



YOLOv8 and Faster R-CNN Performance Evaluation with Super-resolution in License Plate Recognition

Diva Angelika Mulia¹, Sarah Safitri² and I Gede Putra Kusuma Negara^{1,3}

^{1,2,3}Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480

Received 9 Feb. 2024, Revised 16 Apr. 2024, Accepted 20 Apr. 2024, Published 1 Jul. 2024

Abstract: License Plate Recognition (LPR) systems are now indispensable technology for law enforcement, border control, traffic management, and parking facilities, among other sectors. They enable enhancement in security and public safety, streamline traffic management, boost staff productivity, and deliver a seamless experience for customers. However, the current model faces challenges in producing high-quality images from a fixed-angle camera to produce accurate results in recognizing characters on the license plate. Weather conditions and unfavorable LP angles can lead to low-resolution images, causing inaccuracies in recognizing the character. Therefore, improvement is needed. In the past, to solve these issues, researchers have developed Super-resolution (SR) models capable of generating high-resolution images from low-resolution counterparts. In this paper, the authors enhance the LPR technology to become an automatic solution which is called Automatic License Plate Recognition (ALPR) system by incorporating SR, aiming to automatically improve character recognition. The study comprises two phases: the detection phase and the recognition phase. In the detection phase, the authors utilize state-of-the-art object detection models, including the YOLOv8 model, and the Faster R-CNN model that uses detectron2. These models perform well. YOLOv8 achieves 93% accuracy in both train and validation datasets, and 90% in the test dataset. While Faster R-CNN achieves 71%, and 74%, respectively. In the recognition phase, the authors employ stand-alone Tesseract-OCR and SRGAN-enabled Tesseract-OCR. The end-to-end pipeline achieves a Character Error Rate (CER) of 53.9% (stand-alone Tesseract-OCR) and 51.7% (SRGAN-enabled Tesseract-OCR). At the same time, Levenshtein distance achieves 3.6% (stand-alone Tesseract-OCR) and 3.5% (SRGAN-enabled Tesseract-OCR). This highlights the effectiveness of SRGAN in enhancing image quality and, consequently, improving the performance of OCR engines. The insights gained from this study can contribute to the development of robust license plate recognition systems for real-world deployment.

Keywords: Super resolution, license plate detection, license plate recognition, YOLOv8, SRGAN.

1. INTRODUCTION

In recent years, the application of License Plate Recognition (LPR) under computer vision technology, has experienced a substantial increase. Advancements in this technology have facilitated the translation of pixel values into actionable outcomes using sophisticated algorithms. LPR stands out as a prominent solution that processes images to capture the vehicle's license plate images, identifies license plate characters, extracts relevant information from these images, and converts it into machine-readable formats, such as text strings. This information can then be processed and organized in databases which in the end provide analytical and meaningful information for the decision-making process for the stakeholders.

LPR has emerged as a pivotal technology in various fields, playing a vital role in applications such as intelligent transportation systems, border control systems, smart city systems, and even law enforcement systems. This technol-

ogy also plays a key role in modern tasks such as traffic control, vehicle tracking, and security surveillance.

In today's context, LPR holds immense importance due to its wide-ranging applications. For instance, the capability to extract information quickly and accurately from license plates enables tasks that would be impractical and time-consuming for humans, such as searching for a specific car across an entire street in a matter of seconds rather than minutes [1]. This efficiency makes LPR a crucial tool in enhancing the speed and precision of various operations in the current technologically driven society.

The conventional LPR methods commonly follow a four-step process involving license plate localization, character segmentation, feature extraction, and character recognition. Typically applied to images obtained from fixed cameras [2][3], these methods require relatively high-quality images as input. However, the challenges of capturing these



images in real-world scenarios, influenced by other factors such as the surrounding environment, technical constraints, and weather conditions, often result in distorted images. Relying on a captured image for identifying plate characters has drawbacks, particularly when dealing with low-quality and low-resolution images [4][5]. This conventional LPR technology poses a significant challenge for subsequent advanced computer vision applications, especially in dynamic scenarios with moving vehicles and complex environments. This emphasizes the essential requirement for robust computer vision methods in license plate identification systems. To ensure flexibility and reliable performance in real-world scenarios, overcoming these challenges requires enhancements in LPR techniques, marking a significant step forward in the evolution of License Plate Recognition technology.

Low-resolution image poses a significant challenge, impacting the overall effectiveness of LPR systems and reducing their reliability in practical applications. To address this issue, efforts have been made to explore solutions that enhance the resolution of captured images, aiming to improve the robustness of LPR systems in real-world scenarios. In recent years, the enhancement of image resolution through the application of Super-resolution (SR) techniques has been developed. SR is a computational task aimed at restoring HD (high-definition) images from the original low-resolution counterparts [6] to boost the accuracy of image analysis programs, both for single-frame and multi-frame applications [7]. The intrinsic value of HR images lies in their capacity to retain finer details, leading to superior visual quality, making them essential in various domains. To extend the use of HD images to various sectors, such as license plate detection, digital medical imaging, digital satellite imaging, and digital security imaging [6]. The pursuit of advancements in SR technology is motivated by its significant potential to enhance image quality. This enhancement, in turn, contributes to its effectiveness in crucial areas of image analysis and interpretation.

In this paper, the author's main objective is to enhance the adaptability of the LPR system to a wide range of image-capturing solutions. The authors achieve this objective by integrating cutting-edge object detection to detect the plate license, Tesseract-OCR to recognize the characters on the license plate, and SR to enhance the image resolution. The object detection phase is specifically utilized to detect the license plate bounding box, assisting the LPR in precisely locating the license plate information. On the other hand, the SR model is utilized to enhance low-resolution images into HD images. Simultaneously, Tesseract is utilized to identify the characters of the license plate. The combination of these three solutions allows LPR systems to identify the characters on the plate with even greater performance.

This paper used a mix of quantitative and qualitative research methodology. This paper provides a quantitative analysis of various research that has been done in the past.

On the other hand, this paper also provides a quantitative analysis that highlights the performance of each research.

2. RELATED WORK

In real-life scenarios of LPR systems, computers encounter significant challenges in distinguishing vehicle license plates from surrounding symbols and logos that exist on the roads, in addition to other factors like angle, distance, and color exposure. Therefore, the output depends on these images' original quality to enhance its image-learning and image-capturing capabilities [8]. To address these challenges and improve the recognition process, there is a need to employ and advance sophisticated deep-learning algorithms. This means the authors are not just relying on the camera, but also providing the computer systems with Artificial Intelligence capability to enhance its capability to accurately identify the license plates and recognize the character on the license plate.

In contrast, the Super-resolution (SR) technique plays a crucial role in improving the quality of the captured images. This technique contributes significantly by providing clearer and more detailed information visually. This technique enhances the resolution of the images, ensuring that the computer algorithms have access to sharper and more precise data, ultimately aiding in better detection and recognition of license plate details.

So, the solution in this paper includes the following techniques: License Plate Detection, License Plate Recognition, and Super-resolution.

A. License Plate Detection

The solution in this detection task uses You Only Look Once (YOLO) and Faster Region-Convolutional Neural Networks (Faster R-CNN) models, which are part of the Convolutional Neural Networks (CNN) technique that is rapidly expanding in the era of deep learning. Convolutional Neural Networks (CNN) have become the most effective deep learning technique for detection tasks. Among the prominent CNN-based algorithms, YOLO introduced by Redmon in 2015 [8], has gained significant popularity. YOLO adopts an object detection approach framed as a regression problem, excelling in predicting bounding boxes and class probabilities. Its ability to capture general representations of objects surpasses traditional methods like the Deformable Part Model (DPM) and Recurrent-Convolutional Neural Network (R-CNN) across diverse domains, making it a preferred choice for researchers [8].

In the study conducted by I. R. Khan et al., [9] YOLOv5 was employed to detect license plates in real-world traffic videos, and a customized Convolutional Neural Network (CNN) was utilized for the recognition of alphanumeric characters on the license plates. The outcomes revealed a notable improvement in accuracy compared to traditional object detection models. This innovative approach not only showcases the effectiveness of YOLOv5 in real-world scenarios but also highlights the significance of lever-

aging custom CNNs for precise alphanumeric recognition on license plates, contributing to advancements in object detection technology within traffic surveillance applications. Another notable study by T. Ma, Z. Liu, et al. [10] explored license plate detection using PSA-YOLO, a variant based on YOLOv5, revealing a significant improvement compared to Single-shot Detector (SSD) and YOLOv4 as the deep learning model for object detection and localization. The findings of the paper revealed a remarkable improvement, showing an increase of 4.8% (PSA-YOLO) and 2.3% (SSD and YOLOv4).

In addition to YOLO, another advanced CNN-based algorithm worth mentioning is Faster R-CNN. A recent innovation by N. Omar et al. [11], introduces a novel approach based on the fusion of multiple Faster R-CNN architectures. This method excels in pinpointing the precise location of license plates in images, achieving an impressive accuracy rate of 97%. The integration of Faster R-CNN in license plate detection signifies a notable advancement in object localization accuracy, demonstrating its capability to enhance the precision and reliability of identifying license plates within images. Furthermore, a noteworthy study by Z. Mahmood et al. [12] focuses on the detection of license plates on public vehicles using Faster R-CNN, combined with digital image processing techniques. This approach has yielded impressive levels of accuracy, precision, and recall, surpassing the performance of several recently developed methods. In alignment with these findings, M. Shahidi Zandi and R. Rajabi [13] conducted a study employing Faster R-CNN and YOLOv3 for vehicle license plate detection. Both algorithms demonstrated excellence in the realm of object detection, showcasing competitive values of mean Average Precision (mAP), accuracy score, and recall score if compared to recently developed methods.

The impressive performance of these algorithms indicates their strength and usefulness in the dynamic field of object detection methods. It means they can handle different situations very well and are versatile. This holds significance because, with technological advancements, these algorithms retain their effectiveness and can adapt to emerging challenges. The ability to work well in various scenarios makes them valuable tools in computer vision and object detection.

B. License Plate Recognition

In today's AI-enabled world, LPR systems play a key role. These systems rely on intelligent algorithms that perform well in different situations, ensuring accurate identification of license plates in various scenarios and sectors. Expanding on the comprehensive study led by E. C. Huallpa et al. [14], which explores license plate detection and recognition, the research employs the Tesseract-OCR engine as a tool for character detection on vehicle plates. Notably, Tesseract showcases its effectiveness by accurately recognizing characters even when images are captured from varying distances and angles. This past research not only

underscores the reliability of Tesseract in the character recognition phase but also highlights its adaptability in handling diverse image conditions, contributing valuable insights to the field of license plate detection and recognition. Moreover, the study conducted by W. Swastika et al. [15] explores the enhancement of recognition accuracy for vehicle license plate numbers through Vehicle Image Reconstruction using a Super-resolution Convolutional Neural Network (SRCNN). This research employs two recognition methods, Tesseract and Stereography Projection Network (SPNet). The utilization of SRCNN to construct high-resolution images proves impactful, resulting in a substantial increase in the average accuracy of vehicle license plate number recognition. Specifically, the accuracy improves by an impressive increment of 16.9% when using Tesseract and 13.8% when using SPNet.

C. Super-resolution (SR)

Image Super-resolution (SR) and deblurring have been dynamic research areas in computer vision [16]. In previous approaches, image processing techniques heavily depended on using sharpening filters and interpolation methods like bicubic and bilinear interpolations [17]. Despite being effective benchmarks, these methods frequently resulted in excessively smooth textures in the reconstructed images.

With the advent of Convolutional Neural Networks (CNNs), the SR landscape witnessed a transformative shift. The seminal Super-Resolution Convolutional Neural Network (SRCNN) [18] demonstrated impressive results by applying convolutional layers to enhance low-resolution images. Subsequently, the Very Deep Super-resolution (VDSR) model [19] introduced a deep network with 16 convolutional layers and residual learning, outperforming SRCNN. Despite the success of CNN-based methods in increasing Peak Signal-to-noise Ratio (PSNR) and reducing Mean Square Error (MSE) between SR and HD images, the emergence of Generative Adversarial Networks (GANs) marked a significant paradigm shift.

The concept of GANs was introduced by I. J. Goodfellow et al., [20], revolutionized SR development. The Super-Resolution GAN (SRGAN) [21] by C. Ledig et al., demonstrated the ability to infer photorealistic images at 4x up-scaling levels. Further innovations include the Least Squares GAN (LSGAN) [22], which generates higher-quality images with increased stability during the learning process. X. Mao et al. introduced the Enhanced Deep SR (EDSR) network [23], optimizing the SR image generation process compared to original GANs. T. C. Wang et al. [24] proposed a novel approach using conditional GANs for synthesizing high-resolution images from semantic label maps, achieving visually appealing results with unique adversarial loss and multi-scale architectures.

Next, the authors will make a comparison of prior studies based on their characteristics explained before. We will break down the strengths and weaknesses of each study below.

The study by I. R. Khan et al., [9], has weaknesses due to using the old version of YOLOv5. Its strength is enhance the quality of images in each phase of LPD, LPR, capable of detecting trac videos. Study by T. Ma, Z. Liu, et al. [10], has limitations due to the old version of YOLOv5 (PSA-YOLO). Its strength is capable of providing signi cant improvement compared to Single-shot Detector (SSD) and YOLOv4. The study by N. Omar et al. [11], have limitations due to using multiple conventional Faster R-CNN architectures. While its strength can improve then accuracy rate of 97%.

From this perspective, the authors believed that the model used in the paper used modern technology, Faster R-CNN using Detectron2 which is the newest technology introduced in 2018. On the other side, even though its strength achieved high accuracy, the authors' technique reached 74% but using current technology, making it versatile tool for a range of computer vision tasks for today's needs.

In the study by Z. Mahmood et al. [12], the limitation is due to using stand-alone Faster R-CNN. While its strength can be combined with digital image processing techniques. From authors' perspective, the solution offered in this paper has outperformed the prior study. The authors' solution uses the Faster R-CNN technique with Detectron2 and enables SRGAN technology which is the state-of-the-art technology.

The study by M. Shahidi Zandi and R. Rajabi [13] has limitations due to using Faster R-CNN and the old version of YOLOv3. Its strength can improve mean Average Precision (mAP), accuracy score, and recall score if compared to recently developed methods. From this perspective, the solution offered by the authors in this paper can provide better results (74% and still can be improved).

The study led by E. C. Huallpa et al. [14], has limitations using stand-alone Tesseract-OCR. Its strength is accurately recognizing characters on the plate even when images are captured from varying distances and angles. From this perspective, the solution offered by the authors through this paper can provide better results using SRGAN-enabled Tesseract-OCR which achieved high-quality images based on its original (low) quality image. This solution can produce more accurate results in recognizing characters on the license plate.

Subsequently, the study conducted by W. Swastika et al. [15], has limitations because of using a Super-resolution Convolutional Neural Network (SRCNN) which is a conventional technique. However, it uses two methods, Tesseract and Stereography Projection Network (SPNet). From this perspective, the solutions offered by the authors using both SRGAN-enabled Tesseract-OCR and stand-alone Tesseract, with more complex combinations of metrics such as CER and Lavenshtein, which in the end able to recognize 100% characters on the license plate (CER 0%, Lavenshtein 0%).

3. PROPOSED METHOD

In this section, the authors will explore the methods and techniques used in this paper, providing detailed explanations of the object detection models, character recognition with OCR engine, and the Super-resolution model the authors utilized. Nevertheless, the authors will initiate the discussion by outlining the data collection (acquisition, splitting, annotation), the selection of object detection model (YOLOv8 and Faster R-CNN), data cleaning (cropping the object bounding box), pre-processing according to the requirement, data processing (Tesseract-OCR and Tesseract-OCR with SR). In the following Figure 1 is an overview of the steps and approaches taken by the authors.

Figure 1. Overview steps and approach taken by the authors

A. Data Acquisition and Preparation

Dataset: To facilitate the paper objectives, the authors utilized the Stanford Cars dataset sourced from Kaggle, consisting of 16,200 images for training and testing with a ratio of 50:50. This dataset consists of a diverse set of car images captured from different angles, under varying color exposure (brightness, contrast, etc.) and sourced from various countries. The dataset includes car images with and without license plates. Annotations for this dataset are provided in MAT format, including bounding box information for

both the car and license plate. The authors implemented several modifications to adjust the dataset to meet the authors' requirements and approach:

Data Partitioning: the authors divided the original dataset into training, validation, and testing with a ratio of 50:25:25. This partitioning strategy is essential for robust model training, validation, and evaluation.

Annotation Format Conversion: In alignment with the chosen methodology, the authors adapted the dataset by transforming the original annotations into formats that are compatible with YOLO format. This conversion is crucial to seamlessly integrate the dataset with the selected detection models, namely YOLOv8 and Faster RCNN. The use of YOLO formats ensures that the models receive input data in the required specification.

Custom Character Annotations: To increase the dataset's enrichment and relevance to this paper's specific objectives, the authors generate custom character annotations. These custom annotations are meant to identify the existence of characters on each license plate within the training, validation, and testing datasets. This additional information contributes to the model's capability to learn, detect, and analyze specific features.

Tools: The authors used the following tools in conducting the paper.

Google Google Collab Pro A100 GPU version. This tool is used for processing the Faster R-CNN technique because involving a large dataset and complex model.

Visual Studio Code (local and cloud). This tool is used for processing the YOLOv8 technique..

Pytorch. Pytorch library is used as a base library for both Faster R-CNN and YOLOv8 techniques.

GitHub. GitHub is used as a collaboration tool among authors.

Microsoft Office 365 Microsoft Office 365 is used as a analyzing tool.

B. Data Pre-processing

As shown in Figure 2, the image will be processed further and is obtained from a cropped license plate bounding box detected by the object detection process. This pre-processing step is executed to enhance the character recognition using Tesseract-OCR. The goal is to optimize Tesseract's ability to interpret license plate characters, a task that would be less effective if the authors were to input the original image without any preliminary pre-processing.

The cropped bounding box image, provided to both Tesseract-OCR and SRGAN is enlarged four times from its original size. This ensures that the input and output images

have identical sizes. Following this, the authors simplify the cropped image by converting it to grayscale, making it more readable for the computer. Subsequently, the authors apply blurring operations, such as Gaussian and Median Blur, to eliminate noise and create a smoother image. These blurred images are then processed using Otsu's thresholding, which is a technique that categorizes pixels into foreground and background classes based on their grayscale intensity values. Finally, morphological operations, like dilation, are applied to expand object boundaries, fill small gaps, and merge overlapping objects. Additionally, contour detection is also employed to help the program easily identify the boundaries of objects in the image. This series of pre-processing steps aims to enhance the image quality and facilitate the subsequent object recognition process.

In a further section, the authors will investigate a comprehensive overview of our key methodology, which is divided into three key phases: (1) License Plate Detection, (2) Character recognition using Tesseract-OCR, and (3) Improvement of character recognition using SRGAN. Each phase plays a crucial role in enhancing the overall effectiveness of the approach, ensuring accurate and robust license plate detection and character recognition.

C. License Plate Detection

In this section, the authors will conduct a comparative analysis between YOLOv8 and Faster R-CNN for license plate detection tasks. The objective is to evaluate and contrast the performance, accuracy, and efficiency of these two popular object detection models in the specific context of recognizing license plates.

1) YOLOv8

The architecture of YOLOv8 draws inspiration from its predecessor, YOLOv5, with significant refinements and novel additions [25].

CSPDarknet53 Feature Extractor: CSPDarknet53 provides an efficient backbone tailored for YOLOv8's needs. By leveraging the strengths of Darknet while making strategic changes like using a smaller initial conv kernel, the CSPDarknet53 feature extractor balances improved feature learning with reduced computational requirements. The CSPDarknet53 extracts features using conv layers, batch norm, and SiLU activations. It modifies Darknet by replacing the initial 6x6 conv with a more efficient 3x3 conv,

Figure 2. Pre-processing phase

enhancing feature learning while reducing computations. Overall, CSPDarknet53 provides an optimized feature extraction backbone tailored for YOLOv8.

and loss computations, YOLOv8 can more recently re-tune the model during the training phase for cutting-edge object detection.

C2f Module (Cross-Stage Partial Bottleneck): C2f module enables effective cross-stage communication of feature information. By combining outputs from the bottleneck blocks, it allows contextual and semantic feature fusion which improves the quality of the final feature representations used for object detection. The C2f module fuses different feature levels by aggregating outputs from bottleneck blocks. Each block contains two 3x3 conv layers with residual connections. By combining high-level and low-level features, C2f enables cross-stage communication of global and local context information. This contextual fusion enhances feature representations for improved object detection. Overall, C2f integrates multi-scale feature maps through a simple yet effective architectural design.

2) Faster R CNN

The next algorithm chosen for comparison is Faster R-CNN. As explained in the previous section, Faster R-CNN has demonstrated excellence in the field of object detection [13]. The strength of the Faster R-CNN technique can be seen in its architecture and components. As illustrated in Figure 4, these components consist of (1) the convolution layers, (2) the Region Proposal Network (RPN), and (3) ROI Pooling. The input image is initially processed through convolution layers serving as the backbone network, with a Feature Pyramid Network (FPN) being used in this instance. FPN extracts features and generates a feature map, which is then fed into the Region Proposal Network (RPN). This distinctive feature sets Faster R-CNN apart from other models like Fast R-CNN and R-CNN.

The RPN network determines anchors belonging to the background and foreground classes from the input image. Integrating RPN into Faster R-CNN significantly enhances the speed of the detection process compared to using the traditional sliding window approach [26]. The object proposals generated by RPN are then mapped onto the feature map. Subsequently, this feature map is fed into the ROI pooling layer to extract feature vectors corresponding to each object proposal.

Detection Head: As illustrated in Figure 3, YOLOv8 adopts an anchor-free detection strategy, eliminating the reliance on predefined anchor boxes and directly predicting object centers. The detection head is structured with independent branches for objectness, classification, and regression tasks. Each branch processes its designated task separately, allowing for a focused approach that contributes to overall detection accuracy.

The RPN network determines anchors belonging to the background and foreground classes from the input image. Integrating RPN into Faster R-CNN significantly enhances the speed of the detection process compared to using the traditional sliding window approach [26]. The object proposals generated by RPN are then mapped onto the feature map. Subsequently, this feature map is fed into the ROI pooling layer to extract feature vectors corresponding to each object proposal.

Figure 3. YOLO Model

Figure 4. Faster R-CNN Architecture

In the output layer, YOLOv8 predicts the likelihood that an object will be in a bounding box using the sigmoid activation for the objectness score. It uses the SoftMax function for class probabilities to show how likely it is that an object will belong to each class. YOLOv8 incorporates binary cross entropy for classification and CloU and DFL for bounding box regression tasks to improve performance. Better convergence and handling of challenging objects are possible when box regression is performed using CloU and DFL losses. The binary cross entropy loss aids in improving the precision of classification. Through this customization of the activation functions

Instead of implementing Faster R-CNN from scratch, the authors have implemented it using Detectron2. Detectron2 is an object detection model developed by Facebook AI Research and [27] was newly introduced in 2018. Detectron2 is written in PyTorch, and its main areas of interest include the detection of key points, object detection, and semantic segmentation [28], making it a versatile tool for a range of computer vision tasks. Detectron2 offers a variety of base models for each of its areas of interest [29]. In the case of Faster R-CNN, there are few base models available [30], two popular base models are R101-FPN and X101-FPN. For this study, the selected base model is X101-FPN because it

has demonstrated better box Average Precision (AP) on the ImageNet benchmark.

D. Character Recognition using Tesseract

In general, the Tesseract-OCR engine can recognize over 100 languages and offers support for various output formats such as plain text, HTML, PDF, and more. To align with the first approach, the authors use Tesseract-OCR to identify alphanumeric characters on the detected license plate. As illustrated in Figure 1, the input for this step is the cropped bounding box the authors detected earlier from the object detection model, which has already undergone pre-processing to enhance Tesseract's accuracy, as also explained in section 3B. The result obtained from using a stand-alone Tesseract-OCR will be compared to the outcome when the authors combine Tesseract with the Super-resolution model. You can explore this in more detail in the next section.

E. Improvement of Character Recognition using SRGAN

Improving the quality of license plate images is crucial for accurately recognizing the characters on the plate. For the second approach, the authors use a super-resolution model called Super-resolution Generative Adversarial Network (SRGAN) before applying the Tesseract-OCR engine. This step is taken to significantly improve the resolution and clarity of the detected license plate. The SRGAN works by generating high-resolution images from low-resolution inputs, thereby improving the overall quality of the license plate image. This enhancement is designed to make it easier for the subsequent Tesseract-OCR engine to accurately recognize alphanumeric characters on the plate.

After the super-resolution enhancement step, the Tesseract-OCR engine is then applied. This combination using SRGAN and Tesseract will boost recognition accuracy further. The result of this process is the combination of characters and numbers typically found on a license plate [31].

F. Evaluation Metrics

To evaluate the object detection models, the authors use Average Precision (AP) as the authors' metric. AP integrates precision and recall. Precision is the ratio of True Positive (TP) divided by the amount of data predicted as TP and False Positive (FP). Precision describes the level of accuracy between the desired data and the prediction results generated by the model. Recall is the ratio of TP divided by the number of data predicted as TP and False Negative (FN). Recall describes the success rate of the model in re-identifying relevant information. The trade-off between recall and precision at various object detector confidence thresholds is represented by the precision-recall curve. Recall may suffer from high confidence levels, which reduce false positives, but recall is increased when more positives are accepted, usually at the expense of precision. As recall rises, a good detector keeps its high precision, demonstrating its capacity to find all ground-truth objects and

identify pertinent objects at the same time. High precision and high recall are reflected in the area under the precision-recall curve (AUC); however, practical curves frequently have zigzag patterns, making AUC estimation challenging. To rectify this, the curve is smoothed to eliminate zigzag behavior before the AUC computation, guaranteeing a more precise assessment of detector performance. The 11-point interpolation and the all-point interpolation are the two methods for doing this [31]. Equation (1) is the calculation for 11-point interpolation.

Where:

$p(k)$ is the precision at position.

k , which is the number of relevant items (correctly identified) divided by the total number of items up to position.

$\max(p(k), p(k-1))$ selects the maximum precision between the current position.

k and the previous position

$k-1$. This accounts for any potential drops in precision in the list.

$r(k)-r(k-1)$ calculates the difference in ranks between the current and previous positions.

Equation (2) is the calculation for all points.

Where:

$p(r)$ represents the precision at a specific recall level r . Precision at a recall level r is the ratio of relevant items (correctly identified) to the total number of items retrieved at or before recall level r .

Several IoUs (Intersection over Union) are used to evaluate the AP. IoU measures the overlapping area between the predicted bounding box and the ground truth bounding box divided by the area of union between them. It can be computed for ten IOUs with steps of 5%, ranging from 50% to 95%; the result is typically reported as AP@50:5:95. It can also be assessed using IOU single values, the most popular of which are 50% and 75%, denoted as AP0.5 and AP0.75, respectively. In this paper, AP0.5 and AP 0.5-0.95 are used.

In the license plate recognition phase, the authors conducted a comparison between recognition when applying both stand-alone Tesseract-OCR and SRGAN-enabled

Tesseract-OCR engines. To compare these two methods, the authors utilized two evaluation metrics: the Levenshtein distance algorithm and the Character Error Rate (CER). The Levenshtein distance calculates the minimum number of insertions, deletions, or substitutions required to transform one sequence of characters into another [31]. The smaller the Levenshtein distance, the closer the recognized outputs are to the ground truth label. A Levenshtein distance of 0 indicates that the output sequence is identical to the corresponding ground truth. On the other hand, CER is also based on the Levenshtein distance concept. CER operates by counting the minimum number of characters needed to transform the ground truth text into the OCR output. Equation (3) and (4) is the formula to calculate CER.

Where in the formula for CER:

S represents the number of substitutions,

D represents the number of deletions,

I represents the number of insertions, and

N stands for the ground truth.

4. RESULT AND DISCUSSION

For license plate detection, the authors utilized two models: YOLOv8 and Faster R-CNN to predict and detect the bounding box of the license plate of the vehicles. Once the bounding box is detected, the next step is to proceed to the license plate recognition phase where two approaches are used. In the first approach, the authors use stand-alone Tesseract-OCR exclusively to recognize the license plate characters. In the second approach, the authors enhance the result of the first approach by incorporating a Super-resolution Generative Adversarial Network (SRGAN) to enhance the image clarity before applying character recognition using Tesseract-OCR.

A. License Plate Detection Result

In the license plate detection phase, the authors employed Average Precision (AP) at IoU thresholds of 0.5 and AP across the range of 0.5 to 0.95 as authors' evaluation metrics. As shown in Table I, the authors found that YOLOv8 exhibited superior performance across the training, validation, and test datasets. This model outperformed Faster R-CNN, which had a training and validation accuracy of 71%, and slightly improved to 74% when tested on the test dataset.

Due to YOLOv8's superior accuracy compared to Faster R-CNN, the authors have chosen to exclusively use the bounding boxes detected by YOLOv8 for the subsequent

TABLE I. YOLOV8 AND FASTER R-CNN ACCURACY COMPARISON

	YOLOv8		Faster R-CNN	
	AP0.5	AP0.5-0.95	AP0.5	AP0.5-0.95
Train	93%	71%	71%	50%
Validation	93%	70,8%	71%	50%
Test	90,6%	68,6%	74%	51%

For the loss function of YOLOv8, as shown in Figure 5, the box loss and classification loss during training decrease as the number of epochs (train) increases. This suggests that the YOLOv8 model is learning and improving over time. Similarly, the results on the validation set also show a positive trend, with losses decreasing as the number of epochs (train) increases.

Figure 5. YOLOv8 training and validation loss function

The decreasing trends in box loss and classification loss indicate that the model is learning and improving over successive epochs (train).

In terms of image detection, YOLOv8 proved to be a powerful technique to detect license plate bounding boxes. On the other hand, Faster R-CNN using Detectron2 can be useful as an object detection tool to refrain from developing it from scratch which is time-consuming. Also, Detectron2 offers a few base models. Each base model has its own training and inference times, which could lead to different perspectives on the output.

B. License Plate Recognition Result

In Table II, the authors present the license plate images alongside the OCR results for both the SRGAN-enabled Tesseract-OCR and the stand-alone Tesseract-OCR engines.

TABLE II. CHARACTER RECOGNITION RESULT

LP without SRGAN	LP with SRGAN	%Similarity			
		Stand-alone Tesseract-OCR		SRGAN-enabled Tesseract-OCR	
		CER (%)	Lev (%)	CER (%)	Lev (%)
		33.3	2	0	0
		14.2	1	0	0
		57.1	4	28.5	2

The CER and Levenshtein (Lev) distance decreased when SRGAN was applied, reaching a value of 0, indicating a 100% match between the predicted characters and the actual numbers. In comparison, stand-alone Tesseract exhibited poor accuracy. Furthermore, the authors conducted mean similarity calculations for all datasets, including the training, validation, and test sets. In Table III, the authors observe a decrease in mean similarity when plates are processed using SRGAN.

TABLE III. MEAN SIMILARITY VALUES

Similarity Metrics	LP without SRGAN	LP with SRGAN
CER (%)	53.9%	51.7%
Levenshtein	3.6%	3.5%

A lower value signifies better performance. This implies that recognition using SRGAN enhances accuracy, as indicated by the decreasing numerical values. In terms of character recognition, the authors use SRGAN technology to transform low-quality images to HD which is then combined with Tesseract-OCR technology to recognize any character on any condition accurately.

5. CONCLUSION AND FUTURE WORK

This study explores license plate detection and recognition using cutting-edge models. The object detection models experienced training, evaluation, and testing on a customized Stanford Cars dataset. YOLOv8 showed better performance in the detection phase compared to Faster R-CNN. Subsequently, the authors utilized the bounding boxes generated by YOLOv8, cropped the images, and applied them to the recognition phase. In the recognition process, employing SRGAN-enabled Tesseract-OCR significantly enhanced accuracy compared to using stand-alone Tesseract. Although some images may not be accurately recognized, particularly when they are too tilted or contain excessive noise, the use of SR models remains beneficial. This comprehensive investigation highlights the importance of pre-processing images and leveraging SR models, such as SRGAN, for optical character recognition in license plate systems. The authors' contributions to this paper provide valuable insights into the ongoing development of license plate detection and recognition systems.

Besides the advancement of the solution provided in this paper, the authors also notice the drawbacks of the approach, including:

Not all datasets have annotation available publicly.

Not all datasets are already merged in public. This means the researcher must merge the required databases separately using a specific solution.

Due to the limited timeframe, this study is completed within three months. The authors believed, that to



gain more reliable results, the study ideally be conducted for at least six months.

In future research, the authors plan to investigate other detectron2 base models for Faster R-CNN, such as R101-FPN or R101-DC5. Each base model has its own training and inference times, which could lead to different perspectives on the output.

REFERENCES

- [1] A. L. C. Bazzan and F. Klügl, *Introduction to Intelligent Systems in Traffic and Transportation*. Springer Nature, 2022.
- [2] L. Anil, "Automatic number plate detection in fog-haze environments," *Social Science Research Network*, 2023.
- [3] N. C. J. F. Gakkai, "International fuzzy systems association," *IEEE Systems*, 2017.
- [4] S. M. Silva and C. R. Jung, "A flexible approach for automatic license plate recognition in unconstrained scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, p. 5693–5703, 2022.
- [5] M. H. et al., "Intelligent image super-resolution for vehicle license plate in surveillance applications," vol. 11, no. 4, p. 892, 2023.
- [6] J. Cai, Z. Meng, J. Ding, and C. M. Ho, "Real-time super-resolution for real-world images on mobile devices," pp. 127–132, 2022.
- [7] C. Zhou and Z. Jiang, "Artificial intelligence-based super-resolution reconstruction algorithm for pulsed multi-frame images," pp. 01–07, 2020.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.
- [9] I. R. K. et al., "Automatic license plate recognition in real-world traffic videos captured in unconstrained environment by a mobile camera," *Electronics (Switzerland)*, vol. 11, no. 9, 2022.
- [10] M. Tian, Z. Liu, Y. Si, and C. Fu, "Psa-yolo: License plate detection method based on pyramid segmentation attention in complex scenes," *International Conference on Image, Vision and Computing*.
- [11] N. Omar, A. M. Abdulazeez, A. Sengur, and S. G. S. Al-Ali, "Fused faster rcnns for efficient detection of the license plates," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, 2020.
- [12] Z. Mahmood, K. Khan, U. Khan, S. H. Adil, S. S. A. Ali, and M. Shahzad, "Towards automatic license plate detection," *sensors*, *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, 2022.
- [13] M. S. Zandi and R. Rajabi, "Deep learning based framework for iranian license plate detection and recognition," *Multimed Tools Appl*, vol. 81, no. 11, 2022.
- [14] E. C. Huallpa, A. A. S. Macalupu, J. E. O. Luque, and J. Sánchez-Garcés, "Automatic recognition and license plate detection model based on opencv and machine learning," *Applications in Software Engineering*, 2022.
- [15] W. Swastika, E. R. F. Sakti, and M. Subianto, "Vehicle images reconstruction using srcnn for improving the recognition accuracy of vehicle license plate number," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 4, 2020.
- [16] J. Scanvic, M. Davies, P. Abry, and J. Tachella, "Self-supervised learning for image super-resolution and deblurring," 2024.
- [17] M. Jahnavi, D. R. Rao, and A. Sujatha, "A comparative study of super-resolution interpolation techniques: Insights for selecting the most appropriate method," p. 504–517, 2024.
- [18] Y. Li, B. Sixou, and F. Peyrin, "A review of the deep learning methods for medical images super resolution problems," *IRBM*, vol. 42, no. 2, 2021.
- [19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations, ICLR*, 2015.
- [20] I. J. G. et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, p. 2672–2680, 2014.
- [21] C. L. et al., "Photo-realistic single image super-resolution using a generative adversarial network," *Computer Vision and Pattern Recognition*, 2017.
- [22] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *IEEE International Conference on Computer Vision*, 2017.
- [23] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," *Computer Vision and Pattern Recognition Workshops*, 2017.
- [24] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *Computer Vision and Pattern Recognition*, 2018.
- [25] J. Terven and D. M. Cordova-Esparza, "A comprehensive review of yolo: From yolov1 and beyond a preprint," 2023.
- [26] Detectron2. [Online]. Available: <https://github.com/facebookresearch/detectron2/blob/main/docs/tutorials/models.md>
- [27] B. Liu, W. Zhao, and Q. Sun, "Study of object detection based on faster r-cnn," *Chinese Automation Congress*, 2017.
- [28] M. Butt, N. Glas, J. Monsuur, R. Stoop, and A. de Keijzer, "Application of yolov8 and detectron2 for bullet hole detection and score calculation from shooting cards," vol. 5, no. 1, p. 72–90, 2023.
- [29] J. F. Restrepo-Arias, P. Arregocés-Guerra, and J. W. Branch-Bedoya, "Crops classification in small areas using unmanned aerial vehicles (uav) and deep learning pre-trained models from detectron2," p. 273–291, 2022.
- [30] detectron2 model zoo. [Online]. Available: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md
- [31] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," *International Conference on Systems, Signals, and Image Processing*, 2020.

