Advancing Utility Pole and Sign Detection Through Deep Learning

Carl Dickinson

In the fulfilment of the requirement for the degree of

Master of Philosophy

Centre for Intelligent Dynamic Communications
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow

2026

Abstract

This thesis addresses the need for safer, faster, and more consistent inspection of wooden utility poles in electrical distribution networks. Manual inspections are costly, labour-intensive, and potentially hazardous, while automated inspection from street-level imagery must cope with background clutter, occlusion, variable lighting, small warning signs, and visually ambiguous wooden poles that may resemble trees or other vertical structures. To address these challenges, this thesis develops and evaluates a deep learning pipeline for automated utility pole detection, warning-sign detection, pole segmentation, and image-plane lean-angle estimation using Google Street View (GSV) imagery.

The thesis makes three main contributions. First, it curates the OHL-UK dataset, comprising 4,570 annotated GSV images with bounding boxes, segmentation masks, and lean-angle annotations for wooden utility poles and attached electrical warning signs. Second, it benchmarks established object-detection models, including RetinaNet, You Only Look Once version 3 Tiny (YOLOv3-Tiny), Faster Region-based Convolutional Neural Network (Faster R-CNN), and Detection Transformer (DETR). The DETR-based approach achieved mean average precision (mAP) values of 90.43% for pole detection and 88.26% for warning-sign detection. Third, it extends the DETR-based framework with segmentation capability, enabling pole masks to be used for image-plane lean-angle estimation. On the test set, lean angle was estimated for 1,367 out of 1,433 true-positive poles, with a mean absolute error of 1.01°.

Together, these contributions demonstrate a scalable framework for automated visual assessment of wooden utility poles and provide a reusable dataset and benchmark for future research in electrical infrastructure monitoring.

Contents

List of Figures

List of Figures

List of Figures

List of Tables

List of Tables

List of Abbreviations

Abbreviation	Meaning
AP	Average Precision
API	Application Programming Interface
AUC	Area Under the Curve
BMVC	British Machine Vision Conference
CC BY	Creative Commons Attribution
CE	Cross-Entropy
CIGRE	International Council on Large Electric Systems
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CPU	Central Processing Unit
CR	Condition Rating
CSV	Comma-Separated Values
DETR	Detection Transformer
DINO-DETR	DETR with Improved deNoising anchOr boxes
DGPS	Differential Global Positioning System
DNO	Distribution Network Operator
DPN	Deep Parsing Network
DRA	Dynamic Range Adjustment
DSP	Digital Signal Processing

List of Abbreviations

ELEVA	Electric power Line Exploration using Aerial Vehicle
ENA	Energy Networks Association
ESQCR	Electrical Safety, Quality and Continuity Regulations
FCN	Fully Convolutional Network
FL	Focal Loss
FN	False Negative
FLOPs	Floating-Point Operations
FOV	Field of View
FP	False Positive
FPN	Feature Pyramid Network
GFLOPs	Giga Floating-Point Operations
GIoU	Generalised Intersection over Union
GIS	Geographic Information System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GSV	Google Street View
GT	Ground Truth
HI	Health Index
HOG	Histogram of Oriented Gradients
HPC	High-Performance Computing
HV	High Voltage
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IET	Institution of Engineering and Technology
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IoU	Intersection over Union
JSON	JavaScript Object Notation
kV	kilovolt

List of Abbreviations

L1	Least Absolute Deviations norm
LOB	Line of Bearing
LTI	Lost Time Incident
MAE	Mean Absolute Error
mAP	mean Average Precision
MB	Megabytes
MLP	Multi-Layer Perceptron
MV	Medium Voltage
OHL	Overhead Line
ORB	Oriented FAST and Rotated BRIEF
PASCAL VOC	Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes
PCNN	Pulse Coupled Neural Network
PPE	Personal Protective Equipment
PSO	Particle Swarm Optimisation
R-CNN	Region-based Convolutional Neural Network
R-FCN	Region-based Fully Convolutional Network
RGB	Red-Green-Blue
SDK	Software Development Kit
SSH	Secure Shell
SSD	Single Shot Detector
SVM	Support Vector Machine
TP	True Positive
UAV	Unmanned Aerial Vehicle
UK	United Kingdom
UKPN	UK Power Networks
UPC	Utility Poles with Crossarms
URL	Uniform Resource Locator

List of Abbreviations

VGG	Visual Geometry Group
VIA	VGG Image Annotator
YOLO	You Only Look Once

Chapter 1

Introduction

Utility poles are an essential part of the infrastructure used to support power distribution systems, telecommunications, and other critical public services. These poles are subject to environmental stressors such as wind, rain, and material decay, which can cause them to lean or deteriorate over time. The lifespan of wooden utility poles in the UK depends on several factors, including the type of treatment, environmental conditions, and maintenance practices. Creosoted poles, for instance, are widely used and have an estimated lifespan ranging from 30 to 60 years [1]. Other poles, treated with preservatives since 2005, may have a shorter expected lifespan of 15 to 20 years due to changes in preservation methods [2]. If left unchecked, leaning or decaying utility poles pose serious hazards, potentially leading to service interruptions, accidents, or catastrophic failures. Regular inspection and maintenance of utility poles are therefore crucial for ensuring the stability and safety of the electrical grid [3].

Providing customers with a reliable supply of electricity is an expensive undertaking for power companies. Inspection and maintenance activities cost, on average, £53.5 million annually [4]. However, these costs are necessary to prolong the lifespan of equipment and reduce the rate of faults, which can cost power companies more than £130 million annually in repairs, fines, and compensation [5].

In the UK, for instance, customers are eligible for compensation of £70 if adverse weather causes a power outage lasting more than 24 hours, with additional compensation awarded for every 12-hour period thereafter [5].

The importance of regular inspection was underscored by the impact of Storm Arwen in November 2021. The storm left 59,101 homes in the UK without power for more than 48 hours, and 3,032 homes were without power for a week or more, with some waiting as long as 11 days to be reconnected [6]. This resulted in £24.5 million in compensation payouts. Most of the faults were caused by trees falling on lines, snapped wooden poles, and lines severed by ice accumulation. This event highlighted the critical role that inspection activities play in identifying vulnerable equipment, such as excessively leaning poles, before they fail and cause costly disruptions.

Traditional methods of inspecting utility poles involve manual labour, where inspectors travel on foot to visually examine each pole for structural defects, lean angles, and signs of decay. These inspections are time-consuming, labour-intensive, and prone to human error. Inspectors often face hazardous conditions, such as navigating uneven or slippery terrain, carrying heavy equipment, and working in proximity to high-voltage lines, which pose significant risks to their safety. On average, twelve members of the public are seriously injured or killed each year due to contact with overhead lines, with haulage and construction workers particularly at risk [7]. Furthermore, slips, trips, and falls account for 37% of workplace injuries in the UK [8]. The repetitive nature of these inspections, combined with the dangers involved, makes this task not only hazardous but also inefficient.

Recent advances in computer vision and deep learning have presented opportunities to automate the detection and assessment of utility poles, potentially reducing the time and cost of manual inspections. By leveraging data from sources like Google Street View [9], automated systems can identify and assess utility

poles more efficiently. However, current automated methods still face challenges, particularly when it comes to estimating lean angles accurately in complex environments. Factors such as road gradients and background clutter often introduce errors into lean angle estimations, making current solutions unreliable in many real-world scenarios [10, 11]. This research aims to address these limitations by developing a novel deep learning based framework that can more accurately detect utility poles and estimate their lean angles, even in challenging conditions.

1.1 Motivation

The accurate estimation of the lean angle of utility poles is essential for maintaining the structural integrity and safety of electrical infrastructure. Lean angles that deviate significantly from the vertical may indicate potential failures or weaknesses in the poles, leading to service interruptions, property damage, and safety hazards [3]. Regular inspections are necessary to mitigate these risks, particularly as the increasing demand for reliable electricity in both urban and rural areas places greater strain on power distribution systems.

Despite advances in automated inspection methods, significant challenges remain. Existing deep learning techniques, such as YOLO v3-tiny [12] and RetinaNet [13], perform well in detecting utility poles but often fail to account for complex environmental factors, such as background occlusions or varying contextual features. These factors can skew lean angle estimations, especially when poles are located in areas with uneven terrain or when part of the pole is obscured by trees or buildings [10, 11]. Additionally, current methods do not integrate critical inspection data, such as the presence of warning signs, which are legally required in many regions for safety reasons. The lack of integration of these important elements reduces the overall reliability and comprehensiveness of existing automated systems.

The motivation for this research is to develop a more accurate and efficient system for pole detection and lean angle estimation that addresses the shortcomings of existing methods. By focusing on detecting both poles and attached warning signs, this work seeks to fill the gaps left by current approaches and improve the overall reliability of automated pole inspection systems.

1.2 Objectives

This thesis aims to:

- Develop a state-of-the-art deep learning model for the detection of utility poles and their attached warning signs from publicly and readily accessible images, such as those available through Google Street View. This model will be benchmarked against existing utility pole detection methods, including RetinaNet, YOLO v3-tiny, and Faster RCNN [10, 11, 14–17].
- Enhance the accuracy of lean angle estimation by leveraging precise segmentation to address factors such as occlusions, background clutter, and multiple pole components, overcoming limitations of models that rely on coarse bounding box detection or multi-stage pipelines [10, 11, 14, 15, 17].
- Automate the process of detecting and estimating lean angles to improve the efficiency, accuracy, and safety of electrical infrastructure maintenance while reducing human error and associated costs.

1.3 Contributions

The novel contributions of this thesis include:

- **Curation and public release of the OHL-UK dataset for utility pole analysis.** OHL-UK is a domain-specific corpus of 4,570 Google

Street View images (640×640 pixels) containing 6,773 annotated wooden utility poles and 1,805 electrical warning signs. Each instance is labelled with COCO-format bounding boxes, polygonal segmentation masks, and per-pole lean-angle annotations, and the data are organised into stratified train/validation/test splits (Section 3.2, Table 3.1). The dataset, together with the trained detection and segmentation models, is openly available via the University of Strathclyde PURE research data repository under a CC BY 4.0 licence [18], providing the first publicly shareable GSV corpus of wooden poles without crossarms for this application domain. A peer-reviewed conference paper based on this dataset and the associated DETR-based detection and lean-angle estimation framework has been accepted for publication at the 36th British Machine Vision Conference (BMVC 2025) [19].

- **Development of a deep learning-based methodology for the detection and segmentation of utility poles and their warning signs.** Traditional methods of utility pole detection often suffer from inaccuracies due to environmental occlusion and the blending of poles with their surroundings, such as trees and buildings [20]. In Chapter 3 of this thesis, a deep learning model using Detection Transformer (DETR) is introduced, designed specifically for challenging image environments like Google Street View. Unlike previous models, which primarily focus on poles with crossarms, this work successfully detects straight wooden poles without crossarms—a more challenging task due to their visual similarity to tree trunks. Furthermore, the model identifies attached warning signs, which are critical for ensuring safety compliance. By training on a robust dataset, the DETR model outperforms existing approaches like YOLO v3-tiny and RetinaNet, achieving a 13-point improvement in detection accuracy.

- **Development of a novel lean angle estimation model for utility poles.** The object detection model described in Chapter 3 is extended in Chapter 4 to perform utility pole segmentation, enabling lean angle estimation. This approach improves upon existing methods by generating higher-quality segmentation masks, which lead to more accurate measurements.

  Previous approaches, such as those by Zhu et al. [10] and Alam et al. [17], relied on SegNet for segmentation and applied the Hough Transform to estimate lean angles. Although promising, their models were evaluated on small datasets and struggled in accuracy when tested on larger image sets. Similarly, Kim et al. [11] used AlexNet with edge detection techniques, but low-resolution images from Google Street View hindered the accuracy, with 75% of angle errors being less than 1 degree.

  In contrast, the method presented in this thesis, described in Chapter 4, refines pole segmentation using advanced deep learning techniques, providing a solid baseline for lean angle estimation without relying on filtering techniques that often introduce noise. This method addresses challenges such as background occlusions, varying lighting and weather conditions, and differences in pole materials and colours. By focusing on improved image segmentation and processing, this approach overcomes limitations in existing methods and provides more reliable lean angle estimates for real-world conditions.

1.4 Organisation of the Thesis

This thesis is organised as follows:

- **Chapter 2** reports a comprehensive literature review of existing methods

for utility pole detection and lean-angle estimation, including manual inspection techniques, early automated systems, and the role of deep learning in modern approaches. The chapter also discusses the gaps in the literature that this research then addresses.

- **Chapter 3** presents the development and evaluation of a custom deep learning model for utility pole detection, focusing on identifying poles in challenging environments, such as those found in Google Street View. This chapter also discusses the dataset collated for this purpose, which includes labelled Google Street View images collected at specific geographic coordinates provided by UK Power Networks. Each location was captured through a 360-degree sweep to ensure a high likelihood of including the utility pole, addressing issues of occlusion and out-of-frame objects.

- **Chapter 4** describes the extension of the utility pole detection model for precise segmentation, enabling improved lean angle estimation and fault detection. This chapter focuses on enhancing segmentation quality and accuracy, addressing limitations of previous methods, and improving robustness to environmental challenges such as occlusions and lighting conditions.

- **Chapter 5** concludes the research with a discussion of the results, limitations, and potential future directions for improving automated utility pole inspections and maintenance.

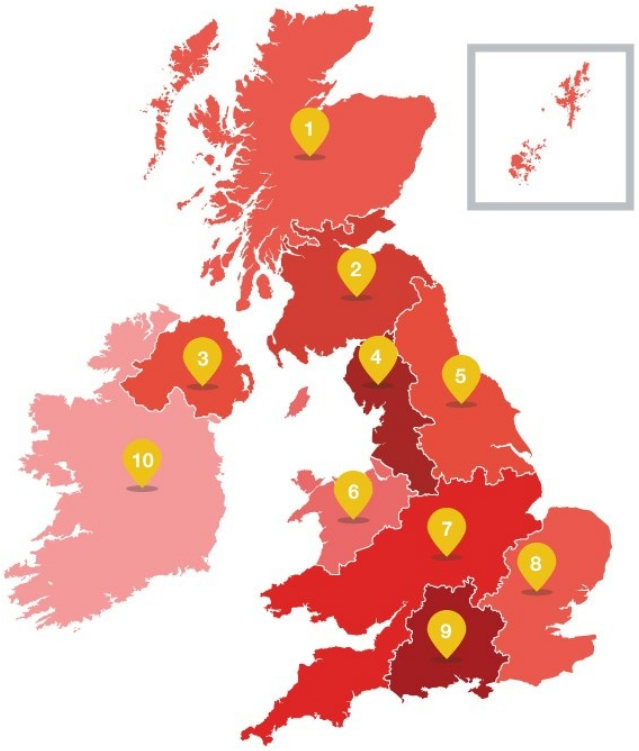
Chapter 2

Literature Review

2.1 Introduction

Overhead lines (OHLs) transport electrical energy across a power system. In Great Britain, OHLs operate at either 400kV or 275kV across the transmission network and 132/33/11kV across the distribution network [21]. Modern society is becoming increasingly dependent upon a stable and reliable source of electrical energy, where power outages are a major concern for power network operators and their customers and result in huge financial losses. In order to detect defects in OHLs as early as possible and to plan for appropriate maintenance to be carried out, power networks are inspected regularly. In the UK and Ireland, the electricity distribution network is operated by fourteen licenced Distribution Network Operators (DNOs), where each operator is responsible for distribution services in a particular geographical area. The fourteen DNOs are owned by six different groups [22], namely Electricity North West Limited, Northern Powergrid, Scottish and Southern Energy, ScottishPower Energy Networks, UK Power Networks and National Grid Electricity Distribution [22]. A map of the Distribution Network Operators and their operating regions, as represented by the Energy Networks Association (ENA), is shown in Figure 2.1, with the DNO key

included alongside the map. Each of the DNOs have a legal, moral and ethical duty to maintain all of their equipment for the duration of its intended use. This is to ensure that customers receive the best quality of service from their networks and to ensure that equipment is operating in a safe manner. To uphold these commitments, assets must be inspected regularly such that preventative maintenance or asset replacement can be implemented where needed. All of the DNOs implement visual inspection techniques of their network assets; in this context, this research focuses on the visual inspection of wooden poles in OHLs.



Map key

No.	Network operator
1	Scottish and Southern Electricity Networks
2	SP Energy Networks
3	Northern Ireland Electricity Networks
4	Electricity North West
5	Northern Powergrid
6	SP Energy Networks
7	National Grid Electricity Distribution
8	UK Power Networks
9	Scottish and Southern Electricity Networks
10	ESB Networks

Figure 2.1: Map of ENA Electricity Distribution Network Operators with the DNO key included alongside the map [23].

The methods used in visual inspection of OHLs can be categorised into manual and automated techniques, which are shown in Figure 2.2. Manual techniques include foot / vehicle patrols and aerial surveying. Automated approaches use robots to carry out the visual inspection and involve Unmanned Aerial Vehicles

(UAVs) and / or climbing robots. This research considers the visual inspection of OHLs, which can be a hazardous and problematic task for human inspectors. Each year people are killed or seriously injured when coming into contact with OHLs. Common causes of injuries or fatalities include equipment touching the lines (e.g. scaffolding, ladders), vehicles coming into contact with the lines (e.g. cranes, combine harvesters) and through human work activities (e.g. loading, stacking) and people working near the lines e.g. when carrying out line inspections.

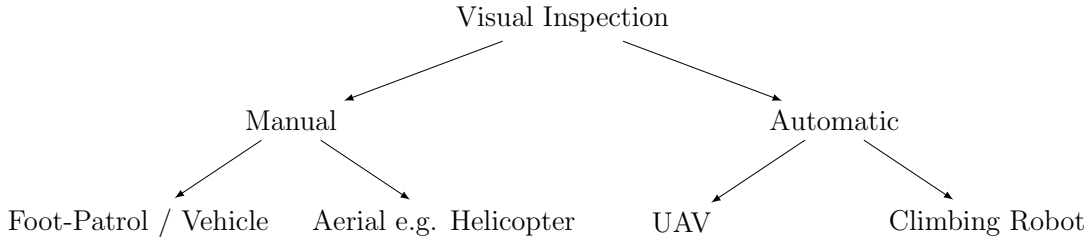


Figure 2.2: Visual Inspection overview.

As mentioned in the GS6 Guidance Note 'Avoiding Dangers from Overhead Lines' [24], one of the biggest problems is that people do not tend to notice the power lines, especially when fatigued or when bad weather makes the lines difficult to see. The very nature of a foot patrol also has health and safety implications for the inspector, where uneven or wet ground can account for slips, trips and falls leading to Lost Time Incidents (LTIs). Also, the fact that inspectors may be carrying heavy equipment can make them unstable and prone to back injuries. As previously mentioned, human visual inspection of OHLs can also be problematic in that some areas of the transmission and distribution networks can spread over hundreds and thousands of kilometres and can be installed in locations inapproachable to humans e.g. a dam on a river or a power grid in a thick forest; this can lead to some parts of the power network being inspected less often than others. Also, the visual inspections carried out by humans are subjective, for example due to lighting conditions, mood and visual acuity. In an effort to reduce

LTIs, improve visual inspection speed and save money for consumers, partially automating the visual inspections of OHLs has been attempted using for example UAVs [25] [26] and crawling robots [27]. In an effort to remove the human from the visual inspection loop altogether, achieving a fully autonomous, cost effective, high-performance, state-of-the-art visual inspection system is the ultimate goal.

2.2 Current Approaches

The visual inspection of OHLs are typically carried out by personnel on the ground or in the air. In the former, this is done by foot patrols or within vehicles. In the latter this is done from helicopters or aeroplanes. Whatever method of inspection is chosen, safety is of the utmost importance. As such, several regulations are in place to ensure the safety of persons, livestock and property [28] and to provide a quality of service to the end user. These regulations are as follows:

1. The Electricity at Work Regulations 1989 [29];
2. The Electrical Safety, Quality and Continuity Regulations (ESQCR) [30];
3. BS 7671:2018 - Requirements for Electrical Installations. IET Wiring Regulations [28].

Visual inspection criteria must meet the requirements of the aforementioned standards and regulations and may also include recommendations, guidelines and best practices from well-known technical organisations such as the Institute of Electrical and Electronic Engineers (IEEE) [31], engineering institutions such as the Institution of Engineering and Technology (IET) [32] and other reputable organisations such as the Energy Networks Association (ENA) [33] and the French organisation, the International Council on Large Electric Systems (Conseil International des Grands Réseaux Électriques) (CIGRE) [34]. Part V of the ESQCR sets mandatory requirements for OHLs in which there are five categories:

1. General restriction on the use of overhead lines;
2. Minimum height of overhead lines, wires and cables;
3. Position, insulation and protection of overhead lines;
4. Precautions against access and warnings of dangers;
5. Fitting of insulators to stay wires.

However, the ESQCR gives no guidance on the frequency or method of inspection of OHLs. OHLs are regularly inspected in order to obtain the latest information about its condition and its immediate surroundings, especially the growth of trees and other vegetation. Inspections should identify any damaged conductors, insulators, fittings and structures [35] and problems with vegetation such as overgrowing, hiding of warning signs or encroachment onto the line. Other things to consider are loose timber, landslides, washouts and anything that may potentially cause damage to the line or support [35].

Foot patrols as a method of protection, usually involves one or two inspectors who walk along a specified route performing a visual inspection from ground level [36]. In urban or congested industrial areas, foot patrols are favoured over aerial methods of inspection due to restrictions in height of flight [35]. Foot patrol inspections can be highly accurate but only for line equipment that can be seen well from the ground. The nature of the work and ground to cover means that visual inspections can be slow, tedious, monotonous and subjective, so that even larger defects can sometimes be overlooked [27].

Helicopter-assisted inspections are another common method of visual inspection. The pilot flies the helicopter whilst a camera operator takes images of the conductors, insulators, poles, stays and objects below the lines etc. Helicopters have the advantage of gaining access to remote locations and can cover large areas quickly and with ease, where an efficiency of up to 300 km per day can be easily

covered [35]. The advantage of helicopter patrols is that the inspector can look down on the lines instead of up towards the bright sky and the support fastenings at the top of towers and wooden poles can easily be identified and inspected; this would be more difficult to do with a foot patrol [35]. Although they have a high operational cost [27], helicopters provide a cost-effective solution due to the number of assets they can cover across the network in a much shorter time than foot patrol methods [36]. Regardless of the inspection method used, safety of the inspectors, public and livestock is of the utmost importance. Transmission and Distribution network operators will have their own procedures and Personal Protective Equipment (PPE) that should comply to the regulations mentioned previously and be suitable for the job at hand. There is no standard or regulation which specifies an inspection checklist for OHLs, however common defects are found on conductors, fittings, insulators, supports and foundations [37]. A description of the categories followed by some examples are presented below.

Conductors: The conductors transport electrical energy across the distribution network at high voltages, until they are stepped down to suitable voltage levels for consumers. Observable conductor defects include: fallen or sagging conductors, broken, twisted, or clashed (touching) conductors, corrosion, joint defects, and weather damage (e.g. wind, lightning, etc.). Typical examples of such defects are illustrated in Figure 2.3.

Wooden Pole: The wooden pole supports conductors that operate at distribution level voltages i.e. 33kV, 11kV and below. Being a natural material, wood is susceptible to a variety of defects, the more detectable ones include: woodpecker damage, rotting, broken / Damaged Pole. Examples of these defects are shown in Figure 2.4.

Supports: The supports, for example the stay wires in wooden pole OHLs, are responsible for keeping the pole upright and in a stable position. The supports are typically made from steel and some of their observable defects are: reduction

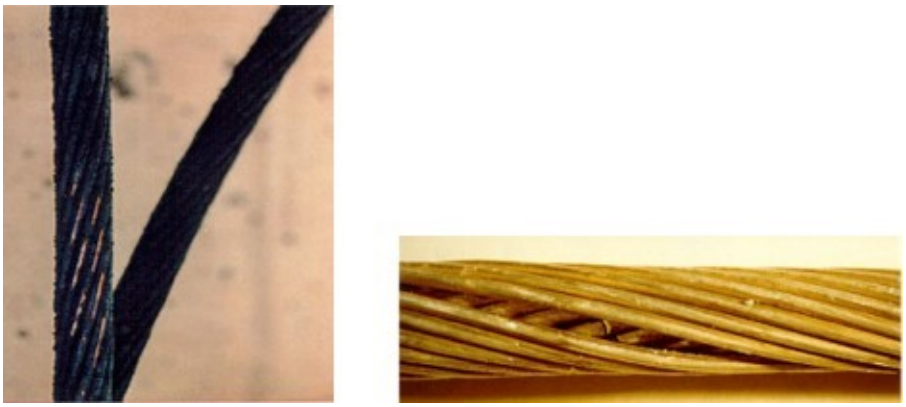


Figure 2.3: Conductor corrosion (left) and broken inner conductors (right) [37].

Figure 2.4: (UK Power Networks) Woodpecker damage (left), Pole rot (right).

in tension, corrosion, snapped / broken wires.

Insulators: The overhead line insulators electrically separate the conductors from the tower / pole structure and support the weight of the conductors. Insulators are commonly made from porcelain but are also available in glass and composite materials. The common observable defects found in insulators are: porcelain, corrosion, erosion of the glass and pin, premature ageing or internal manufacturing defects in composite materials.

Vegetation: Many OHLs are situated in and around plant life such as trees and bushes. Vegetation can pose a problem for OHLs and customers as excessive growth could hide warning signs such as 'Danger of Death', occlude the OHL altogether and falling trees could damage the OHL and disrupt power to customers.

Observable problems caused by vegetation include: pushing conductors together, signs of burning / burnt vegetation, fallen trees on lines.

Security: In many cases where high voltage equipment operates close to the public, security measures are taken to protect the public from harm and to protect the asset from vandals. Examples of security measures are fences, guards and lockable gates. Detecting a compromise in security may suggest that the protected equipment has been tampered with. Observable compromises in security include: broken security fences, open gates, damaged guards.

Clearance Distances: Clearance distances refer to the mandatory minimum height of OHLs, wires and cables as set out in the ESQCR [30]. With respect to clearance distances, defects can be observed by measuring the clearance distances and comparing them against the minimum standard.

From a visual inspection point of view, inspectors are given an inspection schedule containing a check sheet that is used to verify that the OHLs are in visually good condition. As part of the visual inspection process, high resolution photographs are taken of suspected damaged/ failed/ non-conforming parts to provide evidence of the potential defect. The photographs are then examined by a team of experienced engineers to assess the condition of the OHL [36]. Each inspected item is then given a Health Index (HI) and/or Condition Rating (CR) [38] with CR levels 1 to 4 indicating a condition from 'as new' to 'extremely poor'. Not only are visual inspections of OHLs time consuming but they are quite laborious for the inspectors. Some studies [3] [39] have shown that the performance of human visual inspection is influenced by several factors and can be broken down into four main categories, namely subject factors, physical and environmental factors, task factors and organisational factors. Table 2.1 has been reproduced from [3] and highlights the factors that might influence a human inspector. From the same paper [3], four factors from Table 2.1 were considered as being the most significant in determining inspection accuracy as they have been validated in in-

dustrial tasks; these factors were: Visual acuity, Lighting, Inspection time and Feedback. It can be appreciated that due to the many factors that can influence a visual inspector, the accuracy and performance between different inspectors inspecting the same object can vary greatly and hence repeatability of results even with the same inspector can differ.

To overcome the aforementioned disadvantages in manual visual inspection, an automated process would be more accurate, less expensive and at least as fast as helicopter-assisted inspections [27]. Early attempts to automate the visual inspection of OHLs using helicopters [26] tried to address the issue of repeated sightline adjustment of the camera operator, whilst also realising the many advantages of autonomous inspection as descried previously. One of the tasks which can introduce a source of error into the visual inspection process is that of manual sightline adjustment. A camera operator is responsible for visually locating an asset and trying to focus the camera on it. Camera operators can become disorientated as they try to stabilise the camera quickly. This behaviour could result in the asset being lost from the operators field of view or getting confused as to which asset was in focus where assets look similar. Manual adjustment of the sightline is not practical and an automated approach could alleviate some of the factors that might influence an inspectors performance and therefore reduce operator stress whilst improving the inspection speed and accuracy. Three objectives were defined in [26] to bring an automated solution for sightline adjustment into realisation:

- Quickly localise the pole within the camera's Field of View (FOV).
- Identify the location of the pole within the image, relative to the camera's FOV.
- Track the pole continuously as the helicopter moves.

Table 2.1: Factors that might influence inspector performance [3].

Subject Factors	**Task Factors continued**
Visual acuity	Paced vs unpaced
Static	Direction of movement
Dynamic	Viewing area
Peripheral	Shape of viewing area
Colour vision	Density of items
Eye Movement	Spatial distribution of items
Scanning strategies	Fault probability
Age	Fault mix
Experience	Fault conspicuity
Personality	**Organisational Factors**
Gender	Number of inspectors
Intelligence	Briefing/instructions
Physical and Environmental Factors	Feedback
Lighting	Feedforward
General	Training
Surround luminance	Selection
Lighting for colour	Standards
Aids	Time-on-task
Magnification	Rest pauses
Overlays	Shift
Viewing screen	Sleep deprivation
Closed-circuit TV	Social factors
Partitioning of display	General
Automatic scanner	Isolation of inspectors
Background noise	Working in pairs
Music-while-you-work	Effects on sampling schemes
Workplace design	Motivation
Task Factors	Incentives
Inspection time	Product price information
Stationary	Job rotation
Conveyor paced	

The initial position of the pole was located and brought into the cameras FOV using a Differential Global Positioning System (DGPS) which measures the helicopters position using satellites. The camera operator selects the location of the pole from the Geographic Information System (GIS). A transformation is then applied to the pole location and used to extract the angles required to position the camera directly at the pole. Once the camera has been focused onto the pole location, image processing techniques are used identify the pole location within the image. The coordinates of the pole within the image are then used in a closed loop system to automatically fix the position of the camera onto the pole and follow it smoothly, regardless of the helicopters movements. In order to identify the pole within an image, an image processing technique known as feature extraction was used to identify the main characteristics of the pole and to use these to locate poles within the images. Examples of pole features are the vertically straight parallel lines of the pole and 'blob' like shapes of insulators. These features are used to identify a pole within an image and to disambiguate them from similar looking objects such as trees and fences etc. The tracking algorithm was implemented on a scaled laboratory test rig on three video sequences of varying quality, good, moderate and poor. In each sequence the reported success rate in detecting the horizontal position and top of the pole is given in Table 2.2.

Table 2.2: Fraction of Successfully Located Video Streams [26].

Video Sequence	**Horizontal Position**	**Top of Pole**
Good	95%	92%
Moderate	88%	65%
Poor	75%	66%

However, Jones et al. [26] demonstrated that pole recognition did not work well when the background was cluttered. At the time of that study, the Digital Signal Processing (DSP) unit used in the experimental system could only achieve processing rates of 2–8 images per second, which was too slow for practical use.

Jones et al. [26] therefore suggested that multiple DSP units could be used to increase processing speeds to approximately 20–25 images per second.

In an effort to achieve better inspection results than traditional foot patrols or helicopter-assisted inspections whilst also aiming towards developing an autonomous visual inspection system, mobile robots was seen as one such option. Mobile robots [27] can be categorised into two types, a flying robot that flies above the power lines and a climbing robot that travels along the conductors known as UAVs and Rolling on Wires Robots (RWRs) respectively [40].

2.3 Assisted Visual Inspection of OHLs

Using UAVs for the visual inspection of OHLs has some similarities with automated helicopter inspections. In particular, both methods suffer from the technical problems of camera stabilisation, pole tracking and automatic defect detection [27]. As described in [27] a project known as the Electric power Line Exploration using Aerial Vehicle (ELEVA) [41] aimed to develop a UAV which would fly alongside a powerline being guided by a stereo computer vision system. Szeliski [42] describes stereo matching as the process of taking two or more images and estimating a 3D model of the scene by finding matching pixels in the images and converting their 2D positions into 3D depths. ELEVA developed a control system that controlled the UAVs position using inertial sensors and a 3D computer vision system that used power transmission line images to calculate the trajectory of a UAV. The computer vision system needed to detect the cable and also determine the position of the cable relative to the helicopter. These were done using the vector-gradient Hough transform and stereo vision respectively. The Hough transform is a well-known edge detection technique which can not only detect lines but also other arbitrary shapes such as circles or ellipses [43], [44]. In [20] a three-stage algorithm was developed to allow a UAV to

be able to automatically detect and completely extract power lines from aerial images that contain complex and varying natural surroundings. The focus of the algorithm was to minimise the number of missed-detection of line segments. The three stages of the algorithm were the isolation of power lines in various light conditions, select a candidate set of edges from which power lines have to be detected and the final stage worked to detect 'stripes' in an image matrix that corresponded to power lines.

Adaptive thresholding was used to isolate the power lines from the background. The algorithm used a different threshold for every pixel in each frame according to the pixel intensity in the neighbourhood. The algorithm used a blur function to determine the neighbourhood intensity as well as a Gaussian kernel to calculate the weighted mean function in a 3 x 3 neighbourhood to filter more background whilst keeping the power lines. A morphological erosion operator was used to retain the edges of a foreground region, especially straight-line segments. Finally, robust heuristics were used on pixel intensities to detect stripes in the image that corresponded to power lines. The algorithm was tested on real outdoor video footage of a 320-meter coverage of a power grid using a UAV. At least a few lines were detected in each of the frames for which the initial length is a minimum of 17 pixels, which is more than the minimum required length of 1.44 pixels for overhead detection in video monitoring [20]. There was also a reported 0% missed detection rate when it came to important line segments.

An alternative to using UAVs to perform visual inspections of OHLs is to use climbing robots, which are able to climb along the conductor of an OHL and overcome obstacles along its inspection route. Obstacle crossing, however, is one of the main challenges in climbing-robot design. The main advantage of the climbing robot is its close proximity to the line, which can support greater inspection accuracy and higher-quality image acquisition when compared with images obtained from foot-patrol, helicopter, and UAV inspection techniques.

Goncalves et al. [45] presented a review of mobile inspection units and proposed a hybrid system combining the advantages of a UAV and a climbing robot. The climbing robot is completely modular in that its main component parts can be disconnected and transported by the UAV. The UAV transport module is the docking station for the UAV where it is transported across the line. Following behind it are the modules of the climbing robot. Once an obstacle is identified, the UAV carries each module of the robot across it and back onto the line to resume its function. Modules of different types can be designed for inspection and maintenance tasks. The authors argued that this concept could provide a complete solution to inspection and maintenance and allow obstacles to be overcome. However, the work was theoretical in nature, with the design and construction of the proposed system left for future work. Due to their close proximity to the power line, climbing robots are susceptible to electromagnetic interference and potential damage to sensitive electronic components. Hence, adequate shielding is required to protect the robot from strong electric and magnetic fields.

With regards to visual inspection, one of the most important areas of research with climbing robots is the automatic detection of an obstacle on a conductor and determining its position. The work in [27] summarises several projects which implemented computer vision to automatically detect obstacles on the conductor where [46] was able to determine the types of obstacles directly from images. Once an object was believed to be detected, it was represented by simple shapes, an ellipse for a suspension insulator string, two circles left and right of the conductor for a strain insulator string. Once the object was detected, its position was determined using stereo vision. The method was applied to a real power conductor where a 7% error was achieved. In [47] stereo vision was used to extract the position and distance between the robot claws and the line. The algorithm used was able to process 480 x 320 images in near real-time which enabled the extraction of the 3D positions of the conductors.

2.4 Computer Vision-based Navigation and Inspection Systems

Generally speaking, computer vision for power line detection / tracking has consisted of six stages [48]. First is the enhancement of the input images to remove noise. Second is the removal of backgrounds, third is the use of edge detectors (such as the Canny edge detector) to produce edge maps. Fourth is the detection of straight lines from the edge maps using for example, the Hough transform. Fifth is the use of clustering algorithms for example K-means to eliminate spurious lines and to detect power lines. The sixth and final step is to track the detected power lines using visual tracking methods such as using Kalman filters and fuzzy logic. In [49], a knowledge-based approach to automatically detect power lines was proposed. A Pulse Coupled Neural Network (PCNN) filter was used to remove unnecessary information from an image where a Hough transform was then applied to detect straight lines in the filtered image. K-means clustering was then used to differentiate between power lines and other mistakable objects. Li et al. [49] concluded that an experiment on image data captured from a UAV demonstrated that the proposed method was effective in automatic power line detection. In [50] a strategy for detecting and tracking electric towers during aerial inspections was presented. Machine learning was used to automatically detect the tower whilst computer vision techniques were used to track the tower. A two-class Multi-Layer Perceptron (MLP) was trained to detect and classify the background of the towers based on Histogram of Oriented Gradients (HOG) features. The network was then applied as a sliding window detector. Tower tracking was implemented by using the Hierarchical Multi-Parametric and Multi-Resolution Inverse Compositional algorithms. The performance of the power line inspection system was tested under very difficult conditions that included various types of towers and backgrounds including varying degrees of poor-quality

images, some of which contained strong motion blur. Recalling that the system consisted of a tower detection stage and a tower tracking stage, the performance of these were investigated individually and then as a combination on High Voltage (HV) and Medium Voltage (MV) towers. When applied to HV towers, the detection stage was able to detect the position of the towers in 83% of the total frames. The combined detection + tracking stages detected towers in 87% of the total frames. When the system was applied to MV towers, the detection stage only managed to detect the position of the towers in 52% of the total frames. However, when the detection and tracking stages were combined, the detection rate rose to 88%, clearly showing that when used together, detection rates are improved. Not all the frames that contained towers were detected by the detection stage or the combined stage. Martinez et al. [50] explain that this was due to the assumptions made about tower position and size when monitoring the detection and tracking algorithms. That is, the towers in the frames were smaller or in a different position than assumed and hence the algorithms deemed the information to be irrelevant. Their reported results show that the majority of false positives occurred in the MV sequences. The authors attribute this to poor image quality and the simple structure of the towers, which led to HOG features that were not sufficiently discriminative for these tower types. Martinez et al. [50] concluded that the detection and tracking stages provided a robust strategy for detecting, localising, and tracking towers in very challenging conditions. Wang and Zhang [51] proposed an insulator recognition system that used Support Vector Machines (SVMs) trained on features extracted by Gabor filters to suppress cluttered backgrounds and classify insulator types. Testing the insulator recognition system on 96 different aerial images showed that with background suppression, the classification failure rate dropped significantly. This is attributed to the fact that on images without background suppression, the algorithm has to search across the entire image, looking for insulators. However, in images

with background suppression, most of the irrelevant information is removed and the algorithm only needs to search around the areas of the position of the insulator. Wang and Zhang [51] state that without background suppression, the average computation time was 26.57 seconds, compared with 11.87 seconds with background suppression, representing a reduction of 53.02%. The disadvantage of this technique is the training process for the SVM in which the classification results are sensitive to the choice of training samples. A significant part of the project itself was finding a good balance between positive and negative training samples (a common trend with most machine learning algorithms). Wang and Zhang [51] highlighted the importance of collecting enough positive and negative training images in various conditions, such as both good and damaged insulators shown at different angles and lighting conditions. The approaches mentioned so far with regards to the visual inspection of OHLs using computer vision have not been well-suited for high-speed, fully autonomous vision-based systems, due to their slow speed and inaccuracies.

2.4.1 Limitations in Computer Vision Inspection Methods

Many attempts have been made to develop a fully automated visual inspection system for OHLs, however none so far have been able to identify a wide range of faults. Much of the work done so far has been to address specific problems in navigation and/or inspection with varying degrees of accuracy. Being problem specific, many of the current projects on automated visual inspection of OHLs do not generalise well due to, for example, a lack of training data and poor image quality and is one of the main drawbacks that has prevented a fully autonomous visual inspection system from being developed. The majority of the current visual inspection projects rely on hand-crafted feature extractors being created

to address a particular problem which can be laborious and time consuming (such as defining and extracting the features of an insulator for example, its shape, relative size etc). These issues can be addressed by using machine learning techniques that can learn features, generalise well and provide a greater level of accuracy. This branch of Machine Learning, known as Deep Learning, uses representation learning that is a set of methods that allows a machine to take in raw data and automatically determine the representations needed for detection or classification [52].

2.5 Deep Learning for the Automatic Visual Inspection of OHLs

One of the most effective and ground-breaking deep learning techniques is the Convolutional Neural Network (CNN or ConvNet). ConvNets have the ability to learn features rather than be fed hand-designed features, which can greatly reduce the effort in designing solutions for every sub-task in an OHL inspection. CNNs generalise well and have been used in many advanced visual recognition systems, such as in medicine for classifying types of lung disease [53], traffic sign detection [54], and hand gesture recognition [55]. The adaptability of deep learning models to new input data makes them suitable for other related tasks. In the next section, the role of deep learning, and particularly the use of CNNs in achieving a fully autonomous visual inspection system for OHLs, will be addressed.

State-of-the-art deep learning techniques used for object recognition, such as Faster R-CNN [56], RetinaNet [13], YOLO v3-Tiny [12], and image classification, such as RESNET [57], can work together to detect, classify, and map OHL components. The next step would be to use deep learning semantic segmentation to extract the component of interest from the image and remove irrelevant background content. Finally, inspections can be performed on the segmented images

using, for example, texture analysis and/or vision-based anomaly detection [48]. State of the art deep learning techniques used for object recognition such as Faster R-CNN [56] and image classification such as RESNET [57] can work together to detect, classify and map OHL components. The next step would be to use deep learning semantic segmentation to extract the component of interest from the image and removing the irrelevant background content. Finally, inspections can be performed on the segmented images using for example, texture analysis and/or vision based anomaly detection etc [48]. Zhao et al. [58] present a methodology for inspecting power line insulators using deep learning convolutional networks. Their approach used a CNN model with a pre-trained multi-patch feature extractor to represent insulator condition. These features were then used to train an SVM, which classified each sample into one of six categories: normal, crack, defect, dust, sky, and tower. Zhao et al. [58] concluded that the pre-trained model achieved higher classification accuracy than hand-crafted feature-based models, reporting an mAP of 98.71%.

2.5.1 Key Components of Deep Learning and Convolutional Neural Networks

A CNN comprises several essential components, each playing a vital role in detecting and classifying complex visual patterns. For automated visual inspection of OHLs, the primary components include convolutional layers, pooling layers, activation functions, and fully connected layers, all of which enable CNNs to effectively learn and generalise from visual data.

Convolutional Layers

The **convolutional layers** are core feature extractors in a CNN. Each convolutional layer applies a set of small filters (kernels) to the input image, sliding them

across spatial dimensions to produce feature maps. These filters detect various patterns, such as edges or textures in earlier layers and more complex shapes in deeper layers. Key hyperparameters include the number of filters, filter size, stride (determining the step size of the filter), and padding (e.g., zero-padding to maintain spatial dimensions). Convolutional layers are fundamental in capturing hierarchical feature representations needed for detecting OHL components under different conditions.

An illustrative example of a convolution operation is shown in Figure 2.5, where a filter is applied to a local receptive field of the input to produce an activation in the output feature map.

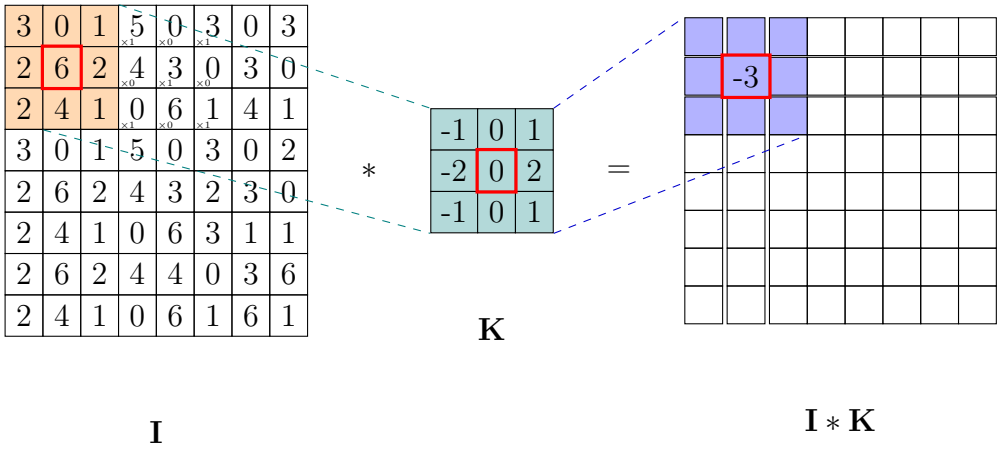


Figure 2.5: Visualisation of a 2D convolution operation. A 3×3 kernel is applied to a local patch of the input matrix, and the resulting activation is stored in the corresponding location of the output feature map. Highlighted cells show the active receptive field and the contributing kernel weights.

Pooling Layers

Pooling layers reduce the dimensionality of the feature maps, retaining essential features while minimising computational complexity. Max pooling, which selects the maximum value within a specified region, and average pooling, which computes the average, are common types. Pooling helps reduce overfitting and adds

some degree of translation invariance. A typical pooling configuration involves a 2×2 window with a stride of 2, as illustrated in Figure 2.6.

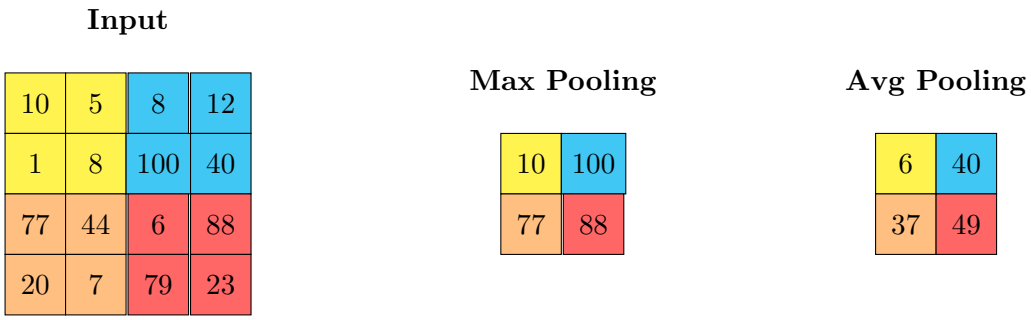


Figure 2.6: Illustration of 2x2 Max Pooling and Average Pooling (stride 2) applied to a 4x4 input matrix.

Activation Functions

Activation functions introduce non-linearity, enabling the network to learn complex, non-linear patterns. The Rectified Linear Unit (ReLU) is the most widely used activation function in CNNs [59] due to its computational efficiency and ability to mitigate the vanishing gradient problem. ReLU is defined as $f(x) = \max(0, x)$, and its variant, Leaky ReLU, introduces a small negative slope for values of $x < 0$ to keep inactive neurons responsive. These functions are crucial for modelling intricate patterns necessary for robust OHL inspection under varied conditions.

Fully Connected Layers

Fully connected layers are typically found at the end of the network and serve as the classifier. After the convolutional and pooling layers have extracted spatial features, the output is flattened into a vector and passed through dense layers. The final fully connected layer typically applies a softmax activation function for multi-class classification or a sigmoid function for binary classification, enabling the network to categorise detected objects.

Semantic Segmentation

In addition to these core layers, CNNs can incorporate other processes and methods to further enhance their utility. **Semantic segmentation** techniques, such as Fully Convolutional Networks (FCNs) and U-Net [60] architectures, enable CNNs to perform pixel-wise classification of images, effectively separating relevant components (such as power lines and insulators) from the background. This process enhances the accuracy of OHL component detection by isolating objects of interest for further analysis and can help filter out visual noise.

Transfer Learning

Transfer learning allows the CNN to leverage knowledge from pre-trained models on related tasks to improve performance on new, but similar tasks, even when labelled data is limited [61]. For OHL inspection, a CNN pre-trained on general object detection datasets (e.g., ImageNet) can be fine-tuned with smaller, domain-specific datasets to achieve high accuracy in detecting OHL components. Domain adaptation, a subset of transfer learning, specifically adapts the model to target domains with limited training data by aligning shared feature spaces between related tasks.

Image Augmentation

Image augmentation techniques generate additional variations of existing data to overcome limited dataset sizes. Common transformations include rotation, flipping, contrast and brightness adjustments, which simulate diverse real-world conditions and improve the model's robustness by exposing it to a broader range of input variations [62]. Examples of these augmentations are shown in Figure 2.7.

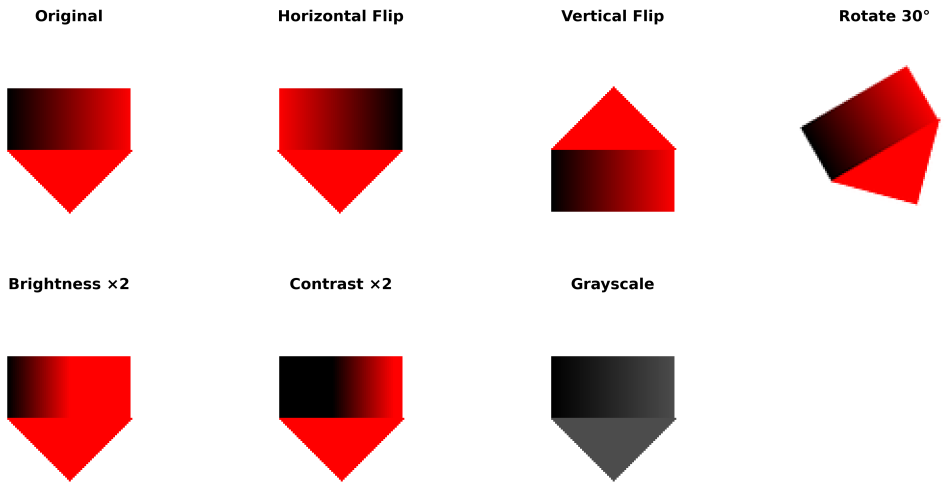


Figure 2.7: Augmentations

Evaluation Metrics

To assess the performance of CNNs in object detection tasks, several evaluation metrics are utilised, including precision, recall, average precision, and the F1 score [63]. These metrics are calculated over a range of Intersection over Union (IoU) values from 0 to 1.

When presented with an image, the model makes three predictions for each object it detects:

- **Object Class**: Identifies the type of object, such as a pole or sign.
- **Bounding Box Coordinates**: Specifies the coordinates of the bounding box, describing a rectangular area in the image that contains the predicted object.
- **Confidence Score**: Represents the model's confidence in its prediction, ranging from zero to one.

Intersection over Union (IoU) The IoU measures the accuracy of the bounding box predictions. Perfectly aligned bounding boxes have an IoU value of one, whereas non-overlapping bounding boxes have an IoU value of zero. Each prediction pairs with each ground-truth object, and the IoU calculates their alignment.

A correct prediction, known as a True Positive (TP), occurs when the IoU value exceeds a threshold. Following popular benchmarks such as COCO [64], PASCAL VOC [65], and ILSVRC [66], an IoU threshold of 0.5 is commonly used.

If multiple true positives exist for the same object, the one with the highest IoU value remains, while the remaining predictions become False Positives (FP). If more than one prediction has the same largest IoU value among a set of predictions, the one with the greatest confidence score is considered a TP, and the others are FPs. A false positive also occurs when the model makes a prediction that does not match any ground-truth object. When the model fails to detect a ground-truth object, this object is a False Negative (FN).

Precision and Recall Precision and recall are two key performance metrics that provide measures of the quality and quantity of predictions, respectively. Precision is the ratio of true positives to the total number of predictions, while recall is the ratio of true positives to the total number of ground-truth objects. They are defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN} \tag{2.1}$$

Average Precision Practitioners often use a metric known as average precision (AP) to summarise the model's precision across different recall levels. An algorithm interpolates precision over 101 recall values from 0 to 1, then divides the area under the resulting precision-recall curve by the number of recall values to compute the AP [64]:

$$AP = \frac{1}{N} \sum_{r \in \{0, \frac{1}{N-1}, \ldots, 1\}} P_{\text{interp}}(r) \tag{2.2}$$

where N is the number of recall values, r is the recall value, and $P_{\text{interp}}(r)$ is the interpolated precision at recall r.

F1 Score The F1 score, also known as the harmonic mean, combines precision and recall into a single metric. It gives more weight to low values, resulting in a high score only when both precision and recall are high:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{TP}{TP + \frac{FN+FP}{2}} \tag{2.3}$$

Generalised Intersection over Union (GIoU)

The **Generalised Intersection over Union (GIoU)** is an advanced metric that improves upon the standard IoU by considering the distance between bounding boxes, which is crucial for optimising non-overlapping predictions [67]. It is defined as:

$$\text{GIoU} = \text{IoU} - \frac{A^c - U}{A^c} \tag{2.4}$$

where A^c is the area of the smallest convex shape enclosing both the predicted and ground-truth bounding boxes, and U is the union area of the bounding boxes.

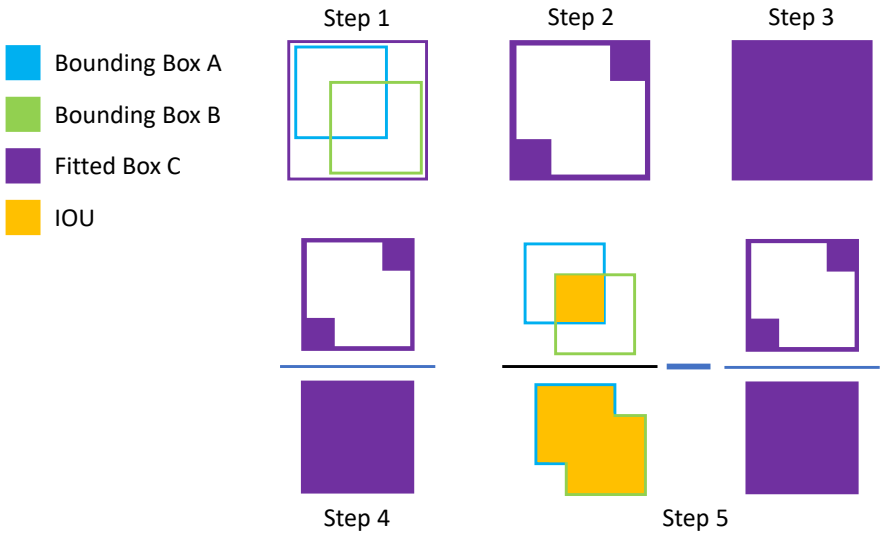


Figure 2.8: Generalised Intersection Over Union (GIoU).

GIoU helps in situations where the predicted and ground-truth bounding boxes do not overlap, providing a more informative loss function for training object detection models.

2.5.2 The Detection Transformer (DETR) Model

The Detection Transformer (DETR) [68] is a novel model that treats object detection as a direct set prediction task, removing the need for traditional region proposals or anchor boxes. DETR's design is especially well-suited for detecting wooden utility poles and warning signs, effectively handling fine-grained disambiguation and complex scenes. Its key advantages include:

1. **Attention mechanism.**
   DETR leverages the Transformer's [69] self-attention to focus contextually on parts of an image, effectively handling object relationships and variances.

2. **CNN Synergy**: By integrating CNNs for feature extraction and Transformers for contextual understanding, DETR combines local and global perspectives.

3. **Simplified Pipeline**: By bypassing anchor boxes and region proposals, DETR greatly streamlines the detection process.

4. **Parallelisation**: The Transformer enables parallel processing of the entire image, enhancing both speed and holistic understanding of image content.

By unifying global self-attention with CNN feature extraction, DETR excels at reasoning about objects in cluttered scenes. The model's parallel nature also speeds up training and inference. These advantages are particularly valuable when distinguishing objects with subtle differences, such as poles vs. tree trunks. DETR first uses a CNN backbone (e.g. ResNet-50) to extract feature maps from an input image. These feature maps are passed into a Transformer encoder-decoder, allowing the model to reason holistically about the image. Unlike earlier methods, DETR predicts a fixed number of objects in a single pass, thereby dispensing with anchor boxes and non-maximum suppression (see Figure 2.9).

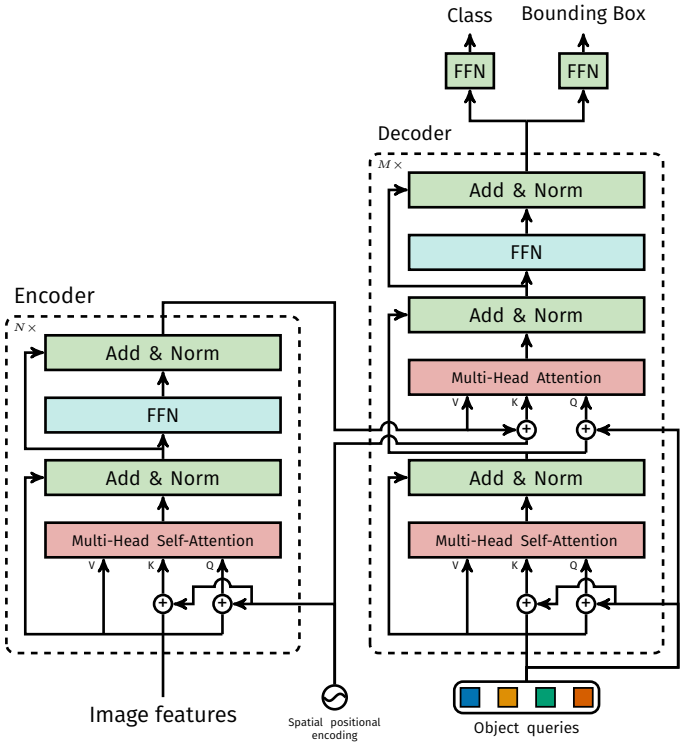


Figure 2.9: DETR Architecture [68].

Feature Extraction An input image of size $H \times W$ with three colour channels (red, green, blue) is passed through a CNN backbone (ResNet-50). The resulting feature map

$$\mathbf{f} \in \mathbb{R}^{C \times H \times W}$$

is then reduced from C to d channels by a 1×1 convolution, forming z_0. Figure 2.10 shows how the input image is broken down into its RGB channels.

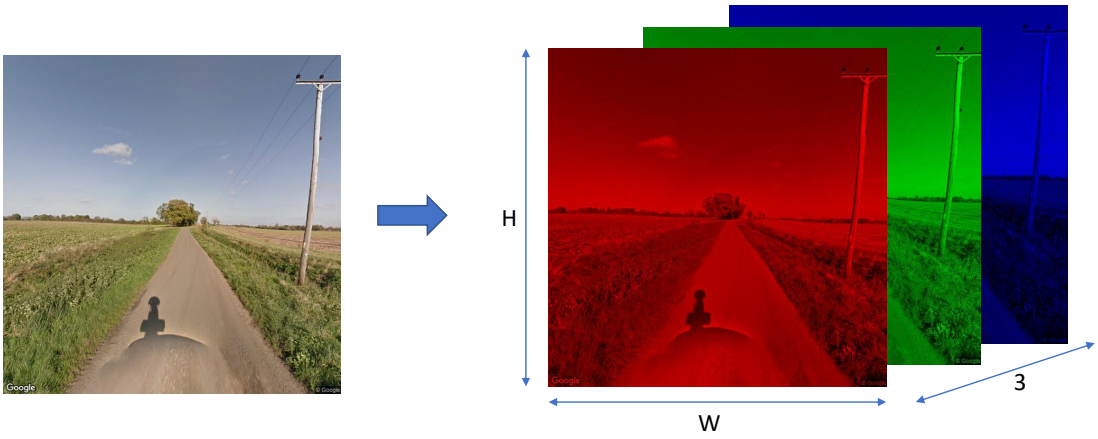


Figure 2.10: Input Image Decomposed into Three Channels.

These feature maps are extracted by a ResNet-50 backbone as illustrated in Figure 2.11.

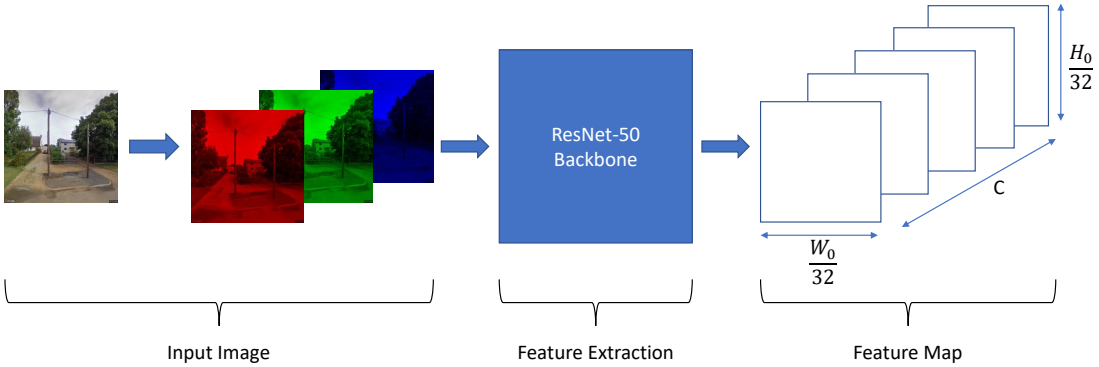


Figure 2.11: Feature Extraction Using ResNet-50 Backbone.

Positional encodings are then added to z_0 to retain spatial information, and

the feature maps are reshaped into a sequence of size $d \times HW$ for the Transformer encoder (Figure 2.12).

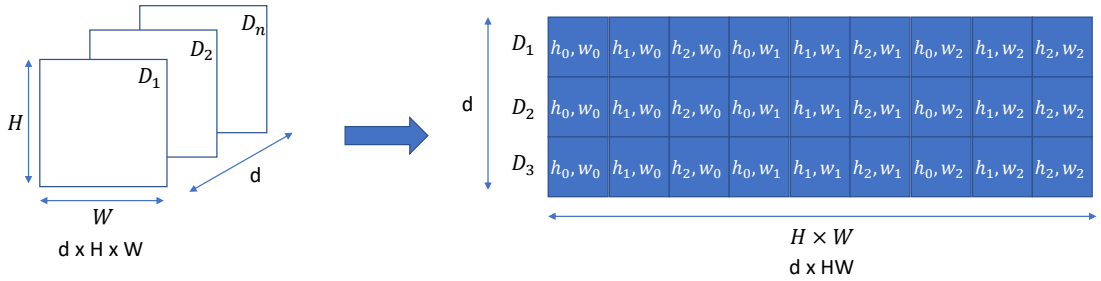


Figure 2.12: DETR Input Sequence Prepared for the Transformer.

Transformer Encoder–Decoder Originally introduced by Vaswani et al. [69], the Transformer architecture underpins DETR's ability to capture long-range dependencies. Its encoder and decoder collaborate to generate final object predictions.

Transformers originated in natural language processing [69], but DETR exemplifies their adaptability to computer vision. Key components include:

- **Multi-Head Attention**:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O, \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \end{aligned}$$

  where Q, K, V are query, key, and value matrices extracted from the input, and W^Q, W^K, W^V, W^O are learned projection parameters.

- **Positional Encodings**: Since the Transformer is inherently permutation-invariant, DETR requires explicit position information. For each position pos in the sequence and each hidden dimension index i,

$$\text{PE}_{(\text{pos},2i)} = \sin\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right), \quad \text{PE}_{(\text{pos},2i+1)} = \cos\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right),$$

where d_{model} is the dimension of the Transformer embeddings. These sine and cosine functions inject a notion of "spatial ordering" in the sequence, helping the model attend to relative positions in the image. Sine and cosine encodings are used instead of a single normalised position value because they provide a bounded, multi-frequency representation of location. This allows nearby and distant positions to be distinguished across several spatial scales, while also allowing relative position relationships to be represented through phase differences. A single normalised coordinate would encode absolute order only and would provide less information to the self-attention layers about relative spatial structure.

- **Self-attention mechanism**:

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\top}{\sqrt{d_k}}\Big)\, V,$$

where d_k is the dimension of the keys K. This defines how each token (or feature location) attends to all others, allowing DETR to compute correlations across the entire image.

The Transformer encoder uses several self-attention layers and position-wise feed-forward networks to process the feature sequence z_0. It captures dependencies across the image, which is critical for distinguishing visually similar objects such as poles and tree trunks. Figure 2.13 shows the multi-head self-attention used in the encoder.

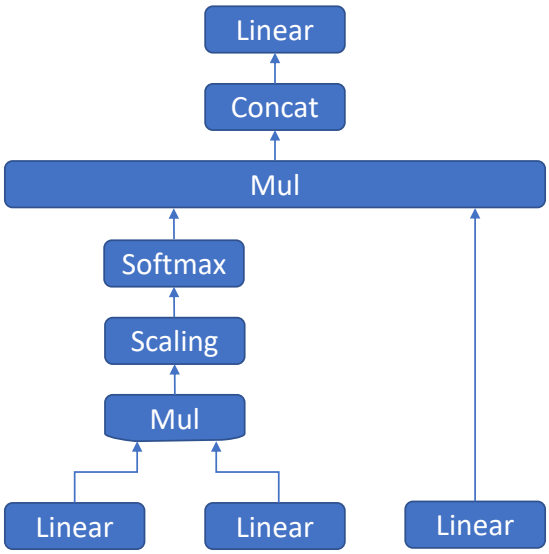


Figure 2.13: Multi-Head Self-Attention in DETR's Encoder.

The Transformer decoder receives a set of learned object queries, which interact with encoder outputs via multi-head attention. Processed in parallel, these queries yield predictions for class labels and bounding boxes (see Figure 2.14).

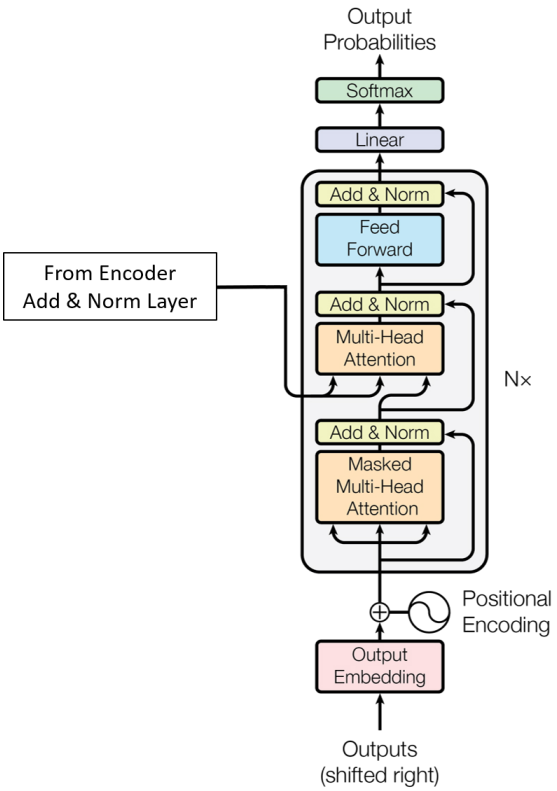


Figure 2.14: Decoder Architecture in DETR.

End-to-End Object Detection with DETR DETR formulates detection as set prediction. A fixed number of objects is predicted in one go, with bipartite matching (via the Hungarian Algorithm [70]) assigning predictions to ground truth. This obviates the need for anchor boxes and post-processing steps such as non-maximum suppression.

To match predicted objects with ground truth objects optimally, DETR employs the Hungarian Algorithm, which minimises a global cost L_{match}. This cost includes classification errors and bounding box errors (including the GIoU metric [67]). The optimal matching $\hat{\sigma}$ is found by:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i=1}^{N} L_{\text{match}}\big(y_i, \hat{y}_{\sigma(i)}\big), \tag{2.5}$$

where:

- N is the number of objects,
- y_i is the ground truth for the i-th object (including class label c_i and bounding box b_i),
- $\hat{y}_{\sigma(i)}$ is the predicted object matched to the i-th ground truth object via permutation σ.

The matching cost L_{match} is defined as:

$$L_{\text{match}}\big(y_i, \hat{y}_{\sigma(i)}\big) = -1_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) \; + \; 1_{\{c_i \neq \varnothing\}} L_{\text{box}}\big(b_i, \hat{b}_{\sigma(i)}\big), \tag{2.6}$$

where:

- $\hat{p}_{\sigma(i)}(c_i)$ is the predicted probability of the correct class c_i,
- L_{box} is a bounding box regression loss function (detailed below),
- $1_{\{c_i \neq \varnothing\}}$ is an indicator function that is 1 if the object is not a "no object" class, and 0 otherwise.

Loss Function DETR's overall loss combines classification and bounding box terms, including both L_1 and GIoU components, weighted by hyperparameters (e.g. λ_{GIoU} and λ_{L1}). For a predicted bounding box $\hat{b}_i$ matched to a ground truth box $b_{\sigma(i)}$, the bounding box loss is:

$$\mathbf{L}_{\text{box}}\big(b_{\sigma(i)}, \hat{b}_i\big) = \lambda_{\text{GIoU}}\, \mathbf{L}_{\text{GIoU}}\big(b_{\sigma(i)}, \hat{b}_i\big) + \lambda_{L1}\, \|b_{\sigma(i)} - \hat{b}_i\|_1, \tag{2.7}$$

where:

- $b_{\sigma(i)}$ is the ground truth bounding box for the i-th object,
- $\hat{b}_i$ is the predicted bounding box,

- $\mathbf{L}_{\text{GIoU}}$ is the GIoU loss measuring overlap quality,
- $\|\cdot\|_1$ is the L_1 distance,
- λ_{GIoU} and λ_{L1} control the relative weighting of each loss term.

Training Strategy

DETR uses a composite loss that combines classification, L_1, and GIoU terms, scaled by hyperparameters (e.g. $\lambda_{\text{bbox}} = 5$ and $\lambda_{\text{GIoU}} = 2$). An AdamW [71] optimiser with different learning rates for the backbone and Transformer ensures fine-grained control of model updates. Training loss is monitored per epoch to guide hyperparameter tuning and mitigate overfitting.

DETR exemplifies the synergy between CNNs and Transformers for object detection, operating end-to-end without hand-crafted proposals or anchor mechanisms. Through global self-attention, DETR excels in detecting objects in challenging settings, paving the way for further research on Transformer-based methods in computer vision.

2.5.3 Challenges of Deep Learning in the Visual Inspection of OHLs

The applications of deep learning in the power industry are fairly recent, and as such, there are a number of challenges associated with the automatic visual inspection and assessment of OHLs. One significant challenge is the limited availability of training data on which to train deep learning models. Furthermore, each image in the dataset must be labelled so that the network can learn to distinguish correct detections from incorrect classifications; this labelling process is time-consuming. One approach to addressing this limitation is to manually create a dataset consisting of positive and negative examples of OHL components,

as well as other visually similar objects and background features, such as vegetation, buildings, street furniture, and other vertical structures. These challenges are illustrated in Figure 2.15, which shows examples of vegetation clutter, small warning signs, pole-like objects, and partial occlusion in utility-pole imagery.

(a) A narrow wooden pole can blend with vegetation and other vertical scene structure.

(b) Warning signs are small objects and can occupy only a small part of an inspection image.

(c) Pole-like objects such as street lights, road signs, chimneys, and masts can be visually ambiguous.

(d) Occlusion, partial visibility, and competing scene content can affect localisation and segmentation.

Figure 2.15: Graphical examples of deep-learning visual-inspection challenges encountered in utility-pole imagery.

By including both positive and negative examples in the dataset, the model can learn not only to recognise OHL components but also to distinguish them from

other visually similar elements that might appear in inspection images. Positive examples represent the components of interest, such as insulators, conductors, and towers, while negative examples include objects that could be misidentified as OHL components, such as tree branches or structural elements unrelated to the power lines. This structure helps the model learn more robustly, making it better equipped to identify genuine components and ignore irrelevant or distracting elements.

Once the dataset with positive and negative examples is prepared, it is used in the training phase of the model. The labelled images are fed through the network, where each image is processed by the convolutional layers to extract visual features. During training, the model gradually learns to associate specific features with either positive or negative labels, effectively building a decision boundary between OHL components and non-component objects. This enables the model to classify new, unseen images with greater accuracy by recognising which features are indicative of an OHL component and which are not.

The positive and negative dataset is particularly useful for developing a model that can perform well in real-world scenarios where varied and cluttered backgrounds are common. This approach, combined with additional techniques like data augmentation, helps improve the model's ability to generalise and reduces the likelihood of misclassifications due to irrelevant background elements. In practice, this means the trained model will be more reliable in detecting and assessing OHL components, even in complex environments, ultimately enhancing the automation of the visual inspection process. The problem with manually creating a data set however is that it is very time consuming, expensive and tedious. This is due to the fact that a lot of good quality images are needed of each component and other objects of interest. Also, the images in each class must be of different shapes, sizes, orientations, lighting conditions such that the deep learning model can robustly identify the component with ease and generalise. In cases where

finding examples of objects in different configurations proves difficult, image augmentation software [72] [73] can be used to take in an image and output many different configurations (flipping, cropping, inverting, rotating etc) of the same image to greatly expand the dataset. Acquiring thousands of images per class is a feat in itself. To tackle the problem of having only a small dataset, domain adaption and transfer learning [74] could be one such approach. Domain adaption is a method of transfer learning which uses labelled data in one or more source domains, to learn a classifier for unseen or unlabelled data in a target domain. Naturally, the two domains must be related so that class labels are meaningful across both. It then becomes a standard machine learning problem whereby a model is trained on the images from related tasks and is then fine-tuned for the target task that has the very few training examples.

Another challenge is the detection of small OHL components and therefore the detection of any faults associated with them. Many popular object detection architectures such as Faster-RCNN [56], R-FCN [75], SSD [76] and YOLO [77] perform poorly on detecting very small objects [78]. Also, it has been observed [78] that the input image resolution can significantly impact detection accuracy. In fact, decreasing the resolution by a factor of two, consistently lowers accuracy and inference time by an average of 15.88% and 27.4% respectively. Therefore the problem of object detectors performing rather poorly at recognising small objects can be improved by using high resolution input images, which [78] confirms. Once the small objects have been detected in the high-resolution image, they can be cropped and used as inputs to fault detection models.

Another challenge is the detection of OHLs and their components in cluttered backgrounds. Unfortunately, the images captured by UAVs or helicopters can show the power conductors as very thin lines and components as very small objects which does not present many features to a deep learning model on which to learn from. Another factor hindering the detection of OHLs and their components

is the weather and lighting conditions. Snow for example can cover up a particular component and the light being too bright or too dull can make an object completely unidentifiable. Tackling the problem of the powerline showing as a thin line in images and therefore lacking rich features on which to train an object detection model, several image processing techniques can be used to enhance the image, using edge detection such as the Canny edge detector [79], DeepEdge [80] and DeepCountour [81] and line detection such as using the Hough Transform [44] and the Radon Transform [82]. Isolating the power lines and removing clustered backgrounds can be achieved by using techniques such as DPN [83] and Mask R-CNN [84]. Spurious lines can then be removed using techniques such as K-means clustering [85] and fuzzy C-means clustering [86].

In some instances, during a visual inspection, a new type of component or fault could appear which the deep learning network has not encountered before and is therefore unable to correctly classify it.

2.5.4 Present Work in Utility Pole Detection Using Deep Learning

Accurate detection and localisation of utility poles is essential for maintaining and enhancing power distribution networks. Zhang et al. [14] developed an automated process to detect Utility Poles with Crossarms (UPC) using deep learning techniques applied to Google Street View images. Their two-stage approach first leveraged RetinaNet, a deep learning model commonly used in object detection, which was trained on 2,500 images of UPCs and evaluated on 500 validation images. The model performed well, achieving an mAP score of 0.78 on a test set of 500 images with an IoU threshold of 0.3. This lower-than-usual IoU threshold was necessary due to the complexity of the detection task and the variability of UPC visibility in the images. In the second stage, Zhang et al. employed a brute-

force-based line-of-bearing (LOB) technique to estimate the location of the poles by calculating azimuth angles from multiple images with different viewpoints. Such angles were calculated with an image-centred approach, taking into account the pixel offset from the image centre to estimate angles. This method allowed the model to estimate pole locations with a 10-metre accuracy on average, across a test set of 500 images. By automating both pole detection and localisation, this approach demonstrated significant time-saving advantages over traditional, labour-intensive surveys, while maintaining sufficient accuracy for practical applications in urban planning and utility management.

Expanding on this work, Zhang et al. [16] introduced a new approach that aimed to improve the detection of smaller, critical components of utility poles, particularly pole caps. These small components protect the poles from environmental damage, and their detection is essential for assessing the poles' overall condition. The authors developed a three-stage process. First, a Faster R-CNN model was used to detect utility poles in images. Then, a two-stage zoom-in detection method focused on detecting the pole cap, which involved cropping the top part of the pole image, where the cap is typically located. The challenge of class imbalance—since there are far more examples of poles with intact caps than missing caps—was addressed through negative sample resampling and re-weighting [87] [88]. This ensured that the model could better handle the rare instances of missing caps. Their model achieved an mAP score of 0.796, demonstrating high accuracy in detecting small objects like pole caps, despite the inherent difficulty of the task. Furthermore, the authors incorporated oversampling and transfer learning to enhance the model's performance, allowing it to generalise better with limited data. This work highlighted the importance of using advanced techniques such as zoom-in detection and class re-weighting to improve detection accuracy for small objects in utility pole inspections, showcasing the potential of deep learning to address complex, real-world inspection tasks.

Meanwhile, Park et al. [15] took utility pole inspection a step further by mounting a pole detection model on a vehicle equipped with multiple cameras. Their system used two stereo cameras in the front and six lateral cameras mounted on the sides of the vehicle, which enabled the capture of high-resolution images of poles, transformers, switches, and insulators as the vehicle moved through urban and rural areas. Their YOLO v3-tiny object detection model was trained on 1,384 images and validated on 73 images to detect utility poles and related equipment. Additionally, stereo cameras enabled the 3D localisation of poles by using ORB descriptors to match keypoints between the detected bounding boxes of objects. By triangulating these keypoints and integrating GPS data, the system could accurately geolocate utility poles in real-world coordinates. This method offered an innovative way to automate the visual inspection of power distribution infrastructure, reducing the time and labour required for manual inspections. The authors also implemented a dynamic adjustment of the tilt angle of lateral cameras to ensure that images captured were optimally aligned with the poles. The proposed system achieved high accuracy in detecting poles and associated equipment, with an mAP of 75.75% for utility poles, which further supports its practicality in automating routine inspections of power distribution networks.

Zhu et al. [10] explored the use of UAVs for automatically measuring the inclination angle of utility poles, which is crucial for assessing pole stability. Traditionally, measuring the inclination angle required manual effort, which was time-consuming and hazardous for workers. Zhu et al. employed UAVs to capture high-resolution images of poles and then applied a segmentation model (SegNet) [89] to extract the poles from the background. The extracted poles were processed using the Hough Transform, which fitted a line through the pole's silhouette, allowing for the calculation of its inclination angle. The authors trained their segmentation model on the CamVid dataset [90], adapted to segment utility poles, and applied it to their own dataset of 64 images. The system performed

well, with 80.95% of the angle errors being less than one degree when compared to ground truth measurements. This precise calculation of lean angles is critical for identifying poles that are at risk of failure, as there is not a universally defined maximum permissible lean angle for utility poles. The acceptable degree of lean depends on various factors, including the pole's design, the load it supports, and environmental conditions. Even a slight lean can indicate underlying issues such as soil erosion or structural weakness, which may compromise the pole's integrity. However, the limited size of the dataset used in this study suggests the need for further testing on larger datasets to validate the model's robustness. Despite this limitation, the work demonstrated the effectiveness of UAVs combined with deep learning techniques in automating the assessment of pole conditions, particularly in hazardous environments where manual inspection would be risky.

In another significant development, Alam et al. [17] proposed a method to predict the future resilience of utility poles using a combination of UAV-based imagery, computer vision, and machine learning. Their approach involved capturing images of utility poles with UAVs and analysing the inclination angle of the poles to assess their stability. To predict potential pole failure, Alam et al. applied bending moment theory, which calculates the forces exerted on poles by wind and gravity. The bending moment, along with the inclination angle, was then used as input to an Support Vector Machine classifier, which categorised poles into three resilience levels: resilient, moderately resilient, and non-resilient. This classification system was tested on 173 poles in Beaumont, Texas, where the method achieved a classification accuracy of 98.27%, proving highly effective in identifying poles at risk of failure. The proposed system provides utility companies with a powerful tool for preventive maintenance, reducing the likelihood of power outages caused by pole failure during extreme weather conditions. However, the authors noted that factors such as soil conditions and pole material ageing were not considered in this study, suggesting areas for future research to

further enhance the system's predictive capabilities.

Kim et al. [11] addressed the vulnerability of wooden utility poles to extreme weather events by developing a large-scale, visual-data-driven method to assess pole conditions. Using Google Street View images, the authors proposed a method for calculating the lean angle of utility poles and integrating this information into a 3D virtual city model. The model was used to simulate the structural stability of poles under varying wind conditions, providing a risk assessment framework for power distribution infrastructure. Kim et al. tested their method on a dataset of utility poles in Houston and Bryan, Texas, and found that the lean angle of poles was a key indicator of their vulnerability. They used computer vision techniques such as edge detection and the Hough Transform to analyse the pole images and calculate their leaning angles. The study demonstrated that poles with significant leaning were at higher risk of failure during high wind events, and the model could be used to prioritise poles for maintenance before extreme weather struck. Although some poles were difficult to assess due to occlusions in the images, the study showed strong potential for using visual data in predictive maintenance and disaster planning for power distribution systems.

Stefenon et al. [91] explored the application of deep learning models, particularly the Inception v3 [92] architecture, to improve the detection and classification of defects in rural electrical distribution networks. Their focus was on identifying faults such as missing insulator components or contamination, which can lead to malfunctions in the distribution system. The authors developed a dataset of 240 photographs taken during field inspections and used it to train their model, which was evaluated using several performance metrics. The Inception v3 model achieved an accuracy of 91.67%, outperforming traditional models like VGG-16 and ResNet50 in both precision and recall. Additionally, the authors implemented image pre-processing techniques such as edge detection to reduce the impact of background noise on classification accuracy. By comparing their method with

other segmentation techniques, the authors found that Sobel and Canny edge detection produced the best results. This study highlighted the potential of deep learning to revolutionise the inspection process for rural power distribution networks, offering more efficient and accurate detection of structural defects and thereby facilitating preventive maintenance efforts.

Chen et al. [93] focused on improving the recognition and localisation of utility poles in aerial images, which is a significant challenge in China's efforts to construct a smart grid. They proposed an image enhancement algorithm based on CNNs to tackle the issues of backlight and chaotic pole distribution in Google Earth images. The algorithm, DRA-Retinex, decomposed backlit images into reflectance and light maps, preserving critical details while enhancing image quality. The authors further applied particle swarm optimisation (PSO) to optimise CNN parameters, improving the model's accuracy and performance. Their method was tested against other recent algorithms and outperformed them in both subjective and objective evaluations, achieving high classification accuracy on datasets such as CIFAR10 and CIFAR100. This research demonstrated the potential of combining image enhancement with deep learning to improve the localisation of utility poles in complex environments, contributing to more efficient planning and deployment of power infrastructure in China's smart grid projects.

Dong et al. [94] tackled the problem of achieving accurate long-term localisation of mobile autonomous systems in urban environments, where utility poles serve as key landmarks for navigation. Their approach used range images generated from LiDAR scans to extract poles from the environment. They proposed a geometry-based pole extractor that leveraged the spherical projection of LiDAR point clouds to detect poles based on their geometric properties. The detected poles were then used as pseudo labels to train a deep learning network for online pole segmentation. Dong et al. demonstrated that their method outperformed baseline techniques in terms of both accuracy and runtime, achieving more reliable

pole extraction and better localisation performance. The results were validated using publicly available datasets, and the authors released their implementation to encourage further research in this area. By developing a lightweight, efficient method for pole extraction, this work represents a significant advancement in the field of autonomous navigation and urban infrastructure management.

Lastly, Manninen et al. [95] introduced a multi-stage deep learning network for the automated assessment of transmission towers using fly-by images. The proposed method used high-resolution aerial images to detect defects in transmission infrastructure, including insulators and concrete poles. The multi-stage approach included recursive region of interest extraction and ensemble modelling, which allowed the system to process large images efficiently without downsampling. Three state-of-the-art object detection networks—YOLOv5, EfficientDet, and CenterNet—were used in the detection pipeline. The authors evaluated the system's performance across three levels: tower detection, component disaggregation, and defect detection. The results showed that the ensemble method improved both precision and recall for detecting defects, with the system achieving more than 80% precision and 90% recall. Despite some challenges in detecting minor defects, the method demonstrated substantial potential for automating the inspection of transmission lines, offering a cost-effective alternative to traditional methods and enabling more frequent and detailed inspections of power distribution infrastructure.

2.6 Summary

This literature review has examined the various methods employed for the identification and assessment of utility poles, ranging from manual inspections to early automated systems and modern deep learning approaches. While traditional inspection techniques such as foot patrols and helicopter surveys have been ef-

fective to some degree, they are labour-intensive, time-consuming, and prone to human error. Automated systems, including those leveraging UAVs, have demonstrated potential in improving efficiency, but they are often limited by high costs, difficult-to-access areas, and technical challenges, such as stabilising cameras and navigating obstacles.

The review has also explored the application of deep learning models for utility pole detection and lean angle estimation. Convolutional Neural Networks, such as YOLO and RetinaNet, have become the cornerstone of modern object detection, showing improvements in accuracy and speed. However, these methods still face limitations when dealing with environmental factors like background occlusion, and the low-resolution images typical of real-world datasets. Current models also tend to overlook critical safety features, such as attached warning signs, which are legally required for compliance but are rarely integrated into detection systems. Additionally, existing methods often rely on edge detection and filtering techniques that introduce noise, reducing their accuracy in complex environments.

A key shortcoming identified in the literature is the lack of comprehensive models that integrate environmental factors into lean angle estimations. Lean angles are crucial for assessing the structural integrity of utility poles, but current techniques often produce inaccurate estimates due to their failure to account for terrain-related variations. Additionally, models trained on small, static datasets have been shown to struggle when scaled up to more diverse and real-world data, highlighting the need for more robust training sets and methods.

This thesis seeks to address these gaps by introducing a deep learning-based framework for utility pole detection and lean angle estimation. By incorporating advanced segmentation techniques, this research aims to enhance the accuracy and reliability of pole lean angle estimations in real-world conditions. Furthermore, by focusing on the detection of both poles and their attached warning signs, the proposed system offers a more comprehensive and legally compliant approach

to utility pole inspections. The novel contributions of this work promise to improve the efficiency, accuracy, and safety of electrical infrastructure maintenance, overcoming the limitations identified in the existing literature.

Chapter 3

Deep Learning for Identifying and Assessing Utility Poles and Attached Warning Signs

3.1 Introduction

The accurate identification and assessment of utility poles, particularly those with attached warning signs, are critical for maintaining the safety and integrity of power distribution networks. Utility poles that exhibit structural weaknesses or carry compromised warning signs can pose significant risks, potentially leading to power outages or safety hazards. Traditionally, these assessments have relied on manual inspections, a process that is both time-consuming and prone to human error. The emergence of deep learning technologies provides an opportunity to automate this process, enhancing both efficiency and accuracy.

Before introducing a novel deep learning approach, this chapter presents a thorough benchmarking of existing utility pole detection models, specifically RetinaNet [13], Faster R-CNN [56], and YOLO v3-Tiny [12]. These models, which

have been documented extensively in prior research [14–16], were rigorously replicated for this study, with careful adherence to the original configurations and hyperparameters detailed by their respective authors. The benchmarking involved training, validating, and testing these models on the dataset used in this research to enable a fair comparison with the novel DETR model. This systematic evaluation provides critical insights into the relative performance of established models and sets the stage for understanding the advantages of the DETR model proposed in this research.

Following this benchmarking, the chapter introduces the Detection Transformer (DETR), a cutting-edge deep learning architecture that integrates CNNs with Transformer models. The DETR model is applied to detect utility poles and evaluate the condition of attached warning signs across a large dataset provided by UK Power Networks (UKPN). This dataset, consisting of over 670,000 geographic coordinates, was meticulously processed to extract images from Google Street View, resulting in a curated dataset of 4,570 annotated images for training and evaluation.

The DETR model was benchmarked against the previously mentioned object detection models, with the objective of achieving superior accuracy and computational efficiency. The findings revealed that the DETR model significantly outperformed its counterparts, achieving an mAP of 89.35%, even when working with the lower-quality images typically provided by Google Street View. This performance highlights DETR's robustness in handling complex visual environments, making it a powerful tool for automating utility pole inspections. In particular, this chapter presents the curated UKPN–Google Street View dataset in Section 3.2, which underpins all subsequent benchmarking and experiments.

3.2 Dataset

3.2.1 Image Collection

Before training the machine learning model, two key steps were undertaken: data collection and preparation. The dataset, provided by UKPN, comprised a CSV file containing over 670,000 geographic coordinates of utility poles. A Python script was purposely developed to query Google Street View [96] for each coordinate and download four images—each at a 90-degree heading (0, 90, 180, and 270)—with a resolution of 640 ×640 pixels. Capturing images at these four orientations maximised the chance that each pole would appear in at least one view, even if it was obscured or out of frame in the others.

An example of a wooden utility pole used in this study is shown in Figure 3.1a, while a sample of the "Danger of Death" safety sign associated with these poles is shown in Figure 3.1b.

(a) Wooden Utility Pole.

(b) Danger of Death Sign.

Figure 3.1: Examples of a Wooden Utility Pole and a Danger of Death Sign.

The Python code used to automate this image collection process is provided in Appendix E.1. It reads each set of coordinates from the CSV file and uses the Google Street View Static API to download and save four images per location,

named according to their latitude, longitude, and heading.

3.2.2 Dataset Annotation Procedure

The initial list of over 670,000 geographic coordinates did not yield a usable training image for every candidate location or heading. Each coordinate could produce up to four Google Street View images, but many candidate views either did not contain a visible wooden utility pole, contained a pole that was too occluded for reliable annotation, or did not contain an attached warning sign. The final dataset of 4,570 annotated images is therefore a manually curated subset of the candidate imagery rather than an exhaustive annotation of every coordinate and heading.

Label annotation is pivotal for the deep learning model's performance in identifying and assessing the utility poles. Annotations were made using VGG Image Annotator (VIA) version 2.0.8 [97]. For the detection experiments, axis-aligned bounding boxes were drawn around each visible wooden utility pole and warning sign. Figure 3.1 provides examples of the two target object classes used during annotation.

Each bounding box is defined by four values and stored in a JavaScript Object Notation (JSON) file. The structure of the JSON file includes the image ID, the object ID, the category ID (e.g., 0 for utility poles and 1 for warning signs), and the bounding box coordinates. In the COCO annotation convention used here, `bbox` is the short name for bounding box. The bounding box coordinates are represented as:

```
{
  "image_id": 917,
  "id": 1484,
  "category_id": 0,
```

```
  "bbox": [475.0, 0.0, 14.0, 90.0],
  "iscrowd": 0,
}
```

In this format, the `bbox` array contains four values: `[x, y, width, height]`, where:

- **x**: The x-coordinate of the top-left corner of the bounding box.
- **y**: The y-coordinate of the top-left corner of the bounding box.
- **width**: The width of the bounding box.
- **height**: The height of the bounding box.

The bounding boxes are axis-aligned. Therefore, when an object is visibly tilted or when an image is rotated during augmentation, the recorded bounding box is the smallest upright rectangle that encloses the object rather than a rotated rectangle. This convention is compatible with the COCO-style object-detection pipeline, but it can cause neighbouring or rotated objects to have substantially overlapping bounding boxes. Such cases were retained as separate object instances in the annotation file, with each instance identified by its own object ID and category. During evaluation, predictions are matched to ground-truth instances using the IoU-based matching procedure described in Section 2.5.1; however, heavily overlapping boxes remain a known source of ambiguity and are considered when interpreting localisation errors.

3.2.3 Training-Validation Split

For model training and evaluation, the dataset was partitioned into training, validation, and test subsets using image-level splitting, adhering to an 80:20 split ratio as suggested by [98]. Specifically, 20% of the images were allocated for

testing, while the remaining 80% of the images were further divided into training and validation sets using the same 80:20 ratio. All object annotations belonging to a given image were kept in the same split. This image-level partition avoids leakage between objects from the same image, although images captured from nearby coordinates or different headings may still share similar environmental context. To enhance the robustness of these sets, stratified cross-validation was employed [99].

Cross-validation is a widely used model evaluation technique in machine learning and deep learning. It involves splitting the dataset into multiple folds and repeatedly training and validating the model on different subsets to evaluate its generalisation ability. In k-fold cross-validation, the dataset is divided into k folds, where k-1 folds are used for training, and the remaining fold is used for validation. This process is repeated k times, with each fold being used once as a validation set. The final model performance is often averaged over the k runs. Cross-validation helps in reducing overfitting and ensures the model's robustness on unseen data.

Stratified cross-validation is a variant of cross-validation that is particularly useful when dealing with imbalanced datasets. It ensures that each fold of the dataset has approximately the same proportion of each class as the original dataset. In this study, the dataset consists of two classes: wooden utility poles and warning signs. Since there are more images of wooden utility poles compared to warning signs, stratified cross-validation was used to ensure that each fold contains a balanced representation of both classes. This prevents the model from becoming biased towards the majority class (utility poles) and improves its performance on the minority class (warning signs).

By using stratified cross-validation, the model was trained and validated on folds that maintained the same class proportions as the overall dataset, which is particularly important in scenarios where accurate detection of both classes is

critical. In this context, accurate detection of warning signs is just as important as detecting the utility poles, as the safety signs carry crucial information.

Details regarding the dataset split are elaborated in Table 3.1.

Table 3.1: Dataset Split.

Split	**Main Ratio**	**Sub Ratio**	**Objects**		**Images**
			Poles	**Signs**	
Train	80	80	4,269	1,008	2,920
Valid		20	1,068	250	730
Test	20		1,490	552	920
Total	100		6,773	1,805	4,570

3.2.4 Data Augmentation

Data augmentation [100] was a key technique employed to artificially enhance the diversity of the dataset and improve the model's generalisation performance. Various image transformations were applied, each designed to modify the images in ways that introduce useful variations without compromising the integrity of the original features of the wooden utility poles and safety signs. These transformations helped the model become more robust to different viewing angles, lighting conditions, and minor occlusions in real-world scenarios.

In this research, the selection of image transformations was inspired by the work of Cubuk et al. [101] and Mounsaveng et al. [102]. These studies identified effective augmentations for the Imagenet dataset [66], which was also used for the pre-training of the DETR model employed in this study. The augmentations, detailed in Table 3.2, include equalisation, rotation, colour jitter, and horizontal flip. These were chosen due to their demonstrated success in enhancing model performance.

The default set of four transformations included Random Horizontal Flip,

Random Resize (to a maximum of 1333 pixels), Random Size Crop, and Random Resize (fixed). In addition, other transformations such as Colour Jitter, Equalise, and Rotation were incorporated to further diversify the training data. Table 3.2 summarises each transformation and its effect on the images.

Table 3.2: Transformations.

Transformation	**Effect**
Random Horizontal Flip	Randomly flips an image along the vertical axis with a given probability.
Random Resize	Randomly resizes an image with a given set of dimensions.
Random Size Crop	Crops the image to a random size and aspect ratio.
Colour Jitter	Randomly changes the brightness, contrast, saturation, and hue of an image.
Equalise	Equalises the image histogram.
Rotate	Rotates the image by a given angle in either direction.

Figure 3.2 demonstrates the visual effects of these transformations. Each image shows a different transformation applied to a sample utility pole image, illustrating how the dataset is augmented to increase variability and model robustness.

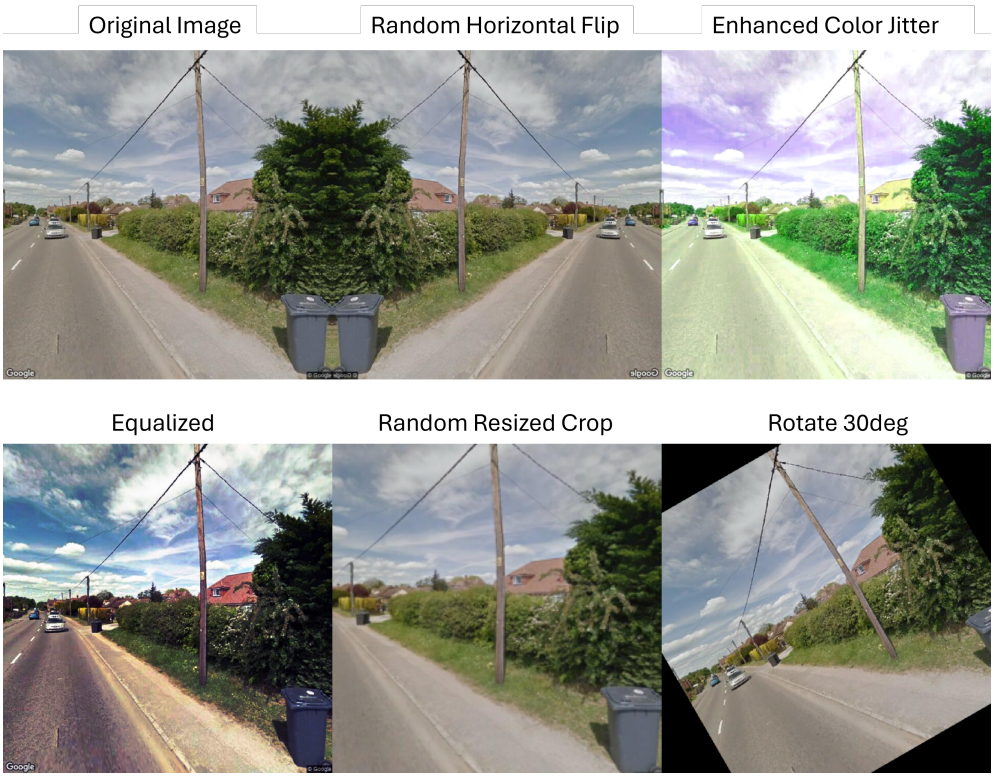


Figure 3.2: Transformations.

The Random Horizontal Flip transformation flips the image along its vertical axis with a given probability, helping the model learn invariance to left-right orientations. Random Resize and Random Size Crop transformations vary the image dimensions and aspect ratios, ensuring that the model can handle utility poles that appear at different scales and sizes. Colour Jitter alters brightness, contrast, saturation, and hue, simulating different lighting conditions. Equalisation adjusts the image histogram, distributing pixel intensity values more evenly, while Rotation introduces variability by rotating the image at random angles, ensuring robustness against different camera orientations.

3.3 Hardware and Software

The experiments were conducted using ARCHIE-WeSt [103], the Lenovo high-performance computing (HPC) facility at the University of Strathclyde. It is designed to support multi-disciplinary research, particularly in engineering and physical sciences, while also extending its capabilities to other fields and industrial collaborations.

For this project, access to the supercomputer was achieved via SSH, utilising one of the graphics processing unit (GPU) nodes equipped with an NVIDIA A100 40 GB GPU.

The software environment was configured with the NVIDIA SDK version 21.3, Python 3.9.7, and the deep learning libraries PyTorch, TorchVision, and Keras. This setup provided a robust and flexible platform for running the experiments and conducting data analysis.

The combination of ARCHIE-WeSt's powerful hardware capabilities and the advanced software stack allowed for efficient handling of computationally intensive tasks, ensuring reliable and reproducible results.

3.3.1 Computational-Cost Comparison

In addition to detection accuracy, the models were compared in terms of computational cost using the trained model checkpoints, training records, and same-HPC inference measurements where the trained-model runtime supported the measurement. Table 3.3 reports the number of trainable parameters, estimated floating-point operations (FLOPs) for a 640×640 input where supported by the runtime, recorded training time, per-image central processing unit (CPU), HPC, and GPU inference latency, and GPU memory. The training time corresponds to the documented wall-clock training duration for the final trained model configuration. The GPU latency values were measured on the OHL-UK test split in the

same HPC environment to provide a consistent comparison of deployment cost.

Table 3.3: Computational-cost evidence for the trained OHL-UK models. Parameters are in millions, floating-point operations (FLOPs) are reported as giga floating-point operations (GFLOPs), training time is in hours, latency is in ms/image, and GPU memory is in megabytes (MB).

Model	**Task**	**Params**	**GF**	**Train**	**CPU**	**HPC**	**GPU**
DETR	Object detection	41.61	77.4	19.9	–	44.0	539.0
DINO-DETR	Object detection	43.99	–	6.6	–	–	638.0
Faster R-CNN	Object detection	136.67	–	22.1	–	29.1	38242.0
RetinaNet	Object detection	55.50	–	24.6	–	56.5	38242.0
YOLOv3-Tiny	Object detection	8.68	13.1	2.7	103.3	49.9	0.0
YOLOv8	Object detection	9.84	23.6	0.6	299.7	9.2	210.1
DETR	Segmentation	42.94	224.9	10.9	–	42.3	4658.0
YOLOv8	Segmentation	10.49	35.1	0.4	409.4	10.6	258.7

Note: HPC latency and GPU memory were measured on the same HPC test environment where the trained-model runtime supported the measurement. DINO-DETR denotes DETR with Improved deNoising anchOr boxes; its GPU-memory value is taken from the training record because the same-HPC inference runtime used for timing did not support a compatible memory measurement. Dashes indicate unsupported measurements.

Figure 3.3 visualises the same-HPC GPU latency component of Table 3.3, making the deployment speed trade-off easier to compare across models.

The results show the expected trade-off between model complexity and runtime. The fastest measured GPU inference is given by YOLOv8 for both the

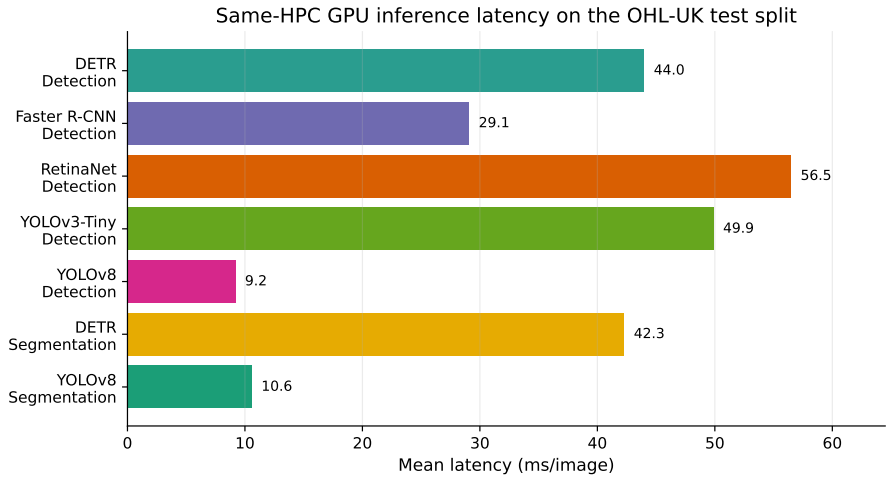


Figure 3.3: Same-HPC median GPU inference latency for the trained OHL-UK models evaluated on the held-out test split where the trained-model runtime supported the measurement.

detection and segmentation checkpoints, while DETR offers stronger pole and warning-sign accuracy at a higher inference cost. The legacy Faster R-CNN and RetinaNet checkpoints require substantially more GPU memory in the evaluated runtime, whereas YOLOv3-Tiny is lightweight but less accurate on this dataset. These cost measurements provide practical deployment context for the accuracy comparisons in the following sections.

3.4 Benchmarking Deep Learning Models for Pole Detection

This section presents the outcomes derived from utility pole detection models documented in literature [14–16]. An exhaustive replication of these models has been undertaken, adhering meticulously to the specifications detailed in the respective publications. This includes the replication of hyperparameters, image transformations, and other pertinent methodologies as presented by the original authors. Subsequently, the models were trained, validated, and tested on the

dataset utilised in this study. Such a methodological approach ensures a rigorous and equitable comparison between the performances of the previously established models and the novel DETR model introduced in this research.

3.4.1 Common Experimental Setup

This section details the key experimental factors—such as ablation study, cross-validation, and test set evaluation—that are consistently applied to all models studied.

Dataset and Hyperparameters

All experiments were conducted on a curated dataset comprising wooden utility poles in various real-world conditions. A total of 2,920 images were used for training, with 730 images set aside for validation. A separate test set of 920 images was reserved for final evaluation. The exact image augmentation techniques, training duration, and other hyperparameters were matched to the original studies where possible, ensuring a fair comparison.

Table 3.4 summarises the primary hyperparameters, which were the same for all models unless otherwise specified.

Table 3.4: Summary of Common Hyperparameters Across All Models.

Hyperparameter	**Value**
Batch size	1
Epoch count	200
Image augmentations	Horizontal flip (p=0.5)
Learning rate range (ablation)	1×10^{-3} to 1×10^{-7}
Cross-validation folds	5

The batch size was fixed at 1 for the replicated RetinaNet, YOLO v3-Tiny, and Faster R-CNN benchmarks. This setting preserved the published training protocols as closely as possible and avoided memory-pressure differences across the

legacy implementations, particularly for the larger two-stage and dense-detector models. A separate batch-size ablation was not performed because the aim of this benchmark was to compare the replicated detectors under matched literature settings while varying only the learning-rate schedule and cross-validation split.

Ablation Study

An ablation study was performed to explore how different learning rates affect model performance, while keeping other parameters (Table 3.4) fixed. The primary goal was to identify the optimal learning rate for each model under near-identical settings to the original research. By varying the learning rate in the range $[1 \times 10^{-3}, 1 \times 10^{-7}]$ and monitoring the diagnostic outputs produced by each training workflow, including training optimisation loss, validation loss when logged by the workflow, and validation-set mAP, the analysis identified the rate that maximised detection performance. Learning-rate selection was based on validation-set mAP because mAP directly measures object-detection quality by combining classification and localisation performance. Validation loss was used diagnostically where the training workflow produced a separate validation-loss trace, but it was not used as the primary selection criterion because the loss terms and scales are not directly comparable across RetinaNet, YOLO v3-Tiny, Faster R-CNN, and DETR. The corresponding plots are included in Appendices A.1–C.1, with separate plots for RetinaNet, YOLO v3-Tiny, and Faster R-CNN (see sections A.1, B.1, and C.1 respectively). These plots show training optimisation loss, validation loss where logged by the workflow, and mAP on the y-axis against epochs on the x-axis, with a separate curve provided for each learning rate setting.

Cross-Validation

Each model was subjected to a 5-fold cross-validation. In each fold, the model was trained and validated on a distinct partition of the dataset, allowing for a rigorous assessment of model robustness. The mAP values reported for the cross-validation tables are validation-fold metrics, not final held-out test-set metrics. Key metrics, including logged optimisation losses, validation-loss traces where produced by the training workflow, validation-fold mAP, and epochs to peak performance, were recorded. The mean and standard deviation of the mAP across folds provide an overview of each model's consistency. The corresponding cross-validation plots are included in Appendices A.2–C.2, with separate sections for RetinaNet, YOLO v3-Tiny, and Faster R-CNN (see Sections A.2, B.2, and C.2, respectively). These plots show training optimisation loss, validation loss where logged by the workflow, and validation mAP against epochs for each fold, using the optimum learning rate identified during the ablation study. For Faster R-CNN, the diagnostic output from the training workflow recorded the training optimisation loss and validation mAP used for model selection, but did not include a separate validation-loss curve in the same output format; validation performance is therefore reported through validation mAP and fold-summary tables rather than by an additional loss trace.

Evaluation on the Test Set

After completing cross-validation, the best-performing model configuration from each method was evaluated on a test set of 920 images containing 1,490 wooden utility poles (previously unseen by the model). For IoU values ranging from 0 to 1, the number of True Positives (TP), False Positives (FP), and False Negatives (FN) were recorded, alongside precision, recall, and the F1 score. At an IoU threshold of 0, a prediction is counted as a TP only if it is assigned to an unmatched ground-

truth object of the correct class under the confidence-ranked matching procedure; the threshold does not require spatial overlap, but duplicate predictions for the same object are still counted as FPs. This threshold therefore measures coarse object presence rather than localisation quality. The analysis across higher IoU thresholds offers insights into each model's sensitivity (recall), precision, and localisation performance under increasingly stringent overlap requirements.

A training-data sensitivity analysis was also carried out to assess how detector performance changes when the available training data is reduced. This analysis is reported for YOLOv8 as a representative retrainable detector because its training workflow allowed the same five-fold reduced-data experiment to be repeated consistently. The earlier RetinaNet, YOLOv3-Tiny, Faster R-CNN, and DETR benchmark results remain on the fixed train/validation/test split so that their comparison stays aligned with the replicated literature protocols. For the YOLOv8 experiment, detection models were trained across five folds using 50%, 60%, and 80% of the training data under a consistent evaluation protocol. The results in Table 3.5 and Figure 3.4 show that performance degrades only modestly as the training proportion is reduced, although the full 80% training setting still provides the strongest average performance.

Table 3.5: Representative YOLOv8 detection sensitivity to reduced training-data proportions across five folds. Values are mean ± standard deviation.

Training data	mAP@0.50:0.95	AP@0.50	F1
50%	49.86 ± 2.04	89.38 ± 2.49	84.49 ± 1.74
60%	50.72 ± 1.89	89.99 ± 3.11	84.89 ± 1.65
80%	51.50 ± 2.07	90.72 ± 2.86	85.32 ± 2.22

3.4.2 RetinaNet

In their study *Using Deep Learning To Identify Utility Poles with Crossarms* [14], Zhang et al. addressed the automation of utility equipment mapping through deep

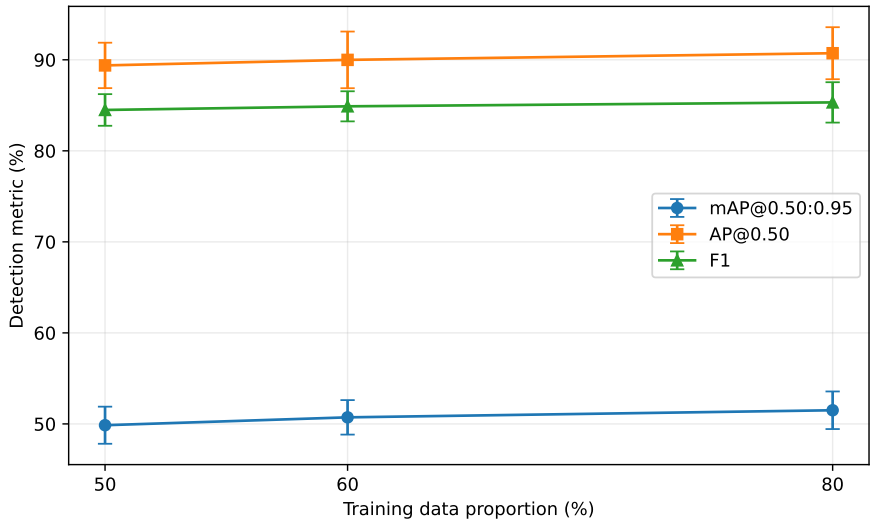


Figure 3.4: Effect of reducing the training-data proportion on representative YOLOv8 detection performance.

learning. They chose RetinaNet for its proficiency in detecting objects across diverse scales and for its focal loss function, which is designed to address class imbalance, a prevalent challenge in object detection. Their choice was driven by the need for a robust model capable of accurately identifying UPC in varying and complex urban landscapes, as captured in Google Street View images.

Ablation Study Results

Table 3.6 and the accompanying discussion present the outcome of applying the ablation study procedure (Section 3.4.1) to RetinaNet specifically. The batch size, epoch count, and other hyperparameters remained as in Table 3.4.

From Table 3.6, a learning rate of 1×10^{-5} yields the highest mAP (26.00%), confirming it as the optimal choice for subsequent cross-validation.

Table 3.6: RetinaNet ablation results on the validation set: maximum mAP observed at different learning rates. Values are percentages.

Learning Rate	**Max mAP (%)**	**Epoch of Max mAP**
1×10^{-3}	0.00	– (No Learning)
1×10^{-4}	17.00	25
1×10^{-5}	26.00	48
1×10^{-6}	14.00	42
1×10^{-7}	0.00	– (No Learning)

Cross-Validation Performance

With 1×10^{-5} determined as the best learning rate, RetinaNet was trained across each of the five data folds. Table 3.7 summarises the maximum mAP and the corresponding epoch for each fold.

Table 3.7: RetinaNet cross-validation results: maximum validation-fold mAP per fold and corresponding epoch. Values are percentages.

Fold	**Max mAP (%)**	**Epoch**
0	23.00	8
1	24.00	9
2	22.00	6
3	22.00	10
4	22.00	12
Mean	23.00	9
Std Dev	1.00	2.24

The results indicate that the average mAP achieved prior to overfitting across the five folds stands at 23.00%. This consistency is underscored by a low standard deviation of 1.00 percentage point, suggesting minimal variation and a reliable performance metric across different folds. The optimal mAP was typically attained by the 9th epoch on average, although this exhibited slightly more variability, as reflected by a standard deviation of 2.24. This indicates a moderate spread in the number of epochs needed to reach the peak mAP across the different cross-validation folds. The plots of the model performance are in Section A.2.

Comparison with Original RetinaNet Results

The original authors trained RetinaNet on a dataset of 2,500 images, achieving an mAP of 78.00% at an IoU threshold of 0.3 [14]. While this performance appears high, it benefits from a relatively lenient IoU threshold. A lower IoU threshold permits detections with less overlap with the ground truth to be classified as correct, inflating the mAP score. An IoU of 0.3, in particular, may not sufficiently penalise misaligned predictions, potentially leading to an overestimation of the model's precision in practical applications.

In contrast, when RetinaNet was applied to the dataset curated in this work —comprising 2,920 training images and 730 validation images—it achieved a maximum mAP of 23.00% under a more stringent IoU threshold of 0.5. This threshold is widely adopted in object detection as it provides a more balanced evaluation of recall and precision, ensuring that only detections with a high degree of overlap with the ground truth are considered as correct.

A key difference between the datasets lies in the structural features of the utility poles. In the original dataset, the presence of crossarms provides a distinctive feature that enhances the model's ability to detect poles. These horizontal structures create a clear silhouette, making the poles easier to distinguish against varying backgrounds. In computer vision, prominent and consistent object features aid model learning and generalisation, improving detection accuracy. Conversely, the poles in the dataset curated in this work lack crossarms, making detection more challenging. Without these defining features, utility poles can resemble other vertical structures, such as tree trunks, increasing false positives and reducing precision. This lack of distinctiveness requires the model to rely on more subtle visual cues, which is particularly challenging under a stricter IoU threshold.

The larger dataset size introduces both advantages and challenges. A broader dataset provides greater variability, which can improve generalisation by exposing

the model to diverse instances of utility poles. However, increased variability also means more complex backgrounds and scenarios, which may make detection less robust. The interplay between dataset complexity and detection performance must therefore be considered when evaluating model effectiveness.

This comparative analysis highlights that while mAP is a critical metric for assessing object detection models, its interpretation must be contextualised with the IoU threshold and dataset characteristics. A higher mAP does not necessarily imply superior performance, particularly if evaluated under a lenient IoU threshold. Additionally, dataset-specific attributes, such as object distinctiveness and variability, significantly impact detection performance. These factors underscore the importance of rigorous evaluation protocols to facilitate fair comparisons across different studies.

Evaluating on the Test Set

The RetinaNet model was evaluated on a dataset of 1,490 unseen wooden utility poles. Performance was assessed across IoU thresholds ranging from 0 to 1, with TP, FP, FN, precision, recall, and the F1 score recorded, as summarised in Table 3.8.

At an IoU threshold of 0, RetinaNet successfully detects 704 utility poles, corresponding to 47.2% of the dataset. However, this comes at the cost of 532 false positives and 786 false negatives, indicating that while the model is relatively permissive in classifying objects as poles, its recall remains below 50%.

As the IoU threshold increases, the model becomes more precise but at the expense of recall. By an IoU of 0.5, the number of correctly detected poles drops to 469, while false negatives rise to 1,021. This trend suggests that the model's localisation accuracy is limited; stricter thresholds cause it to reject many detections that are only slightly misaligned with the ground truth.

A significant trade-off between precision and recall is observed. At lower IoU

Table 3.8: RetinaNet performance metrics on the held-out test split at different IoU thresholds. Precision, recall, and F1 are reported as fractions.

IoU	TP	FP	FN	Precision	Recall	F1
0	704	532	786	0.57	0.47	0.52
0.1	653	583	837	0.53	0.44	0.48
0.2	641	595	849	0.52	0.43	0.47
0.3	615	621	875	0.50	0.41	0.45
0.4	560	676	930	0.45	0.38	0.41
0.5	469	767	1,021	0.38	0.31	0.34
0.6	339	897	1,151	0.27	0.23	0.25
0.7	220	1,016	1,270	0.18	0.15	0.16
0.8	113	1,123	1,377	0.09	0.08	0.08
0.9	32	1,204	1,458	0.03	0.02	0.02
1	0	1,236	1,490	0.00	0.00	0.00

thresholds, the model prioritises recall by accepting more detections, including a substantial number of false positives. As the threshold increases, RetinaNet becomes more conservative, rejecting imperfect matches, leading to a sharp rise in false negatives.

The F1 score, which balances precision and recall, reflects this trade-off, starting at 0.52 for IoU 0 and dropping to 0.34 at IoU 0.5. Beyond this, the model's ability to correctly classify poles deteriorates rapidly, with an F1 score of nearly zero with an IoU of 1, indicating it cannot achieve perfect localisation.

From a real-world perspective, these results highlight potential limitations of RetinaNet in utility pole detection. High false negative rates, where the model fails to detect a significant number of poles, present a critical issue in applications requiring high recall, such as safety monitoring in infrastructure networks. Similarly, high false positive rates at lower thresholds suggest that the model sometimes detects poles where none exist, which could be problematic in cluttered environments, such as those with trees and other vertical structures.

The significant decline in F1 score at higher IoU thresholds indicates that RetinaNet struggles with precise localisation. This performance degradation suggests

that the model may not be well suited for applications requiring strict localisation accuracy. The inability to balance precision and recall effectively signals the need for potential improvements, such as enhanced feature extraction, data augmentation, or alternative detection architectures.

Although RetinaNet detects some utility poles at lower IoU thresholds, its overall performance is constrained by its high false negative rate and declining F1 score at stricter thresholds. These findings suggest that further refinement or alternative detection models may be necessary for precise and comprehensive utility pole detection.

3.4.3 YOLO v3-Tiny

The next model from the literature to be benchmarked was YOLO v3-Tiny, recognised for its capability to balance rapid processing with robust detection accuracy. Park et al. [104] demonstrated its real-time performance, making it particularly suited to the dynamic and demanding milieu of electric power distribution systems, where timely detection of equipment and potential anomalies is paramount.

Ablation Study Results

Table 3.9 and the accompanying discussion present the outcome of applying the ablation study procedure (Section 3.4.1) to YOLO v3-Tiny specifically. The batch size, epoch count, and other hyperparameters remained as in Table 3.4.

Table 3.9: YOLO v3-Tiny ablation results on the validation set: maximum mAP observed at different learning rates. Values are percentages.

Learning Rate	**Max mAP (%)**	**Epoch of Max mAP**
1×10^{-3}	77.00	18
1×10^{-4}	77.00	49
1×10^{-5}	73.00	181
1×10^{-6}	46.00	169
1×10^{-7}	12.00	160

From Table 3.9, the highest mAP (77.00%) appears at two learning rates, 1×10^{-3} and 1×10^{-4}. However, a learning rate of 1×10^{-3} converges much more quickly (epoch 18), making it preferable for efficient training. Notably, beyond roughly 27 epochs, further training yields diminishing returns in terms of improved mAP. Therefore, a learning rate of 1×10^{-3} and a training duration of around 27 epochs were chosen for subsequent experiments.

Cross-Validation Performance

With 1×10^{-3} established as the optimal learning rate, YOLO v3-Tiny was trained and validated using five-fold cross-validation. Table 3.10 summarises the highest mAP achieved for each fold, alongside the epoch at which it was attained.

Table 3.10: YOLO v3-Tiny cross-validation results: maximum validation-fold mAP per fold and corresponding epoch. Values are percentages.

Fold	**Max mAP (%)**	**Epoch**
0	76.00	10
1	78.00	19
2	78.00	16
3	77.00	9
4	78.00	9
Mean	77.00	12
Std Dev	1.00	4

The mean mAP across the five folds is 77.00%, with a small standard deviation of 1.00 percentage point, indicating a high level of consistency in detection performance. This suggests that YOLO v3-Tiny is robust to slight variations in the training data partitioning, reinforcing its stability across different subsets of the dataset.

On average, the highest mAP is reached around the 12^{th} epoch, though there is some variability (Std Dev = 4), implying that the optimal convergence point differs between folds. This variability may be influenced by the specific distribution of objects within each fold, suggesting that while the model reliably learns

relevant features, training dynamics can vary slightly depending on data composition.

The consistency observed in the mAP values is crucial for practical applications, as it indicates that YOLO v3-Tiny maintains a stable level of performance regardless of dataset segmentation. Such stability is particularly valuable when deploying the model in real-world environments where data distributions may shift. Additionally, the observed variation in epoch convergence times highlights an area for further optimisation, potentially through adaptive learning rate scheduling to ensure more efficient training across different folds.

In conclusion, the cross-validation results confirm YOLO v3-Tiny's efficacy and reliability in object detection. The low standard deviation in mAP values demonstrates strong generalisation ability, while the variability in epoch convergence suggests opportunities for fine-tuning training strategies. These findings establish YOLO v3-Tiny as a viable candidate for real-world deployment, where consistent detection performance across diverse conditions is essential.

Comparison with Original YOLO v3-Tiny Results

Park et al. [104] trained YOLO v3-Tiny on 1,384 images and used 73 images for validation, achieving a maximum mAP of 75.60% at an IoU threshold of 0.5 after 20 epochs. Their dataset included utility poles equipped with crossarms, which create a distinct horizontal silhouette. In contrast, the dataset curated for this work does not feature crossarms, making pole detection more challenging. Nevertheless, YOLO v3-Tiny here achieved a mean mAP of 77.00% across five folds, converging on average at epoch 12.

The presence of crossarms in Park et al.'s data likely streamlined detection by providing an additional visual cue. That YOLO v3-Tiny performed comparably or slightly better with a more difficult dataset (i.e. no crossarms) underscores the model's robustness and capacity to generalise effectively, even under more

stringent conditions.

Evaluating on the Test Set

The final YOLO v3-Tiny model configuration was evaluated on a dataset of 1,490 unseen wooden utility poles. Table 3.11 presents the model's performance at different IoU thresholds.

Table 3.11: YOLO v3-Tiny performance metrics on the held-out test split at different IoU thresholds. Precision, recall, and F1 are reported as fractions.

IoU	TP	FP	FN	Precision	Recall	F1
0	842	221	648	0.79	0.57	0.66
0.1	839	224	651	0.79	0.56	0.66
0.2	837	226	653	0.79	0.56	0.66
0.3	819	244	671	0.77	0.55	0.64
0.4	792	271	698	0.75	0.53	0.62
0.5	753	310	737	0.71	0.51	0.59
0.6	654	409	836	0.62	0.44	0.51
0.7	472	591	1,018	0.44	0.32	0.37
0.8	247	816	1,243	0.23	0.17	0.19
0.9	55	1,008	1,435	0.05	0.04	0.04
1	1	1,062	1,489	0.00	0.00	0.00

At lower IoU thresholds (0–0.2), the model correctly detects over half of the utility poles, with approximately 840 true positives and an F1 score of 0.66. However, the presence of 648 false negatives at IoU 0 highlights that a substantial fraction of poles remain undetected. This suggests that while the model is relatively lenient in classifying objects as poles, it still struggles with complete coverage.

As the IoU threshold increases, the number of true positives declines sharply, indicating that the model becomes more conservative in its detections. By an IoU of 0.5, the number of true positives drops to 753, while false negatives rise to 737. This trade-off suggests that the model's ability to generalise detections weakens when more precise localisation is required.

The F1 score, which reflects the balance between precision and recall, follows a downward trend as the IoU threshold increases. While YOLO v3-Tiny maintains an F1 score above 0.6 up to an IoU of 0.4, it drops to 0.37 at IoU 0.7 and approaches zero at an IoU of 1. This highlights the model's decreasing recall at higher thresholds, as stricter localisation criteria cause many detections to be excluded.

From a practical standpoint, YOLO v3-Tiny demonstrates strong performance in scenarios that prioritise detection speed and moderate localisation accuracy. The model's ability to detect objects at lower IoU thresholds makes it well suited for applications requiring rapid identification of utility poles without a stringent need for high-overlap precision. However, its sharp decline in recall at higher IoU values suggests limitations in tasks where accurate localisation is critical, such as structural inspections or fine-grained object recognition.

The high number of false negatives at increasing IoU thresholds suggests that the model may struggle with distinguishing poles from similar vertical structures, such as trees or background elements. This limitation could potentially be mitigated through enhanced feature extraction techniques or improved training strategies, such as incorporating additional contextual information in the dataset.

In conclusion, YOLO v3-Tiny provides a balance between detection efficiency and accuracy, excelling in rapid object detection but facing challenges in scenarios requiring strict localisation. The model's consistent performance at moderate IoU thresholds supports its viability for real-time applications, while its declining F1 score at higher IoU levels highlights areas for potential refinement. Future work could explore architectural improvements or alternative training methodologies to enhance detection robustness, particularly in high-precision applications.

3.4.4 Faster R-CNN

Faster R-CNN is renowned for its two-stage detection mechanism, striking a balance between accuracy and computational cost. Liu et al. [16] showcased its capability in identifying small objects within highly imbalanced datasets, achieving a notable mAP of 79.60%. Such results highlight the model's flexibility in specialised tasks, such as detecting pole caps in electric power distribution systems.

Ablation Study Results

Table 3.12 and the discussion below summarise the outcome of applying the ablation study procedure (Section 3.4.1) to Faster R-CNN specifically. The batch size, epoch count (200), and other hyperparameters remained as presented in Table 3.4, while the learning rate was varied from 1×10^{-3} to 1×10^{-7}.

Table 3.12: Faster R-CNN ablation results on the validation set: maximum mAP observed at different learning rates. Values are percentages.

Learning Rate	**Max mAP (%)**	**Epoch of Max mAP**
1×10^{-3}	65.11	41
1×10^{-4}	61.98	89
1×10^{-5}	58.64	143
1×10^{-6}	43.50	196
1×10^{-7}	18.38	198

From Table 3.12, the best overall performance (mAP = 65.11%) arises at a learning rate of 1×10^{-3}, converging around epoch 41. Lower rates, such as 1×10^{-4} or 1×10^{-5}, also lead to feasible mAP values, but require more epochs to converge and still do not surpass 65.11%. At very small rates (1×10^{-6} and 1×10^{-7}), training becomes less effective, plateauing at modest mAP levels. Consequently, 1×10^{-3} was deemed optimal for subsequent experiments, with little gain observed beyond approximately 50 epochs.

Cross-Validation Performance

With the learning rate of 1×10^{-3} and a maximum training length of about 50 epochs selected from the ablation results, Faster R-CNN was trained using a 5-fold cross-validation scheme. Table 3.13 presents the maximum mAP in each fold, along with the epoch at which it was attained.

Table 3.13: Faster R-CNN cross-validation results: maximum validation-fold mAP per fold and corresponding epoch. Values are percentages.

Fold	**Max mAP (%)**	**Epoch**
0	64.55	9
1	63.17	24
2	59.29	15
3	62.81	34
4	69.25	41
Mean	63.82	24
Std Dev	3.22	11

Across the five folds, the mean maximum mAP is 63.82%, with a standard deviation of 3.22 percentage points. This result suggests moderately consistent performance, although some variation is apparent (for instance, Fold 4 achieves 69.25%, whereas Fold 2 settles at 59.29%). The model converges around epoch 24 on average (Std Dev = 11), indicating that although 50 epochs are available, most folds do not require the entire range to reach peak mAP.

Comparison with Original Faster R-CNN Results

Liu et al. [16] reported a high mAP of 79.60% when applying Faster R-CNN to pole cap detection in a highly imbalanced dataset. The poles in that dataset often featured crossarms and other distinct components, making them visually distinctive. By contrast, in the dataset used in this research, poles lack such features, leading to greater ambiguity (e.g. confusing poles with tree trunks). Despite this increased difficulty, the mean mAP achieved in the present study's

cross-validation (63.82%) demonstrates robust detection. The best fold performance (69.25%) underscores the model's capacity to adapt, though the average remains lower than in Liu et al. due to the subtle nature of the target objects.

Test Set Evaluation

The final configuration of Faster R-CNN was then tested on 1,490 unseen wooden utility poles. Table 3.14 presents the model's performance at different IoU thresholds.

Table 3.14: Faster R-CNN performance metrics on the held-out test split at different IoU thresholds. Precision, recall, and F1 are reported as fractions.

IoU	TP	FP	FN	Precision	Recall	F1
0	1,324	230	166	0.85	0.89	0.87
0.1	1,309	245	181	0.84	0.88	0.86
0.2	1,280	274	210	0.82	0.86	0.84
0.3	1,241	313	249	0.80	0.83	0.82
0.4	1,149	405	341	0.74	0.77	0.75
0.5	994	560	496	0.64	0.67	0.65
0.6	718	836	772	0.46	0.48	0.47
0.7	462	1,092	1,028	0.30	0.31	0.30
0.8	204	1,350	1,286	0.13	0.14	0.13
0.9	58	1,496	1,432	0.04	0.04	0.04
1	3	1,551	1,487	0.00	0.00	0.00

At the lowest IoU threshold (0), Faster R-CNN correctly identifies most poles (TP = 1,324) with an F1 score of 0.87. However, as the threshold tightens, the number of True Positives drops, reflecting the model's challenge in producing perfectly aligned bounding boxes. At the standard IoU of 0.5, the model retains 994 TP and achieves an F1 of 0.65. By IoU = 1.0, the overlap criterion is so strict that only 3 poles are detected correctly.

Faster R-CNN achieves a reliable balance of accuracy and speed, performing best at a learning rate of 1×10^{-3} within approximately 50 epochs. Cross-validation mAPs average around 63.82, while test set performance remains strong

at lower IoU thresholds but diminishes significantly at higher ones. Compared with studies where objects carry more distinct features (e.g. crossarms), the model faces greater difficulty here, yet still demonstrates versatility and robustness. These results offer valuable insights for practitioners seeking a dependable two-stage detection framework in contexts where object definition may be subtle or ambiguous.

3.5 DETR Model

This chapter focuses on the implementation and evaluation of the DETR model [105] for the task of detecting wooden utility poles and warning signs. Building on the discussion provided in Section 2.5.2, where the DETR model's architecture, training strategies, and loss functions were examined, this section provides a detailed account of how the model was adapted and fine-tuned for the target classes.

In contrast to previous research, which has primarily focused on utility pole detection, this study treats utility poles and warning signs as separate classes. This distinction allows for a more precise comparison of the results with those reported in earlier studies. The chapter includes an ablation study to optimise hyperparameters, an evaluation of performance across cross-validation folds and test sets, and a comparison with other state-of-the-art object detection models. These evaluations provide insights into the model's effectiveness and potential for practical applications.

3.5.1 Implementation

The *DETR* model was trained on two distinct classes, namely wooden utility poles and warning signs. The ablation study for the DETR model was an extensive investigation aimed at optimising the model's hyperparameters to enhance

detection accuracy for wooden utility poles and warning signs. This comprehensive study involved a systematic variation of several key parameters, beginning with the learning rate. The learning rates tested included 0.0001, 1.00×10^{-5}, and 1.00×10^{-6}, with the model being trained for 300 epochs at each rate, maintaining a fixed batch size of 4. The Adam optimisation algorithm was employed for training, and a range of weight decay values—0.0001, 1×10^{-4}, 1×10^{-3}, 1×10^{-1}, and 1×10^{-5}—were explored to assess their impact on the model's performance. Regularisation techniques were also a focal point of the study, particularly the use of dropout to prevent overfitting. The dropout rate was varied from its default value of 0.1 to 0.2. Furthermore, the study incorporated a variety of data augmentation techniques to ensure model robustness, including default transformations such as Random Horizontal Flip, Random Resize (up to 1,333 pixels), Random Size Crop, and Random Resize (fixed). Additional transformations like Colour Jitter, Equalise, and Rotation were tested to evaluate their effects on the model's performance.

The mAP for each configuration was meticulously recorded, with the highest performing settings selected for further experiments. On average, each model configuration in the ablation study required approximately 34 hours of training. The results of the ablation study are presented in Table 3.15. The greatest mAP achieved in the study was 90.46%, with specific class performances of 90.90% for utility poles and 90.03% for warning signs, observed at 102 epochs. Consequently, the hyperparameters selected for subsequent training and evaluations were a learning rate of 1×10^{-5}, and the combination of transforms A, B, C, D, E, F, I, shown in Table 3.15).

Table 3.15: Summary of Ablation Study Results: Hyperparameter Tuning and Model Performance Metrics.

LR	**TFMS**	**Best mAP Values (%)**			**Epoch**
		Poles	**Signs**	**Mean**	
1E-4	Default	80.61	74.65	77.63	55
1E-05	Default	88.13	87.96	88.05	37
1E-06	Default	87.44	87.06	87.25	43
1E-05	F,I	90.14	89.75	89.94	91
1E-05	C,F,I	89.65	88.06	88.85	119
1E-05	C,E,F,I	90.17	87.38	88.77	131
1E-05	A,B,C,D,E,F,I	90.9	90.03	90.46	102
1E-05	A,B,C,D,F,H	90.3	89.85	90.08	92
1E-05	A,C,D,F,G	88.62	81.47	85.04	92
1E-05	A,B,C,D,E,F,J	89.88	89.79	89.83	96
1E-05	A,B,C,D,E,F,I,K	89.56	88.09	89.92	101

Note: A = Random Horizontal Flip, B = Colour Jitter, C = Equalise, D = Random Resize, E = Random Size Crop, F = Rotation, G = Weight Decay 1e-1, H = Weight Decay 1e-3, I = Weight Decay 1e-4, J = Weight Decay 1e-4, K = Dropout (0.2).

The selected DETR configuration combines a moderate learning rate with geometric and photometric augmentation. The learning rate of 1×10^{-5} provided a stable compromise between convergence and overfitting, while the selected transforms expose the detector to horizontal viewpoint changes, colour variation, contrast differences, scale changes, crops, and rotations that are common in Google Street View imagery. The weight decay term further regularises the transformer and backbone parameters, which is important given the comparatively high model capacity. The corresponding DETR cross-validation plots are provided in Appendix D, Figures D.1–D.5.

3.5.2 Utility Pole Detection

The evaluation results for the DETR model across the five cross-validation folds, as presented in Table 3.16, demonstrate a high level of accuracy and consistency in detecting wooden utility poles. The model achieved an mAP of 90.43% across the folds, with a low standard deviation of 0.61, highlighting its robustness and reliability in varying data subsets.

Examining the epochs required to achieve the highest precision for each fold reveals some variability, with epoch counts ranging from 74 to 102 and a mean of 90.60. The relatively high standard deviation in epochs (10.95) indicates some sensitivity in training duration, which may be attributed to differences in data complexity or convergence patterns across folds.

The results underscore the model's effectiveness in maintaining a balance between precision and recall, evidenced by the consistently high mAP across folds. This level of performance positions DETR as a strong candidate for deployment in practical scenarios involving utility pole detection, despite the computational demands associated with its training.

Table 3.16 presents the model's evaluation results for each fold of data.

Table 3.16: Evaluation Results.

Fold	**Poles mAP (%)**	**Epoch**
1	91.35	82
2	90.90	102
3	89.75	101
4	89.85	74
5	90.31	94
Mean	90.43	90.60
Std Dev	0.61	10.95

The DETR model was evaluated on 1,490 wooden utility poles. These were utility poles that the model has not seen before. For IoU values ranging from 0 to 1, the number of True Positives, False Positives, False Negatives, precision,

recall, and F1 score were recorded as shown in Table 3.17.

Table 3.17: DETR utility-pole performance metrics on the held-out test split at different IoU thresholds. Precision, recall, and F1 are reported as fractions.

IoU	**TP**	**FP**	**FN**	**Precision**	**Recall**	**F1**
0	1,433	86	57	0.94	0.96	0.95
0.1	1,431	88	59	0.94	0.96	0.95
0.2	1,428	91	62	0.94	0.96	0.95
0.3	1,398	121	92	0.92	0.94	0.93
0.4	1,344	175	146	0.88	0.90	0.89
0.5	1,295	224	195	0.85	0.87	0.86
0.6	1,214	305	276	0.80	0.81	0.81
0.7	1,041	478	449	0.69	0.70	0.69
0.8	746	773	744	0.49	0.50	0.50
0.9	330	1,189	1,160	0.22	0.22	0.22
1	24	1,495	1,466	0.02	0.02	0.02

The DETR model was evaluated on a test set consisting of 1,490 utility poles, offering a comprehensive view of its detection capabilities. The performance metrics, presented in Table 3.17, underscore the model's efficacy across varying IoU thresholds.

At lower IoU thresholds (0 to 0.2), the model achieved an impressive F1 score of 0.95, with TPs slightly decreasing from 1,433 to 1,428 as the IoU threshold increased. This high level of performance indicates the model's robust detection capabilities with a significant rate of correct classifications, even with minimal bounding box overlap.

As the IoU threshold increased to 0.5, which is often considered a balanced threshold for object detection tasks, the model maintained a strong performance, with 1,295 TP and an F1 score of 0.86. This demonstrates the DETR model's effectiveness in maintaining a high precision-recall balance, even as the criterion for detection accuracy becomes more stringent.

However, at higher IoU thresholds (0.7 to 1), there is a marked decline in model performance. Specifically, at an IoU of 1, the model identified only 24

TPs, with a significant increase in FPs to 1,495 and FNs to 1,466, resulting in a low F1 score of 0.02. This dramatic decrease indicates the model's challenge in achieving perfect alignment with ground truth at high precision levels, a common issue in object detection models.

Overall, the DETR model exhibits strong performance at lower IoU thresholds, with diminishing accuracy at higher thresholds. This trend highlights the need for a careful selection of the IoU threshold based on the specific requirements of utility pole detection tasks. Figure 3.5 provides representative examples from the held-out test split, including correctly detected poles, low-IoU predictions, false positives, and false negatives. These examples show that some low-IoU predictions still correspond to visually plausible pole detections, particularly where the predicted bounding box captures only part of a pole or where the annotation boundary is narrow.

Comparison with Previous Work

Training and Validation The comprehensive analysis of the four state-of-the-art object detection models, namely DETR, YOLO v3-Tiny, Faster R-CNN, and RetinaNet, reveals insightful disparities and strengths in their performance on the task of wooden utility pole detection. The dataset, meticulously partitioned into training and validation sets comprising 5,269 and 1,068 poles across 2,920 and 730 images respectively, served as the benchmark for assessing each model's efficacy.

The comparison should be interpreted as a controlled benchmark on the same thesis dataset rather than a perfectly like-for-like reproduction of every design choice in the original papers. RetinaNet, YOLO v3-Tiny, and Faster R-CNN were replicated as closely as possible from their published settings, whereas the DETR model was developed and tuned more extensively for the OHL-UK pole and warning-sign task. This difference is important when interpreting the scale of

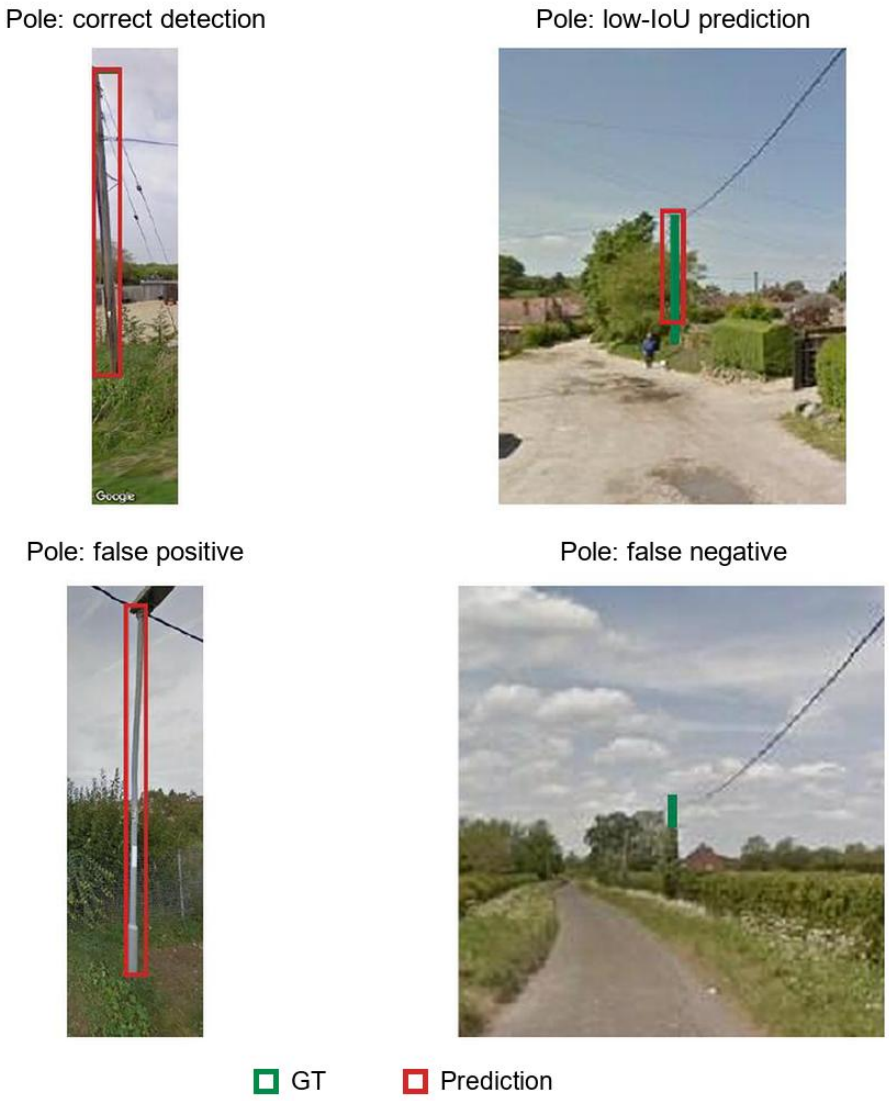


Figure 3.5: Representative DETR utility-pole detection outcomes on the held-out test split, including a correct detection, a low-IoU detection, a false positive, and a false negative. Green boxes show ground truth and red boxes show predictions.

the DETR improvement: the benchmark demonstrates the achieved performance of the final thesis pipeline, while the legacy models provide strong replicated baselines rather than fully re-optimised alternatives.

DETR emerged as the most proficient model, boasting a superior mAP of 90.43%, coupled with a minimal standard deviation of 0.61, denoting not only high precision in pole detection but also remarkable consistency across different data folds. However, this high accuracy comes with a trade-off in terms of training epochs, in that on average, it takes about 90 epochs to converge, which suggests a longer training duration compared to its counterparts.

YOLO v3-Tiny, though lagging behind DETR with an average mAP of 77.00%, showcased a low standard deviation of 1.00 percentage point, reflecting a highly consistent performance across the dataset. Moreover, the model's swift convergence, evident from the average epoch of 12, highlights its efficiency in reaching optimal performance swiftly, albeit at a lower accuracy threshold. This rapid convergence, especially in comparison to DETR, can be attributed to its fully convolutional architecture, which forgoes the computationally expensive global reasoning and self-attention mechanisms inherent in DETR's transformer-based design. YOLO v3-Tiny simplifies the detection pipeline with anchor-based predictions and avoids the iterative query refinement and bipartite matching loss used by DETR. While DETR leverages these features for superior accuracy and context awareness, YOLO v3-Tiny's lightweight and computationally efficient design makes it better suited for scenarios prioritising speed over precision.

Faster R-CNN presented a dichotomy of results, with a mean maximum mAP of 63.82%, the lowest among the models, juxtaposed with a higher standard deviation of 3.22. This indicates a less consistent performance, compounded by a moderate average epoch count of 24, which underscores a slower learning pace relative to YOLO v3-Tiny and RetinaNet.

RetinaNet registered a notably low mAP of 23.00%, despite its rapid con-

vergence, mirrored in an average epoch count of 9. The consistency of its performance, however, is underlined by a low standard deviation of 1.00 percentage point, suggesting that while the model is reliable across different subsets, its efficacy in accurately detecting utility poles is considerably lower than its counterparts.

In summary, while each model exhibits unique attributes in terms of convergence speed and performance consistency, **DETR** unequivocally stands out as the superior model in this comparative study, with respect to the training data.

Test Set Results: DETR demonstrates consistently high performance, with TP counts commencing at 1,433 at IoU 0 and diminishing to 24 at IoU 1. The model maintains robust F1 scores, consistently at 0.9 up to an IoU of 0.5, denoting a strong balance between precision and recall at lower thresholds. However, the increase in FP and FN at higher thresholds indicates a decrease in detection accuracy under stringent conditions.

Conversely, YOLO v3-Tiny initiates with moderate TP counts of 842 at IoU 0, sharply declining to 1 at IoU 1. F1 scores for YOLO v3-Tiny start at 0.66 but exhibit a significant reduction as the IoU threshold intensifies, alongside substantial increases in FP and FN. This pattern highlights the model's difficulties in maintaining precision and recall, particularly at higher IoU thresholds.

Faster R-CNN exhibits commendable TP counts, notably higher than those of YOLO v3-Tiny and RetinaNet, initiating at 1,324 at IoU 0 and concluding at 3 at IoU 1. F1 scores begin at 0.87 but demonstrate a declining trend as the IoU threshold escalates, paralleled by rising FP and FN counts. This suggests that while Faster R-CNN achieves high precision and recall at lower IoU thresholds, its performance diminishes more rapidly than DETR as the threshold becomes more stringent.

RetinaNet records the lowest TP counts and F1 scores among the models, with TP starting at 704 at IoU 0 and reducing to 0 at IoU 1. F1 scores for RetinaNet

commence at 0.52 and consistently decrease with increasing IoU thresholds. The high numbers of FP and FN underscore pronounced challenges in the model's object detection accuracy and localisation precision.

In summary, DETR emerges as the leading model, maintaining high precision and recall across a wide range of IoU thresholds. Its performance, as quantified by F1 scores, is notably superior, especially at more stringent IoU thresholds. Although Faster R-CNN exhibits potential at lower IoU thresholds, its performance wanes more markedly than that of DETR as the threshold increases. YOLO v3-Tiny and RetinaNet, despite some efficacy in certain scenarios, display considerable limitations, particularly at higher IoU thresholds. Figure 3.6 summarises this behaviour by plotting F1 score as a function of IoU threshold for the utility-pole detectors using the TP, FP, and FN counts reported in the test-set tables.

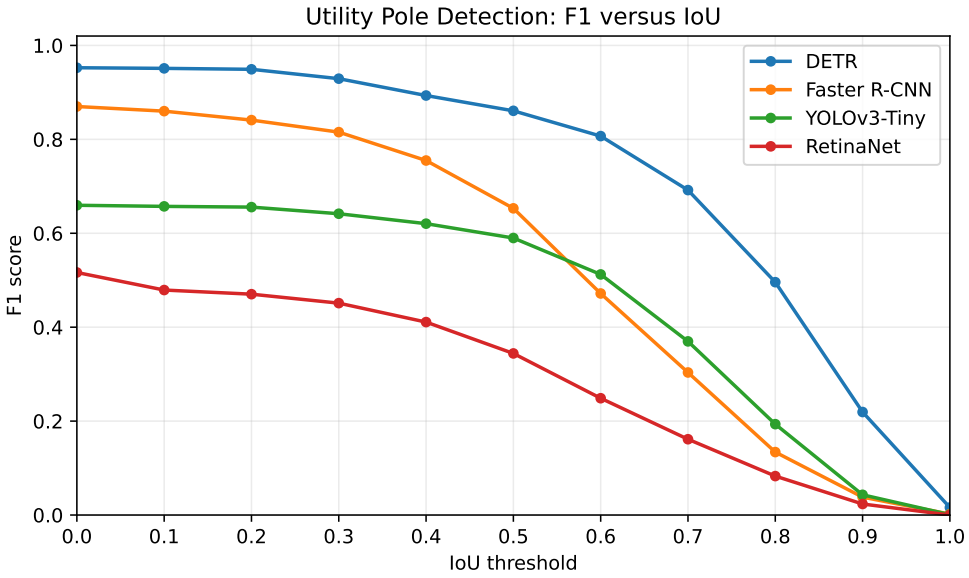


Figure 3.6: F1 score as a function of IoU threshold for the utility-pole detectors on the held-out test split.

The comprehensive assessment of DETR, YOLO v3-Tiny, Faster R-CNN, and RetinaNet across training, validation, and test sets elucidates their distinct per-

formance dynamics in the task of wooden utility pole detection.

DETR distinguishes itself as the leading model, demonstrating superior precision and recall. This is evidenced by its high mAP, consistent F1 scores across a spectrum of IoU thresholds, and robust TP counts. The model's intricate architecture, however, necessitates longer training epochs, indicative of a comprehensive learning process.

In contrast, YOLO v3-Tiny, while exhibiting notable consistency in performance, falls short in overall detection accuracy. Its challenges in maintaining precision and recall, particularly at higher IoU thresholds, are manifested in its moderate TP counts and declining F1 scores.

Faster R-CNN, with commendable TP counts and initial F1 scores, exhibits a performance decline at stricter IoU thresholds. The dichotomy of its results, characterised by a lower mean maximum mAP and higher standard deviation, places it behind DETR in terms of overall efficacy.

RetinaNet, despite its rapid training convergence, registers the lowest TP counts, F1 scores, and mAP, highlighting pronounced challenges in object detection accuracy and localisation precision.

3.5.3 Warning Sign Detection

Training on Cross-Validation Folds

Table 3.18 provides a detailed analysis of the DETR model's performance on the warning sign class, assessed through a five-fold cross-validation approach. This rigorous evaluation method ensures a comprehensive understanding of the model's generalisation capabilities.

In each fold, the model demonstrates consistent proficiency in recognising warning signs, as indicated by the high mAP values. Specifically, the mAP values fluctuate narrowly around the high 80s, implying that the model reliably detects

Table 3.18: Warning Signs Evaluation Results.

Fold	**Warning Signs mAP (%)**	**Epoch**
1	87.61	82
2	90.03	102
3	89.61	101
4	87.75	74
5	86.32	94
Mean	88.26	90.60
Std Dev	1.37	10.95

and classifies warning signs across different subsets of the data. The peak mAP recorded is 90.03 in the second fold, signifying an exemplary performance in that specific partition of the dataset.

The 'Epoch' column reflects the model's learning progression, revealing the epoch at which the maximum mAP was attained for each fold. It is noteworthy that the optimal performance is achieved at varying epochs, ranging from as early as epoch 74 in the fourth fold to as late as epoch 102 in the second fold. This variation suggests that while the model consistently reaches high precision, the convergence rate can differ depending on the specific data subset it is trained on.

The mean (denoted by $\bar{x}$) of the mAP across all folds stands at 88.26%, indicating a strong overall performance. However, the standard deviation (denoted by σ) of 1.37 percentage points in the mAP signifies some variability in the model's precision across different folds. This variability is relatively small, underscoring the model's robustness but also hinting at potential room for optimisation in certain folds.

Similarly, the mean epoch number of 90.60, coupled with a standard deviation of 10.95, underscores the model's varied learning pace across folds. This variability in convergence epochs may prompt a closer examination of the training dynamics or the consideration of early stopping criteria to optimise training efficiency.

In conclusion, the table encapsulates a strong and stable performance of the

DETR model on the warning sign class, with indications of consistent precision and insightful nuances in the model's learning behaviour across different cross-validation folds.

Evaluation on the Test Set

Table 3.19 provides a comprehensive evaluation of the DETR model's performance in detecting warning signs within a test set of 552 instances. At the lowest IoU thresholds (0.0 to 0.2), the model attains an F1 score of 0.99, signifying its high sensitivity in identifying warning signs. These thresholds are less strict, focusing more on the model's ability to detect the presence of warning signs rather than the precision of the bounding box placement.

Table 3.19: DETR warning-sign performance metrics on the held-out test split at different IoU thresholds. Precision, recall, and F1 are reported as fractions.

IoU	**TP**	**FP**	**FN**	**Precision**	**Recall**	**F1**
0.0	542	5	10	0.99	0.98	0.99
0.1	542	5	10	0.99	0.98	0.99
0.2	542	5	10	0.99	0.98	0.99
0.3	541	6	11	0.99	0.98	0.98
0.4	541	6	11	0.99	0.98	0.98
0.5	537	10	15	0.98	0.97	0.98
0.6	511	36	41	0.93	0.93	0.93
0.7	416	131	136	0.76	0.75	0.76
0.8	225	322	327	0.41	0.41	0.41
0.9	51	496	501	0.09	0.09	0.09
1.0	0	547	552	0.00	0.00	0.00

A critical observation is the model's performance at an IoU threshold of 0.5, a standard benchmark in object detection tasks. The model exhibits a com-

mendable F1 score of 0.98, indicating its proficiency in accurately localising and classifying warning signs even under more stringent criteria for a correct detection.

However, the model's performance declines notably as the IoU threshold increases beyond 0.5. At an IoU of 0.7, the F1 score drops to 0.76, and it further decreases to 0.41 at an IoU of 0.8. This trend illustrates the model's challenge in maintaining high precision and recall as the criteria for a correct detection become more rigorous, requiring a closer alignment between the predicted and ground-truth bounding boxes.

The most stringent scenario, an IoU of 1.0, sees the model unable to correctly identify any warning signs, as indicated by the F1 score of 0.00. This result is not unexpected, given the practical challenges of achieving a perfect overlap between the predicted and actual bounding boxes.

In conclusion, the DETR model demonstrates robust performance in detecting warning signs, particularly at lower IoU thresholds, where the emphasis is on detection sensitivity. The model's ability to maintain high accuracy at an IoU of 0.5 is particularly noteworthy. Nonetheless, the diminishing performance at higher IoU thresholds suggests potential avenues for enhancing the precision of the model's bounding box predictions, thereby improving its overall efficacy in practical applications. Figure 3.7 visualises the same trend as an F1-versus-IoU curve, while Figure 3.8 provides qualitative examples of warning-sign detections on the held-out test split.

3.5.4 Further Analysis

Given that the DETR model excels in the detection of utility poles and demonstrates strong performance in identifying warning signs, this section delves further into the test results. It meticulously examines the influence of pole sizes on the quality of predictions, and formulates recommendations for the application of this

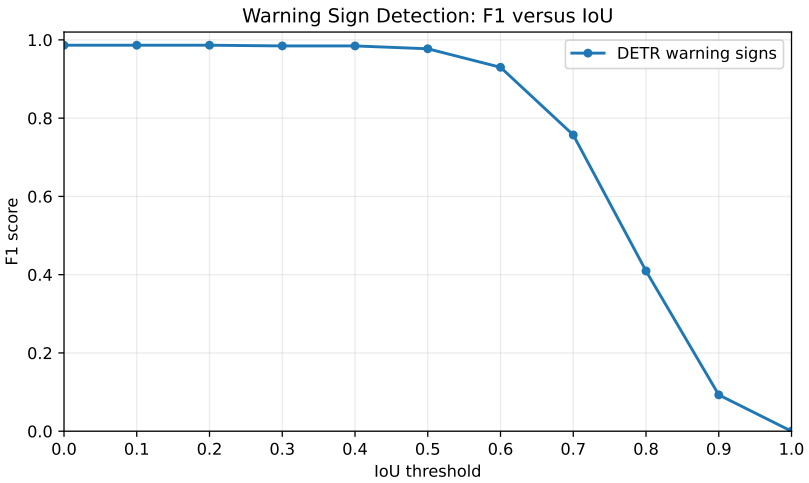


Figure 3.7: F1 score as a function of IoU threshold for DETR warning-sign detection on the held-out test split.

model in real-world scenarios.

This section is intended as a complementary diagnostic analysis rather than a repeat of the preceding benchmark tables. The earlier tables report headline mAP, precision, recall, and F1 values, whereas the following figures and tables decompose the same held-out DETR predictions by confidence threshold, IoU threshold, object size, false-positive category, and false-negative distribution. Retaining these views helps explain why the headline scores behave as they do and supports the later deployment recommendations.

The optimal confidence threshold produces a curve with the greatest Area Under the Curve (AUC). As a result, with an AUC of 36.68, the optimal confidence threshold for the class of utility poles is 0.9. The warning sign class has a maximum AUC of 38.02 and occurs at two confidence levels, 0.7 and 0.8. Since 0.8 is the larger confidence value, it is the confidence threshold for warning signs. For a confidence value of 0.8, the utility pole class produced an AUC of 36.45. This is a small difference of 0.23 for a confidence threshold of 0.9. Therefore, both the utility pole and the warning sign classes have a confidence threshold set to 0.8, without a significant impact on performance.

Figure 3.8: Representative DETR warning-sign detection outcomes on the held-out test split, including a correct detection, a low-IoU detection, a false positive, and a false negative. Green boxes show ground truth and red boxes show predictions.

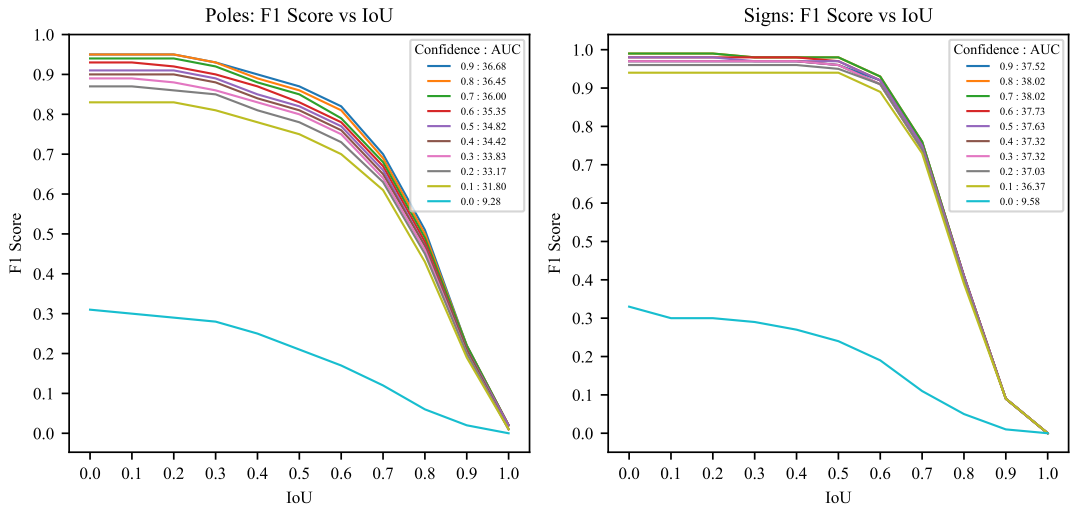


Figure 3.9: Pole and Signs F1 Score.

Utility Pole Detection Analysis

As the IoU threshold increases, the predicted bounding boxes must align more precisely with the ground-truth bounding boxes, in order to be considered a true positive. As a result, increasing the IoU threshold decreases the number of true positives. Conversely, increasing the IoU threshold increases the number of false positives and negatives. Figure 3.10 illustrate this behaviour for the utility pole class.

The performance of the model in detecting utility poles is detailed in Table 3.20, with a particular emphasis on True Positive detections. In this table, the percentage row reports the number of TPs as a percentage of the 1,490 ground-truth utility poles in the test set. The model exhibits a remarkable detection rate, identifying 96.17% of the poles in the test set at an IoU value of 0.0. This high level of detection accuracy gradually declines as the IoU threshold is increased, a common trend in object detection tasks, where higher IoU values denote more stringent criteria for classifying a detection as a True Positive.

Notably, even at an IoU threshold of 0.4, the model successfully detects 90.20%

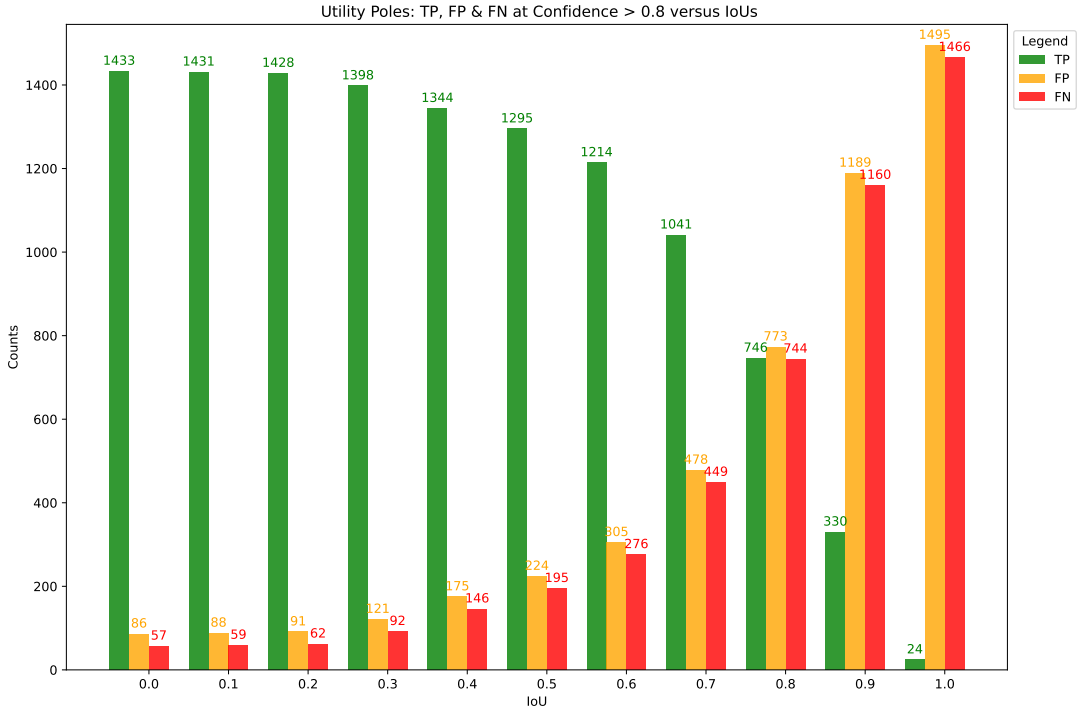


Figure 3.10: Poles: TP, FP and FN at Confidence 0.8 versus IoUs.

of the poles, implying that the predictions are generally in good alignment with the actual locations of the poles. The detection rate remains robust, staying above 80% up to an IoU of 0.6, which is indicative of the model's effective performance in accurately locating poles with a reasonable level of precision.

However, there is a significant decline in the detection rate as the IoU threshold further increases. At an IoU of 0.7, the model detects fewer than 70% of the poles, and this percentage sharply falls to just over 50% at an IoU of 0.8. The performance at the highest IoU values of 0.9 and 1.0 is markedly lower, with only 22.15% and 1.61% of poles detected, respectively. These figures suggest that while the model is highly proficient at detecting poles with moderate spatial accuracy, its capability to precisely match the exact location, as required at higher IoU levels, is somewhat limited.

Further analysis of the test data, using the standard correlation coefficient,

Table 3.20: Utility-pole true-positive counts by IoU threshold.

IoU	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
TPs	1,433	1,431	1,428	1,398	1,344	1,295	1,214	1,041	746	330	24
%	96.17	96.04	95.84	93.83	90.20	86.91	81.48	69.87	50.07	22.15	1.61

revealed notable linear relationships between the IoU values and the physical dimensions of the poles, as detailed in Table 3.21.

Table 3.21: Pole IoU Correlation Coefficients.

Metric	**Correlation**
GT Widths	0.66
GT Area	0.51
GT Heights	0.50

The correlation coefficients provide insightful observations. Pole widths, with a correlation coefficient of 0.66, exhibit the strongest impact on IoU values. This suggests that wider poles are more likely to be accurately detected and aligned with the ground-truth data. Following this, pole areas and heights also show positive correlations with IoU values, with coefficients of 0.51 and 0.50 respectively. These findings align with the expectation that larger poles, offering more distinct and discernible features, are easier to detect and assess accurately.

Figure 3.11 visually demonstrates these relationships. It illustrates the sizes of each detected pole, with emphasis on their respective IoU values. The figure corroborates the findings presented in Table 3.21, vividly highlighting the positive correlation between the size of the poles and their IoU values. The visual evidence thus reinforces the conclusion that larger poles, due to their more prominent features, are associated with higher IoU values in the model's detections.

Figure 3.12 shows the distribution of pole sizes for the true positives, grouped by their IoU value. As shown, both the IoU value and the number of detected poles increase as the pole width increases. Likewise, both the IoU value and the

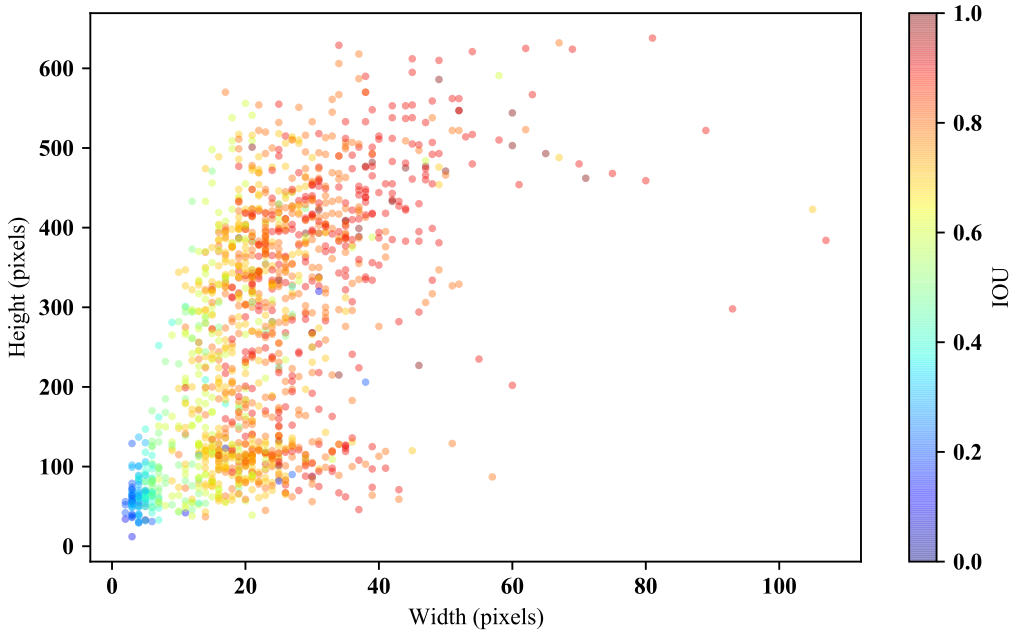


Figure 3.11: True Positives against Pole Widths and Heights.

number of detected poles increase as the pole height increases, but to a lesser degree.

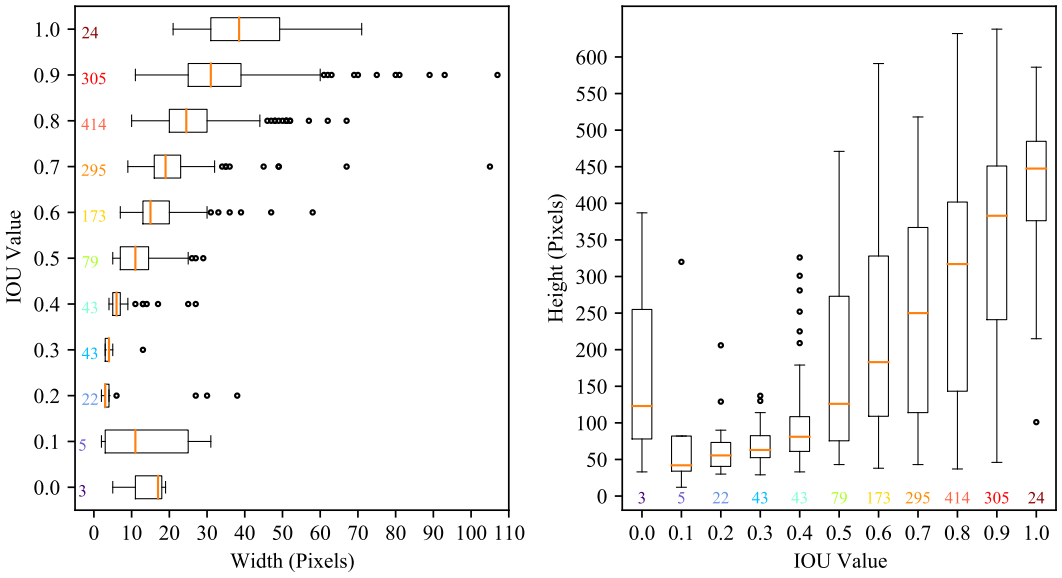


Figure 3.12: Distribution of Pole Sizes per IoU Value.

The FPs identified by the model across various IoU values are detailed in Table 3.22. In this table, the percentage row reports the FP count as a percentage of the total number of predictions considered at that threshold, not as a percentage of the ground-truth pole count. This analysis provides a nuanced understanding of the model's precision in detecting utility poles.

Table 3.22: Utility-pole false-positive counts by IoU threshold.

IoU	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
FPs	86	88	91	121	175	224	305	478	773	1,189	1,495
%	5.66	5.79	5.99	7.97	11.52	14.75	20.08	31.47	50.89	78.27	98.42

The data reveals that the proportion of false positives varies notably with the IoU threshold. At lower IoU values (0.0 to 0.2), the FP rate is relatively modest, staying below 6%. This suggests that the model has a lower tendency to incorrectly identify objects as utility poles under less stringent detection criteria.

However, as the IoU value increases, there is a significant rise in the FP rate. Notably, at an IoU of 0.7, the FP rate exceeds 30%, and it escalates further at higher IoU values, reaching nearly 51% at 0.8 and escalating to 98.42% at the maximum IoU of 1.0. This steep increase indicates a substantial number of false detections at higher precision levels, suggesting that the model might be overly sensitive or prone to misidentifying objects as poles under strict alignment conditions.

With only 5.77% of the predictions identified as false positives, these have been categorised into five distinct groups, as outlined in Table 3.23. Figure 3.13 provides visual examples of each category.

Moreover, the false positives attributed to poles, totalling 50, are further broken down into three subcategories as presented in Table 3.24. Illustrative examples of these subcategories are depicted in Figure 3.14.

In the case of the 33 false positives classified as 'Partially Detected Pole', only

Table 3.23: False Positive Categories.

Category	Count	%
Poles	50	58
Street Lights	18	21
Road Signs	5	6
Chimneys	3	3
Other*	10	12

*Other includes 1 of each: advertising sign, building, flag mast, plants, roof pole, tree, and unknown.

Figure 3.13: Examples of False Positive Categories.

Table 3.24: False Positive Poles Breakdown.

Category	Count
Partially Detected Pole	33
Poor IoU	14
Disjointed Pole as Single Pole	3

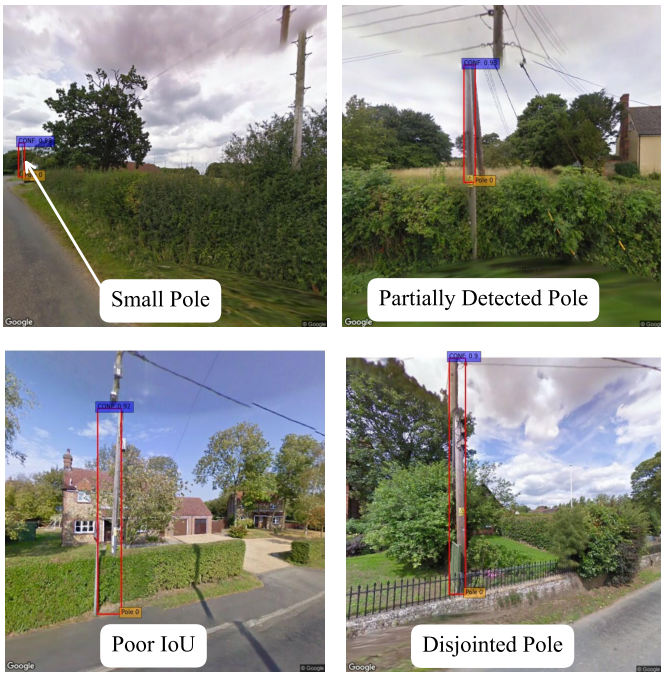


Figure 3.14: False Positive Poles Examples.

a portion of a pole was detected. For the 14 instances categorised under 'Poor IoU', the Intersection over Union value fell below the set threshold. Finally, three cases were identified as 'Disjointed Pole as Single Pole', where the image-stitching process in Google Street View erroneously fragmented a single pole into multiple parts.

Warning Sign Detection Analysis

Similarly, Figure 3.15 illustrate the behaviour of IoU thresholds on warning sign detection. Intuitively, the larger the sign, the easier it should be to detect. Similarly, the smaller the sign, the harder it should be to detect. The next section investigates this idea by analysing the true positives and their relationship to the sign sizes.

The analysis of True Positives in relation to IoU values is a critical metric for understanding the performance of object detection models. In Table 3.25, the

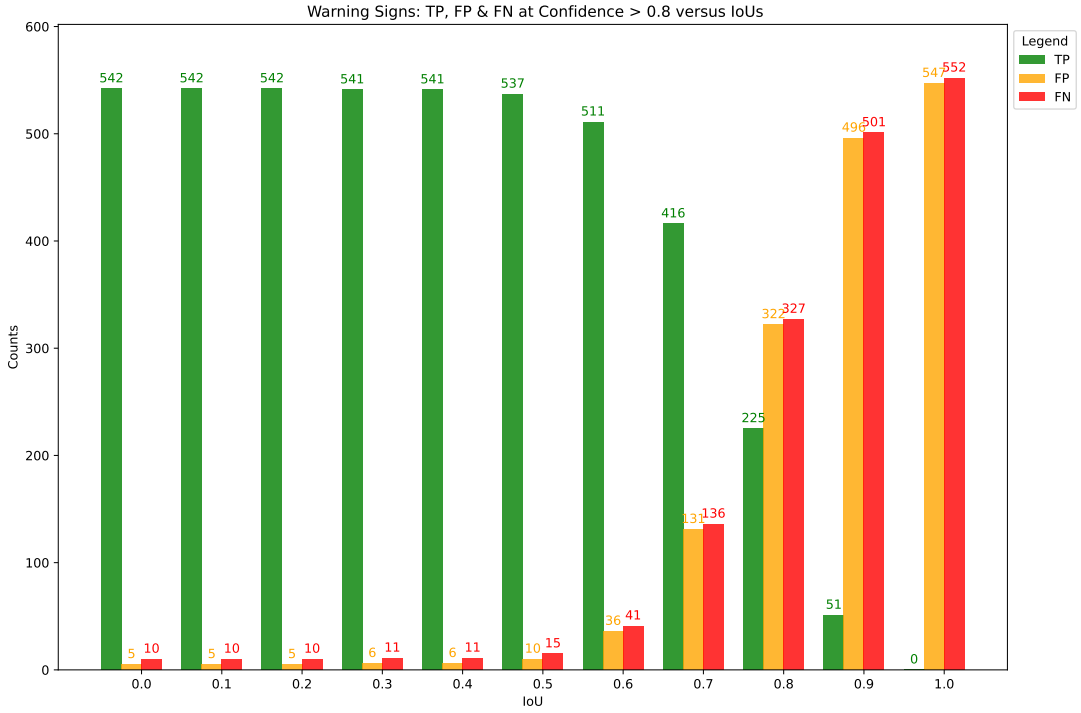


Figure 3.15: Signs: TP, FP and FN at Confidence 0.8 versus IoUs.

percentage row reports the TP count as a percentage of the 552 ground-truth warning signs in the test set. The provided data, as shown in Table 3.25, reflects the TP count and its corresponding percentage out of a total of 552 ground truths for various IoU thresholds.

At lower IoU thresholds (0.0 to 0.4), the model exhibits a remarkably high TP rate, maintaining a consistent count of 542 TPs, equating to 98.19% of the ground truths. This indicates a strong capability of the model to detect objects with a broad acceptance margin. As the IoU threshold increases to 0.5, a slight decrease in TPs is observed (537 TPs, 97.28%), suggesting a minor reduction in model performance when stricter spatial overlap criteria are applied.

A more significant reduction in TPs is evident as the IoU threshold reaches 0.6 and above. The TP count drops to 511 (92.57%) at 0.6 and continues to decrease sharply, with only 416 TPs (75.36%) at 0.7 and 225 TPs (40.76%) at 0.8.

This trend highlights the model's decreasing effectiveness in accurately localising objects under stringent overlap conditions.

At extremely high IoU values of 0.9 and 1.0, the TP counts drop drastically to 51 (9.24%) and 0 (0%) respectively. These results indicate a severe limitation in the model's ability to produce highly precise bounding boxes that almost perfectly or completely match the ground truth. The decrease from 98.19% at IoU 0.0 to 0% at IoU 1.0 is therefore not a contradiction in the results. At IoU 0.0, a correctly classified prediction can be matched to an unmatched ground-truth warning sign even without requiring spatial overlap, so the value mainly reflects object-presence detection. At IoU 1.0, the predicted and ground-truth boxes would have to coincide exactly in position and size, which is an unrealistically strict requirement for small signs because even a one- or two-pixel boundary difference reduces the IoU below 1.0.

Table 3.25: Warning-sign true-positive counts by IoU threshold.

IoU	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
TPs	542	542	542	541	541	537	511	416	225	51	0
%	98.19	98.19	98.19	98.01	98.01	97.28	92.57	75.36	40.76	9.24	0

Furthermore, the correlation coefficients presented in Table 3.26 offer additional insights into the relationship between the ground truth (GT) dimensions and the IoU performance. The correlation values for GT Width (0.38), GT Height (0.37), and GT Area (0.36) are moderately positive. This implies a modest but significant relationship between the size of the ground truth bounding boxes and the IoU scores achieved. Larger bounding boxes, as indicated by width, height, and overall area, tend to have a slightly higher IoU score, suggesting that the model may perform better with larger objects. This correlation, while not strong, is noteworthy and can inform adjustments to the model or its deployment strategy, particularly in environments where object size variability is a key factor.

Table 3.26: Sign IoU Correlation Coefficients.

Metric	Correlation
GT Width	0.38
GT Height	0.37
GT Area	0.36

Figure 3.16 shows each detected sign by its IoU value against the height and width of the detected sign and verifies the results of Table 3.26.

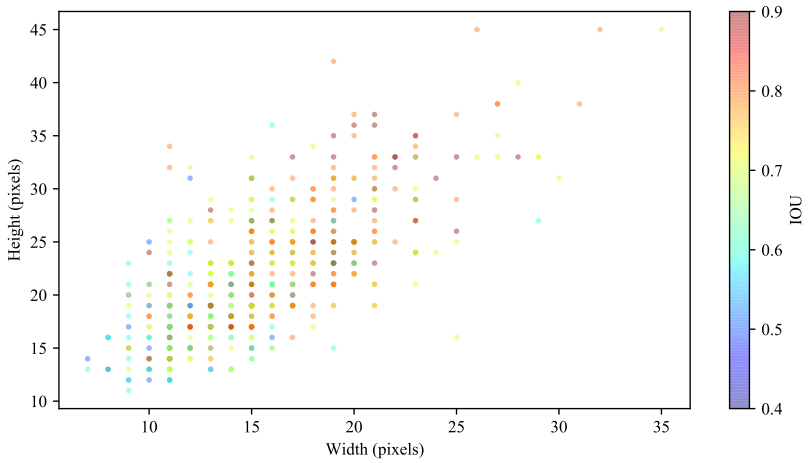


Figure 3.16: True Positives against Sign Widths and Heights.

Figure 3.17, shows the distribution of sign heights and widths for the true positives, grouped by their IoU value. As expected, the larger the sign sizes, the greater the IoU value. Similarly, there are more detections in IoU groups that have larger sign sizes.

The analysis of False Positives in relation to various IoU thresholds provides essential insights into the specificity of an object detection model. Table 3.27 details the FP count and its corresponding percentage out of a total of 547 sign predictions at different IoU thresholds.

At the lowest IoU thresholds (0.0 to 0.4), the model demonstrates a low FP rate, with counts hovering at 5 and 6, translating to 0.91% and 1.10%, respec-

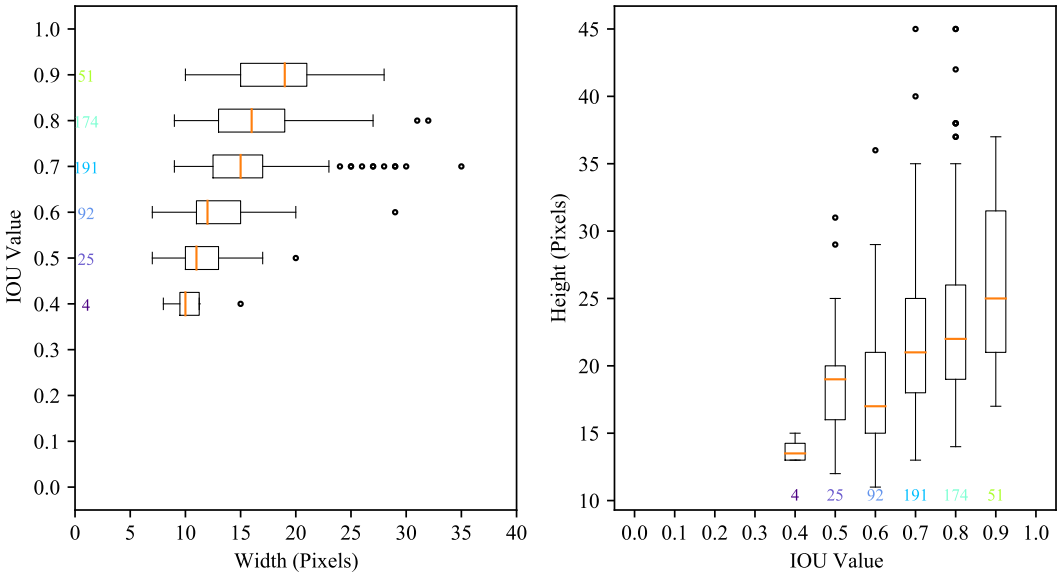


Figure 3.17: Distribution of Sign Sizes per IoU Value.

tively. This indicates a high specificity, where the model is less prone to falsely identifying objects as signs at these IoU thresholds. However, it is important to note that lower IoU thresholds are generally more lenient, potentially allowing for less precise object localisation to be deemed correct.

A noticeable increase in FPs is observed at the 0.5 IoU threshold, where the count rises to 10 (1.83%). This increment signifies the model's decreasing specificity as the IoU criterion for object detection becomes more stringent.

A substantial increase in the FP rate is evident as the IoU threshold reaches 0.6 and above. The FP count rises sharply to 36 (6.58%) at 0.6, followed by a more significant escalation to 131 (23.95%) at 0.7 and 322 (58.87%) at 0.8. These trends highlight a notable decline in the model's specificity, with a substantial proportion of detections being falsely identified as signs under stricter spatial overlap conditions.

At extremely high IoU values of 0.9 and 1.0, the FP counts escalate dramatically to 496 (90.68%) and 547 (100%), respectively. It is essential to recognise

that achieving near-perfect or perfect overlap with ground truth is exceedingly challenging for any object detection model. These high IoU thresholds are often unrealistic in practical applications. Therefore, while the FP rates are high at these thresholds, it does not necessarily indicate a deficiency in the model. Rather, it underscores the inherent challenge in achieving extremely high spatial accuracy in object detection tasks.

In conclusion, the model exhibits strong specificity at lower IoU thresholds, with an expected reduction in accuracy as the IoU criterion becomes more stringent. The high FP rates at higher IoU values reflect the rigorous and often impractical standards set by such thresholds rather than the model's inadequacy. This analysis suggests that the model performs well within the reasonable expectations of object detection tasks.

Table 3.27: Warning-sign false-positive counts by IoU threshold.

IoU	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
FPs	5	5	5	6	6	10	36	131	322	496	547
%	0.91	0.91	0.91	1.10	1.10	1.83	6.58	23.95	58.87	90.68	100.00

Only 0.9% of the sign predictions made by the model were false positives. These are grouped into the three categories shown in Table 3.28, Figure 3.18 displays each category.

Table 3.28: Signs False Positive Categories.

Category	**Count**
Possible Sign	3
Yellow Pole Box	1
Yellow Square	1

- Three false positives were yellow squares. The low resolution of the image made it difficult to determine if they were warning signs.

- One false positive was a small yellow box attached to the pole.
- The remaining false positive was a yellow square on the pole.

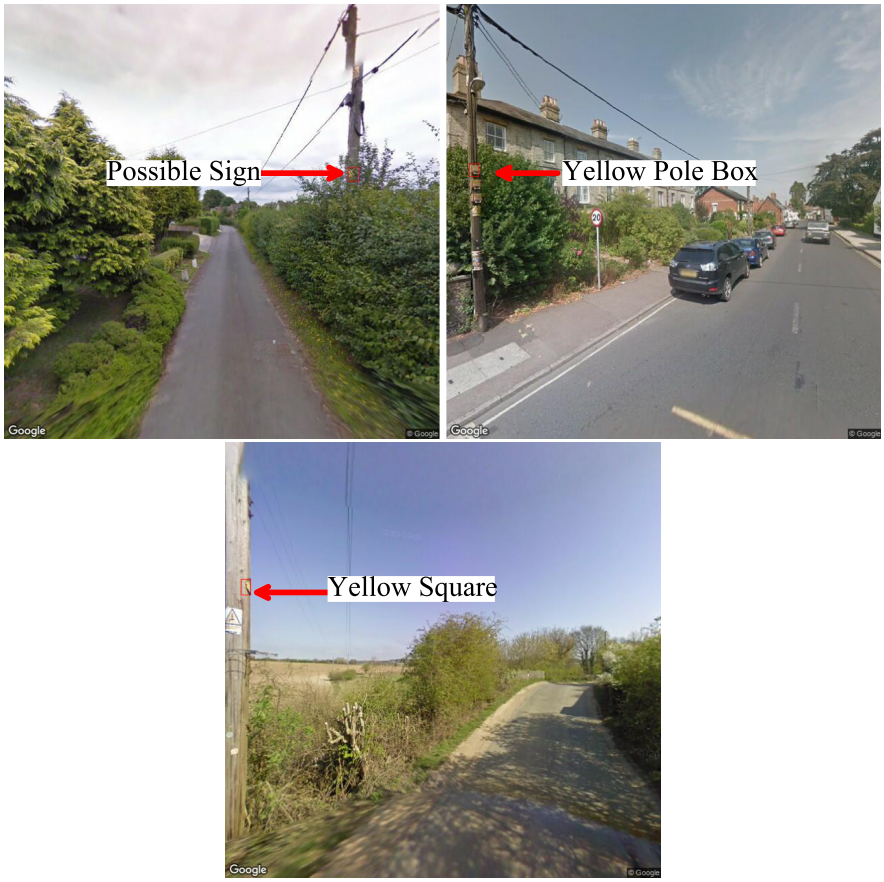


Figure 3.18: Examples of Sign False Positive Categories.

An analysis of False Negatives in relation to various IoU thresholds is crucial for understanding the sensitivity of an object detection model. The data presented in Table 3.29 illustrates the count of False Negatives and their corresponding percentages out of a total of 552 ground-truth signs, across different IoU thresholds.

At the lower end of the IoU spectrum (0.0 to 0.4), the FN count remains relatively stable, with only 10 to 11 instances, constituting about 1.81% to 1.99% of the total ground truths. This low rate of FNs indicates that the model has a

high detection sensitivity, successfully identifying most of the true objects even when the spatial overlap criterion is lenient.

As the IoU threshold increases to 0.5, there is a slight increase in FNs to 15, which is 2.72% of the ground truths. This suggests that as the spatial accuracy requirement becomes moderately stringent, the model starts to miss a few more objects, but still maintains a relatively high detection sensitivity.

A notable increase in FNs is observed as the IoU threshold reaches 0.6 and beyond. The FN count escalates to 41 (7.43%) at an IoU of 0.6, indicating a decrease in the model's sensitivity. This trend becomes more pronounced at an IoU of 0.7, where the FNs increase significantly to 136, accounting for 24.64% of the ground truths. The model's sensitivity is substantially reduced at this threshold, failing to detect a considerable number of true objects.

At higher IoU thresholds of 0.8, 0.9, and 1.0, the model exhibits a substantial decline in sensitivity. The FN counts rise dramatically to 327 (59.24%), 501 (90.76%), and 552 (100%) respectively. It is important to acknowledge that achieving high precision in object detection, especially at these thresholds, is inherently challenging. Particularly at an IoU of 1.0, where perfect overlap is theoretically required, it is unrealistic to expect any model to align perfectly with all ground truth instances. Therefore, the high FN rate at this threshold should not be seen as a failure of the model but rather as an indication of the extremely stringent and often impractical standards set by such high IoU values.

In conclusion, the model demonstrates high sensitivity at lower IoU thresholds, efficiently detecting most of the true objects. However, as the IoU criterion becomes more demanding, a reduction in the model's ability to detect objects accurately is observed. This trend, particularly at higher IoU values, underscores the limitations inherent in the pursuit of perfect spatial accuracy in object detection tasks.

These false negatives had a sign size distribution shown in Figure 3.19. The

Table 3.29: Warning-sign false-negative counts by IoU threshold.

IoU	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
FNs	10	10	10	11	11	15	41	136	327	501	552
%	1.81	1.81	1.81	1.99	1.99	2.72	7.43	24.64	59.24	90.76	100.00

average missed sign had a width of 13 pixels and a height of 18.5 pixels (ignoring outliers). Seventy-five percent of the missed signs had a width below 16 pixels and a height below 22.75 pixels.

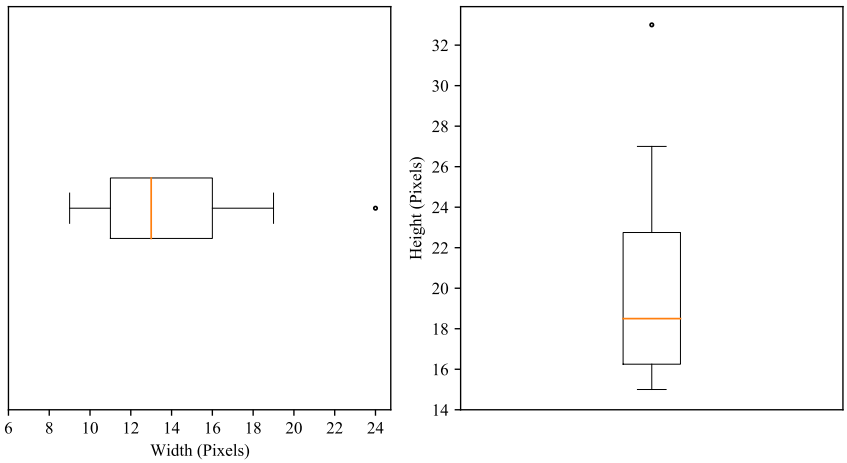


Figure 3.19: Boxplot for False Negative Sign Sizes.

Figure 3.20 shows the false negatives, plotted on top of the heat map of the true positives. This distribution of ground-truth signs highlights both the missed and detected signs. The false negatives are marked with high-contrast red crosses to make the missed cases visible against the heat-map background.

Unlike the poles where low IoU values cluster together, there is no clustering of IoU values for the signs. Because of this, there are no further recommendations for pole sizes.

Model predictions are more accurate when the pole sizes have widths greater than 8 pixels and heights greater than 35 pixels. The thresholds in Figure 3.21 removed one correct sign detection; error counts did not change.

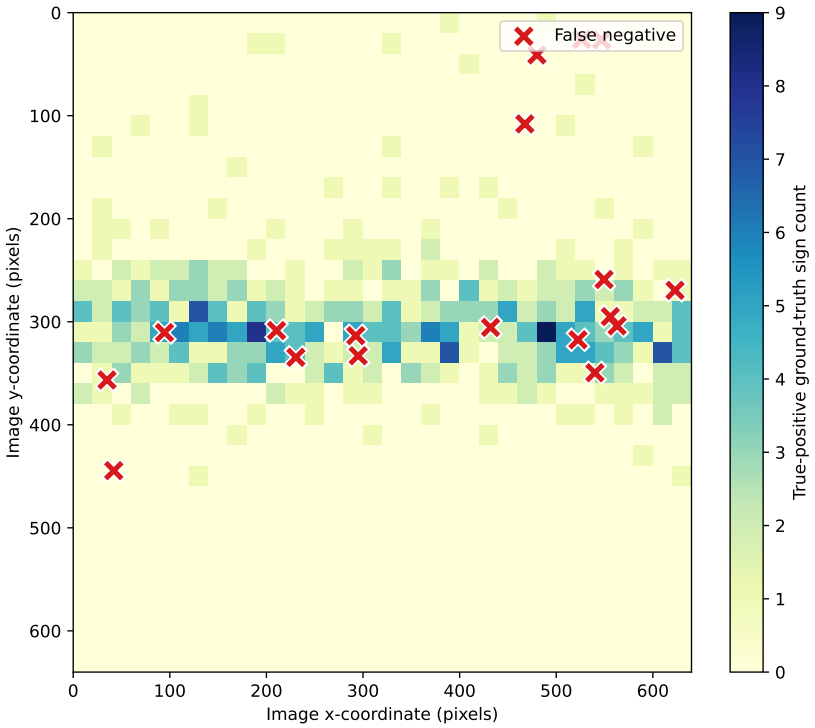


Figure 3.20: Signs ground-truth heat map with false negatives marked using high-contrast red crosses.

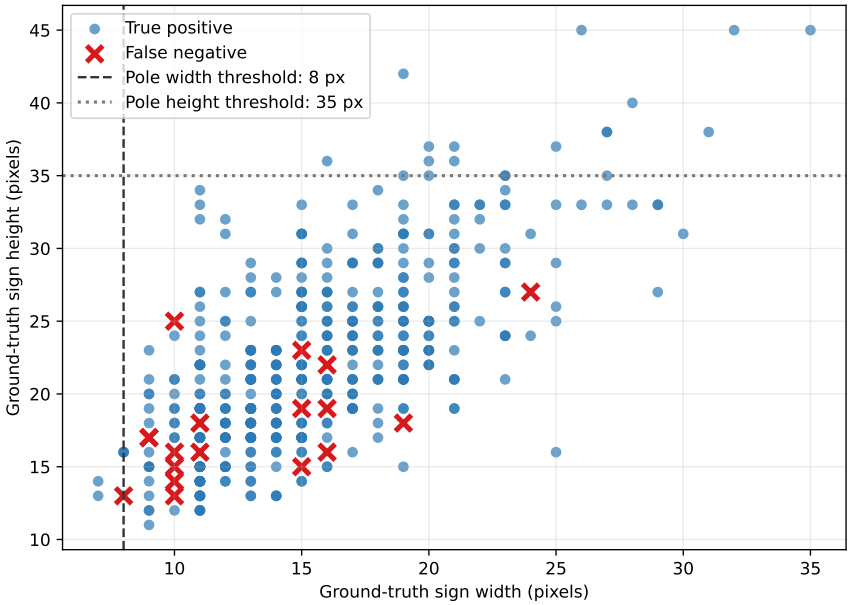


Figure 3.21: Pole size threshold on signs with false negatives marked using high-contrast red crosses.

3.6 Summary

This chapter has presented a comprehensive study on the application of deep learning models for the detection and assessment of wooden utility poles and attached warning signs. The research involved the creation of a large and meticulously annotated dataset, leveraging over 670,000 geographic coordinates provided by UKPN, and the utilisation of images extracted from Google Street View. A rigorous benchmarking of existing object detection models—RetinaNet, Faster R-CNN, and YOLO v3-Tiny—was conducted.

The findings from the benchmarking phase revealed that while traditional models exhibited varying degrees of success, they faced challenges in accurately detecting utility poles, particularly in the absence of distinctive features like crossarms. RetinaNet achieved a mAP of 23.00%, and Faster R-CNN reached an mAP of 63.82%, both demonstrating limitations in handling the complex-

ity of the dataset. YOLO v3-Tiny performed better, with an mAP of 77.00%, but still fell short in achieving the high levels of precision required for reliable automated inspections.

The introduction of the DETR model marked a significant advancement in this domain. By integrating convolutional neural networks with transformer architectures, DETR was able to capture both local and global features of the images, enabling it to more effectively distinguish utility poles from visually similar objects like trees or lampposts. The DETR model achieved an impressive mAP of 90.43% for utility pole detection and 88.26% for warning sign detection, outperforming the other models by a substantial margin.

This benchmark should be read as an applied comparison on the thesis dataset: the DETR pipeline was tuned for OHL-UK, while the legacy detectors were reproduced as strong literature baselines rather than exhaustively re-optimised variants.

The superior performance of the DETR model can be attributed to its end-to-end training approach and the use of the Hungarian algorithm for direct set prediction without the need for anchor boxes or non-maximum suppression. This architecture allowed DETR to handle the complex visual environments presented in the dataset, demonstrating robustness in detecting objects of varying sizes and orientations, even in images of lower quality from Google Street View.

Further analysis highlighted the correlation between the size of the utility poles and the detection accuracy, with larger poles being detected more reliably. The study also identified and categorised false positives and negatives, providing insights into areas where the model could be further improved.

In conclusion, this research demonstrates that the DETR model improves automated utility pole and warning sign detection in challenging Google Street View imagery. Its application contributes to computer vision practice and offers the utility sector a scalable route for infrastructure inspection and maintenance.

The methods and findings provide a foundation for future work on more difficult scenes, additional infrastructure classes, and real-time deployment.

Chapter 4

Lean Angle Estimation of Utility Poles

4.1 Introduction

Utility pole detection and segmentation have become critical tasks in many industrial applications, particularly for infrastructure monitoring and maintenance. Accurate detection of utility poles is essential for ensuring structural integrity, conducting safety assessments, and planning interventions. Segmentation, the process of delineating the exact boundaries of objects, plays a key role in providing more detailed and precise analyses compared to simple detection tasks. Unlike bounding box detection, segmentation provides pixel-level information, enabling advanced spatial analyses of objects.

Semantic segmentation methods have seen significant advancements with the introduction of FCNs [106], which revolutionised the field by enabling dense, pixel-wise predictions. FCNs laid the foundation for subsequent segmentation approaches by demonstrating that convolutional architectures could be effectively adapted to produce segmentation masks rather than just classification or bounding boxes. In this context, modern detection frameworks such as DETR (DEtec-

tion TRansformer) [68] extend their capabilities to include segmentation without requiring architectural modifications. By leveraging the inherent strength of transformers in object detection, DETR generates segmentation masks alongside bounding boxes in a unified, end-to-end manner.

This work focuses on applying DETR for the segmentation of wooden utility poles and their attached warning signs. The pixel-level masks produced by DETR provide a detailed spatial representation of the poles' boundaries, which is particularly valuable for downstream tasks such as lean angle estimation. Lean angle estimation is an important metric in assessing the physical condition of utility poles, offering insights into potential structural issues caused by factors such as wind-related disasters [107]. The angular deflection of utility poles is a recognised indicator of structural resilience and can help power distribution companies identify poles at risk of failure and plan corrective measures effectively.

Compared to the state-of-the-art approaches reviewed in the literature, the proposed method using DETR for segmentation offers several key advantages. Traditional methods such as those presented by Zhang et al. [14] and Liu et al. [16] primarily rely on bounding box detection followed by localisation or focused component analysis. While effective for coarse localisation or detection of smaller pole components, these methods do not provide the detailed boundary information necessary for precise lean angle estimation. Other approaches, such as the UAV-based segmentation technique by Zhu et al. [10], rely on classical segmentation models like SegNet, coupled with post-processing using the Hough Transform to calculate lean angles. However, the use of DETR removes the need for such multi-stage pipelines by unifying detection and segmentation in a single framework, reducing complexity and computational overhead.

Furthermore, vehicle-based methods such as Park et al. [15], which combine stereo vision and triangulation for 3D pole localisation, are hardware-dependent and require specialised equipment. In contrast, the DETR-based segmentation

approach presented here relies solely on high-resolution 2D images, making it more adaptable for practical deployment in various settings without the need for additional hardware infrastructure.

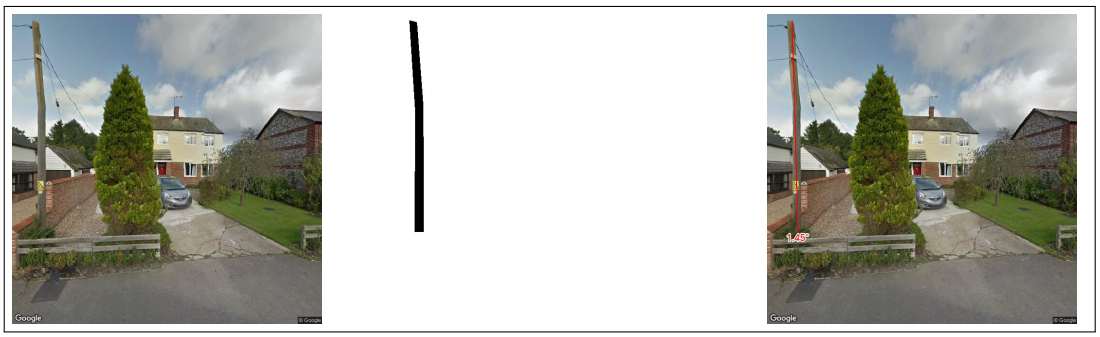
The evaluation of DETR's segmentation performance is conducted using the specialised dataset introduced in Section 3.2. By utilising segmentation annotations, DETR is fine-tuned to produce high-quality masks of utility poles and attached warning signs. The results demonstrate that the proposed approach achieves a higher level of detail and accuracy in segmenting utility poles compared to bounding box-based methods and conventional multi-stage pipelines. This capability enables robust and precise lean angle estimation, contributing to improved structural assessments and maintenance planning for utility poles.

4.2 Dataset Preparation

The dataset utilised to advance the DETR model's capabilities for both detection and segmentation of wooden utility poles and their warning signs is an extension of the dataset described in Chapter 3.2. The core structure, including the geographic coverage, data splits, and five-fold configuration, remains unchanged. The primary enhancement is the addition of segmentation annotations, which provide polygonal boundaries for the objects. These additional annotations enable a more detailed analysis, facilitating precise segmentation and the subsequent computation of ground-truth lean angles for structural assessment of the utility poles.

The ground-truth lean angle is defined as the absolute deviation of the annotated pole centreline from the vertical direction in the two-dimensional image plane. It is estimated using a two-stage process, as illustrated in Figure 4.1. Segmentation points are first converted into a polygonal boundary, after which a line of best fit is derived using the OpenCV fit line method. The lean angle is

then calculated from the fitted line orientation using trigonometric principles and reported in degrees. This definition measures apparent image-plane lean only; it does not measure three-dimensional lean towards or away from the camera. No assumption is made that the Google Street View camera is perfectly horizontal to the ground. Instead, image vertical is used as a practical proxy for gravity, meaning that camera roll, road camber, terrain slope, and viewpoint can introduce image-plane angle error.



(a) Original Image. (b) Convert Segmentation to Polygon. (c) Fit Line and Estimate Angle.

Figure 4.1: Ground-Truth Lean Angle Estimation Process.

Segmentation annotations are captured within the JSON files as an array of coordinates, detailing the polygonal boundary of the objects. The structure of these annotations is expressed in the 'segmentation' field with coordinates paired sequentially as x and y values: `[[x1, y1, x2, y2, ..., xn, yn]]`. These pairs, when connected, form a polygon that accurately outlines the shape of the utility poles, allowing for precise segmentation.

These segmentation coordinates fulfil a dual purpose. Firstly, they enable the DETR model to achieve detailed segmentation of the utility poles. Secondly, they are integral to calculating the ground truth lean angle of the poles. The ground truth angles are derived through a two-stage process, initially utilising the fit line method from the OpenCV library to estimate a line of best fit through the

segmentation points. This method performs a regression analysis, optimising the line placement to minimise the distances of the points to the line.

The lean angle is computed using trigonometry, specifically the arctangent of the line's gradient, which is given by the formula:

$$\theta = \arctan\left(\frac{\text{rise}}{\text{run}}\right) \tag{4.1}$$

where "rise" represents the change in the y-coordinate and "run" indicates the change in the x-coordinate of the line vector. The 'angle' field in the JSON entry records this calculated lean angle, providing an essential metric for the poles' structural analysis.

In this thesis, θ is treated as an image-plane apparent lean angle measured in degrees relative to the vertical image axis. The fitted line orientation is converted to the acute deviation from vertical, so the recorded ground-truth angle lies in the range $0^\circ \leq \theta \leq 90^\circ$, where 0° corresponds to a vertical pole in the image and larger values indicate stronger apparent lean. Prediction error is reported as absolute angular error against the annotated ground truth:

$$\theta_{\text{Error}} = \left|\theta_{\text{predicted}} - \theta_{\text{ground truth}}\right|. \tag{4.2}$$

For instance, the JSON entry for an object is:


```
{
  "image_id": 917,
  "id": 1484,
  "category_id": 0,
  "segmentation": [[475.0, 0.0, ..., 476.0, 0.0]],
  "bbox": [475.0, 0.0, 14.0, 90.0],
  "iscrowd": 0,
  "angle": 0.67
```

```
}
```

In this entry, the ‘angle‘ is determined to be 0.67 degrees. This meticulous calculation allows for an intricate examination of the poles, beyond simple detection and segmentation, laying the groundwork for in-depth structural analysis.

4.3 Segmentation Model

To incorporate segmentation capabilities into the DETR model devised in Chapter 3, attention is placed on the additional layers and training adaptations necessary to segment wooden utility poles. The following explanation builds upon the technical specifics and advancements presented for DETR.

Architectural Extensions The DETR model, initially developed for object detection in Chapter 3, is augmented with a segmentation head to facilitate instance segmentation tasks. Inspired by the mask prediction branch in Mask R-CNN [84], this segmentation head is implemented as a small fully convolutional network (FCN) appended to the last layer of the transformer's decoder [68].

The segmentation head processes the features output by the transformer, utilising the learned positional encodings and attention mechanisms inherent to the transformer model.As illustrated in Figure 4.2, the decoder produces a set of per-object box embeddings that, via multi-head attention over the encoded image features, give rise to attention maps; these attention-weighted features are then combined with multi-scale ResNet features and refined by a Feature Pyramid Network (FPN)-style convolutional network to generate the final mask logits [13]. The FPN, originally proposed to handle multi-scale feature extraction in object detection tasks, processes features at different resolutions and combines them to preserve fine-grained spatial details while maintaining semantic richness. This refined feature representation is then used to output pixel-wise mask logits. A

pixel-wise argmax operation is applied to generate the final binary masks for each detected object, indicating the presence or absence of the object at each pixel.

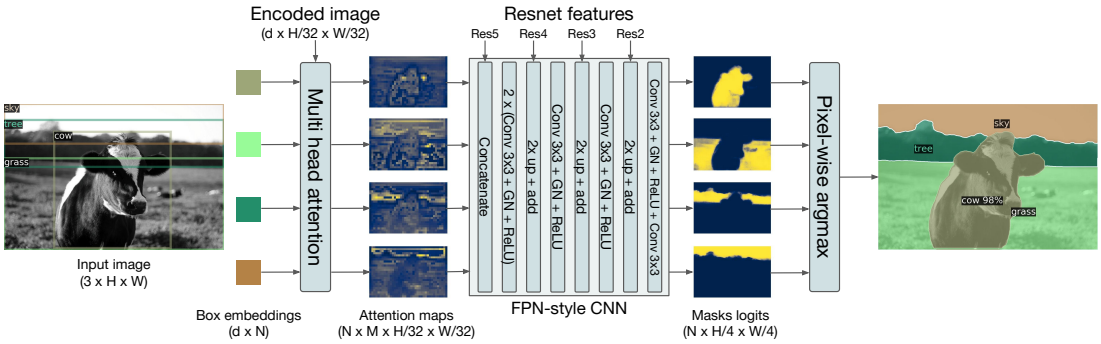


Figure 4.2: Illustration of the Panoptic Head [68].

Loss Functions for Segmentation To supervise the predicted masks against the ground truth masks, DETR employs a combination of losses, which are defined as follows:

- **Hungarian Loss:** The Hungarian loss is primarily used for object detection and includes:

  1. Negative log-likelihood loss for class predictions.
  2. A linear combination of the L1 loss and the Generalised Intersection over Union (GIoU) loss for bounding box regression.

  See Section 2.5.2 for more details on the Hungarian Algorithm.

- **Dice Loss (F-1 Loss):** First introduced in the context of volumetric medical image segmentation [108], the Dice loss is a differentiable approximation of the Dice coefficient, defined as:

$$\mathcal{L}_{\text{DICE}}(m, \hat{m}) = 1 - \frac{2m\sigma(\hat{m}) + 1}{\sigma(\hat{m}) + m + 1}, \tag{4.3}$$

  where:

- m represents the ground truth mask.
- $\hat{m}$ is the predicted logits.
- σ denotes the sigmoid activation function.

The addition of 1 in both the numerator and denominator ensures numerical stability. This formulation directly optimises the overlap between predicted and ground truth masks, making it particularly effective in handling class imbalance [108].

- **Focal Loss:** The Focal loss [13] is a variant of the Binary Cross-Entropy loss that down-weights easy examples and focuses on hard-to-classify pixels. It is defined as:

$$CE = \begin{cases} -\log(p), & \text{if } y = 1, \\ -\log(1-p), & \text{otherwise.} \end{cases} \tag{4.4}$$

Focal Loss introduces the modulating factor $(1 - p_t)^\gamma$, where p_t is the estimated probability for the ground truth class:

$$p_t = \begin{cases} p, & \text{if } y = 1, \\ 1-p, & \text{otherwise.} \end{cases} \tag{4.5}$$

Therefore, the Focal Loss is expressed as:

$$FL(p_t) = -\alpha_t (1-p_t)^\gamma \log(p_t), \tag{4.6}$$

where:

- α_t is a balancing factor to address class imbalance.
- $\gamma > 0$ is the focusing parameter that down-weights the loss for well-classified pixels.

The total loss function for DETR during the segmentation training phase combines the Hungarian loss for object detection and the mask-specific losses. It is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Hungarian}} + \lambda_{\text{DICE}}\mathcal{L}_{\text{DICE}} + \lambda_{\text{Focal}}\mathcal{L}_{\text{Focal}}, \tag{4.7}$$

where:

- λ_{DICE} and λ_{Focal} are hyperparameters that balance the contributions of the Dice and Focal losses during training.

By combining these loss components, DETR is able to jointly optimise object detection and segmentation tasks, ensuring spatial accuracy for bounding boxes and high-quality mask predictions.

Training Procedure for Segmentation To train DETR for segmentation, the pre-trained object detection model undergoes fine-tuning with the segmentation head using the combined loss functions. The dataset's annotations, previously comprised of bounding boxes, are enriched with segmentation masks in the form of polygons detailed in the JSON files. These masks delineate the exact outline of the utility poles and any attached warning signs.

The hyperparameters for the segmentation task were kept identical to those used for the utility pole and warning sign detection model described in Chapter 3. The only adjustment was to the number of epochs, which was set to **10**, as determined through an ablation study.

During fine-tuning, the segmentation head learns to produce high-quality masks that align with the detected objects' bounding boxes. The model is trained using backpropagation, where the gradients from the loss functions are used to update the weights of both the transformer and the segmentation head. The training process involves balancing the weights of the different loss components

to ensure both accurate object detection and precise segmentation.

Inference and Evaluation In the inference phase, DETR generates bounding boxes and corresponding segmentation masks for the detected utility poles. The performance is then evaluated using standard metrics such as IoU for the segmentation masks, complementing the object detection metrics discussed in Chapter 3.

In summary, the extension of DETR for segmentation involves the integration of a mask head, the introduction of additional loss functions suited to segmentation, and fine-tuning the model with enriched annotations. This enables the model not only to detect but also to segment the utility poles, providing detailed spatial information essential for analyses such as lean angle estimation.

4.3.1 Segmentation Model Fine-tuning

The segmentation task leveraged the pre-trained DETR model introduced in Chapter 3, where it was originally trained to detect utility poles and warning signs. For this task, we fine-tuned the model's segmentation head while freezing the detection weights. The fine-tuning focused specifically on refining the model's ability to segment utility poles by accurately delineating their shapes and boundaries. This was achieved using segmentation annotations in the dataset, which provided precise polygonal outlines of the poles as arrays of x and y coordinates stored in JSON files.

The process was efficient; by running the model for a limited number of epochs—10 in this case—the segmentation head reached a level of performance that was sufficient for detecting the boundaries of the poles with high accuracy. The minimal training time required demonstrates that DETR's underlying architecture, when fine-tuned appropriately, can quickly adapt to segmentation tasks without the need for a lengthy retraining process. This is especially advanta-

geous for tasks where a pre-trained object detection model already exists and segmentation is an extension of that detection.

The segmentation evaluation includes a standalone mask-quality assessment and an epoch-ablation analysis. Table 4.1 summarises the mask metrics, while Table 4.2 reports the epoch-ablation results. The selected checkpoint at epoch 3 achieved both the best mask AP@0.50 and the lowest matched-angle error in the sweep, indicating that the early fine-tuning stage was sufficient for the available segmentation data. Although later epochs can improve individual measures such as recall, they did not improve the combined mask-quality and lean-angle objective used for checkpoint selection. Figures 4.3 and 4.4 show the corresponding mask and angle-error trends. The DETR/DINO terminology in the standalone mask table denotes the transformer-family segmentation checkpoint used for the epoch sweep; the YOLOv8 rows document standalone mask quality for the trained YOLOv8 segmentation checkpoint alongside the DETR-based lean-angle pipeline.

Table 4.1: Standalone segmentation metrics.

Source	**Mask mAP@0.50:0.95**	**Mask AP@0.50**	**Mask recall**
Segmentation DETR/DINO epoch sweep	8.60	29.11	18.08
YOLOv8 segmentation five-fold mean	32.76	81.22	–
YOLOv8 segmentation phase-B fold 1	42.98	76.91	67.40

Table 4.2: Segmentation epoch-ablation results for validation loss, mask quality, and lean-angle error.

Epoch	**Val. loss**	**Mask mAP@0.50:0.95**	**Mask AP@0.50**	**Mask AR@100**	**Matched masks**	**Angle MAE**
1	25.50	7.20	25.30	15.90	803	8.90
2	28.10	5.60	19.00	17.10	800	10.30
3	25.90	8.60	29.10	18.10	795	5.89
4	26.60	8.20	27.70	18.60	824	7.40
5	26.60	6.10	21.40	16.70	835	7.30
6	28.90	7.40	25.20	17.20	813	6.90
7	32.30	3.00	11.50	15.60	797	6.80
8	25.80	6.20	22.20	17.10	814	9.20
9	24.60	6.60	23.30	18.20	835	6.70
10	26.70	5.90	19.60	18.70	825	6.20
11	26.00	8.30	25.80	22.20	828	6.30

The trained YOLOv8 segmentation checkpoint was also evaluated directly on

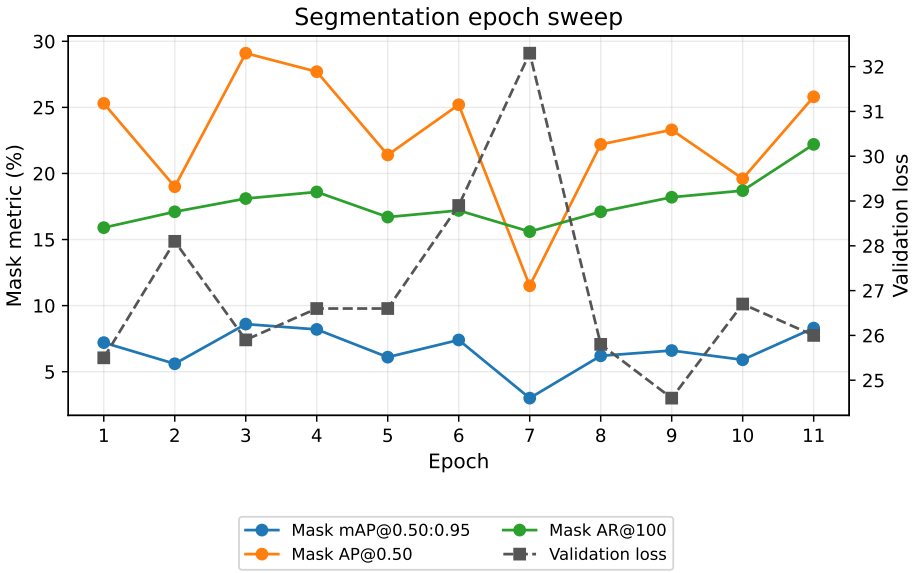


Figure 4.3: Validation loss, mask AP, and mask recall across the segmentation epoch sweep.

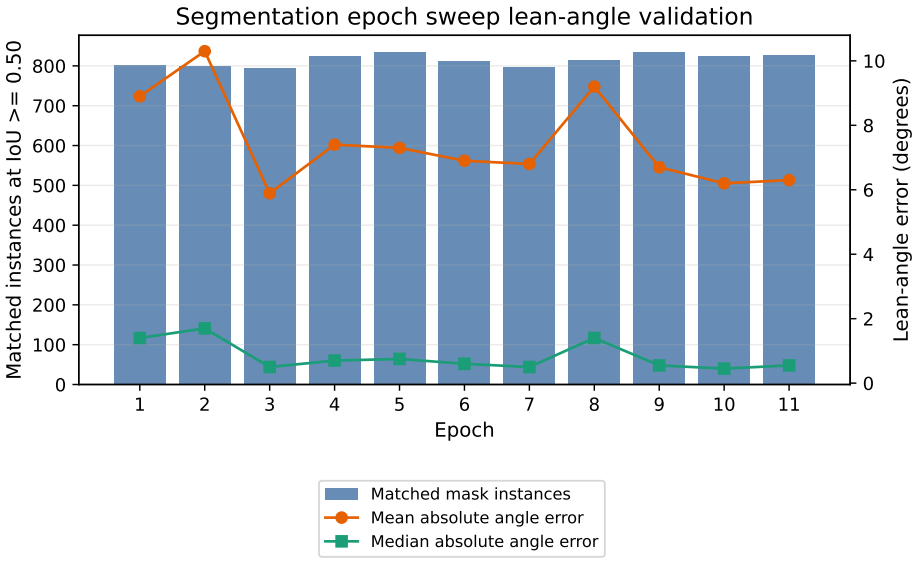


Figure 4.4: Lean-angle error for the segmentation epoch sweep. Angle errors are reported where predicted masks matched ground-truth pole masks at IoU ≥ 0.50.

the OHL-UK test split to quantify mask quality independently of the downstream lean-angle calculation. Table 4.3 reports instance-level IoU, Dice, precision, and recall, and Figure 4.5 shows the distribution of matched-mask IoU and Dice values. This confirms that the segmentation component is not evaluated only indirectly through angle error.

Table 4.3: Standalone instance-mask metrics for the trained YOLOv8 segmentation model on the OHL-UK test split.

Metric	**Value**
Images evaluated	920
Ground-truth pole masks	1,490
Predicted pole masks	1,458
Matched masks at IoU ≥ 0.5	1,274
Mean mask IoU	0.806
Median mask IoU	0.827
Mean Dice	0.890
Median Dice	0.905
Mean mask precision	0.873
Mean mask recall	0.917
Instance precision at IoU ≥ 0.5	0.874
Instance recall at IoU ≥ 0.5	0.855

The training-data proportion sensitivity experiment was repeated for segmentation. As shown in Table 4.4, the 80% training setting produced the strongest mean mask F1 and the lowest mean angle error, but the 50% and 60% settings remained close in angular accuracy. Figures 4.6 and 4.7 show the corresponding mask-quality and lean-angle trends.

Table 4.4: YOLOv8 segmentation sensitivity to reduced training-data proportions across five folds. Values are mean ± standard deviation.

Training data	Box mAP@0.50:0.95	Mask mAP@0.50:0.95	Mask F1	Mean angle error
50%	46.60 ± 0.87	30.40 ± 4.84	72.06 ± 1.74	1.95 ± 0.64
60%	47.70 ± 1.08	31.97 ± 5.06	73.36 ± 1.20	1.95 ± 0.68
80%	48.61 ± 1.13	31.91 ± 3.98	74.21 ± 2.01	1.92 ± 0.74

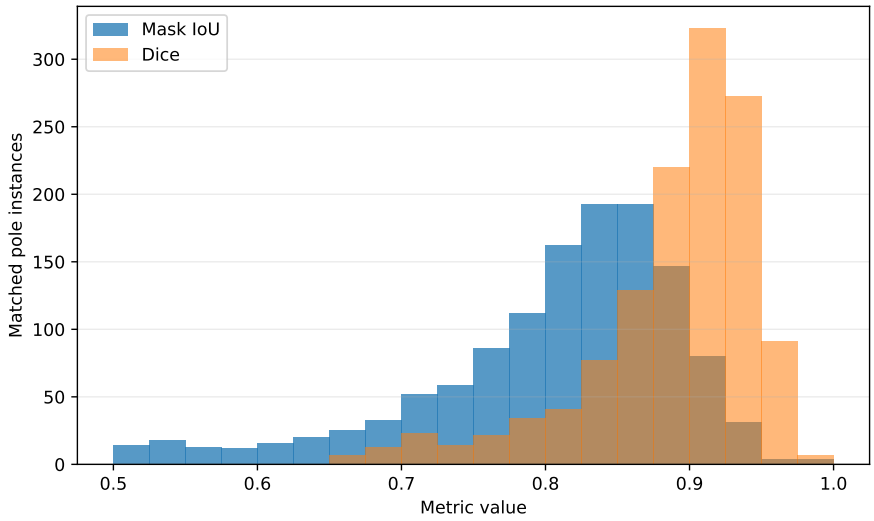


Figure 4.5: Distribution of matched-instance mask IoU and Dice scores for the trained YOLOv8 segmentation model on the OHL-UK test split.

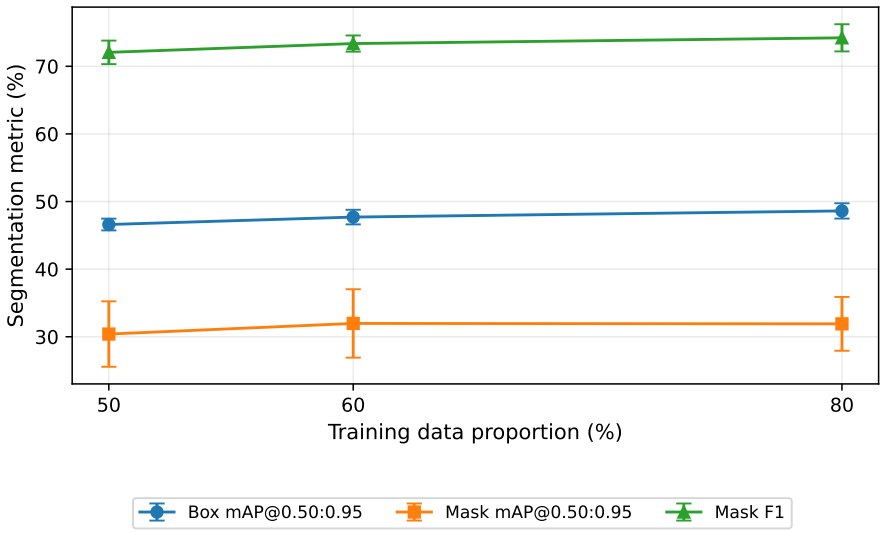


Figure 4.6: Effect of reducing the training-data proportion on YOLOv8 segmentation performance.

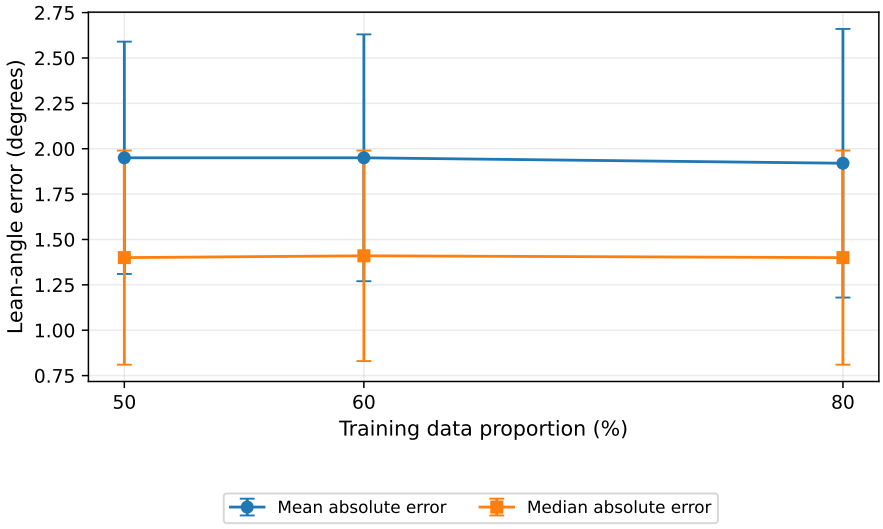
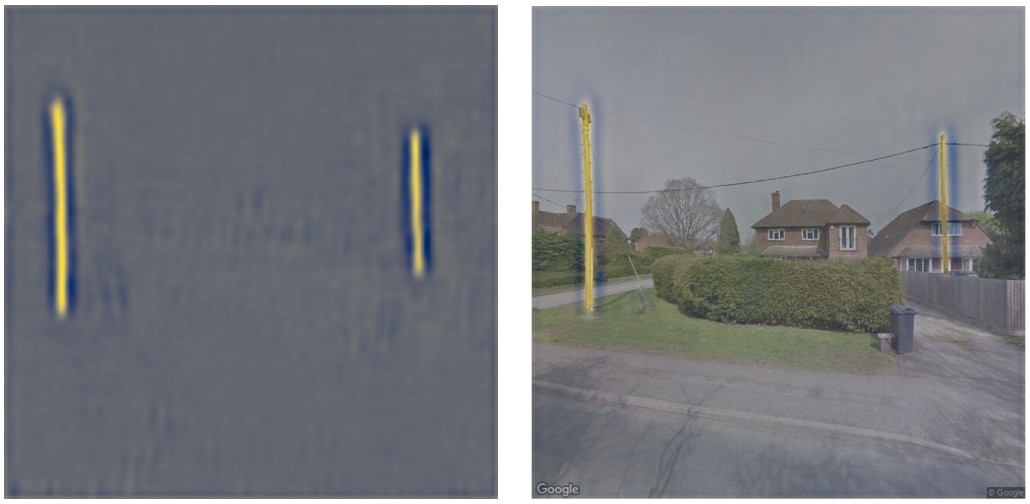


Figure 4.7: Effect of reducing the training-data proportion on YOLOv8 segmentation lean-angle error.

4.4 Utility Pole Lean Angle Estimation

For a given image, the DETR model returns a single segmentation image or prediction mask. The model employs a transformer-based architecture, consisting of an encoder and decoder. The encoder processes the input image features, and the decoder uses learned object queries to attend to these features and predict the final segmentation masks. The segmentation process involves predicting the coordinates of the segmentation points, which outline the utility poles.

When superimposed onto the original image, the result is shown in Figure 4.10.



Segmentation Image. Segmentation Image Superimposed.

Figure 4.10: Segmentation Image Superimposed on Original Image.

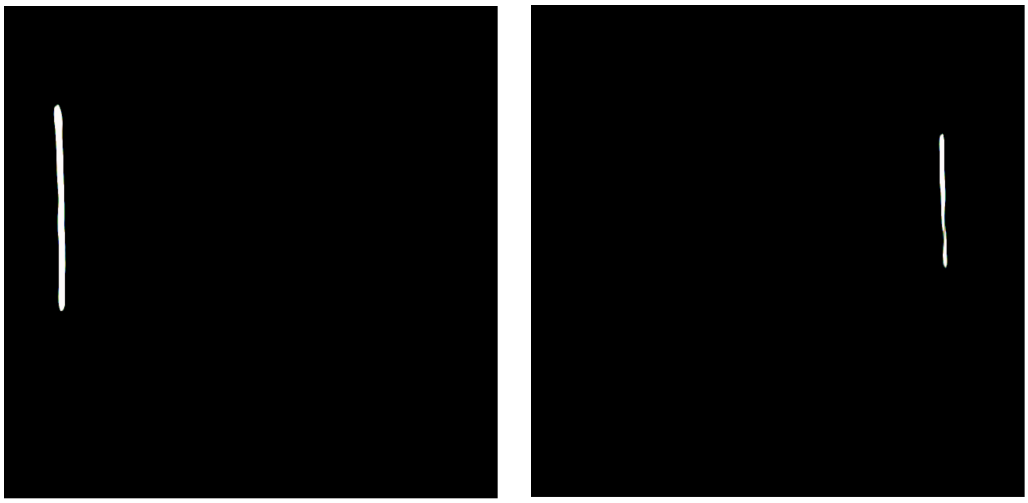
The predicted bounding boxes are used as coordinates to crop the segmentation mask. This is implemented in Python using libraries such as OpenCV and NumPy. The segmentation image is converted to greyscale and subjected to Otsu thresholding to convert the image to black and white pixels. The white pixels represent the segmentations, while the black pixels represent the background. The thresholding operation can be described mathematically as follows:

Given a greyscale image I, Otsu's method finds the threshold t that minimizes the intra-class variance, defined as:

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

where: - $\omega_1(t)$ and $\omega_2(t)$ are the probabilities of the two classes separated by the threshold t, - $\sigma_1^2(t)$ and $\sigma_2^2(t)$ are the variances of these two classes.

The segmentation results are shown in Figure 4.13.



Left Pole. Right Pole.

Figure 4.13: Individual Poles After Otsu Thresholding.

Small utility pole objects can be detected, but after thresholding, small segmentation masks often disappear. This does not pose a problem since measuring the angles of distant poles is inherently inaccurate.

As an example, consider Figure 4.17. Two poles and a sign are detected; however, the pole in the distance is too small to get any mask segmentations from it, and no angle estimation is provided. The figure shows the original image with detections and the mask images for each of the two poles, with the small pole having no mask.

Image with Predictions. Small Pole No Mask. Right Mask.

Figure 4.17: Example of Predictions with Empty Mask.

For each of the individual masks, the ‘fitLine‘ method from the OpenCV library is used to plot the line of best fit through the masks. The ‘fitLine‘ function calculates the optimal line that minimizes the orthogonal distances to all the segmented points. The mathematical representation of the ‘fitLine‘ method is:

$$\text{dist}(x_i, y_i) = |ax_i + by_i + c|$$

where (x_i, y_i) are the coordinates of the segmented points, and a, b, and c are the coefficients of the line equation. The function minimizes the sum of these distances to find the best fit line.

This line is then used to determine the lean angle of the utility pole using the gradient from the straight-line equation. The lines are clipped to the minimum and maximum heights of the predicted bounding boxes. In terms of azimuth and altitude, the gradient of the line provides an azimuth measurement, while the altitude can be considered in context of the vertical tilt with respect to the zenith.

An example is shown below in Figure 4.18.

Figure 4.18: Angle Estimation Using Best Fit Line.

By leveraging segmentation masks and employing the ‘fitLine‘ method, the lean angle estimation of utility poles is improved, providing a more reliable analysis for structural assessments.

Figure 4.19 further illustrates the segmentation results used for angle estimation.

Figure 4.19: Angle Estimation Using Segmentations.

4.4.1 Testing and Results

The segmentation model was tested on the same 920 test images that were used to evaluate the Utility Pole detection model, with the exception that the test set included polygon information specifically for the wooden utility poles. The model's performance was evaluated based on its ability to estimate the lean angles of utility poles, using both ground-truth segmentation points and predicted segmentation masks. The evaluation focused on the accuracy of these predicted

angles compared to the ground truth, as detailed in the previous section.

Lean Angle Estimation

The model excelled at estimating the lean angles for 1,367 out of the 1,433 true positive poles, leaving only 66 small poles without an angle calculation. The average error between the predicted and ground-truth angles was a mere 1.01 degrees, with a standard deviation of 1.95 degrees. Despite a maximum error of 37.69 degrees for one pole, the majority of predictions were highly accurate. Table 4.5 summarises the test-set lean-angle evidence.

Table 4.5: DETR segmentation lean-angle summary on the thesis test set.

Metric	Value
Pole detections with predicted angle	1,367
Pole detections without predicted angle	66
Unique images with predicted angle	917
Mean absolute angle error (degrees)	1.013
Standard deviation (degrees)	1.946
Median absolute angle error (degrees)	0.470
Maximum absolute angle error (degrees)	37.690
Mean IoU for angle-evaluable detections	0.729

As detailed in Table 4.6, 98% of the angle predictions had errors within 5 degrees, and just 2% of poles exhibited errors greater than 5 degrees. Furthermore, Table 4.7 shows that 90% of the poles had an angle error within 2 degrees, with 75% of predictions falling within 1 degree of the actual angle.

These results highlight the robustness of the fine-tuned DETR segmentation model for accurately estimating the lean angles of utility poles, even when compared to prior research.

Table 4.6: Pole Lean Angle Error.

Angle Error (Degrees)	**Count**
No Predictions	66
$0 \leq \theta_{\text{Error}} \leq 5$	1,333
$5 < \theta_{\text{Error}} \leq 10$	26
$10 < \theta_{\text{Error}} \leq 15$	3
$15 < \theta_{\text{Error}} \leq 20$	3
$25 < \theta_{\text{Error}} \leq 30$	1
$30 < \theta_{\text{Error}} \leq 40$	1

Table 4.7: Breakdown of Top Pole Angle Error Count.

Angle Error (Degrees)	**Count**
$0 \leq \theta_{\text{Error}} \leq 1$	1,002
$1 < \theta_{\text{Error}} \leq 2$	203
$2 < \theta_{\text{Error}} \leq 3$	68
$3 < \theta_{\text{Error}} \leq 4$	35
$4 < \theta_{\text{Error}} \leq 5$	25

Figure 4.20 confirms that the large-error cases are rare relative to the main concentration of predictions below 5°. Figure 4.21 shows representative highest-error examples. The largest deviations occur where mask quality, occlusion, partial pole visibility, non-standard pole geometry, surface defects, or ambiguity in the visible pole centreline affects the fitted line. These cases illustrate that the method is strongest when the visible mask follows a clean, approximately cylindrical pole centreline and less reliable when the pole is fragmented, partly hidden, damaged, fitted with substantial attachments, non-cylindrical in silhouette, or viewed under difficult image-plane geometry. This limitation is important for de-

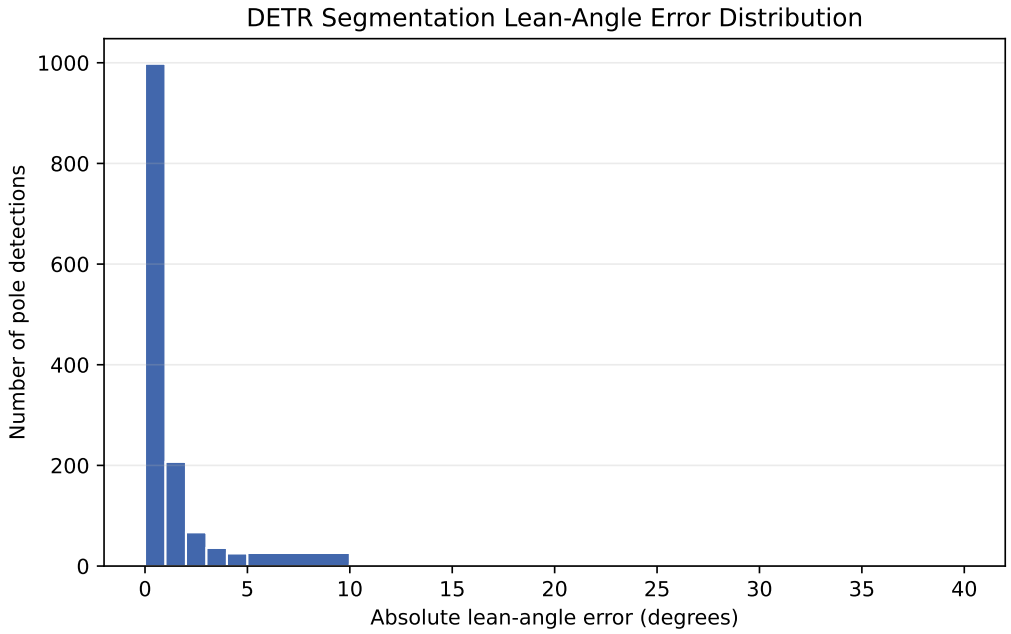


Figure 4.20: Distribution of DETR segmentation lean-angle errors on the thesis test set.

ployment because chipped, decayed, bent, or otherwise irregular poles can cause the fitted line to follow the visible defect boundary rather than the structural centreline of the pole.

Comparison with Prior Work

A direct same-dataset comparison with the closest prior lean-angle studies could not be reported because the cited studies do not provide reusable public image and annotation datasets. The comparison is therefore limited to the information available from the published papers: dataset scale, data accessibility, imaging context, and reported lean-angle distributions. This limitation is important because apparent differences in angle-estimation accuracy can be affected by pole height in the image, imaging viewpoint, occlusion, and the proportion of near-vertical poles in the test set. The OHL-UK row in Table 4.8 is included as the

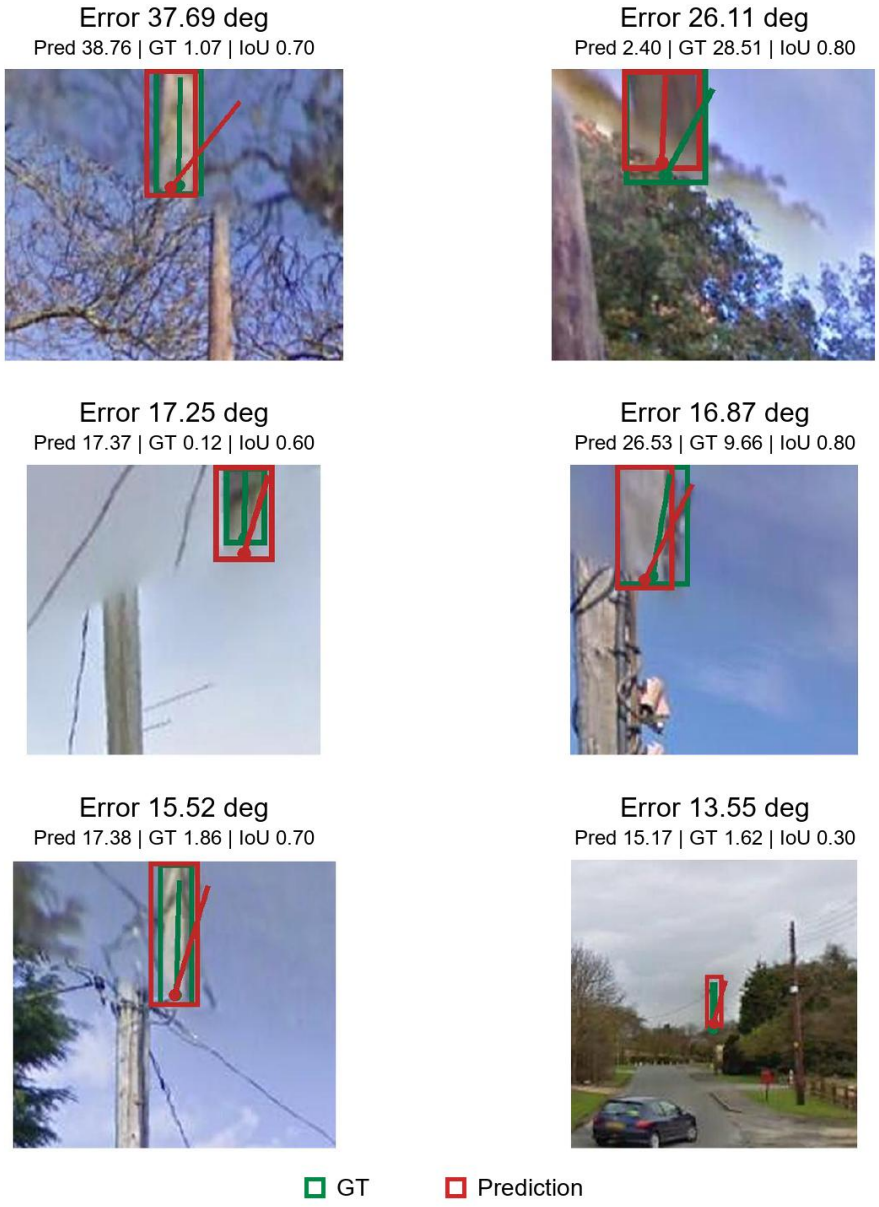


Figure 4.21: Highest-error DETR segmentation lean-angle examples. Green indicates the ground truth and red indicates the prediction.

benchmark introduced in this thesis, not as prior work.

Table 4.8: Dataset accessibility and comparison basis for lean-angle evaluation.

Study or benchmark	Reported data	Direct-comparison status
Zhu et al. [10]	64 UAV images containing 84 utility poles.	No reusable public image or annotation dataset was identified, so direct evaluation on the same images was not possible.
Alam et al. [17]	UAV case study in Beaumont, Texas, reporting lean-angle categories for 173 utility poles.	No reusable public image or annotation dataset was identified; comparison is restricted to reported angle-bin distributions.
This thesis: OHL-UK dataset	920 held-out test images containing 1,490 pole angle annotations, with 1,433 true-positive detections used for lean-angle evaluation.	Dataset and trained models are made publicly available, enabling independent evaluation and reuse.

The OHL-UK test annotations have a mean absolute lean angle of 2.19°, a median of 1.58°, and a maximum of 28.51° across 1,490 annotated poles. In angle-bin terms, 91.54% of annotated poles fall within 0°–5°, 6.98% fall within 5°–10°, and 1.48% exceed 10°. Figure 4.22 compares this distribution with the reported angle categories in Alam et al. [17]. The OHL-UK evaluation therefore provides a larger and openly reusable benchmark, but it should not be interpreted as a direct replication of prior studies on identical imagery.

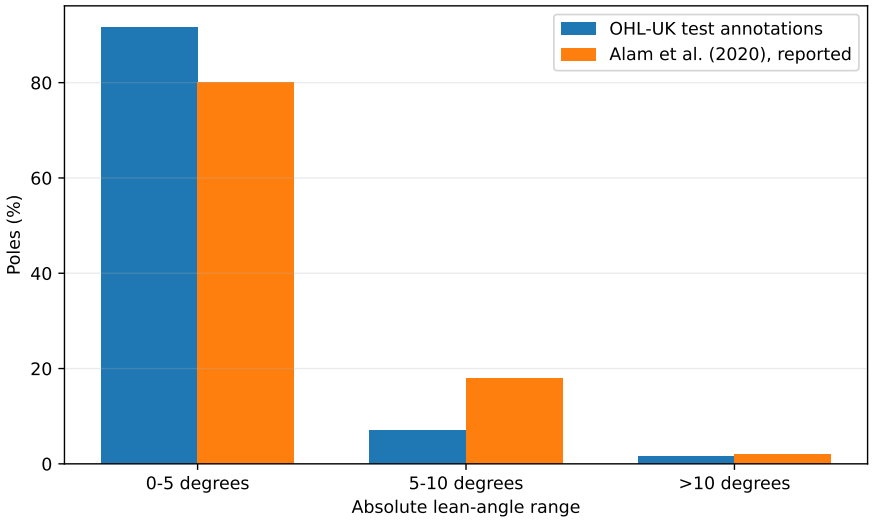


Figure 4.22: Comparison between the OHL-UK lean-angle distribution and the angle-bin distribution reported by Alam et al. [17].

4.5 Summary

This work has successfully extended the DETR model to perform both detection and precise segmentation of wooden utility poles and their associated warning signs. By incorporating a segmentation head and fine-tuning the model on a dataset enriched with pixel-level annotations, the model achieved accurate mask generation and enabled calculation of important structural metrics such as lean angle—an essential factor for pole assessment and maintenance.

The lean angle estimation results were particularly strong. The model provided predictions for 1,367 out of 1,433 poles, with an average error of just 1.01 degrees and a standard deviation of 1.95 degrees. Notably, 98% of angle predictions had errors within 5 degrees, and 90% were within 2 degrees. The evaluation is larger than the directly comparable published studies and is supported by an openly reusable dataset and model release, although the absence of public image-level data from prior studies prevents a direct same-dataset comparison.

These results demonstrate the robustness and scalability of the fine-tuned DETR segmentation model for utility pole analysis. The integration of segmentation within the object detection pipeline provides significant advantages, enabling detailed spatial understanding and precise structural assessments under real-world conditions.

More broadly, this study shows that a pre-trained detection model can be adapted to deliver fine-grained, task-specific measurements such as lean angle without the need for a bespoke architecture. Evaluating deployment speed and resource usage in realistic industrial settings is an important direction for future work.

Chapter 5

Conclusion

This thesis has presented a comprehensive investigation into the application of deep learning techniques for the detection and assessment of wooden utility poles and their attached warning signs using publicly available imagery, such as Google Street View. Motivated by the limitations of traditional inspection methods and the shortcomings of prior automated systems, this work introduced a robust, end-to-end pipeline that combines detection and segmentation to support scalable and reliable infrastructure monitoring.

Three core objectives guided this research: (1) to develop a state-of-the-art deep learning model for detecting utility poles and their attached warning signs; (2) to support accurate lean angle estimation through precise segmentation; and (3) to automate the inspection process in order to improve efficiency, reduce costs, and enhance safety in electrical infrastructure maintenance.

5.1 Contributions

The contribution of this thesis is applied and methodological rather than the proposal of a wholly new neural-network architecture. Its novelty lies in curating and releasing a domain-specific dataset, adapting and benchmarking established

detection and segmentation models for wooden utility pole inspection, and integrating mask-based image-plane lean-angle estimation into the same inspection workflow.

To achieve these aims, the thesis introduced a detection and segmentation system based on the DEtection TRansformer (DETR), a transformer-based architecture that unifies convolutional and attention mechanisms. The DETR model was benchmarked against established detectors—YOLO v3-Tiny, RetinaNet, and Faster R-CNN—on a large, diverse dataset constructed from over 670,000 geographic coordinates provided by UK Power Networks. DETR significantly outperformed these models, achieving a mean average precision of 90.43% for utility pole detection and 88.26% for warning sign detection.

A further contribution of this work is the extension of DETR to include segmentation capabilities, enabling the estimation of pole lean angles directly from the predicted masks. This segmentation-based approach was applied to 1,433 true positive poles, producing lean angle estimates with an average error of 1.01 degrees and 98% of errors falling within 5 degrees. Relative to previous studies, this work provides a larger openly documented evaluation and reports the accessibility limitation that prevents direct same-dataset replication of the published lean-angle methods.

In addition to the model architecture, this thesis contributes a large, high-quality dataset of utility pole and warning sign annotations tailored to real-world conditions. Unlike prior work focused on poles with crossarms or drone imagery, this dataset includes challenging examples of straight wooden poles without crossarms, captured from diverse angles and lighting conditions using ground-level imagery. The inclusion of warning sign detection further improves the legal and operational relevance of the system.

In summary, this thesis demonstrates that deep learning models—particularly transformer-based architectures—can be effectively applied to large-scale, real-

world infrastructure assessment tasks. By integrating detection and segmentation into a unified framework, the proposed approach offers a scalable, accurate, and cost-effective alternative to manual or drone-based inspection methods. While further work is needed to address extreme clutter and urban complexity, the methods developed here lay a strong foundation for future research and industrial deployment.

5.2 Future Work

The work presented in this thesis has made significant strides in automating the detection and assessment of utility poles and their attached warning signs using advanced deep learning models. While the methods developed have demonstrated high accuracy in challenging environments such as Google Street View images, several areas for future research and improvement remain.

One potential avenue for future work is the integration of additional safety-critical features into the detection pipeline. Although this study focused on identifying utility poles and attached warning signs, other important elements, such as insulators, transformers, or crossarms, could be included to create a more comprehensive system for infrastructure monitoring. Expanding the dataset to cover a broader range of pole-mounted equipment and refining the model to detect these additional components would provide utility companies with a more detailed understanding of the overall health of the poles and their associated systems.

Another area for further research is the application of advanced transfer learning techniques. By leveraging pre-trained models on related tasks, such as object detection in street-level imagery, the proposed model could be fine-tuned to detect previously unseen objects or faults with minimal training data. This approach would be particularly valuable for detecting rare or emerging issues, such as newly

introduced warning signs or pole attachments. Investigating one-shot or few-shot learning techniques would enable the model to quickly adapt to new conditions without the need for extensive retraining.

In addition to improving the range of detectable features, future work could also focus on enhancing the robustness of the system in more diverse geographical and environmental conditions. For instance, rural environments with different types of poles and terrain present unique challenges that may not be fully addressed by the current model. By incorporating additional datasets from varying climates, seasons, and geographic regions, the model could be further generalised to handle a wider array of real-world conditions.

Finally, a future extension could involve the inclusion of other geospatial data, such as weather conditions or soil type, which could affect the structural integrity of utility poles over time. By incorporating these external factors into the analysis, the system could provide predictive insights that help to identify poles at higher risk of failure due to environmental factors.

By addressing these avenues of future work, the proposed model can be further developed into a more comprehensive, adaptable, and predictive tool for automating the inspection and maintenance of utility pole infrastructure.

In conclusion, this thesis contributes to automated visual inspection of electrical infrastructure by offering a scalable, cost-effective solution for utility pole detection and lean angle estimation. The methodologies developed here provide a foundation upon which future work can build, with the goal of achieving fully autonomous, intelligent systems for infrastructure management and maintenance.

References

[1] "Poles & piles : Timber development uk." [Online]. Available: https://timberdevelopment.uk/products-database/poles-piles/?

[2] "Why do wooden utility poles fail? — polesaver." [Online]. Available: https://polesaver.com/blog/why-do-wooden-utility-poles-fail/?

[3] E. D. Megaw, "Factors affecting visual inspection accuracy," *Applied Ergonomics*, vol. 10, pp. 27–32, 1979.

[4] R. Witcomb, C. Potter, J. Richardson, J. Ward, S. Darcy, M. Hurst, V. McGregor, G. Cooper, S. Horne, G. Strbac, R. Hull, U. Kaltenbronn, J. Hubbard, H. Parker, R. Blomfield-Smith, A. Bryce, and J. Baldwin, "Riio-2 challenge group independent report for ofgem on riio-2 business plans," Independent report for Ofgem, 2020.

[5] "Get compensation if you have a power cut - citizens advice." [Online]. Available: https://www.citizensadvice.org.uk/consumer/energy/energy-supply/problems-with-your-energy-supply/get-compensation-if-you-have-a-power-cut/

[6] "Storm arwen: Electricity firms 'unacceptable' response criticised - bbc news." [Online]. Available: https://www.bbc.co.uk/news/uk-england-60419238

[7] "Western power distribution - be aware of what's around you." [Online]. Available: https://www.westernpower.co.uk/customers-and-community/health-safety/be-aware-of-whats-around-you

[8] "Health {&} safety statistics for great britain 2018/19 — peoplesafe." [Online]. Available: https://peoplesafe.co.uk/blogs/health-safety-statistics-for-great-britain-2018-19/

[9] Google, "Google street view – explore natural wonders and world landmarks." [Online]. Available: https://www.google.co.uk/intl/en-GB/streetview/https://www.google.com/streetview/

[10] Z. Zhu, J. Zhang, M. M. Alam, B. E. Tokgoz, and S. Hwang, "Automatic utility pole inclination angle measurement using unmanned aerial vehicle and deep learning," *IISE Annual Conference and Expo 2019*, 2019.

[11] J. Kim, M. Kamari, S. Lee, and Y. Ham, "Large-scale visual data–driven probabilistic risk assessment of utility poles regarding the vulnerability of power distribution infrastructure systems," *Journal of Construction Engineering and Management*, vol. 147, p. 4021121, 2021.

[12] P. Adarsh, P. Rathi, and M. Kumar, "Yolo v3-tiny: Object detection and recognition using one stage improved model," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 687–694.

[13] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 2999–3007, 2017.

[14] W. Zhang, C. Witharana, W. Li, and C. Zhang, "Using Deep Learning to Identify Utility Poles with Crossarms and Estimate Their Locations from Google Street View Images," *Sensors*, vol. 18, no. 8, p. 2484, 2018.

[15] J. Park, U. Shin, G. Shim, and K. Joo, "Vehicular multi-camera sensor system for automated visual inspection of electric power distribution equipment," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 281–288.

[16] L. Liu, T. Zhang, K. Zhao, A. Wiliem, K. Astin-Walmsley, and B. Lovell, "Deep inspection: An electrical distribution pole parts study via deep neural networks," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2019-Septe, pp. 4170–4174, 2019.

[17] M. M. Alam, Z. Zhu, B. E. Tokgoz, J. Zhang, and S. Hwang, "Automatic assessment and prediction of the resilience of utility poles using unmanned aerial vehicles and computer vision techniques," *International Journal of Disaster Risk Science*, vol. 11, pp. 119–132, 2020. [Online]. Available: https://doi.org/10.1007/s13753-020-00254-1

[18] C. Dickinson, "Ohl-uk: Wooden utility pole and electrical sign corpus with trained detection and segmentation models," University of Strathclyde research dataset, 2025, dataset openly available under CC BY 4.0 licence. [Online]. Available: https://doi.org/10.15129/df7cc895-1091-4cd6-bf2c-7b5ebd203e55

[19] C. Dickinson and G. Di Caterina, "Advancing utility pole and sign detection through deep learning," in *36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24–27, 2025*. BMVA, 2025. [Online]. Available: https://bmva-archive.org.uk/bmvc/2025/assets/papers/Paper_976/paper.pdf

[20] H. Sharma, V. Adithya, T. Dutta, and P. Balamuralidhar, "Image analysis-based automatic utility pole detection for remote surveillance," *2015 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2015*, 2015.

[21] Energy Networks Association, "FutureSmart Consultation Report," Online report, 2019. [Online]. Available: https://innovation.ukpowernetworks.co.uk/wp-content/uploads/2019/07/FutureSmart-Consultation-Report.pdf

[22] Ofgem, "The gb electricity retail market," Online report, 2016. [Online]. Available: https://www.ofgem.gov.uk/electricity/distribution-networks/gb-electricity-distribution-network

[23] Energy Networks Association, "Who is my network operator?" 2026, accessed: 31 May 2026. [Online]. Available: https://www.energynetworks.org/customers/find-my-network-operator

[24] Health and Safety Executive, "Avoiding danger from overhead power lines," Guidance Note GS6 (Fourth edition), 2013, accessed: 2026-06-01. [Online]. Available: https://www.hse.gov.uk/pubns/gs6.pdf

[25] I. Ce, "Power line inspection - an uav concept," Project report, pp. 55–83, 2000.

[26] D. I. Jones, C. C. Whitworth, A. W. G. Duller, and G. K. Earp, "Aerial video inspection of overhead power lines," *Power Engineering Journal*, vol. 15, pp. 25–32, 2001. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/pe_20010103

[27] J. Katra and F. Pernu, "A survey of mobile robots for distribution power line inspection," Conference paper, pp. 485–493, 2010.

[28] Bsi, *Requirements for electrical installations.* Institution of Engineering and Technology, 2008, vol. 2.

[29] UK Government, "The electricity at work regulations 1989," Statutory instrument, 1989. [Online]. Available: http://www.legislation.gov.uk/uksi/1989/635/contents/made

[30] UK Parliament, "The Electricity Safety, Quality and Continuity Regulations 2002," Statutory Instrument 2002 No. 2665, 2002, made 24 October 2002; came into force 31 January 2003. [Online]. Available: https://www.legislation.gov.uk/uksi/2002/2665/pdfs/uksi_20022665_en.pdf

[31] IEEE, "IEEE: Advancing Technology for Humanity," accessed: 2026-06-01. [Online]. Available: https://www.ieee.org/

[32] Institution of Engineering and Technology, "The Institution of Engineering and Technology," accessed: 2026-06-01. [Online]. Available: https://www.theiet.org/

[33] Energy Networks Association, "Energy Networks Association Homepage," accessed: 2026-06-01. [Online]. Available: https://www.energynetworks.org/

[34] CIGRE, "CIGRE Homepage," accessed: 2026-06-01. [Online]. Available: https://www.cigre.org/

[35] CIGRE Study Committee B2, *Overhead Lines.* Springer, 2017.

[36] G. Earp, R. Eyre-Walker, A. Ellam, and A. Thomas, "Advanced aerial inspection and asset management of electricity towers," *Proceedings of the IEEE International Conference on Transmission and Distribution Construction and Live Line Maintenance, ESMO*, 2011.

[37] CIGRE Working Group 22.13, "Cigre technical brochure 175 - management of existing transmission lines," CIGRE, Tech. Rep. Technical Brochure 175, 2000.

[38] CIGRE Canada, "2015 cigre canada conference," Conference proceedings, 2015.

[39] J. E. See, "Visual inspection: a review of the literature," Sandia National Laboratories, Tech. Rep. SAND2012-8590, 2012. [Online]. Available: https://prod-ng.sandia.gov/techlib-noauth/access-control.cgi/2012/128590.pdf

[40] A. Pagnano, M. Höpf, and R. Teti, "A roadmap for automated power line inspection. maintenance and repair." *Procedia CIRP*, vol. 12, pp. 234–239, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.procir.2013.09.041

[41] P. Campoy, P. J. Garcia, A. Barrientos, J. D. Cerro, I. Aguirre, A. Roa, R. Garcia, and J. M. Muñoz, *An stereoscopic vision system guiding an autonomous helicopter for overhead power cable inspection.* Springer, 2001.

[42] R. Szeliski, *Computer Vision.* Springer, 2011, vol. 42. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20549881%5Cnhttp://link.springer.com/10.1007/978-1-84882-935-0

[43] A. S. Hassanein, S. Mohammad, M. Sameer, and M. E. Ragab, "A survey on hough transform, theory, techniques and applications," *arXiv preprint arXiv:1502.02160*, p. 18, 2015. [Online]. Available: http://arxiv.org/abs/1502.02160

[44] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, 1972.

[45] R. S. Gonçalves and J. C. M. Carvalho, "Review and latest trends in mobile robots used on power transmission lines," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1–14, 2013.

[46] Y. Zhang, S. Fu, X. Zhao, Z. Liang, M. Tan, and Y. Zhang, *Visual Navigation for a Power Transmission Line Inspection Robot*. Springer, Berlin, Heidelberg, 2006, pp. 887–892. [Online]. Available: http://link.springer.com/10.1007/11816171_111

[47] L. Tavares and J. S. Sequeira, "Riol - robotic inspection over power lines," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 6, pp. 108–113, 2007.

[48] V. N. Nguyen, R. Jenssen, and D. Roverso, "Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning," *International Journal of Electrical Power and Energy Systems*, vol. 99, pp. 107–120, 2018. [Online]. Available: https://doi.org/10.1016/j.ijepes.2017.12.016

[49] Z. Li, Y. Liu, R. Hayward, J. Zhang, and J. Cai, "Knowledge-based power line detection for uav surveillance and inspection systems," *2008 23rd International Conference Image and Vision Computing New Zealand, IVCNZ*, pp. 1–6, 2008.

[50] C. Martinez, C. Sampedro, A. Chauhan, and P. Campoy, "Towards autonomous detection and tracking of electric towers for aerial power line inspection," *2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings*, pp. 284–295, 2014.

[51] X. Wang and Y. Zhang, "Insulator identification from aerial images using support vector machine with background suppression," *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, pp. 892–897, 2016.

[52] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[53] B. Microbiana, D. Hidalgo, M. Anthimopoilos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network," *Ieee Transactions on Medical Imaging*, vol. 35, pp. 1207–1216, 2016.

[54] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 2110–2118. [Online]. Available: http://ieeexplore.ieee.org/document/7780601/

[55] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2015, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/7301342/

[56] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Arxiv.Org*, 2015.

[58] Z. Zhao, G. Xu, Y. Qi, N. Liu, and T. Zhang, "Multi-patch deep features for power line insulator status classification from aerial images," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2016-Octob, pp. 3187–3194, 2016.

[59] A. Krizhevsky, I. Sutskever, and H. G. E., "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems 25 (NIPS2012)*, p. 1–9, 2012. [Online]. Available: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[60] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015.

[61] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[62] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, pp. 1–48, 2019.

[63] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457309000259

[64] T.-Y. Lin, G. Patterson, M. R. Ronchi, Y. Cui, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, L. Zitnick, and P. Dollár, "Coco - common objects in context," *COCO Dataset*, 2018.

[65] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

[66] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei,

"Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015. [Online]. Available: http://link.springer.com/10.1007/s11263-015-0816-y

[67] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 658–666, 2019.

[68] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *arXiv:2005.12872*, 2020. [Online]. Available: http://arxiv.org/abs/2005.12872

[69] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 5999–6009, 2017.

[70] H. W. Kuhn, *The Hungarian method for the assignment problem.* Springer Berlin Heidelberg, 2010.

[71] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.* International Conference on Learning Representations, ICLR, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[72] "imgaug — imgaug 0.2.6 documentation." [Online]. Available: https://imgaug.readthedocs.io/en/latest/

[73] "Image preprocessing - keras documentation." [Online]. Available: https://keras.io/preprocessing/image/

[74] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 1–35. [Online]. Available: http://arxiv.org/abs/1702.05374

[75] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387. [Online]. Available: http://arxiv.org/abs/1605.06409

[76] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. yang Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, 2016, pp. 21–37.

[77] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788. [Online]. Available: http://arxiv.org/abs/1506.02640

[78] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3296–3305, 2017.

[79] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, 1986.

[80] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[81] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[82] J. Radon, "On the determination of functions from their integral values along certain manifolds," *IEEE Transactions on Medical Imaging*, 1986.

[83] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[84] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[85] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[86] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy Systems*, 1995.

[87] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," arXiv:1402.3722, 2 2014. [Online]. Available: http://arxiv.org/abs/1402.3722

[88] T. Maiti, "Horvitz–thompson estimator," *International Encyclopedia of Statistical Science*, pp. 637–638, 2011. [Online]. Available: https://link.springer.com/referenceworkentry/10.1007/978-3-642-04898-2_291

[89] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1511.00561

[90] Brostow, Shotton, Fauqueur, and Cipolla, "Camvid - object recognition in video dataset." [Online]. Available: http://web4.cs.ucl.ac.uk/staff/g.brostow/MotionSegRecData/http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/

[91] S. F. Stefenon, K. C. Yow, A. Nied, and L. H. Meyer, "Classification of distribution power grid structures using inception v3 deep neural network," *Electrical Engineering*, vol. 104, pp. 4557–4569, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s00202-022-01641-1

[92] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12 2016, pp. 2818–2826. [Online]. Available: http://arxiv.org/abs/1512.00567

[93] T. Chen, M. M. Othman, X. Wang, Y. Zhu, Z. Zhu, and J. Xiao, "Ecml driven geographical location of utility poles in smart grid: Data analysis and high-definition recognition," *Applied Soft Computing*, vol. 134, p. 109973, 2023.

[94] H. Dong, X. Chen, S. Särkkä, and C. Stachniss, "Online pole segmentation on range images for long-term lidar localization in urban environments," *Robotics and Autonomous Systems*, vol. 159, p. 104283, 2023.

[95] H. Manninen, C. J. Ramlal, A. Singh, J. Kilter, and M. Landsberg, "Multi-stage deep learning networks for automated assessment of electricity transmission infrastructure using fly-by images," *Electric Power Systems Research*, vol. 209, p. 107948, 2022.

[96] G. Developers, "Street view static api," 2021. [Online]. Available: https://developers.google.com/maps/documentation/streetview/overview

[97] A. Dutta and A. Zisserman, "The vgg image annotator (via)," *ArXiv*, 2019.

[98] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2017.

[99] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, pp. 111–147, 1974.

[100] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, 12 2019.

[101] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 113–123, 2019.

[102] S. Mounsaveng, I. Laradji, I. B. Ayed, D. Vazquez, and M. Pedersoli, "Learning data augmentation with online bilevel optimization for image classification," *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pp. 1690–1699, 2021.

[103] "Introduction - archie-west documentation." [Online]. Available: https://docs.hpc.strath.ac.uk/user-guide/

[104] J. Park, U. Shin, G. Shim, and K. Joo, "Vehicular multi-camera sensor system for automated visual inspection of electric power distribution equipment vehicular multi-camera sensor system for automated visual inspection of electric power distribution equipment," *IEEE International Conference on Intelligent Robots and Systems*, vol. (2019) 281, 2019.

[105] "End-to-end object detection with transformers." [Online]. Available: https://ai.facebook.com/blog/end-to-end-object-detection-with-transformers/

[106] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 640–651, 4 2017.

[107] M. M. Alam, B. E. Tokgoz, and S. Hwang, "Framework for measuring the resilience of utility poles of an electric power distribution network," *International Journal of Disaster Risk Science*, vol. 10, pp. 270–281, 6 2019.

[108] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings of the Fourth International Conference on 3D Vision*, 6 2016, pp. 565–571. [Online]. Available: http://arxiv.org/abs/1606.04797

Appendix A

RetinaNet

A.1 Ablation Study Plots

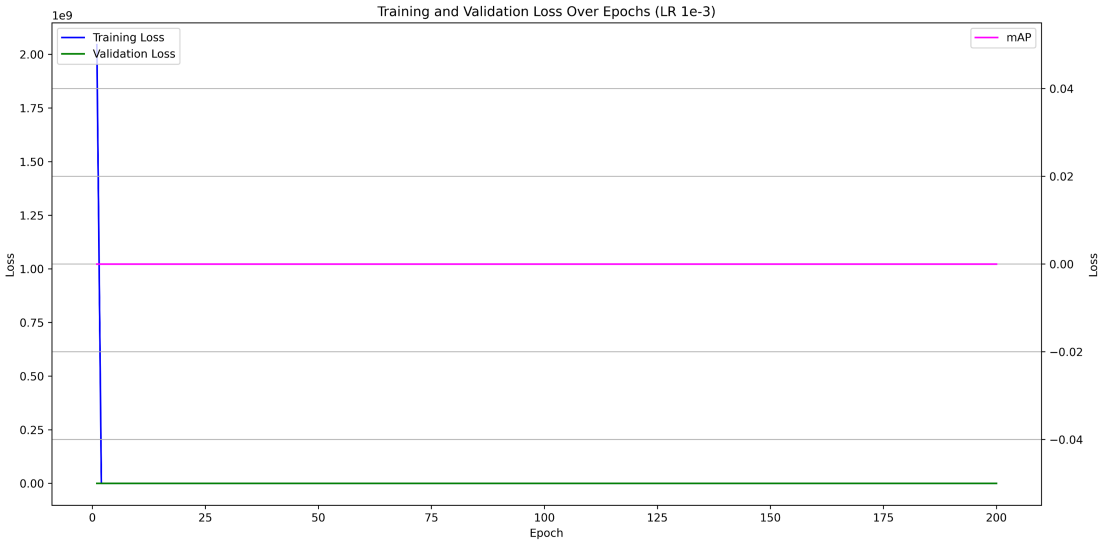


Figure A.1: Training and validation losses and mAP against epochs for a learning rate of 1×10^{-3}.

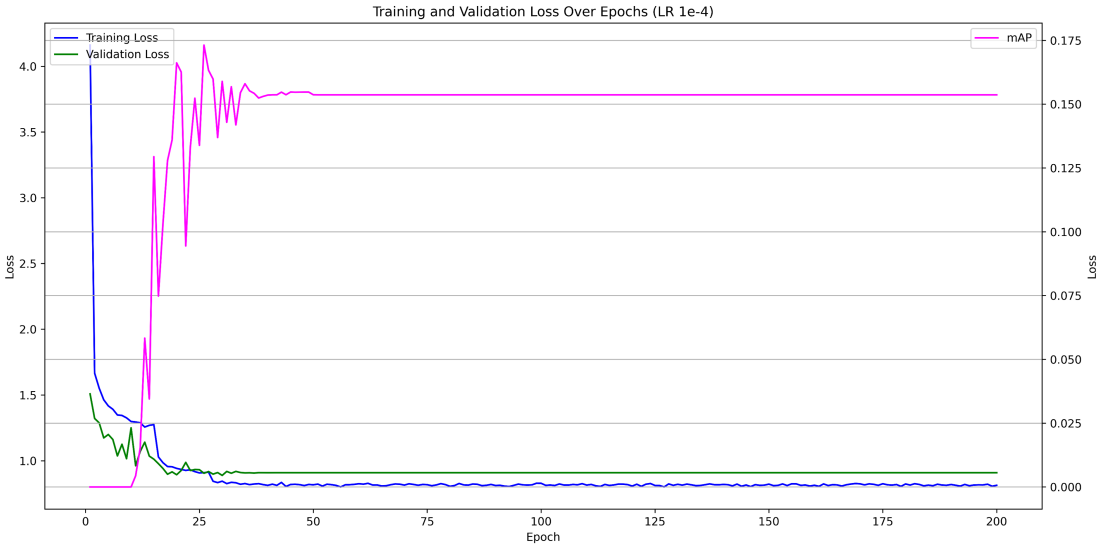


Figure A.2: Training and validation losses and mAP against epochs for a learning rate of 1×10^{-4}.

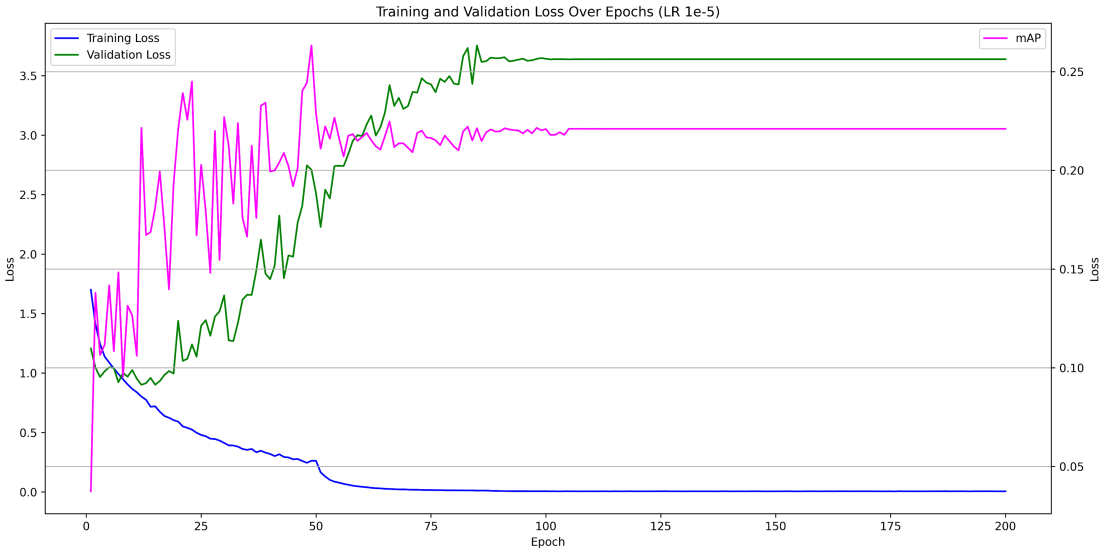


Figure A.3: Training and validation losses and mAP against epochs for a learning rate of 1×10^{-5}.

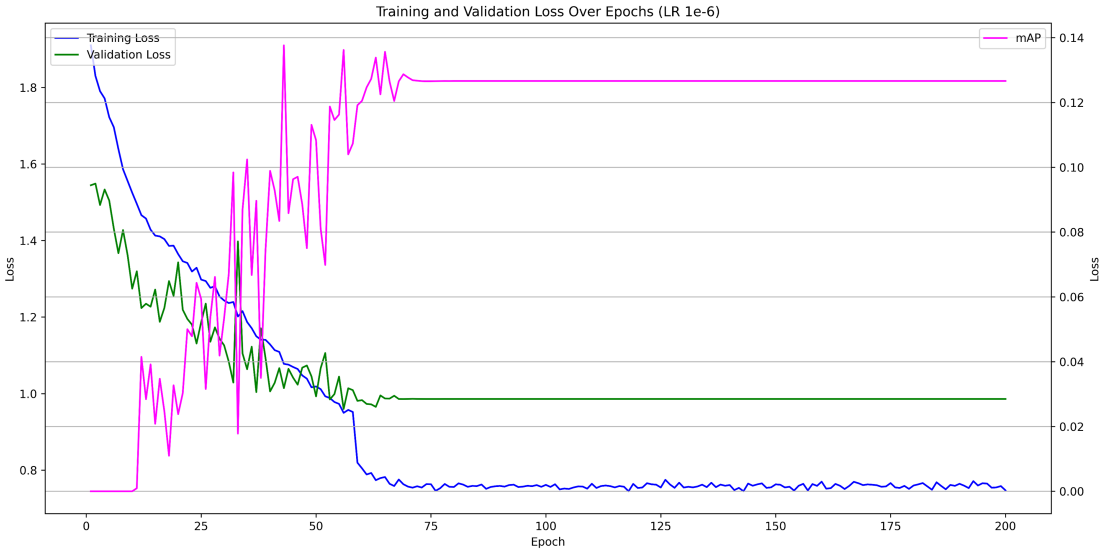


Figure A.4: Training and validation losses and mAP against epochs for a learning rate of 1×10^{-6}.

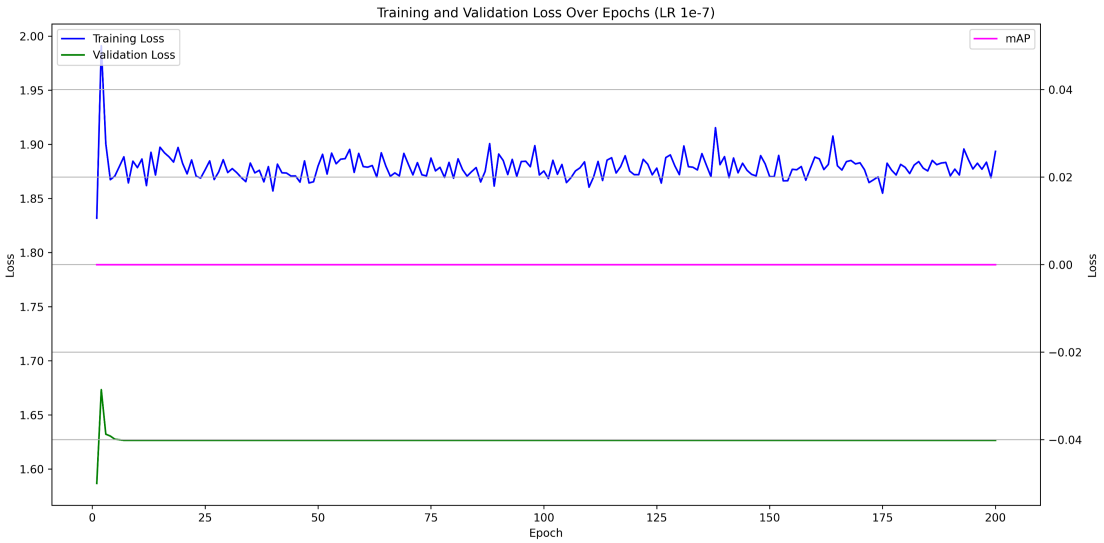


Figure A.5: Training and validation losses and mAP against epochs for a learning rate of 1×10^{-7}.

A.2 Cross-Validation Plots

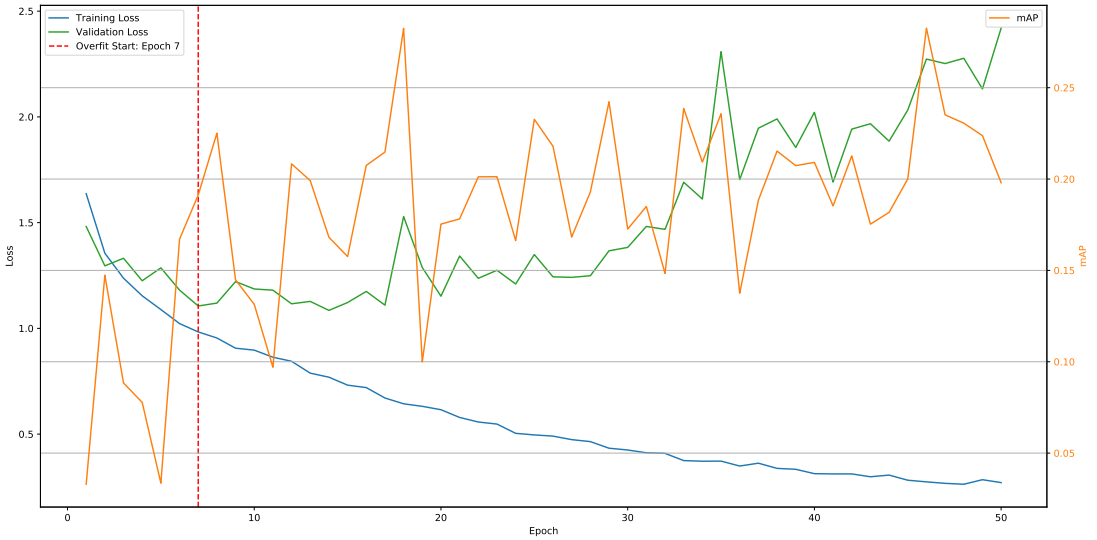


Figure A.6: Training and validation losses and mAP against epochs for Fold 0.

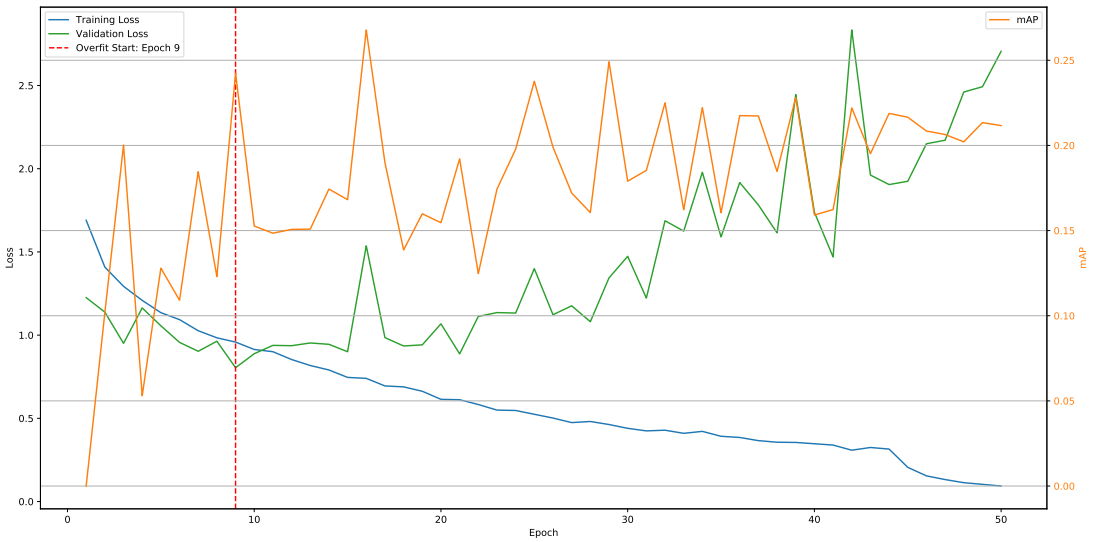


Figure A.7: Training and validation losses and mAP against epochs for Fold 1.

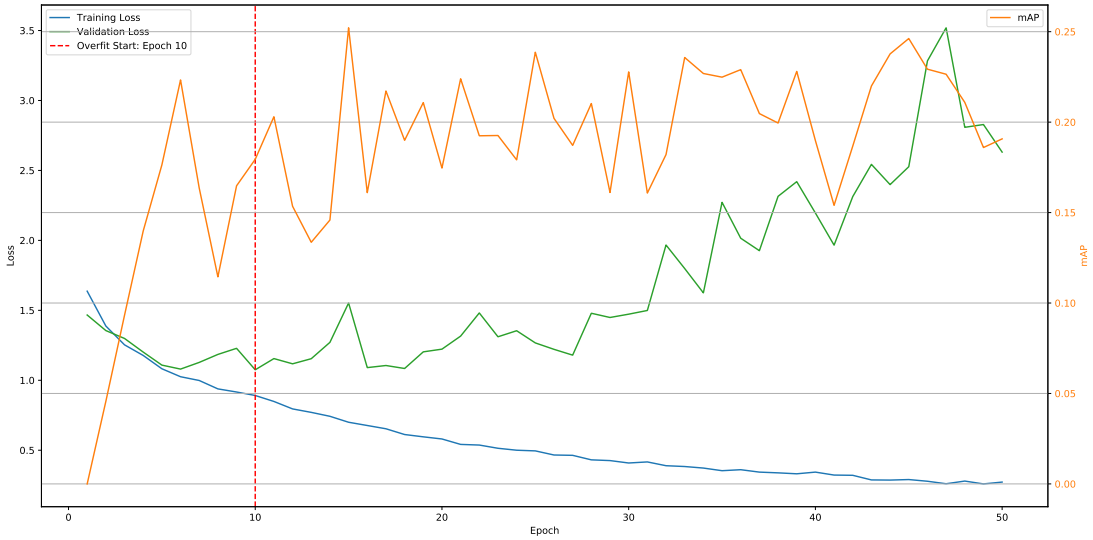


Figure A.8: Training and validation losses and mAP against epochs for Fold 2.

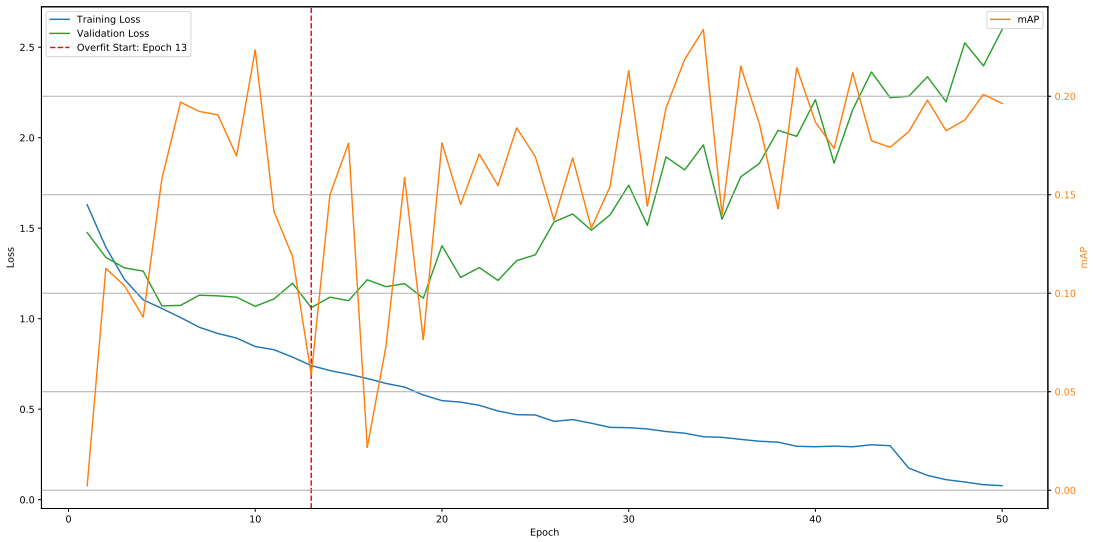


Figure A.9: Training and validation losses and mAP against epochs for Fold 3.

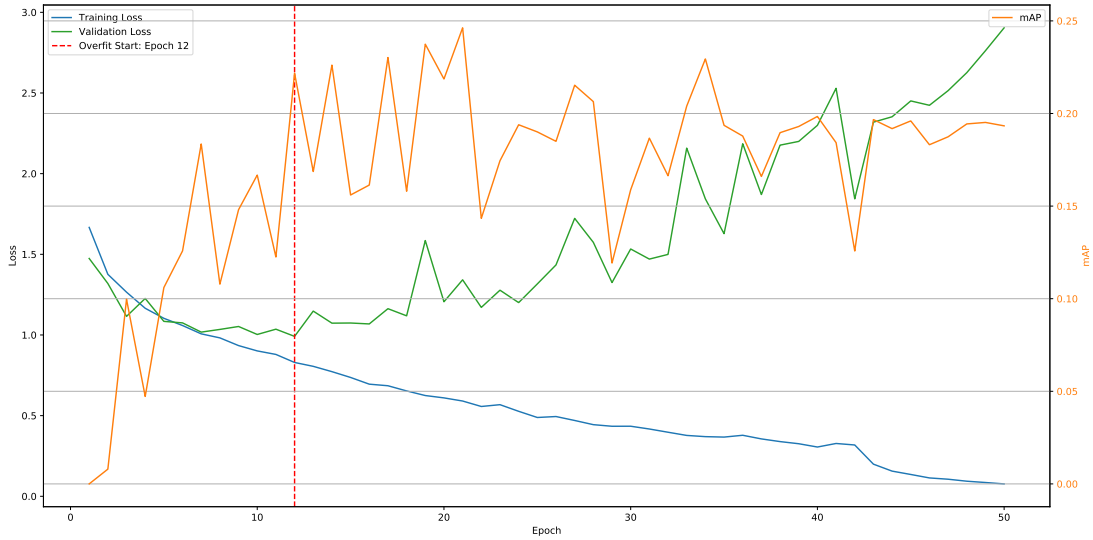


Figure A.10: Training and validation losses and mAP against epochs for Fold 4.

Appendix B

Yolo v3-Tiny

Plot note: The YOLO v3-Tiny ablation figures report the logged training optimisation loss together with validation mAP. The cross-validation figures reproduce the loss traces generated during each fold; where the plot axis uses "Average Loss", this denotes the optimisation loss recorded during training, while validation performance is represented by the mAP curve and fold-summary tables.

B.1 Ablation Study Plots

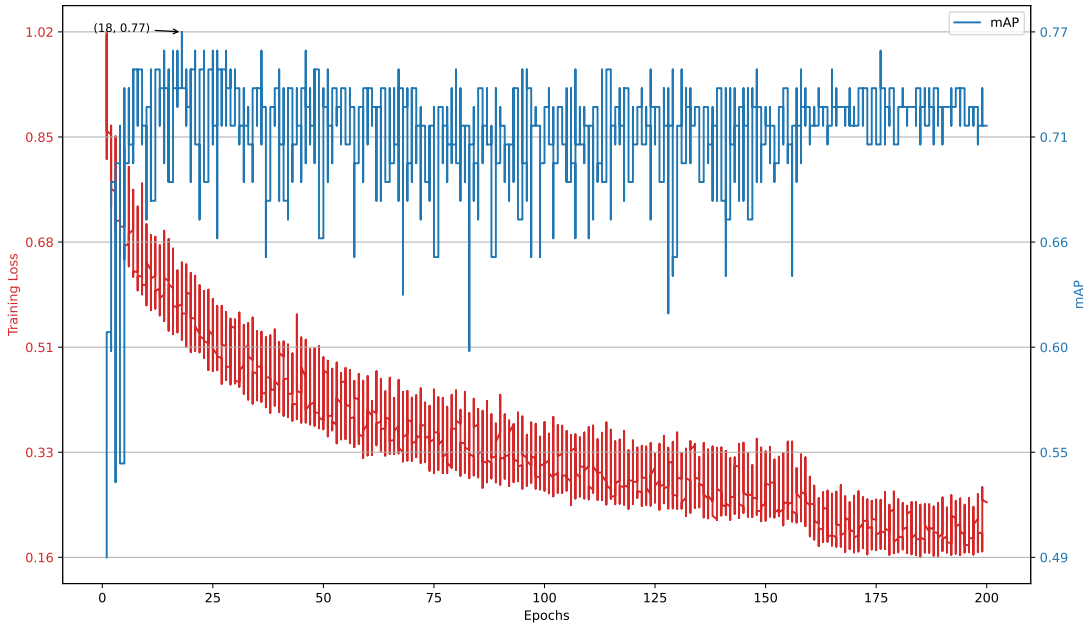


Figure B.1: Training losses and mAP against epochs for a learning rate of 1×10^{-3}.

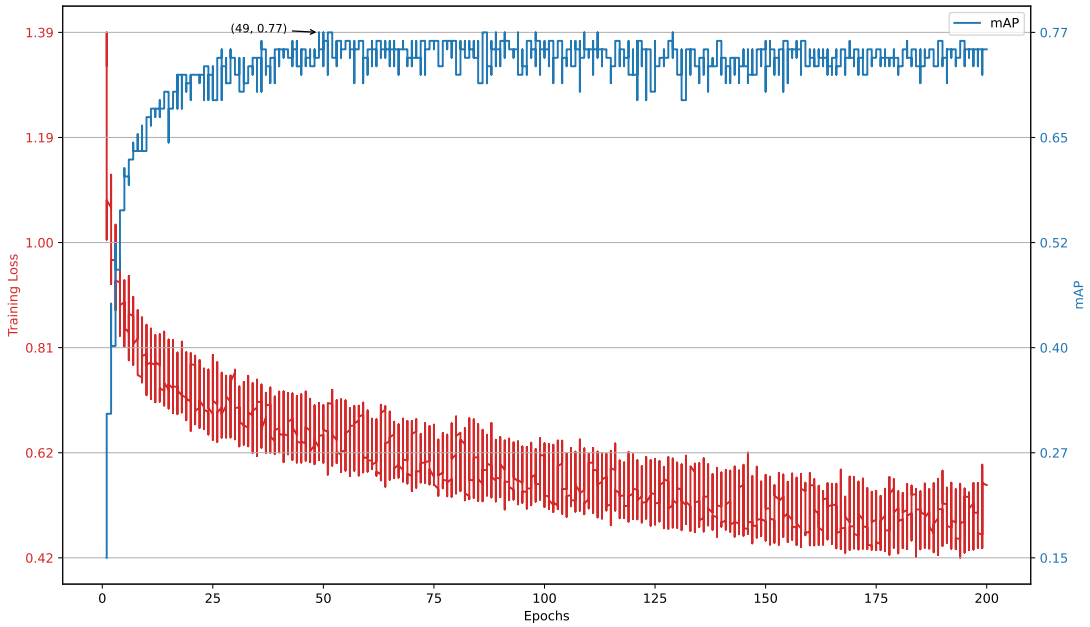


Figure B.2: Training losses and mAP against epochs for a learning rate of 1×10^{-4}.

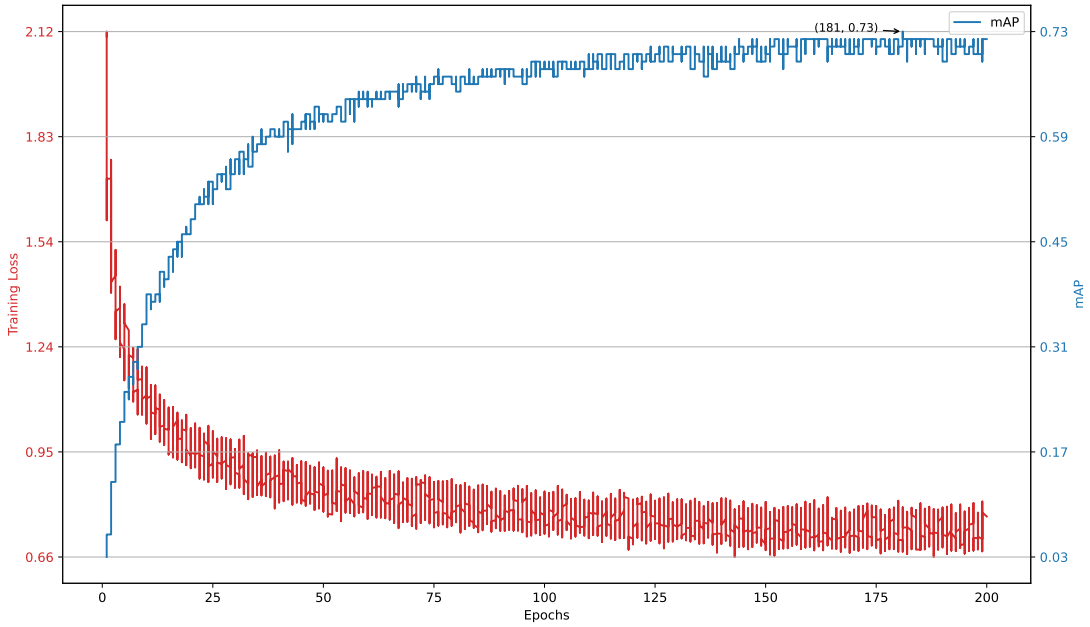


Figure B.3: Training losses and mAP against epochs for a learning rate of 1×10^{-5}.

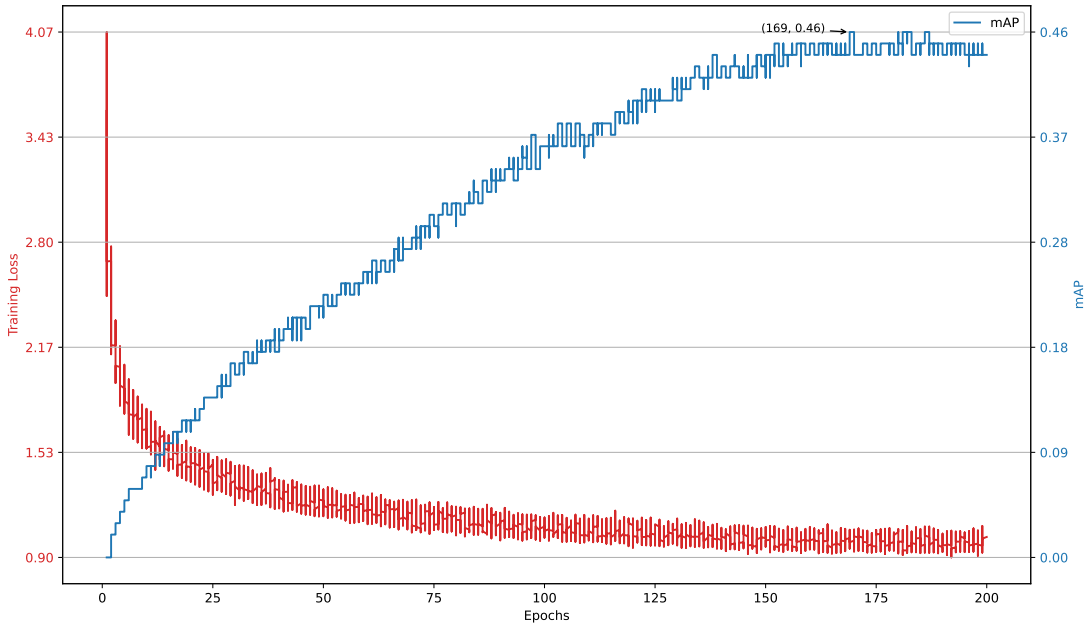


Figure B.4: Training losses and mAP against epochs for a learning rate of 1×10^{-6}.

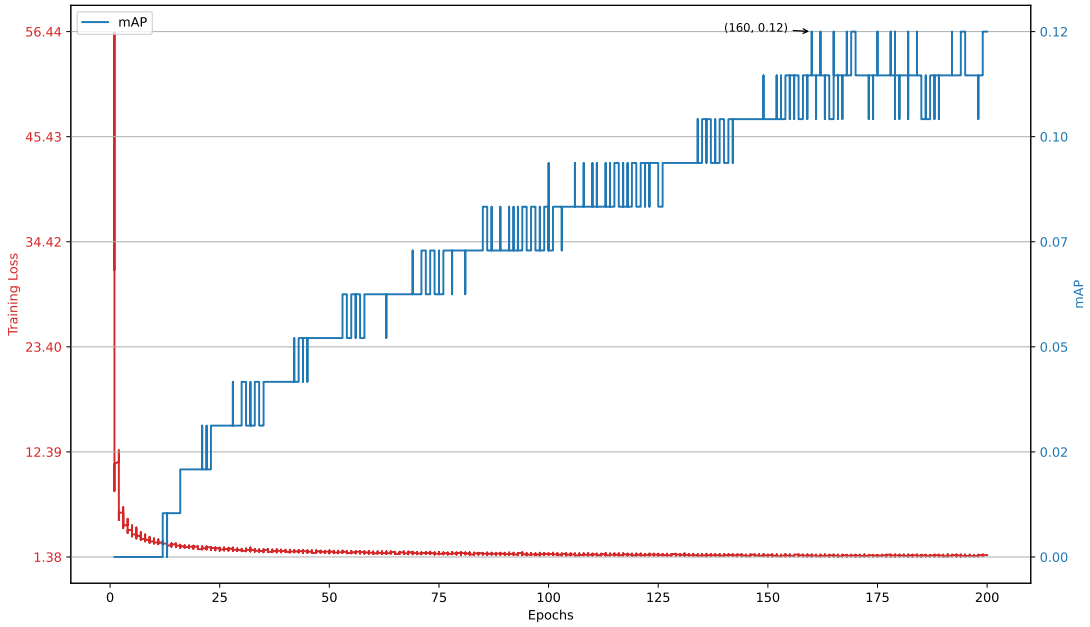


Figure B.5: Training losses and mAP against epochs for a learning rate of 1×10^{-7}.

B.2 Cross-Validation Plots

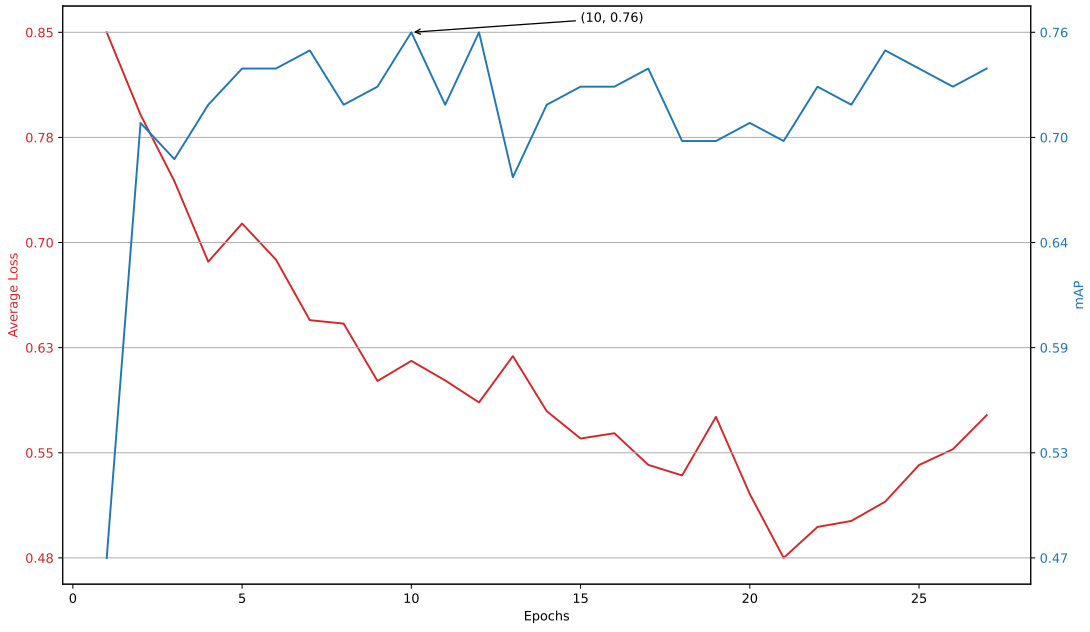


Figure B.6: Training and validation losses and mAP against epochs for Fold 0.

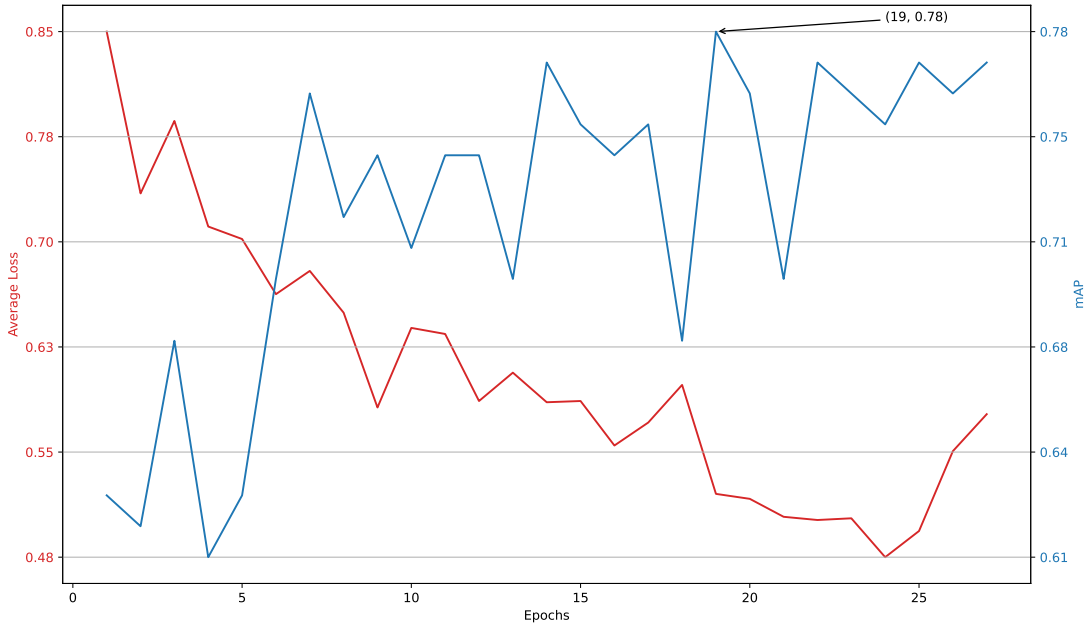


Figure B.7: Training and validation losses and mAP against epochs for Fold 1.

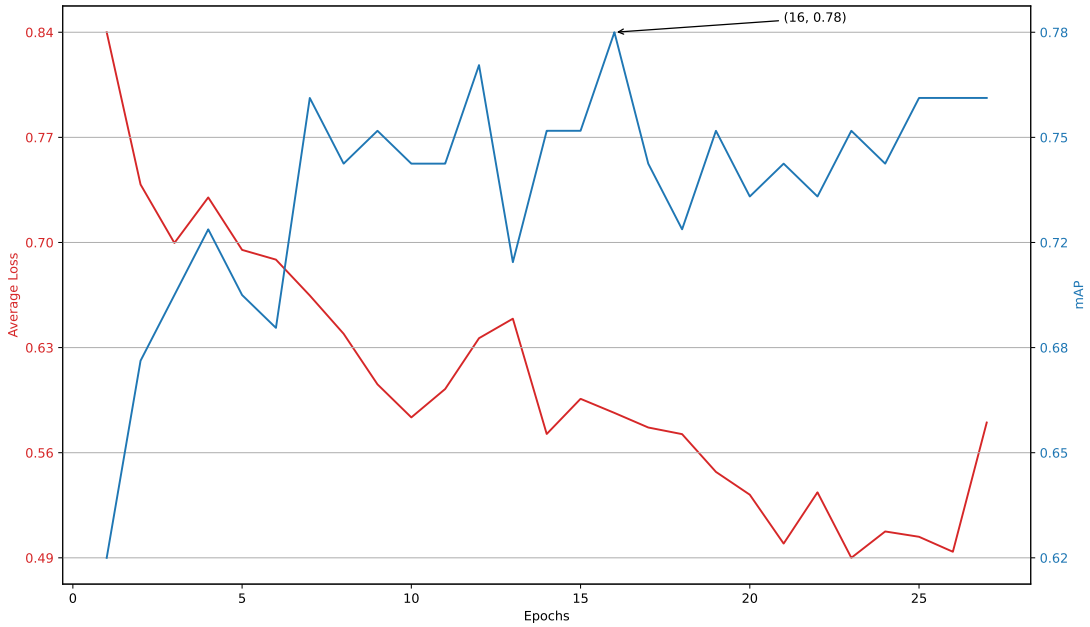


Figure B.8: Training and validation losses and mAP against epochs for Fold 2.

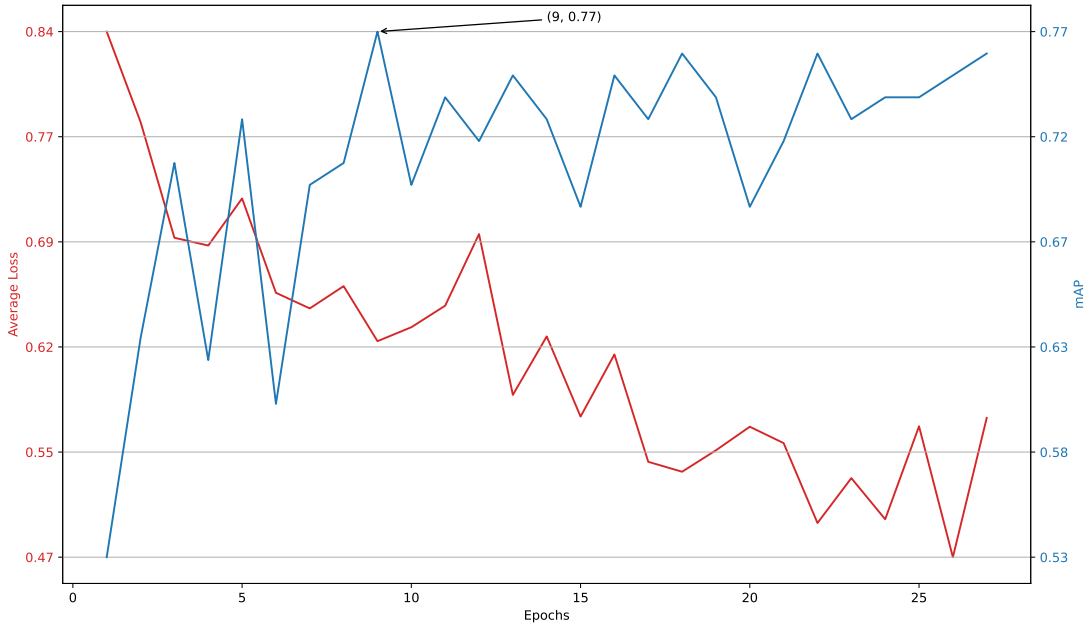


Figure B.9: Training and validation losses and mAP against epochs for Fold 3.

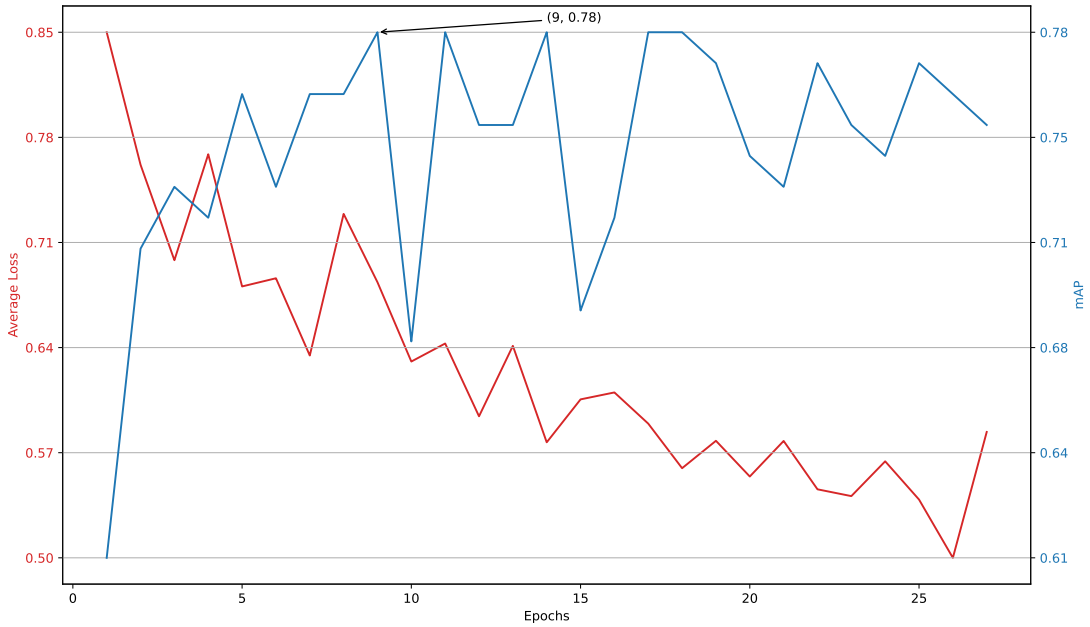


Figure B.10: Training and validation losses and mAP against epochs for Fold 4.

Appendix C

Faster RCNN

Plot note: The Faster R-CNN figures report the training optimisation loss generated during training and the validation mAP used for model selection. A separate validation-loss curve was not part of the diagnostic output for this training workflow, so validation performance is reported through the mAP curve and fold-summary tables rather than by an additional loss trace.

C.1 Ablation Study Plots

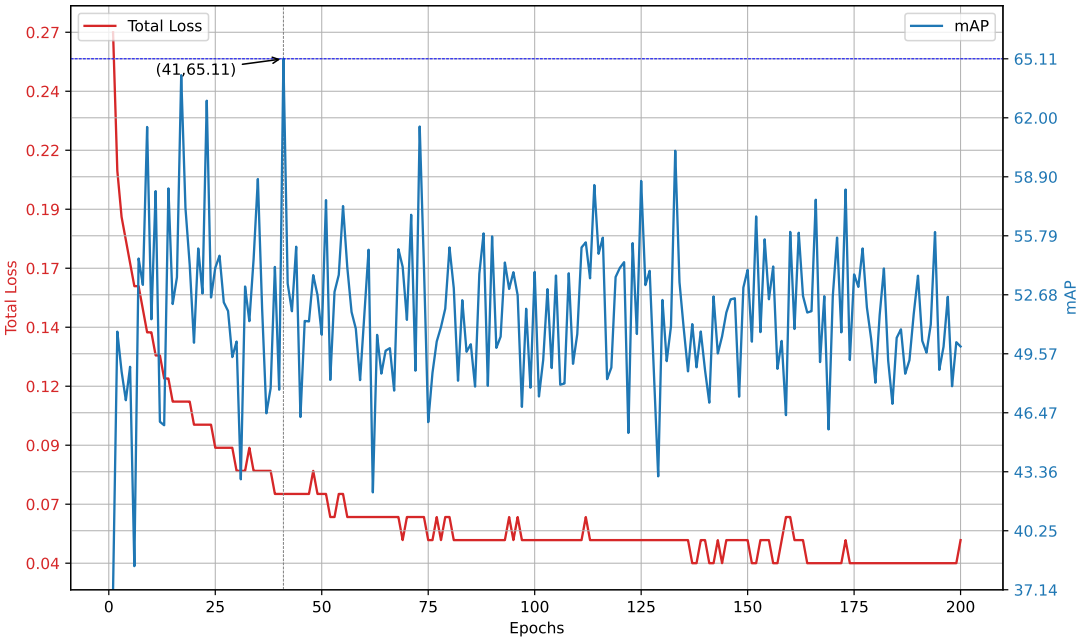


Figure C.1: Training losses and mAP against epochs for a learning rate of 1×10^{-3}.

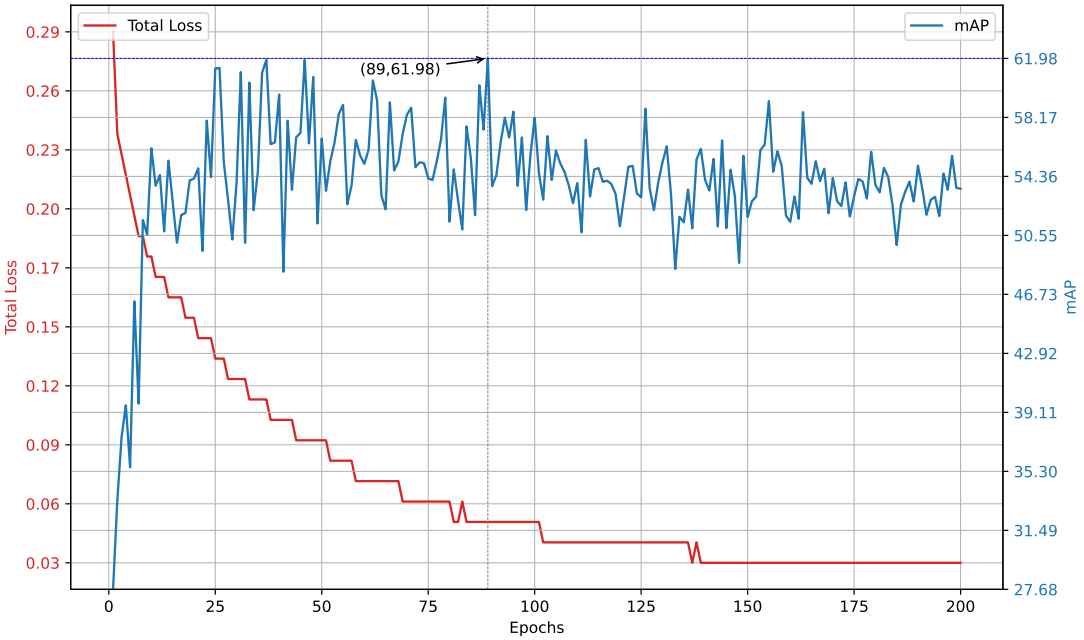


Figure C.2: Training losses and mAP against epochs for a learning rate of 1×10^{-4}.

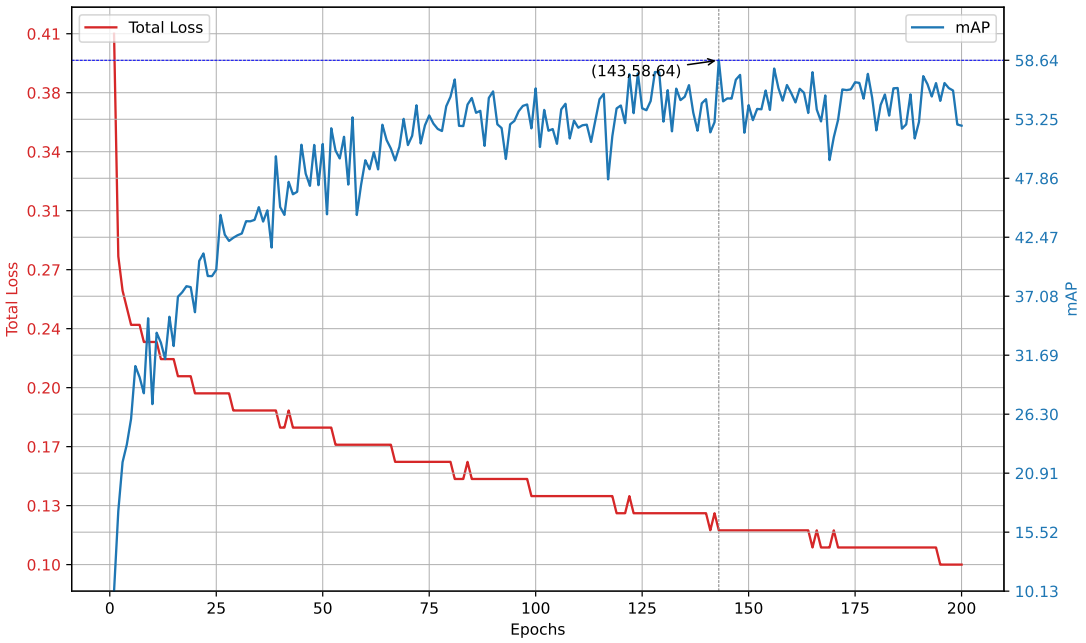


Figure C.3: Training losses and mAP against epochs for a learning rate of 1×10^{-5}.

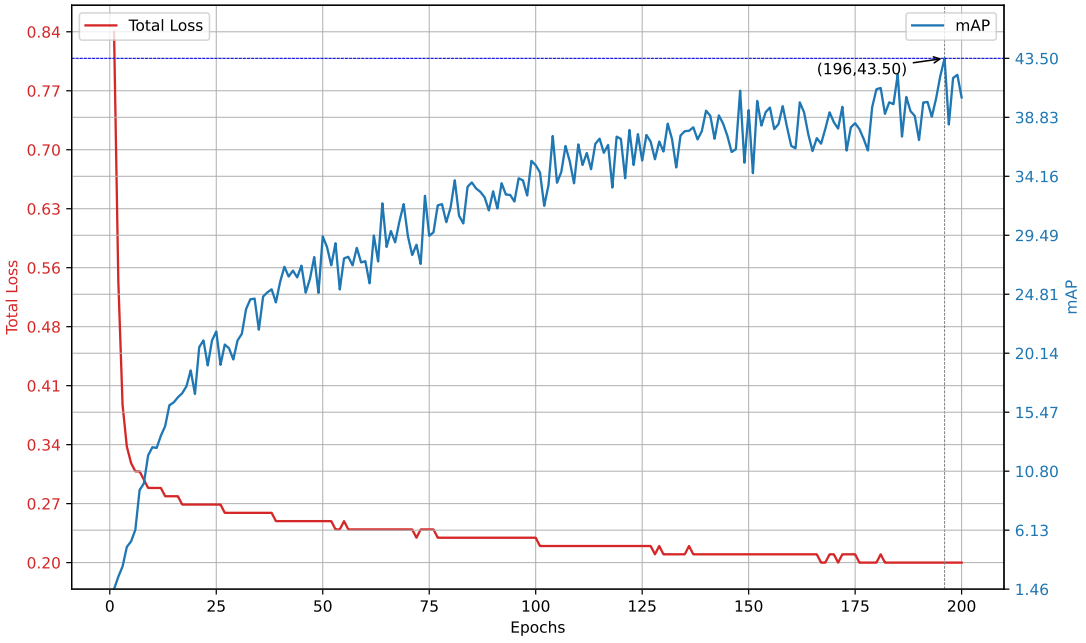


Figure C.4: Training losses and mAP against epochs for a learning rate of 1×10^{-6}.

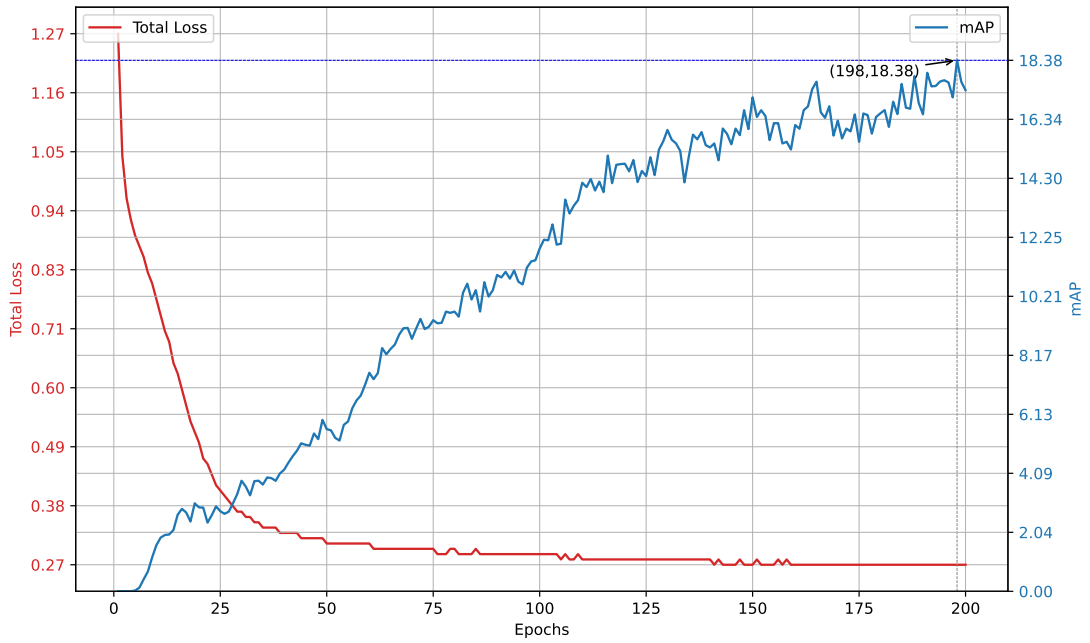


Figure C.5: Training losses and mAP against epochs for a learning rate of 1×10^{-7}.

C.2 Cross-Validation Plots

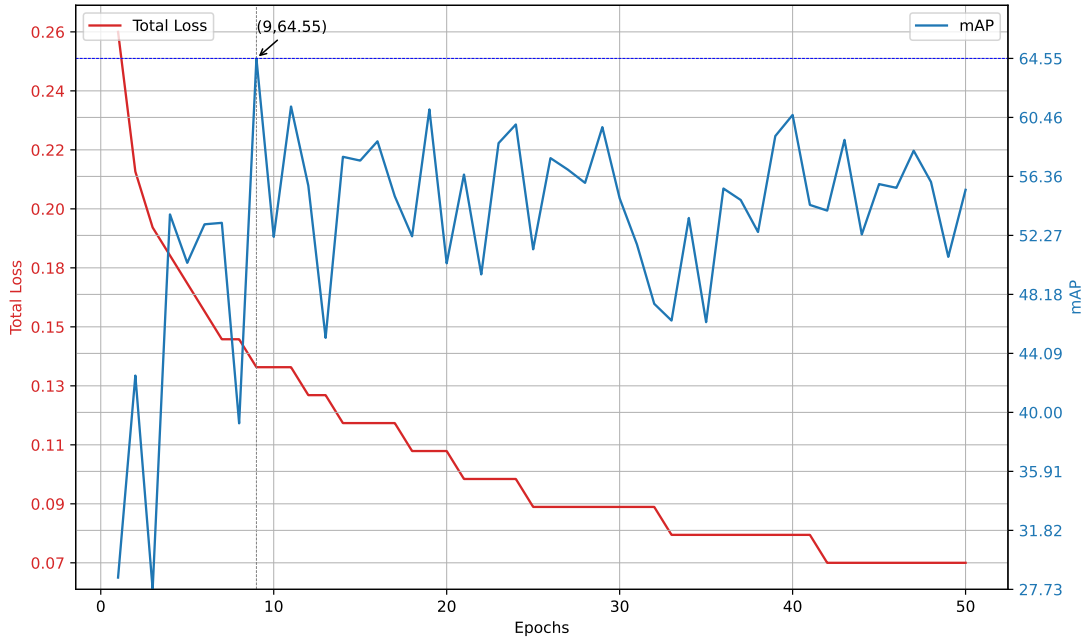


Figure C.6: Training loss and mAP against epochs for Faster RCNN Fold 0.

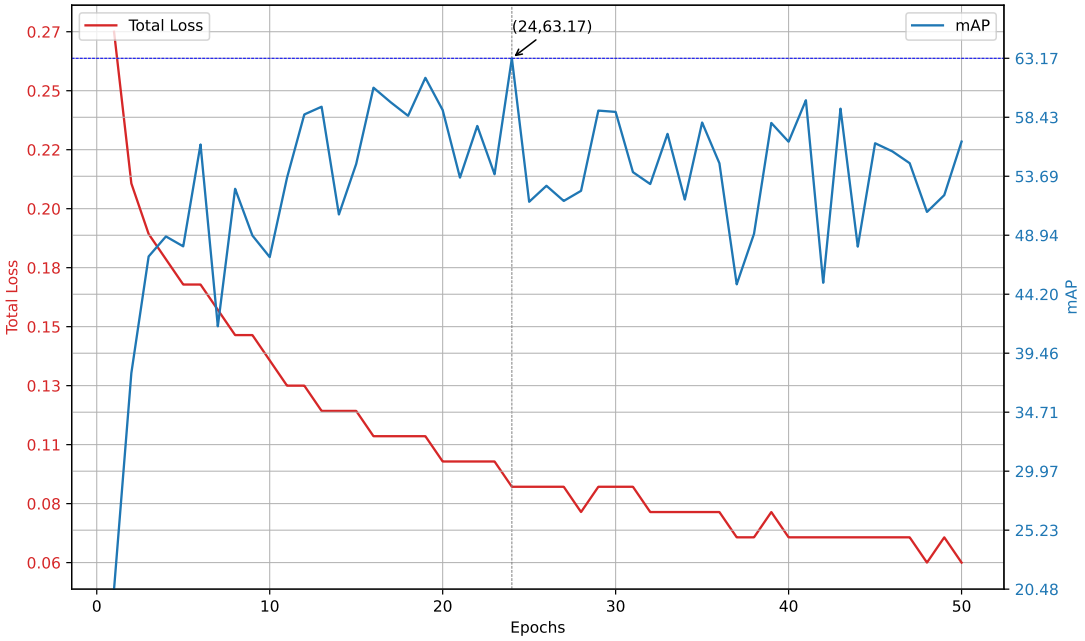


Figure C.7: Training loss and mAP against epochs for Faster RCNN Fold 1.

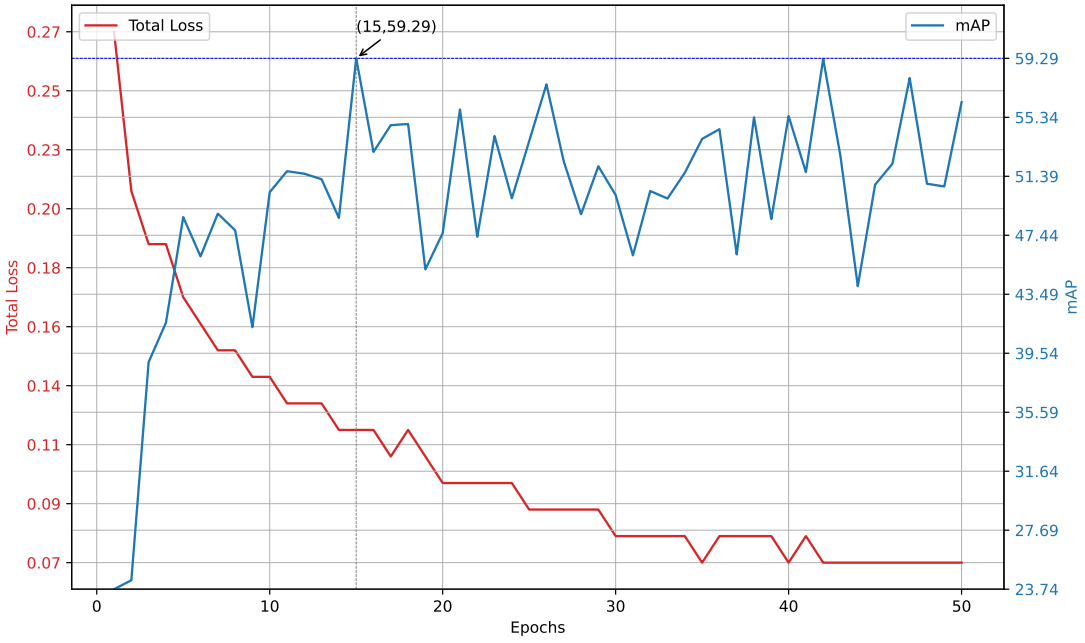


Figure C.8: Training loss and mAP against epochs for Faster RCNN Fold 2.

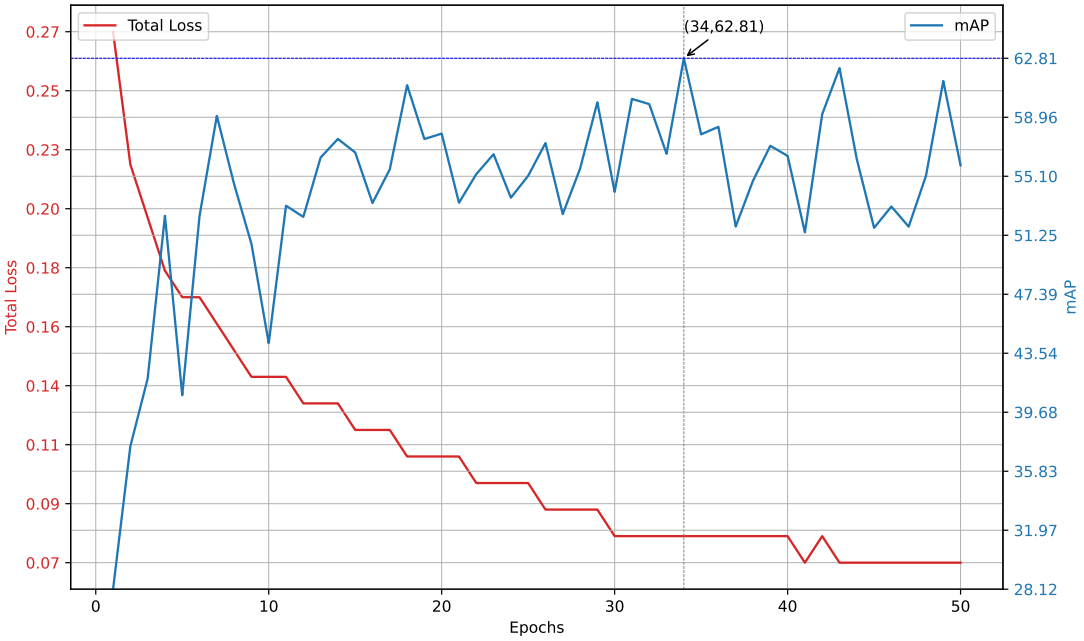


Figure C.9: Training loss and mAP against epochs for Faster RCNN Fold 3.

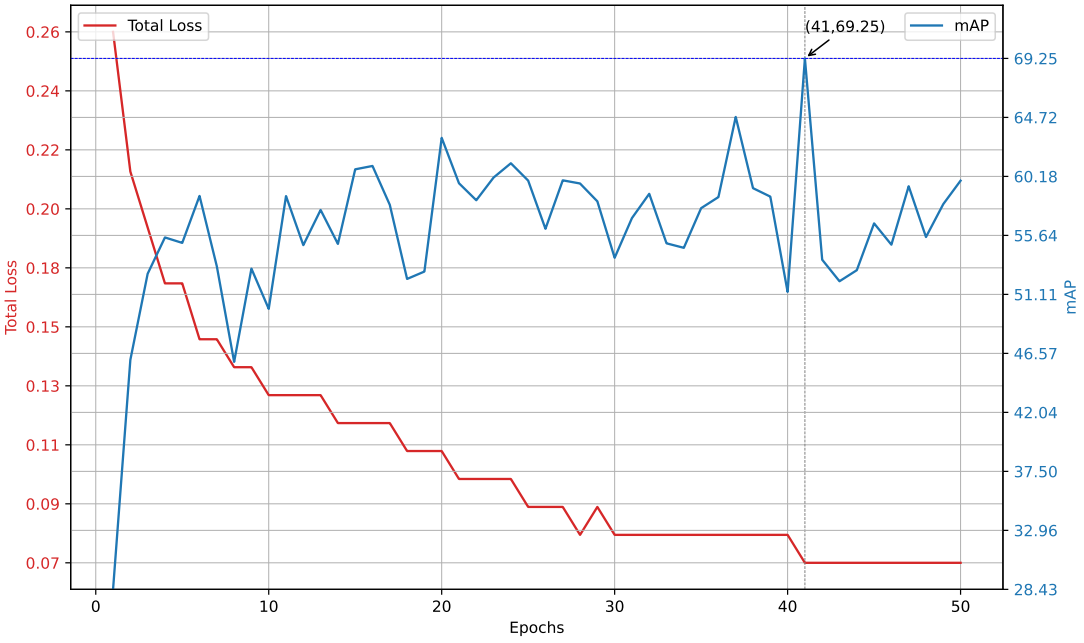


Figure C.10: Training loss and mAP against epochs for Faster RCNN Fold 4.

Appendix D

DETR

D.1 Cross-Validation Plots

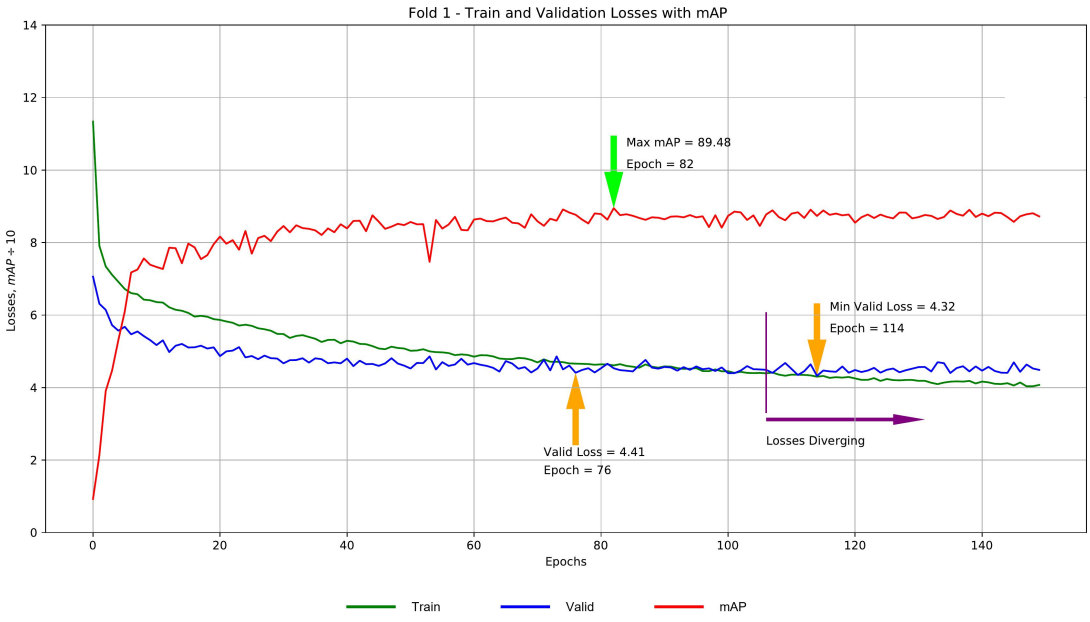


Figure D.1: Training loss and mAP against epochs for DETR Fold 1.

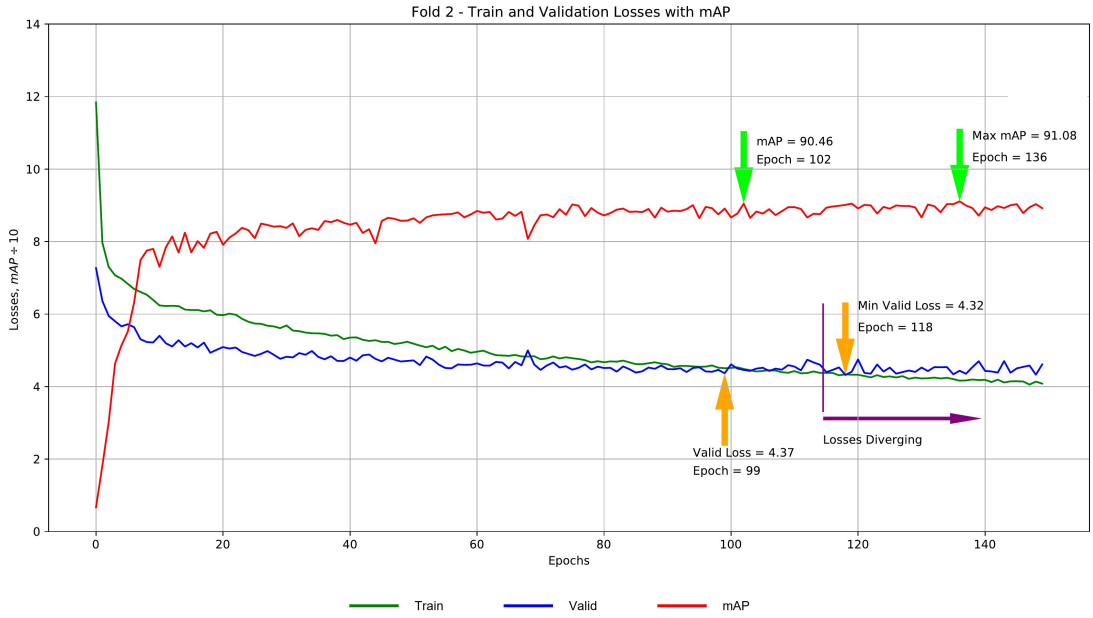


Figure D.2: Training loss and mAP against epochs for DETR Fold 2.

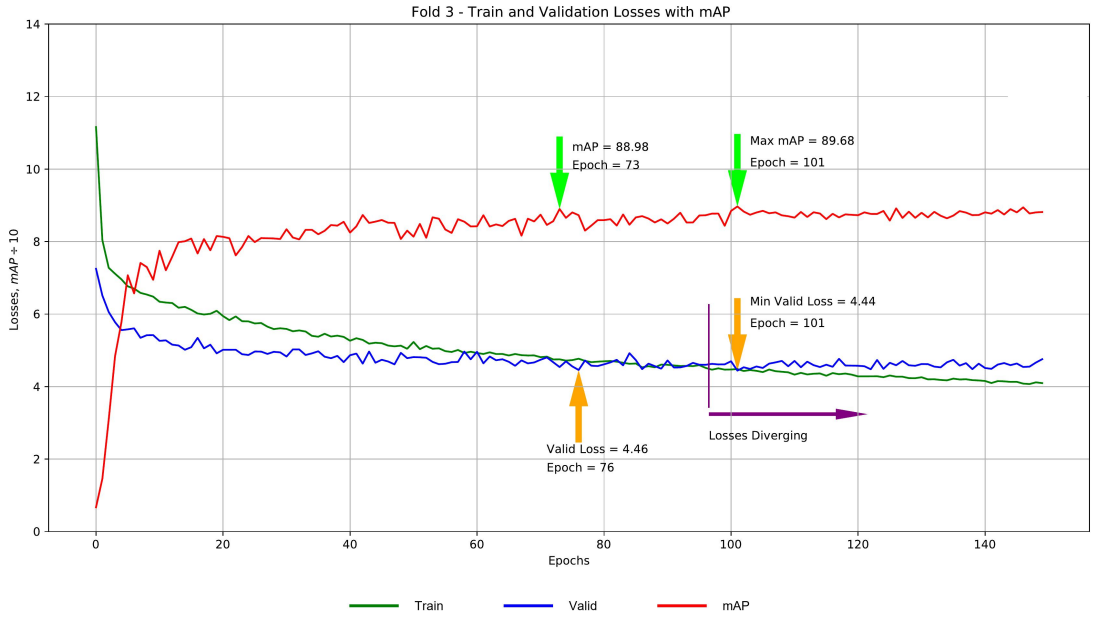


Figure D.3: Training loss and mAP against epochs for DETR Fold 3.

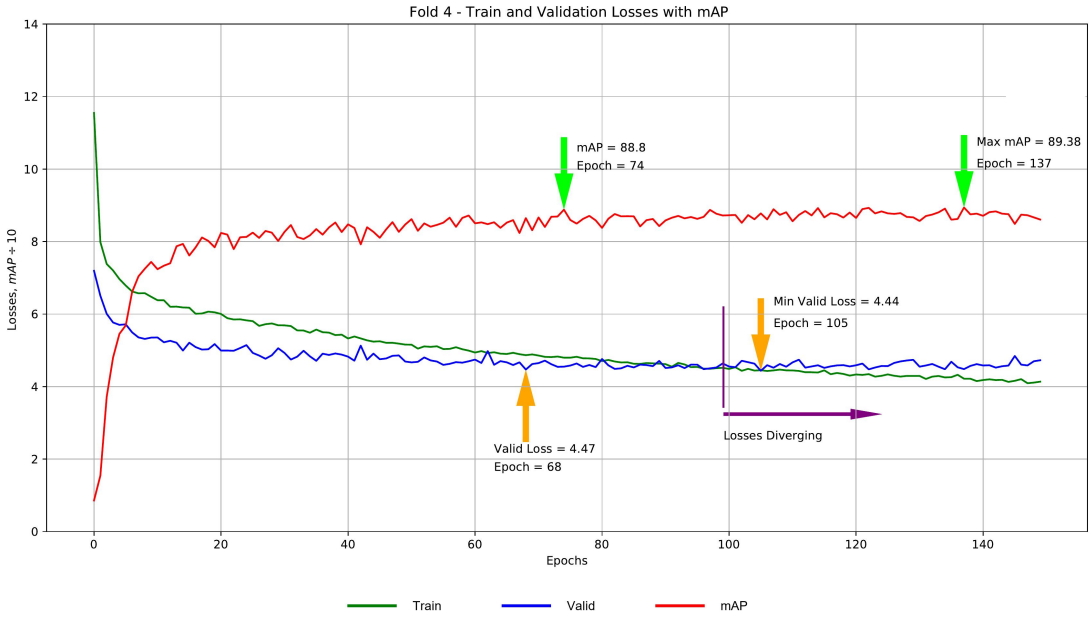


Figure D.4: Training loss and mAP against epochs for DETR Fold 4.

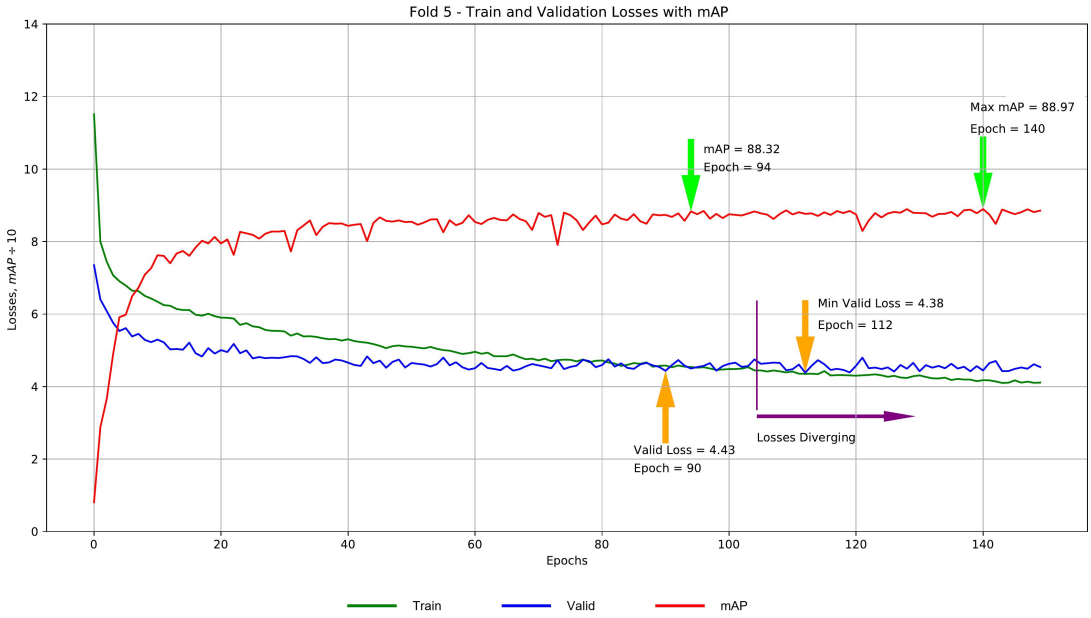


Figure D.5: Training loss and mAP against epochs for DETR Fold 5.

Appendix E

Code

E.1 GSV Image Retrieval - From Section 3.2

Listing E.1: Python script to download Google Street View images at multiple headings

```
import googlemaps
import csv
import os
import requests

gmaps = googlemaps.Client(key='YOUR_API_KEY_HERE')

def download_street_view_images(lat, lng, location_name, save_directory="images"):
    if not os.path.exists(save_directory):
        os.makedirs(save_directory)

    headings = [0, 90, 180, 270]

    for heading in headings:
        url = (
            "https://maps.googleapis.com/maps/api/streetview"
```

```
            f"?size=640x640&location={lat},{lng}"
            f"&heading={heading}&fov=90&pitch=0"
            f"&key=YOUR_API_KEY_HERE"
        )
        image_filename = f"{location_name}_{heading}.jpg"
        image_path = os.path.join(save_directory, image_filename)

        response = requests.get(url)
        if response.status_code == 200:
            with open(image_path, 'wb') as file:
                file.write(response.content)
            print(f"Image saved: {image_path}")
        else:
            print(f"Failed to download image for heading {heading} at ({lat}, {lng})")

def process_coordinates_csv(csv_file):
    with open(csv_file, newline='') as file:
        reader = csv.DictReader(file)
        for row in reader:
            lat = row['latitude']
            lng = row['longitude']
            location_name = row.get('name', f"location_{lat}_{lng}")
            download_street_view_images(lat, lng, location_name)

process_coordinates_csv("coordinates.csv")
```

The function `download_street_view_images` constructs the appropriate URL with parameters such as image resolution, field of view (fov), and heading (direction of the camera). The resulting images are stored in a designated folder, which the script creates if it does not already exist.

Finally, the script's `process_coordinates_csv` function reads the CSV file containing the coordinates and calls the download function for each location. This automated process significantly streamlines the collection of imagery, enabling

efficient preparation of the dataset for training the deep learning model.