Automatic Aircraft Visual Inspection (AAVI)

A thesis presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours)

Dean Ockenden

BE(Hons) Degree Candidate in Mechatronic Engineering

Academic Supervisor: Mason Brown

University of Technology Sydney

Sydney NSW

27/05/2024

Contents

List of Figures

List of Tables

Abstract

Visual inspection (VI) is a key process employed within aerospace to ensure the overall flight readiness of aircraft. Research on the human visual inspection performance of a Boeing-737 showed that 32% of defects were missed due to time constraints, fatigue and lack of inspector expertise. To improve the reliability and accuracy of this process, the aerospace industry can greatly benefit from automated VI techniques. Deep convolutional neural networks (DCNNs) have been recognised as an especially promising image-processing technique that excels in defect detection.

This research aims to train DCNN models on stationary photographs of aircraft defects to create an automated aircraft visual inspection (AAVI) system. AAVI is composed of two parts. The first specialises in defect detection in real-time (DDRT) using a model based on YOLO-NAS architecture. The DDRT is designed to minimise inference latency and computational complexity to be deployed on the edge. The second model prioritises recall and precision performance to focus on defect detection post-hoc (DDPH). The DDPH adopts a DCNN with neural architecture search (NAS) and hyper-parameter optimisation (HPO) based on the clonal selection algorithm (CSA).

Initial results of the DDRT model show a mAP@50 of 84.67%. F2-score peaks at a confidence level of 47%, achieving a score of 82.09%, with a recall and precision of 81.67% and 83.82%, respectively. These preliminary results of the DDRT system indicate that AAVI could complement manual inspection processes through the reduction of false negatives (FNs) and inspection time to allow human inspectors to dedicate their expertise and limited

attention to problem areas. Future improvements in dataset quality, model performance and computation efficiency may eventually allow for the complete replacement of manual inspection with automated techniques.

Further research is recommended to construct a comprehensive dataset of aircraft defects, the assessment of a viable automated visual inspection framework, including the adoption of AAVI by human inspectors, the use of mobile robotics such as UAVs, and the alignment of an AAVI system with existing regulations.

Acknowledgements

I would like to express huge thanks to my supervisor, Mason Brown, who helped guide me through the entirety of this project offering sage advice and patiently fielding questions.

Chapter 1

Introduction

Automating the aircraft visual inspection process could improve the accuracy and reliability of fault detection in the aerospace industry, preventing accidents and reducing loss of life. AAVI would also take a huge burden of highly stressful, error-prone and labour-intensive work off the plates of line maintenance technicians (LMTs) who are solely responsible for keeping aircraft airworthy between scheduled maintenance. An autonomous solution could perform inspections more frequently than the status quo and use past inspection imagery to learn from its mistakes. There have been recent improvements in the capabilities of computer vision thanks to breakthroughs in deep convolutional neural network (DCNN) design. This makes DCNNs an obvious candidate primed to tackle autonomous visual inspection. Despite this potential, to design an effective AAVI solution, several challenges need to be overcome.

An AAVI system needs to provide actionable insight and analysis of aircraft condition before its subsequent flight. In the world of commercial air travel, where aircraft typically spend only 20-40 minutes at the gate between flights, this feedback needs to be real-time. The inspection system needs to be sufficiently mobile to capture details of intricate control surfaces along with the entirety of the aircraft fuselage and undercarriage. Preliminary research has demonstrated the potential of drones to perform visual inspections much faster than human inspectors. However, the best object detection models require significant computational time and resources, which is inherently limited in mobile robotics. Finally, DCNN models rely on large amounts of high-quality image data to train themselves, which can be challenging to acquire. The quality, diversity, breadth and balance of the dataset are paramount to constructing a model that can detect different types of defects and generalise on unseen data.

This thesis outlines a proposal for an AAVI system comprising a data acquisition network (DAN) and data analysis and sharing hub (DASH). The DAN consists of drone mounted and stationary cameras to inspect the aircraft exterior and undercarriage, respectively. Image data captured by the DAN is analysed locally on small form factor computers by the defect detection in real-time (DDRT) DCNN model based on YOLO-NAS-L architecture. The data is then transferred to the DASH and processed by a second DCNN model for defect detection post-hoc (DDPH). The AAVI system is designed to perform a complete inspection of aircraft condition between flights and distribute improvements generated from any AAVI instance to all other nodes applying the DDRT and DDPH models. The development of the proposed AAVI system in this work is limited to the validation of DDRT model performance.

The developed DDRT model is trained on a dataset curated from a range of publicly available datasets to recognise dents, cracks and missing fasteners on aircraft. The model's performance is compared to three other lightweight DCNN models in the YOLO family and human inspectors. Results show that the DDRT model exhibits the highest recall and F2 score across all of the evaluated YOLO models. The DDRT model also beats the detection ability of human inspectors and has the capability to run real-time deployed on the edge. An in-depth evaluation of DDRT model performance is detailed in section 4.2

1.1 Background

1.1.1 Aircraft Line Maintenance Paradigm

Fault inspection schemes are in place within aerospace (*SAE Standard AS9110*, 2003), but they are not without limitations. In 2014, damage occurred to an ATR 72-600 horizontal stabiliser; the fault went undetected during inspections and was finally identified 13 flights later (Australian Transport Safety Bureau, 2014). Similarly, it has been confirmed that the loss of China Airlines Flight 611 in 2002 was due to fuselage cracks that went unnoticed (Aviation Safety Council, 2002). As prime companies push for automation and progress towards Industry 5.0 (Barata & Kayser, 2023), the main fault detection technique is still visual inspection (VI), performed manually by human inspectors.

Commercial passenger aircraft in the US are subject to comprehensive periodic maintenance that is scheduled into A, B, C and D checks. The most frequent A check occurs after an aircraft has completed between 400-600 flight hours or 200-300 flights and requires up to two days to complete, depending on the aircraft. Between these checks, the airworthiness of aircraft is the sole responsibility of LMTs, who have to perform an extensive array of tasks. General visual inspections (GVIs) require examination of the fuselage and undercarriage to check for any damage caused during flight, commonly by bird or lightning strikes. Hydraulic fluids and oils need to be checked and refilled. Onboard computers and flight sensors need to be tested and cleared of faults. Line replaceable units (LRUs) must be fixed or replaced, including defibrillators, tray tables, toilets, oxygen tanks, tyres, and landing gear. Pilots must be debriefed to gather reports on any in-flight anomalies requiring investigation. Lastly, aircraft may need to be re-positioned and taxied to different gates.

All of these responsibilities need to be completed in the 20-40 minute window that an aircraft has between flights (Stig Aviation, 2023) or during overnight pre-flight servicing, which takes place in a hangar after a plane has accrued between 24-60 flight hours. Furthermore, airline scheduling requires that this work takes place in all weather conditions and occurs around the clock, which can disrupt circadian rhythms. Research by Marais & Robichaud (2012) shows that inspection reliability is highly dependent on inspection conditions, inspector fatigue and LMT expertise. Fault inspection processes can be aided by automation to reduce missed defects, lower inspection time and decrease the burden of work that falls on LMTs.

1.2 Research Objectives

The following research questions were developed after conducting a systematic literature review on aircraft visual inspection and defect detection techniques. The first research question recognises the literature gap in comparing human inspectors' performance to SOTA DCNN defect detection models. Recall (R) denotes the percentage of detected defects compared to total defects present. Precision (P) refers to the percentage of correct predictions divided by the total number of predictions. The second research question aims to bridge one of the four gaps identified in a systematic review of aircraft visual inspection conducted by Yasuda et al. (2022): "the lack of consideration for complete inspection systems in the literature".

Research Question 1

Can a VI system based on modern DCNNs deliver performance in recall and precision comparable to or better than human inspectors?

Research Question 2

Can an automated aircraft visual inspection (AAVI) system be devised to aid or replace manual processes?

1.3 Problem Formulation

Optimisation of aircraft VI can be formulated as optimising function 1.1.

$$\begin{aligned} \max_{-T,A,E} \quad & F(-T, A, E) \\ \text{subject to} \quad & T \leq \text{Ground Time of Aircraft} \end{aligned} \tag{1.1}$$

In this context, T denotes the time taken for VI to take place, constrained by the ground time of aircraft, which is dictated by airline schedules. The variable A signifies accuracy, measured as the percentage of defects detected by the VI process compared to the current accuracy of GVIs during line maintenance. E represents the robustness and reliability of the system under various environmental conditions.

Defects should be recognised immediately after the flight that caused the damage to occur to prevent the endangerment of passengers and crew or mission failure. The inspection process should not interfere with current flight scheduling. Ideally, non-destructive testing (NDT) should be employed. The accuracy and reliability of the system should at least match current VI techniques or the automated system should decrease overall inspection time. Ideally, data gained from the system should allow the proposed solution to be optimised and built upon without a complete overhaul.

Chapter 2

Related Work

2.1 Literature Review

This review assesses the current state of aircraft visual inspection. The review is split into seven sections: 1) current visual inspection techniques, 2) automated means of defect detection, 3) UAV for use in visual inspection, 4) CNN architectures for real-time defect detection, 5) CNN architectures for higher precision defect detection, 6) prior research, and 7) research gaps.

2.1.1 Literature Review Methodology

Using the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology (Page et al., 2021), this review investigates the topic of automated aircraft fault detection. The inclusion criteria were: (i) publication date from 1997 through 2023; (ii) being an empirical study, thesis, or project report; (iii) written in English; and (iv) published in a scholarly peer-reviewed journal or conference proceedings. The search topics included "fault detection" or "visual inspection" or "object detection" or "convolutional neural networks" or "line maintenance" or "line maintenance technicians" or "aircraft maintenance". "fault detection", "convolutional neural networks", or "automated object detection". Ultimately, 22 articles across three databases, The Lens, Google Scholar and UTS Library, were selected for review in this work.

2.1.2 Manual Visual Inspection

The aerospace industry relies heavily on visual inspection (Yasuda et al., 2022). A human reliability analysis of visual inspection by Chen and Huang (2014) found that the five factors contributing most to the efficacy of visual inspection are visual information, fatigue, equipment, detection distance, and job training. Furthermore, the effectiveness of visual inspection is more related to factors involving the inspector rather than what is being inspected (Chen & Huang, 2014). This aligns with research conducted by Wang & Chuang (2014), who found that 21% of all reported incidents in the Aviation Safety Reporting System

(ASRS) are fatigue-related. Research by Drury et al. (1997) examined the effectiveness of 12 professional aircraft inspectors in identifying defects on a Boeing-737, where only 68% were detected. Research by See (2015) found that visual inspectors incorrectly rejected 35% of acceptable precision-manufactured parts. See (2015) indicates an unacceptable level of error in visual inspection, which, if extrapolated to the visual inspection of aircraft, could result in unnecessary maintenance, causing catastrophe. The Federal Aviation Authority (FAA) records conclude that inspection and maintenance errors are within the top three causes of aircraft accidents. Furthermore, accidents resulting from maintenance are 6.5 times more likely to be fatal and result in 3.6 times more fatalities when compared to accidents in general (Marais & Robichaud, 2012). Due to the notable limitations inherent in visual inspection for faults and defects, automating this process has recently garnered significant interest.

2.1.3 Automated Techniques for Defect Detection

Automated techniques for industrial defect detection have shifted from conventional image processing techniques to modern deep convolutional neural networks (DCNNs) (Tulbure et al., 2022). Work by Wang et al. (2020) compared conventional machine learning (ML) against DCNNs and found ML techniques to be better suited to use cases with limited data. DCNNs perform better once the data limitation is overcome. For virtual inspection, Rice et al. (2018) explored using stationary RGB-D cameras for high-quality visual inspection by mapping aircraft onto a 3D model (Rice et al., 2018). While automated techniques are being explored in the literature, aircraft VI is still performed manually.

2.1.4 Unmanned Aerial Vehicle (UAV) for Aircraft Inspection

Work by Sun and Ma (2022) demonstrates the possibility of using drones for aircraft visual inspection. The authors propose a two-stage approach where the UAV follows a predefined path to survey the aircraft at a distance to generate a rough model for optimal path planning much closer to the aircraft surface. A digital experiment indicates that approximately 70% of the aircraft surface can be inspected within an hour. This timeframe aligns with the maximum flight times of modern enterprise drones and the time commercial aircraft often spend at airport gates between flights (Picchi Scardaoni et al., 2021). Research by Liu et al. (2022) validates this concept by using ArUco markers to orient the drone before performing surface defect detection using YOLOv4 CNN architecture. Their results show 87 defects detected by their system compared to 31 by manual inspection. However, their study design may be misleading as the human inspector likely grouped many of the individual defects identified by their system. Most notably, their inspection was achieved in 12 minutes compared to 3.1 hours manually. The use of UAVs for visual inspection of aircraft structures is further explored by Tzitzilonis et al. (2019) and work by Doğru et al. (2020), who employ Mask R-CNN and a pre-classifier approach to achieve an F1-score of 67.5% for fuselage dent detection. Conventional DCNNs require significant computational resources that would not be applicable to a UAV. Additionally, DCNNs such as Mask R-CNN adopted by Doğru et al. (2020) exhibit latency that makes them unsuitable for real-time object detection (Huang et al., 2016).

2.1.5 CNN Architectures for Real-Time Defect Detection

The YOLO CNN architecture addresses the need for real-time defect detection due to its low computational complexity and latency (Terven et al., 2023). Variations of this architecture have inspired works related to automated defect detection in aerospace (Li et al., 2019; Qu et al., 2023). Mask R-CNN is another architecture that has gained attention for defect detection (Doğru et al., 2020; Ding et al., 2022) and can apply instance segmentation. De-

spite this, Mask R-CNN is far more resource-intensive than YOLO-based architectures and cannot be processed quickly enough for real-time use cases. Faster R-CNN exhibits lower latency than Mask R-CNN and has similar capabilities (Aleksandr G., 2022), although it exhibits significantly higher latency than YOLO-based architectures. The YOLO architecture has proven effective for drone applications (Li et al., 2019; Jung & Choi, 2022; Li et al., 2023). Finally, Deci (2023) built upon prior YOLO architectures with YOLO-NAS, a YOLO-based DCNN architecture with neural architecture search (NAS) capability. Deci's model demonstrates improved mAP scores compared to other YOLO-based models while further reducing latency (Deci, 2023). A state-of-the-art (SOTA) YOLO-based architecture deployed on drones could be an effective solution for real-time defect detection on aircraft.

2.1.6 CNN Architectures for High-Precision Real-Time Detection

Malekzadeh et al. (2017) proposed an aircraft defect detection deep learning model trained using high-resolution 20-megapixel images. Their model is based on the VGG-F architecture to produce feature maps, which are classified using a linear support vector machine (SVM), achieving accuracy greater than 96%. Several works find that the performance of DCNN architectures is highly dependent upon the dataset (Bressem et al., 2020; Yang et al., 2021; Zhao et al., 2022). Work by Al Bataineh et al. (2023) proposes the clonal selection algorithm (CSA) with NAS that can automatically generate an optimised DCNN architecture for each specific use case. An implementation of the Al Bataineh et al. CSA significantly outperforms other SOTA DCNN models on the EMNIST dataset. DCNN architectures generated by the CSA may be well-suited to a model prioritising high-precision defect detection.

2.1.7 Prior Research

Doğru et al. (2020) explore aircraft maintenance visual inspection, implying drone image capture. Despite this, the work only focuses on fuselage dents. A pre-classifier approach powered by an SVM is used to determine whether a dent is present within an image, achieving a test F1 score of 88.7% despite a dataset of only 56 images before augmentation. The work demonstrates proof of concept; unfortunately, the Mask R-CNN model is too computationally complex for real-time object detection on a mobile robot such as a drone. Li et al.'s 2019 work proposes YOLOv3-Lite as a lightweight CNN architecture for crack detection in the aircraft structure. Their method combines feature pyramids and YOLOv3 with a modified backbone that employs depthwise separable convolutions to reduce latency by more than 50% compared to YOLOv3. The model performs detection in the image in 0.1s with an average precision (AP) of 38.7%. Qu et al. (2023) use a modified YOLOv5 model to study surface defects. Their model replaces the C3 module with C3-Faster based on the FasterNet network, which results in more than a 10% improvement in AP, parameter quantity and weight volume compared to standard YOLOv5. Liu et al. (2022) validate the viability of using drones for visual inspection by orienting the drone using fiducial markers before performing visual inspection on various aircraft. Their system reduced manual inspection time by 93.5% (Liu et al., 2022).

2.1.8 Research Gaps

In conclusion, the transition from manual to automated aircraft VI is underway. The findings of this review align with research conducted by Yasuda et al. (2022), identifying several key gaps in the research. Namely, there is a lack of a complete aircraft VI system and insufficient validation of the performance of automated VI techniques compared to human inspectors. There is a clear indication of certain promising solutions in the literature, especially if synthesised. These technologies include droned-based visual inspection utilising DCNNs with a YOLO-based architecture for real-time defect detection. The CSA also represents a SOTA solution to developing DCNN architectures that excel in recall and precision. In summary, the literature suggests that an intelligent drone-based aircraft inspection system can lead to more accurate, reliable, time and cost-efficient validation of aircraft condition that negates the limitations of current manual processes.

Chapter 3

Methodology

3.1 Automated Aircraft Visual Inspection (AAVI)

The AAVI system takes advantage of the speed, reliability and scalability of digital systems to focus on the reduction of inspection time and false negatives while improving the reliability and repeatability of defect detection.

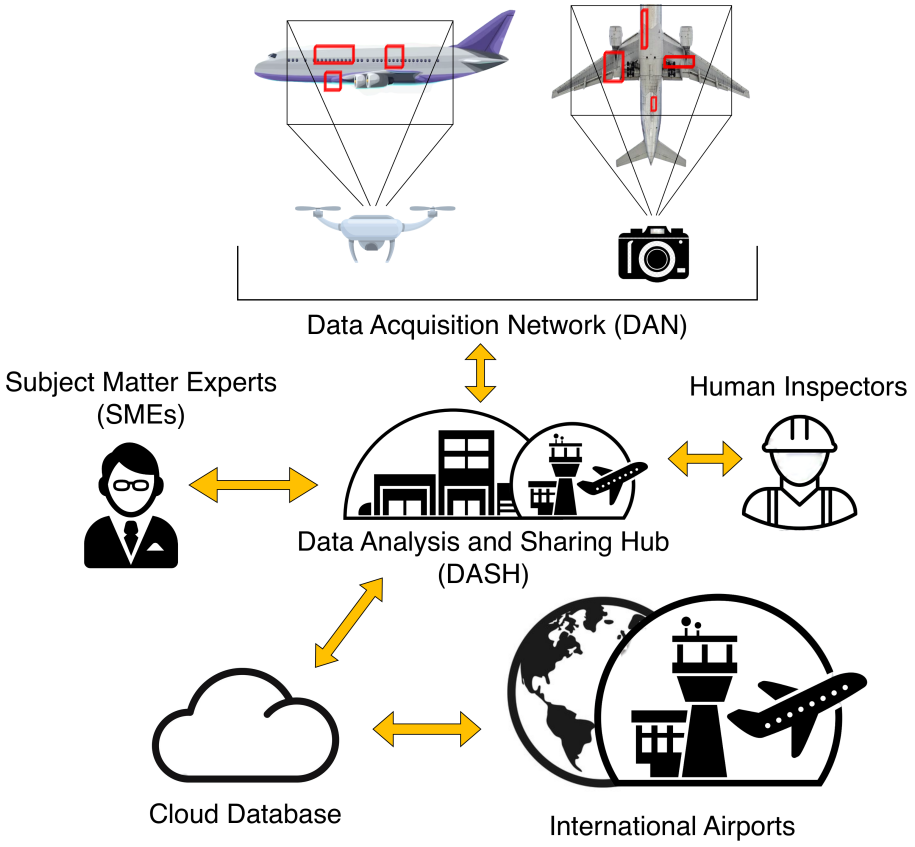


Figure 3.1: AAVI System Diagram

3.1.1 Data Acquisition Network (DAN)

To minimise inspection time T, AAVI's DAN consists of two parts:

- UAVs scanning the outer and upper aircraft frame
- Stationary cameras imaging the aircraft's undercarriage and landing gear

Images from the DAN will be streamed directly to the data analysis and sharing hub (DASH) for analysis by SMEs and two DCNN models.

Figure 3.2: Depiction to aid conceptual understanding of the DAN. Left: A DJI Matrice drone being used to inspect an aircraft.
Middle & Right: Example image capture of aircraft undercarriage

3.1.2 Data Analysis and Sharing Hub (DASH)

The primary function of the DASH is to analyse the images captured by the DAN and determine whether any defects have been identified along with any regions of interest (ROIs) that require further inspection by line maintenance technicians. This is accomplished in three parts. First, the captured image data is passed to a lightweight DCNN model for defect detection in real time (DDRT). SMEs in the DASH can review the defects identified by the DDRT and identify regions that may require further inspection by a line maintenance technician. Concurrently, the image data is also fed through another DCNN model, which prioritises performance in recall and precision for defect detection post-hoc (DDPH). This system prioritises data pertaining to regions of the aircraft that were deemed clear of defects by the DDRT to minimise the chance of FNs while the aircraft is at the gate.

Finally, the image data gathered from the scans will be labelled, and the DCNN models will be periodically trained on this data to improve their performance. The dataset will be uploaded to a cloud database allowing international airports to improve their own models and for the development of alternate analysis techniques that may exhibit superior performance compared to the DCNN solution. Following an air accident, images of the plane in question could be scrutinised to ascertain whether any defects went undetected to further improve the model.

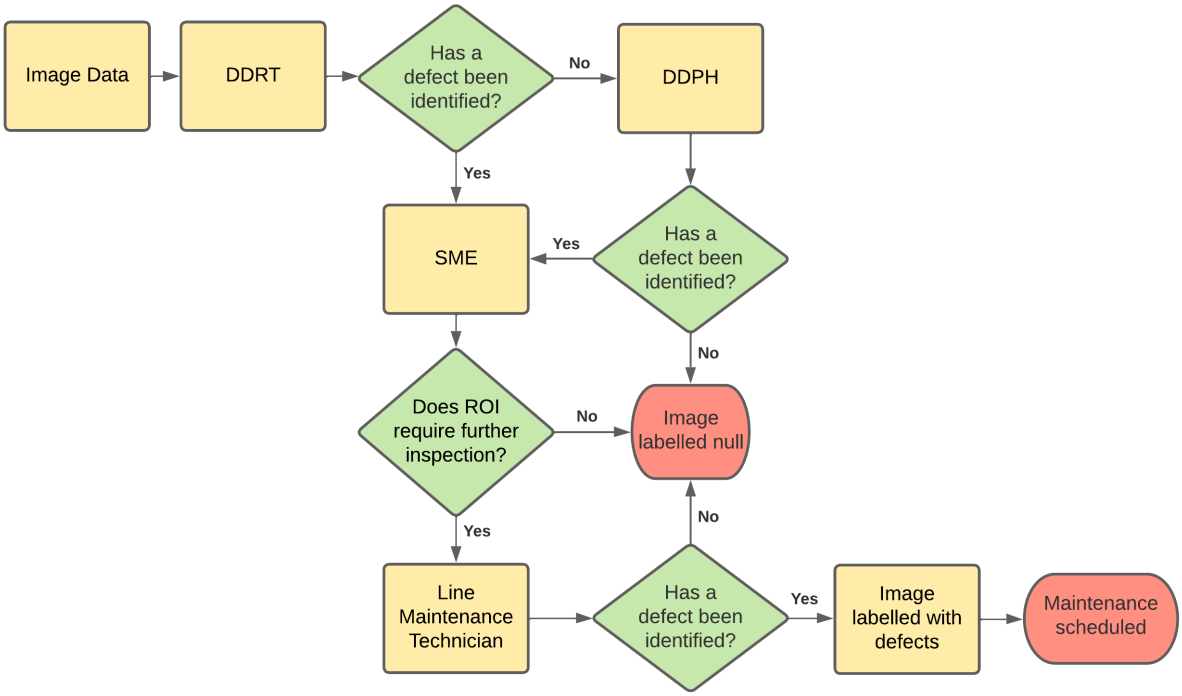


Figure 3.3: AAVI Flowchart

3.1.3 Defect Detection in Real-Time (DDRT) Model Architecture

The DCNN architecture underpinning the model used for DDRT will be based on the foundation model YOLO-NAS (Deci, 2023). YOLO stands for "You Only Look Once" and is a lightweight CNN architecture introduced in 2015. YOLO-NAS has shown promise for real-time object detection, as inference can be performed in milliseconds (Deci, 2023). While conventional DCNN architectures require significant manual tuning and experimentation to find a promising architecture, Deci used their proprietary AutoNAC technology, which applied a neural architecture search (NAS) to construct a YOLO variant that was optimised for both accuracy and lower complexity specifically for the NVIDIA T4 GPU. This allows YOLO-NAS to achieve SOTA performance and an average increase of 0.5 mAP@50 with decreased latency compared to other YOLO variants (Deci, 2023). NAS models try to optimise a balance between model accuracy, computational complexity and model size (Casas et al., 2023). The YOLO-NAS model is pre-trained on both Objects365 (Shao et al., 2019)

and MS-COCO (Lin et al., 2014), making it an excellent foundation model to be fine-tuned for specific tasks. YOLO-NAS has small, medium and large variants that trade increasing performance for inference time and can be selected depending on the application. YOLO-NAS-L has been selected for this use case as it maximises performance while being sufficiently fast for real-time analysis. DDRT model performance and latency are discussed further in chapter 4.

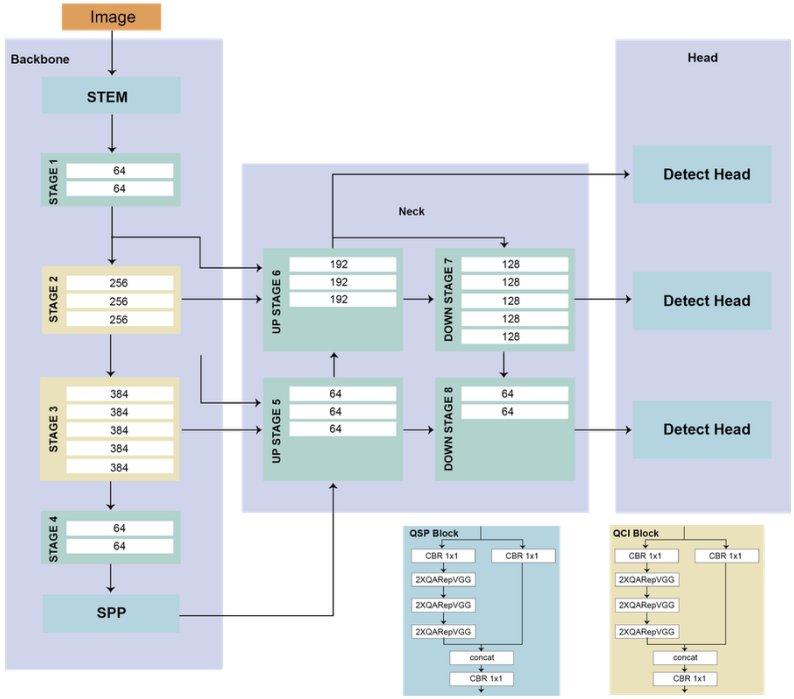


Figure 3.4: Model Architecture of YOLO-NAS-L

3.1.4 Defect Detection Post-Hoc (DDPH)

The model underpinning the DDPH should also take advantage of the benefits that AutoML has to offer. As such, algorithms such as the Clonal Selection Algorithm (CSA) proposed by Al Bataineh et al. (2023) are promising candidates that have demonstrated SOTA performance on the EMNIST dataset. The CSA performs NAS and HPO to optimise its own architecture using a system that mimics the selection of clonal antibodies in the human immune system. This research does not consider model architecture for the DDPH and is suggested for future works.

Chapter 4

Experiments and Results

4.1 Experimental Setup

4.1.1 Dataset Construction

Dataset Acquisition and Selection

To construct an aircraft defect dataset, images were compiled from publicly available datasets pertaining to aircraft defects on the Roboflow platform (Dwyer et al., 2024). The datasets were briefly checked for duplicates, augmentations and unsuitable images. Many of the datasets available on Roboflow pulled images from each other, making it impossible to ascertain the original source.

Dataset Name	Reference	Image Count
Aircraft Dent&Crack Dataset	Detection, 2022	96
Yolov7 Dataset	alex, 2023	284
aircraft Dataset	school, 2023	448
Aircraft Dataset	Debele, 2023a	909
Aircraft-Surface-Damage Dataset	Debele, 2023b	1200
Detect Mix Dataset	Thesis, 2023	1448
aircraft-ruptures	Unknown	107
	Total:	4492

Table 4.1: Publicly available datasets sourced from Roboflow and compiled into a single aircraft defect dataset. The final dataset, "aircraft-ruptures", was removed from Roboflow and cannot be referenced.

The compiled dataset contained classes such as "dents", "Dent" "fissures", "cracks", "scratches", "paint-off", "missing fasteners", "missing heads", "fastener damage", "rupture", and "corrosion". A dataset of exactly 4492 images was initially constructed.

Dataset Preparation

Due to a lack of images and class imbalance, the "paint-off", "scratches", and "corrosion" classes were dropped. The "fissures", "cracks", and "ruptures" classes were combined into a single "Crack" class. Similarly, the "missing fasteners", "missing head" and "fastener damage" class were combined to create the "Missing Fastener" class. Lastly, the "dents" and "Dent" classes were consolidated into a "Dent" class. To prepare the dataset, the images were downloaded and thoroughly inspected to remove duplicates and augmented images. Many invalid images were also removed, such as screen captures from video games, 3D renders, low-resolution images, images captured too far or close to the subject and images with inaccurate annotations. To standardize the image outputs, the dataset was auto-oriented and resized to 640x640 pixels.

Dataset Health and Composition

The final dataset consisted of 1749 images containing 1720 instances of Missing Fasteners, 1345 instances of Cracks, and 915 instances of Dents.

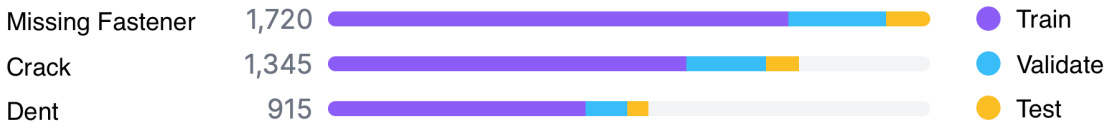


Figure 4.1: Annotation quantity for each class in the dataset.

The dent class is underrepresented in the training data but surprisingly yielded the best performance, as detailed in section 4.2. A review of the dataset showed all of the photographs to be taken in daytime or well-lit conditions in dry environments. Few of the images were blurred or out of focus, and most were taken at a front-on angle perpendicular to the defect.

Train-Test-Validation Split

The dataset was divided into train, validate and test splits with 1363, 263 and 123 images, respectively. This equates to 78% of the dataset devoted to training, 15% to validation and 7% to test model performance. Together, validation and test splits comprise 22% of the dataset before augmentations, which is lower than the recommended 30% (Solawetz, 2020). This split was selected to maximise the amount of image data used to fine-tune the model while allowing sufficient images to evaluate model performance.

Augmentations

An article published by Brems (2021) trialled varying levels of image augmentations, generating between 3x and 20x augmented outputs per training sample. The experiment concluded that the effects of augmentations were most pronounced on smaller datasets and that more augmentations almost universally led to better training outcomes. To test the effects of augmentations on the compiled dataset, two versions were created.

The first version applied seven data augmentation techniques with four outputs per training sample, resulting in a total dataset of 5858 images. The augmentations include horizontal flips, crops of 0-20% zoom, horizontal and vertical shear of ±10%, exposure adjustment of ±10%, cutouts of four boxes each of 10% size, a mosaic applied in a 2x2 grid and horizontal bounding box flips. These augmentations were selected to improve the model's robustness to the slight rotation of camera orientation, defect orientation, visual occlusion, changes in lighting and the size of a defect relative to image size.

(a) No augmentations. Example of the "Dent" class.

(b) Mosaic, cutout and horizontal bounding box flip augmentations

(c) No augmentations. Examples of the "Dent" and "Missing Fastener" classes.

Figure 4.2: a: Example image used in test and validation stages. b and c: Example images used to fine-tune the DDRT model.

The second version added two extra pre-processing steps, converting the images into greyscale and tiling them 2x2 to improve the model's ability to detect small objects. Vertical flips were added to the augmentations while cutouts and bounding box flips were dropped. This resulted in a final dataset of 28804 images.

Training was commenced on both versions of the dataset and initial results showed version one significantly outperforming version two after 120 epochs. Version one also required 1/5th of the training time of version two per epoch. Due to this, version one was selected as the final dataset. The dataset created in relation to this project is publicly available on Roboflow (Ockenden, 2024).

4.1.2 DDRT Implementation

Hardware Configuration

The workstation used to train the DDRT models consisted of a 12th generation Intel Core i7-12700K 20 core CPU, 31GB RAM, and an NVIDIA RTX A2000 with 12GB VRAM. Google Colab connected to a runtime with an NVIDIA T4 GPU was used to test the inference speed of the fine-tuned YOLO-NAS-L model.

Software Configuration

The fine-tuning was performed using Ubuntu 20.04 with CUDA 10.1 and CUDNN 8.9 in a Conda virtual environment based on Python 3.10. Key packages include Supergradients 3.17, PyTorch 2.2, Roboflow 1.1, Supervision 0.18 and Ultralytics 8.2.

Hyper Parameter Setting

To maximise training results, a number of hyper-parameters were adjusted and monitored across various training runs to improve model performance.

Training Parameter	**Value**
metric_to_watch	mAP(50)
average_best_models	True
score_threshold	0.01
Evaluation Parameter	**Value**
score_thres	0.5
include_classwise_ap	True
calc_best_score_thresholds	True
nms_threshold	0.65

Table 4.2: Table displaying the most influential training and evaluation parameters when fine-tuning YOLO-NAS

In terms of training parameters, YOLO-NAS allows for the "metric_to_watch" to be set, which instructs the model on how to evaluate its own performance during training. Mean average precision at 50% confidence (mAP@50) was selected as it encapsulates performance in both precision and recall. A feature that distinguishes YOLO-NAS from other YOLO variants is the ability to enable "average_best_models", which constructs a version of the model by averaging the model weights across the ten best checkpoints during fine-tuning. Training with this setting enabled consistently demonstrated significantly better results with a nearly 10% increase in mAP(50) observed on the best fine-tuning run compared to the single "best" epoch. The "score_threshold" variable found within the "PPYOLOEPost-PredictionCallback" sets the limit at which predictions below this confidence level will be ignored in Top-K and non-maximum suppression (NMS). This value was set at 0.01 to allow for maximal recall.

For evaluation, the "score_thres" variable found within the "DetectionMetrics_050" and "DetectionMetrics_050_095" modules was set to 0.50 and acts as the confidence level that is used to evaluate the model following each epoch, which is logged to the tensor boards. Enabling the "include_classwise_ap" setting logs the average precision (AP) for each class, offering insight into the performance of each class and whether the dataset needs improvement. The "calc_best_score_thresholds" variable returns the confidence level that maximises the F1-score of the model during evaluation. Finally, the "nms_threshold" denotes the percentage of overlap of the intersection over union (IoU) that is tolerated before NMS removes predictions that are too similar. This value was manually tuned to 0.65, which offered the best results during evaluation.

4.1.3 Tutorial Notebooks

All of the code generated to fine-tune and evaluate the DDRT YOLO-NAS-L model was adapted from the notebook linked on the Roboflow blog posted by Skalski (2023) titled "How to Train YOLO-NAS on a Custom Dataset". The Ultralytics package and documentation were used to fine-tune YOLOv5, YOLOv8 and YOLOv9 for comparison.

4.1.4 Comparison Methods and Evaluation Metrics

The DDRT model's performance will be assessed using four evaluation metrics: precision (P), recall (R), mean average precision (mAP) and F2-score, according to standard definitions and measured at 50% confidence unless otherwise specified. Average precision (AP), along with a confusion matrix, will also be used to understand how the DDRT model is performing across classes. Frames per second (FPS) and parameter count will be used to measure the ability of the model to run in real time and quantify model complexity.

Testing the recall of the model allows for the evaluation of how many defects the DDRT model can identify out of the total number of defects present which is especially useful for comparison against human inspectors. Precision quantifies the percentage of correct predictions that the model is making over total predictions. For the DDRT model, unidentified defects could cause disaster and loss of life; as such, it makes sense to evaluate the model with an emphasis on performance in recall rather than precision, which can be accomplished by calculating the F2-score. Average precision (AP) is calculated using the area under the P-R curve for each class, allowing us to gain an understanding of the trade-off between these two key metrics. Mean average precision (mAP) aggregates the AP performance of each of our three classes, giving a holistic overview of model performance in P and R for all classes. The mAP and AP will be calculated at IoU thresholds of 50% and 50:95%. These thresholds pertain to the overlap of the model's bounding box prediction on the ground truth image labels. An IoU threshold of 50% is the minimum to be considered a TP while looking at mAP@50:95 allows us to understand the model's performance to better localise defects that are present.

4.2 Results

As shown in the table below, at 50% confidence, YOLO-NAS-L proved to be the most performant model in terms of R and tied first in F2-score with YOLOv9-e. It is marginally edged out in mAP(0.5) by YOLOv8-L. YOLO-NAS-L exhibits the lowest score in precision and second last in mAP(0.5:0.95). Detailed training graphs depicting model comparisons between recall, precision, AP and mAP can be found in the appendix.

Model	**R**	**P**	**F2**	**mAP(0.5)**	**mAP(0.5:0.95)**	**Parameters (M)**
YOLOv5-L	0.759	0.877	0.7800	0.825	0.466	53.2
YOLOv8-L	0.784	0.88	0.8014	**0.848**	**0.563**	**43.7**
YOLO-NAS-L	**0.7969**	0.8411	**0.8053**	0.8467	0.4815	66.9
YOLOv9-e	0.779	**0.881**	**0.8053**	0.84	0.546	58.1

Table 4.3: YOLO family model evaluations at 50% Confidence listed in order of release.

DDRT model performance directly correlates to the percentage of detected defects in the AAVI system. As such, the confidence threshold should be set to maximise the F2 score. The best score threshold to optimise the F1 score was calculated at 47%, which, through trial and error, was also found to maximise the F2 score.

Model	**R**	**P**	**F1**	**F2**
YOLO-NAS-L: 47% Confidence	0.8167	0.8382	0.8260	0.8209

Table 4.4: YOLO-NAS Model Evaluation at an optimised confidence threshold of 47%

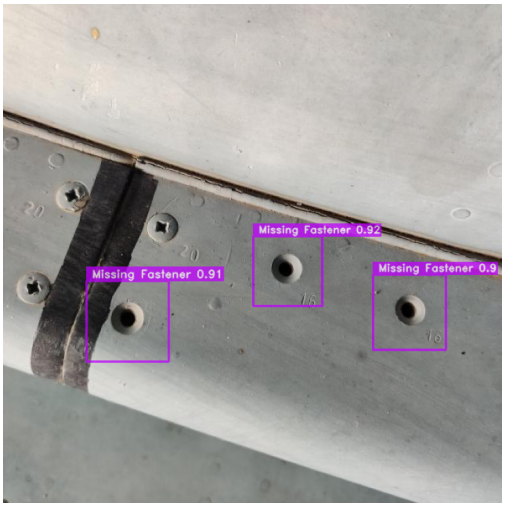
4.2.1 Inference Latency

Latency testing of the DDRT model was performed on Google Colab connected to a runtime with an NVIDIA T4 GPU.

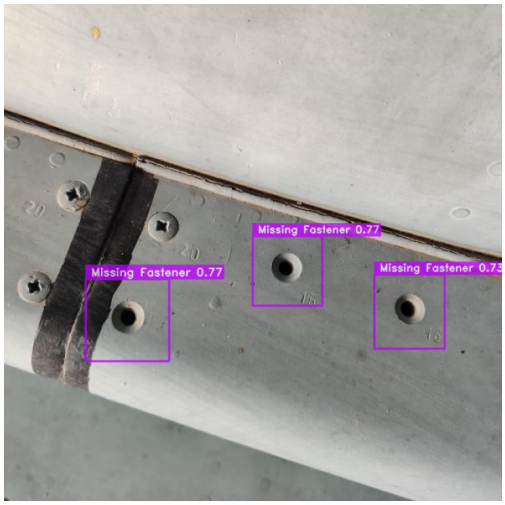
YOLO-NAS-L Version	Latency (ms)	FPS
PyTorch	64.85	15.42
FP16 Quantization	10.15	98.52
Int8 Quantization	7.95	125.78

Table 4.5: Performance of the DDRT Model fine-tuned on aircraft defects dataset tested using Google Colab on an NVIDIA T4 GPU in PyTorch and TensorRT FP16 and Int8 quantised formats.

Inference testing with annotations allowed for the comparison of predictions and confidence between the PyTorch and quantised FP16 and Int8 models.



(a) PyTorch and Quantised FP16



(b) Quantised Int8

Figure 4.3: Inference tests performed on PyTorch, quantised FP16 and Int8 models. Prediction confidence is identical for PyTorch and FP16 precision. The Int8 quantised model demonstrates an average drop in prediction confidence of 15% in this test.

Model Comparison

Model (TensorRT FP16)	Latency (ms)	FPS
YOLOv5-L	Unknown	Unknown
YOLOv8-L	**7.315**	**136.71**
YOLO-NAS-L	10.15	98.52
YOLOv9-e	13.61	73.45

Table 4.6: Performance of the compared YOLO family models tested on an NVIDIA T4 GPU in TensorRT FP16 quantised format. Results for YOLOv8-L and YOLOv9-e are cited from (ausk, 2024).

4.2.2 Classwise Evaluation

To understand how individual classes perform in the DDRT model, it is helpful to look at classwise AP scores and a confusion matrix calculated at the best score threshold of 47%.

IoU Threshold	mAP	AP: Crack	AP: Dent	AP: Missing Fastener
50%	0.8467	0.7708	0.9372	0.8321
50:95%	0.4815	0.3877	0.6313	0.4254

Table 4.7: Average Precision across and between classes at varying confidence thresholds

The DDRT model exhibits the best performance detecting the "Dent" class at IoU thresholds of both 50% and 50:95%. This is most likely due to the quality of image data in the "Dent" class, especially as it is underrepresented in the dataset.

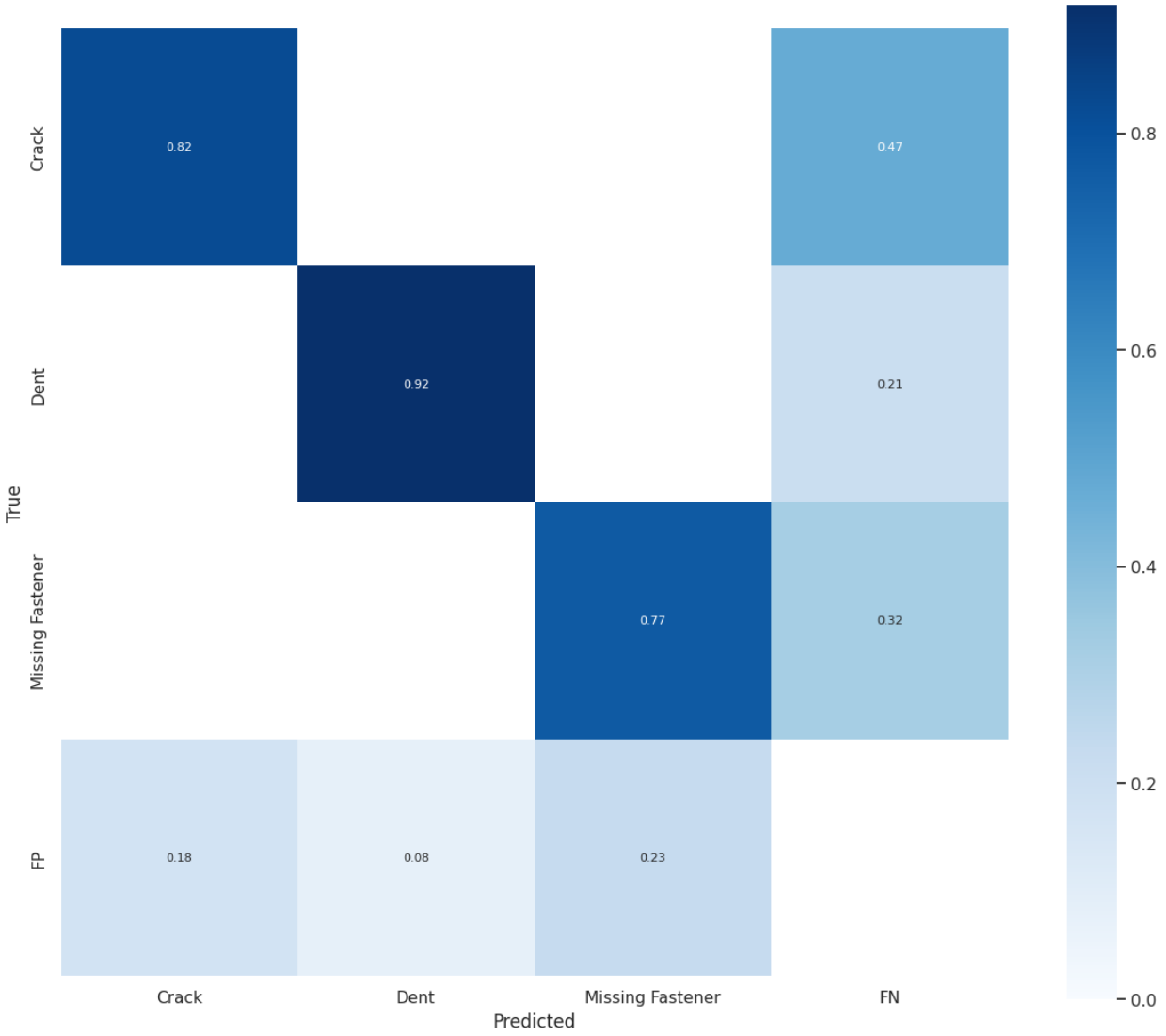


Figure 4.4: Confusion Matrix at 47% Confidence to Optimise F1 Score

4.2.3 Implications

As missing a defect could lead to catastrophe, FNs are of most concern for the DDRT model. Here, it is clear that the model performs best in the Dent class and worst in the Crack class, missing 21% and 47% of these defects, respectively. When averaged across the three classes, the average rate of FNs is 33%, which matches the performance of human inspectors in the study performed by Drury et al. (1997).

The DDRT model excels in detecting Dents, with the least FPs and FNs at only 8% and 21%, respectively. Similarly, 92% of the Dents predicted by the model are correct. The Missing Fastener class shows the most FPs predicted by the model at 23% and the second highest FNs at 32%. 77% of the Missing Fastener predictions made by the model are correct.

For the three classes that the DDRT model predicts of Cracks, Dents and Missing Fasteners, the model makes correct predictions for each class 82%, 92% and 77% of the time, respectively. For these same classes, the FN rates are 47%, 21% and 32%, respectively, and the FP rates are 18%, 8% and 23%, respectively.

4.2.4 Limitations

The limitations of this research are numerous. Many aspects of AAVI need to be researched before a viable system can be implemented. This includes DDPH model architecture, appropriate drone and computer hardware configuration, image capture requirements, aircraft mapping by UAV and regulations impacting the use of AAVI in real-world scenarios. Latency measurements were also unable to be performed on a single workstation, although, results were found applying the same GPU.

Chapter 5

Ethical, Indigenous and Sustainability Considerations

5.1 Ethics Evaluation

This section covers the ethical considerations of the proposed AAVI system according to the ethical frameworks of utilitarianism, deontology and care-based ethics. Utilitarianism discerns the morality of actions based on what generates the best outcome (Kay, 2018). Deontology posits that actions can be determined as right or wrong according to rules and professional duties (Barrow & Khandhar, 2023). Finally, the ethics of care emphasises that the significance of moral choices depends on their impact on caring for and nurturing others (Pettersen, 2011).

Adopting the AAVI system aligns with utilitarianism as if successful, it will result in safer air travel, decreased inspection times, less unnecessary maintenance and reduced costs, which are all positive outcomes. Future iterations of the automated system may become so adept as to cause the job loss of SMEs and human inspectors, which could be considered negative according to utilitarianism; however, the decreased loss of life would likely outweigh this aspect.

As the AAVI system seeks to increase the consistency and reliability of the aircraft inspection process, it aligns with deontology. This extends as the goal of the system is founded on the duty of the airline and maintenance workers to ensure the safety of passengers. Deontology stresses adherence to ethical rules, and if the AAVI system is proven to reduce the likelihood of air accidents, it could be deemed unethical under deontology for an airline not to adopt it.

Care-based ethics are based on how decisions impact the care and nurture of people and relationships. The AAVI system aims to ensure the flightworthiness of aircraft and, therefore, the safety of all passengers and crew onboard. If AAVI is eventually adopted as an entirely autonomous system, which leads to the displacement of workers, then AAVI could result in potential harm to the families and lives of human inspectors, which would be viewed unfavourably under this framework. This impact would again need to be weighed against the reduction in loss of life due to safer air travel.

5.2 Indigenous Dimension

This technology must consider the perspectives of Indigenous Australians who have cultural and historical connections to both land and sky. Cultural sensitivity should be applied to the input from Indigenous communities about the proposed technology, and any environmental impact this technology could have should be considered. Implementing the technology could promote employment and training opportunities for First Nations people to decrease the large unemployment gap between indigenous and non-indigenous Australians.

5.3 Sustainability Reflection

5.3.1 Sustainable Use of Resources

All aspects of AAVI require significant resource outlay, such as drone, computer, camera, and server hardware, which constitute the DAN and DASH. The proposed AAVI system requires labour from SMEs, LMTs and other employees to maintain datasets and cloud infrastructure. The system will also require electricity to operate. These resources are expensive monetarily and contain various electrical components such as processors, batteries and polymers made from rare earth or non-renewable materials, not to mention man hours. The cost of these assets must be weighed against the losses that occur due to air accidents. Most important is the cost of mortality, which, if viewed purely from a monetary perspective, leads to less productive companies, wasted education and increased financial pressure on households suffering the loss of a loved one. Typical passenger aircraft cost in excess of $80M USD, not to mention the cost of cleanup crews and air crash investigations. Finally, it is worth considering the labour hours LMTs can re-invest into other work areas due to reduced inspection times.

5.3.2 Designing a Sustainable System

Systems should be designed for longevity, allowing them to be improved and incorporate new technology without a complete overhaul. The AAVI system applies DCNNs as a technique for defect detection and possesses the capability to improve itself as dataset quality, model architecture and compute power increase over time. The drone platform that AAVI takes advantage of can also house additional sensors, allowing new defect-detection techniques to be designed and tested. The ever-growing dataset used to train the DDRT and DDPH models can also be used to design models specific to aircraft defect detection. New instances of AAVI can also be quickly deployed and scaled without significant training or variation in performance. AAVI embodies a future-proofed fault detection system that can be built upon as its underlying technology matures.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This paper has presented AAVI, a complete aircraft VI system suitable for use in modern airports. The approach synthesises the mobility of drone-mounted cameras with the object detection capability of SOTA DCNN models to negate the limitations of manual inspection processes. The proposed YOLO-NAS-L model architecture was evaluated against three other lightweight DCNN models in the YOLO family and demonstrated the most promising results as the foundation of the DDRT by a narrow margin. For deployment on the edge, YOLOv8-L demonstrated comparable performance with lower inference times. The DDRT model also bested the detection ability of human inspectors and validated itself as capable of performing defect detection in real time when deployed on the edge. The adoption of AAVI presents numerous benefits to the aerospace industry such as safer air travel, less unnecessary maintenance and a decreased burden of work for line maintenance technicians. AAVI also fills a research gap identified in a review by Yasuda et al. (2022) as the first proposed complete inspection system in the literature. There remain three main research gaps related to automated aircraft visual inspection solutions that should be explored in future works, along with the specifications of AAVI that are not covered in this work.

6.2 Future Work

6.2.1 Dataset Quality

The performance of neural networks is highly dependent on the dataset on which they are trained and fine-tuned. As such, considerable performance gains could be garnered in defect detection by validating and improving the quality of aircraft defect datasets. Firstly, it is important that the dataset has been cleaned of duplicate images and images that lack sufficient detail. This includes images that may mislead the model or train it on conditions that an aircraft in service would never experience such as aeroplanes installed within a museum. Secondly, it is vital that the dataset has been properly annotated in

an accurate and consistent manner to allow the model to learn appropriately and ensure that the evaluation accurately reflects the model performance and not the dataset quality. Another improvement in the dataset could be the labelling of defect severity. This would allow the model to make predictions of whether defects will require maintenance or can be left as is prior to the next flight. This could also be achieved by a separate model that is dedicated to this purpose. Additionally, a model can only be expected to predict well on data similar to that it has been trained on. A larger dataset captured in a wider array of environmental conditions will allow the model to improve its generalisability and improve its robustness in real world scenarios. Finally, the addition of extra classes should be considered to detect defects such as corrosion, scratches, evidence of leaks and missing paint.

6.2.2 AAVI Research Gaps

DDPH Model Design

YOLO-NAS excels in low latency object detection, but its lightweight architecture means that its performance could likely be improved by a model not constrained for defect detection in real-time. It is recommended that research be dedicated to the development of DDPH model architecture.

DDRT Hardware

To accomplish the AAVI system it is necessary to run the DDRT model deployed on the edge. Preliminary research, such as similar implementations from Mainblades (Horstink, 2022), suggests that the DJI Matrice 350 RTK and NVIDIA Jetson Xavier computer may possess the potential for use in the DDRT. However, this topic requires significant further validation.

Appendix A

Appendix

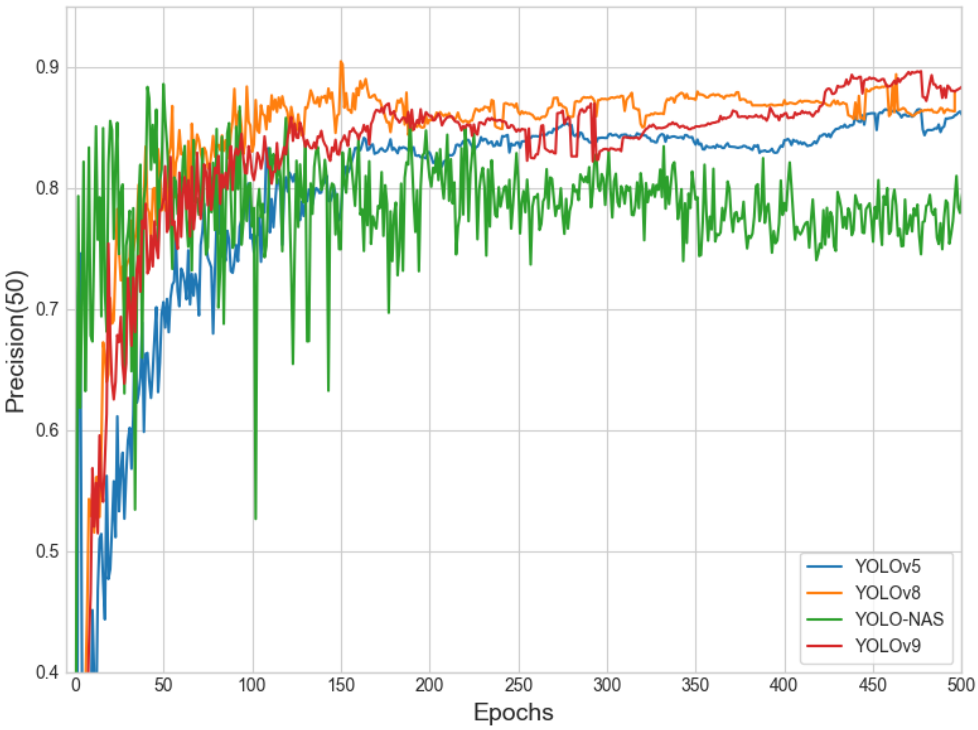


Figure A.1: Training Graph of Precision at 50% Confidence

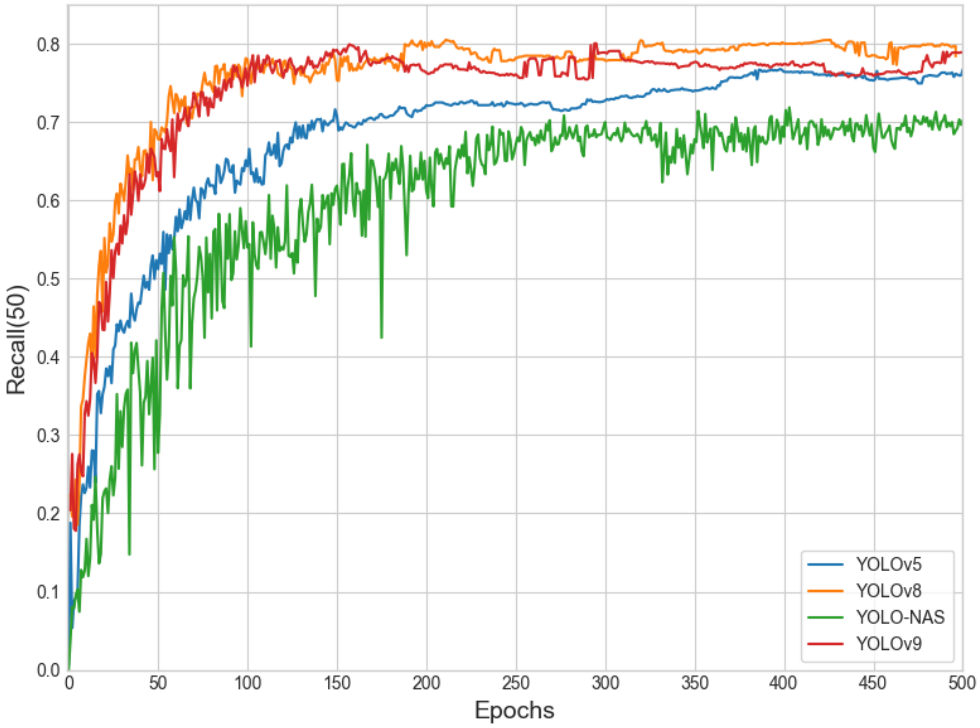


Figure A.2: Training Graph of Recall at 50% Confidence

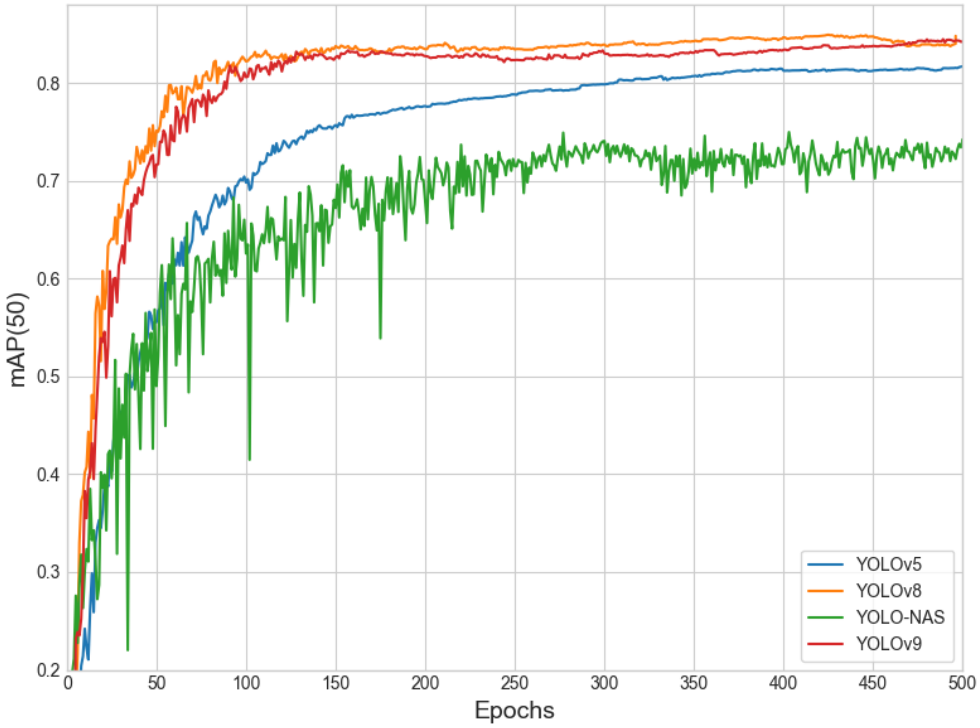


Figure A.3: Training Graph of mAP at IoU Threshold: 50%

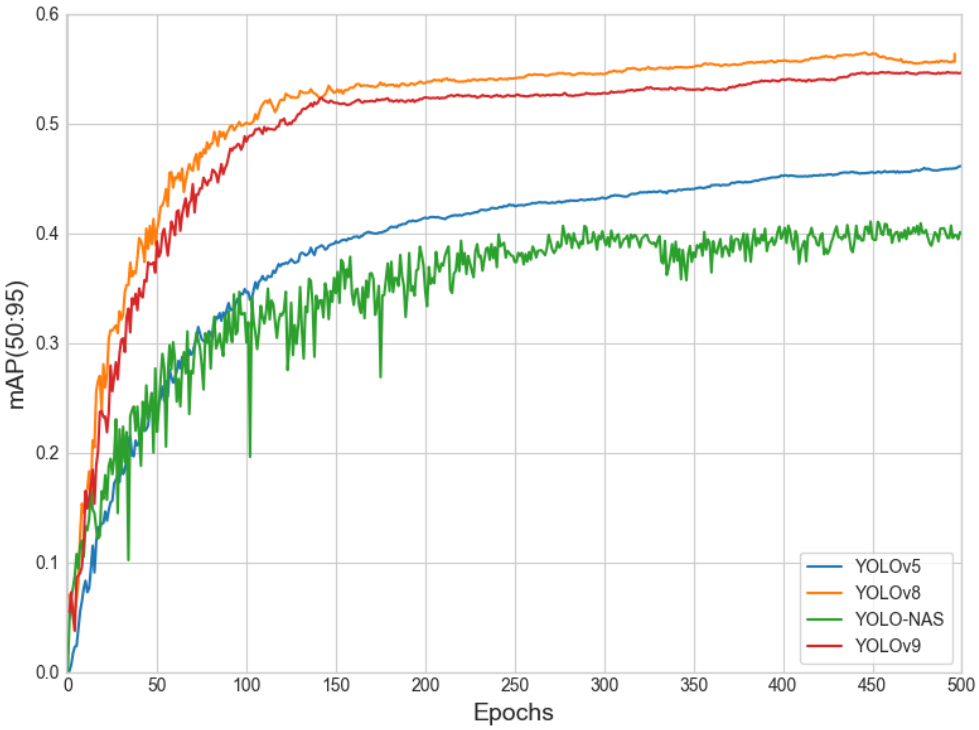


Figure A.4: Training Graph of mAP at IoU Threshold: 50:95%

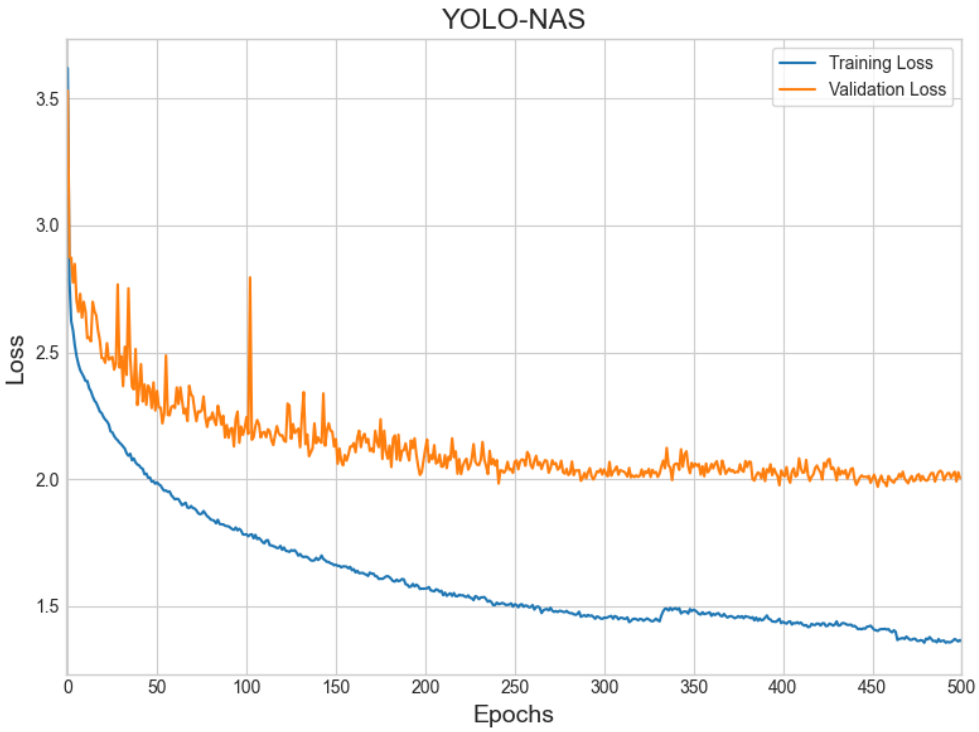


Figure A.5: YOLO-NAS-L: Training vs Validation Loss

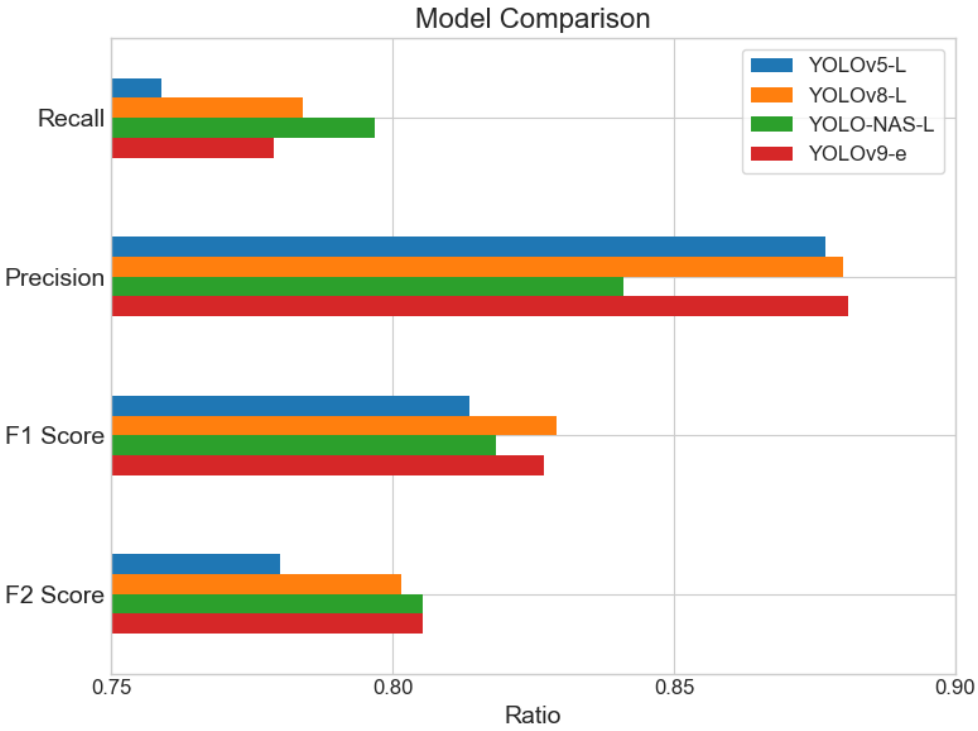


Figure A.6: Model Comparisons of Recall, Precision and F-beta Scores

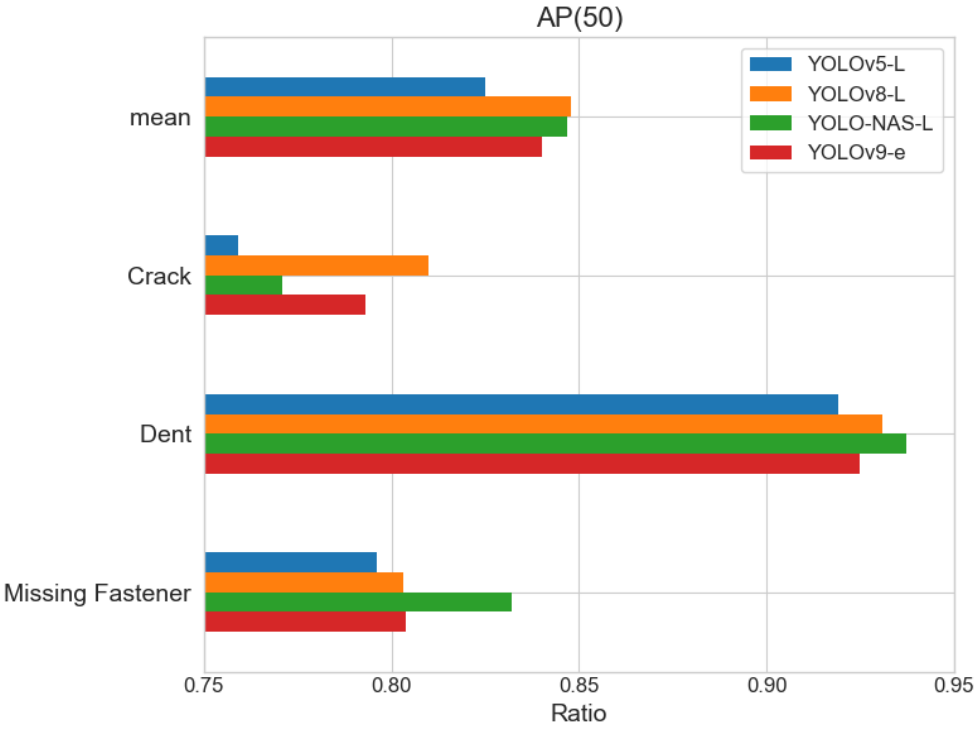


Figure A.7: Model Comparisons of mAP at IoU Threshold: 50%

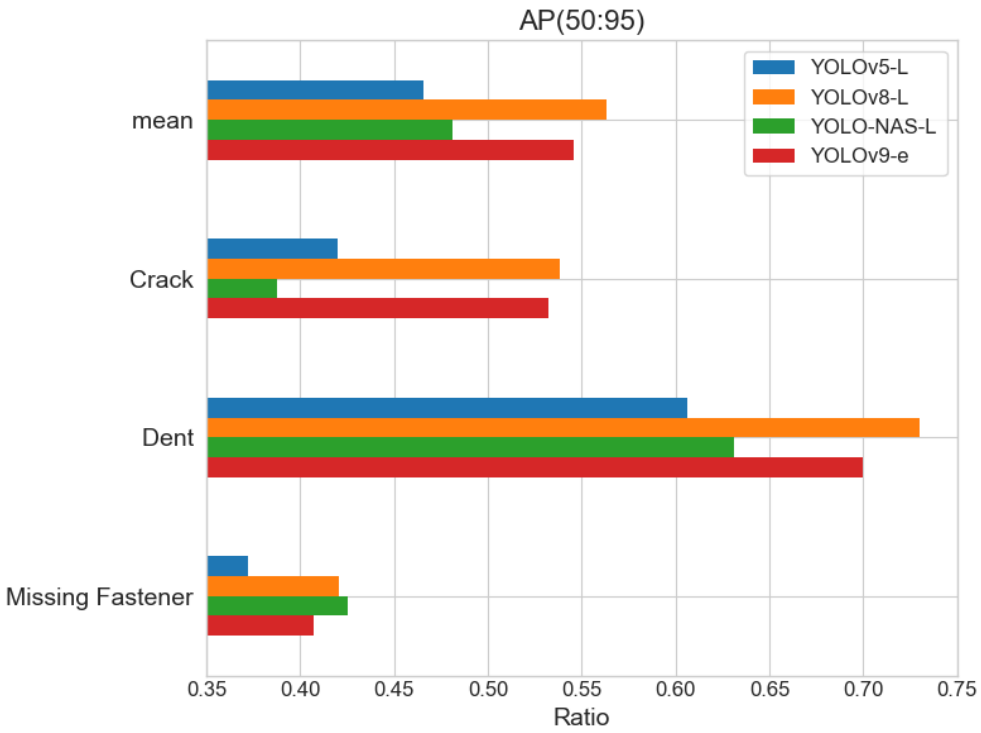


Figure A.8: Model Comparisons of mAP at IoU Threshold: 50:95%

Bibliography

Aharon, S., Louis-Dupont, Ofri Masad, Yurkova, K., Lotem Fridman, Lkdci, Khvedchenya, E., Rubin, R., Bagrov, N., Tymchenko, B., Keren, T., Zhilko, A., & Eran-Deci. (2021). Super-gradients. https://doi.org/10.5281/ZENODO.7789328

alex. (2023, August). Yolov7 dataset [visited on 2024-05-24]. https://universe.roboflow.com/alex-oqpwt/yolov7-oyimw

Debele, L. (2023a, December). Aircraft dataset [visited on 2024-05-25]. https://universe.roboflow.com/lemi-debele/aircraft-m3c6e

Debele, L. (2023b, December). Aircraft-surface-damage dataset [visited on 2024-05-25]. https://universe.roboflow.com/lemi-debele/aircraft-surface-damage

Detection, A. (2022, September). Aircraft dentcrack dataset [visited on 2024-05-24]. https://universe.roboflow.com/aircraft-detection-hpzth/aircraft-dent-crack

Jocher, G. (2020). *Ultralytics yolov5* (Version 7.0). https://doi.org/10.5281/zenodo.3908559

Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics yolov8* (Version 8.0.0). https://github.com/ultralytics/ultralytics

Ockenden. (2024, May). Defects cleaner dataset [visited on 2024-05-27]. https://universe.roboflow.com/capstone-tqtck/defects-cleaner

school. (2023, August). Aircraft dataset [visited on 2024-05-24]. https://universe.roboflow.com/school-sapxp/aircraft-oeted

Thesis. (2023, October). Detect mix dataset [visited on 2024-05-24]. https://universe.roboflow.com/thesis-vnyio/detect-mix

Wang, C.-Y., & Liao, H.-Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information.