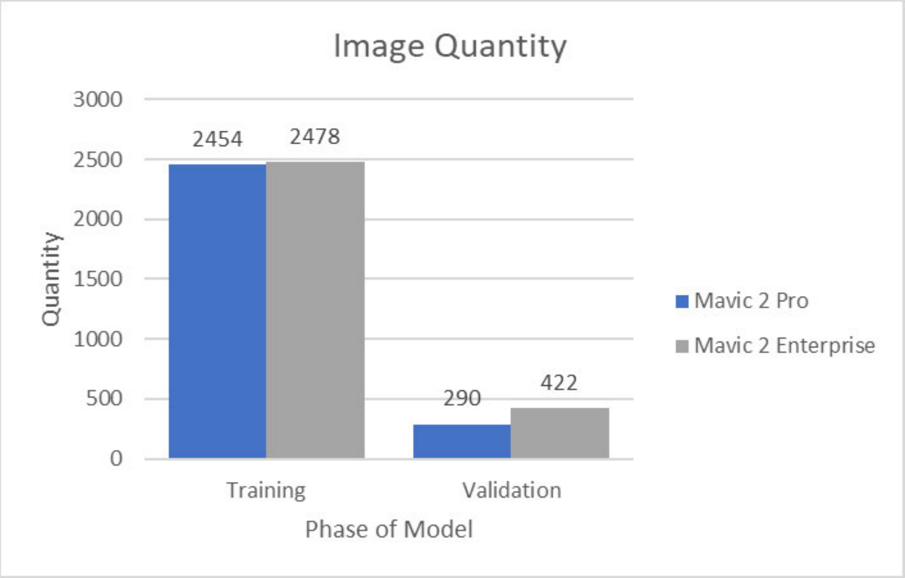


Figure 5. Image Quantity Comparison

The overall percentage of the lighting and resolution distribution was also graphed to highlight potential issues in accuracy of annotation and detection. As seen in Figure 6 below, the low and medium lighting for both models were reasonably similar. A slight decline in high lighting quality was seen in the Mavic Enterprise data set with respect to the Mavic pro dataset. This is most likely due to the number of images which were over exposed due to the brighter light and higher concentration of light in the frame. In terms of resolution, the Mavic Enterprise has a higher number of high-resolution images, which is to be expected with higher camera quality. However, it can be seen that for the medium and low-resolution images, the Mavic Pro dataset had a slightly higher number of images. Again, this is not unexpected as the difference is somewhat accounted for in the higher resolution images of the Mavic Enterprise dataset. Overall, it can be seen that resolution was largely improved in the Mavic Enterprise dataset in comparison with the Mavic Pro set. It can also be seen that the lighting did not necessarily affect the resolution quality of the images. Further information on the image quality distribution can be found in Table 2Table 3 andTable 4 in Appendix D.

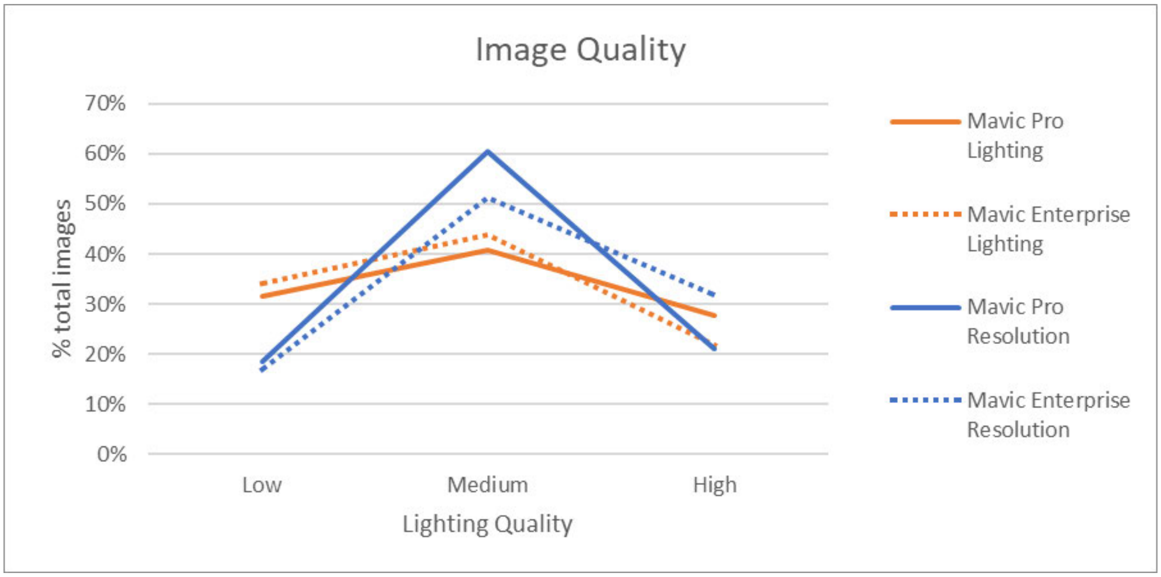


Figure 6. Distribution of Image Quality

4.3. Results analysis

The final analysis was conducted to show the number of images in each class compared to the precision, recall and mean average precision(mAP) values for different confidence intervals as shown in Figure 7. Across all classes, the Mavic Pro model showed better precision, however the Mavic Enterprise performed better in precision and mAP. The Mavic Enterprise dataset also had a larger number of validation images than the Mavic Pro dataset. From this we can infer that whilst the Mavic Pro model had a higher precision over all cases, the ability to detect defects and the confidence in these detected defects was higher in the Mavic Enterprise model.

The surface defect class was the class with the lowest instances in both models. It can be seen that the Mavic Pro dataset had slightly more images than the Mavic Enterprise dataset and this is reflected in the precision, recall and mAP statistics. Due to the small number of instances, this class is not taken into consideration for the overall performance analysis.

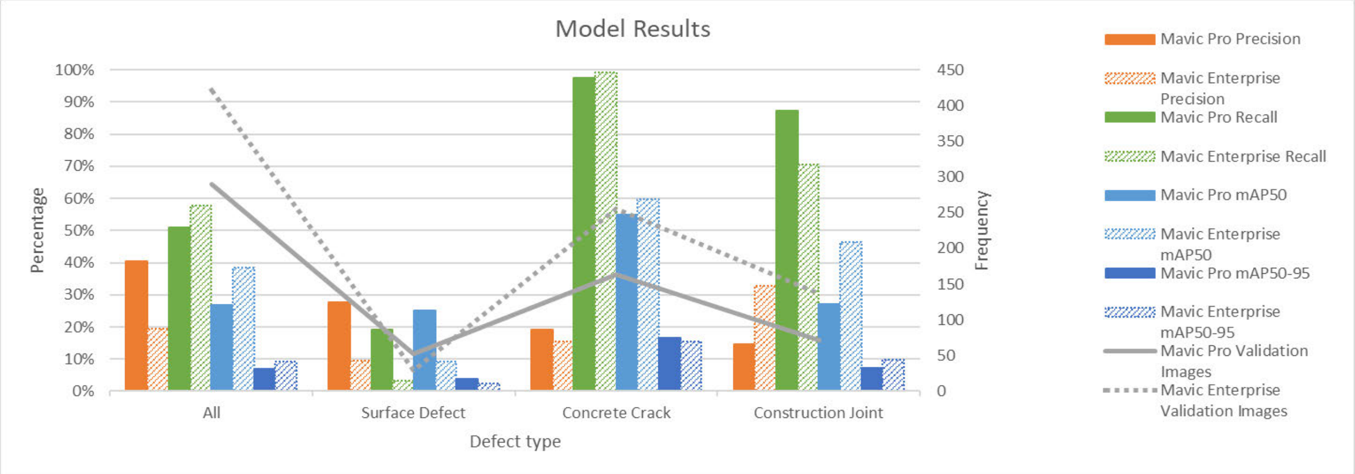


Figure 7. Model Results

In the concrete crack class, the Mavic Enterprise again had a higher number of images than the Mavic Pro dataset. Similar to the overall values, the precision was lower than the Mavic Pro model but the recall and mAP at 50% confidence were higher. The mAP for confidence 50-95% was lower for the Mavic Enterprise model than the Mavic Pro model meaning that despite the larger number of images, the Mavic Enterprise model was not as accurate as the Mavic Pro model for the concrete crack class. Although, the recall being higher for the Mavic Enterprise model means that this model is more likely to detect defects.

The final class was the construction joint. As mentioned earlier, this is not so much a defect as teaching the models that crack like images with a certain degree of linearity may not be a concrete defect. The Mavic Enterprise model again had a larger number of images for this class and performed better in the precision and both mAP confidence intervals. Interestingly, it didn't perform as well in the recall which is different to most other classes in this analysis. It is unclear why this occurred as it is expected that a larger number of training examples would lead to a better ability to detect the defect. One explanation may be a confusion between the construction joint and concrete crack classes. This can be seen in the confusion matrix as both models had a large percentage of images confused for concrete cracks. This may be due to the significant difference in training images between the concrete cracks and other classes making the concrete class overfitted for the models in comparison to the other classes.

3.3

Chapter 5. Conclusions

This project has shown that whilst the effects of lighting and resolution can be accounted for in the training of a model, they can have an impact on the annotators ability to identify defects prior to a model being trained. To correct this, the consistency of lighting and resolution should be controlled by taking the manual operation of the UAV out of the equation and ensuring that the lighting source covers the entire frame of the video and is not concentrated in on location. Having more consistency in the UAV operation would also allow for values such as defect dimension and crack widths to be calculated as they can be scaled when the distance from the wall and the frame size is known.

To better assess the effects of resolution and lighting on the quality of training, more data needs to be collected and annotated to ensure that all classes have the same number of instances and the distribution of quality in each class is also consistent. This would require approximately nine models to be run, using the low, medium and high-quality labels, to ensure that the only variable between each model is the quality of the lighting or resolution.

Whilst this project has not directly contributed to the ultimate industry goal for an integrated infield drone controller and live in-field inspection device, it has helped to gain an understanding of the effects of the challenging variables that are encountered in a tunnel environment.

References

Alsharqawi, M, Dawood, T, Abdelkhalek, S, Abouhamad, M & Zayed, T 2022, 'Condition assessment of concrete-made structures using ground penetrating radar', *Automation in Construction*, vol. 144, p. 104627.

Altabey, WA & Noori, M 2022, 'Artificial-Intelligence-Based Methods for Structural Health Monitoring', *Applied Sciences*, vol. 12, no. 24, viewed 26/5/23,<.

Anupoju, S 2023, 'Tunnel Construction Techniques and Their Details', *The Constuctor*, viewed 27/05/23,<https://theconstructor.org/construction/tunnel-construction-methods/17167/>.

Arts Queensland 2020, *Ernest Junction Railway Tunnel*, viewed 12/7/23, <https://www.arts.qld.gov.au/case-studies/ernest-junction-railway-tunnel>.

Centre d'Études des Tunnels 2015, *Road tunnel civil engineering inspection guide,* Bron France, viewed 23/5/23, <https://www.cetu.developpement-durable.gouv.fr/IMG/pdf/road_tunnel_inspection_guide_book2_hv.pdf>.

CrossRiver Rail 2023, *Guide to tunnelling*, viewed 27/05/23, <https://crossriverrail.qld.gov.au/construction/guide-to-tunnelling/>.

DJI 2023a, *Mavic 2*, viewed 25/10/23, <https://www.dji.com/au/mavic-2/info>.

DJI 2023b, *Mavic 2 Enterprise Advanced*, viewed 25/10/2023, <https://enterprise.dji.com/mavic-2-enterprise-advanced/specs>.

Flah, M, Suleiman, AR & Nehdi, ML 2020, 'Classification and quantification of cracks in concrete structures using deep learning image-based techniques', *Cement and Concrete Composites*, vol. 114, p. 103781.

Guo, L, Li, R, Jiang, B & Shen, X 2020, 'Automatic crack distress classification from concrete surface images using a novel deep-width network architecture', *Neurocomputing*, vol. 397, pp. 383-92.

Linkt 2023, *About Queensland toll roads*, viewed 25/5/23, <https://www.linkt.com.au/using-toll-roads/about-brisbane-toll-roads/brisbane>.

Louis, L 2018, *Guide to Road Tunnels Part 3: Operations and Maintenance,* Austroads, Austroads, Sydney NSW 3000 Australia, viewed 30/4/23, <https://austroads.com.au/publications/tunnels/agrt03>.

Maayan, GD 2022, 14/10/2022, 'Complete Guide to Image Labeling for Machine Learning', *Dataversity*, viewed 27/05/23,<https://www.dataversity.net/complete-guide-to-image-labeling-for-machine-learning/#>.

Menendez, E, Victores, J, Montero, R, Martinez, S & Balaguer, C 2018, 'Tunnel structural inspection and assessment using an autonomous robotic system', *Automation in Construction*, vol. 87, pp. 117-26.

Milwaukee 2021, *M12™ LED HIGH PERFORMANCE FLASHLIGHT (TOOL ONLY)*, viewed 25/10/2023, <https://www.milwaukeetool.com.au/power-tools/lighting/personal-lighting/M12MLED-0.html>.

Moore, T 2019, 14 October 2019, 'No congestion-busting benefit from Brisbane's $10 billion toll roads: expert', *Brisbane Times*, viewed 25/5/23,<https://www.brisbanetimes.com.au/politics/queensland/no-congestion-busting-benefit-from-brisbane-s-10-billion-toll-roads-expert-20191014-p530m4.html>.

Nguyen, L 2023, *Yolov5*, Google Colabs.

Northeast Maglev 2023, *Tunnels and Transport: A Hsitory of the World*, viewed 27/05/23, <https://northeastmaglev.com/2018/10/08/cool-stuff-to-know-about-tunnels/#:~:text=Currently%2C%20the%20more%20common%20way,types%20of%20soil%20and%20rock.>.

Prasanna, P, Dana, KJ, Gucunski, N, Basily, BB, La, HM, Lim, RS & Parvardeh, H 2016, 'Automated Crack Detection on Concrete Bridges', *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591-9.

Protopapadakis, E, Makantasis, K, Kopsiaftis, G, Doulamis, N & Amditis, A 2016, *Crack Identification Via User Feedback, Convolutional Neural Networks and Laser Scanners for Tunnel Infrastructures*.

Qin, H, Xie, X, Vrugt, JA, Zeng, K & Hong, G 2016, 'Underground structure defect detection and reconstruction using crosshole GPR and Bayesian waveform inversion', *Automation in Construction*, vol. 68, pp. 156-69.

Ren, Y, Huang, J, Hong, Z, Lu, W, Yin, J, Zou, L & Shen, X 2020, 'Image-based concrete crack detection in tunnels using deep fully convolutional networks', *Construction and Building Materials*, vol. 234, p. 117367.

run.ai 2023, *Deep Convolutional Neural Networks: A Guide*, viewed 27/05/23, <https://www.run.ai/guides/deep-learning-for-computer-vision/deep-convolutional-neural-networks#:~:text=Deep%20convolutional%20neural%20networks%20(CNN,the%20visual%20cortex%20of%20animals.>.

Ting, KM 2010a, 'Precision and Recall', in C Sammut & GI Webb (eds), *Encyclopedia of Machine Learning*, Springer US, Boston, MA, pp. 781-.

Ting, KM 2010b, 'Confusion Matrix', in C Sammut & GI Webb (eds), *Encyclopedia of Machine Learning*, Springer US, Boston, MA, pp. 209-.

Transurban 2022, *Corporate Report for the year ended 30 June 2022*, viewed 25/5/23, <https://online.flippingbook.com/view/493118867/170/>.

'True Positive', 2010, in C Sammut & GI Webb (eds), *Encyclopedia of Machine Learning*, Springer US, Boston, MA, pp. 999-.

Yu, Q-m, Zhou, H-l, Wang, Y-h & Duan, R-x 2016, 'Quality monitoring of metro grouting behind segment using ground penetrating radar', *Construction and Building Materials*, vol. 110, pp. 189-200.

Yu, S-N, Jang, J-H & Han, C-S 2007, 'Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel', *Automation in Construction*, vol. 16, no. 3, pp. 255-61.

Appendices

Appendix A Equipment Photos

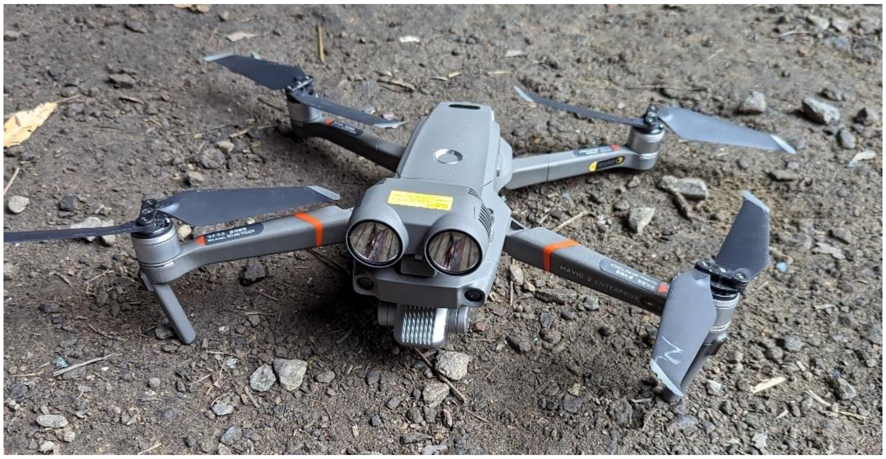


Figure 8. Mavic 2 Enterprise Advanced with spotlight attached at tunnel site

Figure 9. Milwaukee 12V LED torch image from(Milwaukee 2021)

Figure 10. Mavic 2 Pro Drone image from(DJI 2023a)

Appendix B Annotation and Model Training Software

Figure 11. Example of low resolution, low lighting image

Figure 12. Example of low resolution, low lighting image with -20% contrast and +40% brightness

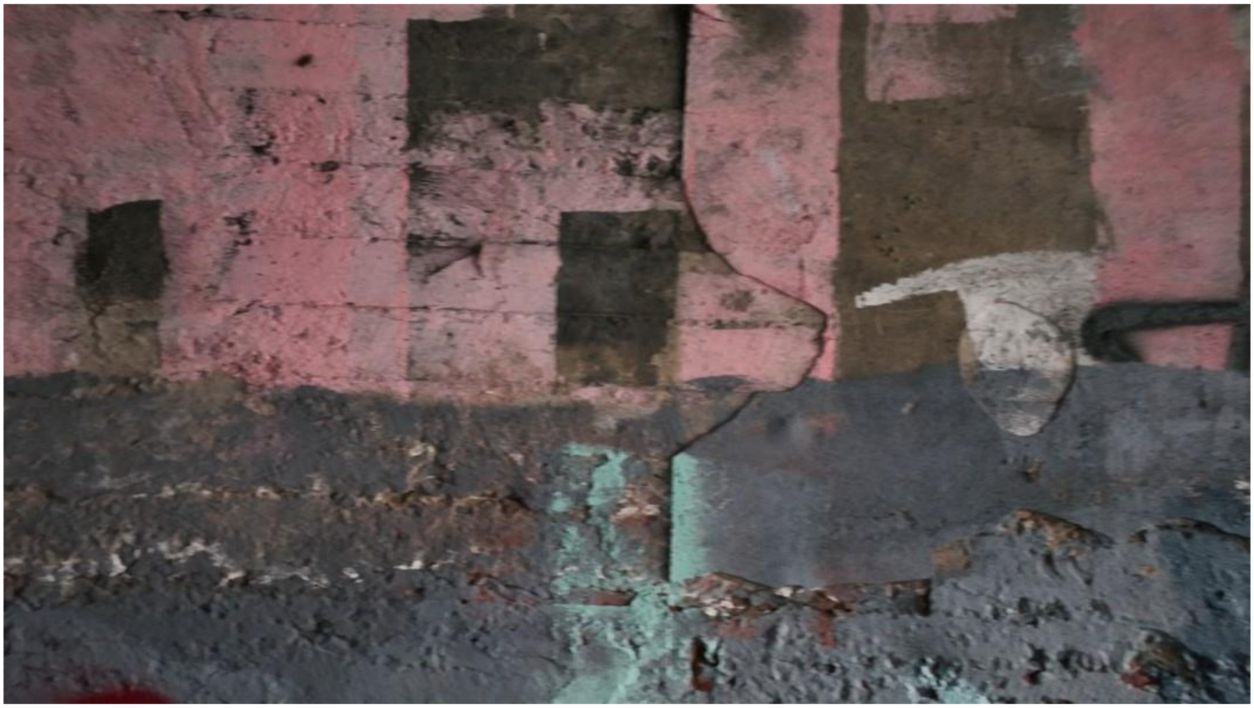


Figure 13. Example of medium resolution, high lighting image

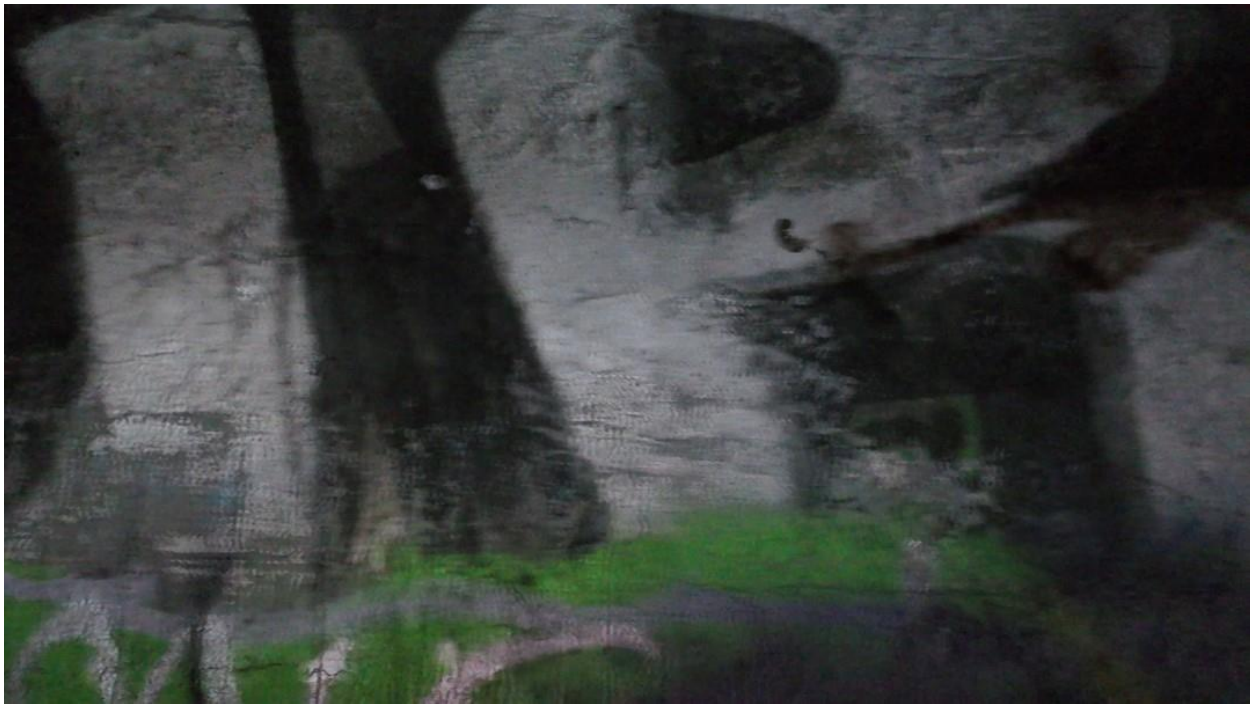


Figure 14. Example of low resolution, medium lighting image

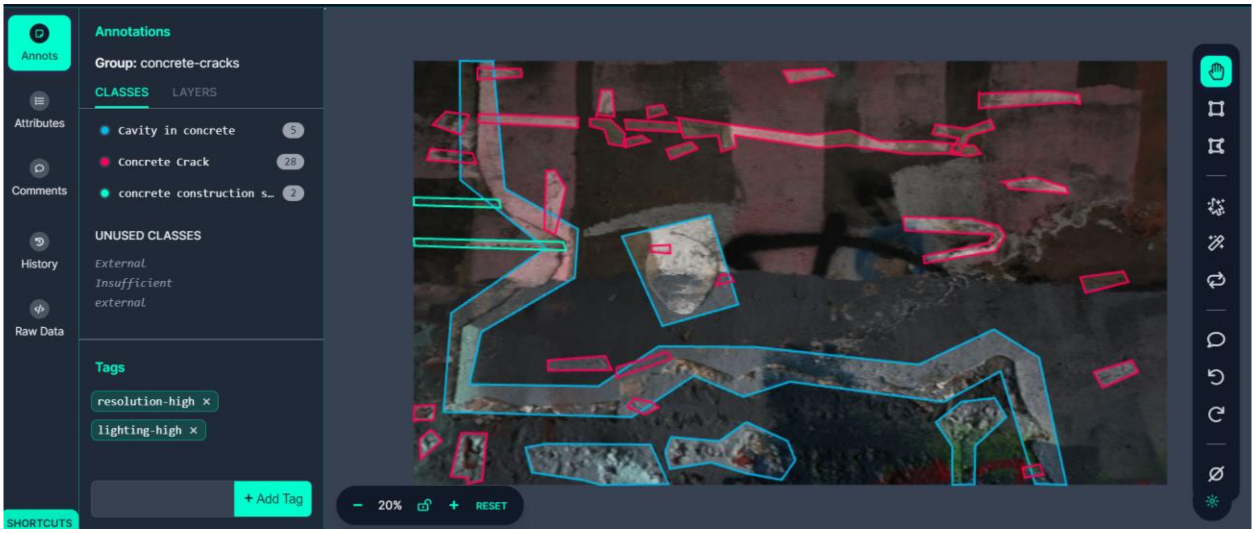
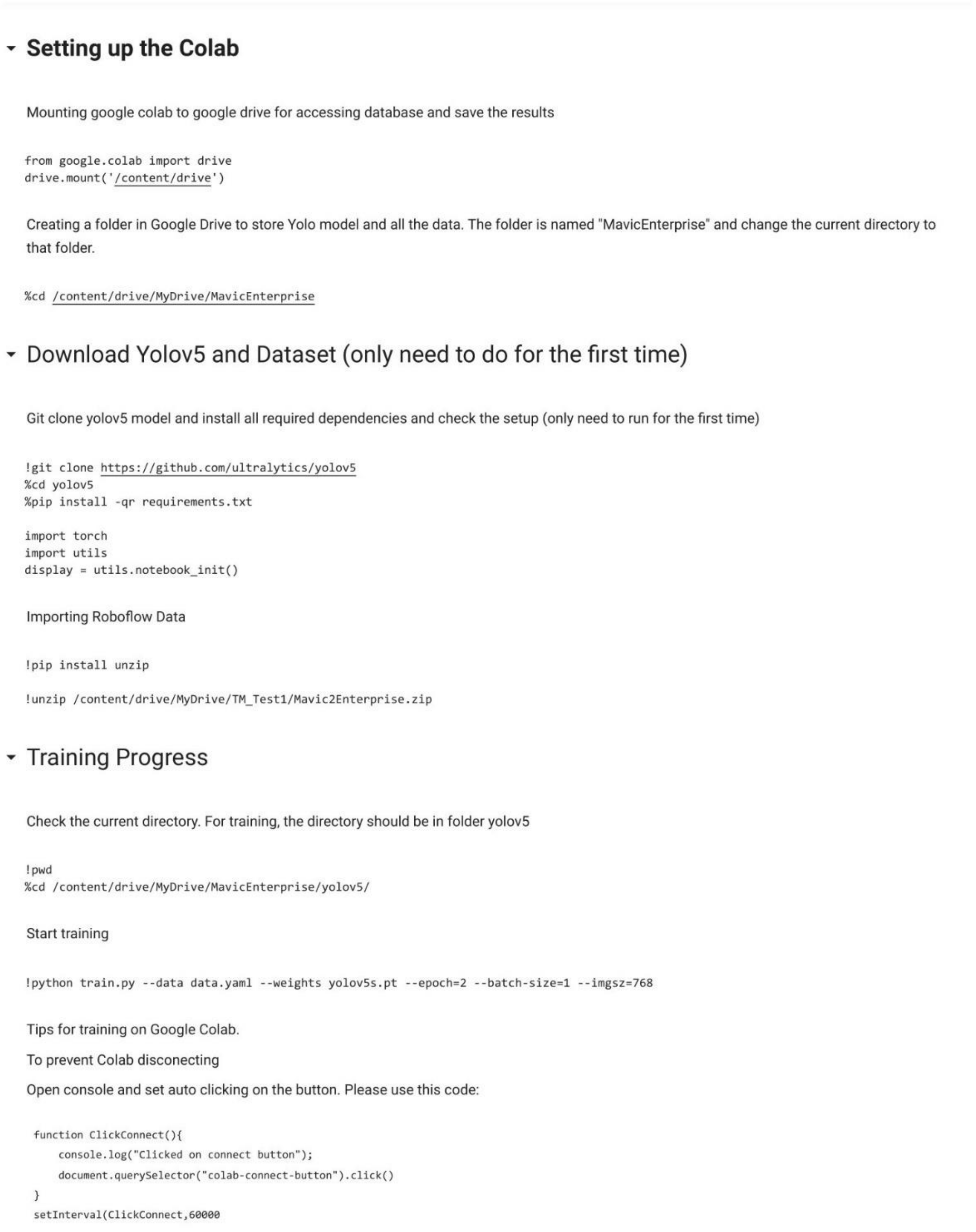


Figure 15. Example of annotated image in Roboflow



Setting up the Colab

Mounting google colab to google drive for accessing database and save the results

```
from google.colab import drive
drive.mount('/content/drive')
```

Creating a folder in Google Drive to store Yolo model and all the data. The folder is named "MavicEnterprise" and change the current directory to that folder.

```
%cd /content/drive/MyDrive/MavicEnterprise
```

Download Yolov5 and Dataset (only need to do for the first time)

Git clone yolov5 model and install all required dependencies and check the setup (only need to run for the first time)

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt

import torch
import utils
display = utils.notebook_init()
```

Importing Roboflow Data

```
!pip install unzip
```

```
!unzip /content/drive/MyDrive/TM_Test1/Mavic2Enterprise.zip
```

Training Progress

Check the current directory. For training, the directory should be in folder yolov5

```
!pwd
%cd /content/drive/MyDrive/MavicEnterprise/yolov5/
```

Start training

```
!python train.py --data data.yaml --weights yolov5s.pt --epoch=2 --batch-size=1 --imgsz=768
```

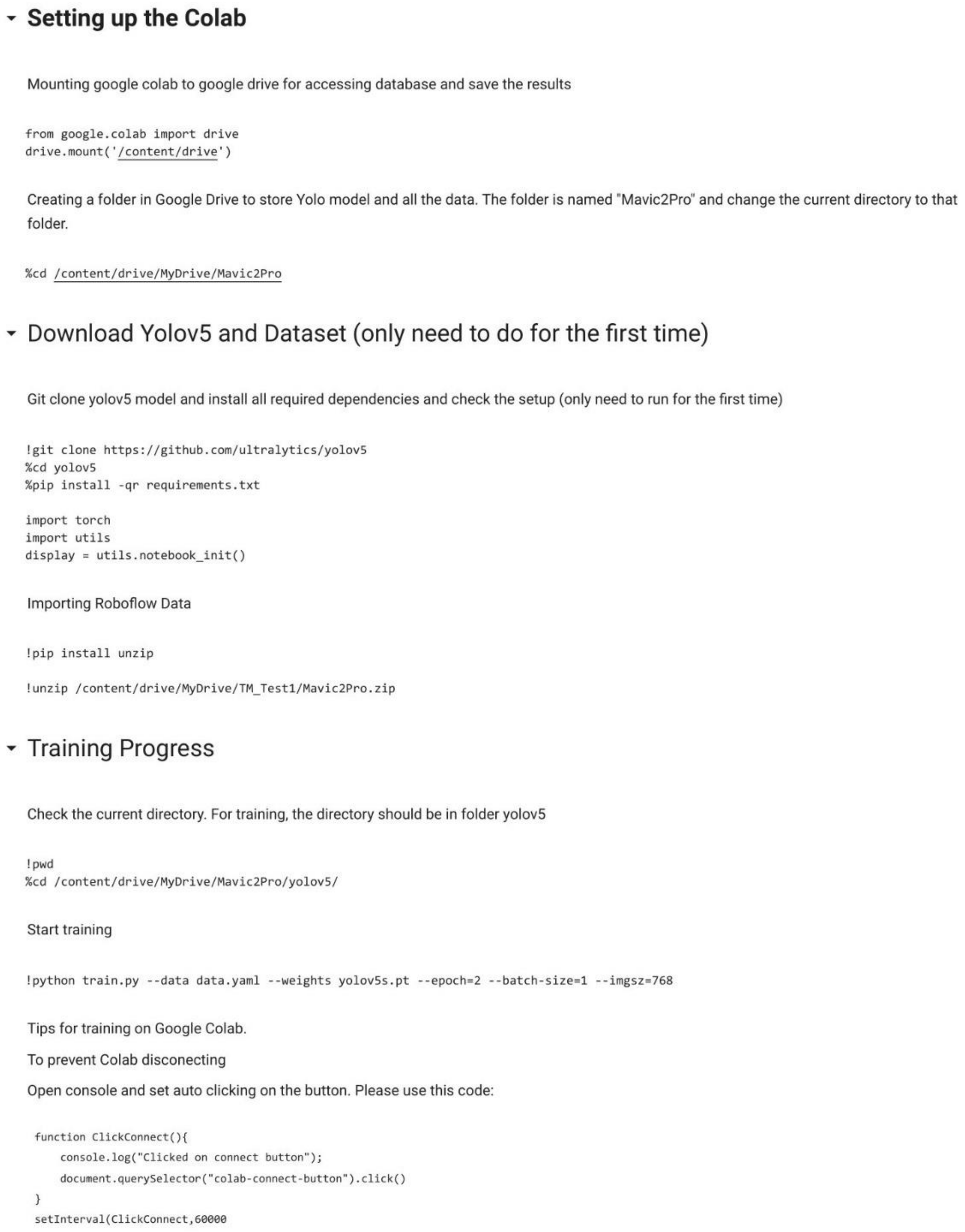
Tips for training on Google Colab.

To prevent Colab disconecting

Open console and set auto clicking on the button. Please use this code:

```
 function ClickConnect(){
     console.log("Clicked on connect button");
     document.querySelector("colab-connect-button").click()
 }
 setInterval(ClickConnect,60000
```

Figure 16. Google Colabs program for modelling training using Mavic Enterprise images.



Setting up the Colab

Mounting google colab to google drive for accessing database and save the results

```
from google.colab import drive
drive.mount('/content/drive')
```

Creating a folder in Google Drive to store Yolo model and all the data. The folder is named "Mavic2Pro" and change the current directory to that folder.

```
%cd /content/drive/MyDrive/Mavic2Pro
```

Download Yolov5 and Dataset (only need to do for the first time)

Git clone yolov5 model and install all required dependencies and check the setup (only need to run for the first time)

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt

import torch
import utils
display = utils.notebook_init()
```

Importing Roboflow Data

```
!pip install unzip
```

```
!unzip /content/drive/MyDrive/TM_Test1/Mavic2Pro.zip
```

Training Progress

Check the current directory. For training, the directory should be in folder yolov5

```
!pwd
%cd /content/drive/MyDrive/Mavic2Pro/yolov5/
```

Start training

```
!python train.py --data data.yaml --weights yolov5s.pt --epoch=2 --batch-size=1 --imgsz=768
```

Tips for training on Google Colab.

To prevent Colab disconecting

Open console and set auto clicking on the button. Please use this code:

```
function ClickConnect(){
    console.log("Clicked on connect button");
    document.querySelector("colab-connect-button").click()
}
setInterval(ClickConnect,60000
```

Figure 17. Google Colabs image of program for modelling training using Mavic Pro images.

Appendix C Model Validation Outputs

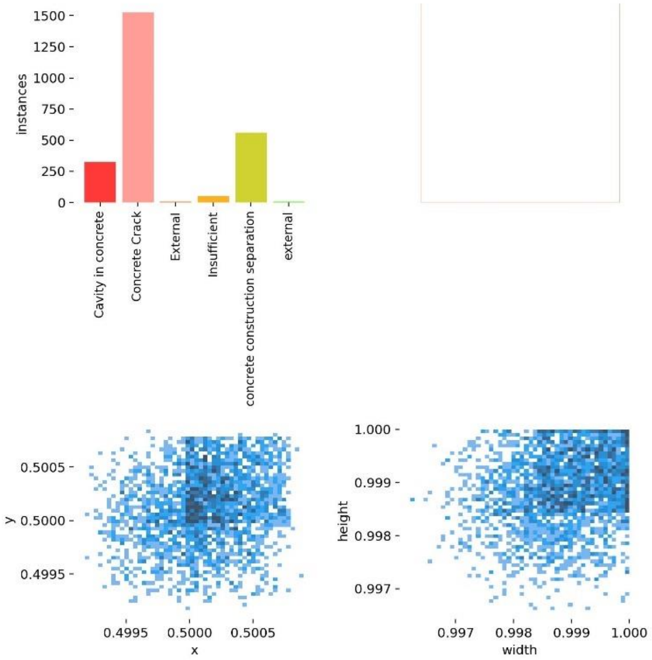


Figure 18. Mavic Pro Model Validation Output: Label Frequency

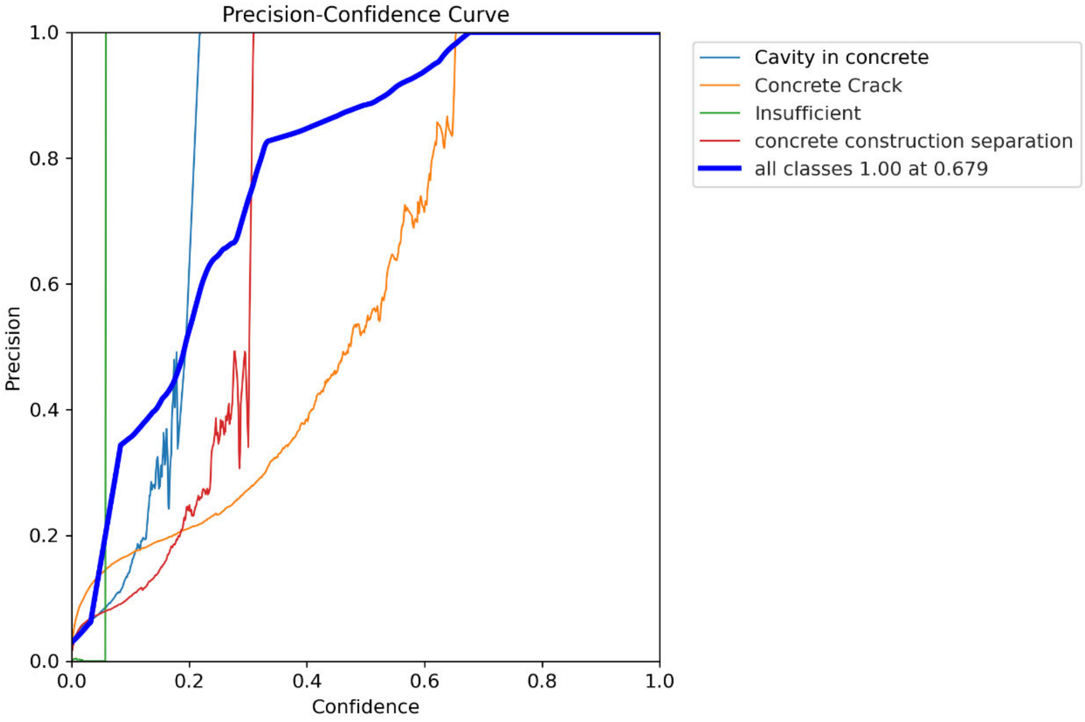


Figure 19. Mavic Pro Model Validation Output: Precision-Confidence Curve

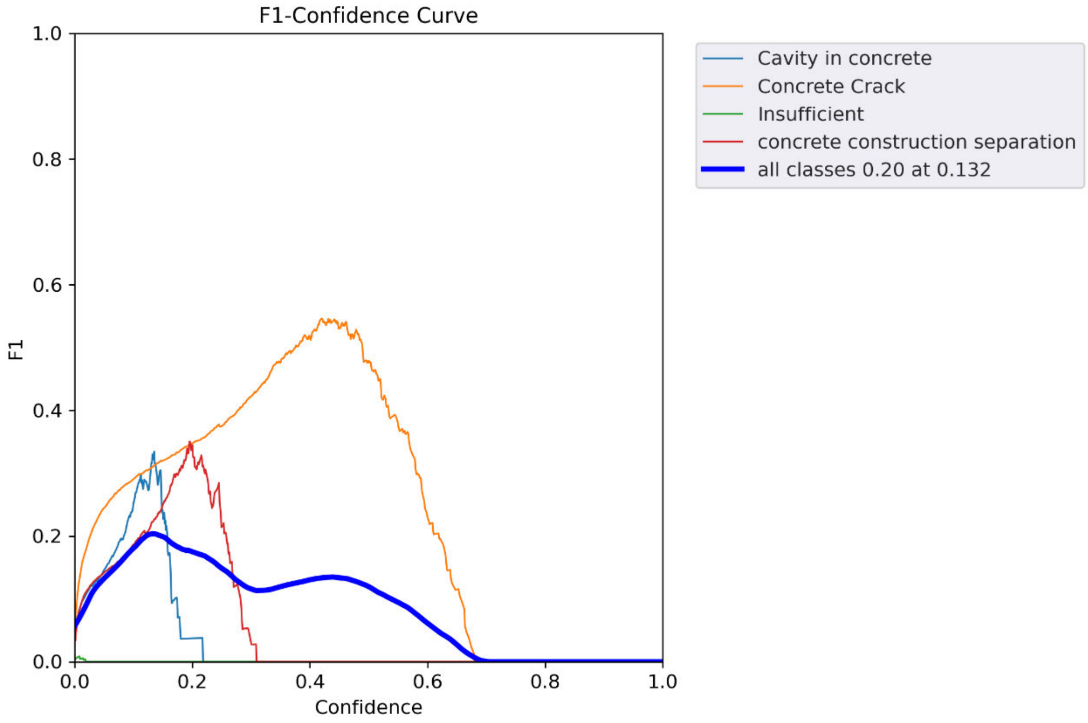


Figure 20. Mavic Pro Model Validation Output: F1-Confidence Curve

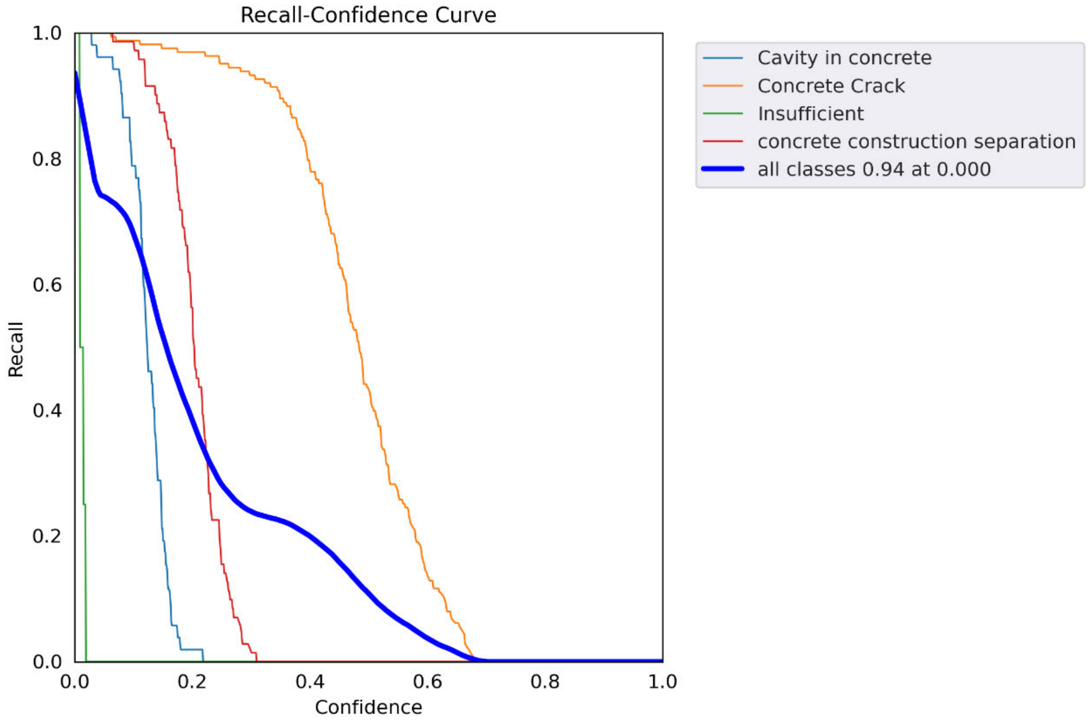


Figure 21. Mavic Pro Model Validation Output: Recall-Confidence Curve

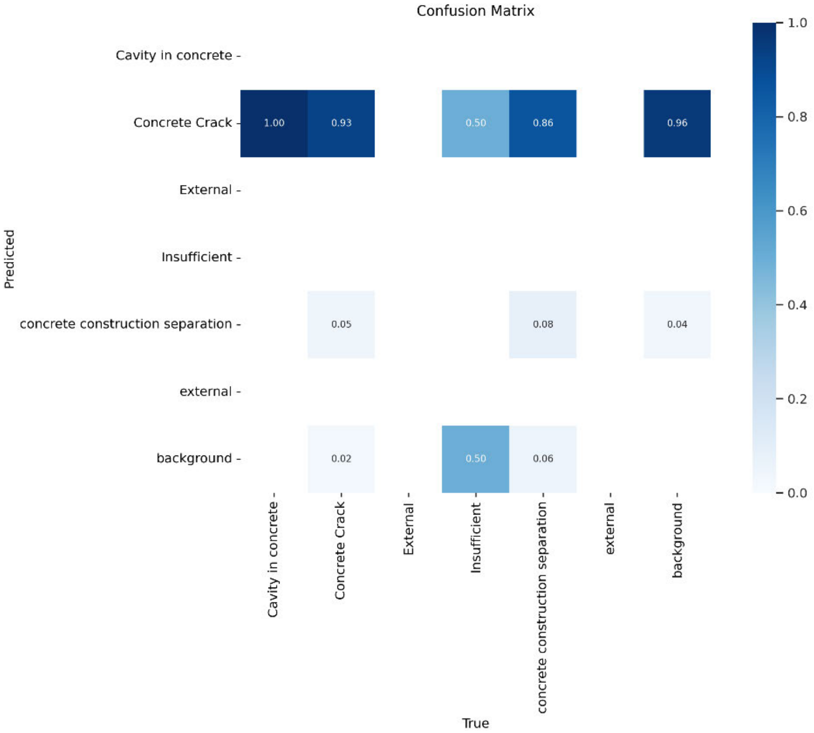


Figure 22. Mavic Pro Validation Output: Confusion Matrix

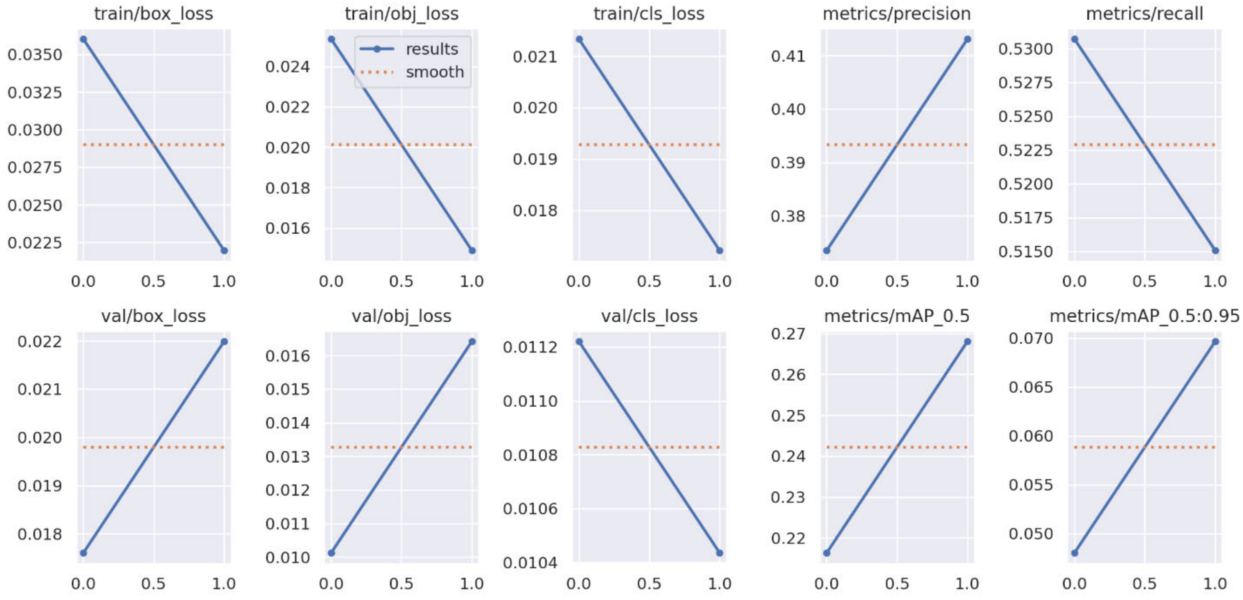
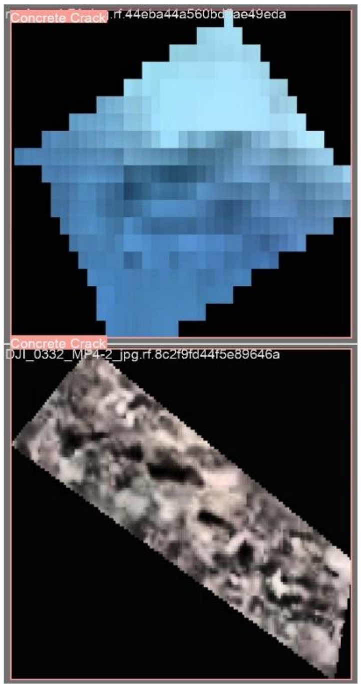


Figure 23. Mavic Pro Model Validation Output: Results Graph


.rf.44eba44a560bd ae49eda
Concrete Crack
Concrete Crack
DJI_0332_MP4-2_jpg.rf.8c2f9fd44f5e89646a

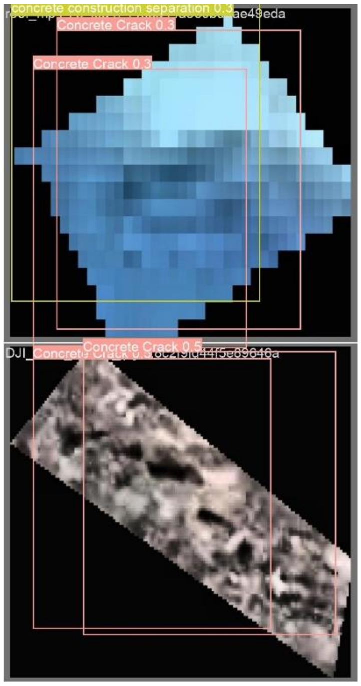
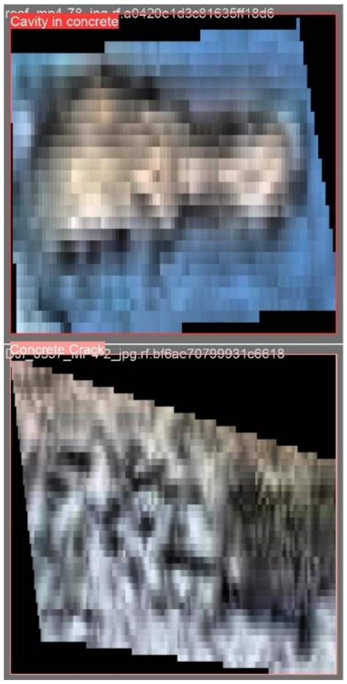


Figure 24. Mavic Pro Model Validation Output: Labeled Image Example 1

Figure 25. Mavic Pro Model Validation Output: Predicted Labels Example 1


roof_mp4_78_jpg.rf.a0420c1d3c81635ff18d6
Cavity in concrete
Concrete Crack
_jpg.rf.bf6ae70799931c6618

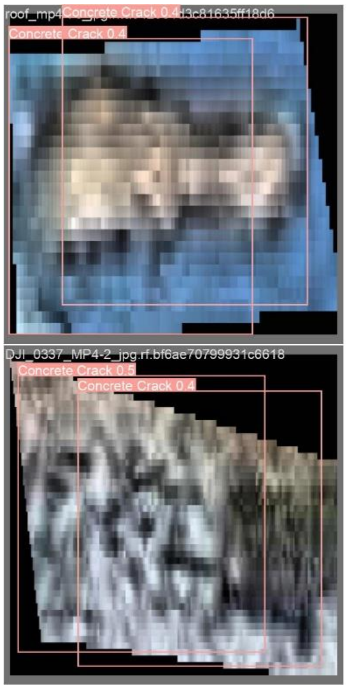


Figure 26. Mavic Pro Model Validation Output: Labeled Image Example 2

Figure 27. Mavic Pro Model Validation Output: Predicted Labels Example 2

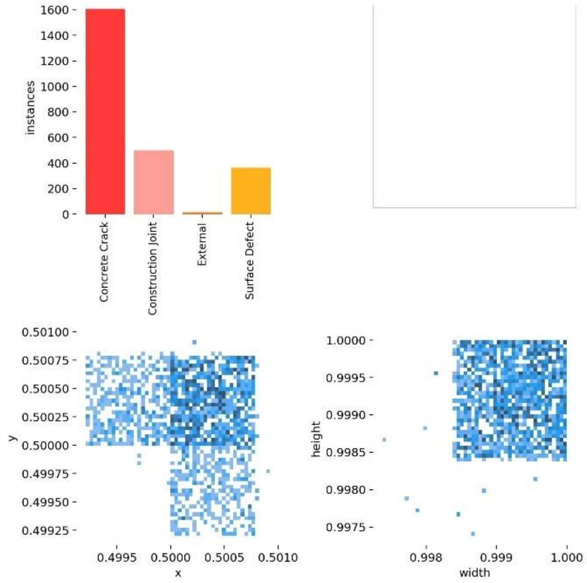


Figure 28. Mavic Enterprise Validation Output: Label Frequency

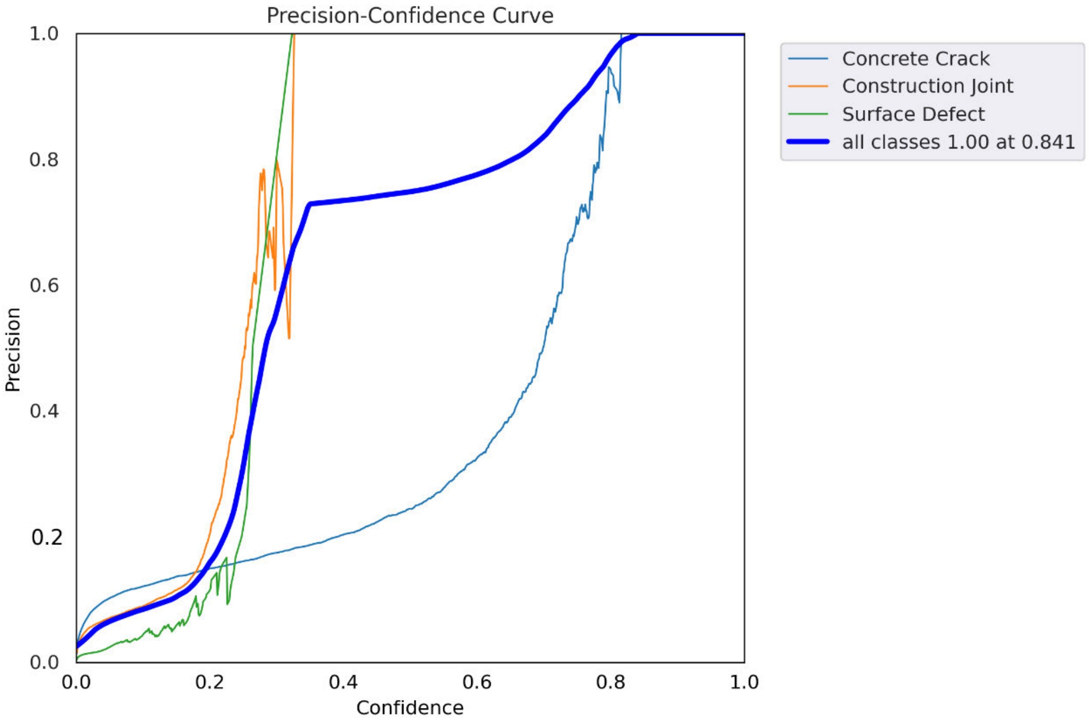


Figure 29. Mavic Enterprise Validation Output: Precision-Confidence Curve

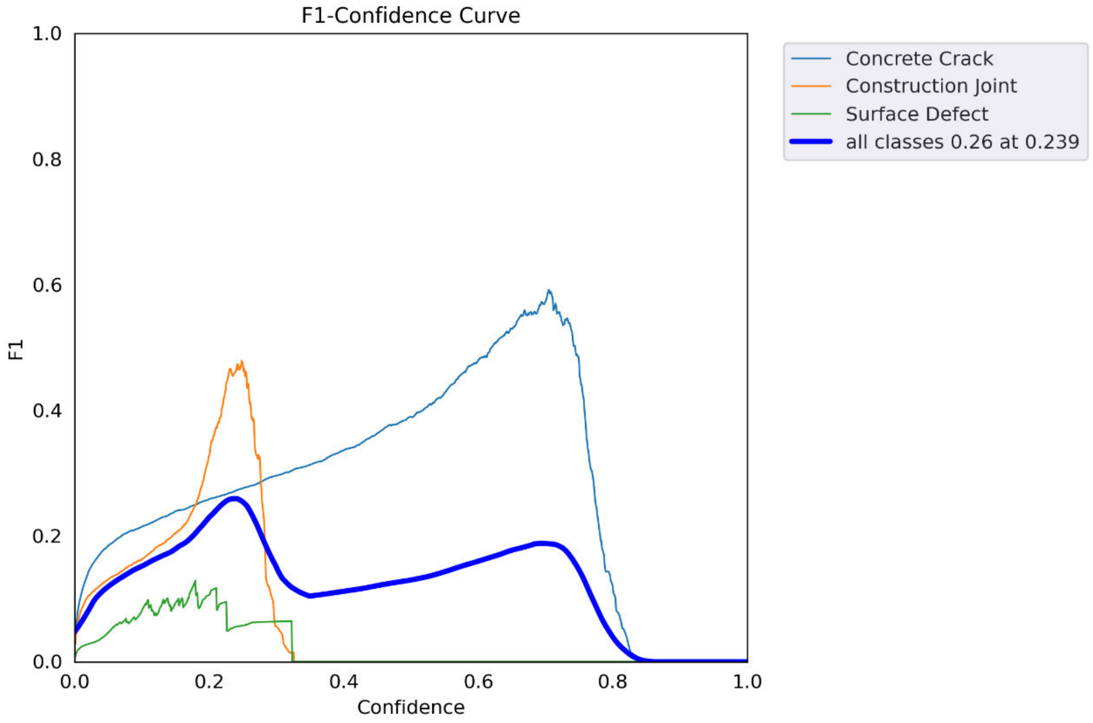


Figure 30. Mavic Enterprise Validation Output: F1-Confidence Curve

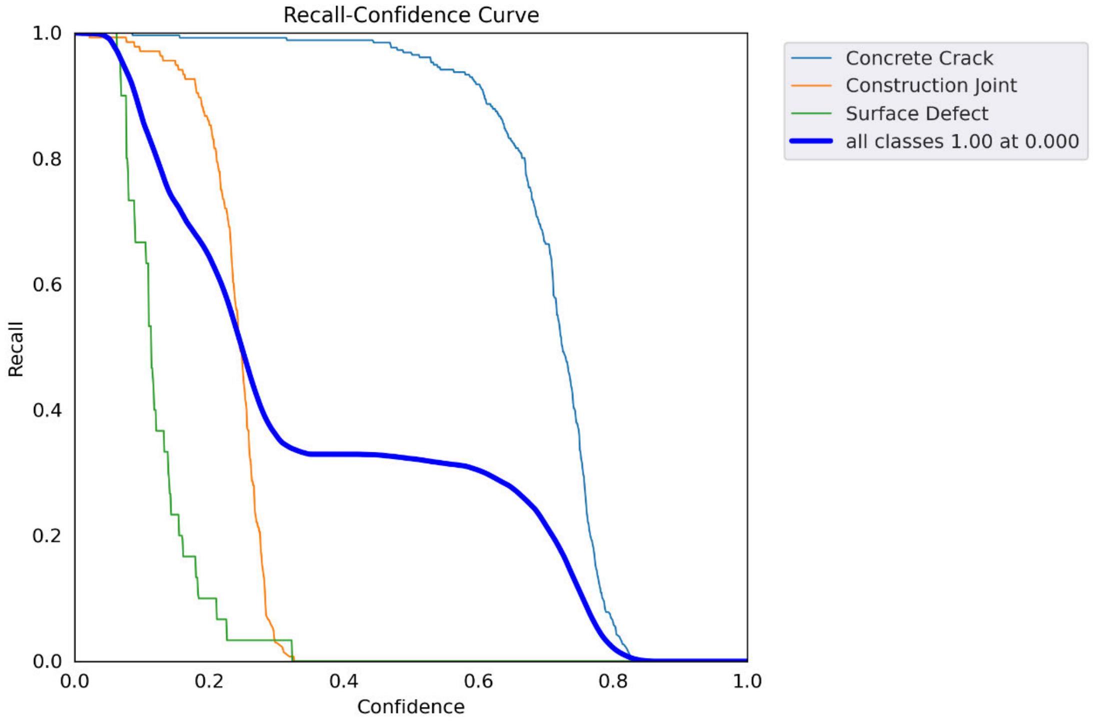


Figure 31. Mavic Enterprise Validation Output: Recall-Confidence Curve

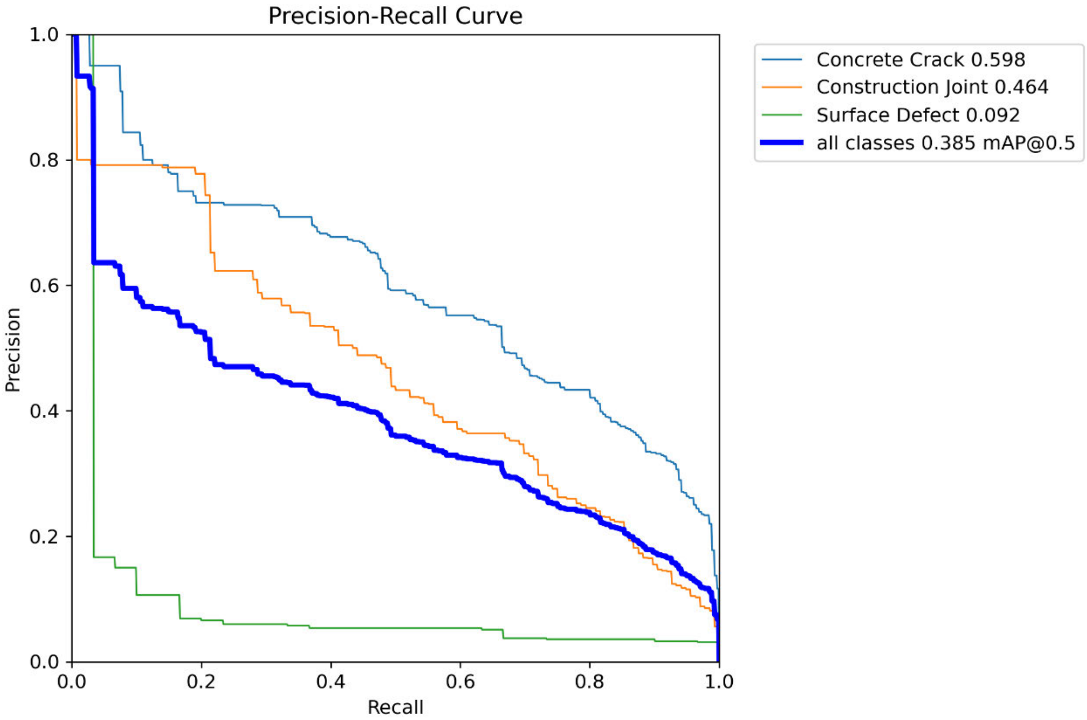


Figure 32. Mavic Enterprise Validation Output: Precision-Recall Curve

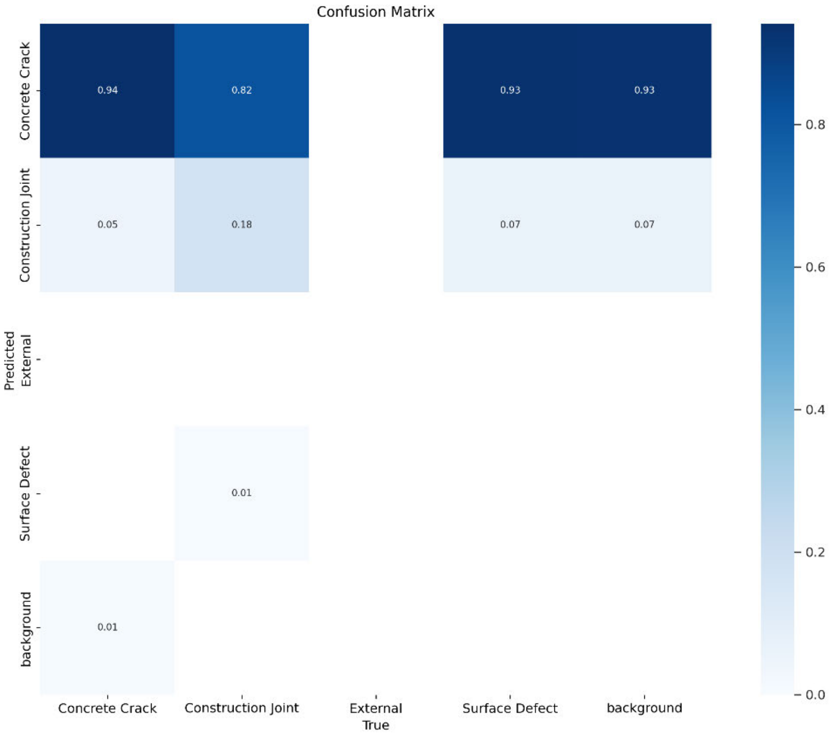


Figure 33. Mavic Enterprise Validation Output: Confusion Matrix

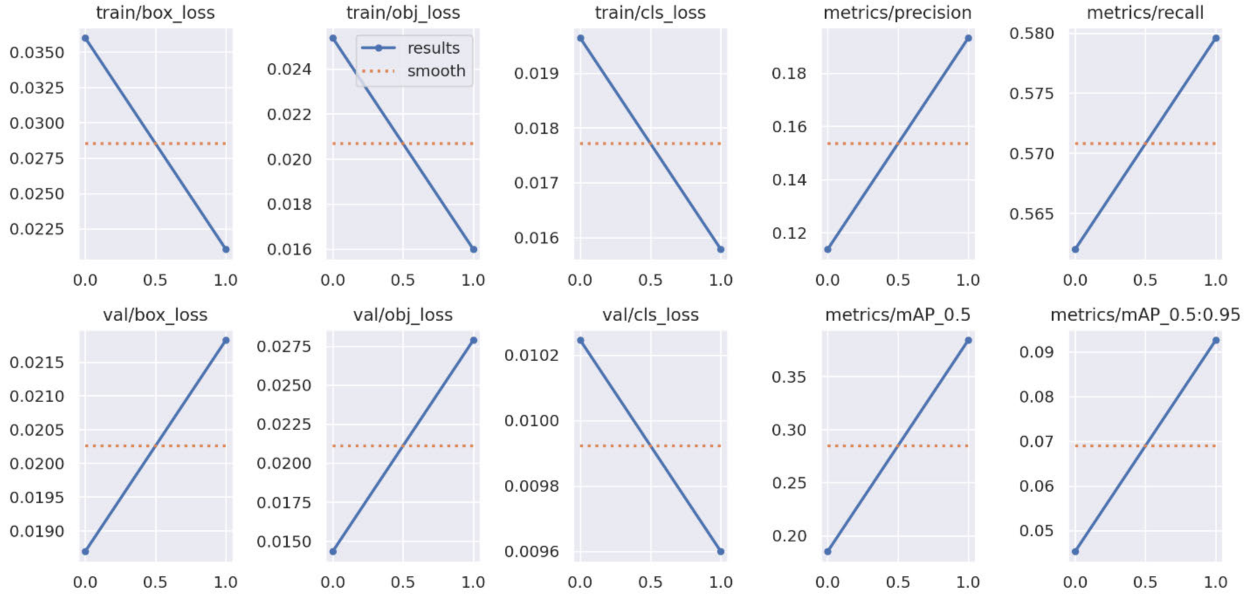
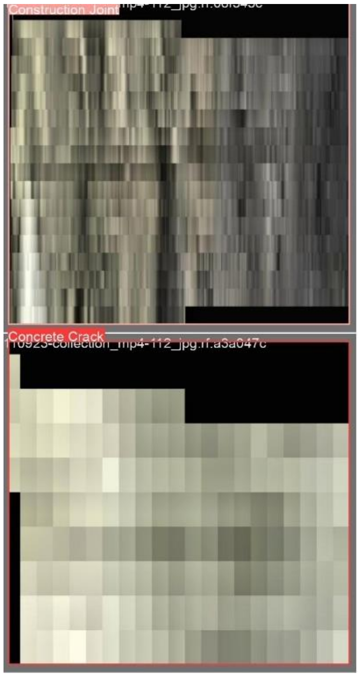


Figure 34. Mavic Enterprise Validation Output: Results Graph



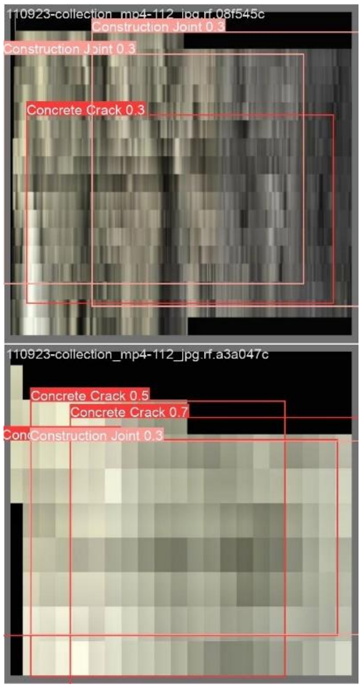
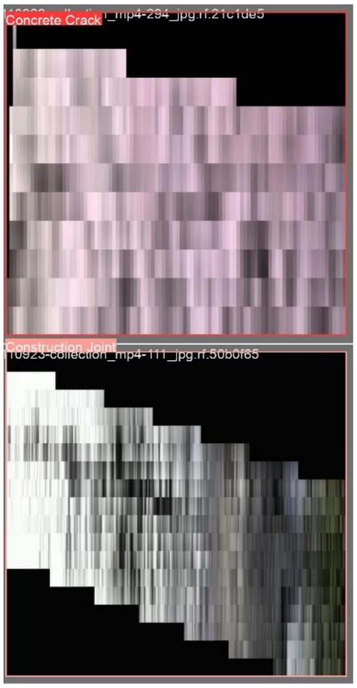


Figure 35. Mavic Enterprise Model Validation Output: Labeled Image Example 1

Figure 36. Mavic Enterprise Model Validation Output: Predicted Labels Example 1



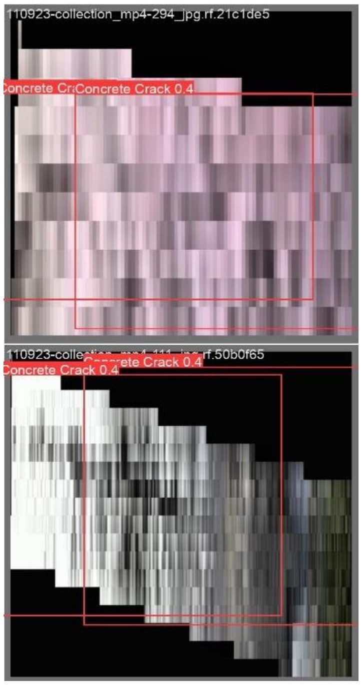


Figure 37. Mavic Enterprise Model Validation Output: Labeled Image Example 2

Figure 38. Mavic Enterprise Model Validation Output: Predicted Labels Example 2

Appendix D Dataset Distribution and Analysis

Table 1. Distribution of Training and Validation Sets

Image Quantities	Mavic 2 Pro	Mavic 2 Enterprise
Training	2454	2478
% Training	84	80
Validation	290	422
% Validation	10	4

Table 2. Image Quality Comparison - Quantity

Mavic 2 Pro		Resolution			
		Low	Medium	High	SUM
Lighting	Low	8	16	0	24
	Medium	4	24	3	31
	High	2	6	13	21
	SUM	14	46	16	76

Mavic 2 Enterprise		Resolution			
		Low	Medium	High	SUM
Lighting	Low	2	7	5	14
	Medium	5	9	4	18
	High	0	5	4	9
	SUM	7	21	13	41

Table 3. Image Quality Comparison - Percentage

Mavic 2 Pro		Resolution			
		Low	Medium	High	SUM
Lighting	Low	11%	21%	0%	32%
	Medium	5%	32%	4%	41%
	High	3%	8%	17%	28%
	SUM	18%	61%	21%	100%

Mavic 2 Pro		Resolution			
		Low	Medium	High	SUM
Lighting	Low	5%	17%	12%	34%
	Medium	12%	22%	10%	44%
	High	0%	12%	10%	22%
	SUM	17%	51%	32%	100%

Table 4. Overall Resolution and Lighting Percentage Comparison

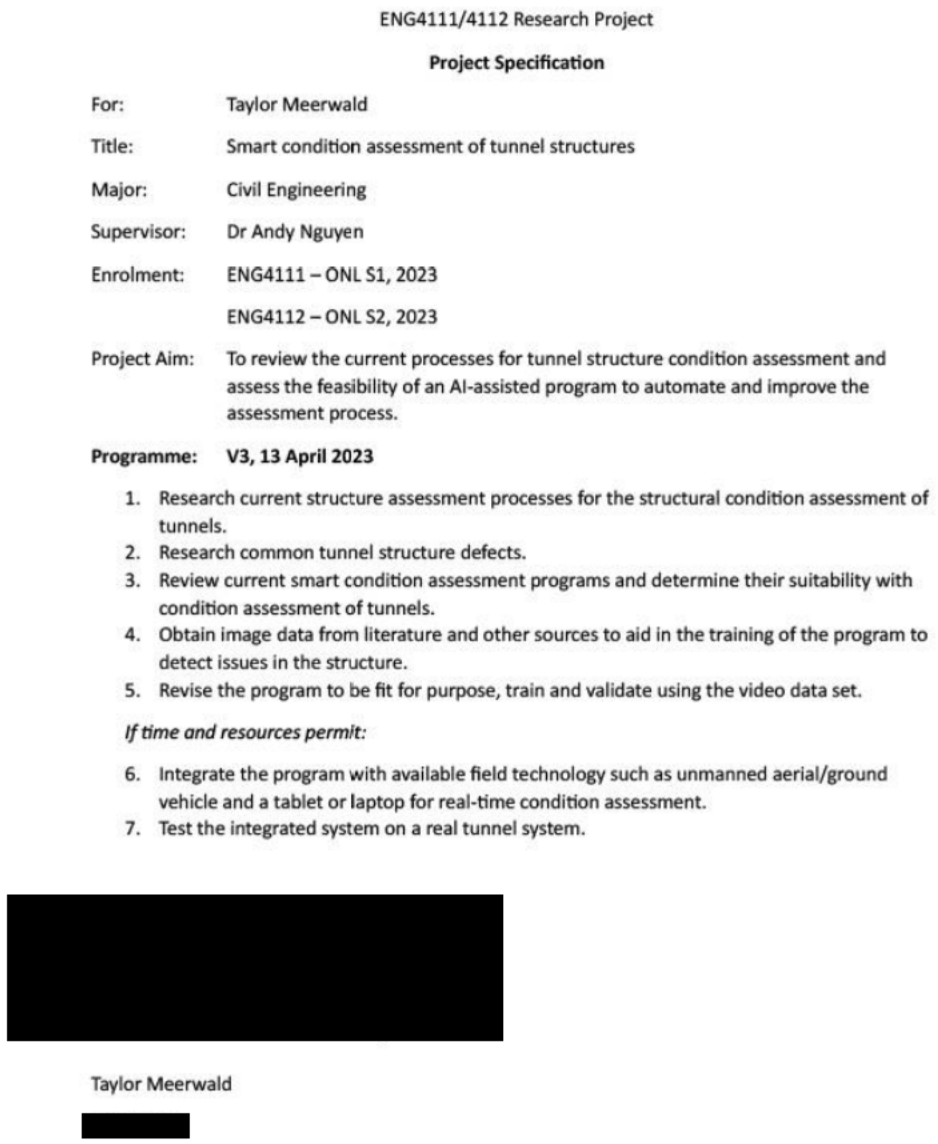
Lighting	Mavic Pro	Mavic Enterprise
Low	32%	34%
Medium	41%	44%
High	28%	22%

Resolution	Mavic Pro	Mavic Enterprise
Low	18%	17%
Medium	61%	51%
High	21%	32%

Table 5. Model Results

	Mavic 2 Pro						Mavic Enterprise					
Class	Images	Instances	P	R	mAP50	mAP50-95	Images	Instances	P	R	mAP50	mAP50-95
All	290	290	40%	51%	27%	7%	422	422	19%	58%	39%	9%
Surface Defect	290	52	28%	19%	25%	4%	422	30	9%	3%	9%	2%
Concrete Crack	290	163	19%	98%	55%	17%	422	256	16%	99%	60%	16%
Construction Joint	290	71	15%	87%	27%	7%	422	136	33%	71%	46%	10%

Appendix E Project Specification



ENG4111/4112 Research Project

Project Specification

For:	Taylor Meerwald
Title:	Smart condition assessment of tunnel structures
Major:	Civil Engineering
Supervisor:	Dr Andy Nguyen
Enrolment:	ENG4111 – ONL S1, 2023 ENG4112 – ONL S2, 2023
Project Aim:	To review the current processes for tunnel structure condition assessment and assess the feasibility of an AI-assisted program to automate and improve the assessment process.
Programme:	**V3, 13 April 2023**

1. Research current structure assessment processes for the structural condition assessment of tunnels.
2. Research common tunnel structure defects.
3. Review current smart condition assessment programs and determine their suitability with condition assessment of tunnels.
4. Obtain image data from literature and other sources to aid in the training of the program to detect issues in the structure.
5. Revise the program to be fit for purpose, train and validate using the video data set.

If time and resources permit:

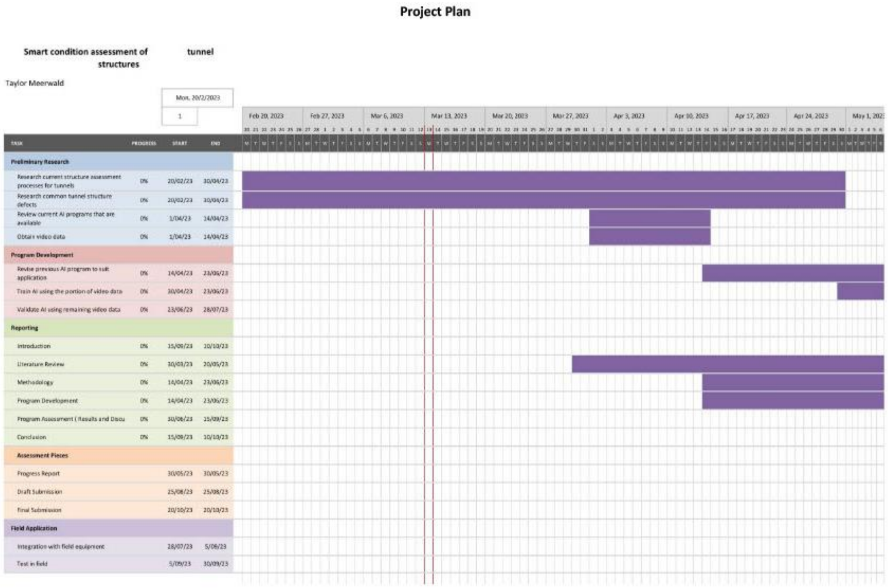
6. Integrate the program with available field technology such as unmanned aerial/ground vehicle and a tablet or laptop for real-time condition assessment.
7. Test the integrated system on a real tunnel system.

[redacted]

Taylor Meerwald

[redacted]

Figure 39. Project Specification Page 1



Project Plan

Smart condition assessment of structures tunnel

Taylor Meerwald

Mon, 20/2/2023

1

TASK	PROGRESS	START	END
Preliminary Research			
Research current structure assessment processes for tunnels	0%	20/02/23	30/04/23
Research common tunnel structure defects	0%	20/02/23	30/04/23
Review current AI programs that are available	0%	1/04/23	14/04/23
Obtain video data	0%	1/04/23	14/04/23
Program Development			
Revise previous AI program to suit application	0%	14/04/23	23/06/23
Train AI using the portion of video data	0%	30/04/23	23/06/23
Validate AI using remaining video data	0%	23/06/23	28/07/23
Reporting			
Introduction	0%	15/09/23	10/10/23
Literature Review	0%	30/03/23	20/05/23
Methodology	0%	14/04/23	23/06/23
Program Development	0%	14/04/23	23/06/23
Program Assessment (Results and Discu	0%	30/06/23	15/09/23
Conclusion	0%	15/09/23	10/10/23
Assessment Pieces			
Progress Report		30/05/23	30/05/23
Draft Submission		25/08/23	25/08/23
Final Submission		20/10/23	20/10/23
Field Application			
Integration with field equipment		28/07/23	5/08/23
Test in field		5/09/23	30/09/23

Figure 40. Project Specification Page 2

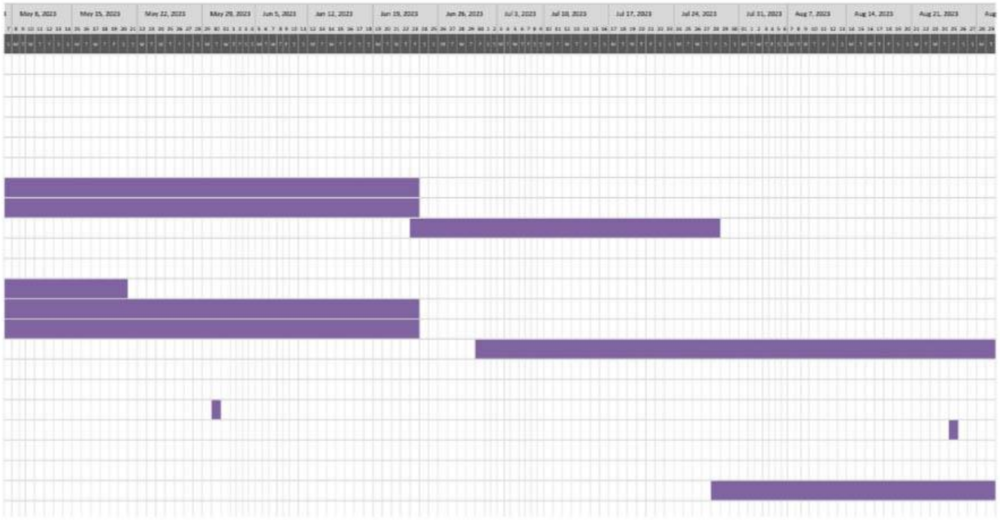


Figure 41. Project Specification Page 3

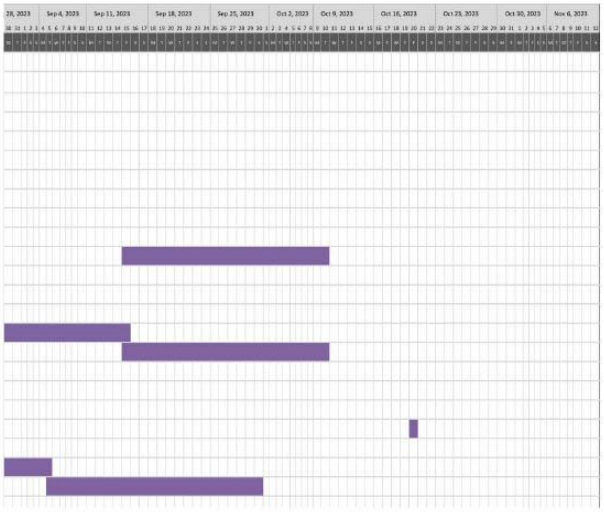
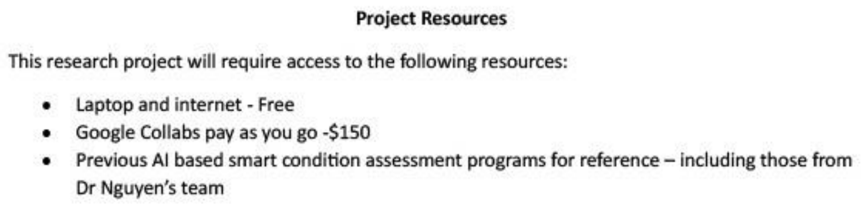


Figure 42. Project Specification Page 4



Project Resources

This research project will require access to the following resources:

- Laptop and internet - Free
- Google Collabs pay as you go -$150
- Previous AI based smart condition assessment programs for reference – including those from Dr Nguyen's team

Figure 43. Project Specification Page 5

Appendix F Resources

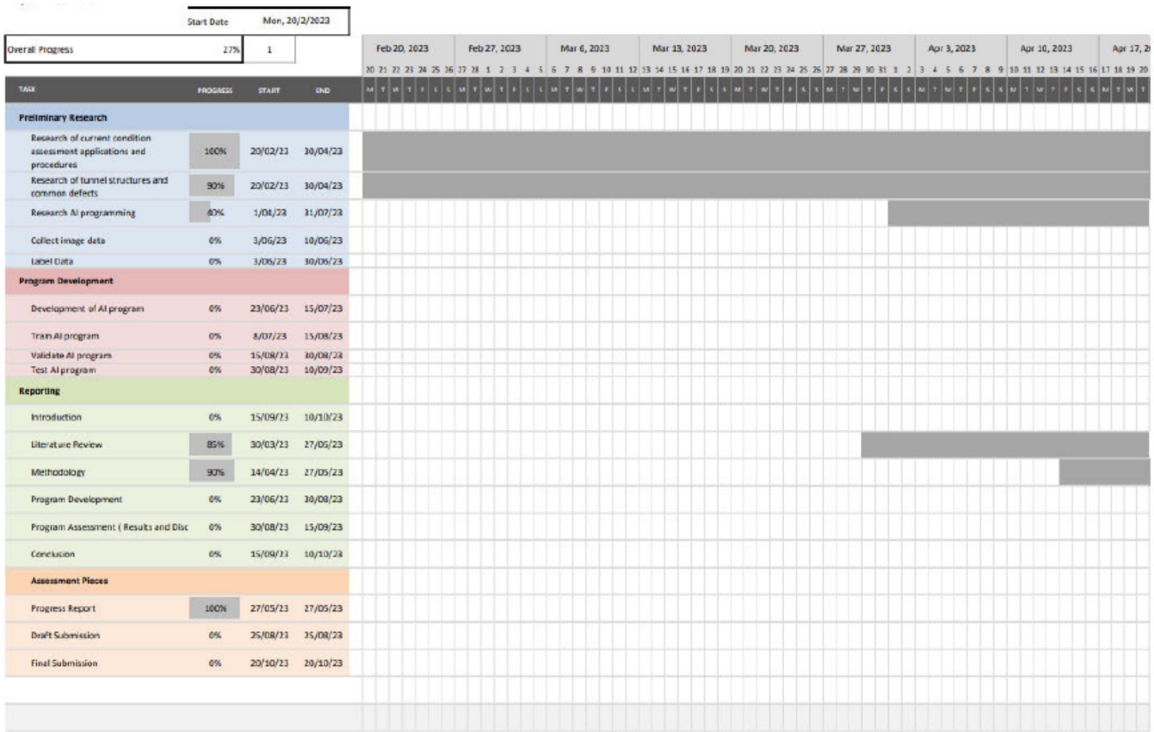
The following resources will be required to complete the research and dissertation:

- For video data capturing:
    - Mavic 2 Pro
    - Milwaukee 12V LED torch
    - Mavic 2 Enterprise Advanced
    - SD card
- For video data labelling:
    - Computer
    - Roboflow
    - Internet connection
- For program writing, testing and final results
    - Computer
    - Google Colabs; $150 for premium access for the period of the project
    - Examples written previously by Mr Long Nguyen (Nguyen 2023)
    - Discussions with and assistance from Mr Long Nguyen and Dr Andy Nguyen for understanding the previous examples and learning how to implement the required alterations for this application
    - Labelled video data for training, validation and testing
    - Possible requirement for video or image data from other sources for severity classification training
- For final results analysis and discussion
    - Computer
    - Microsoft Word for report writing
    - Microsoft Excel for data analysis
    - Endnote for referencing

F.1 Timeline

The project is currently running behind the previous schedule as background research has taken longer than expected. Now that the topic has been clearly defined and researched and the ideas written up in the literature review above, the development of the program can begin and should progress smoothly to complete before the final deadline in ENG4112. The project would benefit from more regular check-ins with the supervisor to ensure that deadlines are met and that the project progresses appropriately.

Figure 44-Figure 46 show the current milestone dates and progress.



	Start Date	Mon, 20/2/2023	
Overall Progress	27%	1	
TASK	PROGRESS	START	END
Preliminary Research			
Research of current condition assessment applications and procedures	100%	20/02/23	30/04/23
Research of tunnel structures and common defects	90%	20/02/23	30/04/23
Research AI programming	40%	1/04/23	31/07/23
Collect image data	0%	3/06/23	10/06/23
Label Data	0%	3/06/23	30/06/23
Program Development			
Development of AI program	0%	23/06/23	15/07/23
Train AI program	0%	8/07/23	15/08/23
Validate AI program	0%	15/08/23	30/08/23
Test AI program	0%	30/08/23	10/09/23
Reporting			
Introduction	0%	15/09/23	10/10/23
Literature Review	85%	30/03/23	27/05/23
Methodology	90%	14/04/23	27/05/23
Program Development	0%	23/06/23	30/08/23
Program Assessment (Results and Disc	0%	30/08/23	15/09/23
Conclusion	0%	15/09/23	10/10/23
Assessment Pieces			
Progress Report	100%	27/05/23	27/05/23
Draft Submission	0%	25/08/23	25/08/23
Final Submission	0%	20/10/23	20/10/23

Figure 44. Project Timeline Part A

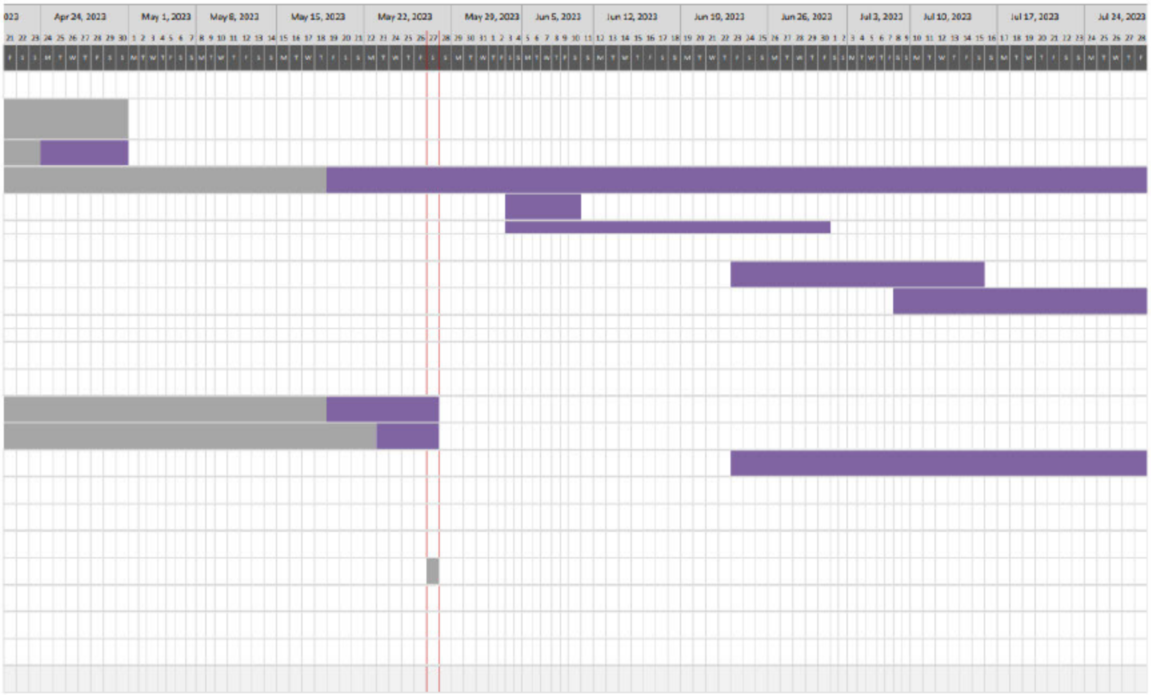


Figure 45. Project Timeline Part B

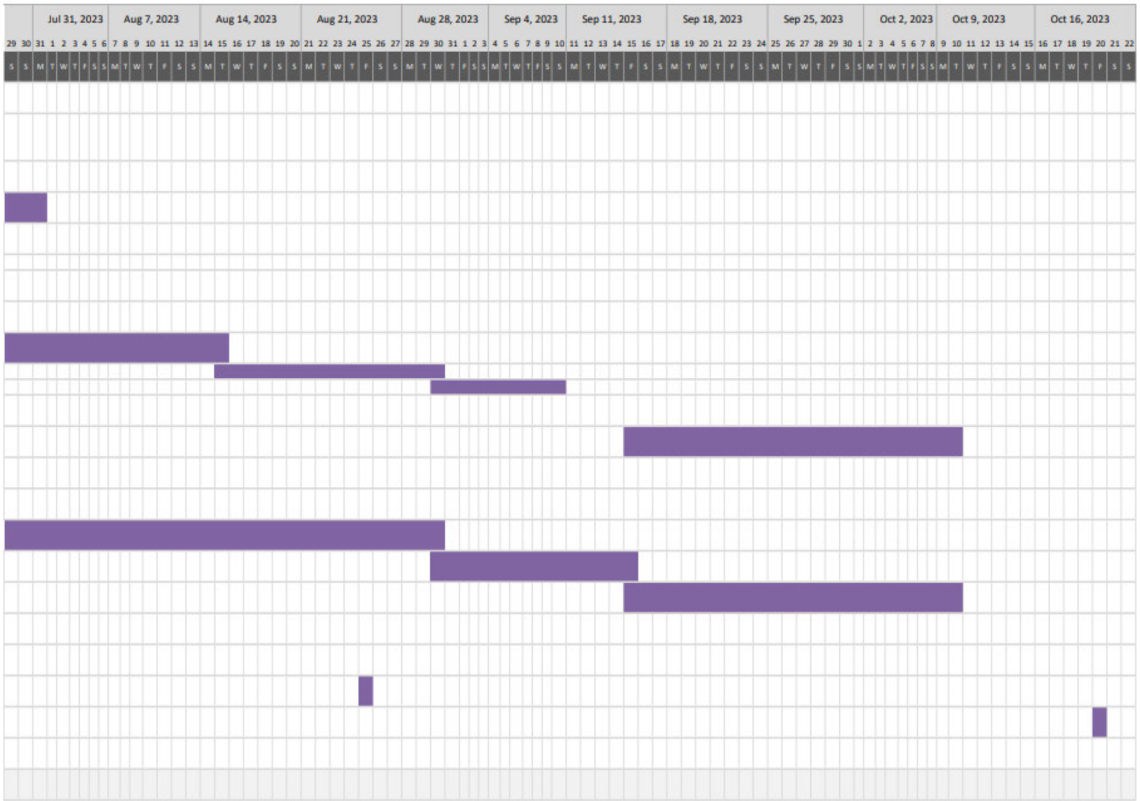


Figure 46. Project Timeline Part C

Appendix G Risk Assessment

As the majority of this project revolves around creating and testing a computer program, there is very little risk involved in the project itself. The main risks are to do with incorrect fitting of the data and insufficient data. The methodology we have adopted has allowed us to accommodate these risks. Other risks may occur when the program is put into use including the user not understanding the limitations of the program or incorrect detection of defects. To negate user error, we will install a disclaimer in the program which will advise the structure and defect types that the program has been trained for. The use of the training, validation and test sets will ensure that the program is sufficiently accurate. However, we will also advise users to do a manual scan through the data for any obvious major defects that the program has missed.

Figure 47 below shows the full risk management plan.

Figure 47. Risk Management Plan