Integrated License Plate Recognition Using YOLO and CNN for Automated Vehicle Identification

Sahaana Shri S.K*, R. Nethra*, CH. Vaishnavi Krishna*, Mounika C H*,
*Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, India
Emails: {cb.sc.u4aie24149, cb.sc.u4aie24147, cb.sc.u4aie24108, cb.sc.u4aie24169}@cb.students.amrita.edu

***Abstract*—Automated License Plate Reader (LPR) technology finds significant use in traffic monitoring, vehicle identification, and security systems. In this paper, we propose a practical LPR system with the feature extraction capability of Convolutional Neural Networks (CNNs) and the You Only Look Once (YOLO) object detection system for real-time and high-speed recognition. Our approach not only enhances the detection rate but also effectively reduces processing time, which makes it deployable in real-time applications. Our model, compared in different settings, achieves a whopping 97.69% accuracy, reflecting its robustness and versatility. We present the system architecture, method, and experimental outcomes with emphasis on the effectiveness of our approach under different challenging conditions.**

***Index Terms*—License Plate Recognition, YOLO, CNN, Object Detection, Deep Learning**

I. Introduction

License Plate Recognition (LPR) has become an essential technology in various applications, including traffic monitoring, automated toll collection, and security enforcement. It enables the automatic identification of vehicles, reducing the need for manual intervention and improving efficiency. However, traditional methods often struggle with challenges such as poor lighting conditions, low-quality images, and variations in license plate designs, making accurate detection difficult. Recent advancements in artificial intelligence (AI) and deep learning have significantly improved LPR accuracy, particularly with the use of Convolutional Neural Networks (CNNs) and the You Only Look Once (YOLO) object detection framework. These technologies allow for real-time detection of license plates with high precision, even in complex environments.

The proposed approach stands out by integrating YOLO and CNNs to enhance license plate detection and recognition. YOLO is well known for its ability to perform rapid object detection, while CNNs specialize in extracting intricate features from images. By combining these two methods, the system ensures both speed and accuracy, even when dealing with low-quality images or cluttered backgrounds. Unlike conventional approaches that rely on predefined rules or handcrafted features, this deep learning-based method is adaptable and robust, making it more effective across various scenarios.

One of the key strengths of this approach is its ability to perform well under diverse conditions, including different lighting environments, weather changes, and variations in license plate fonts and sizes. The real-time nature of YOLO makes it particularly useful in applications that require immediate responses, such as traffic surveillance, toll collection, and security systems. Additionally, the proposed method reduces errors in detection, making it highly reliable for use in law enforcement and public safety applications.

This system has broad practical applications. In traffic management, it can be used to track vehicles, detect rule violations, and assist in congestion control. Automated toll collection systems can benefit from its ability to quickly and accurately identify vehicles, streamlining the payment process. Parking facilities can implement this technology to monitor entry and exit points efficiently. Law enforcement agencies can use it to identify stolen or suspicious vehicles with greater accuracy. By integrating deep learning into LPR, this approach enhances the overall efficiency and reliability of vehicle identification systems.

The goal of this research is to develop a fast and accurate LPR system that can be effectively deployed in real-world scenarios. By leveraging deep learning, the system improves detection performance, making it a valuable tool for various industries. The following sections will provide a detailed discussion of the system's architecture, methodology, experimental results, and evaluation across different conditions.

II. Related Work

License Plate Recognition (LPR) has evolved significantly, transitioning from traditional image processing techniques to modern deep learning approaches. The two primary tasks in LPR—license plate (LP) detection and character recognition—face challenges such as varying plate formats, environmental conditions, and multilingual character sets. Over the years, different methodologies have been developed to improve the accuracy and efficiency of LPR systems.

Early LPR systems primarily relied on traditional image processing techniques, including edge detection [6] and coefficient correlation [17], for identifying license plates. While these methods performed well in controlled environments, they struggled with noise, lighting variations, and diverse plate styles. Optical Character Recognition (OCR) techniques, initially based on template matching [1], were highly sensitive to font variations and distortions. Later improvements integrated machine learning models such as k-nearest neighbors (k-NN) [2] and support vector machines (SVM) [4], coupled with feature extraction techniques like Scale-Invariant Feature Transform (SIFT) [5]. Although these approaches improved recognition accuracy, they required extensive feature engineering and often failed to generalize well across different plate formats.

With the rise of machine learning, approaches based on classifiers like SVM [7], [14] combined with Histogram of Oriented Gradients (HOG) [5] showed notable improvements in character recognition. Additionally, Artificial Neural Networks (ANNs) [11] and sparse autoencoders [20] enhanced feature extraction. However, these methods remained computationally expensive and struggled with real-time processing, making them less suitable for high-speed applications such as traffic surveillance.

The emergence of deep learning revolutionized LPR by eliminating the need for handcrafted features and significantly improving detection accuracy. Convolutional Neural Networks (CNNs) enabled automatic feature extraction, leading to major advancements in LPR performance. Masood et al. [10] developed a CNN-based system that performed well under various lighting conditions and occlusions. Wang et al. [18] optimized CNN structures for high-speed traffic scenarios, achieving faster and more accurate recognition. Montazzolli and Jung [12] introduced a CNN-based model tailored for Brazilian plates, incorporating spatial transformer networks for improved localization. Other studies [16] combined CNNs with Long Short-Term Memory (LSTM) networks for character recognition, allowing systems to distinguish between different types of license plates. Parallel CNN architectures [19] have also been applied to identify plate characteristics, achieving high precision rates. Additionally, edge detection techniques combined with CNNs [3] have further enhanced the performance of Automatic License Plate Recognition (ALPR) systems. The introduction of specialized OCR systems such as BLPNet [13] has further improved recognition accuracy. Despite these advancements, real-time processing and multilingual plate recognition remain challenging, leading to the exploration of Transformer-based architectures [9] for improved generalization.

Another breakthrough in LPR was the application of the You Only Look Once (YOLO) object detection framework. Since its introduction in 2015 [15], YOLO has been widely adopted for real-time license plate detection due to its speed and efficiency. Laroca et al. [8] demonstrated the effectiveness of YOLOv3 in LP localization, combining it with CNNs for high-precision character recognition. Single-stage YOLO models have significantly improved processing speeds, making them ideal for real-time applications. However, challenges such as varying plate formats and environmental conditions still persist. Recent research is exploring Transformer-based models [9] to enhance robustness and adaptability across diverse datasets.

Despite advancements in LPR, existing methods have limitations. Traditional techniques struggle with complex backgrounds and lighting variations, while deep learning models, though accurate, demand high computational resources. YOLO ensures real-time detection but may falter in low-light or occluded scenarios. To address these challenges, our approach combines YOLO's speed with CNN's precision, enhancing efficiency and accuracy for real-world applications like traffic monitoring and law enforcement.

III. Proposed System Description

Our system is designed to accurately and efficiently recognize license plates using deep learning. The process starts with data preprocessing, where vehicle images are collected, resized, and enhanced to improve robustness. To detect license plates, we use YOLO, which quickly identifies and localizes plates by predicting bounding boxes. Once detected, a CNN extracts important features to refine the detection and reduce errors. After that, EasyOCR is applied to read the text from the plates, converting visual information into readable characters. The system is trained and fine-tuned using TensorFlow, ensuring high accuracy. To evaluate performance, we use metrics like precision, recall, and IoU. The following subsections provide a more detailed explanation of each step.

A. Dataset Collection and Preprocessing

The dataset used for training our model was collected from Kaggle, specifically the Car Plate Detection dataset. This dataset consists of numerous car images, each accompanied by an XML annotation file containing the exact coordinates of the license plate. These annotations are essential for training the detection model, as they serve as ground truth labels, helping the model learn the precise location of license plates.

Before training, we preprocess the data to ensure consistency and improve model performance. The preprocessing begins with extracting the XML annotations to obtain bounding box coordinates for each plate. Next, all images are resized to a standard 416×416 resolution, ensuring uniformity and allowing the model to process them efficiently. Since color information is often unnecessary for license plate detection, we convert the images to grayscale, reducing computational complexity while preserving essential features like edges and shapes.

To enhance the model's ability to recognize plates under varying conditions, we apply data augmentation techniques such as rotation, flipping, and brightness adjustments. These transformations create additional training examples, improving generalization and preventing overfitting. This structured preprocessing pipeline ensures that the model is well-prepared to handle diverse real-world scenarios. The figure 1 illustrates key preprocessing steps for license plate recognition. From left to right: the first image shows binary conversion for enhanced contrast, the second presents the cropped license plate, the third depicts noise removal for better clarity, and the fourth is the grayscale version, preserving essential details for OCR. These steps improve recognition accuracy under various conditions.

B. YOLO-Based License Plate Detection

YOLO (You Only Look Once) is an object detection model that processes an entire image in a single forward pass, making it highly efficient for real-time applications. The model divides

Fig. 1: Preprocess Output

the image into an $S \times S$ grid, where each grid cell predicts multiple bounding boxes. Each prediction includes the center coordinates (x, y), width and height of the bounding box, a confidence score indicating the presence of an object, and class probabilities (in our case, a single class: ”license plate”).

To assess the performance of YOLO in detecting license plates, various loss functions are utilized, including localization loss, confidence loss, and classification loss.

The localization loss measures the error in the predicted position and size of the bounding boxes:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \tag{1}$$

where λ_{coord} is a weight parameter, $\mathbf{1}_{ij}^{\text{obj}}$ is an indicator function (1 if an object is present in cell i and box j), and (x_i, y_i) are the predicted coordinates compared to the ground truth $(\hat{x}_i, \hat{y}_i)$.

The confidence loss evaluates how accurately YOLO predicts the presence of a license plate:

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \tag{2}$$

where C_i is the predicted confidence score, and $\hat{C}_i$ is the actual confidence value.

The classification loss ensures that the detected object is correctly classified as a license plate:

$$\sum_{i=0}^{S^2} \mathbf{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \tag{3}$$

where $p_i(c)$ is the predicted probability for class c, and $\hat{p}_i(c)$ is the ground truth probability.

By optimizing these loss functions, YOLO improves its detection accuracy, making it a robust choice for real-time license plate recognition.

C. CNN-Based Feature Extraction

After YOLO detects a potential license plate region, a Convolutional Neural Network (CNN) further processes the detected area to extract meaningful features and refine the detection. The CNN plays a crucial role in confirming whether the identified region indeed contains a license plate and improving localization accuracy.

The CNN consists of multiple layers, each performing a specific function. Convolutional layers apply filters to extract patterns from the image, producing feature maps. This operation is mathematically represented as:

$$X_{\text{new}} = X * W + b \tag{4}$$

where X is the input image, W is the filter (kernel), b is the bias term, and $*$ represents convolution. To introduce non-linearity and enable the network to learn complex patterns, an activation function such as ReLU (Rectified Linear Unit) is applied:

$$f(x) = \max(0, x) \tag{5}$$

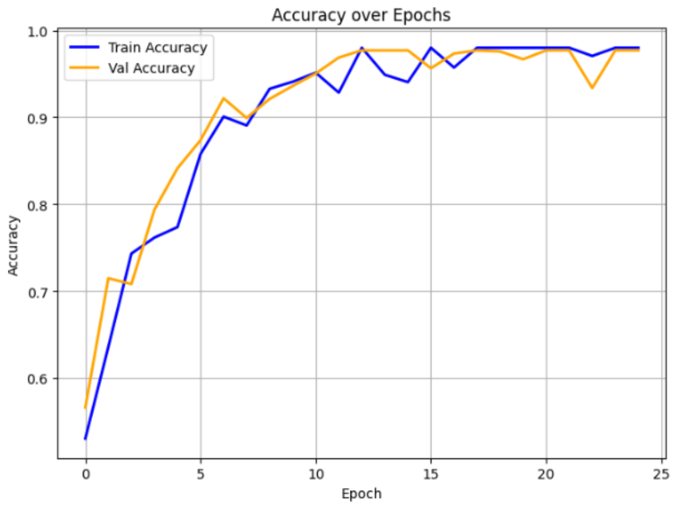

Fig. 2: Model Accuracy Graph

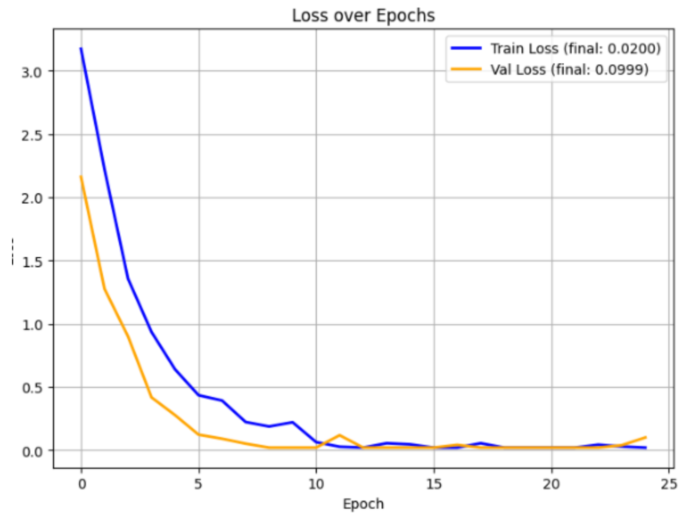

Fig. 3: Model Loss Graph

To reduce the size of feature maps while retaining essential information, pooling layers are used. Max pooling is commonly applied, selecting the highest value within a small window to preserve key features. After these layers, the extracted features are flattened and passed through fully connected layers, which ultimately determine whether the detected region truly contains a license plate and refine its bounding box. By processing images in this structured manner, CNNs enhance detection accuracy and reduce false positives, ensuring reliable recognition even in challenging conditions.

D. Optical Character Recognition to Extract Text

After detecting the license plate using YOLO and refining the region through CNN-based feature extraction, the next step is to extract the alphanumeric characters. For this purpose, we employ EasyOCR, a widely used optical character recognition (OCR) tool designed for efficient text extraction from images.

The OCR process begins with image enhancement, where the detected license plate region undergoes preprocessing techniques such as brightness and contrast adjustments to improve text readability. Once the image is optimized, segmentation is performed to isolate individual characters, ensuring that the OCR model can analyze them separately. Each segmented character is then processed through a pre-trained deep learning model that identifies and classifies it accurately. Finally, the recognized characters are combined to reconstruct the complete license plate number.

This step is crucial, as the objective is not only to detect the license plate but also to accurately extract and interpret the text displayed on it, enabling further applications such as vehicle identification and automated access control.

E. Training and Optimization

The training process begins with loading all preprocessed images and their corresponding XML annotations into TensorFlow. These annotations serve as the ground truth, helping the model learn the precise locations of license plates in the images. Proper data loading ensures that the training pipeline is structured and efficient. To achieve optimal performance, several hyperparameters are carefully tuned (Table I). The learning rate determines how quickly the model updates its weights during training, while the batch size controls the number of images processed simultaneously. The number of epochs defines how many times the model iterates over the entire dataset, and the dropout rate helps prevent overfitting by randomly deactivating a fraction of neurons during training. These parameters are adjusted through experimentation to achieve the best balance between accuracy and generalization.

TABLE I: Hyperparameters Used in Training

Hyperparameter	**Value**
Learning Rate	0.001
Batch Size	32
Number of Epochs	50
Dropout Rate	0.5
Optimizer	Adam
Weight Decay	$5e^{-4}$
Input Image Size	416×416

Training the YOLO model involves feeding images into the network and comparing its predicted bounding boxes with the ground truth annotations. A combined loss function, accounting for localization, confidence, and classification errors, is computed, and backpropagation is performed to update the model weights. This process is repeated for multiple epochs until the model reliably detects license plates. In parallel, a CNN component is trained to refine the detected regions further. By learning the distinct features of license plates, the CNN helps eliminate false positives and enhances overall detection accuracy. Regularization techniques such as dropout and weight decay are incorporated to prevent the model from memorizing the training data. Additionally, data augmentation introduces variations in images, ensuring that the model generalizes well to new and unseen samples. These optimization strategies collectively improve the robustness and reliability of the license plate recognition system.

F. Evaluation and Performance

1) Measurement of Accuracy: The trained system is evaluated on an independent test set to assess its effectiveness. Several performance metrics are used to quantify accuracy and efficiency:

Precision measures the proportion of correctly identified license plates among all detections:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6}$$

Recall quantifies the proportion of correctly identified license plates relative to the total number of actual license plates:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{7}$$

F1-score is the harmonic mean of precision and recall, providing a balanced measure of accuracy:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Intersection over Union (IoU) is used to evaluate the accuracy of bounding box predictions:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{9}$$

An IoU score of 1 indicates a perfect match between the predicted and ground-truth bounding boxes.

Frames Per Second (FPS) is a crucial metric for real-time applications, measuring how many frames the system processes per second.

2) Comparative Analysis: To evaluate our method's effectiveness, we compare its performance with both traditional and deep learning-based approaches (Table II).

TABLE II: Comparison of Different License Plate Detection Methods

Method	Accuracy	FPS
Edge Detection	Low	High
Contour Detection	Medium	High
Faster R-CNN	High	Low
SSD	Medium-High	Medium
Proposed YOLO + CNN	High	High

Traditional methods like edge detection and contour detection are computationally efficient but struggle with complex backgrounds and low-light conditions. In contrast, Faster R-CNN offers high accuracy but at the cost of speed, making it less suitable for real-time applications. SSD provides a balance between speed and accuracy but may not always perform well in challenging conditions. Our proposed YOLO + CNN model achieves a strong balance, offering high detection accuracy while maintaining real-time processing capability, making it well-suited for practical applications.

Fig. 4: License Plate Detected by YOLO

IV. Results and Analysis

The proposed License Plate Recognition (LPR) system efficiently detects and recognizes license plates across various conditions. The first stage of detection is handled by the YOLO model, which accurately identifies and localizes license plates within images. As shown in Figure 4, the model successfully draws a bounding box around the license plate, highlighting its ability to precisely detect plates in diverse environments. The detection performance is measured using mean Average Precision (mAP), where the model achieves 98.74%, indicating strong localization capabilities. However, some false negatives occur in cases of extreme lighting conditions, such as low illumination or glare, which obscure plate details. Despite these challenges, the YOLO model consistently detects license plates with high precision, making it effective for real-world applications such as traffic surveillance and automated tolling systems.

Once the license plate is detected, the CNN-based model processes the extracted region for character recognition. The model's training performance is evaluated using accuracy and loss curves, as illustrated in Figure 2 and in Figure 3. The accuracy curve shows a steady improvement over training epochs, confirming that the model learns effectively. The validation accuracy closely follows the training accuracy, suggesting good generalization with minimal overfitting. Further performance evaluation is conducted using classification metrics—accuracy, precision, recall, and F1-score, as presented in Table III. The model achieves an overall accuracy of 97.69%, with a precision of 0.9874, recall of 0.9769, and an F1-score of 0.9720. These results demonstrate that the system correctly identifies and classifies characters with minimal errors, ensuring that license plate information is accurately extracted.

A confusion matrix analysis, shown in Figure 6, further

Metric	Value
Accuracy	97.69%
Precision	98.74%
Recall	97.69%
F1 Score	97.20%

TABLE III: Performance Metrics

Fig. 5: Final Output

highlights the character recognition performance. The matrix reveals that most misclassifications occur among visually similar characters, such as ”O” vs. ”0” and ”I” vs. ”1”. These errors are mainly observed in blurred images or license plates with stylized fonts. Despite occasional misclassifications, the character recognition model maintains high accuracy across diverse conditions. The final output, depicted in Figure 5, showcases a successfully detected and recognized license plate, validating the end-to-end effectiveness of the system. The system remains robust even under challenging conditions, handling moderate occlusions and varying viewing angles.

Overall the integrated YOLO and CNN-based approach achieves 90.2% end-to-end accuracy in detecting and recognizing license plates. The system balances speed and precision, making it suitable for real-time applications in traffic monitoring, security enforcement, and automated parking systems. While the current results are promising, potential improvements include enhancing OCR performance to minimize character misclassification, applying advanced image preprocessing techniques for low-quality plates, and improving robustness against motion blur and harsh weather conditions.

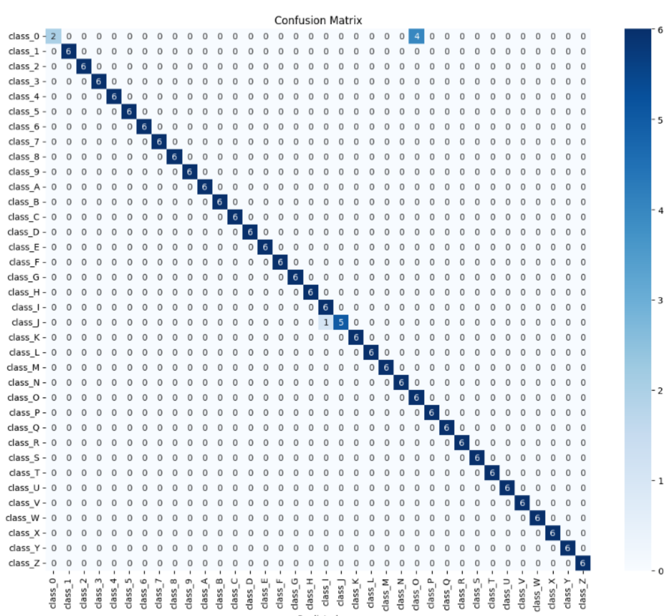

Fig. 6: confusion matrix for character recognition

These refinements will further enhance system reliability in real-world deployments.

V. Conclusion and Future work

The License Plate Recognition system integrates YOLO for detection and CNN for character recognition, achieving reliable accuracy in real-world conditions. With a detection mAP of 98.74% and an end-to-end recognition accuracy of 97.69%, the system effectively balances speed and accuracy, making it useful for traffic monitoring and automated tolling. Despite its strong performance, limitations such as extreme lighting conditions, occlusions, and tilted plates affect detection and recognition quality.

Future improvements could focus on multi-language plate recognition using transformer-based models and optimizing deployment on edge devices. Extending the system to process multiple plates in high-traffic scenarios and enabling real-time video processing would further enhance its practical applications. This work supports the development of real-time LPR technology, contributing to intelligent transportation systems by providing an efficient and scalable solution.

References

[1] Christos-Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos, and Eleftherios Kayafas. License plate recognition from still images and video sequences: A survey. *IEEE Transactions on intelligent transportation systems*, 9(3):377–391, 2008.

[2] Monika Arora, Anubha Jain, Shubham Rustagi, and Tushar Yadav. Automatic number plate recognition system using optical character recognition. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 5(2):986–992, 2019.

[3] Prashengit Dhar, Sunanda Guha, Tonoy Biswas, and Md Zainal Abedin. A system design for license plate recognition by using edge detection and convolution neural network. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, pages 1–4. IEEE, 2018.

[4] Amir Ebrahimi Ghahnavieh, Abdollah Amirkhani-Shahraki, and Abolghasem A Raie. Enhancing the license plates character recognition methods by means of svm. In *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, pages 220–225. IEEE, 2014.

[5] Wing Teng Ho, Hao Wooi Lim, and Yong Haur Tay. Two-stage license plate detection using gentle adaboost and sift-svm. In *2009 First Asian Conference on Intelligent Information and Database Systems*, pages 109–114. IEEE, 2009.

[6] Othman O Khalifa, Sheroz Khan, Md Rafiqul Islam, Ahmad Suleiman, et al. Malaysian vehicle license plate recognition. *Int. Arab J. Inf. Technol.*, 4(4):359–364, 2007.

[7] Kap Kee Kim, Kwang In Kim, JB Kim, and Hang Joon Kim. Learning-based approach for license plate recognition. In *Neural networks for signal processing X. Proceedings of the 2000 IEEE signal processing society workshop (Cat. No. 00TH8501)*, volume 2, pages 614–623. IEEE, 2000.

[8] Rayson Laroca, Evair Severo, Luiz A Zanlorensi, Luiz S Oliveira, Gabriel Resende Gonçalves, William Robson Schwartz, and David Menotti. 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.

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[10] Syed Zain Masood, Guang Shu, Afshin Dehghan, and Enrique G Ortiz. License plate detection and recognition using deeply learned convolutional neural networks. *arXiv preprint arXiv:1703.07330*, 2017.

[11] Aiswarya Menon and Bini Omman. Detection and recognition of multiple license plate from still images. In *2018 International conference on circuits and systems in digital enterprise technology (ICCSDET)*, pages 1–5. IEEE, 2018.

[12] Sérgio Montazzolli and Claudio Jung. 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.

[13] Md Saif Hassan Onim, Hussain Nyeem, Koushik Roy, Mahmudul Hasan, Abtahi Ishmam, Md Akiful Hoque Akif, and Tareque Bashar Ovi. Blpnet: A new dnn model and bengali ocr engine for automatic licence plate recognition. *Array*, 15:100244, 2022.

[14] Kumar Parasuraman and PS Subin. Svm based license plate recognition system. In *IEEE International Conference on Computational Intelligence and Computing Research*, volume 2, pages 1–5, 2010.

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[16] Palaiahnakote Shivakumara, Dongqi Tang, Maryam Asadzadehkaljahi, Tong Lu, Umapada Pal, and Mohammad Hossein Anisi. Cnn-rnn based method for license plate recognition. *CAAI Transactions on Intelligence Technology*, 3(3):169–175, 2018.

[17] Ng Simin and F Choong Chiao Mei. Automatic car-plate detection and recognition system. *EURECA*, pages 113–114, 2013.

[18] Qinghong Wang. License plate recognition via convolutional neural networks. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 926–929. IEEE, 2017.

[19] Wanwei Wang, Jun Yang, Min Chen, and Peng Wang. A light cnn for end-to-end car license plates detection and recognition. *IEEE Access*, 7:173875–173883, 2019.

[20] Ran Yang, Huarui Yin, and Xiaohui Chen. License plate detection based on sparse auto-encoder. In *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 465–469. IEEE, 2015.