Enhancing Licence Plate Recognition for a Robust Vehicle Re-Identification System

Submitted in fulfilment

of the requirements for the degree of

Master of Science

of Rhodes University

Alden Boby

Grahamstown, South Africa

April 7, 2024

Declaration of Authorship

I, Alden Boby, declare that Enhancing Licence Plate Recognition for a Robust Vehicle Re-Identification System is my own work. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- This thesis has not been submitted, in whole or in part, for any degree in any other University.
- Where I have consulted the published work of others, this is always clearly attributed.
- I have indicated by reference and acknowledgement the areas that are not my own work. Except for such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have clarified what others did and what I have contributed to myself.

Abstract

Vehicle security is a growing concern for citizens of South Africa. Law enforcement relies on reports and security camera footage for vehicle identification but struggles to match the increasing number of carjacking incidents and low vehicle recovery rates. Security camera footage offers an accessible means to identify stolen vehicles, yet it often poses hurdles like anamorphic plates and low resolution. Furthermore, depending on human operators proves inefficient, requiring faster processes to improve vehicle recovery rates and trust in law enforcement. The integration of deep learning has revolutionised object detection algorithms, increasing the popularity of vehicle tracking for security purposes.

This thesis investigates advanced deep-learning methods for a comprehensive vehicle search and re-identification system. It enhances YOLOv7's algorithmic capabilities and employs preprocessing techniques like super-resolution and perspective correction via the Improved Warped Planar Object Detection network for more effective licence plate optical character recognition. Key contributions include a specifically annotated dataset for training object detection models, an optical character recognition model based on YOLOv7, and a method for identifying vehicles in unrestricted data.

The system detected rectangular and square licence plates without prior shape knowledge, achieving a 98.7% character recognition rate compared to 95.31% in related work. Moreover, it outperformed traditional optical character recognition by 28.25% and deep-learning EasyOCR by 14.18%. Its potential applications in law enforcement, traffic management, and parking systems can improve surveillance and security through automation.

ACM Computing Classification System

Thesis classification under the ACM Computing Classification System[1] (2012 version valid through 2024):

- **Computing methodologies ∼ Artificial intelligence ∼ Computer vision ∼ Computer vision problems ∼ Object identification**
- Computing methodologies ∼ Artificial intelligence ∼ Computer vision ∼ Computer vision problems ∼ Tracking

General-Terms: Vehicle Security, Deep Learning, Object Detection, Object Identification, Object Tracking, YOLOv7, Licence Plate Character Recognition

[1] https://www.acm.org/publications/class-2012

Acknowledgements

The authors would like to thank the authors of the publicly available datasets used in this paper. This work was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA and Rhodes University. The authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author(s) and that none of the above mentioned sponsors accept liability whatsoever in this regard.

I would also like to acknowledge all those who directly and indirectly supported me through my work. In particular, I would like to extend my acknowledgements to my supervisor, Professor Dane Brown, who has been an exceptional and attentive supervisor and companion. The countless hours you have dedicated to your students are seldom found at this level of education. I would also like to thank my co-supervisor, Mr James Connan, for his input and inspiring intelligence.

Lastly, I would like to thank my parents, Brian and Grace Boby, and my brothers and friends for the emotional support they have provided throughout my academic journey. A special mention to Marc Marais and Luxolo Kuhlane, my peers who maintained a supportive and conducive environment, in and out of the academic space. Without all the mentioned persons involved in this process, this project would not have been possible.

Contents

List of Figures

List of Tables

Listings

Declaration of Publications

The following research papers have been accepted and presented for publication, and parts of its materials are included in the thesis:

1. **A. Boby**, D. Brown and J. Connan. Improving licence plate detection using generative adversarial networks. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, pages 588–601. Springer International, 2022.

2. **A. Boby**, D. Brown, J. Connan and M. Marais. Investigating the Effects of Image Correction Through Affine Transformations on Licence Plate Recognition. In *International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–6. IEEE, 2022.

3. **A. Boby**, D. Brown and J. Connan, M. Marais and L. L. Kuhlane. Licence Plate-Specific Character Recognition Using YOLO. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*. 2022.

4. **A. Boby**, D. Brown and J. Connan, M. Marais. Exploring the Incremental Improvements of YOLOv7 Over YOLOv5 for Character Recognition. In *International Advanced Computing Conference (IACC)*, pages 50–65. Springer Nature Switzerland, 2022.

5. **A. Boby**, D. Brown and J. Connan. Iterative Refinement Versus Generative Adversarial Networks for Super-Resolution Towards Licence Plate Detection. In *Inventive Systems and Control: Proceedings of ICISC 2023*, pages 349–362. Springer Nature Singapore, 2023.

6. **A. Boby**, D. Brown and J. Connan. A Practical Use for AI-Generated Images. In *International Conference on Information, Communication and Computing Technology*, pages 157–168. Springer Nature Switzerland, 2023.

7. **A. Boby**, D. Brown and J. Connan, M. Marais and L. L. Kuhlane. Enabling Vehicle Search Through Robust Licence Plate Detection. In *International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–7. IEEE, 2023.

8. **A. Boby**, D. Brown and J. Connan, M. Marais and L. L. Kuhlane. Cross-Data Vehicle Retrieval Through Robust Licence Plate Detection. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*. 2023.

CCTV	Closed-Circuit Television
CNN	Convolutional Neural Network
DeepSORT	Simple Online and Realtime Tracking with a Deep Association Metric
DiffBIR	Blind Image Restoration with Diffusion Prior
E-ELAN	Extended Efficient Layer Aggregation Network
ESRGAN	Enhanced Super-Resolution Generative Adversarial Network
ERQA	Edge Restoration Quality Assessment
FPS	Frames per Second
GAN	Generative Adversarial Network
IWPOD-NET	Improved Warped Planar Object Detection Network
IoU	Intersection over Union
LPIPS	Learned Perceptual Image Patch Similarity
LPR	Licence Plate Recognition
mAP	Mean Average Precision
MOSI-LPD	Multi-Oriented and Scale-Invariant Licence Plate Detection
MSE	Mean Squared Error
NMS	Non-Maximum Suppression
OCR	Optical Character Recognition
PROVID	Progressive Vehicle Re-Identification
PSNR	Peak-Signal-to-Noise Ratio
R-CNN	Region-Based Convolutional Neural Networks
ROI	Region of Interest
RRDB	Residual-in-Residual Dense Block
Re-Id	Re-Identification
SANRAL	South African National Roads Agency Ltd
SRCNN	Super-Resolution Convolutional Neural Network
SRGAN	Super-Resolution Generative Adversarial Network
SSIM	Structural Similarity Index
YOLO	You Only Look Once

1

Introduction

1.1 Re-Identification and Licence Plate Recognition

In recent years, machine learning and deep learning have been applied to enhance the capabilities of computer vision tasks. The initial method for object detection demanded substantial expertise, relying on manually crafted features and classifiers. This approach offered limited adaptability to unseen data due to its focus on local, low-level characteristics (Han *et al.*, 2019). Image processing solutions are frequently designed to function within familiar, researched, or predefined environments, limiting their ability to handle complex and varied object appearances. On the other hand, machine learning algorithms have gained prominence primarily because of their ability to generalise problems. Therefore, the complexity inherent in diverse vehicle appearances and their licence plates makes it challenging for image processing systems to achieve the efficiency and accuracy levels exhibited by machine learning approaches (Kalake *et al.*, 2022).

Vehicle Re-Identification (Re-Id) and Licence Plate Recognition (LPR) represent subdomains within computer vision that can greatly benefit from integrating deep learning solutions, particularly in the realm of object detection (Al-Batat *et al.*, 2022). Object detection has been significantly advanced by the learning capabilities of Convolutional Neural Networks (CNNs), which can process large amounts of data and extract relevant features for vehicle identification — enabling the utilisation of vehicle-specific attributes for both vehicle and licence plate classification tasks (Björklund *et al.*, 2019). Object detection stands out as one of the most common tasks within the realm of machine learning-based algorithms in literature. The current methods revolve around refining and enhancing established deep learning approaches to create viable solutions, overshadowing traditional machine learning and image processing-based methods of the past.

Specialised CNNs, such as YOLO (You Only Look Once) and Faster R-CNN (Region-Based CNN), can detect objects in real-time (Redmon *et al.*, 2015). Adapting these systems to the target application can benefit surveillance and traffic regulation, especially when leveraging existing security cameras. When appropriately configured with relevant data, these models demonstrate effectiveness in vehicle detection and tracking, which are prerequisites for vehicle Re-Id (de Oliveira *et al.*, 2019). Coupled with a primary reliance on LPR for vehicle identification, vehicle Re-Id can be realised. Present-day LPR systems perform well with constrained data, including cameras positioned for optimal capture conditions. Good ambient lighting and a controlled object perspective create an ideal environment for LPR. However, real-world footage includes challenges such as low-light conditions, motion blur, occlusion and self-occlusion, all of which can impact the performance of object detection systems. Through the use of vast amounts of carefully selected data and advanced machine learning techniques such as Generative Adversarial Networks (GANs) and specialised CNNs, the performance of such systems can be enhanced (Lee *et al.*, 2019, Kim *et al.*, 2021).

The risk of car theft can be mitigated by integrating technology into daily operations requiring vehicle Re-Id (Henry *et al.*, 2020). An automated system would alleviate the workload of human operators, potentially eliminating human error. These systems can identify stolen vehicles or traffic offenders and flag them, storing their licence plates for Re-Id in video footage or stream. Advancing LPR contributes to the adoption of intelligent transport systems worldwide, increasing the efficiency of law enforcement and traffic regulation by utilising state-of-the-art technology.

1.2 Context of Research

In South Africa, vehicle security is a prevalent issue. Criminals continuously exploit vehicle vulnerabilities, highlighting the need for effective security measures (Rondganger, 2023). LPR is a non-trivial task involving several intricate stages, each requiring careful refinement, especially for correctly classifying characters. Extracted data from a licence

plate only transforms into meaningful information upon identification of all its characters. Therefore, an incorrect character can invalidate an entire licence plate, highlighting the pivotal role of precise character classification within the system. Enhancing images through super-resolution and perspective control[1] stands as an effective approach to improving image quality before Optical Character Recognition (OCR), ensuring accurate character identification on a licence plate. Manual observation of 24-hour security camera footage is prone to human error, time-consuming and inefficient for human observers (Laroca *et al.*, 2021a, Kalake *et al.*, 2022). Leveraging existing infrastructure within intelligent transport systems presents an opportunity to strengthen security measures by automating vehicle Re-Id (Al-Batat *et al.*, 2022).

1.3 Motivation for this Research

Increasing carjacking rates cause concern amongst South African residents, especially with low vehicle recovery rates (BussinessTech, 2022). Traditional approaches to counter vehicle theft involve physical car trackers. However, these trackers can be easily deactivated or removed when in proximity to a vehicle, either by locating them or jamming their signals with specialised devices. Moreover, car trackers are optional and placed in vehicles based on the owner's preference. An alternative method that is less easily tampered with is required.

Security camera footage offers an accessible and persistent means of identifying stolen vehicles and traffic offenders, as the data is securely stored and tamper-resistant. Private data security protocols have become more robust with the implementation of the POPI Act[2]. The data can be stored offsite and duplicated, ensuring continuous availability. The infrastructure is readily available but is underutilised. To solve the shortcomings of physical car trackers, vehicle security can be improved by supplementing it with a security camera feed for vehicle identification with vehicle detection and LPR prior. The

[1] The deliberate adjustment or management of an image's perspective or view angle to ensure optimal clarity and accuracy in identifying and interpreting characters on the licence plate.

[2] Protection of Personal Information Act, South Africa's data protection law.

application of such systems is not limited to vehicle theft, as traffic primarily involves vehicles and large amounts of processed data.

Surveillance footage resolution is typically low to maintain reasonable storage costs, as it requires a 24/7 uninterrupted feed. A persistent problem in LPR systems involves data degradation, where variables such as illumination and occlusion result in "dirty data". Road camera lenses, in particular, are prone to data degradation as they operate in unconstrained environments. Licence plates serve as unique identifiers for registered vehicles, enabling their differentiation even when physical properties are shared. These alphanumeric strings can be searched in a database by an authorised party to access information about a vehicle, including ownership details and make/model. All characters on a licence plate need to be distinguishable for accurate recognition. Therefore, specific techniques are required to restore or enhance the clarity of data such that LPR systems can achieve high recall and precision rates. Perspective distortion further complicates LPR in real-world scenarios, exacerbating the data degradation problem and making OCR more challenging due to significant feature alterations such as shapes and sizes (Björklund *et al.*, 2019). Therefore, robust LPR requires several factors to be considered, including variations in licence plate formats, camera angles, lighting and weather/lens conditions.

To effectively leverage video footage for security purposes, a system adept at managing low-quality data constraints is required. This system aims to address vehicle Re-Id and tracking through LPR on varied data, such as security and dashboard footage.

1.4 Problem Statement

This research investigates the effectiveness of locating vehicles using their unique feature, the licence plate, as the first point of identification. However, unconstrained environments, video quality degradation, and obscure angles present a challenging scenario for LPR. The initial stages thus require detecting both licence plates and vehicles. If a licence plate is undetectable, the vehicle's body features, although common and shared across different models from the same manufacturer, provide an additional mode of information

which can enable improved vehicle identification or be used as a re-identification fall-back mechanism.

The hypothesis is that a system can be constructed to improve vehicle identification by supplementing it with accurate LPR, where deep learning methods play a significant role in the initial detection of a licence plate and the subsequent classification of individual licence plate characters. The resulting system functionality includes a fully-fledged vehicle Re-Id and retrieval system extracting information from video/image data.

1.5 Research Question

This research seeks to answer the overarching question: 'Can a real-time vehicle re-identification and retrieval system effectively localise and track vehicles primarily based on licence plate recognition?' The question can be divided into the sub-questions:

1. How can false negatives be eliminated when detecting licence plates in a scene?
2. Is there an effective way to reduce distortion in anamorphic licence plates?
3. Can an object detection model be modified to operate as a better optical character recognition model?
4. How can vehicle identification be optimised when only partial licence plate information is available?

1.6 Research Objectives

The primary aim is to effectively localise and recognise licence plates in unconstrained scenarios for re-identification by achieving the following objectives:

1. Collate datasets of vehicles and licence plates to train a system that recognises licence plates in real-world scenarios.

2. Identify an effective method for detecting licence plates at oblique angles.
3. Create a dataset for training a robust OCR model.
4. Successfully adapt an object detection model to detect and classify characters accurately.
5. Conduct an experiment to select a super-resolution model that improves character recognition accuracy based on indicators such as enhanced readability.
6. Identify a vehicle in footage through its licence plate.
7. Re-identify a vehicle in footage using its licence plate string as input for vehicle retrieval.

1.7 Approach

The summary approach to this thesis is as follows. Various techniques, including deep learning algorithms, image processing methods, and data collection approaches, are explored in the literature. The literature extensively explores models focusing on recent cutting-edge deep learning-based object detectors towards creating an accurate vehicle Re-Id and retrieval system using licence plates as a primary identifier. The most promising approaches are thus selected based on their effectiveness in accurately detecting licence plates and their feasibility for deployment in real-world scenarios.

The system's success is determined mainly by its capacity to capture and interpret characters on a licence plate accurately. Deep learning models are trained through a diverse set of data to enable improved proficiency in unconstrained data. Experiments are conducted to allow the system's performance to be measured using quantitative metrics and qualitative visual inspection, facilitating comparisons and evaluation.

1.8 Assumptions

Data collection for the experiments is conducted under the following assumptions:

- The licence plates in the data are distinguishable and verifiable by the human eye.
- The data is constrained to the Latin alphabet.

1.9 Limitations

- The appearance of an image is subjective and difficult to evaluate quantitatively. Existing metrics are common in literature but are indicators and do not fully quantify visual perception.
- Datasets for vehicle re-identification redact licence plates for ethical reasons, complicating the evaluation of the effects of LPR to supplement vehicle identification. The data available for this scenario is limited to specific samples that were ethically approved.
- The findings of this study apply exclusively to regions utilising the Latin alphabet. Characters from other scripts, such as hanzi[3], will be excluded due to insufficient interpretation and evaluation resources for such subsets.
- While some images can be recovered, others may experience occlusion or truncation, making it challenging to restore missing information with reasonable accuracy.
- As the aim is to use existing infrastructure, the solution leans heavily towards addressing the problem through software. Expensive depth cameras and dedicated hardware will thus not be used, limiting the selection to consumer-grade and more affordable hardware.

[3]Chinese characters.

- As the proposed LPR system is intended as a general solution, using the format of a region to increase the accuracy of OCR is limited to inherent unsupervised learning and not explicitly learned by the system.

1.10 Thesis Outline

The remainder of this thesis is arranged as follows:

Chapter 2: *Concepts and Literature Review*: An overview of literature focused on the prominence of object detection models in the LPR domain. The relevant concepts and effective approaches from related studies are evaluated to propose a new LPR system.

Chapter 3: *System Methodology*: This chapter introduces the proposed multi-stage approach, outlining the stages to achieve robust LPR based on knowledge from existing literature.

Chapter 4: *Implementation & Experimental Setup*: This chapter presents the implementation-specific details of the proposed LPR system and the data curated to train it, followed by the structure of the experiments and quantitative metrics used for performance analysis.

Chapters 5: *Results and Discussion*: The validation of the implemented system is presented, followed by the results achieved during testing with a comprehensive analysis and discussion.

Chapter 6: *Conclusion and Future Work*: This chapter highlights the contributions and discoveries made while concluding the thesis. It offers insights into areas that demand further focus and exploration in future work.

2

Concepts and Literature Review

This chapter details the issues on licence plate detection and explores advancements towards improving the accuracy of LPR systems. Particularly the use of deep learning to automate security surveillance, alleviate human operators' workload and enhance the quality of source images. Furthermore, the chapter highlights the challenges and limitations existing LPR systems face, providing insights into future directions for research in this field.

2.1 YOLO Object Detection

YOLO, a one-stage object detector, gained popularity due to its optimal balance between speed and accuracy (Wang *et al.*, 2022). It is the favoured option compared with R-CNN, and faster R-CNN (Diwan *et al.*, 2022). YOLO's high inference speed can be attributed to its efficient single-shot approach, which eliminates the need for multiple passes over an image to make detections (Miller *et al.*, 2022). YOLO's notable advantage lies in its ability to handle multiple objects within a single image simultaneously. With enough relevant data, it can be finetuned to detect virtually any object (Wang *et al.*, 2022). Hence, it is well-suited for applications like traffic monitoring and surveillance, where multiple objects of interest may appear in an image.

At a high level, YOLO divides an image into an $S \times S$ grid. Larger grid dimensions enable the detection of more objects. A confidence score is assigned to each cell, indicating the probability of an object's presence. If the centre of an object falls within a cell of the grid, the cell adopts the object class. Every bounding box B is associated with five predictions, the centre point coordinates (x,y), width and height (w,h) relative to the

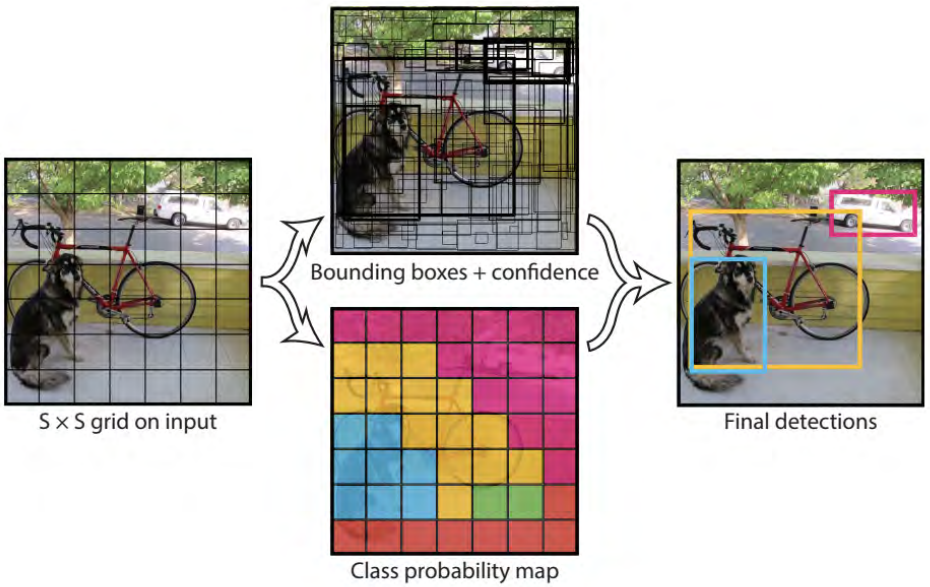


Figure 2.1: An image is divided into an $S \times S$ grid, B bounding boxes are predicted for each cell, confidence scores and C class probabilities. Resulting in a tensor with the shape $S \times S \times (B \times 5 + C)$ (Redmon *et al.*, 2015).

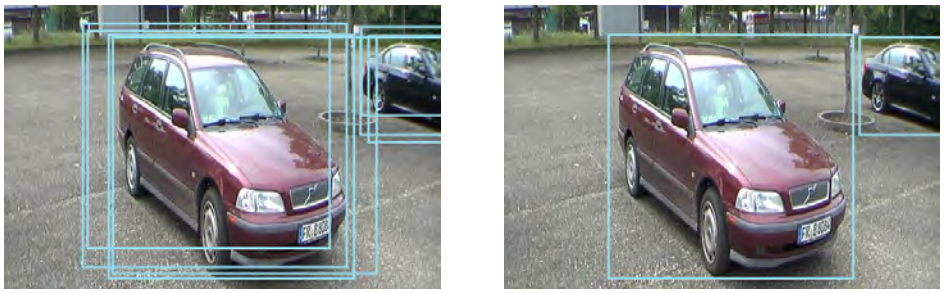
input image size and a confidence score. Lastly, each cell in the grid has an associated class probability C, obtained by a softmax activation function, resulting in a tensor with the shape $S \times S \times (B \times 5 + C)$ (Redmon *et al.*, 2015). An illustration of the YOLO algorithm can be found in Figure 2.1. YOLO accommodates multiple bounding boxes to handle duplicate detections; however, this results in potentially overlapping bounding boxes, which are undesirable. Non-Maximum Suppression (NMS) removes redundant bounding boxes by selecting the most confident box prediction per object.

2.1.1 Non-Maximum Suppression

Due to how YOLO processes images, proposals may include multiple detections for the same object (Bodla *et al.*, 2017). NMS is an algorithm used in the post-processing stage of the YOLO model to refine detections. It selects the bounding box with the highest confidence value and calculates the Intersection over Union (IoU) against overlapping bounding boxes. IoU represents the intersection between a ground truth bounding box and a predicted bounding box. A greater area of intersection signifies a more accurate pre-

diction from the model, shown visually in Figure 2.2. A threshold value for IoU controls the strength of NMS as use cases may require detecting objects very close to each other. Figure 2.3 shows detections before and after applying NMS. Standard NMS suppresses only overlapping bounding boxes belonging to the same class, allowing detection of different object classes in the same region. Alternatively, class-agnostic NMS considers the IoU across classes and prevents duplicate detections in the same region. The applications of these NMS methods are use case specific (Gomes *et al.*, 2022).

Figure 2.2: Different levels of bounding box accuracies.



(a) **(b)**

Figure 2.3: Multiple predictions for a single object are present before NMS and are reduced to one after the algorithm is applied.

2.1.2 Evaluation

Object detectors are commonly evaluated using Mean Average Precision (mAP) with IoU, which are metrics intended for comparing the performance of existing CNN-based solutions (Du, 2018).

Average precision measures the precision and recall trade-off over a range of values with a consistent interval (Casas *et al.*, 2023). This measure calculates the area under a precision-recall graph, as provided in Equation 2.1. The mAP metric aggregates average precision scores across classes, serving as a holistic indicator of the YOLO model's performance, where higher values indicate superior object detection capabilities.

$$AP = \frac{1}{11} \sum_{\text{Recall}_i} Precision\,(\,Recall_i) \tag{2.1}$$

The mAP metric in YOLO commonly exists in two versions: mAP@0.5 and mAP@0.5-0.95. These variations gauge performance by considering the complexity of labelled detections during training. While mAP@0.5 covers a single IoU threshold, mAP@0.5-0.95 assesses more challenging detections, typically in increments of 0.05, across a range from 0.5 to 0.95.

2.1.3 Revisions

YOLO has undergone various architectural refinements, resulting in multiple versions (Wang *et al.*, 2022). This subsection delves into several notable iterations of YOLO; however, it focuses solely on versions accompanied by seminal academic papers.

The initial version of YOLO was based on GoogLeNeT, replacing the inception modules with 1×1 reduction layers followed by 3×3 convolutional layers (Redmon *et al.*, 2015). Modelling object detection as a regression problem proved successful, introducing a single-shot object detector with a high inference speed with minimal compromises to accuracy. However, this version was limited by reduced performance when detecting small objects grouped and an inability to generalise data including varied aspect ratios.

The second iteration, YOLO9000 (YOLOv2), had been extended to detect more than 9000 classes (Redmon and Farhadi, 2017) — the newer version aimed to improve the weaknesses while maintaining accurate classification. YOLOv2 has significant improvements over its first iteration (Park *et al.*, 2022). Batch normalisation was introduced to improve training

efficiency, increasing the mAP of the model by 2%. The backbone was changed to the Darknet-19 architecture and was downsized by pruning convolutional layers from the network to increase the detector's speed. Lastly, anchor boxes were introduced to address missed detections as a direct effect of aspect ratios (Redmon *et al.*, 2015). Anchor boxes are bounding boxes with predefined aspect ratios and shapes. Classes are assigned anchor boxes, and predictions are based on the IoU between an anchor box and a bounding box, allowing the model to detect a broader range of aspect ratios and sizes. The introduction of anchor boxes successfully increased the recall of the model by 7% (Redmon and Farhadi, 2017).

YOLOv3 features minor improvements aimed at increasing the performance of the model. YOLOv3 includes another reformation to the architecture with an update to a larger network, Darknet-53, matching the performance of ResNet-101 and ResNet-152 while still providing up to 2× the performance. Feature pyramid networks were adopted for this version, enabling enhanced detection at varying scales and improving the detection of smaller objects by concatenating feature maps of different scales (Redmon and Farhadi, 2018).

The development of YOLO was taken over by new authors Bochkovskiy *et al.* (2020); YOLOv4 renewed the previous architecture with a backbone called CSPDarknet-53 based on a variant of ResNet tailored for object detection, the Cross Stage Partial Network. The authors divided the YOLO model into a three-part structure. The backbone focuses on general feature extraction, the neck extracts features at varying scales, and the head handles the prediction of anchor boxes (Diwan *et al.*, 2022). YOLOv4 focused on optimising the YOLO algorithm to be the fastest object detector by a substantial margin at the time of publication (Bochkovskiy *et al.*, 2020).

Following YOLOv4, subsequent versions upheld the three-part structure comprising the backbone, head and neck. The rapid release of successive YOLO model iterations, such as YOLOR (Wang *et al.*, 2021a), YOLOX (Ge *et al.*, 2021), YOLOv5, and YOLOv6 (Li *et al.*, 2022), sparked discussions about the reliability and improvements of these

newer systems in comparison to their predecessors[1]. Differences in the new versions thus prompted whether to change the version name, given the comprehensive testing required to establish credibility.

YOLOv7 presented a significant performance gain (Wang *et al.*, 2022). Benchmarked against prior iterations, YOLOv7 demonstrated the fastest and most accurate real-time detection. Moreover, the model achieved this without using pre-trained weights, unlike previous versions. A novel contribution from the authors of YOLOv7 is the introduction of the Extended Efficient Layer Aggregation Network (E-ELAN). The addition of this network improved YOLO's learning capabilities and efficiency during inference (Casas *et al.*, 2023). YOLOv7 also introduces model scaling in which the architectures are scaled to perform on devices with varied computational power, introducing different versions such as YOLOv7-tiny that aim to keep the performance consistent where possible. YOLOv7-tiny does not just remove layers but has careful considerations to reduce inference time, such as using the Leaky ReLU activation function over SiLU and reducing the number of filters for the convolutional layers (Wang *et al.*, 2022).

Regardless of all reformations, YOLO has consistently proven to be the best-performing single-shot object detector for each version at the date of publication (Redmon *et al.*, 2015, Redmon and Farhadi, 2017, 2018, Bochkovskiy *et al.*, 2020, Wang *et al.*, 2022). Over recent years, YOLO has gained traction within LPR stages, surpassing other object detectors as a preferred choice due to achieving state-of-the-art speed and accuracy. The model can be adapted to the various subtasks in a typical end-to-end LPR pipeline, such as vehicle detection, licence plate detection, and OCR.

2.2 Licence Plate Detection

The selection of the YOLO architecture in literature has primarily been influenced by the requirements of the specific object detection task or by author preference (Casas *et al.*,

[1] https://blog.roboflow.com/yolov4-versus-yolov5/

2023). This section surveys related studies by focusing on methods associated with robust LPR systems.

Lee *et al.* (2018) utilised YOLO for two subtasks in LPR, vehicle detection and licence plate detection. Their approach showed that conducting vehicle detection before licence plate detection decreases the likelihood of errors. During the licence plate detection stage, a comparative analysis was conducted between R-CNN and YOLOv2 to assess their performance differences. Their findings favoured YOLO due to its superior balanced accuracy and inference speed. Although Faster R-CNN achieved a high recall of 93.70%, compared to the YOLOv2-based detection model with a recall of 93.20%, the marginal 0.7% difference did not justify its use over YOLO in a practical setting.

An LPR system focusing on deep-learning methods by Montazzolli and Jung (2017) proposed detecting the frontal portion of a vehicle first, effectively reducing the search space, similarly to Lee *et al.* (2018). Based on a preliminary experiment, the authors reported a low recall when attempting to detect licence plates without vehicle detection prior. Their LPR system utilised YOLO at each stage of the pipeline, with a custom variation created for OCR, CR-NET. While inspiration can be taken from this method, detecting only the frontal portion of a vehicle limits the range of vehicle poses, reducing the ability of the overall LPR system. A full vehicle image would enable an extensive range of poses for potential detection (Silva and Jung, 2021, Al-Batat *et al.*, 2022).

Laroca *et al.* (2018) compared Fast-YOLO and YOLOv2 for use in their end-to-end LPR system, concluding that the smaller efficient Fast-YOLO model was sufficient for vehicle and licence plate detection in both less complex and unconstrained scenarios, but that a deeper model such as YOLOv2 is still preferred for the latter scenario.

Most of the models discussed excel in achieving high recall for their specific tasks. Alongside strong model performance, the quality of training data significantly impacts the model's effectiveness. Boby and Brown (2022) evaluated YOLOv3 for LPR by training the model using data representing challenging scenarios to make a robust plate detector. The model produced excellent results when disregarding anamorphic licence plates, thus leaving room for improvement. Other approaches to improving training data include data

augmentations such as rotations, shearing and cutouts (Silva and Jung, 2018, Laroca *et al.*, 2021b, Al-Batat *et al.*, 2022). These approaches increase training samples without additional data collection, discussed further in Section 2.6.1.

A specialised version of YOLO, the Warped Planar Object Detection Network (WPOD-NET), was optimised to detect licence plates (Silva and Jung, 2018). This state-of-the-art algorithm introduced a significant advantage over existing models. WPOD-NET uses bounding parallelograms, which can conform to the shape of a licence plate in unconstrained vehicle poses and successfully exclude background noise. This model is invariant to oblique angles in a frame or image, allowing it to capture licence plates accurately regardless of their orientation. The model allows better licence plate detection and potential gains in OCR. Approaches to detecting oblique licence plates are discussed in the following subsection.

2.2.1 Detecting Oblique Licence plates

Bounding boxes introduce background noise into oblique licence plate crops. OCR often misinterprets background noise as characters, resulting in false positives. Standard object detection algorithms are not flexible enough for the appropriate detection of oblique licence plates. The following subsections explore ways to rectify this problem.

2.2.1.1 Perspective Transform

A perspective transform is an image correction technique used to rectify undesirable perspective distortion in an image. This approach can be applied in the LPR domain to normalise oblique licence plates (Gao and Xiang, 2023).

Correcting for distortion places characters on a licence plate in a horizontal line, presenting fewer complications for OCR (Kim *et al.*, 2021). With reduced distortion, characters are easier to detect based on their learned features. The perspective transform works

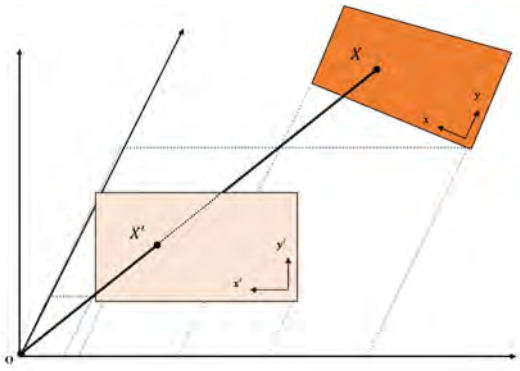


Figure 2.4: Perspective transformation from X to X' (Kim *et al.*, 2021).

optimally with LPR if the coordinates of the region of interest (ROI) are captured accurately. An illustration of the process is shown in Figure 2.4. The perspective correction requires the initial coordinates to form a parallelogram instead of a rectangle or square, typically obtained from a bounding box (Zhang and He, 2007).

Zhang and He (2007) proposed a method to estimate the aspect ratio of a quadrangle by the assumption that it is a rectangle projected in space. Applying this approach towards LPR could automate square and rectangular licence plate detection, simplifying the process of recognising multi-row licence plates.

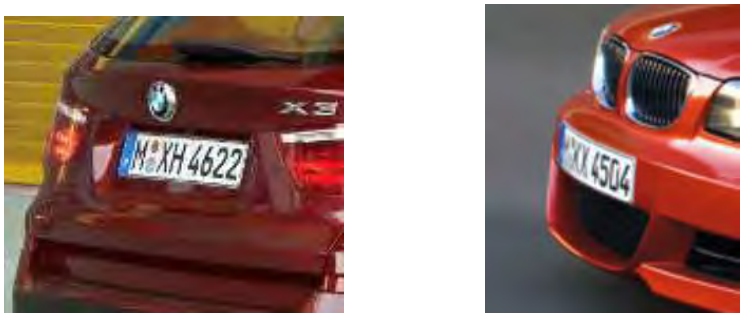
Incorporating perspective transform requires an approach more versatile than a bounding box. Candidates include the WPOD-NET (Silva and Jung, 2018) and the Multi-Oriented and Scale-Invariant Licence Plate Detection System (MOSI-LPD) (Han *et al.*, 2019), which are models in the LPR domain that have been successful at introducing parallelograms for accurate ROI capture.

2.2.1.2 Warped Planar Object Detector

WPOD-NET is a novel CNN crafted to detect and deskew licence plates using coefficient regression to perform affine transformations on an extracted bounding box (Silva and Jung, 2018). The WPOD-NET differentiates itself from other object detectors, such as YOLO and R-CNN, by creating parallelograms instead of bounding boxes. These are useful for complex licence plate detection scenarios. Using bounding parallelograms,

WPOD-NET can utilise coordinates to transform oblique licence plates into rectangular ones akin to those taken from a front-parallel view. Frontal licence plate detection presents minimal distortion challenges and is trivial with existing deep-learning networks. Figure 2.5 shows some less trivial detection tasks, where 2.5b is an edge case.

Augmenting these images enables increased performance even for less robust OCR systems, as there is reduced character distortion, addressing one of the bottlenecks of LPR. The challenges of OCR are addressed further in Section 2.3.



(a) **(b)**

Figure 2.5: Examples of licence plates captured at oblique angles.

The proposed pipeline of the system by Silva and Jung (2018) is shown in Figure 2.6.

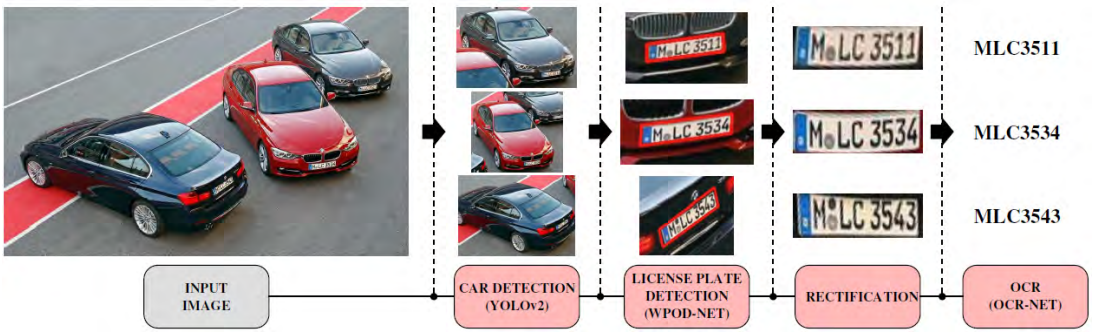


Figure 2.6: WPOD-NET pipeline (Silva and Jung, 2018).

The WPOD-NET utilises shared weights for both classification and detection, which can backpropagate conflicting information from both tasks to preceding layers. Silva and Jung (2021) refined their earlier approach by adding two sub-networks to the end of the model to decouple detection and classification. This new approach was aptly named the Improved Warped Planar Object Detection Network (IWPOD-NET). The new model improved

LPR results by up to 3.7% based on an ablation study comparing the performance of WPOD-NET and IWPOD-NET (Silva and Jung, 2021).

The unique properties of the IWPOD-NET make it an attractive option for implementing perspective correction. As stated in Section 2.2.1.1, perspective transformation requires the exact corner points of a licence plate within an image for the correction to work effectively.

2.2.1.3 Multi-Oriented and Scale-Invariant Licence Plate Detection

MOSI-LPD is based on Faster R-CNN, adapted to output form-fitting bounding parallelograms like the WPOD-NET (Han *et al.*, 2019). Figure 2.7 shows bounding boxes and how they compare to the bounding parallelogram produced by the MOSI-LPD system. Bounding parallelograms were selected because they have free angles and orientation and can fit arbitrary regions as opposed to standard bounding boxes, which use horizontal rectangles, limiting flexibility when labelling a region in an image.

Figure 2.7: From left to right, variations of bounding shapes include a standard bounding box, an oriented bounding box and a bounding parallelogram. The bounding parallelogram is the most effective way to capture the ROI with minimal background noise.

MOSI-LPD and IWPOD-NET present solutions that can be prerequisites for perspective correction. Although both models produce the desired output, the IWPOD-NET holds

two advantages: it is based on the YOLO algorithm, which is faster than R-CNN (Diwan *et al.*, 2022), and it is a more recent and refined implementation of the earlier WPOD-NET approach.

2.3 Licence Plate Optical Character Recognition

OCR is an essential stage of automatic LPR systems as it is typically the end goal. After extracting a licence plate from an image or video stream, the data must be converted into information for storage or string matching. A typical OCR algorithm converts text from an image into processable characters through image pre-processing steps, such as thresholding and canny edge detection (Kessentini *et al.*, 2019).

OCR was developed primarily for digitising documents for electronic storage. The idea of OCR extends beyond documents and can be applied to text detection in unconstrained scenarios. Image-processing methods have been applied to OCR for LPR with varying degrees of success. Dated solutions used TesseractOCR, an open-source multi-platform OCR engine developed in C++ (Smith, 2007). These older methods cannot adapt to data; varied lighting conditions or unfamiliar contrast in licence plates are too complex. Concepts were explored to discover methods to overcome these limitations. Brisinello *et al.* (2017) considered image upscaling to see how it would affect the performance of TesseractOCR. Using bicubic interpolation to upscale their images increased OCR accuracy by up to 20%. Bicubic upscaling preserved more detail than alternative methods, such as bilinear interpolation and had a reasonable computation cost. The fundamental problem with TesseractOCR is it aims to detect text in documents, posters, and similar mediums. Nevertheless, TesseractOCR is still used by commercial systems such as OpenALPR[2] (Špaňhel *et al.*, 2017). A more refined approach is required to detect characters in unconstrained scenes, representing LPR in the real world.

EasyOCR is a deep learning-based OCR model more suited to text detection in unconstrained scenarios (Idrose *et al.*, 2022). The model utilises Character Region Awareness

[2]https://github.com/openalpr/openalpr

For Text detection to separate text from a scene, focusing on locating regions containing text rather than predicting bounding boxes like other models such as CR-NET (Baek *et al.*, 2019). When compared against TesseractOCR on licence plate data, it achieved a 7% higher accuracy. Additionally, the results indicated that TesseractOCR favoured letters while EasyOCR favoured numerical data (Vedhaviyassh *et al.*, 2022). Idrose *et al.* (2022) particularly noted in their findings that both models struggled to detect multi-row licence plates efficiently. A less explored approach to OCR is object detection; its application in related literature is discussed in the following section.

2.3.1 OCR through Object Detection

Object detectors are pre-trained on large datasets of annotated images and can identify objects by learning the patterns and features characteristic of each object class. Their versatility allows them to be trained to detect text with some adjustments to how they predict classes. Approaching OCR through object detection has received minimal attention, and based on results obtained from the literature, YOLO is a strong candidate for OCR.

Montazzolli and Jung (2017) produced CR-NET, an early YOLO-based OCR model derived from a computationally efficient YOLO version (Fast-YOLO). The system was adapted to recognise Brazilian licence plates, achieving an accuracy of 63.18% when detecting a combination of letters and numbers. The model was configured to take in a fixed licence plate shape, limiting its ability to detect square ones. Laroca *et al.* (2018) employed CR-NET in their research, validating the practicality of using YOLO for LPR. Silva and Jung (2020) extended CR-NET by employing data augmentation with synthetic data to increase the representation of weaker classes in their training data. Character swapping addressed character ambiguity successfully using the Brazilian licence plate format, as the domain was restricted to one region. The model detected 35 classes, 0-9 and A-Z, with '0' and 'O' as a joint class, improving recognition by 25.97% (Montazzolli and Jung, 2017) and outperforming a commercially available system by 1.71%. Thus highlighting

the effectiveness of object detection for OCR. However, the fixed aspect ratio preventing square licence plate detection remained unaddressed.

Other approaches include Fast-OCR and YOLOv2. Fast-OCR is a small framework that detects digits within small images for meter readings (Laroca *et al.*, 2021a). Much like CR-NET, FastOCR is based on a lightweight network, Fast-YOLOv4, and achieved an average recognition rate of 95.87% in its target domain. Kessentini *et al.* (2019) compared a Convolutional Recurrent Neural Network with YOLOv2 for OCR using Tunisian licence plates datasets. YOLOv2 outperformed this model by 16.52% on the smallest dataset, demonstrating learning efficiency. Moreover, YOLOv2 was tested on the Tawainese AOLP dataset (Hsu *et al.*, 2012), achieving a high character recognition rate, emphasising the flexibility of a YOLO-based OCR solution. Lastly, Kim *et al.* (2021) favoured YOLOv2 for OCR in their methodology based on their review of existing literature at the time of publication.

Contributions to object detection for OCR have been limited to older versions of YOLO, with implementations revolving heavily around CR-NET and YOLOv2 (Hsu *et al.*, 2012, Montazzolli and Jung, 2017, Laroca *et al.*, 2018). Improvements may be found by exploring OCR with updated versions of YOLO, such as version seven, as defined in Section 2.1. Insight from these papers informs methods to counter challenges associated with OCR, which are discussed in Section 2.3.2.

2.3.2 Challenges

There are a few challenges that make OCR difficult. These are mainly character ambiguity, distortion and resolution, which are closely linked. Several Latin characters share similar features, making them ambiguous at the detection stage (Patel *et al.*, 2012, Silva and Jung, 2020). Unlike human reading, OCR cannot use semantics or context. Instead, it relies on many training samples for performance. This problem becomes more apparent with licence plates as they include alphanumeric characters, which can easily be mistaken for each other during classification. For example, as shown in Figure 2.8, characters such as '0'

and 'O' or '1' and 'I' can be identical in some regions, resulting in an ambiguity problem. An error or misidentification of one character can recognise an invalid licence plate (Silva and Jung, 2020). Perspective distortion can exacerbate ambiguity as characters become more challenging to distinguish from oblique angles, thus making an already tricky case more complex.

(a) **(b)**

Figure 2.8: Characters on Brazilian licence plates such as [0, O] [1, I] are identical (De Oliveira *et al.*, 2021).

Depending on the region, the ambiguity problem may be less prominent as some fonts add more defining features to particularly problematic characters. The example in Figure 2.9 shows how typefaces can be unambiguous. Another way this is done is through serifs, which make ambiguous characters easy to distinguish from one another, as illustrated in Figure 2.10. Unfortunately, the lack of an international standard means different regions use varied typefaces on their licence plates, making LPR much more difficult (Silva and Jung, 2020). Moreover, multi-national LPR makes using a character-swapping algorithm difficult as licence plate formats vary by region.

OCR is sensitive to low-resolution input. Smaller images can cluster pixels together, making unclear characters more challenging to detect and amplifying the ambiguity problem. Colour and busy backgrounds within an input image can also decrease the accuracy of the predictions made by the OCR models. These models perform better when given higher-quality input. Upscaling techniques can be utilised to increase the resolution of the image to increase the clarity. Traditionally, the approach has been methods such as bilinear and bicubic upscaling (Brisinello *et al.*, 2017). Advancements in computing

(a) **(b)**

Figure 2.9: Two separate ways the character zero is differentiated from 'O' on existing licence plates.

Figure 2.10: The appearance of characters with and without serifs.

power enable upscaling images in real-time with deep learning. This presents a use case for super-resolution methods discussed further in Section 2.4.

2.3.3 Measuring the Performance of OCR

Identifying a metric consistently applied in the literature to represent OCR performance accurately poses a smaller challenge in the domain. Generally, the equation involves the number of correctly predicted characters and the number of errors or alterations to align the output with the ground truth. In literature, there are several approaches to measuring character recognition accuracy, making performance comparisons non-trivial.

The character recognition rate, synonymous with accuracy in the OCR domain, is commonly represented as:

$$\text{character recognition rate } = \frac{n-m}{n}. \tag{2.2}$$

Where n represents the number of correctly recognised characters, while m represents

the number of incorrect predictions or deletions and replacements needed to equate an output string to a target string (Shen and Lei, 2015). To remedy this, Shen and Lei (2015) revised the formula to prevent potential zero division errors or negative values. The revised formula includes the total number of characters *all* and is shown in Equation 2.3.

$$\text{character recognition rate } = \frac{n}{all + m}. \tag{2.3}$$

An example of inconsistency in the literature is shown in Equation 2.4:

$$\text{recall } = \frac{n}{all} \tag{2.4a}$$

$$\text{character recognition rate } = \frac{no.\ of\ correctly\ recognised\ characters}{total\ no.\ of\ characters.} \tag{2.4b}$$

Analysing the equation reveals that 2.4b represents the same formula as 2.4a. The work of Kessentini *et al.* (2019) defines it as character recognition rate, while Shen and Lei (2015) defines it as recall. This discrepancy raises the question of the appropriate metric to use, as the application of separate character recognition rate formulas makes results incomparable. Further extending the problem, Montazzolli and Jung (2017) uniquely reported their findings, listing the accuracy relative to the number of correctly guessed characters without providing a formula for reproducibility. Besides character recognition rate, Levenshtein distance can also be used to quantify OCR performance.

Levenshtein distance is the most robust and widely used metric for measuring the similarity of two strings. This metric calculates the disparity between two strings by determining the least amount of edits required to transform one string into another (Spruck *et al.*, 2022). For instance, a variation of one character results in a Levenshtein distance of one. The advantage of this metric is it does not require two strings to be of equal length, allowing for partial detections to be evaluated. Accuracy for character recognition parallels string matching, making Levenshtein distance a suitable choice. Silva and Jung (2021)

used the metric to measure the accuracy of their OCR model, adding another variation to performance calculation, complicating the comparison between different OCR methods.

Nevertheless, the Levenshtein distance can be used to calculate m within the character recognition rate formula (Equation 2.3) proposed by Shen and Lei (2015). The two formulae can be combined to create an accurate method of calculating the character recognition rate.

2.4 Super-Resolution

Blind super-resolution is a deep-learning technique for upscaling images, generating a high-resolution image from a low-resolution input with an unknown degradation. Advances in computational power and resources have allowed deep learning to surpass traditional image processing techniques for image upscaling. Thus enabling models to enhance beyond what is available in the input data.

Using super-resolution to improve accuracy during the OCR stage would benefit LPR. Using a fully computational model mitigates expensive hardware costs. For example, some regions deploy police vehicles equipped with expensive telephoto and infrared cameras tailored for licence plate capture (CTV, 2022). Such solutions may not be accessible in certain countries, thus, creating a software-based solution with comparable performance that can be deployed as an affordable solution would benefit either a government organisation or a private entity.

2.4.1 Image Quality Analysis

Quantifying image quality is a challenging task, emphasising the need for a suitable measure to evaluate the output from super-resolution models. Qualitative analysis is effective yet time-consuming, while quantitatively assessing results enables measuring a model's performance mathematically (Wang *et al.*, 2002, Ye *et al.*, 2023).

Mean Squared Error (MSE) can be used to measure image quality, but as described by Wang and Bovik (2009) MSE does not sufficiently represent human perception. For instance, Figure 2.11 shows two variations of an image with an MSE of 309 but with a drastic difference in quality. This shows quantitative methods are not so easily correlated with human perception.

(a) Original **(b)** Noticeable degradation. **(c)** No easily noticeable flaws.

Figure 2.11: The images 2.11b and 2.11c are perceivably different quality but identical MSE (Wang and Bovik, 2009).

Alternative methods for measuring image quality exist, such as Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Referring back to Figure 2.11, although the two images have an identical MSE, they have distinct SSIM values, 0.580 and 0.987, respectively, showing SSIM is a more accurate measure. Recent literature has explored novel metrics to analyse the performance of super-resolution models beyond SSIM and PSNR, which have been the dominant means of quantitative image analysis (Wang *et al.*, 2020, Kirillova *et al.*, 2021).

One such metric is Edge Restoration Quality Assessment (ERQA), which addresses some of the problems related to these metrics (Kirillova *et al.*, 2021). The subjective scores from visual analysis often do not compare with the calculated values from metrics such as SSIM and PSNR. For instance, Boby *et al.* (2023a) identified a discrepancy in their findings where images upscaled with a diffusion-based model appeared subjectively better to human perception. Still, objective measures such as SSIM and PSNR favoured the corresponding distorted ESRGAN result. Additionally, ERQA offers a tailored metric for evaluating super-resolution models by assessing high-level features between ground truth and super-resolution images and providing visualisations.

Another metric is Learned Perceptual Image Patch Similarity (LPIPS) (Zhang *et al.*, 2018). LPIPS is a deep learning-based perception metric that uses deep features to compare images. The method has a higher correlation with human perception than methods appearing before it. While older than ERQA, LPIPS has increased in popularity and application in a few papers since its creation. Despite this, SSIM and PSNR remain the most popular techniques (Lyapustin *et al.*, 2022).

At the core, all techniques require calculating the difference between two images. A super-resolution model reproduces an image based on supplied training data, and the output will always differ from the original (Lyapustin *et al.*, 2022). For LPR, the goal of the super-resolution is to enhance character clarity for improved recognition (Boby and Brown, 2022). Therefore, alongside quantitative analysis, qualitative analysis is beneficial for inspecting image samples to observe how well they are reconstructed, as currently, no metric can adequately quantify human perception when viewing images (Sara *et al.*, 2019).

2.4.2 Super-Resolution Convolutional Neural Network

The Super-Resolution Convolutional Neural Network (SRCNN) was one of the first successful deep-learning models for super-resolution, drawing inspiration from CNNs (Anwar *et al.*, 2020).

The network consists of three convolutional layers with separate tasks. The first layer creates feature maps, the second layer, dubbed non-linear mapping, converts the feature maps into higher-dimensional feature vectors, and the final layer combines the feature maps to create a high-resolution image (Dong *et al.*, 2014, Anwar *et al.*, 2020).

A synthetic dataset of high-resolution patches downsampled and upsampled with bicubic interpolation was used to create low- to high-resolution counterparts to train the network. During training, the model aimed to minimise MSE between ground truth high-resolution images and the generated high-resolution images (Dong *et al.*, 2014). The model produced

superior-quality images to bicubic upscaling even after a few iterations during training (Dong *et al.*, 2014).

Although MSE methods produce reasonable high-resolution output, they do not utilise CNNs' full potential, resulting in blurry images (Lucas *et al.*, 2019). Furthermore, as discussed in Section 2.4.1, MSE does not equate to the human perception of image quality and fidelity. Adversarial loss was explored as an alternative loss function for super-resolution models.

2.4.3 Generative Adversarial Networks

Due to their learning ability, GANs became increasingly popular for their application to computer vision tasks. They were adopted as state-of-the-art due to accurate super-resolution images surpassing the quality of the SRCNN (Wang *et al.*, 2020, Lyapustin *et al.*, 2022).

A GAN comprises two neural networks: A discriminator and a generator. During training, the discriminator discerns whether generated output images are real or fake (Brock *et al.*, 2018). Depending on the validity of the discriminator's prediction, the model weights are updated according to the adversarial loss function. Eventually, the generator learns to produce images that bypass the discriminator each time, allowing the generator to be used for creating images in a target domain.

A specialised GAN called the SRGAN aims to generate high-resolution images from low-resolution counterparts through the same training process (Ledig *et al.*, 2017, Lee *et al.*, 2019). The literature supports a correlation between super-resolution through SRGANs and increased OCR accuracy. Evidently, Lee *et al.* (2018) used the SRGAN to upscale low-resolution licence plates and successfully reduced false positives at the OCR stage. While Kim *et al.* (2021) reported upscaling led to sharper edges and clearer images, improving the character recognition rate on CCTV footage. GANs are notoriously challenging to train and can suffer from mode collapse during the training process (Srivastava *et al.*,

2017). Mode collapse causes the generator to link several input points to one output, resulting in similar images for a range of varied data — which is undesirable.

The Enhanced Super-Resolution GAN (ESRGAN) mitigates this with a Residual-in-Residual Dense Block (RRDB) for more accessible model training to improve the discriminator's ability to judge image realism, giving it an advantage over the standard SRGAN (Wang *et al.*, 2018). Boby and Brown (2022) used the ESRGAN, observing greater accuracy by enhancing licence plate images prior to OCR. However, Lin *et al.* (2021) proposed some modifications to the ESRGAN to improve licence plate construction but reported lower SSIM scores when compared to the original implementation.

Similar to the SRCNN, the SRGAN and ESRGAN upscale images using low- and high-resolution ground-truth image pairs. Yuan *et al.* (2018) argued that such an end-to-end mapping is impractical in the real world as low-resolution images rarely have a corresponding high-resolution counterpart. Previous super-resolution models use bicubic downsampling, which does not represent real-world scenarios. The Real-ESRGAN attempts to address this by extending the ESRGAN, introducing a more realistic image degradation process in the model's training (Wang *et al.*, 2021b). Moreover, using a Sinc filter removes high frequencies and helps reduce artefacts in high-resolution output images, especially for text and lines. Figure 2.12 compares some existing deep-learning methods to the Real-ESRGAN.

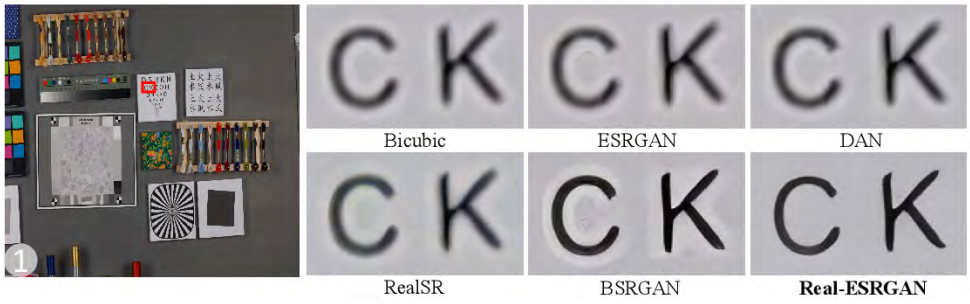


Figure 2.12: The text restoration of the Real-ESRGAN is superior to that of the ESRGAN (Wang *et al.*, 2021b).

The Real-ESRGAN is less widely used than the ESRGAN; with its enhancements, such as Sinc filters, the model can produce more legible results for OCR and may further accelerate progress for LPR.

2.4.4 Diffusion Probablistic Models

A diffusion probabilistic/diffusion model works with an iterative process that performs exceptionally well for image generation. Unlike GANs they do not experience mode collapse — as defined in Section 2.4.3. Hence, recent implementations such as Blind Image Restoration with Diffusion Prior (DiffBIR), SR3, and StableSR have adapted them for super-resolution (Ho *et al.*, 2020, Saharia *et al.*, 2022, Lin *et al.*, 2023), boasting improved results compared to a widely used super-resolution model like the ESRGAN (Lin *et al.*, 2023).

The diffusion process consists of two stages: a forward and backward process. In the forward process, gaussian noise is gradually added to an input image, while the backward process denoises the image. This approach is more stable than generating images with GANs (Ho *et al.*, 2020).

The Markov chain represents the forward diffusion process in Equation 2.5.

$$q\left(x_1, \ldots, x_T \mid x_0\right) := \prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right) \tag{2.5a}$$

$$q\left(x_t \mid x_{t-1}\right) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t I\right) \tag{2.5b}$$

The backward diffusion process is shown in Equation 2.6.

$$p_\theta\left(\mathbf{x}_{0:T}\right) := p\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) \tag{2.6a}$$

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{2.6b}$$

Figure 2.13 models both the processes visually. Each time-step, t, relies on the previous step, gradually reversing the noise in an image, $x_t, x_{t-1}, \ldots$, until a clear image is produced

at x_0. More iterations improve the final output image quality at the expense of inference time and computational resource requirements.

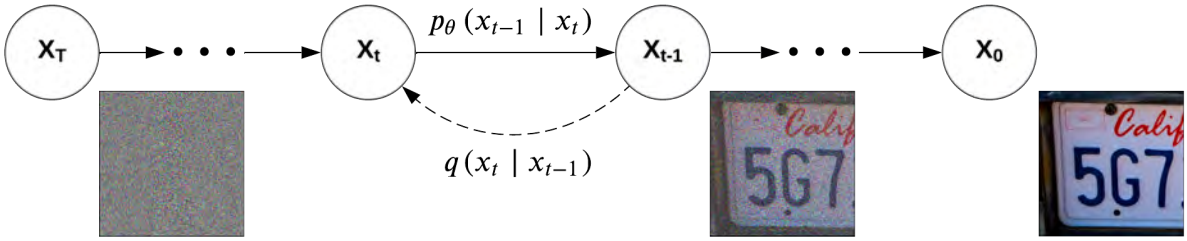


Figure 2.13: A visual representation of the forward and backwards processes.

The capabilities of diffusion models are yet to be applied practically for licence plate restoration. SR3 reports improved restoration with reduced artefacts compared to the ESRGAN (Saharia *et al.*, 2022), only referring to faces and natural images, with no samples reporting the model's performance on images containing text.

Much like the Real-ESRGAN, DiffBIR creates downsampled training images with various modes of degradation more representative of the real world (Lin *et al.*, 2023). DiffBIR follows a two-stage process employing a restoration model followed by a pre-trained stable diffusion model, which performs the upscaling, introducing fine details and sharpness to the image. The model achieves superior performance compared to other super-resolution models, including the Real-ESRGAN (Lin *et al.*, 2023).

Visual results from StableSR, DiffBIR and SR3 all have a high quality and fidelity, but all underperform based on standard PSNR and SSIM metrics; conversely, they achieve higher scores with LPIPS (Saharia *et al.*, 2022, Lin *et al.*, 2023, Wang *et al.*, 2023). Comparing the abilities of diffusion probabilistic models for upscaling to the capabilities of SRGANs could discover a superior method for licence plate super-resolution, which has been dominated by GAN-based methods so far.

2.5 Vehicle Surveillance

Vehicle surveillance involves monitoring vehicles and tracking their movement. LPR can play a significant role in surveillance systems, enabling vehicle tracking, identification and re-identification (Kalake *et al.*, 2022).

2.5.1 Tracking

Vehicle tracking can add an extra layer of effectiveness to surveillance systems by isolating vehicles and continuously performing LPR to provide them with an identity.

Simple Online and Realtime Tracking with a Deep Association Metric (DeepSORT) (Wojke *et al.*, 2017) uses deep learning to track objects in video sequences. It is an extension of the SORT algorithm that integrates a deep appearance descriptor to improve tracking accuracy, especially with occluded or similar-looking objects. The DeepSORT algorithm uses a CNN to generate deep features that describe the appearance of an object, and these are combined with attributes such as location, size, and motion to track the object's movement over time.

The algorithm uses the Hungarian method[3] to associate object detections across multiple frames using IoU to maintain object identities. When objects are occluded, Kalman filtering is applied to predict object locations, relying on the previous state to estimate their trajectory. For the DeepSORT algorithm to perform effective tracking, it needs an initial detection to be supplied.

DeepSORT can integrate well with many existing object detection models such as YOLO (Wei, 2022), with literature considering versions from YOLOv3 to YOLOv7 (Santos *et al.*, 2020, Yang *et al.*, 2022). The application of DeepSORT for vehicle surveillance includes tasks such as vehicle tracking and counting (Santos *et al.*, 2020, Zuraimi and Zaman, 2021). DeepSORT's modularity concerning object detection models makes it a candidate for adoption in an LPR system.

[3] A simple algorithm that solves the assignment problem in polynomial time.

2.5.2 Re-Identification

Vehicle Re-Id seeks to find a vehicle with a known identity within camera footage. This form of surveillance can differentiate vehicles by their unique features. As discussed in Section 1.3, the licence plate is a unique identifier for a vehicle and can be leveraged for vehicle identification and Re-Id across camera footage. Re-Id is helpful in situations with vast amounts of data requiring a particular subject to be found (He *et al.*, 2020). The worst-case surveillance scenario for a human operator would require them to seek through twenty-four hours of footage to locate a target vehicle. With suitable training, automated solutions can accurately identify targets in real-time, at least 30 FPS (frames per second) (Laroca *et al.*, 2018). Furthermore, the efficiency of subject identification in large datasets can be improved, presenting a use case for Re-Id through deep learning. Approaches to Re-Id from literature are discussed in the paragraph that follows.

Progressive Vehicle Re-Identification (PROVID) is a specialised Re-Id system for the vehicle domain (Liu *et al.*, 2017). It aimed to advance vehicle Re-Id by introducing licence plate detection and temporal information, aspects often overlooked in similar systems. PROVID adopts a coarse-to-fine approach when re-identifying a vehicle and uses a Siamese neural network for licence plate verification, trading accuracy for efficiency. Liu *et al.* (2017) argue that constraints of road cameras make using a licence plate for identification unreliable. However, they did not consider PROVID could be improved by incorporating super-resolution as a system component to enhance licence plates (Lee *et al.*, 2019), enabling them to be a reliable primary identification means.

A similar approach was used by de Oliveira *et al.* (2019), where they proposed a two-stream Siamese neural network to identify vehicles using two distinct features: the licence plate and vehicle shape. Their initial model adopted licence plate matching as opposed to OCR. An extended version of their system utilised characters, as the previous licence plate matching method was unreliable (De Oliveira *et al.*, 2021). Using OCR improved the system's performance, showing surveillance systems such as PROVID can benefit from leveraging licence plate information for better tracking and re-identification.

2.6 Data Availability

Various publicly available datasets can be used to train machine learning models for specific tasks. South Africa, in particular, has no public dataset available to train and test models on local licence plates. The most extensive datasets primarily consist of Chinese or Brazilian licence plates (Laroca *et al.*, 2018, Xu *et al.*, 2018). Authors of the respective countries have advanced a large portion of LPR research. For South Africa, data would have to be collected to create an extensive dataset, typically requiring approval from an ethics committee. Not much data is readily available beyond short samples of IP-approved highway data. Beyond ethics approval, physically collecting a comprehensive dataset would be time-consuming. Hence, generalised licence plate data is necessary for a robust system aimed towards multi-national LPR. The lack of data can be compensated for with image synthesis.

2.6.1 Manually Generated Synthetic Data

Data can be generated through procedural programs and rendering software. The following literature has approached data scarcity by manually generating synthetic data.

The model proposed by (Björklund *et al.*, 2019) used synthetic licence plate data to train a CNN for LPR. The generated dataset enabled the authors to eliminate data collection and labelling time constraints. Moreover, the ability to synthesise images allowed them to control variables affecting LPR difficulty, resulting in a model that could successfully detect and generalise well to real-world images.

Kessentini *et al.* (2019) expanded on the approach by extending an existing dataset by supplementing it with 40,000 synthetic licence plate images. As deep-learning models thrive from large amounts of data, combining synthetic and real data can increase the volume of existing datasets. Alternatively, three-dimensional modelling could be utilised to create virtual scenes for training.

Similar to 2D images, a virtual scene offers control over various variables, including lighting intensity and occlusion. This can enable training for situations particularly difficult to recreate in real life. Spruck *et al.* (2022) proposed using synthetic data as there was only one commercial German licence plate dataset at publication. Using a 3D modelling software called Blender and an automated script, they created a novel dataset with randomised licence plates, which did not breach ethical guidelines (Spruck *et al.*, 2022). The approach is adaptable to other countries by replacing the font and templates to match the desired country.

Data augmentation can also be employed to increase occurrences and variability in datasets. Al-Batat *et al.* (2022) augmented their licence plate data to include additional samples representative of complex scenarios by applying blur, brightening, blobs and shadows. Furthermore, they introduced a character permutation augmentation method to increase occurrences of underrepresented classes in licence plate datasets. As licence plate datasets suffer from large class imbalances, this method can be applied to reduce the effect.

2.6.2 Image Synthesis with Generative Models

Due to the unique requirements of vehicle identification data, it is challenging to create large-scale datasets where the same vehicle reappears in multiple scenes. Moreover, some vehicle datasets may have limited poses, restricted to rear or frontal views. In a virtual scene, camera angles and perspectives can be controlled to create novel datasets (Herzog *et al.*, 2023), enabling vehicle tracking and identification models to be tested. A common issue for vehicle Re-Id datasets is that, for privacy reasons, the licence plate characters are redacted. Thus, the dataset cannot be used to test licence plate detection with vehicle tracking. Using synthetic or virtual scenes could help alleviate this problem.

As an alternative to the standard computer-generated methods in Section 2.6.1, synthetic images can be produced through generative models such as GANs and diffusion models. This presents a paradox as deep-learning models require large amounts of data for training; conversely, the problem to be solved is a scarcity of data. Regardless, these models have achieved some success in image synthesis.

Kramberger and Potočnik (2020) proposed a novel dataset for training GANs for vehicle image synthesis. They filtered existing car datasets, selecting only the best images containing features appropriate for training GANs. The results showed their refinements to the dataset successfully yielded a GAN model that can create vehicle images. The resulting model could be used to train deep-learning models for vehicle detection (Kramberger and Potočnik, 2020).

There is a domain gap between synthetic and real-world data, characterised by illumination and image backgrounds. Existing approaches to make synthetic data more robust employ GANs to enhance photo-realism. Zheng *et al.* (2020) used this approach to transform synthetic vehicle images into images more representative of real-world data. They produced a superior model than the one trained solely on real data. The VehicleX dataset is a novel 3D dataset aimed at solving problems related to data acquisition, such as time and cost (Yao *et al.*, 2020), applying the same principles as Zheng *et al.* (2020), the synthetic data domain gap is reduced using a GAN. Richter *et al.* (2022) used a similar approach where they used a GAN to convert rendered city driving data from a video game to real-world footage. Literature supports combining generative models and procedurally generated methods discussed in Section 2.6.1 to synthesise more realistic datasets.

GANs were the predominant model for image generation prior to advancements in the diffusion model space. Dhariwal and Nichol (2021) presents a paper on how diffusion models outperform GANs for image synthesis — describing GANs as the current state-of-the-art regarding image generation while noting diffusion models produce images perceived as higher quality. To control image synthesis with diffusion models, a model is trained using a frozen Contrastive Language-Image Pre-training text encoder to relate words to objects so suitable images can be generated with prompts from an end user (Rombach *et al.*, 2022). This trains the model to associate certain words or phrases with specific features or attributes in the images, which can control the generation process and produce images with specific characteristics (Schuhmann *et al.*, 2022).

Using a diffusion model for image synthesis is particularly useful for generating realistic images, as it considers the natural statistical properties of images, such as the correlation

between neighbouring pixels. Using the forward and backward process described in Section 2.4.4, the model can also ensure the final image is coherent and visually pleasing (Ho *et al.*, 2020).

Diffusion models exhibit great versatility; for example, they can be used to create an image of a car in snow or dust, allowing a licence plate detector to be trained for differing environments, eliminating the need for physical data collection (Boby *et al.*, 2023b). Moreover, the iterative nature of diffusion models does not significantly impact data collection times as these models run in a reasonable time, minutes, as opposed to days compared to collecting real-world data.

2.7 Summary

This chapter reviewed existing literature to gauge the current state of LPR. The importance of accurate OCR was explored, and automation can be achieved by applying deep learning. Such models are essential for accurate LPR in challenging situations.

The utility of object detection models for LPR was assessed, and various models were reviewed to find the most suitable candidate. Based on the literature, the YOLO model is the most developed and recommendable model for vehicle detection and OCR. However, the older YOLOv2 has been widely adopted in related literature. Performance and accuracy improvements could be observed using newer architecture versions and thus should be explored.

Specialised models for licence plate localisation were explored to solve a common problem related to bounding boxes. Bounding parallelograms provide a more accurate means of licence plate localisation, which is necessary to correct distortion in licence plates. Two architectures were investigated for this task: MOSI-LPD and IWPOD-NET. IWPOD-NET presented a superior architecture based on YOLO, with high inference speed.

Moreover, super-resolution and data augmentation were analysed as potential approaches to address low-quality and low-resolution data commonly associated with unconstrained

data from mediums such as CCTV cameras for vehicle tracking and surveillance. In numerous vehicle Re-Id systems, the significance of the licence plate as a distinct identifier was disregarded. Ensuring its legibility is crucial for the efficacy of any identification system. While GANs were suggested, adopting state-of-the-art diffusion models within the LPR domain remains understudied. There is a need to delve into the potential of these models to contribute to the existing knowledge base.

3

System Methodology

This chapter discusses the proposed system intended for reliable LPR while relying on vehicle detection and tracking for a re-identification system. It first provides an overview, followed by a detailed analysis of the system's design in subsequent sections.

3.1 Vehicle Re-Id and Retrieval System Overview

This research proposes an end-to-end vehicle Re-Id and retrieval system using a multi-stage approach to address the requirements in Section 1.6. Since the primary aim of the system is to accurately recognise licence plates for Re-Id in unconstrained scenarios, all of the stages require optimised state-of-the-art methods. Based on the literature survey in Chapter 2, successfully achieving this aim requires additional tasks as a foundation, including vehicle detection, licence plate rectification and super-resolution.

A high-level design of the end-to-end system is illustrated in Figure 3.1.

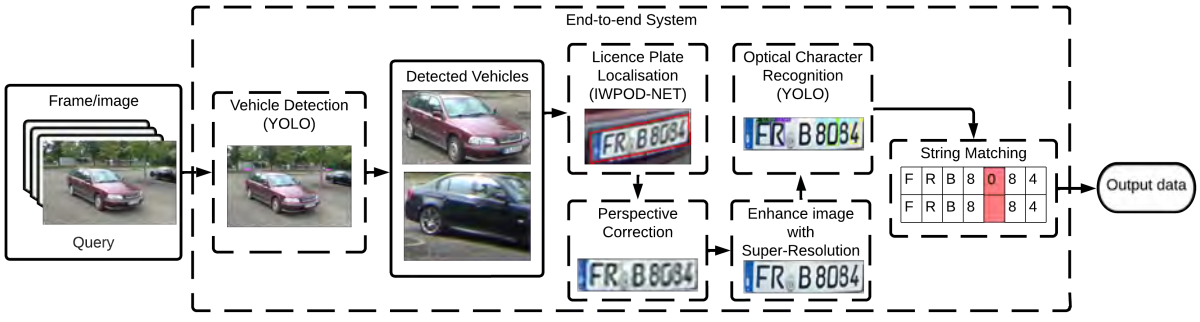


Figure 3.1: The high-level design of the proposed vehicle Re-Id and retrieval system.

In the first stage, vehicles are detected within a scene to minimise background regions, which typically cause false positives in subsequent stages (Al-Batat *et al.*, 2022). The

second stage focuses on localising the licence plate within a vehicle patch obtained from the vehicle detection stage. Perspective transformation is applied to normalise licence plate images to a front-parallel view, horizontally aligning characters for improved recognition. The third stage involves OCR by detecting each alphanumeric character as an object. As discussed in Section 2.3.2, OCR is sensitive to false positives due to many character classes and ambiguity. Therefore, the normalised licence plate is first enhanced using super-resolution, improving the visibility of its characters. The high-level design makes it clear that the vehicle detection, licence plate localisation and OCR stages comprise several methods that require tailoring to meet the system requirements.

The implementation, including method details and modifications, is thus provided in great detail in Chapter 4. Furthermore, each module of the system will be tested to analyse individual performance, followed by a test of the combined performance of the end-to-end system. Hence, the low-level design details are first provided per module in the following sections.

3.2 Vehicle Detection

The foundation of the proposed vehicle Re-Id system design relies on selecting a robust object detection model such that subsequent stages are not prone to failure. The literature survey revealed that vehicle detection to reduce images to vehicle patches is an effective way to increase recall for licence plate localisation in the next stage. YOLO was commonly used, and based on revisions, YOLOv7 emerged as the optimal choice due to its superior performance, particularly in achieving higher mAP on benchmark datasets compared with previous models, detailed in Section 2.1.3. Hence, the improved learning capabilities are expected to benefit vehicle detection results.

Figure 3.2 describes how a vehicle detection stage would work with an input image of arbitrary dimensions. The image is thus resized to a fixed size of 640×640 for compatibility with YOLOv7, which subsequently divides the image into a $S \times S$ grid. Following this, YOLOv7 predicts multiple bounding boxes for objects of interest in the image. NMS

is applied to the predictions from the model to remove redundant detections for the same object, leaving behind a single bounding box.

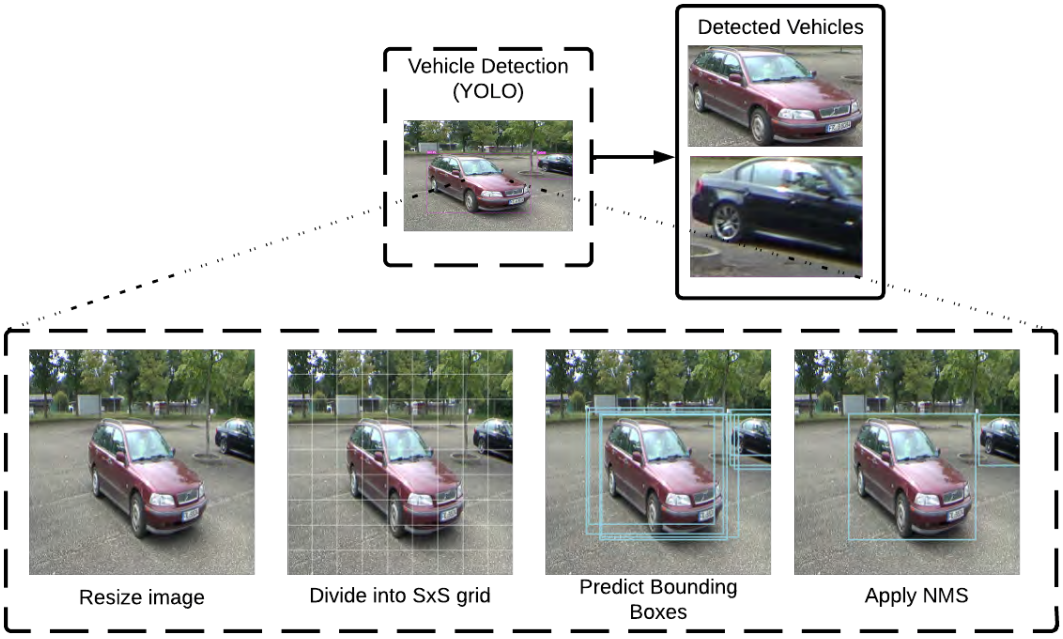


Figure 3.2: Steps for vehicle detection with YOLO.

Since vehicle detection is a single-class problem with relatively large features — constituting a large part of the input image — smaller YOLOv7 models may be beneficial or achieve comparable results to standard or extra-large variants. As described in Section 2.1, these models vary in size and computational requirements and may affect data fitting. Assessing the utility of model scaling in YOLOv7 for vehicle detection is evaluated by comparing the performance of YOLOv7-tiny with standard-sized YOLOv7 to evaluate whether the speed-accuracy trade-off is insignificant or provides real-time performance, including details such as whether the models overfit.

3.3 Licence Plate Localisation

For effective licence plate localisation, the initial licence plate detection result is followed by a geometric normalisation process. This ensures the creation of a consistent ROI, which is essential for subsequent steps.

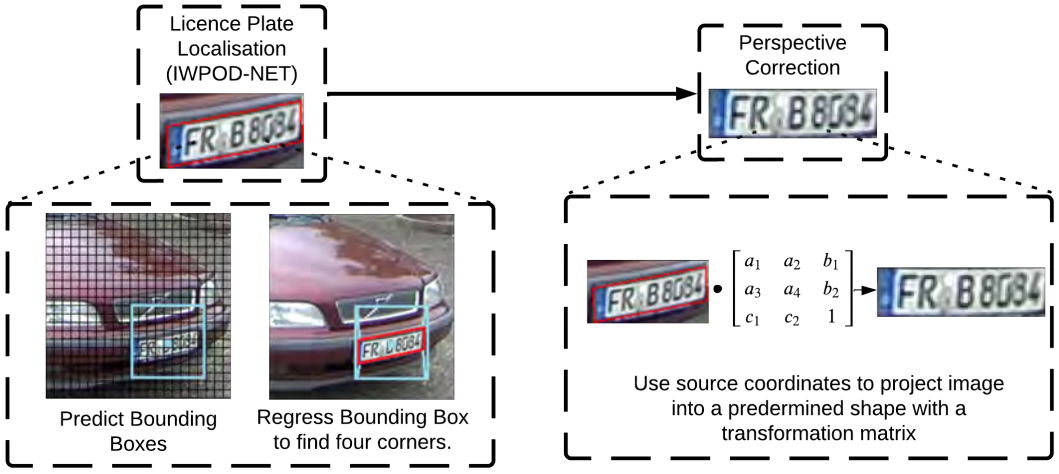


Figure 3.3: Acquiring four corner points enables perspective correction through a homography matrix.

A separate detection model was thus considered based on the requirements in Section 1.6. YOLO, R-CNN and other object detection models are restricted to bounding boxes, which lack the flexibility to apply perspective transformations on licence plates. For this task, the IWPOD-NET is employed. The IWPOD-NET uses bounding parallelograms that provide coordinates which can transform the shape for a localised ROI encapsulating the licence plate as shown in the first stage of Figure 3.3 and detailed in Section 2.2.1.2. As explained in Section 2.2.1.3, MOSI-LPD can also achieve the same task but is based on the outdated, less efficient R-CNN architecture. The significantly higher inference speed of IWPOD-NET is preferred for real-time LPR as part of an end-to-end vehicle Re-Id and retrieval system.

For each detected licence plate, four points are predicted but may not necessarily have the same perspective in camera footage. Perspective transformation utilises these coordinates to correct any distortion caused by the orientation of the licence plate. Figure 3.3 demonstrates the process; the source coordinates are mapped to the desired shape using a homography matrix to transform the angle of the licence plate to match the camera angle, resulting in a frontal view parallel to the camera. The image remains unaltered if coordinates form a rectangle, signifying the licence plate is correctly oriented. This uncommon scenario is explicitly handled within the image transformation function. The transformed image decreases distortion but does not affect its resolution, necessitating

additional correction measures.

3.4 Super-Resolution

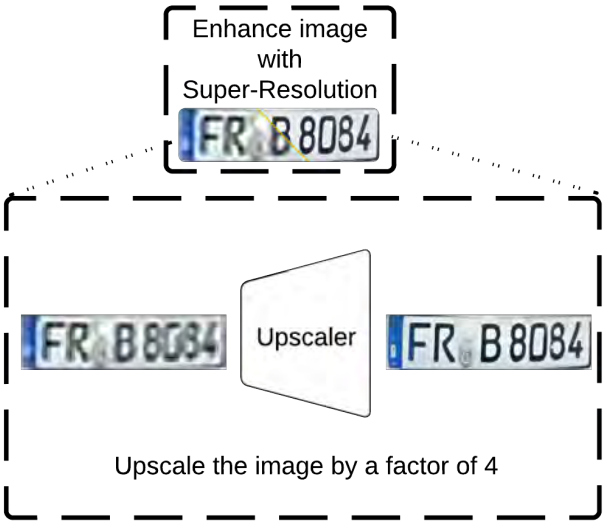


Figure 3.4: Enhancing the clarity of a licence plate image with super-resolution.

The super-resolution model is intended to upscale licence plate images to restore features that distinguish its characters from one another. This process is important due to general image quality challenges in real-world settings and perspective correction discussed in Section 2.3.2. The model needs to be trained on domain-specific data to maximise character reconstruction, as alluded to in Section 2.4. The images are upscaled to 4× their original resolution to improve their quality for effective character recognition as demonstrated in Figure 3.4.

The super-resolution component considers two models for comparison, Real-ESRGAN and DiffBIR. While SRGANs are established in the LPR domain, their limitations and adaptability compared with the state-of-the-art diffusion models remain unstudied at the time of writing. As discussed in Section 2.4.4, DiffBIR is a recently introduced diffusion-based super-resolution model primarily designed for image enhancement and unexplored in licence plate restoration for effective OCR. The similarities between DiffBIR and the Real-ESRGAN concerning image degradation for training influenced the decision to study

their performance. The superior model is selected based on the aggregated performance of the models when evaluated in Chapter 5.

3.5 Optical Character Recognition

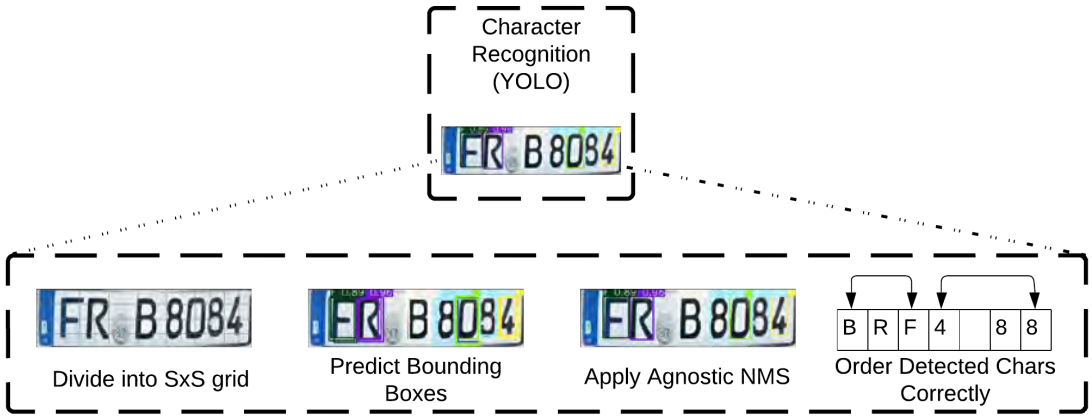


Figure 3.5: OCR process with YOLOv7.

An OCR model uses enhanced licence plate images as input for a custom YOLO architecture. It is an unconventional classifier in that it detects characters as objects and thus functions as a multiclass object detector. Following this, class-agnostic NMS is applied to the image to reduce ambiguity and extra character detections. Since this NMS approach considers all classes instead of a singular class, it is well suited to reducing problems related to the ambiguity discussed in Section 2.3. The final post-processing step orders characters correctly such that they appear in string-matching order for vehicle retrieval rather than object detection order. This step enables YOLO to work as an OCR model in the LPR domain.

Figure 3.5 denotes that a YOLOv7-based model is utilised again, similar to vehicle detection. As older YOLOv2 architectures are used for OCR in existing literature, the improved performance and enhancements in YOLOv7 are expected to yield a better OCR model. Since YOLOv7 inherits focal loss, weights are dynamically adjusted based on class imbalances, enabling directed training of weakly represented classes in the dataset. Real-world licence plate data is skewed, with certain characters appearing more frequently

than others due to region format, as discussed in Section 2.3.2. A high level of accuracy is required for distinguishing characters, and thus, the standard size YOLOv7 model was considered over the computationally efficient YOLOv7-tiny. This is supported by Kessentini *et al.* (2019) stating deeper YOLO models are preferred for finer details. Moreover, OCR cannot be recognised as a single-class problem like vehicle detection.

At the time of writing, there were limited implementations of object detection-based OCR models and related datasets for licence plates. Therefore, much manual data labelling is required to produce a dataset which can be used to train an object detection model to perform OCR, especially for licence plates. Creating such a dataset involves using publicly available data. A variety of datasets, including variations in font, position and colour, were thus considered to represent the real-world licence plate data sufficiently. Augmentations were considered based on techniques from Section 2.6.1. Such diversity increases the data pool the model can learn distinguishing features from, resulting in a system robust to unconstrained scenarios.

3.6 String Matching

String matching is proposed to identify and re-identify vehicles across different data feeds using the integrated system components for LPR. It is the final process in the end-to-end system that enables information to be extracted from a video/image.

Figure 3.6, visualises the comparisons made between a target string (row) and a prediction (column). The Levenshtein distance is calculated at each step in the table, with the final value in the bottom-right cell. The proposed method iterates this operation against all licence plates found in the target data. A string is considered a match when the distance is zero or a partial match when the distance is above a given threshold. Using Levenshtein distance, the character recognition rate is calculated as described in Section 2.3.3. Robust LPR is leveraged with the character recognition rate to identify and re-identify vehicles to be retrieved by the system.

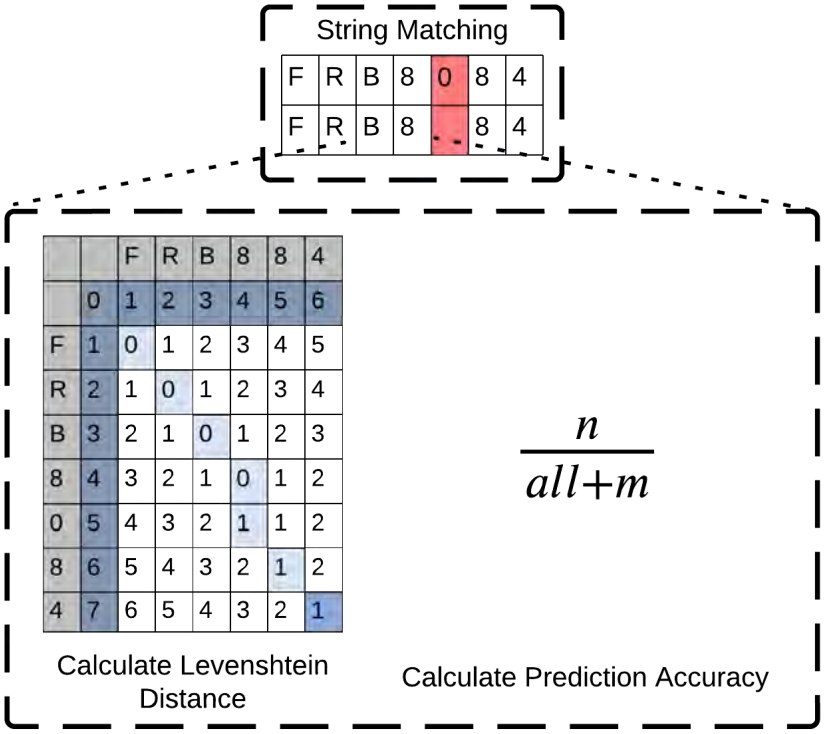


Figure 3.6: String matching with Levenshtein Distance.

3.7 Summary

This chapter summarised the proposed system with a high-level overview and subsequently detailed each component, including vehicle detection, licence plate localisation, super-resolution, OCR and string matching. Moreover, the approaches and architectures for each component are presented and chosen based on feedback from existing literature.

The IWPOD-NET is selected for its flexible bounding parallelograms and the requirements of perspective transformation, while the standard YOLOv7 is selected for OCR due to its high accuracy requirement. Two stages of the pipeline, vehicle detection and super-resolution, experiment with varying architectures. At the time of writing, it is inconclusive how diffusion-based super-resolution models compare against SRGAN models for licence plate reconstruction and if YOLOv7 model-scaling will prove beneficial for consumer-grade hardware. The methodology aims to develop a robust LPR system capable of accurately identifying and tracking vehicles in diverse surveillance scenarios based on information extracted from licence plates.

4

Implementation & Experimental Setup

This chapter details the system's implementation, including tools for annotating training data and system and training specifications. The implementation approach is detailed in line with the end-to-end system's order of execution. Lastly, the experimental setup and test models are defined for evaluation in Chapter 5.

4.1 Data Preparation

In the licence plate domain, publicly available datasets approved for research use are accessible. However, despite their availability, these datasets often require additional preparation. Extensive manual annotation was necessary for the majority of datasets collected for this research to ensure alignment with the models proposed in Chapter 3.

4.1.1 Labelling Data for YOLO

The YOLO object detector requires bounding box coordinates in the format $x, y, x+h, y+h$. All acquired datasets were manually labelled to allow the modified YOLO models to read ground truth bounding box labels. This process involved annotating the objects of interest within images with their bounding box and corresponding class. By providing these annotations during training, the YOLO model learns to predict similar bounding boxes for objects in new, unseen images, enabling the model to detect objects of interest during inference accurately.

All data labelling was completed using Roboflow[1], an online computer vision platform that provides a suite of tools especially geared towards image data annotation for ground truthing. The benefit of Roboflow is its ability to convert annotations to fit several popular object detection formats, such as YOLO. Figure 4.1 shows the platform's annotation user interface.

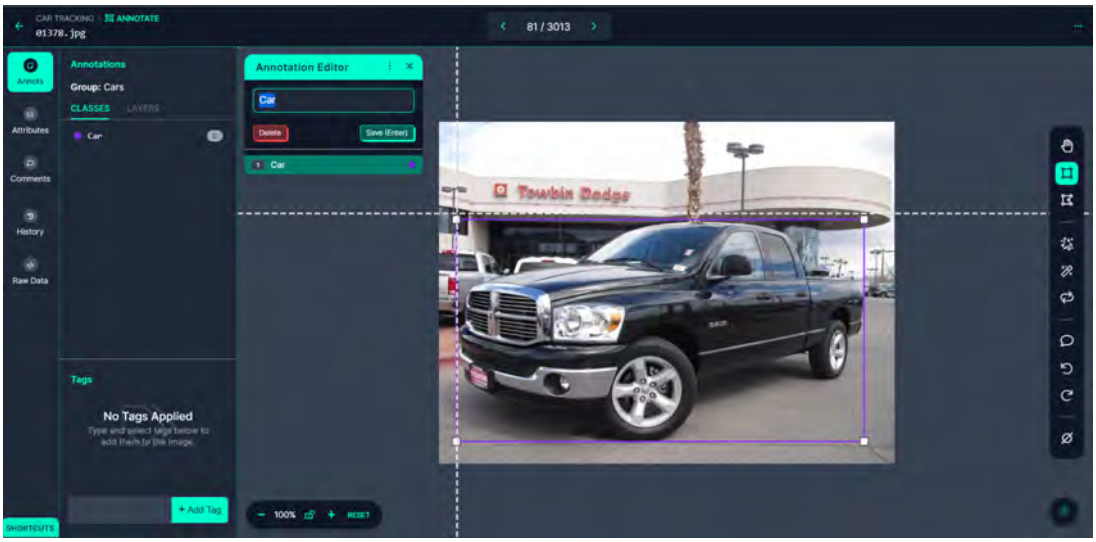


Figure 4.1: Labelling a vehicle and assigning its class in Roboflow.

The ground truth bounding boxes for dataset images were annotated following best practices, mirroring the annotation process akin to the Microsoft Common Objects in Context dataset (Lin *et al.*, 2014). Given its status as the benchmark for image standards, aligning with its annotation methodology for handling object occlusion and truncation within images of interest was essential.

The character labelling process followed a similar methodology, as YOLOv7 architectures were again utilised. Visualised in Figure 4.2, each character received individual labelling as its distinct class, enabling the object detector to identify and detect each character separately. This classification results in out-of-context OCR. As such, further post-processing was necessary for the application of these characters within the LPR domain. Further implementation specifics are discussed in Section 4.3.4.

[1]https://roboflow.com/

Figure 4.2: Distinct classes assigned to each character facilitate OCR following model training.

4.1.2 Labelling Data for the IWPOD-NET

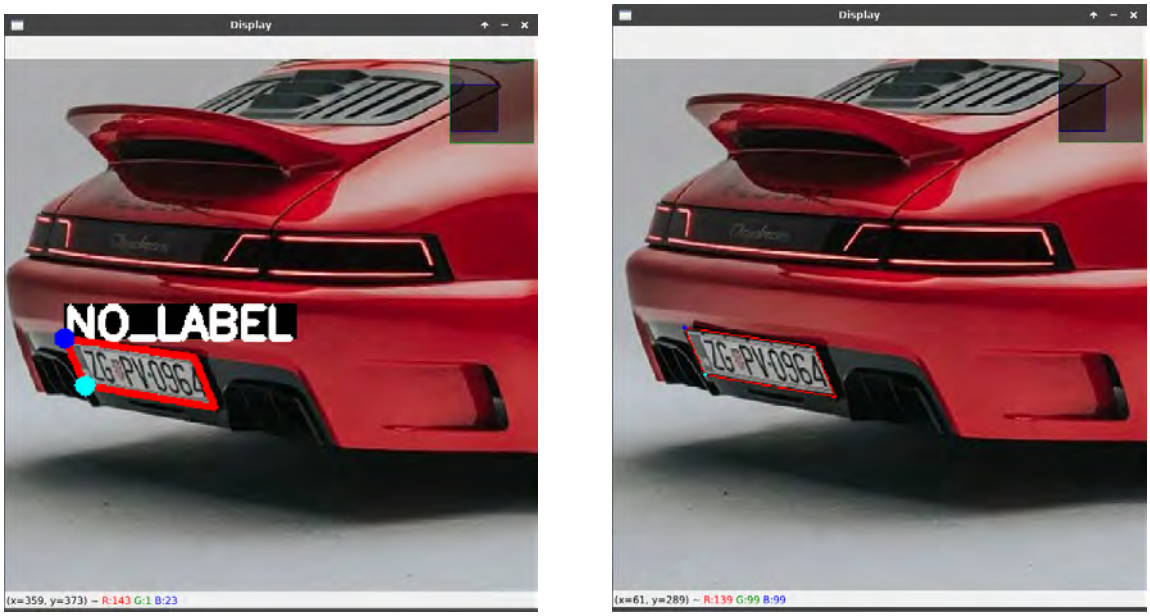
The IWPOD-NET required a custom labelling program as Roboflow does not support bounding parallelograms. A Python program (Silva and Jung, 2018) written with the open-source OpenCV library was utilised to annotate licence plates with bounding parallelograms. Furthermore, due to the limited application of IWPOD-NET in existing literature, no pre-annotated datasets supported its format. Unlike mouse-driven tools such as Roboflow, this program operates using key shortcuts. Labelling thus required the images to be passed through the command line, accompanied by various parameters like image size.

To enhance the program's efficiency and expedite the labelling process, modifications were made to the code. Primarily, adjustments were implemented to enable the program to batch process a directory of images, eliminating the need to specify image names via the command line for each image requiring annotation. Additionally, when creating bounding parallelograms, the outlines and corner points were reduced to one pixel to improve labelling accuracy, overcoming the hindrance caused by the original sizes that occluded finer details. This modification is detailed in Listing 4.1. Minor code adjustments were also necessary to transition the Python program from Python 2 to Python 3, ensuring

compatibility with the Python version installed on the system. Figure 4.3 depicts the changes in the labelling user interface post-modifications.

```
    def drawLine(self,pt1,pt2,color=(255,255,255),thickness=1):
        pt1 = self.__pt2xy(pt1)
        pt2 = self.__pt2xy(pt2)
        cv2.line(self.Idisplay,pt1,pt2,color=color,thickness=thickness)
```

Listing 4.1: The thickness of the lines were changed from three to one in the `drawLine()` parameters as thicker borders occlude the true border of the licence plate.



(a) **(b)**

Figure 4.3: The streamlined labelling process by reducing the thickness of the bounding parallelograms and removing label text which covered corners.

4.1.3 Dataset Construction

The nature of modelled data significantly influences deep learning performance. This research introduces the novel Natural Scene Licence Plate (NSLP) dataset, an evolved version of the licence plate dataset created in earlier works (Boby *et al.*, 2022). This dataset aims to train a YOLO model for OCR on licence plates with an emphasis on real-world representation. The data collection process consisted of open-source images depicting challenging weather conditions and obliquely positioned licence plates. The

initial volume of the dataset was made up of the etiquetadoOlval dataset (Hernandez, 2022b), and the chvaltrainOpenIMGS (Hernandez, 2022a), which contributed 719 images and 306 images respectively. Irrelevant classes were deleted from these datasets, so there were 36 classes A-Z and 0-9. The remainder of the dataset was populated with open-source images[2].

The dataset was implemented in this research over two stages of augmentation. The first addition to the NSLP dataset was specifically designed to increase the instances of under-represented characters. Proposed by the methodology of Al-Batat *et al.* (2022), permutations were applied to images within the dataset to increase character occurrences and positional variations. Candidate characters with low representation identified from Boby *et al.* (2022) were $[Q, O, J, K, U, Z]$. These characters were superimposed onto sufficiently represented characters only in images where they initially occurred, preserving colour consistency and ensuring the images looked as natural as possible. Some samples of the augmentations on the open-source images are shown in Figure 4.4.

(a) 5UFJ26. **(b)** JUFJ26

(c) HVK124C **(d)** HVK1K4C

(e) KBZ099Q **(f)** KBZQZKQ

Figure 4.4: Using ground truth bounding box labels, underrepresented character classes were increased by imposing them over classes with enough samples throughout the dataset.

[2]www.pexels.com

Post-augmentation, the dataset comprised 9,682 labelled characters across 2,018 images, a notable increase from the pre-augmentation dataset, which contained 6,726 character annotations across 1,702 images. Figure 4.5 shows the class frequency diagram compared with the initial values.

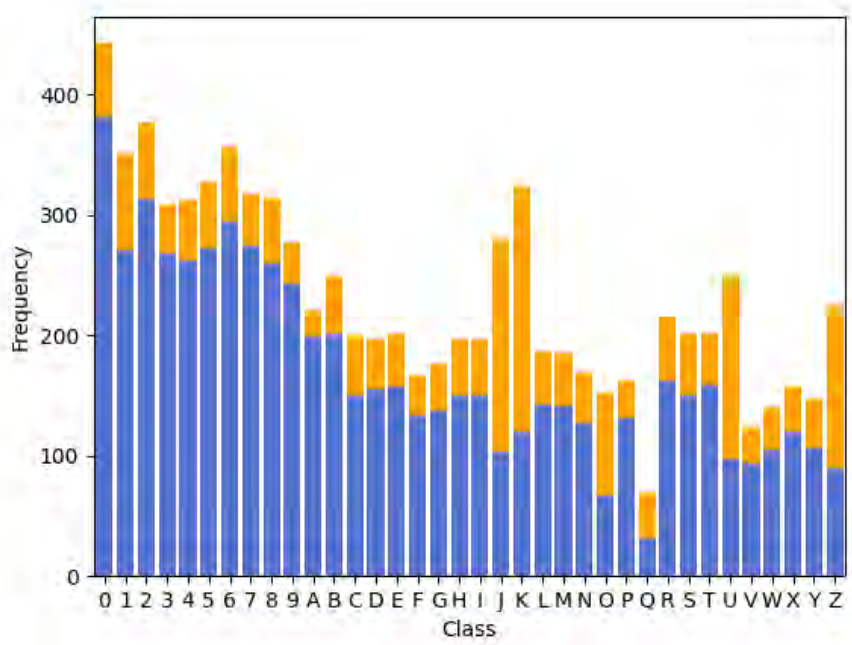


Figure 4.5: The distribution of instances within the dataset, the new distribution is represented by orange bars.

A second augmentation was applied to the resulting first augmentation stage, focusing on incorporating samples representative of challenges and degradations commonly found in unconstrained scenarios into the dataset. The four augmentations applied were:

1. Gaussian blur up to 2.5 pixels.
2. Exposure between 25% and -25%.
3. Noise up to 5%.
4. Random cutouts across the image.

These four augmentations were selected to train the model to handle scenes with blur, low light, overexposure and occlusion in real-world data.

The final version of the dataset thus contained a total of 5,132 images, with 4,761 for training and 461 for validation, all normalised to 640 × 640, the YOLOv7 input size. The assigned classes consisted of 10 numerical characters and 26 alphanumeric characters, totalling 36 classes. The dataset was included along with other publicly available datasets to form the training and evaluation data for each component described in the following section.

4.1.4 Datasets

Datasets were collated to train all the system components, each using its own subset of training data with individual splits. A breakdown of each dataset is presented in this section.

4.1.4.1 NSLP Dataset

A novel dataset was created to train object detection models for detecting Latin licence plate characters consisting of 5,132 images at 640 × 640. A more extensive list of details can be found in Section 4.1.3.

4.1.4.2 Stanford Cars

The Cars196, more commonly known as the Stanford Cars dataset (Krause *et al.*, 2013), contains a diverse collection of stationary vehicles captured from different angles featuring varying resolutions and quality. Its scale and variation make it suitable for training a vehicle detection model. The full dataset includes 16,185 images.

4.1.4.3 Medialab LPR Dataset

The Medialab LPR Dataset (Anagnostopoulos, 2010) consists of 12 sets of images categorised based on the scene's conditions. The categories included are but not limited

to blur, shadows, and dirt. The dataset has resolutions ranging from 640×480 up to 1792×1312. The dataset provides a reasonable spread of data rather than focusing on one particular scene. The images, however, are limited to one region (Greece) and only include black and white licence plates. The dataset totals only 716 images.

4.1.4.4 Croatian Licence Plate Dataset

The Croatian Licence Plate dataset (Srebrić, 2003) is a collection of 509 images of vehicles captured from the rear end. The dataset includes various samples with environmental factors and lighting conditions affecting licence plate detection. The images are categorised into the following groups: sunny, cloudy, rainy, twilight and night light. All images in this dataset have a fixed resolution of 640×480.

4.1.4.5 UFPR-ALPR Datatset

The UFPR-ALPR Dataset (Laroca *et al.*, 2018) is specifically designed for LPR tasks. The dataset comprises 4500 images captured from the front seat of a vehicle. The images form 150 distinct sequences, each focusing on a different car as the subject. The dataset maintains a consistent resolution of 1920×1080.

4.1.4.6 AOLP Database

The Application Oriented Licence Plate (AOLP) Database (Hsu *et al.*, 2012) contains 2049 images divided into three subsets: Access Control (AC), Law Enforcement (LE) and Road Patrol (RP). The licence plate images feature varying degrees of rotation, difficult orientations, as well as multiple licence plate instances in a single image.

4.1.4.7 Caltech Dataset

The Caltech dataset (Weber and Perona, 1999) comprises 126 parked vehicles with visible licence plates. While the subject vehicles are captured from the rear-end background,

vehicles in distal areas of the images have varied poses and illegible licence plates. All images feature a resolution of 896×592.

4.1.4.8 Vehicle-Rear

The Vehicle-Rear dataset aimed to facilitate vehicle identification and Re-Id via licence plates captured across various cameras. The dataset includes 10 videos recorded at a resolution of 1920×1080 at 30 FPS. Many vehicle re-identification datasets typically redact licence plates, making it challenging to assess models that rely on LPR. The dataset is ethically approved and retains all licence plates within the data, as in Brazil, licence plates are linked to vehicles, not owners, mitigating privacy concerns (De Oliveira *et al.*, 2021).

4.1.4.9 SANRAL Sample Data

Aside from public datasets, real-world sample data in the target environment was approved for viewing by the South African National Roads Agency Ltd (SANRAL). The data contains two-minute CCTV footage of vehicles before they enter a stop-and-go section on the R336 near Kirkwood in the Eastern Cape of South Africa.

4.2 System Specifications

Training and testing all the system components required dedicated hardware and software. The configuration of the machine used is as follows.

4.2.1 Hardware

Deep-learning models benefit greatly from hardware acceleration. A small dedicated server was built and used for the duration of this research, consisting of:

- CPU: AMD Ryzen™ 9 3950X 4.7 GHz 16-core
- RAM: 128GB
- GPU: NVIDIA RTX 2080Ti 11GB VRAM

4.2.2 Software

The packages and their versions are:

- Ubuntu 22.04.1 LTS
- Python 3.10.6
- TensorFlow 2.10.0
- OpenCV 4.7.0
- Pytorch 1.12.1

4.3 Model Training and Architectures

The training processes for the selected architectures for each stage of LPR are detailed in the following section.

4.3.1 YOLOv7 Vehicle Detection

The YOLOv7 (standard) and YOLOv7-tiny vehicle detection models were trained using 560 images from the Stanford Cars dataset with a validation and testing set of 120 and 58, respectively. The selected images encompassed various vehicle poses, including rear, frontal, and side views. The YOLOv7 models employed in this process were implemented using PyTorch.

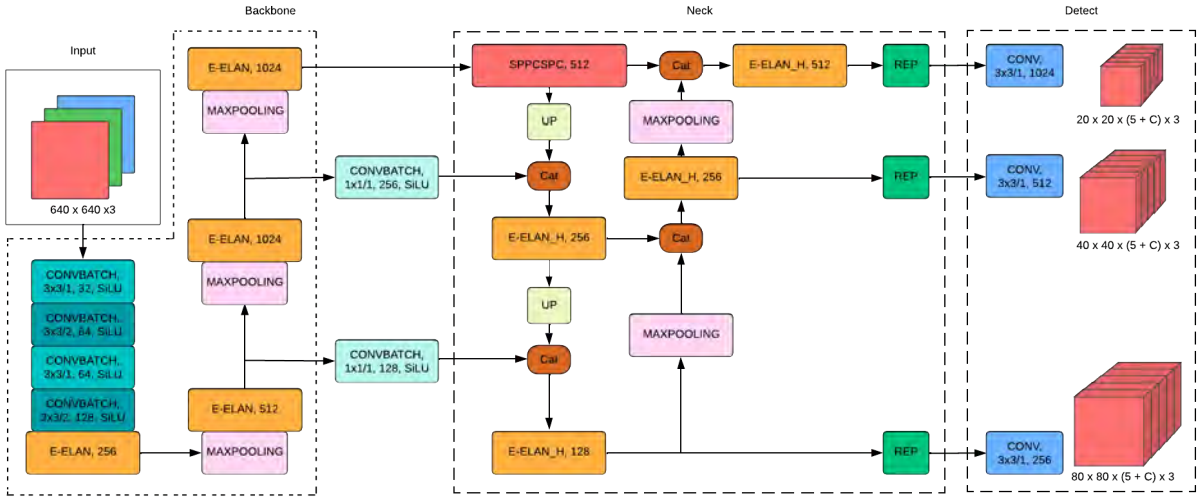


Figure 4.6: YOLOv7 Architecture. Adapted from Yan *et al.* (2022).

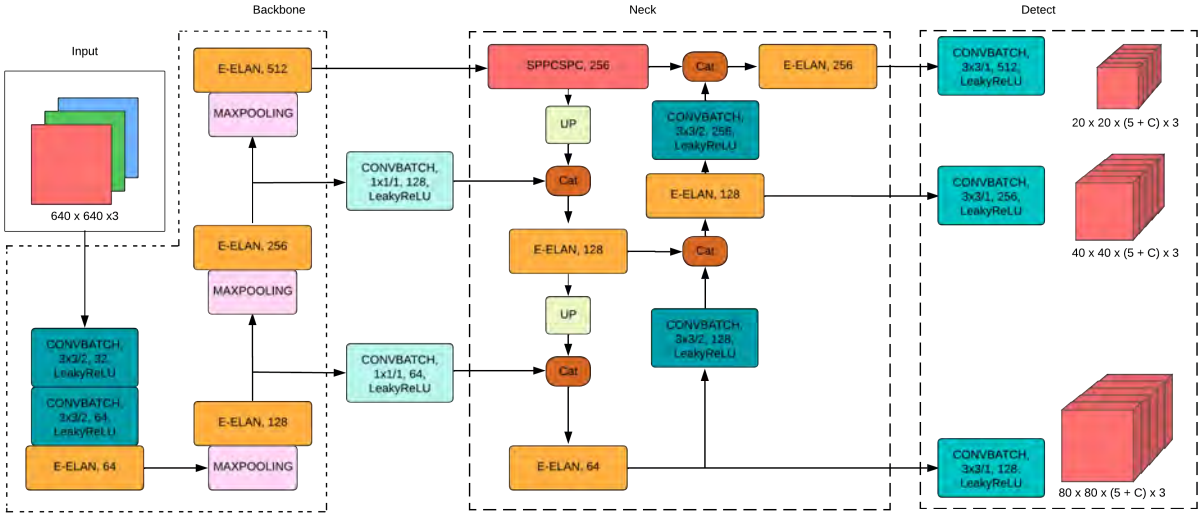


Figure 4.7: YOLOv7-tiny Architecture. Adapted from Yan *et al.* (2022), Li *et al.* (2023).

The architecture of the implemented YOLOv7 models is shown in Figures 4.6 and 4.7. The models were trained with 640×640 RGB input images and split into three separate grid sizes as per the model configuration, 20×20, 40×40 and 80×80 for the Feature Pyramid Network in the detection head (Detect). Both models were trained for 300 epochs, with computed weights saved every twenty-five epochs and generated a best weights file at the final epoch, storing the weights and achieving the top mAP during training. Weight computation for the best performance occurred after 200 epochs to ensure training stability. Optimal hyperparameters were selected based on their consistent application in existing literature (Zhang *et al.*, 2023, Liu *et al.*, 2023, Yang *et al.*, 2023):

- Learning rate: 0.01
- Optimiser: SGD
- Momentum: 0.937
- Weight Decay: 0.0005

4.3.2 Licence Plate Localisation with IWPOD-NET

The IWPOD-NET framework was implemented in TensorFlow and was trained to detect licence plates at oblique angles where characters can be distorted. The output from this model allows postprocessing stages to use the coordinates and transform the image to have the same normalised frontal angle.

The model was trained using a dataset comprising 100 images of cars with challenging-to-detect licence plates, alongside straightforward frontal and rear views. Therefore, this dataset aimed to be robust to the variety encountered in unconstrained scenarios while maintaining effectiveness on near-ideal conditions typically present in standard scenarios. Furthermore, it included diverse weather conditions and varying illumination to simulate real-world complexities, enhancing the model's adaptability to adverse conditions hindering feature detection. Furthermore, control images of vehicles without licence plates were incorporated into the training dataset to minimise false positives in the model. Ten control images from the large Stanford dataset were merged with the training set to generate the trained model.

Following Silva and Jung (2021)'s established parameters for licence plate detection, the model underwent training for 20,000 epochs, commencing with an initial learning rate of 0.001. A learning rate adjustment occurred a third into the training, reducing it by a factor of 5 based on the scheduler. The trained model would provide the coordinates of a licence plate in an image, enabling perspective transformation performed in the next step.

- Learning rate: 0.001
- Optimiser: Adam
- Batch Size: 64

The OpenCV library performed the perspective transformation using four coordinates predicted by the IWPOD-NET. The licence plate images were corrected to match the perspective of a standard licence plate by projecting the four corner points into the shape $(0,0),(w,0),(0,h),(w,h)$ using a calculated homography matrix, where w represents width and h represents height (Zhang and He, 2007). The code used to transform the four coordinates is shown in Listing 4.2.

```
pts1 = np.float32([[x_1, y_1], [x_2, y_2], [x_4, y_4], [x_3, y_3]])
pts2 = np.float32([[0, 0], [(x_dis), 0], [0,(y_dis)], [(x_dis), (y_dis)]])
M = cv2.getPerspectiveTransform(pts1, pts2) #calculate homography matrix
dst = cv2.warpPerspective(IVehicle, M, ((x_dis),(y_dis)))
```

Listing 4.2: Perspective transform applied through OpenCV.

For further evaluation of the trained model, a small script was developed to calculate the IoU for predictions. The code for this is shown in Listing 4.3. To visualise the overlap, the script produces a labelled image showing a green bounding parallelogram for the ground truth and a red parallelogram for the prediction from the model with the IoU printed at the top right corner of the detection. Figure 4.8 shows an example output from the IoU script using a sample from the Medialab data.

```
    from shapely.geometry import box, Polygon
    poly1_xy = [[x0,y0], [x1,y1], [x2,y2], [x3,y3]] #create bounding parallelogram with ground
     truth coordinates
    poly2_xy = [[x_0, y_0], [x_1, y_1], [x_2, y_2], [x_3, y_3]] # create bounding
    parallelogram with predicted coordinates
    poly_gt = Polygon(poly1_xy)
    poly_pred = Polygon(poly2_xy)

    intersection = poly_gt.intersection(poly_pred).area
    union = poly_gt.area + poly_pred.area - intersection
    IoU = intersection / union
```

Listing 4.3: A snippet of the code used to calculate the IoU of bounding parallelograms

Figure 4.8: IoU is represented by the intersection of the ground truth (green) and prediction (red).

4.3.3 Super-Resolution

A common theme in the literature is that any form of image enhancement benefits the detection or classification of objects in images. To verify this claim, two separate generative models were trained for the task to ultimately be compared and evaluated based on their time to upscale an image as well as the quality of the output.

The Real-ESRGAN was trained on 286 high-resolution images subject to various augmentations to allow for robust training models to variable data quality. These included jpeg compression, blur, noise and downsampling to allow the super-resolution model to learn the type of degradation images have besides low-resolution.

The Real-ESRGAN was trained for 1200 epochs using PSNR and SSIM to assess the model's performance during training. The parameters used for training the Real-ESRGAN are listed below.

- Learning rate: 0.0001
- Optimiser: Adam
- Gamma: 0.5
- EMA Decay: 0.999

Figure 4.9 shows the difference between the pre-trained and the finetuned weights, displaying qualitatively sharper and clearer results from the model trained with domain-specific data during preliminary tests.

(a) **(b)**

Figure 4.9: The finetuned Real-ESRGAN output shown on the right has more precise edges and shapes.

For the DIFFBIR model, the same 286 high-resolution images were used, ensuring both models were trained with the same data structure so that neither benefited from a larger or more varied dataset in their training when compared to the other.

DiffBIR was trained for 15,000 iterations. The parameters for training the model are:

- Learning rate: 0.0001
- Optimiser: Adam
- Weight Decay: 0

These super-resolution algorithms are comprehensively tested on the datasets in Chapter 5.

4.3.4 Optical Character Recognition with YOLOv7

The OCR model underwent a similar training process to the vehicle detection model, employing the YOLOv7 object detection model. The target images for OCR portrayed an undistorted view of characters. However, to enhance the model's adaptability and reduce vulnerability to perspective distortion, training images featured characters on angled licence plates. Precision in capturing the exact location of all four corner points of a licence plate by the detector was not guaranteed. Therefore, the OCR aspect was designed to

accommodate this variability, enabling the detection of characters even when not fully restored by perspective correction.

Training aimed at detecting 36 classes, encompassing numerical characters 0 - 9 and alphanumeric characters A – Z. As current licence plates exclusively feature capital letters, lowercase letters were not included in the considerations.

The training spanned 300 epochs, with the best weights saved at 25-epoch intervals after 200. Configuration parameters for OCR within the YOLO model were as follows:

- Learning rate: 0.01
- Optimiser: SGD
- Momentum: 0.937
- Weight Decay: 0.0005

As the YOLO model is not designed for OCR, post-processing corrected the output for the LPR domain. To repurpose YOLO predictions for OCR, the class name is required, along with its coordinates. An algorithm was developed to help order these characters with this information. As text on a licence plate is read from left to right, the x-axis can be used to determine the position of a character amongst other detections within an image. The leftmost character will always have the lowest x value. This assumption is correct for all regular licence plates. When square licence plates are considered, the y-axis adds an extra dimension. A square licence plate may include multiple rows, with the top row taking precedence. The highest y values represent characters in the top row. The code for this character ordering is shown in Listing 4.4.

```
1 def getX(elem):
2     return elem[1]
3
4 def getY(elem):
5     return elem[2]
6
7 def distance(list):
```

```
    return [abs(list[elem][2]-list[elem+1][2]) for elem in range(len(list)-1)]

def biggest(list):
    '''Find the biggest distance between adjacent numbers in a sorted list'''
    biggest = 0
    for elem in range(len(list)):
        distance = list[elem]
        if biggest < distance:
            biggest = distance
    return biggest

def toString(list):
    output = ""
    for row in list:
        for val in row:
            output+=val[0]
    return output

def arrangeChar(charCoords):
    #initialise empty lists to store sorted values
    sorted_Y = []
    top =[] #top row of number plate
    bottom =[] #bottom row of number plate

    charCoords.sort(key=getY)

    distance_list = distance(charCoords)
    biggest_num = biggest(distance_list)

    switch = False
    for elem in range(len(distance_list)):
        if switch:
            bottom.append(charCoords[elem])
        else:
            top.append(charCoords[elem])
        if distance_list[elem] == biggest_num and distance_list[elem] > 10:
            switch = True
    if len(bottom)== 0:
        top.append(charCoords[-1])
    else:
        bottom.append(charCoords[-1])

    top.sort(key=getX)
    bottom.sort(key=getX)
    sorted_Y.append(top)
    sorted_Y.append(bottom)
```

```
54
55      return toString(sorted_Y)
```

Listing 4.4: Character ordering algorithm implemented in Python.

The Levenshtein Python library[3] was used to calculate the distance between two strings. The implemented code is provided in Listing 4.5. It yields the count of characters necessary to convert one string into another. However, this output lacks significance on its own. Hence, the metric is integrated into the character recognition rate, enabling a more meaningful accuracy metric. Additionally, this metric can serve as a confidence value for assessing the quality of predictions during vehicle identification.

```
1   import Levenshtein as lev
2   ...
3   chars = [names[x] for x in categories.astype(int)]
4   found_string = [(chars[item], bbox_xyxy[item][0], bbox_xyxy[item][1]) for item in range(
    len(chars))]
5   final_string = org.arrangeChar(found_string)
6   lev_value = lev.distance(final_string, opt.search_string)
```

Listing 4.5: Computing the Levenshtein between two strings.

4.4 Vehicle Retrieval and Search Functionality

The collective operation of all subsystems facilitates vehicle retrieval from an image, video, or data stream within the implemented Python system. Optimal weights — those with the lowest loss values from training — were integrated into the system. To ensure peak performance, the YOLO models operated at confidence values determined by the F1 score, maximising precision and recall.

The system's functionality involves running a detect file with various parameters that govern specific aspects. Alongside the standard YOLO parameters, new parameters were introduced specifically for vehicle retrieval. The first additional parameter, `--search-string`, accepts a string, typically a licence plate number, initiating a vehicle search

[3] https://pypi.org/project/python-Levenshtein/

based on the characters within that string. The second parameter, `-query-vehicle`, accepts an image expected to contain the target vehicle for searching within the provided data source. If none of these parameters are specified, the system actively detects licence plates.

Upon successfully detecting a vehicle, the system generates an image displaying the instance where a match was located within the source data. This image is stored and can later be utilised for identification within a separate data source.

4.5 Experimental Setup

Based on the objectives set out in Section 1.6 and informed by literature, test model and metric requirements were selected to evaluate each component of the implemented system.

4.5.1 Measuring Object Detection Performance

The standard object detection methods IoU and mAP were used to evaluate the vehicle detection performance, licence plate localisation and OCR components. Moreover, precision and recall are lower-level metrics that can be used to assess per class performance of the OCR model. These metrics, established for quantitative evaluation of object detection model performance, are standard throughout literature and useful for comparing different models (Miller *et al.*, 2022).

4.5.1.1 Precision

Precision indicates what proportion of the detections made by a model are true. A high precision value means that the model has a low false positive (FP) rate, while a low precision value means that many of the detections the model made were the incorrect class. The formula for precision is defined in Equation 4.1.

$$\text{Precision } = \frac{TP}{TP + FP} \tag{4.1}$$

4.5.1.2 Recall

Recall is the number of true positives (TP), correctly identified objects, divided by the sum of true positives and false negatives (FN). The formula for recall is defined in Equation 4.2. A high recall value indicates that a model is good at finding the desired classes. In contrast, a low recall value indicates that the model cannot detect most instances in the data.

$$\text{Recall } = \frac{TP}{TP + FN} \tag{4.2}$$

4.5.1.3 F1 Score

The F1 score is a value that balances precision and recall, and it indicates the optimum confidence value for evaluating data. The formula for the F1 score is defined in Equation 4.3.

$$F_1 = 2 \cdot \frac{\text{precision } \cdot \text{ recall}}{\text{precision } + \text{ recall}} \tag{4.3}$$

4.5.1.4 Intersection Over Union

IoU represents the intersection between a ground truth $Bbox_g$ and a predicted $Bbox_p$ bounding box as a ratio. A greater area of intersection signifies a more accurate prediction from the model. It is formulated as:

$$IoU_{\text{bbox}} = \frac{Bbox_p \cap Bbox_g}{Bbox_p \cup Bbox_g} \tag{4.4}$$

$Bbox_p \cap Bbox_g$ represents the area of intersection and $Bbox_p \cup Bbox_g$ represents the area of overlap.

4.5.1.5 Mean Average Precision

Mean Average Precision measures the average precision across all object classes; it summarises a model's ability to detect objects of various classes within an image, considering both the accuracy of the detections and their consistency across different images. In the case of vehicle detection, there is only one class; hence, AP and mAP would be identical. A higher mAP correlates to better model performance. The mAP is represented by Equation 4.5 where k represents the total number of classes.

$$mAP = \frac{1}{k} \sum_{i}^{k} AP_i \tag{4.5}$$

4.5.2 Measuring Image quality

Qualitative and quantitative methods inspect outputs from two super-resolution models: the Real-ESRGAN and DiffBIR. The quantitative metrics discussed in Section 2.4.1 are described in the following sections.

4.5.2.1 SSIM

Structural Similarity is a metric used to measure the similarity between two images. By measuring the SSIM of a ground truth high-resolution image compared to a super-resolution image, it is possible to estimate how well the upscaling model performed.

4.5.2.2 PSNR

PSNR uses signals within an image to assess to what extent noise distorts an image. The ratio is used to compare the similarity between two images. PSNR is measured in decibels, where a higher value is preferable. Equation 4.6 shows how to calculate PSNR. The metric does not correlate highly with human perception but is used repeatedly in literature.

$$\text{PSNR} = \frac{10 \log_{10} \left(\text{ peakval }^2 \right)}{\text{MSE}} \tag{4.6}$$

Peakval represents the highest value in the input image (peak value).

4.5.2.3 LPIPS

Adopting more advanced metrics other than the commonly used PSNR and SSIM promotes the wide adoption of these new techniques in subsequent research, unlike traditional metrics, which rely solely on pixel-level differences. LPIPS uses learned features from deep neural networks to capture higher-level visual information and low-level pixel differences that contribute to the overall perceptual quality of images — resulting in a metric that is more comparable to human perception.

4.5.2.4 Edge Restoration Quality Assessment

ERQA measures how effectively an image has been upscaled based on its edges. The metric was developed with the hypothesis that edges are an important part of an image regarding restoration. The ERQA assessment produces a score and a visual indicator showing predicted edges versus ground truth edges. Edges in the ERQA evaluation image are colour-coordinated. A white edge represents a true positive, while blue and red edges represent false negatives and false positives.

4.5.3 Measuring OCR Performance

Character recognition rate quantitatively measures a model's performance in correctly identifying alphanumeric characters. The equation for calculating the character recognition rate is expressed as Equation 2.3 in Section 2.3.3.

$$\text{character recognition rate} = \frac{n}{all + m}.$$

The value of m is found by computing the Levenshtein distance between the recognised and ground truth strings (Equation 4.7).

$$\text{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} \text{lev}_{a,b}(i-1,j) + 1 \\ \text{lev}_{a,b}(i,j-1) + 1 \\ \text{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{4.7}$$

4.5.4 Test Models

A series of test models were developed for all system components to measure the system's ability to satisfy the requirements defined in Section 1.6 and address the research question. The tests used the defined performance metrics, allowing a comparison of existing systems and quantitative performance analysis.

4.5.4.1 Vehicle Detection

The methodology requires the model to detect vehicles before the licence plate detection step occurs. YOLOv7 and YOLOv7-tiny are compared to determine the best model for detecting vehicles in data based on inference speed and performance. The performance of the models was measured using the metrics mAP, IoU, precision and recall defined in Section 4.5.1.

4.5.4.2 Licence Plate Localisation

To test the licence plate localisation, the trained IWPOD-NET model was tested on various images to gauge the model's performance outside of validation data. Using recall and IoU for quantitative evaluation, this test model analysed the range of the licence plate detector and its ability to detect licence plates in varying scenarios.

4.5.4.3 Perspective Correction

Image correction, particularly perspective correction, is anticipated to enhance OCR performance. This test aimed to analyse how perspective correction alleviates OCR problems caused by distortion, such as ambiguity and misclassification. To quantify the effects perspective correction had on the character recognition rate, corrected licence plates and their original perspectives were passed into the OCR model to evaluate predictions. The effect of perspective correction is measured across multiple models, including CR-NET and the proposed YOLOv7 OCR model.

4.5.4.4 Super-Resolution

Super-resolution is included in the methodology to improve the visual quality of input images, as unconstrained data may have poor resolution. Two candidate super-resolution models were compared to test visual quality and performance to determine which was the most suitable to use for the end-to-end system.

The effectiveness of this method was judged by quantitatively comparing low-resolution images to their higher-resolution counterparts with the metrics defined in Section 4.5.2. Moreover, a visual inspection was performed to analyse lower-level differences between the models.

4.5.4.5 Optical Character Recognition

The OCR model had a series of tests to gauge its efficacy for the given task. The character recognition rate was used to evaluate the model's performance against labelled ground truth images. The performance of this can be an estimate of the average accuracy the system will have on real-world data. Moreover, the model's performance was compared against existing models used in literature, such as EasyOCR and TesseractOCR. The effect of super-resolution on OCR performance was investigated as well. OCR was performed on upscaled versions of the images initially used to test the OCR model, and the results

from the datasets were evaluated with the character recognition rate and the performance of the two models compared.

4.5.4.6 Vehicle Identification

Through the use of all components of the system, a target vehicle needs to be identified within data to successfully conclude that a vehicle can be identified through robust LPR. The test data included licence plate strings as identifiers of vehicles known to be present in the target videos. The strings were used as search queries on the target data, expecting the vehicle Re-Id and retrieval system to return images containing target vehicles following a licence plate match. Furthermore, Re-Id was tested using the output from the vehicle patches obtained from the identification test. The system was evaluated based on how many target vehicles it could find and the number of vehicles it failed to detect.

4.6 Summary

The approach to data preparation and construction for training the deep-learning models was detailed, providing the labelling tools used to make the images readable by the relevant models and procedures such as data augmentation were employed to allow better generalisation throughout the models and balance data for OCR.

This chapter detailed the implementation of each system component in the methodology, including the training parameters for YOLOv7, IWPOD-NET, DiffBIR and Real-ESRGAN models and the system specifications and environment for conducting training and testing. The best-performing models were saved periodically during training to ensure the optimal model selection for the end system. The models were trained with a diverse range of datasets, which were detailed to explain their contribution to a robust system.

Additionally, intentional test models and selected metrics based on literature were provided to ensure that the system was tested to meet the requirements defined in Section 1.6.

5

Results and Discussion

This chapter includes the data obtained from running the test models proposed in Chapter 4 and analyses the results to evaluate the findings.

5.1 Vehicle Detection

This section evaluates the vehicle detection model by first comparing the standard YOLOv7 and YOLOv7-tiny. Quantitative visual inspection subsequently highlights the learnt salient features followed by additional results.

5.1.1 Model Comparison

The evaluation of vehicle detection performance on the Stanford Cars dataset involved comparing the standard YOLOv7 and its smaller counterpart, YOLOv7-tiny. Given its smaller size, YOLOv7-tiny was anticipated to exhibit reduced mAP. The Stanford Cars dataset only includes cars and trucks; motorcycles were thus not trained or tested. A summary of the scores for both models is given in percentage in Table 5.1.

While the results were comparable, notable disparities were evident in the precision and recall metrics. The larger standard YOLOv7 model displayed superior performance in every metric except precision. Specifically, YOLOv7-tiny exhibited fewer false detections than the larger model, with a 3.89% difference in precision. In vehicle surveillance scenarios, a higher recall diminishes the risk of overlooking vehicles within the system, whereas

Table 5.1: Measuring inference accuracy of both YOLOv7 models based on the Stanford Cars unseen test set.

Metric	Score (%)	
	YOLOv7	***YOLOv7 tiny***
Precision	93.79	**97.68**
Recall	**91.26**	85.91
mAP@0.5	**97.03**	94.50
mAP@0.5:0.95	**85.84**	79.55

prioritising precision may elevate false negatives. The standard YOLOv7 model demonstrated a 5.35% higher recall, indicating better suitability for the surveillance and tracking tasks aligned with the model's intended operations.

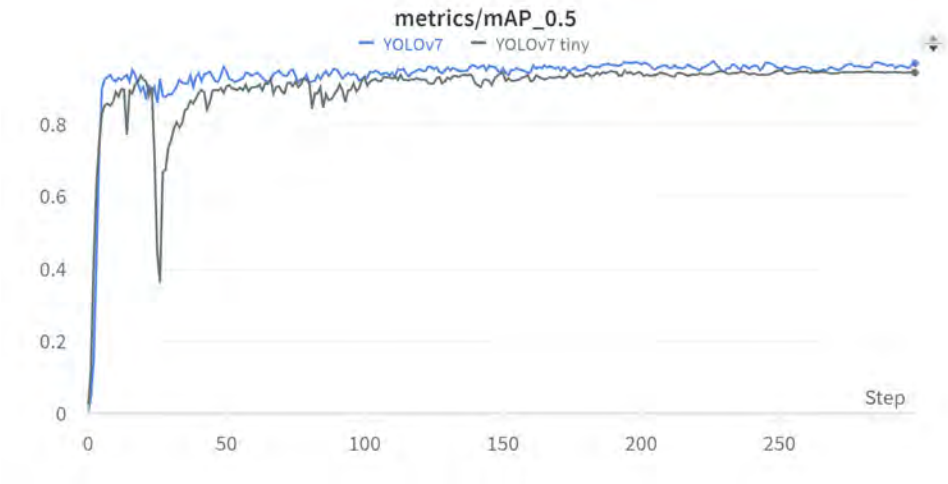


Figure 5.1: Comparison of mAP@0.5 for YOLOv7 and YOLOv7-tiny.

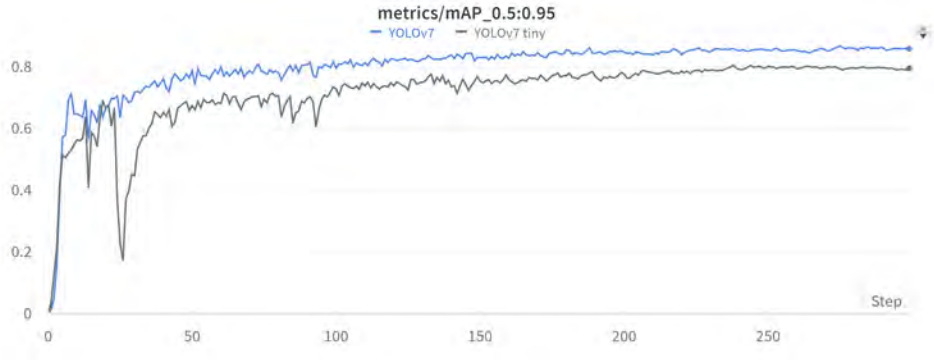


Figure 5.2: Comparison of mAP@0.5-0.95 for YOLOv7 and YOLOv7-tiny.

Analysing the mAP scores from Table 5.1 along with the graphs in Figures 5.1 and 5.2, an additional performance gap between YOLOv7 and YOLOv7-tiny becomes apparent. While similar scores at mAP@0.5 are observed between the models, YOLOv7 demonstrates increased ability when predicting objects under more stringent overlap conditions for mAP@0.5-0.95, reflecting higher accuracy across a range of scenarios.

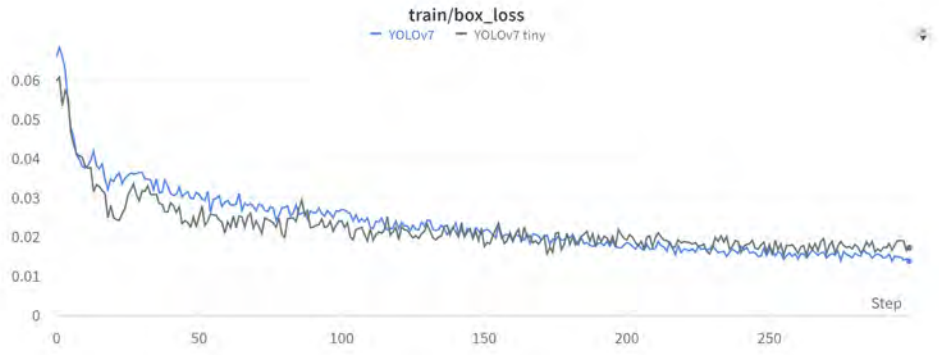


Figure 5.3: Comparison of box loss for YOLOv7 models across training epoch steps.

In Figure 5.3, a larger box loss value implies a lower IoU. As vehicle bounding boxes don't require extreme precision, the difference in IoU between the smaller and larger models is acceptable. Despite the tiny model initially exhibiting a lower loss, this changes after about epoch 180, where the larger model appears to learn the ROI better. At this point, the standard model outperforms the tiny model, which could only be considered an alternative if additional benefits beyond precision are apparent. Since the results have been largely comparable up to this point, a better conclusion can be made after measuring and comparing the FPS of both models during video stream inference.

An additional assessment was thus conducted on the SANRAL sample data videos, focusing on real-time performance measurement using FPS. The outcomes revealed a notable performance enhancement when utilising YOLOv7-tiny, indicated by a discernible difference of up to 12 FPS. Table 5.2 illustrates the total inference time for YOLOv7 and YOLOv7-tiny across a two-minute video containing ten vehicles. YOLOv7-tiny exhibited an inference time of 2.9 ms per frame, while YOLOv7 required 4.3 ms per frame. This disparity in inference speed, amounting to 1.4 ms per frame, resulted in a cumulative difference of 3.8 s in processing the entire video. It is noteworthy that both models suc-

cessfully detected all vehicles in the test footage, even in challenging scenarios involving overlapping vehicles.

Table 5.2: Inference speed of YOLOv7 and YOLOv7-tiny

Model	Average FPS	Inference (per frame)
YOLOv7	78	4.3 ms
YOLOv7-tiny	90	2.9 ms

A trade-off between accuracy and inference speed is necessary. Based on the findings in Table 5.1, it's evident that YOLOv7-tiny performs well in terms of mAP and IoU. Considering the inference speed of YOLOv7-tiny, a surplus of 12 FPS could accelerate overall system performance when integrated with all components. However, both models operate well over 30 FPS, meeting real-time standards for LPR, as defined in Section 2.5.2. It's important to note that the results regarding configuration substitutions pertain solely to vehicle detection, with a single class to detect, simplifying the problem and narrowing the gap between YOLOv7 and YOLOv7-tiny.

5.1.2 Visualising Activations

To further analyse the performance of both models, activation maps were implemented to visualise the model's use of features in image classification. The activation maps in Figure 5.4 range from dark blue for less important regions to orange or red for the most important regions. These visualisations denote which parts of vehicle features impact the model's performance and reveal potential data or modelling deficiencies. Moreover, these activation maps allow for a direct comparison between YOLOv7 and YOLOv7-tiny on the same image, illustrating their respective detection capabilities.

The activation maps for the larger model are more precise, often highlighting only the vehicles of interest in the images. Figure 5.4c shows that the activation extends above the car, whereas the activation encompasses only the target vehicle in Figure 5.4d. This visualises how the larger model is better at detecting vehicles and supplements the difference between the models based on their IoU, observed in Figure 5.3, as bounding boxes

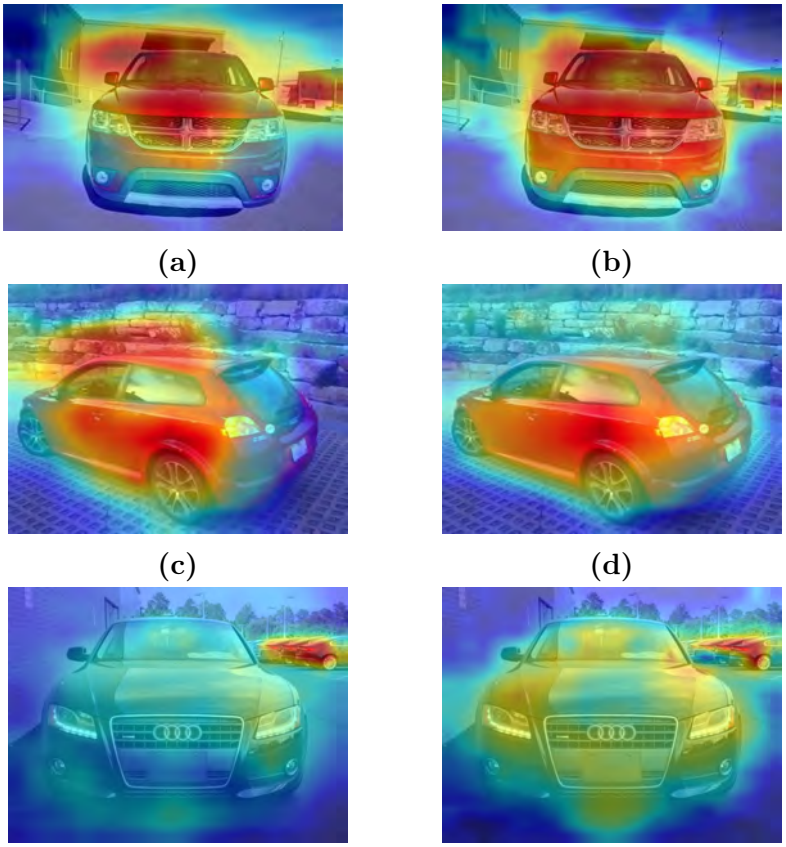


Figure 5.4: The first column shows sample activations for YOLOv7-tiny and the second column shows the activations for YOLOv7.

are created based on the activations. From inspecting Figure 5.4e and 5.4f, it can be seen that both models can detect vehicles in the background — not necessarily beneficial.

5.1.3 Additional Results

The Stanford Cars dataset provided a diverse range of vehicle poses, offering a realistic performance assessment on real-world data. On the other hand, the Caltech Cars dataset primarily contained frontal and rear views of vehicles, which were not abundant in the Stanford Cars dataset. This enabled testing for biases towards the Stanford dataset and ensured excellent detection was maintained for straightforward poses. The results for the tests on the Caltech Cars dataset are shown in Table 5.3.

Despite the relatively simpler poses, evaluating the Caltech Cars dataset is slightly more

Table 5.3: Performance of both YOLO models based on the Caltech cars dataset.

Metric	Score (%)	
	YOLOv7	***YOLOv7 tiny***
Precision	88.00	**92.68**
Recall	**82.10**	77.30
mAP@0.5	**88.60**	85.70
mAP@0.5-0.95	**67.30**	62.60

complex than the Stanford Cars dataset due to the numerous vehicles that occupy minimal space within the images, often facing occlusion from elements like fences, foliage, and other vehicles. While the larger YOLO model showcased better overall performance, there were notable disparities. Both models exhibited similar weaknesses, but the tiny model struggled more with detecting background vehicles as seen in Figure 5.5. On the other hand, the larger model presented a higher rate of false positives, resulting in reduced precision, evident in Figure 5.6. Despite these sporadic false positives, they did not significantly impact the model's efficacy. As highlighted in Section 5.1.1, prioritising higher recall in vehicle detection tasks is preferable, even if it means accommodating more false positives.

(a) **(b)**

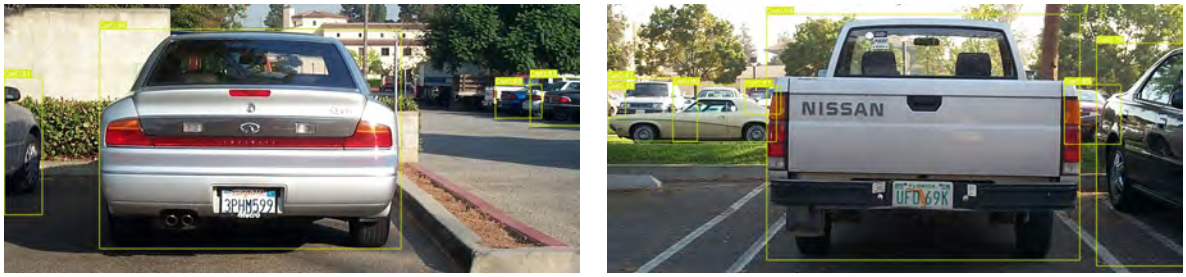
Figure 5.5: The tiny model (5.5b) failed to detect one of the distant vehicles, while the larger model was able to detect it despite it being behind a tree.

Generally, the models faced difficulties detecting vehicles further away from the camera, as shown in Figure 5.7a. More positive detections were observed as the size of the vehicles increased relative to the image. Despite successfully detecting partially visible cars occluded or truncated by other vehicles, the model had limited capabilities; significant amounts of occlusion prevented the detection of some vehicles. This is especially true in

(a) **(b)**

Figure 5.6: Comparing detections for the same image it can be seen that the larger model (5.6b) mistakes some features in the bush for distant vehicles.

crowded scenes, as demonstrated in Figure 5.7b. Many cars close to each other lead to erroneous detections or none at all. The subsequent LPR stage filters these out as they do not contain licence plates.



(a) The red vehicle, which is the furthest away from the camera, is undetected.

(b) A particularly crowded scene with many occluded vehicles.

Figure 5.7: Limitations of the vehicle detection models include crowded scenes with many vehicles.

Motorcycles have separate defining features from vehicles and were relatively underrepresented in this Caltech Cars dataset. Including motorcycles as part of a single class would affect the anchor boxes for other cars used by YOLO due to the differences in aspect ratios. Due to time constraints, motorcycle detection could not be implemented.

Another limitation of this model is that it does not differentiate between real-world cars and images of cars, for example, as seen in Figure 5.8. The model predicts a vehicle within an advertisement in the scene. Unfortunately, this serves no purpose for practical vehicle tracking and can introduce false positives during evaluation, depending on the labelling convention.

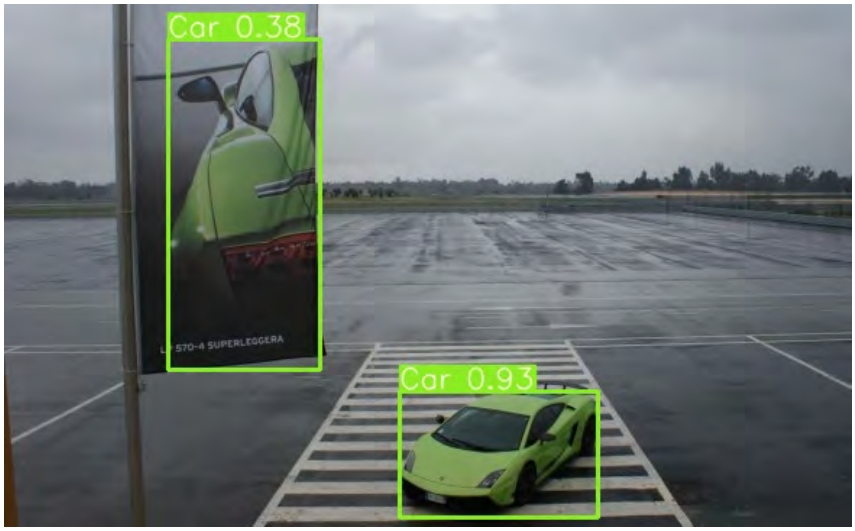


Figure 5.8: An instance of a vehicle was detected within a poster. This detection would be ignored in a practical setting.

5.2 Licence Plate Localisation

Licence plate localisation was tested using data from the Croatian Licence plate, Medialab datasets and vehicle patches from the vehicle detection tests in Section 5.1. Specially selected images from the Stanford Cars were also employed due to their acute camera angles. The varied orientations of the licence plates within the images were beneficial for testing the IWPOD-NET's prediction accuracy.

The average IoU for the IWPOD-NET, when tested on the data, was 89.15%, signifying the model could accurately predict the correct shape and size of detected licence plates. The trained IWPOD-NET model exhibited exceptional abilities to detect licence plates from a wide range of perspectives (Figure 5.9). Further capabilities are shown in Figure 5.10, where multiple licence plates were detected and localised with accurate bounding parallelograms.

The model achieved a recall of 90.9%, missing a few instances of licence plates; this high recall was attributed to the ability to locate small licence plates within an image. Smaller licence plates, however, represented a trade-off; they had less accurate predictions associated with a lower IoU. The model struggled to align the bounding parallelograms in these instances, as shown in Figure 5.11. It is important to note that such small licence

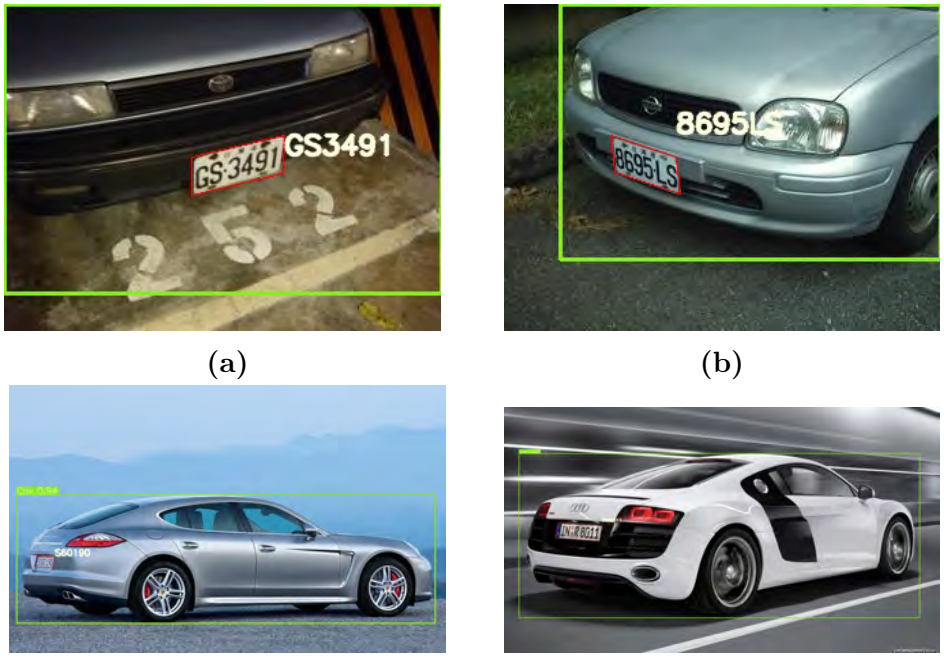


Figure 5.9: Example detections on anamorphic licence plates from the AOLP and Stanford datasets.

Figure 5.10: The IWPOD-NET model could locate all visible licence plates in the image.

plates can rarely be recovered and used to extract information, even after applying image enhancement through super-resolution.

The IWPOD-NET model successfully introduced an efficient way to capture the ROI with minimal background noise, which OCR is particularly sensitive to, solving the oblique licence plate problem. Although the model rarely struggled to achieve a good IoU, there were instances in which it had erroneously predicted skew bounding parallelograms with incorrect tilts. Such detections can result in an incorrect affine transformation at the perspective correction stage, introducing unwanted distortion and negatively impacting

Figure 5.11: The IWPOD-NET model could detect licence plates in the distance, albeit with reduced localisation accuracy.

character recognition accuracy. Examples of these predictions are in Figure 5.12, where a red parallelogram represents a prediction, and a green parallelogram indicates the ground truth. The IoU for those particular detections are shown in the top right corner. Although it is clear that there is tolerance for reduced IoU for licence plate detection, an added complexity with bounding parallelograms is that IoU can also be affected by rotation and skewing, unlike with bounding boxes.

(a) **(b)**

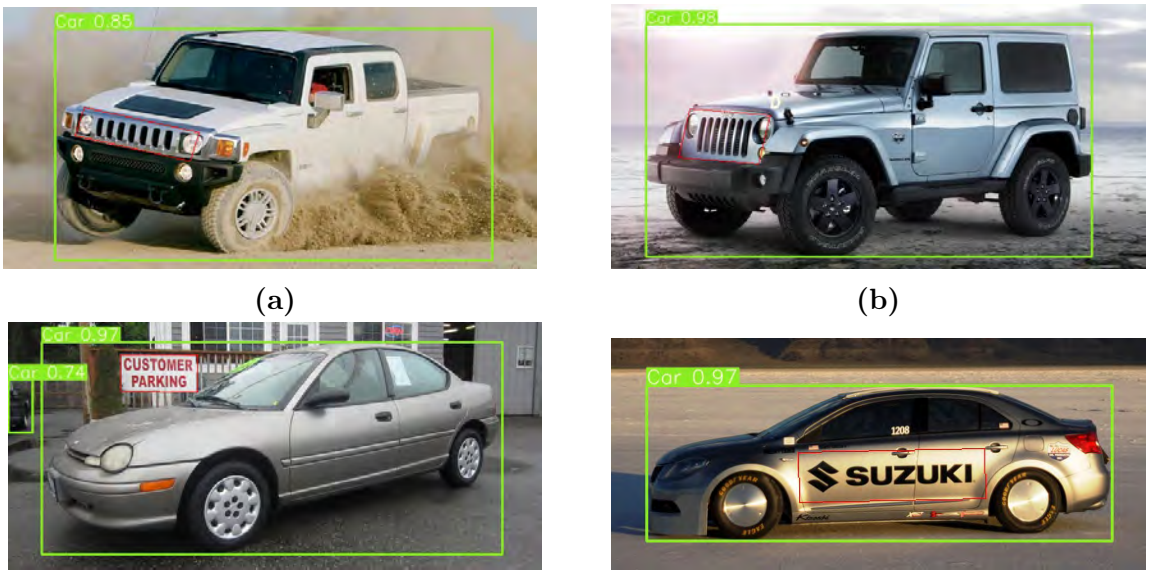
Figure 5.12: Tilted bounding parallelograms can affect perspective correction in later steps of the pipeline.

The IWPOD-NET model had some difficulties predicting licence plates accurately in very bright light. Overexposed images obscured the boundary between a car and its licence plate, particularly when they shared similar colours, eliminating edge features that aid in accurate licence plate detection. Figure 5.13 shows a typical example of this. The predicted bounding parallelogram is tilted, whereas the licence plate is horizontally aligned. Including more training data to cater for this edge case can improve the model's performance. Despite this, the IWPOD-NET model achieved a recall of 100% for the Caltech Cars dataset for all licence plates. The detections such as the one in Figure 5.13

were still counted as true positives as they were still within the IoU threshold, enabling the model to maintain high recall.

Figure 5.13: The red bounding parallelogram does not conform to the border of the licence plate.

The model occasionally produced false positives, which were often text that entered the vehicle patch or signs visually resembling licence plates. Figure 5.14 shows a few samples of these false positives taken from the Stanford Cars dataset. Some other frequent false positives included grills and headlights; these were consistently mistaken for licence plates when not part of a vehicle. Lastly, the licence plate detection model was limited by the vehicle detector; if it could not detect a car, there would be no subsequent licence plate detection either, as each stage relies on the previous stage, further highlighting the importance of high recall for vehicle detection.



(a) **(b)**

(c) **(d)**

Figure 5.14: Sample false positives from the IWPOD-NET model.

5.3 Perspective Correction and Bounding Parallelograms

Perspective correction is a system component that converts licence plate detections to a front-parallel view, the ideal viewpoint for LPR. The tests performed were developed to measure the effects of this correction on the subsequent OCR stage and thus serve as preliminary experiments for the YOLOv7 OCR model. The OCR model is run on oblique and corrected licence plates to compare effects on performance.

This experiment used a subset of the Stanford Cars dataset, containing vehicles with anamorphic licence plates of various angles and degrees of rotation. Figure 5.15 shows typical samples from the subset, illustrating the non-trivial nature of this problem. Bounding boxes were used as a control to represent uncorrected licence plates to show the effectiveness of bounding parallelograms. The ROI captured by each bounding method was passed onto the proposed OCR method to observe the effect of perspective correction on accuracy.

(a) **(b)** **(c)** **(d)**

Figure 5.15: Sample images showcasing diverse licence plate angles used to test the efficacy of perspective correction.

Figure 5.16 shows samples of licence plates before and after perspective correction. As shown in the images, perspective correction successfully aligns licence plate characters horizontally, which helps with more accurate character detection and also enables simplified

semantic arrangement using their coordinates. It can also be observed that background noise is effectively eliminated when using bounding parallelograms. As briefly mentioned in Section 5.2, excess noise outside the ROI can lead to erroneous detections. Furthermore, the perspective correction can also rectify rotation, apparent in Figure 5.16h.

(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)** **(h)**

Figure 5.16: Licence plates before and after the perspective transform is applied.

Evident from the quantitative results in Table 5.4, there was a positive correlation between perspective correction and character recognition rate. There was a 2.17% increase in accuracy for the YOLOv7 OCR model when using perspective correction as opposed to unaugmented images from bounding boxes used by standard object detectors. Improved results were also observed for comparable models from existing literature EasyOCR and CR-NET (Laroca *et al.*, 2018), with even larger 5.63% and 5.23% increases in character recognition rate. EasyOCR and CR-NET benefit the most from perspective correction, indicating that these methods have reduced capabilities on more difficult data. As the YOLO model was trained on a robust dataset, it can recognise characters at oblique

Table 5.4: The effect of perspective correction on the results of character recognition rate.

Approach	Character Recognition Rate (%)	
	Unaugmented	***Corrected***
EasyOCR	48.24	53.87
CR-NET	79.53	84.76
Proposed	**83.90**	**86.07**

angles; hence, it achieved higher initial scores on the unnormalised bounding box images. The abilities of the OCR models are contrasted in more detail in Section 5.5. Based on the results, the extent of benefits from perspective correction depends on the weaknesses of the OCR method. However, an improved character recognition rate was observed for all tested OCR methods.

It was discovered that bounding box predictions affect the output of the YOLOv7 OCR model. The character ordering algorithm does not work efficiently when lines of text are not aligned horizontally. Oriented licence plates such as the one in Figure 5.17a demonstrate this, as the licence plate within the bounding box cannot be normalised with perspective transformation, hence variations in the y-axis cause the last three letters to be detected as an additional row of text. This issue is mitigated with bounding parallelograms, as using the four corners of a licence plate, undesired rotation can be removed (Figure 5.17b).

Perspective correction increased the accuracy as ambiguous characters were correctly predicted due to distinguishing features becoming clearer. As shown in Figure 5.18b, predicted strings from normalised bounding parallelogram did not confuse the characters 'M' and 'N', but the bounding box prediction led the OCR model to confuse the two. Additional pairs showing improved prediction of ambiguous characters by normalising the licence plate are shown in Figure 5.18.

The effectiveness of the licence plate patches produced by the IWPOD-NET are limited when the model does not accurately predict all the corners of a licence plate, resulting in an inaccurate reconstruction of the front-parallel licence plate. This rare scenario may

(a) WSR652 **(b)** 652WSR

Figure 5.17: Capturing a licence plate with a bounding box makes it difficult to place characters in the correct order.

(a) 5MPH921 **(b)** 5NPH921

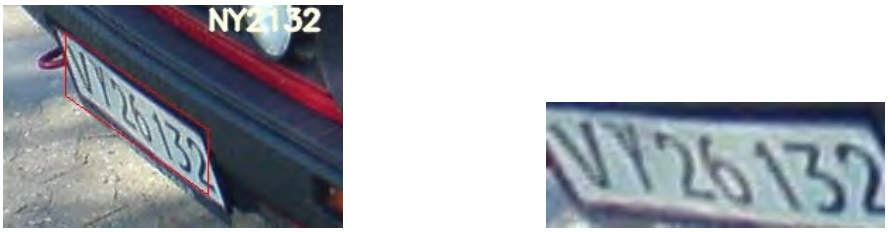
(c) 63DX765 **(d)** 6SDX765

(e) BM851LK **(f)** BN851LX

Figure 5.18: Using perspective correction improves ambiguous character predictions.

increase character classification difficulty. Figure 5.19 demonstrates an instance where the detector failed to accurately predict the lower corners of a licence plate, subsequently

affecting the perspective correction and creating an image that did nothing substantial to remove distortion.



(a) **(b)**

Figure 5.19: Due to incorrect input parameters from weak corner detection, the output image remains distorted.

Lastly, edge cases include deformed licence plates and uncommon shapes. The perspective correction method assumes that a licence plate has four corners. However, deformed licence plates do not maintain a quadrilateral shape, meaning the IWPOD-NET algorithm would need to be adjusted to detect additional vertices, thus enabling deformed licence plates to be unravelled. Moreover, these cases are challenging to detect; depending on the viewing angle, self-occlusion from bent corners may obscure information on the licence plate. Figure 5.20 shows that specific shapes are too complex to perfectly transform with just four corner points.

(a) **(b)**

Figure 5.20: The IWPOD-NET cuts off portions of the licence plate to accommodate a quadrilateral shape. While the YOLOv7 model can still read the licence plate, the characters in the image are not horizontally aligned, which can cause incorrect detections in less trivial scenarios.

5.4 Super Resolution

A number of metrics were used to measure the quality of the generated super-resolution images. To quantitatively assess the output, PSNR and SSIM were used due to their

regular use in literature. LPIPS and ERQA were included in this evaluation as they were developed to supersede the older metrics to be more effective in quantifying quality. A major issue is that all four measures require a ground truth image to assess model performance. In a real-world scenario, there is no comparable ground truth. Since surveillance and several other applications use camera footage operators, visual inspection of real-world test images can provide a good estimate of how the model will perform in the field. For this reason, qualitative results are also included.

5.4.1 Quantitative Results

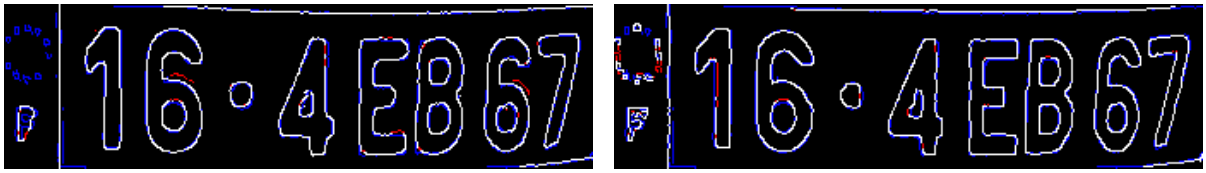
Table 5.5 summarises the performance of each model based on the performance metrics.

Table 5.5: Image quality assessment scores for the SR models, for LPIPS a lower score indicates better performance.

Super-resolution method	PSNR ↑	SSIM ↑	LPIPS ↓	ERQA ↑
Diffusion (DiffBIR)	**20.11**	**0.7354**	0.1583	**0.6547**
GAN (Real-ESRGAN)	18.39	0.6760	**0.1448**	0.6509

Between the two models, DiffBIR achieved the highest overall scores. The only metric where the ESRGAN outperformed the diffusion model was LPIPS, with a difference of 0.0135. The performance of the two super-resolution models was not substantially different, indicating similar abilities to upscale images.

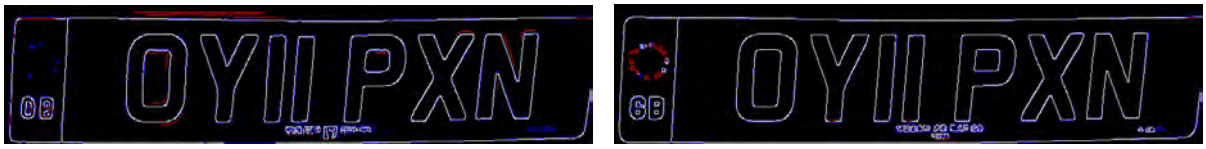
Figure 5.21 shows ERQA diagrams of the same image for both models. The image allows low-level analysis to reveal why the models achieved different scores. Looking closely at 5.21a, it can be seen that the GAN achieved better edge restoration with minimal false positives, showing that the upscaled text output closely resembles the ground truth. The DiffBIR model is better at restoring smaller, finer details. When comparing the two images, it can be seen that the top left corner in the ESRGAN result was not restored at all, as indicated by the blue outline, which denotes false negatives. The marginal difference in performance can likely be characterised by the diffusion model's ability to upscale finer details, which added to its final score.



(a) Real-ESRGAN: 0.7580 **(b)** DiffBIR: 0.7577

Figure 5.21: The ESRGAN has a slightly higher ERQA value for this image due to more accurate edge restoration and fewer false negatives.

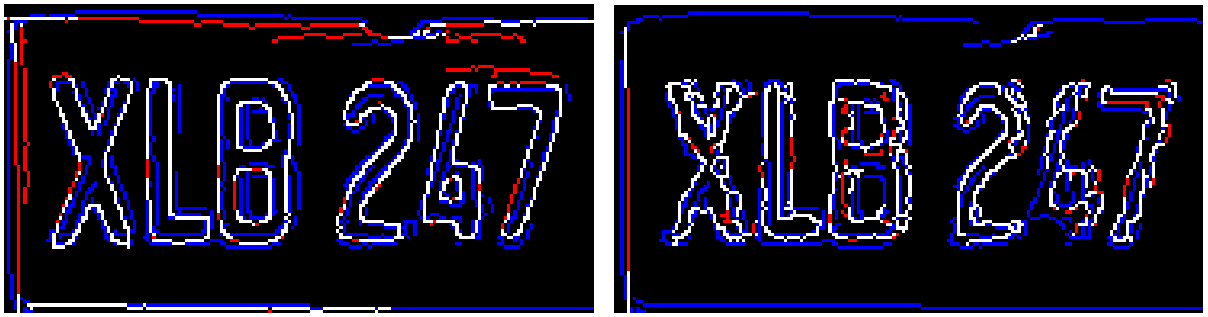
On images with higher dimensions prior to super-resolution, the diffusion model consistently produces higher-quality upscaled images. Figure 5.22 demonstrates this through comparison as the diffusion model produced more accurate edge restoration and almost no false negatives. Moving onto even lower-quality images, which represent some of the most difficult data to restore, the disparity between the abilities of the models becomes clear. A sample is shown in Figure 5.23; the diffusion model could not produce well-defined edges. Conversely, the ESRGAN produced edges that made the characters recognisable. Although the ESRGAN is better at restoring lower-resolution images, this is not reflected in the averages of the metrics presented in Table 5.5 or the ERQA in Figure 5.23b despite the results being visually more accurate.



(a) Real-ESRGAN: 0.7381 **(b)** DiffBIR: 0.8242

Figure 5.22: The DiffBIR model achieves near-perfect restoration, showing only a few red lines where the model tried to reconstruct fine details.

Based on the quantitative metrics, DiffBIR is the better super-resolution model. These metrics look at the holistic reconstruction of an image compared to its ground truth counterpart. In practice, certain aspects of the image are not important for improving OCR. As seen in Figure 5.23, the misconstruction of the borders of the licence plate penalised the ERQA score despite the ESRGAN having much better edge restoration, which is more important for OCR. PSNR, SSIM and LPIPs are not supplemented with visual results like ERQA, which is why visual inspection is still necessary.



(a) Real-ESRGAN: 0.5435 **(b)** DiffBIR: 0.5866

Figure 5.23: The comparison reveals a shortcoming of the image-quality metrics, despite having worse edge restoration 5.23b achieved a higher score.

5.4.2 Qualitative Results

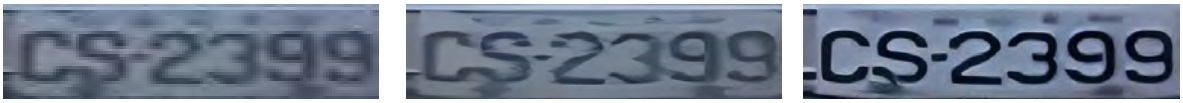
The results in this section do not contain ground truth high-resolution counterparts; therefore, quantitative analysis was not possible, instead, visual inspection was used as an alternative to gauge the capabilities of the super-resolution models. Figure 5.24 shows that results produced by the diffusion model had some superior qualities, such as preserved finite details on upscaled images and no erosion of the character edges. However, the diffusion model reproduced high-quality images inconsistently, often introducing artefacts and misrepresenting features found in low-quality images. In the example in Figure 5.24c, DiffBIR had a more authentic reconstruction of the original image. Whereas the Real-ESRGAN output, while still readable, did not retain the exact shape of the characters and instead made them thinner. For the reference image in Figure 5.24a, it is likely that the blur in the image caused erosion of the edges seen in the output image from the Real-ESRGAN. Moreover, in Figure 5.24b, the second last letter has been transformed from an '8' to a 'B', which would introduce errors in the OCR stage.

(a) Low-resolution **(b)** Real-ESRGAN **(c)** DiffBIR

Figure 5.24: Upscaled images from both models compared against the original low-resolution image.

Finetuning the ESRGAN with a set of high-quality images from the target domain can ef-

fectively increase the readability of the output. Figure 5.25 compares the output from the standard ESRGAN and the finetuned model. Finetuning the model enhanced the features that are crucial for identifying characters. Low-resolution images may have washed-out colours and less contrast between edges; this issue is addressed effectively by the finetuned model. Characters are clearer with reduced artefacts in the background and darkened text, enabling more separation between the foreground and the background. These darkened character colours make edges more prominent, effectively making OCR easier to perform in the subsequent stage.



(a) Low-resolution **(b)** Standard **(c)** Finetuned

Figure 5.25: The Real-ESRGAN adapts well to the target domain.

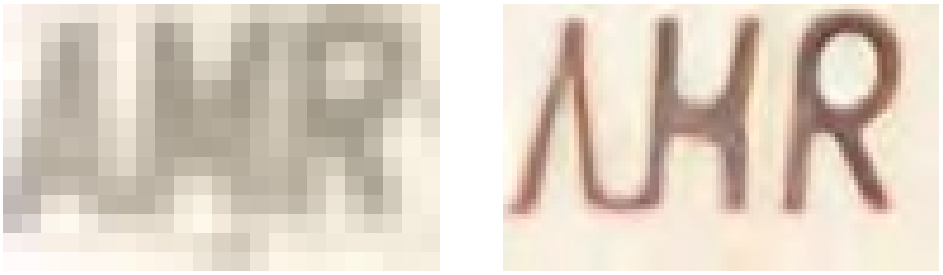
When the resolution of an image is significantly low, the diffusion model begins to predict pixels as blocks in the resulting upscaled image; this is the cause of the misconstructed edges seen in the ERQA analysis. While the Real-ESRGAN also struggles to upscale significantly low-resolution images, it is more stable, and largely unaffected by pixelation, thus producing smoother images with fewer artefacts. Moreover, the diffusion model is susceptible to aliasing, while the GAN is more robust in this regard. This is demonstrated in Figure 5.26, where aliasing is present in the original image and is handled effectively once processed by the Real-ESRGAN. Whereas, with the DiffBIR model, this data was transferred to the high-resolution output, resulting in a fuzzy image.

(a) Original **(b)** Real-ESRGAN **(c)** DiffBIR

Figure 5.26: The Real-ESRGAN eliminates low-level degradations while they are incorrectly preserved by DiffBIR.

An unintended effect of super-resolution is the conversion of characters. The effect is infrequent and requires certain conditions to be met; it occurs when the resolution of an image is considerably low and contains one or more ambiguous characters, as discussed in

Section 2.3.2. Characters with similar features become almost indistinguishable at a low resolution and thus are easily mistaken for one another. This effect is hard to mitigate, unlike the case where ambiguity stems from distortion, which has a proposed solution. In Figure 5.27, the character 'M' is transformed into an 'H' in the upscaled image. This outcome is unfavourable as it subsequently affects the OCR stage as the super-resolution effectively changes the unique identifier of a vehicle, which can lead to vehicle lookups with no matches. The Real-ESRGAN is more prone to this error, with another example of character confusion observed earlier in Figure 5.24. The frequency at which it occurs is very low and does not affect accuracy significantly enough for super-resolution to be disregarded for practical application towards robust LPR.



(a) **(b)**

Figure 5.27: Again, erosion is present in the output from the Real-ESRGAN. The side effect in this scenario is a conversion from one character to another.

The restoration metrics are used to assess the quality of reconstructed images; they do not give a clear indication of how the upscaled images affect OCR performance. Traditionally, for image restoration, it is desirable for an image to be as close as possible to the ground truth high-resolution image. However, for LPR, only a few features are required to classify the characters. As observed earlier from the ERQA results, edge restoration takes precedence over other features, such as small logos. Poor reconstruction of the smaller features reduces the scores achieved by the models as the metrics measure the distance between two images. In practice, there is some room for slight error in restoration; it is more important that the shape of the LR characters are restored to match the recognisable features of their respective class.

5.5 Optical Character Recognition

The OCR model was evaluated with object detection metrics on unseen test data as they can inform the best confidence thresholds to maximise the models' balance between precision and recall. From an object detection perspective, 95.1% mAP@0.5 and for tighter bounding boxes, 75.5% mAP@0.5-0.95 were achieved. The character recognition rate is also measured to analyse the model's LPR capabilities on unseen data.

For OCR, a 92.6% recall and 93.3% precision were obtained on the NSLP dataset, which is promising. An accurate OCR algorithm must minimise false detections, as that equates to incorrect characters in a resulting string. One wrong character could lead to an incorrect vehicle identification, and high recall increases the chances of this happening. For this reason, precision is prioritised over recall for OCR. The lowest precision was observed for the character 'O' with a precision of 31.2%. The reason for this is the complexity of differentiating 'O' and '0'. Moreover, the NSLP dataset contains Brazilian licence plate characters where 'O' and '0' are known to be identical.

The training graphs for the model are shown in Figures 5.28 and 5.29. The model was trained for 300 epochs, as mentioned in Section 4.3.4. It can be seen from the graphs that both metrics start to plateau around 250 training epoch steps, and thus, training past 300 epochs was not conducted to avoid overfitting the training data. These values indicate the modified object detection model successfully functions as a fully-fledged OCR model.

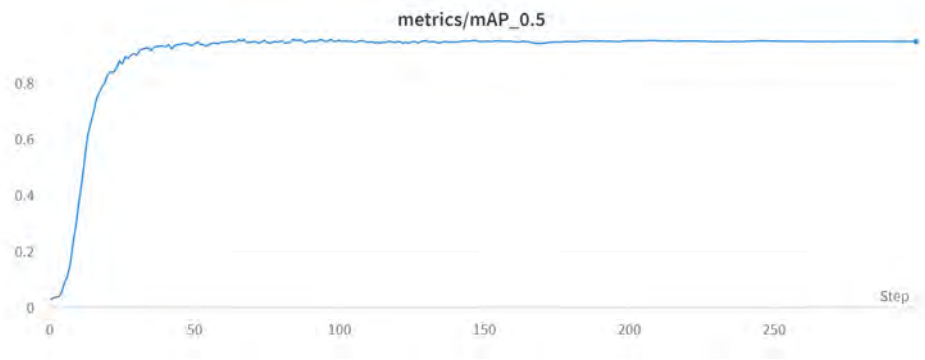


Figure 5.28: mAP@0.5 for the YOLOv7 OCR model.

Figure 5.30 and 5.31 show the training loss and validation loss on the same plot to verify

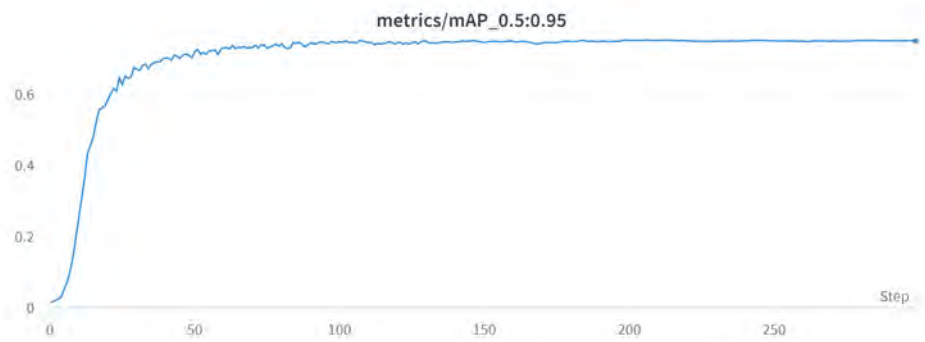


Figure 5.29: mAP@0.5-0.95 for the YOLOv7 OCR model.

no overfitting occurs prior to 300 epochs. It is clear that for training and validation, curves for class loss intersect at around 275 epochs, while the curves for box loss intersect at around 200 epochs, with some overfitting thereafter. Therefore, training halted before significant overfitting for OCR, but some overfitting was allowed for object detection. It is better to have a slightly less generalisable bounding box detector on unseen data than an inaccurate class prediction on the licence plate string — as that translates to the wrong characters being predicted.

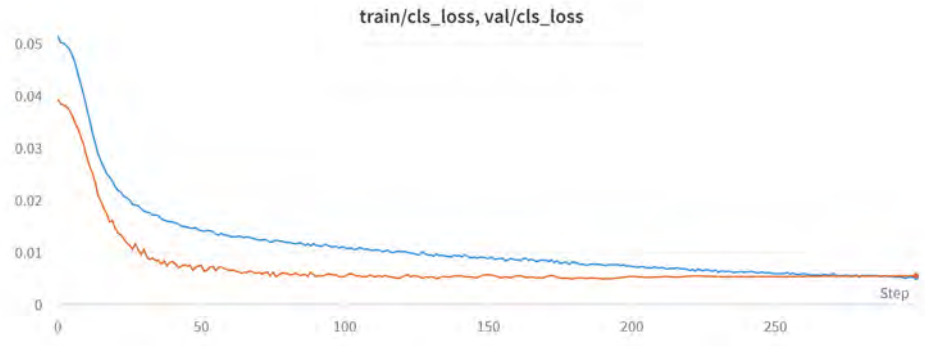


Figure 5.30: Class loss during training for the YOLOv7 OCR model.

The confusion matrix from the test data is shown in Figure 5.32, where the model predicts most characters in the set with great accuracy of above 80%. However, the model struggled particularly with the character 'O', the most commonly confused character in the alphabet [A-Z, 0-9]. On the samples in the test set, the letter was commonly confused with 'D' and '0'. Interestingly, the rate for misclassification between '0' and 'D' was much lower, meaning the character confusion was mostly one-sided. The reason for this is characters with higher representation in the dataset tend to overpower characters with

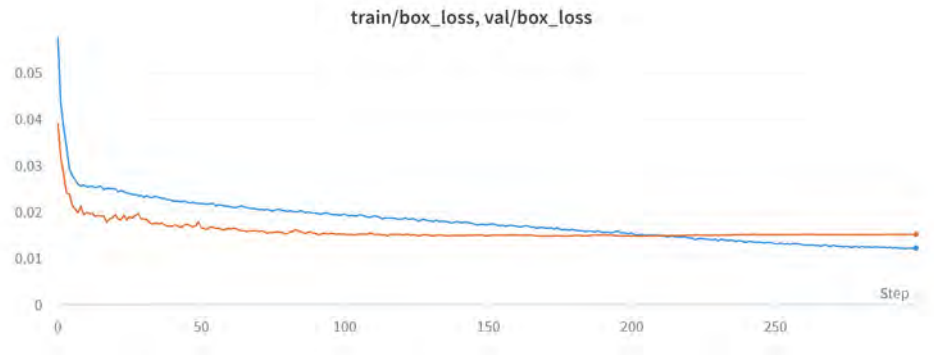


Figure 5.31: Box loss during training for the YOLOv7 OCR model.

similar features. For instance, observing the distribution in Figure 4.5, the abundance of instances for '0' resulted in 'O' being incorrectly predicted 60% of the time. One way to mitigate this would be to detect '0' and 'O' as one class, similarly to Montazzolli and Jung (2017). The difficulty of replicating this is that their data was constrained to Brazilian licence plates, allowing them to swap characters based on the licence plate format known beforehand. The proposed model is designed to perform detection and classification on a wide range of regions, which would require an extra parameter on the system to filter predictions using a regular expression or a user-specified format.

An additional confusion matrix was produced based on predictions on the unseen UFPR-ALPR dataset, as shown in Figure 5.33. Observing the data in the chart shows that for the UFPR-ALPR dataset, the model struggles to differentiate the character 'O' from '0', the same trend observed with the NSLP test set. The higher false positive rate is because UFPR-ALPR is a complete Brazilian dataset, so it is impossible for the model to differentiate between '0' and 'O' in this case. Some additional characters the model had difficulty classifying were 'G', 'I', 'F' and 'B', which are commonly confused, and this is acknowledged in the literature. There is an apparent dataset imbalance in the training data stemming from the varied frequency of certain numbers and characters caused by real-world data distribution. Despite adopting an approach to increase low-represented classes (Al-Batat *et al.*, 2022), this limitation arises because during augmentation, an underrepresented character could only superimposed on an image in which it initially occurred, ensuring that the augmented images looked natural and maintained colour consistency. A more balanced dataset is still needed to address this issue. A novel

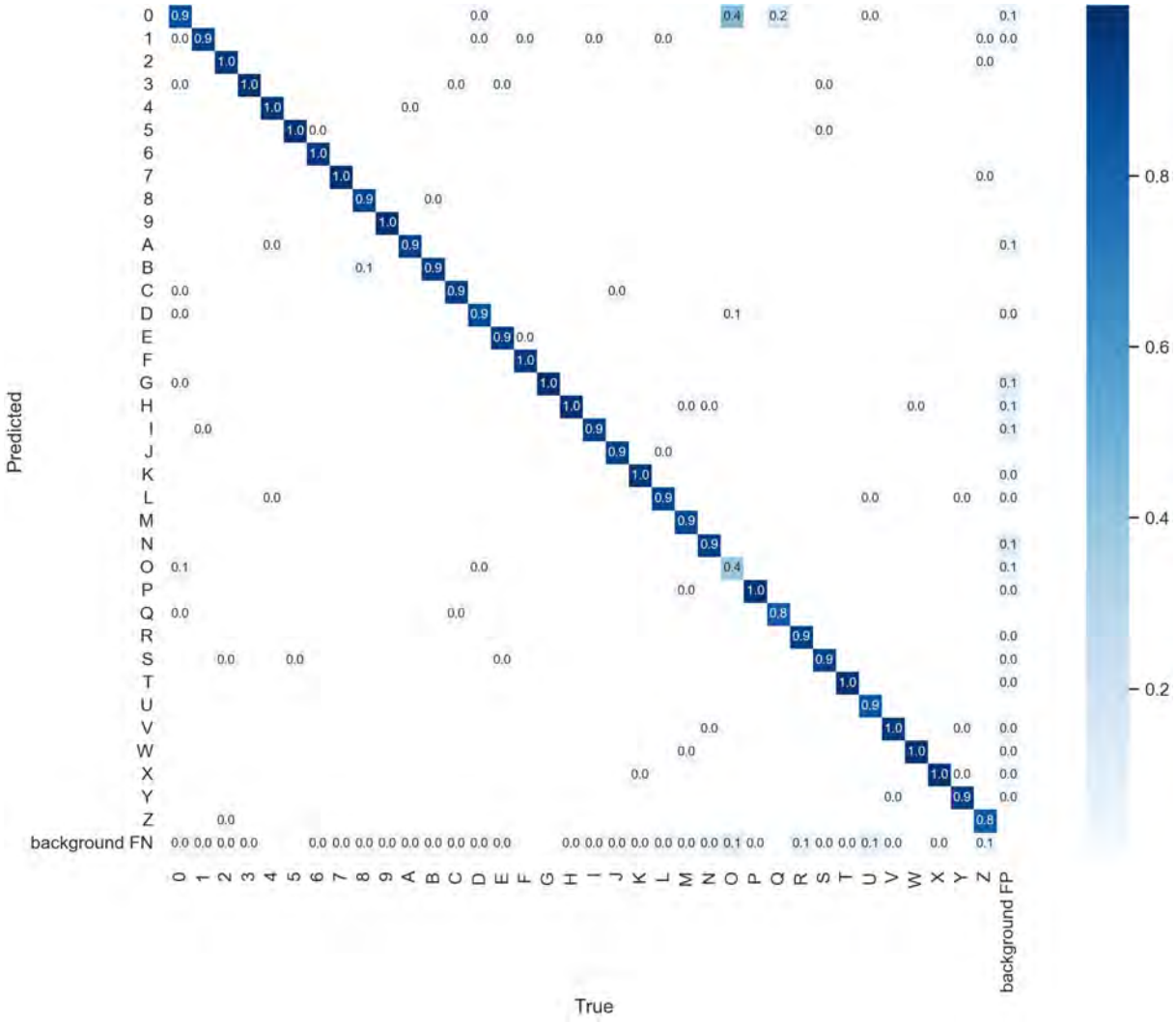


Figure 5.32: Confusion matrix for the NSLP dataset.

approach could involve image synthesis with diffusion models; however, these models still need additional development and active learning to reproduce readable text consistently and effectively.

The character recognition rate was subsequently tested on the Taiwanese AOLP dataset. Two subsets were used to test the performance of the proposed YOLOv7 OCR model. Subset AC and Subset RP: the AC set contains relatively average perspectives for licence plate detection, while the RP set contains more difficult angles. Additionally, the OCR methods were tested on the UFPR-ALPR dataset, which contains a more varied and difficult test set for LPR. The YOLO model was compared with the image processing-

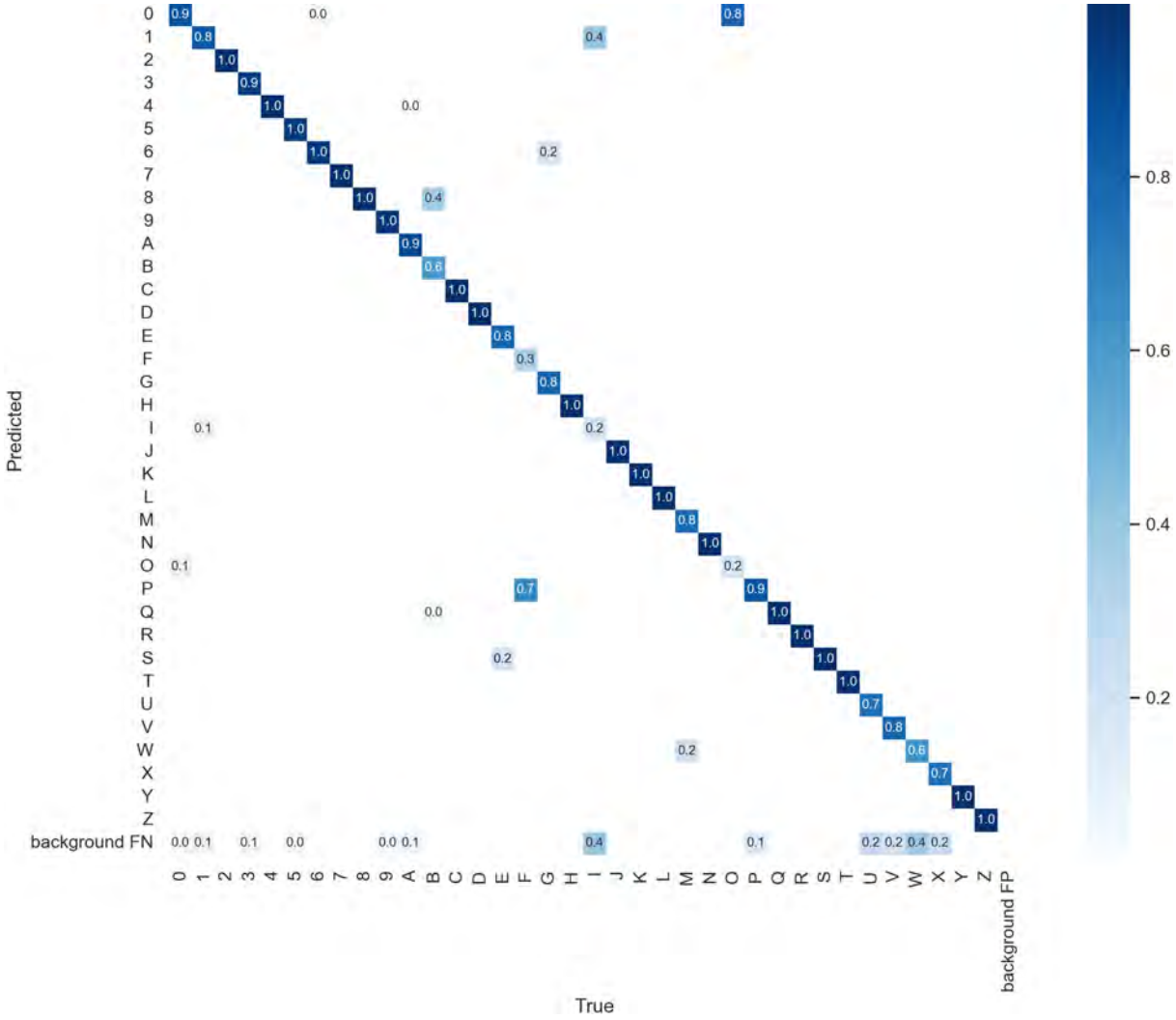


Figure 5.33: Confusion matrix for the UFPR-ALPR dataset.

based PyTesseract[1] and EasyOCR, which uses a neural network. For fairness in the results, both models were restricted to the same alphabet the YOLO model was trained on, preventing false positives in their predictions as they can predict special characters not typically found on licence plates. The results from the experiment are summarised in Table 5.6.

The results indicate that the YOLO-based OCR method is superior to both PyTesseract and EasyOCR. The YOLO model outperformed the image-processing-based PyTesseract by 28.25% and CNN-based EasyOCR by 14.15% accuracy on the AOLP AC dataset. The results again show that using an object detector is an intuitive way to create an effective

[1]A Python wrapper for TesseractOCR.

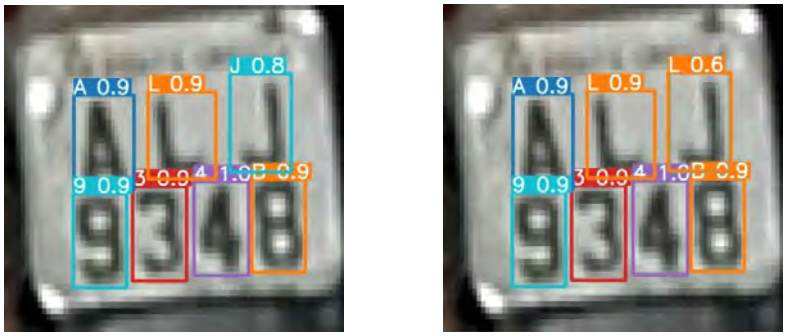
Table 5.6: Comparison of performance on datasets based on character recognition rate (%)

Approach	AOLP		UFPR-ALPR
	AC	***RP***	
EasyOCR	80.07	77.72	73.78
PyTesseract	65.97	67.23	43.47
Proposed	**94.22**	**92.02**	**88.56**

OCR system.

The proposed YOLO-based OCR model saw notable improvements from using class-agnostic NMS. Particularly for character recognition, it reduces excess or duplicate characters from being detected on a licence plate, increasing precision. In several instances, it was observed that ambiguous characters were correctly predicted when class-agnostic NMS was employed rather than the standard NMS. Figure 5.34 shows predictions from the OCR model differ whilst using both variations of NMS. In some instances, the model could even predict particularly ambiguous characters such as 'O' and '0'. An example from the AOLP dataset in Figure 5.35 demonstrates how standard NMS can unnecessarily increase the length of output strings. The character 'R' was detected thrice as different characters with similar features, such as 'B' and 'P'. With class-agnostic NMS, the additional predictions are eliminated from the final output as 'R' has the highest confidence value in that region. The approach has some drawbacks, as in some cases, the incorrect prediction may have the highest confidence, causing the correct character to be excluded from the output string. However, the frequency at which this occurred was negligible due to the OCR model's high precision.

The OCR model was flexible enough to predict licence plates with varied aspect ratios, differentiating between square and rectangular ones. This detection is possible by calculating the approximate aspect ratio of the detected licence plates from the IWPOD-NET. No additional input from the user is required to detect square licence plates, as opposed to the system created by Silva and Jung (2021). Some examples of correct detection of less typical square licence plates are shown in Figure 5.36. Note that the OCR model



(a) ALJ934B **(b)** ALL934B

(c) AXV9857 **(d)** AXV9B57

(e) AVI0949 **(f)** AVID949

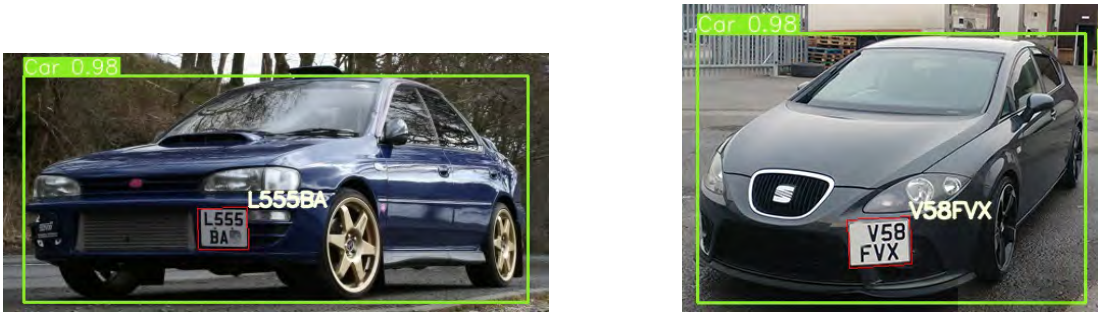
Figure 5.34: Applying class-agnostic NMS reduces confusion between ambiguous characters. The sample results, including class-agnostic NMS, are shown in the first column, and the results without it are shown in the next column.

(a) 9932RPBK **(b)** 9932RK

Figure 5.35: Without class-agnostic NMS, predicted strings from the model are longer and inaccurate as every detected character is included in the string, including overlapping predictions.

detects and orders the characters correctly.

Given the impressive results, some weaknesses affecting the performance of the OCR system were discovered. The problems include the effects of occlusion and truncation, specifically relating to confusion between characters with similar features. Truncation



(a) **(b)**

Figure 5.36: Multi-row licence plates can successfully be detected by the YOLOv7 OCR model.

within an image can leave certain characters ambiguous. For example, in Figure 5.37, truncation of the character '8' means it can either be detected as an '8' or a 'B' unless prior knowledge from previous frames is extrapolated. However, for a single image, it is challenging to verify the character, given that several characters potentially match that shape. An approach to solving this for a single image would be to convert the ambiguous characters based on the region's alphanumeric format and to prefer licence plates that are fully visible or away from the border of an image. As mentioned, the former approach is not easily applied to multi-national LPR. On the other hand, occlusion can only be handled with prior knowledge of the scene, as there is a possibility that the hidden character could be one of 36 classifiable characters. An example of occlusions that limited the effectiveness of the OCR model is shown in Figure 5.38.

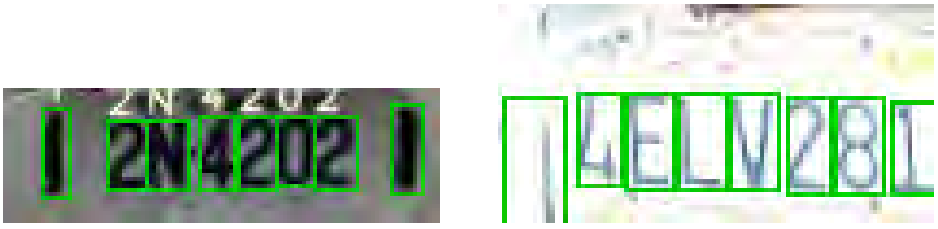
Another problem observed with the proposed OCR model is its sensitivity to noise in the background. When the ROI is not captured exactly along the borders of the licence plate, background features can confuse the OCR model, leading to erroneous detections which affect the final licence plate string prediction. Figure 5.39a shows an example from the AOLP dataset where false positives result from this occurrence. While the OCR model could be more accurate, this issue is effectively dealt with by training the IWPOD-NET to create more precise bounding parallelograms, ensuring noiseless licence plate patches are passed through to the OCR stage of the system.

(a) The first character of the licence plate is truncated because it is out of the camera's field of view.

(b) When the vehicle licence plate is in full view, there is no issue correctly classifying the character.

Figure 5.37: Two samples from the AOLP AP dataset featuring the same vehicle but with two different detected licence plates due to truncation.

Figure 5.38: The fourth character on the licence plate is blocked by the tow bar on the vehicle, resulting in a missed detection.



(a) 12N42021

(b) L4ELV281

Figure 5.39: False positive caused by features outside the ROI.

5.5.1 The Effect of Super-Resolution on Character Recognition Rate

The tests from Section 5.5 were re-run with upscaled images from both DiffBIR and the Real-ESRGAN. Table 5.7 presents the character recognition rate results from running the OCR on high-resolution images produced by the ESRGAN and DiffBIR. The initial results for the OCR model without super-resolution are available in Table 5.6.

As referenced in Section 2.3.3, Kessentini *et al.* (2019) used recall to define their character recognition rates. If the recall is calculated to match their formula, the proposed system achieved an accuracy of 98.7% compared to their 95.31% on the AOLP AC dataset. It is important to note that precision is favoured for OCR, as a higher precision results in fewer false positives caused by ambiguity.

Table 5.7: The effect of super-resolution on the results of character recognition rate.

	Real-ESRGAN			**DiffBIR**		
	AOLP		**UFPR-ALPR**	**AOLP**		**UFPR-ALPR**
Approach	***AC***	***RP***		***AC***	***RP***	
EasyOCR	**88.21**	**82.82**	**81.02**	85.22	80.96	81.00
PyTesseract	81.25	**73.14**	**65.80**	**81.82**	70.78	64.86
Proposed	**96.93**	**94.30**	**89.19**	96.13	94.02	88.94

The super-resolution models improved the character recognition rate for all tests. The Real-ESRGAN improved performance on the AOLP AC and RP dataset by 2.71% and 2.28%, respectively, and the UFPR-ALPR dataset saw only a small increase of 0.63%. Although the performance gap between the models is marginal, the results favoured the Real-ESRGAN. For example, on the AOLP dataset, there is less than a 1% difference in accuracy between the two models, with a difference of 0.8% and 0.2% for the AC and RP subsets, respectively. A similar distance between scores was observed for the more challenging UFPR-ALPR dataset. Even so, the diffusion model struggled to upscale some of its images. As mentioned in Section 5.4.2, the diffusion model tends to interpret unwanted features, such as the blocky appearance of aliased edges and translates them

to the resultant high-resolution image. Figure 5.40 is an example of the diffusion model misinterpreting the blocky appearance of the pixels in the low-resolution and forming a mosaic pattern in the upscaled image.

Another observation was that artefacts from the upscaling method could introduce false positives in the detections, affecting the accuracy of the prediction. In Figure 5.41, there is a large contrast in size between the true positives and false positives. These errors could be reduced by introducing an algorithm that suppresses bounding boxes by their relative size. As licence plate characters have the same vertical height, this can be leveraged to remove false detections based on size and location.

(a) Original **(b)** Real-ESRGAN **(c)** DiffBIR

Figure 5.40: Pixelation being misinterpreted by DiffBIR on a UFPR-ALPR sample.

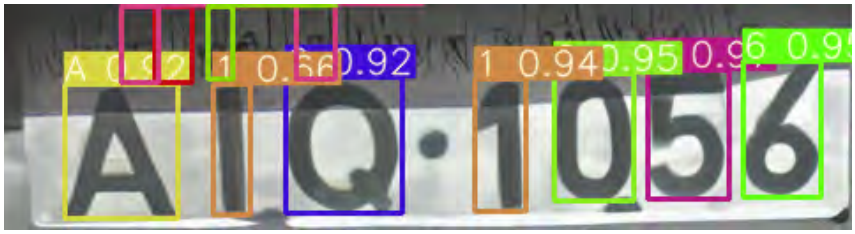


Figure 5.41: GUHGA1Q1056

The changes in accuracy were relatively small when looking at the proposed system. However, when looking at the other OCR methods, EasyOCR and PyTesseract, the benefit of super-resolution becomes more apparent. There are large gains in the character recognition rate, especially when using the Real-ESRGAN. The biggest gains were seen from PyTesseract on the UFPR-ALPR dataset with a 22.33% increase in accuracy. The initial OCR results when testing the effects of perspective transformation had a similar outcome, meaning the same conclusion from Section 5.3 can be extended to the super-resolution results. While all OCR models will see improvements from super-resolution, the biggest improvements are seen when coupling it with weaker OCR methods.

Both super-resolution models provide performance boosts to the OCR models, and the

difference between their ability to improve accuracy is small, ranging from 0.25-0.8%, with the Real-ESRGAN having slightly better performance. Moreover, DiffBIR is prone to introducing artefacts which introduce false positives to the OCR stage. The largest differentiating factor is that the diffusion model takes significantly more resources to run; therefore, it is much slower and less suited for a real-time system. Considering resources and performance, the Real-ESRGAN is much more suited to the task. Diffusion models need further optimisation and improvements before being considered for real-time tasks, which is one of the end-to-end system's requirements.

5.6 End-to-End System

The models work together to form an end-to-end vehicle search system. This section presents minor limitations and findings from running the end-to-end model on datasets from the individual component testing stages.

As mentioned in the evaluation of the vehicle detector in Section 5.1, the vehicle classification portion of the LPR pipeline was treated as a single-class problem. However, when running the end-to-end system, motorcycle licence plates captured within a vehicle bounding box were detected by the IWPOD-NET, processed and read correctly despite their shape. Figure 5.42, demonstrates with a sample from the UFPR-ALPR dataset. Currently, motorcycle detection is limited by the initial detection stage.

A shortcoming was identified when running the full system. Under specific conditions, there would be erroneous output. In the video feed, when two or more vehicles pass each other, an intersection of bounding boxes can enclose two licence plates belonging to separate vehicles, resulting in classification within the same bounding box. The system is designed to expect one licence plate per vehicle bounding box, thus when there is this overlap, the system cannot distinguish which of the detected licence plates belongs to which car. Consequently, one of the vehicles will be labelled with the wrong licence plate. The intersection of overlapping bounding boxes, including a licence plate or a portion of a licence plate, can also create duplicate detections. The IWPOD-NET processes

(a) **(b)**

Figure 5.42: The vehicle detection model is not trained with motorbikes in mind. It is limited to trucks, pickups and cars.

each vehicle patch separately, thus it cannot handle this case. Figure 5.43 shows this occurrence.

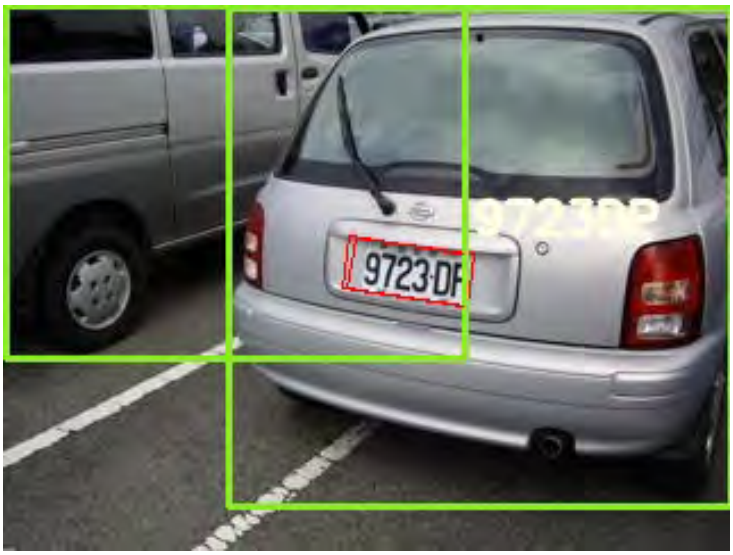
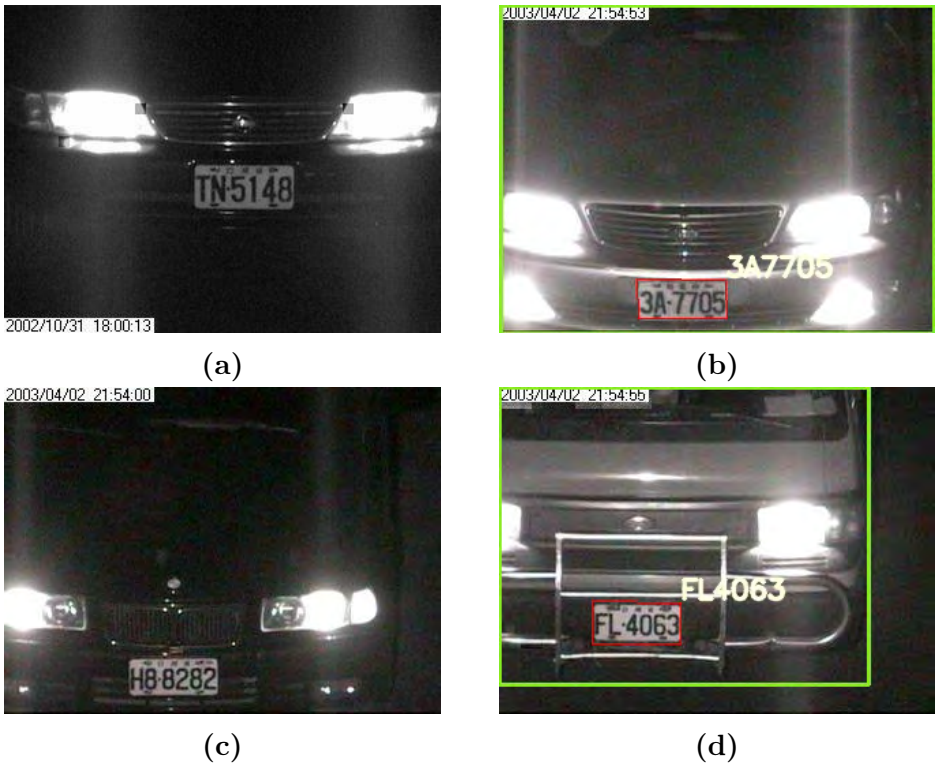


Figure 5.43: The intersection of the two-vehicle bounding boxes includes the same licence plate, resulting in a duplicate detection.

Moreover, the vehicle detection system had trouble detecting vehicles in particularly dark scenes. Figure 5.44 demonstrates this with images from the AOLP dataset. It can be seen that very few of the vehicle-defining features are visible in the image, which causes the vehicle detection to miss the vehicle, preventing any of the other models from performing any detections. Conversely, when there is more light in the image, the vehicle detector

has no difficulty detecting the image, even when it is just a partial image of the car.



(a) **(b)**

(c) **(d)**

Figure 5.44: Dark scenes where few features of a car were visible presented a challenge for the vehicle detection model.

5.6.1 Vehicle Retreival

The vehicle searching function uses a string, which can be supplied as input or extracted from an image query. Using character recognition rate and Levenshtein distance, a confidence threshold is used to validate licence plate predictions from the LPR system for matching. Enabling the model to search for a vehicle within a sequence of frames or still images. Alternatively, the system will actively detect licence plates in any image or video stream when no search query is given. It is difficult to read licence plates directly from footage as a human observer. The system makes this more efficient as it is unaffected by the frame rate and does not need to pause the video to read licence plates, enabling a less laborious automated vehicle search.

The Vehicle-Rear dataset was employed to test the vehicle identification and retrieval

capabilities of the system. The dataset consists of footage from traffic cameras and represents a real-world scenario. Five videos were used to test the full pipeline. For each video, a target vehicle known to be present was selected for a search. The input was a search string containing the desired vehicle's licence plate, and the output was an image from the video containing the target vehicle. Moreover, the vehicles were re-identified in a different camera stream using output images from the initial search as new input. The second stage tests the system's ability for re-identification. The results for the vehicle retrieval tests are shown in Table 5.8. Due to the limited vehicle poses in the Vehicle-Rear dataset, additional data was used to demonstrate the system's ability to detect vehicles in motion from an alternate angle. The feed was limited to only the samples of CCTV footage on R366 Kirkwood that were approved for viewing and supplied by SANRAL.

Query	Appearance	Result
AKE9113		Found AKE9113
ATC1182		Found ATC1189
AYI9058		Found AYI9058
AXX1612		Found AXX1612
AGZ1215		Found AGZ1215

(a) Vehicle identification.

Query	Re-Id	Result
		Found
		Found
		Found
		Not found
		Found

(b) Vehicle re-identification.

Table 5.8: Search queries followed by the appearance of identified vehicles.

The end-to-end system demonstrated effective functionality, utilising all models in the pipeline to search for a vehicle using a provided string. Evident from Table 5.8a, the

system located all but one target vehicle. The system erroneously retrieved a vehicle with the identifier ATC1189 when the query was ATC1182. The difference between the search query and the output equated to a Levenshtein distance of one. Allowing the incorrect vehicle to exceed the confidence threshold for a match. Moreover, the system could not find the correct vehicle as not all characters were classified with a high enough accuracy to confidently capture the vehicle as a positive match. Hence, only the second vehicle was found. The intended procedure when a similar licence plate is found is to return two items, effectively narrowing down the work a human operator would have to do. Similarly, at the Re-Id stage in Table 5.8b, the vehicle with the identity AXX1612 could not be re-identified and the system returned no results. To mitigate this, an adaptive confidence threshold could be implemented, where searches are re-run with a reduced threshold when no matches are found in the initial search. This approach has a higher recall but more coarse results.

Visually inspecting the other retrieved vehicle patches, it can be verified that the correct vehicles were located as the search queries matched the licence plates in the output images. All the correct sample detections were found with a 100% character recognition rate. However, accurate character recognition is limited to vehicles within a certain range; some small characters at greater distances may go undetected even with super-resolution. While some low-quality information cannot be recovered, the system can still detect and track licence plates across frames.

Figure 5.45 displays a sample image from the SANRAL test video. The system has no difficulty with South African licence plates as they have light backgrounds, dark text and an unambiguous typeface across provinces. When a vehicle matches the search query, a window displays the vehicle (Figure 5.45b). A red bounding box indicates a confidence value above 85% for the string match.

To conclude, all components of the vehicle retrieval system can work together to perform LPR, enabling further functionality such as vehicle searching, identification and re-identification. Through the enhanced LPR, as long as the vehicle's licence plate was visible, cross-data vehicle searching through re-identification was enabled.

(a) A sample of a car identified through a match against the query string.

(b) Output from the system displaying a vehicle match in a separate window to the video feed.

Figure 5.45: The state of the system when a match is found, the bounding box changes from green to red to signify a match (5.45a). Then, an image of the vehicle is stored (5.45b).

5.7 Summary

This chapter presented and discussed the results obtained from the experiments to evaluate if the proposed system addressed the research question. The experimentation was done modularly and aligned with the research objectives, testing each component of the end-to-end vehicle Re-Id and retrieval system.

It was discovered that the YOLOv7 object detector is a flexible architecture that can be extended for OCR. The results solidified that deep learning-based methods have superseded image-processing methods in LPR tasks. The YOLOv7 OCR model consistently outperformed PyTesseract, averaging a performance gain of 32.71% on all the datasets tested. Moreover, the model showed its relevance and enhanced OCR capabilities compared to another deep-learning model, EasyOCR, achieving an average character recognition rate that was 14.41% higher. While the proposed model effectively detects characters, the confusion between similar characters must still be solved. The training data was still unbalanced even after employing methods from literature for uniform representation of classes in an LPR dataset.

In the tests on image enhancement to improve OCR, perspective correction strengthened performance and reduced confusion between ambiguous characters, resulting in higher

character recognition rates. Moreso, weaker OCR methods experienced larger performance gains from image enhancement. The same effect was observed when using super-resolution prior to OCR. The super-resolution models improved the clarity of images, allowing higher confidence values when classifying characters, leading to improved performance. The system can enhance LPR capabilities with perspective transformation and super-resolution.

Some limitations of the system included reliance on accurate bounding parallelograms for perspective transformation. Occasionally, an erroneous bounding parallelogram may introduce more distortion to a licence plate. The super-resolution methods, specifically DiffBIR, achieved impressive results for decently clear images prior to super-resolution but struggled greatly with very low-resolution images. Moreover, these generative methods can alter low-resolution data and introduce false characters, an extended symptom of ambiguity made more prominent with low-resolution images.

When placed in the pipeline, the models could work together to produce a fully functional vehicle Re-Id and retrieval. The end-to-end system could take in queries as both images and strings, enabling a vehicle to be tracked across different camera feeds, thus enhancing LPR and, subsequently Re-Id, answering the research question.

6

Conclusion and Future Work

This chapter concludes the thesis, highlights the contributions made towards the research, and provides directions for future work.

6.1 Conclusion

This research combined deep-learning and image-processing methods to investigate the feasibility of searching for a vehicle within data and tracking it using the licence plate as a key point of identification. The system used ideas and effective methods from related literature to construct a robust system using YOLOv7, Real-ESRGAN and the IWPOD-NET to enhance LPR such that the output strings could be used to track vehicles.

Six tests were conducted to test the capabilities of the proposed system based on the research objectives, which enabled the research questions to be answered. Each system component was tested separately, allowing individual analysis, followed by a full run of the end-to-end system on real-world data. Each model was integrated into the pipeline to test the ability of the proposed system to complete the task from start to finish.

The YOLOv7 object detector was used for two stages in the end-to-end system. Firstly, it was employed for vehicle detection. Given that this was a single-class problem, the feasibility of using a smaller, more computationally efficient YOLOv7-tiny model was explored. The results showed that both models performed similarly with regard to precision and recall. However, when inspected more closely, it is possible to see the shortcomings of the scaled model. These include a reduced ability to detect smaller objects, as well as less accurate bounding boxes. There are performance gains with these trade-offs as the

smaller model has a faster per-frame inference time of 2.6 ms compared to 4.6 ms from the larger model, which enabled faster performance of the overall end-to-end system.

A suitably labelled and varied dataset was required to perform OCR using a modified object detection algorithm. This research introduced a new dataset specifically for scene text recognition for licence plates containing 9,682 labels for training an object detection model to detect characters. The dataset underwent augmentations such as blurring and exposure to cater to edge cases that make OCR within a scene difficult. To effectively make the class count spread more evenly, the data was augmented to increase the frequency of characters with a low representation in the dataset. Furthermore, datasets commonly used in literature were then collated to be used as test data to evaluate the performance of the OCR model based on the YOLOv7 architecture.

Only the larger YOLOv7 model was considered because OCR is a multi-class problem which requires small objects within an image to be detected as well. Using the proposed NSLP dataset for training, the model achieved favourable performance on the unseen AOLP AC and RP datasets with a character recognition rate of 94.22 and 92.02%, respectively, prior to any perspective correction or super-resolution, which are additional components to the system. These results outlined how much better the YOLOv7-based OCR model was than an image processing method PyTesseract and a similar deep-learning-based model made specifically for OCR — EasyOCR. The proposed YOLOv7 model outperformed both these models on the aforementioned dataset by highly significant margins of 26.52% and 14.23%, respectively.

A study was conducted to see the effects a perspective transformation would have on the results of OCR. It was found that there was a positive correlation between deskewing licence plate images and character recognition rate. The increase in accuracy was observed not only with the proposed YOLOv7 model but also with the other OCR models, CR-NET and EasyOCR. It was discovered that the weaker OCR methods experienced more gains when the image was corrected for distortion, allowing for improved recognition rates of up to 5.23%. The smaller gains from the stronger models were attributed to them already being more robust, allowing them to deal with some scenarios even before

the perspective correction. It was also found that the perspective correction reduced confusion amongst characters with similar features such as 'K' and 'X'. Furthermore, it was shown that perspective correction is a necessary step for OCR with an object detector as the character ordering algorithm does not work as intended when used on an angled licence plate as it creates an irregular ordering of characters along the y-axis.

The last portion of image enhancement involved evaluating the performance of the super-resolution model. This experiment followed a two-part qualitative and quantitative analysis. The qualitative analysis involved inspecting the images from two super-resolution models to see which produced more realistic images, as the evaluation metrics are not exactly equivalent to human perception. The qualitative analysis revealed that the Real-ESRGAN was much better at upscaling extreme low-resolution images characterised by pixelation. In contrast, DiffBIR struggled to upscale these images, often preserving aliasing even in the upscaled image. The diffusion model is prone to introducing artefacts into an image but effectively preserves finer details.

An additional study was conducted to see how upscaling affected the character recognition rate. It was found that super-resolution provides improved accuracy to a larger extent than perspective correction. Similar to perspective correction, it was also observed that a weaker OCR method sees greater benefits from image enhancement. In this experiment, the largest increase in OCR was 22.33%, which is a significant increase. Separate from the quality metrics, DiffBIR and the Real-ESRGAN model performed similarly in the OCR tasks, with the Real-ESRGAN marginally improving the character recognition rate compared to DiffBIR. The Real-ESRGAN was selected based on its results and lower resource consumption than the DiffBIR model.

To conclude, this research highlighted the significance of dataset selection and model architecture in developing effective LPR and vehicle searching systems. YOLOv7 demonstrated improved performance and robustness with regard to LPR. The system utilised character recognition rate and Levenshtein distance as performance metrics to assess the similarity between detected licence plates and search queries. Evaluation using diverse datasets showcased the system's ability to handle challenging conditions and achieve re-

liable licence plate detection and OCR. The system effectively enhanced LPR, allowing vehicles to be tracked across different data, answering the overarching research question. However, the study identified shortcomings in the training data, specifically the uneven distribution of characters, which affected the detection performance of underrepresented characters and the lack of ability to detect motorcycles, introducing a bottleneck that prevented other stages in the pipeline from working.

6.2 Contributions

The seven defined objectives were successfully achieved during this study:

1. Multiple public datasets were collated, allowing a fully functional LPR system to be trained. The model's capabilities were influenced by the diversity in the dataset encompassing different conditions that represent real-world scenarios.

2. Oblique licence plate detection was achieved using a state-of-the-art licence plate detector (IWPOD-NET). Bounding parallelograms produced by the model allowed free angles and orientation, allowing the perspective of oblique licence plates to be corrected.

3. OCR was successfully achieved by creating a dataset tailored towards training an object detection model (NSLP dataset). The dataset was created using practices informed by existing literature, including character permutations for increased character representation and other augmentation such as blur and exposure.

4. A robust model based on YOLOv7 was trained and extended to allow for OCR by adding an algorithm to automatically order detections from left to right in the cartesian plane.

5. Multiple experiments were run to evaluate two super-resolution models, Real-ESRGAN and DiffBIR. Modern image perception metrics were used to evaluate the two models quantitatively and qualitatively. It was concluded that the Real-ESRGAN was the most suitable based on its consistency and speed of inference.

6. An end-to-end vehicle search system was created by linking together deep learning models, forming a pipeline that could effectively detect a vehicle, find its licence plate, perform OCR and use a string matching algorithm to identify a target vehicle.

7. The end-to-end system had the functionality to take in a string as input as well as an image; from the image, the search could be run on a target dataset to see if the vehicle was present in the data, enabling re-identification from the use of pre-stored data.

The completed objectives yielded three significant contributions to the body of knowledge:

1. A novel dataset constructed to train object detection models for character detection. The dataset is specifically constructed to cater to difficult cases and class imbalances.

2. An OCR model based on the YOLOv7 architecture aimed at LPR with high precision and recall.

3. A new way to search for vehicles in a dataset unapplied in previous literature, a search string is used to identify any target vehicle.

4. Vehicle Re-Id through LPR instead of licence plate verification.

6.3 Future Work

Future work could include possible objectives such as:

1. Using generative models such as Diffusion and GANs to create training data for OCR. Such models have poor performance in text generation and, therefore, are unreliable for producing synthetic licence plates at the time of writing.

2. Experiment with high-fidelity three-dimensional scenes and models to create licence plate and vehicle datasets. The high fidelity of these models can enable the creation of varying scenarios with a high level of environmental control.

3. Improve the method used for detecting licence plates by allowing deformed ones to be deskewed; this requires more than just the four corners of a licence plate.

4. Create a system that allows cars to be identified by make and model; this requires an up-to-date, extensive dataset encompassing car brands.

5. Use segmentation for vehicle detection instead of bounding boxes to prevent duplicate detections caused by overlapping bounding boxes.

6. The use of infrared camera data is subject to availability. These images improve visibility in night-time scenarios and other low-light conditions.

7. The model could be advanced to search for more than one vehicle at a time.

References

Al-Batat, R., Angelopoulou, A., Premkumar, S., Hemanth, J., and Kapetanios, E. An end-to-end automated license plate recognition system using YOLO based vehicle and license plate detection with vehicle classification. *Sensors*, 22(23):9477, 2022.

Anagnostopoulos, C.-N. Medialab licence plate recognition database. 2010. URL http://www.medialab.ntua.gr/research/LPRdatabase.html

Anwar, S., Khan, S., and Barnes, N. A deep journey into super-resolution: A survey. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

Baek, Y., Lee, B., Han, D., Yun, S., and Lee, H. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374. 2019.

Björklund, T., Fiandrotti, A., Annarumma, M., Francini, G., and Magli, E. Robust license plate recognition using neural networks trained on synthetic images. *Pattern Recognition*, 93:134–146, 2019.

Boby, A. and Brown, D. Improving licence plate detection using generative adversarial networks. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 588–601. Springer, 2022.

Boby, A., Brown, D., and Connan, J. Iterative refinement versus generative adversarial networks for super-resolution towards licence plate detection. In *Inventive Systems and Control: Proceedings of ICISC 2023*, pages 349–362. Springer, 2023a.

Boby, A., Brown, D., and Connan, J. A practical use for ai-generated images. In *International Conference on Information, Communication and Computing Technology*, pages 157–168. Springer, 2023b.

Boby, A., Brown, D., Connan, J., and Marias, M. Exploring the incremental improvements of YOLOv7 over YOLOv5 for character recognition. In *International Advanced Computing Conference*, pages 50–65. Springer, 2022.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

Bodla, N., Singh, B., Chellappa, R., and Davis, L. S. Soft-NMS–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569. 2017.

Brisinello, M., Grbić, R., Pul, M., and Andelić, T. Improving optical character recognition performance for low quality images. In *2017 International Symposium ELMAR*, pages 167–171. IEEE, 2017.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

BussinessTech. Hijacking is on the rise in south africa – these are the cars that criminals are after. 2022.
URL https://businesstech.co.za/news/lifestyle/638799/hijacking-is-on-the-rise-in-south-africa-these-are-the-cars-that-criminals-are-after/

Casas, E., Ramos, L., Bendek, E., and Rivas-Echeverría, F. Assessing the effectiveness of YOLO architectures for smoke and wildfire detection. *IEEE Access*, 2023.

CTV. Windsor police getting $1.5 million for more automated licence plate readers. 2022.
URL https://windsor.ctvnews.ca/windsor-police-getting-1-5-million-for-more-automated-licence-plate-readers-1.6163687

de Oliveira, I. O., Fonseca, K. V., and Minetto, R. A two-stream siamese neural network for vehicle re-identification by using non-overlapping cameras. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 669–673. IEEE, 2019.

De Oliveira, I. O., Laroca, R., Menotti, D., Fonseca, K. V. O., and Minetto, R. Vehicle-rear: A new dataset to explore feature fusion for vehicle identification using convolutional neural networks. *IEEE Access*, 9:101065–101077, 2021.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Diwan, T., Anirudh, G., and Tembhurne, J. V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, pages 1–33, 2022.

Dong, C., Loy, C. C., He, K., and Tang, X. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

Du, J. Understanding of object detection based on CNN family and YOLO. In *Journal of Physics: Conference Series*, volume 1004, page 012029. IOP Publishing, 2018.

Gao, Z. and Xiang, J. Real-time location, correction and segmentation algorithm based on tilted license plate. *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, 16(4):395–406, 2023.

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. YOLOX: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

Gomes, H., Redinha, N., Lavado, N., and Mendes, M. Counting people and bicycles in real time using YOLO on jetson nano. *Energies*, 15(23):8816, 2022.

Han, J., Yao, J., Zhao, J., Tu, J., and Liu, Y. Multi-oriented and scale-invariant license plate detection based on convolutional neural networks. *Sensors*, 19(5):1175, 2019.

He, L., Liao, X., Liu, W., Liu, X., Cheng, P., and Mei, T. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020.

Henry, C., Ahn, S. Y., and Lee, S.-W. Multinational license plate recognition using generalized character sequence detection. *IEEE Access*, 8:35185–35199, 2020.

Hernandez, M. chvaltrainOpenIMGS dataset. Jan 2022a. Visited on 2022-09-29. URL https://universe.roboflow.com/mario-hernandez/chvaltrainopenimgs

Hernandez, M. etiquetadoOIval dataset. Jan 2022b. Visited on 2022-09-29. URL https://universe.roboflow.com/mario-hernandez/etiquetadooival

Herzog, F., Chen, J., Teepe, T., Gilg, J., Hörmann, S., and Rigoll, G. Synthehicle: Multi-vehicle multi-camera tracking in virtual cities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1–11. 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hsu, G.-S., Chen, J.-C., and Chung, Y.-Z. Application-oriented license plate recognition. *IEEE transactions on vehicular technology*, 2012.

Idrose, H., AlDahoul, N., Karim, H. A., Shahid, R., and Mishra, M. K. An evaluation of various pre-trained optical character recognition models for complex license plates. In *Multimedia University Engineering Conference (MECON 2022)*, pages 21–27. Atlantis Press, 2022.

Kalake, L., Wan, W., and Dong, Y. Applying ternion stream dcnn for real-time vehicle re-identification and tracking across multiple non-overlapping cameras. *Sensors*, 22(23):9274, 2022.

Kessentini, Y., Besbes, M. D., Ammar, S., and Chabbouh, A. A two-stage deep neural network for multi-norm license plate detection and recognition. *Expert systems with applications*, 136:159–170, 2019.

Kim, T.-G., Yun, B.-J., Kim, T.-H., Lee, J.-Y., Park, K.-H., Jeong, Y., and Kim, H. D. Recognition of vehicle license plates based on image processing. *Applied Sciences*, 11(14):6292, 2021.

Kirillova, A., Lyapustin, E., Antsiferova, A., and Vatolin, D. ERQA: Edge-restoration quality assessment for video super-resolution. *arXiv preprint arXiv:2110.09992*, 2021.

Kramberger, T. and Potočnik, B. LSUN-stanford car dataset: enhancing large-scale car image datasets using deep learning for usage in gan training. *Applied Sciences*, 10(14):4913, 2020.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561. 2013.

Laroca, R., Araujo, A. B., Zanlorensi, L. A., De Almeida, E. C., and Menotti, D. Towards image-based automatic meter reading in unconstrained scenarios: A robust and efficient approach. *IEEE Access*, 9:67569–67584, 2021a.

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., and Menotti, D. A robust real-time automatic license plate recognition based on the YOLO detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2018.

Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., and Menotti, D. An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intelligent Transport Systems*, 15(4):483–503, 2021b.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. *et al.* Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690. 2017.

Lee, D., Lee, S., Lee, H., Lee, K., and Lee, H.-J. Resolution-preserving generative adversarial networks for image enhancement. *IEEE Access*, 7:110344–110357, 2019.

Lee, Y., Yun, J., Hong, Y., Lee, J., and Jeon, M. Accurate license plate recognition and super-resolution using a generative adversarial networks on traffic surveillance video. In *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 1–4. IEEE, 2018.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W. *et al.* YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

Li, Z., Zhou, Y., Zhao, C., Guo, Y., Lyu, S., Chen, J., Wen, W., and Huang, Y. Design of a cargo-carrying analysis system for mountain orchard transporters based on rgb-d data. *Applied Sciences*, 13(10):6059, 2023.

Lin, M., Liu, L., Wang, F., Li, J., and Pan, J. License plate image reconstruction based on generative adversarial networks. *Remote Sensing*, 13(15):3018, 2021.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

Lin, X., He, J., Chen, Z., Lyu, Z., Fei, B., Dai, B., Ouyang, W., Qiao, Y., and Dong, C. DiffBIR: Towards blind image restoration with generative diffusion prior. *arXiv preprint arXiv:2308.15070*, 2023.

Liu, K., Sun, Q., Sun, D., Peng, L., Yang, M., and Wang, N. Underwater target detection based on improved YOLOv7. *Journal of Marine Science and Engineering*, 11(3):677, 2023.

Liu, X., Liu, W., Mei, T., and Ma, H. PROVID: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia*, 20(3):645–658, 2017.

Lucas, A., Lopez-Tapia, S., Molina, R., and Katsaggelos, A. K. Generative ad-

versarial networks and perceptual losses for video super-resolution. *IEEE Transactions on Image Processing*, 28(7):3312–3327, 2019.

Lyapustin, E., Kirillova, A., Meshchaninov, V., Zimin, E., Karetin, N., and Vatolin, D. Towards true detail restoration for super-resolution: A benchmark and a quality metric. *arXiv preprint arXiv:2203.08923*, 2022.

Miller, D., Moghadam, P., Cox, M., Wildie, M., and Jurdak, R. What's in the black box? the false negative mechanisms inside object detectors. *IEEE Robotics and Automation Letters*, 7(3):8510–8517, 2022.

Montazzolli, S. and Jung, C. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 55–62. IEEE, 2017.

Park, S.-H., Yu, S.-B., Kim, J.-A., and Yoon, H. An all-in-one vehicle type and license plate recognition system using YOLOv4. *Sensors*, 22(3):921, 2022.

Patel, C., Patel, A., and Patel, D. Optical character recognition by open source OCR tool tesseract: A case study. *International Journal of Computer Applications*, 55(10):50–56, 2012.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.
URL http://arxiv.org/abs/1506.02640

Redmon, J. and Farhadi, A. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271. 2017.

Redmon, J. and Farhadi, A. YOLOv3: An incremental improvement. 2018.

Richter, S. R., AlHaija, H. A., and Koltun, V. Enhancing photorealism enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1700–1715, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695. 2022.

Rondganger, L. These are south africa's six most hijacked cars. 2023. URL https://www.iol.co.za/news/crime-and-courts/these-are-south-africas-six-most-hijacked-cars-5fc6eb2f-9fe9-4458-a16b-e9f8943f56d0

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Santos, A. M., Bastos-Filho, C. J., Maciel, A. M., and Lima, E. Counting vehicle with high-precision in brazilian roads using YOLOv3 and deep SORT. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 69–76. IEEE, 2020.

Sara, U., Akter, M., and Uddin, M. S. Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M. *et al.* Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Shen, M. and Lei, H. Improving OCR performance with background image elimination. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 1566–1570. IEEE, 2015.

Silva, S. M. and Jung, C. R. License plate detection and recognition in unconstrained scenarios. In *Proceedings of the European conference on computer vision (ECCV)*, pages 580–596. 2018.

Silva, S. M. and Jung, C. R. Real-time license plate detection and recognition using deep convolutional neural networks. *Journal of Visual Communication and Image Representation*, 71:102773, 2020.

Silva, S. M. and Jung, C. R. A flexible approach for automatic license plate recognition in unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Smith, R. An overview of the tesseract OCR engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

Špaňhel, J., Sochor, J., Juránek, R., Herout, A., Maršík, L., and Zemčík, P. Holistic recognition of low quality license plates by cnn using track annotated data. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.

Spruck, A., Gruber, M., Maier, A., Moussa, D., Seiler, J., Riess, C., and Kaup, A. Synthesizing annotated image and video data using a rendering-based pipeline for improved license plate recognition. *arXiv preprint arXiv:2209.14448*, 2022.

Srebrić, V. Croation licence plate dataset. 2003.
URL https://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. VEEGAN: Reducing mode collapse in GANs using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

Vedhaviyassh, D., Sudhan, R., Saranya, G., Safa, M., and Arun, D. Comparative analysis of EasyOCR and TesseractOCR for automatic license plate recognition using deep learning algorithm. In *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pages 966–971. IEEE, 2022.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

Wang, C.-Y., Yeh, I.-H., and Liao, H.-Y. M. You only learn one representation: Unified network for multiple tasks. *arXiv preprint arXiv:2105.04206*, 2021a.

Wang, J., Yue, Z., Zhou, S., Chan, K. C., and Loy, C. C. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023.

Wang, X., Xie, L., Dong, C., and Shan, Y. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1905–1914. 2021b.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Loy, C. C. ESRGAN: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*. September 2018.

Wang, Z. and Bovik, A. C. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

Wang, Z., Bovik, A. C., and Lu, L. Why is image quality assessment so difficult? In *2002 IEEE International conference on acoustics, speech, and signal processing*, volume 4, pages IV–3313. IEEE, 2002.

Wang, Z., Chen, J., and Hoi, S. C. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020.

Weber, M. and Perona, P. Caltech cars 1999. Apr 1999. doi:10.22002/D1.20084.

Wei, C. Vehicle detecting and tracking application based on YOLOv5 and DeepSort for bayer data. In *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 843–849. IEEE, 2022.

Wojke, N., Bewley, A., and Paulus, D. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

Xu, Z., Yang, W., Meng, A., Lu, N., and Huang, H. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 255–271. 2018.

Yan, J., Zhou, Z., Zhou, D., Su, B., Xuanyuan, Z., Tang, J., Lai, Y., Chen, J., and Liang, W. Underwater object detection algorithm based on attention mechanism and cross-stage partial fast spatial pyramidal pooling. *Frontiers in Marine Science*, 9:1056300, 2022.

Yang, F., Zhang, X., and Liu, B. Video object tracking based on YOLOv7 and DeepSORT. *arXiv preprint arXiv:2207.12202*, 2022.

Yang, X., Zhang, B., and Lien, K.-C. SATPlate: A germany license plate detection dataset and baselines. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3329–3333. IEEE, 2023.

Yao, Y., Zheng, L., Yang, X., Naphade, M., and Gedeon, T. Simulating content consistent vehicle datasets with attribute descent. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 775–791. Springer, 2020.

Ye, S., Zhao, S., Hu, Y., and Xie, C. Single-image super-resolution challenges: A brief review. *Electronics*, 12(13):2975, 2023.

Yuan, Y., Liu, S., Zhang, J., Zhang, Y., Dong, C., and Lin, L. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710. 2018.

Zhang, L., Du, X., Zhang, R., and Zhang, J. A lightweight detection algorithm

for unmanned surface vehicles based on multi-scale feature fusion. *Journal of Marine Science and Engineering*, 11(7):1392, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595. 2018.

Zhang, Z. and He, L.-W. Whiteboard scanning and image enhancement. *Digital signal processing*, 17(2):414–432, 2007.

Zheng, Z., Jiang, M., Wang, Z., Wang, J., Bai, Z., Zhang, X., Yu, X., Tan, X., Yang, Y., Wen, S. *et al.* Going beyond real data: A robust visual representation for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 598–599. 2020.

Zuraimi, M. A. B. and Zaman, F. H. K. Vehicle detection and tracking using YOLO and DeepSORT. In *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pages 23–29. IEEE, 2021.