Encompassing YOLOv8 and EfficientNet B7 for Automatic License Plate Identification

Rohan Reddy B
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham Chennai-601103, India
rohanbadugula@gmail.com

Gunti Swathi
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham Chennai-601103, India
swathigunti4@gmail.com

Engu Himesh
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham Chennai-601103, India
enguhimesh01@gmail.com

Southarrajan S
Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham Chennai-601103, India
s_sountharrajan@ch.amrita.edu

***Abstract*—The rapid urbanization and increased vehicular population in metropolitan areas have presented a formidable obstacle to the manual identification of license plates. The surge in vehicular activity has led to a complex task, requiring efficient surveillance of each vehicle for vital functions such as theft prevention and traffic regulation. In response to this intricate challenge, the deployment of an Automatic License Plate Recognition (ALPR) system emerges as a crucial necessity for smart city transportation. This research aims to develop an efficient ALPR system employing YOLOv8, the latest iteration in the YOLO series of object detection models, for license plate localization, and utilizing EfficientNet B7 for character recognition on the detected plates. The YOLOv8 license plate detection module achieved a mean Average Precision (mAP 50) of 99.5%, ensuring robust and reliable license plate localization. The character recognition component, utilizing EfficientNet B7, achieved an accuracy of 98.22%, further enhancing the system's overall performance and determining the amalgamation of YOLOv8 and EfficientNet B7 as an optimal solution. To validate the efficacy of our ALPR system, a diverse set of test data comprising various license plate types is employed. The system demonstrates remarkable identification accuracy, achieving a recognition rate of 96.6%. These results highlight the practical feasibility and efficacy of our proposed ALPR system in real world applications.**

***Index Terms*—Automatic License Plate Recognition, YOLOv8, EfficientNet B7, License plate localization, Character recognition, mean Average Precision**

I. Introduction

The advancement of Automatic License Plate Recognition (ALPR) systems has brought about a significant transformation in modern traffic control and safety protocols [1]. The conventional practice of manually identifying vehicles is not only time-consuming but also monotonous and repetitive. Consequently, there is a pressing need for automated systems capable of swiftly and accurately identifying vehicles through their license plate numbers. The applications of ALPR technology are far-reaching, encompassing tasks such as detecting traffic violations, streamlining toll collection, enabling vehicle tracking, and managing unmanned parking facilities [2].

In the realm of scene analysis, the progression and implementation of ALPR systems revolve around addressing distinct layers of analysis. These layers are essential for accurate identification and recognition of license plates within a given scene. They encompass multiple crucial stages, including Number plate detection, Character segmentation, and character recognition [3]. Traditionally, studies in computer vision have approached these challenges using methods like color texture analysis [4], edge detection [5], morphological operations [6] and template matching [7]. However, these techniques often prove to be complex, inefficient, and susceptible to issues such as uneven lighting, noise, and variations in font styles. To enhance the accuracy and reliability of ALPR systems in real-world settings, there's a growing emphasis on adopting advanced deep learning models [8, 9]that can effectively learn intricate features, patterns, and variations present in license plates.

In this research, we propose a two-stage license plate recognition system. In the first stage, we employ YOLOv8 (You Only Look Once Version 8) [10], the latest iteration of the YOLO algorithm initially introduced in 2016, to detect the license plate from the image. The detected license plate undergoes processing through various image processing techniques and is then passed to the second stage. EfficientNet B7 [11], a pretrained CNN model, is utilized in the second stage to recognize and extract the characters present on the processed license plates. The integration of YOLOv8 with EfficientNet B7 results in significant enhancements in both performance and accuracy.

II. Related Works

The escalating importance of license plate recognition has propelled it to a prominent position, particularly within the realm of smart transportation systems. The automated process of extracting license plates comprises two indispensable components: license plate detection and license plate recog-

ISBN: 979-8-3503-0611-8/23/$31.00 ©2023 IEEE

nition, functioning as separate yet interconnected processes. Traditional methods encompass image processing techniques like color texture analysis [4], edge detection [5] and template matching [7]. Hongilong et al.(2004) [6], employed morphological operations applied to a contour detector to identify rectangular regions corresponding to license plates. Similarly, Zhang et al.(2015) [12] utilized a color difference model to binarize images, applying the Adaboost algorithm for precise license plate localization. Kim et al.(2002) [4] focused on low-light image conditions, using contour extraction to localize license plates. Khalil et al.(2010) [7] employed Template Matching-based Methods with modules for image acquisition, license plate extraction, segmentation and character recognition utilizing template matching for segmentation-free accuracy and enhancing information extraction from number plate images.

Initially, the recognition process entailed sequential character segmentation of license plates, followed by optical character recognition for character identification. The approach is subsequently extended through the integration of Machine Learning and Deep Learning models. Pavani et al.(2023) [13] presented a license plate recognition system using image processing and KNN algorithm for authorized vehicle verification at restricted entry points, demonstrating accurate differentiation between real and fake license plate images. Omar et al.(2020) [14] utilized preprocessing procedures, including Gaussian filtering and adaptive contrast enhancement, to optimize the quality of input images, and a semantic segmentation network to identify license plate regions using deep encoder-decoder architecture. In another work Shatnawi et al.(2016) [15] adopted the honeybee algorithm for character segmentation, followed by support vector machine (SVM) for character recognition. Zhang et al.(2023) [16] integrated Hough transform for correction and segmentation, along with the fusion of Firefly Algorithm and Back Propagation for character recognition. Patil et al.(2023) [17] introduced an improved license plate recognition system using OCR. The method involves fine-tuning a ResNet50 model and training a CNN model on a custom dataset. The CNN model outperformed the ResNet50 model, offering higher accuracy and faster computation for real-time applications. Slimani et al.(2020) [18] combined two-dimensional wavelet transform for plate detection, validation of the detection by a CNN classifier designed for discriminating between plates and non-plates, followed by character segmentation and classification using additional CNNs. Dang et al.(2023) [19] utilized MATLAB and employs image processing techniques to address unclear license plate challenges, enabling precise deciphering of Chinese characters, numbers, and letters by investigator.

Previous machine learning and deep learning approaches [9] exhibited shortcomings in terms of both accuracy and speed. To address this, comprehensive object detection models like YOLO and Residual-CNN were introduced for accurately locating and recognizing license plates in real-world scenarios. Nandhakumar et al.(2023) [20] used YOLOv2 and YOLOv3 object detection models for license plate recognition with OCR. Notably, YOLOv3 exhibited superior performance over YOLOv2. Rattanakorphan et al.(2022) [21] presented a LPR system specialized for Thai license plates utilizing YOLOv5, morphology, and CNNs. Notably, a particular study [22] combines Image processing and deep learning tools like OpenCV, YOLO, PaddleOCR, Tesseract OCR to gauge their efficacy in detecting and identifying number plates across diverse environmental conditions. Aljelawy et al.(2022) [23] introduced a car license plate detection and recognition system comprising image processing, YOLOv4-based plate detection, character extraction and Python-based recognition using OpenCV and easy OCR libraries. Shi et al.(2023) [24] proposed an end-to-end deep learning model that enhances license plate recognition accuracy and speed. It integrates improvements to YOLOv5, reduces parameters, and applies GRU+CTC for segmentation-free recognition. Sinthia et al.(2023) [8] Integrated YOLOv6 with BLPNET (VGG-19-RESNET-50), for license plate identification and character recognition and proven its superiority on the RCNN and MobileNet.

The research findings indicate that the YOLO model, utilized for object detection, have demonstrated significant improvements in both speed and accuracy. Building upon this our work extends these advancements by implementing the latest iteration of YOLO, known as YOLOv8, for object detection tasks. Additionally, we have incorporated the EfficientNet B7 architecture to enhance recognition capabilities on the identified objects.

III. Data Description

The selection of appropriate datasets for license plate identification is a critical determinant of accuracy, contingent upon various factors, with dataset diversity emerging as paramount for model robustness. To address this, we have leveraged the Artificial Mercosur License Plates Dataset [25], encompassing license plates from five distinct South American countries, thereby enriching the Detection phase of our proposed model. This dataset encompasses a comprehensive collection of 3,840 images, sourced from monitoring systems and parking lots, meticulously annotated according to the YOLO standard. These images originate from a publicly accessible traffic monitoring camera video stream, which also operates at a resolution of 800×600. For the Recognition of the data in the Detected license plate, we utilized an open-source character recognition dataset [26] that consists of all the digits and letters, encompassing 36 distinct classes that are encoded numerically from 0 to 35. Comprising a total of 37,623 monochromatic images each of 28×28 pixels images , showcasing a balanced distribution across characters, rendering it ideal for effective training and evaluation in license plate character recognition.

IV. Methodology

In our research, we have implemented cutting-edge technologies from various relevant domains to achieve the precise detection and recognition of license plate numbers.This intricate process encompasses license plate detection within images, followed by the application of image processing

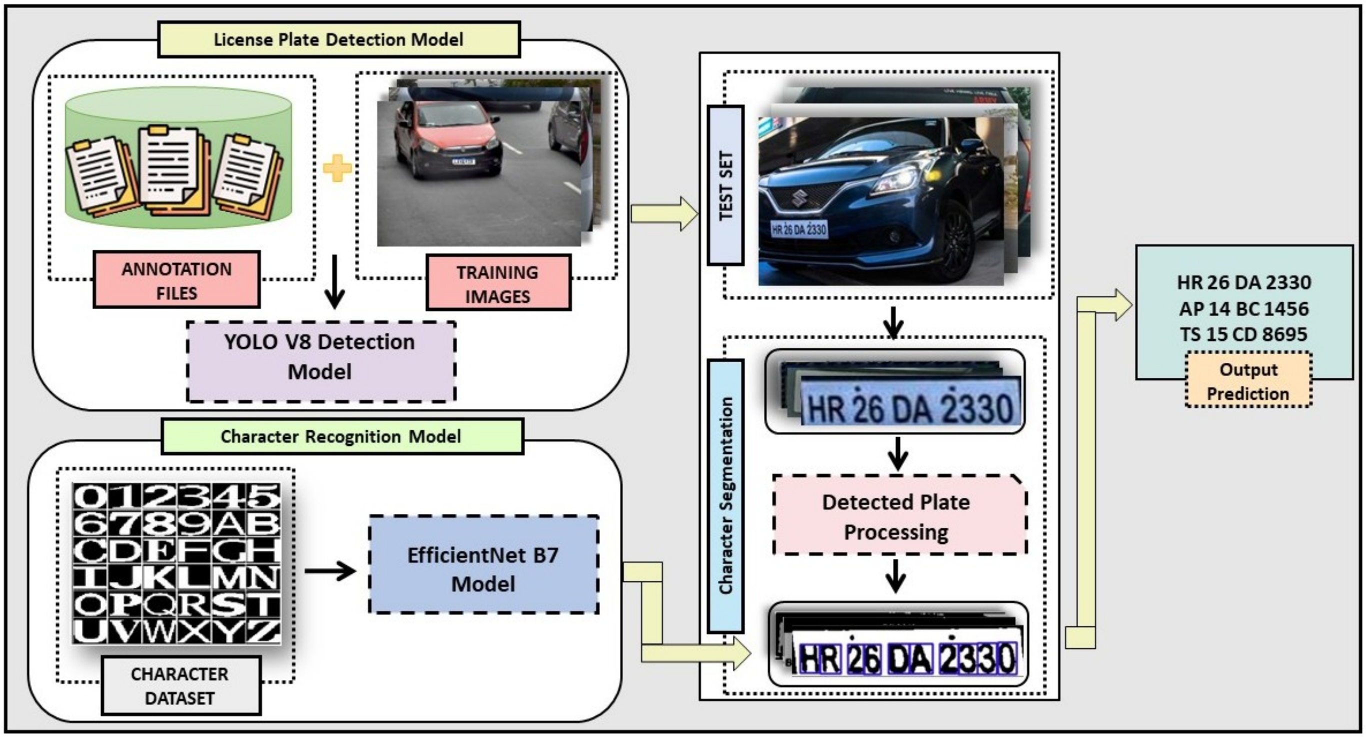

Fig. 1: Proposed Framework: YOLOv8 - EfficientNet B7 for License Plate Identification

techniques on the plate to make them compactable for the classifier. Subsequently, contour segmentation is employed to isolate individual characters, enabling the recognition and assembly of these characters into a coherent alphanumeric string, ultimately resulting in the extraction of the license plate number. A comprehensive depiction of the proposed methodology is provided in Figure 1.

A. License plate detection

We have employed YOLOv8 for license plate detection in images. YOLOv8 represents an advanced single-step object detection algorithm developed by Ultralytics [10]. Building upon the foundation of YOLOv5, YOLOv8 incorporates a hierarchical structure that enhances its capacity to learn from a wide range of classes, enabling precise object detection, even within complex images. This architecture introduces significant and novel improvements, including an anchor-free detection head, a novel backbone network, a pioneering loss function, and various other enhancements. The introduction of a novel spatial pyramid pooling layer within the architecture not only reduces computation requirements but also enhanced accuracy. Notably, the algorithm's unique mix-up of data augmentation techniques contributes to enhanced performance even with limited training data.

In contrast to the traditional anchor-based approach that relies on fixed anchor boxes, YOLOv8 employs an anchor-free methodology. This approach avoids the use of pre-determined anchor boxes and instead predicts bounding boxes directly, leading to significantly improved accuracy in object detection.

The license plate detection using YOLOv8 algorithm takes an image containing the vehicle, along with its associated annotation file, as an input and produces a set of bounding boxes, class labels and confidence scores for the detected objects within the image. In our work the focus is solely on license plate detection, entailing the classification of two classes: license plate and the background. According to the YOLOv8 algorithm the image is first divided into a grid of cells. For each cell the presence of an object along with its associated bounding box is detected. Subsequently, the image is passed through a single neural network that extracts high-level features from an image. These features play a pivotal role in predicting the co-ordinates of the multiple bounding boxes, the probability of each class in the box and the confidence score. A non-maximum suppression is employed on the predicted bounding boxes to filter out overlapping boxes. At the end the results are post-processed to remove the low-confidence bounding boxes and merging overlapping boxes corresponding to the same object.

B. Image Processing for Classification

In the post-detection phase, the processing of the image identified by YOLOv8 necessitates the refinement of object detection outcomes. Specifically, when dealing with license plate detection, the initial step involves isolating the license plate region by cropping the bounding box from the original image. Subsequently, a series of image processing techniques are applied to this isolated region, aimed at extracting individual characters present on the license plate. The "findContours"

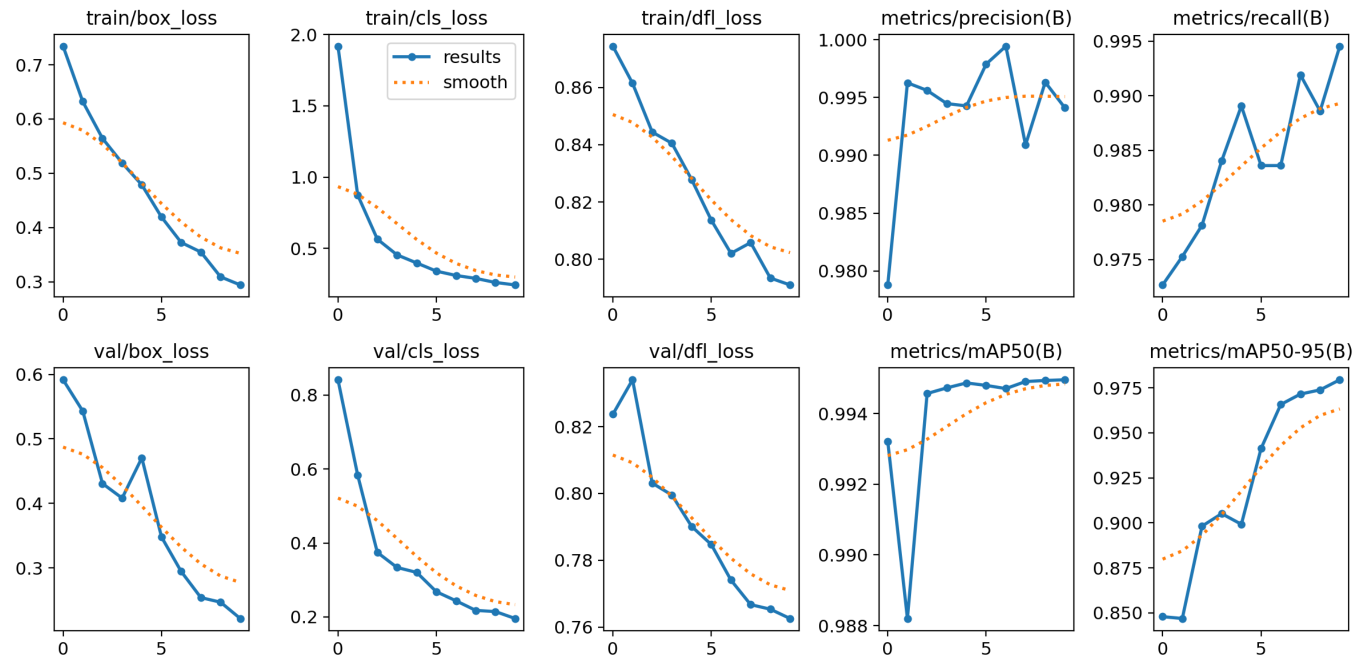

Fig. 2: Visual Analysis of YOLOv8 Performance Metrics During Training and Validation

method from the CV2 library [27] is utilized to detect the contours within the cropped image. By applying certain parameter adjustments and fine-tuning, we had identified and retained only those contours that correspond to rectangular shapes containing the characters of interest. The identified contours need to be disassembled into discrete entities for character-level recognition, so further preparatory procedures are implemented on the contoured image. Initially, image resizing was conducted to enhance character visibility, succeeded by grayscale conversion to prepare for subsequent stages. The grayscale image was then subjected to thresholding, transforming it into a binary representation using a threshold value of 200. Subsequent erosion operations aimed to eliminate superfluous boundary pixels, while dilation served to fill absent pixels within the image. Further refinement involved rendering image boundaries white, potentially eliminating any external pixels. A predefined list of dimensions facilitated the screening of pertinent characters. The processed images are made compactable for character recognition model after the implementations of these processing techniques.

C. Character Recognition of processed License plate

The character recognition model serves as a crucial component within the ALRS system, responsible for identifying individual characters from processed and segmented images. To accomplish this task, we have employed the EfficientNet B7 model. EfficientNet is a pre-trained CNN model that has undergone training on the Imagenet dataset [28]. It utilizes the Mobile Inverted Bottleneck MBConv as its core component for building its architecture. This framework has 8 variations, spanning from EfficientNet B0 to EfficientNet B7. As the model index increases, there's a remarkable surge in accuracy without a proportional escalation in the number of parameters. Notably, EfficientNetB7 stands out as the top-performing model, surpassing prior well established CNNs in terms of ImageNet accuracy. Moreover, it achieves this while being 6.1 times faster and 8.4 times more compact than the best in comparison to the most advanced existing CNN models [11]. The character recognition model serves as the final component of the ALRS system, generating an output string that identifies the license plate number present in the initial provided image.

V. Experiment and Results

The experimentation of the all the models were conducted using a NVIDIA TESLA P100 GPU that provides 1.6x more GFLOPs and stacks 3x the memory bandwidth of the K80. The automatic license plate detection and recognition task necessitated the utilization of two specific datasets as detailed in the Dataset Section. A split of 80% for training and 20% for testing was employed for both datasets. The YOLOv8 model was employed, leveraging pre-trained weights, and underwent 10 epochs of training. Detailed performance metrics for the training and validation of the detection model are graphically represented in Figure 2. The Y-axis in each graph represents the range of corresponding metrics as indicated in the title of each sub-graph, while the X-axis corresponds to the number of epochs. The bounding box loss, class loss, and detection loss values being recorded as 0.9697, 0.9219, and 1.197, respectively.

In the pursuit of developing a plate recognition model, several widely recognized and ImageNet-proven pretrained CNN models [29], including EfficientNet B0, MobileNet V2, ResNet50, EfficientNet B7 and Inception V3 are utilized. These models were trained on the character recognition dataset delineated in the Dataset Section, each undergoing 15 epochs of training. The compilation of all models was performed using the Adam optimizer [30]. Throughout the training phase, the disparity between the actual class labels and the predicted class probabilities was measured using the categorical cross-entropy loss function with a learning rate of 0.0001. The Validation losses of all the models are shown in Figure 3.

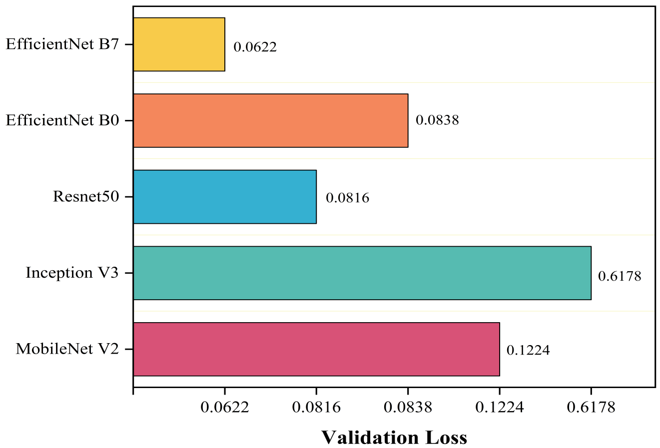

Fig. 3: Validation Loss Analysis of Various Pre-trained CNN

To assess the efficacy of the YOLOv8 detection model, a comprehensive analysis was conducted, encompassing metrics such as Mean Average Precision (mAP), Bounding Box Precision (BBP), and Bounding Box Recall (BBR).

$$BBP = \frac{\text{Number of Correctly Localized Boxes}}{\text{Total Number Of Predicted Boxes}} \tag{1}$$

$$BBR = \frac{\text{Number of Correctly Localized Boxes}}{\text{Total Number of Ground Truth Boxes}} \tag{2}$$

mAP quantifies the accuracy of predictions by computing the average precision across different object classes, providing a comprehensive overview of a model's detection capabilities. mAP50 focuses on a specific aspect of detection precision, considering predictions where the Intersection over Union (IoU) between predicted and ground-truth bounding boxes exceeds a threshold of 0.5. On the other hand, mAP50-95 provides a broader perspective by considering a range of IoU thresholds from 0.5 to 0.95, reflecting the model's performance across varying degrees of spatial overlap between predictions and ground-truth objects. The outcomes corresponding to the metrics mentioned above for the YOLOv8 model, evaluated on the test dataset are presented in Figure 4. The evaluation of pretrained CNN recognition models involved a comprehensive examination of their performance through the prism of Accuracy, Precision, Recall, and F1 score.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{3}$$

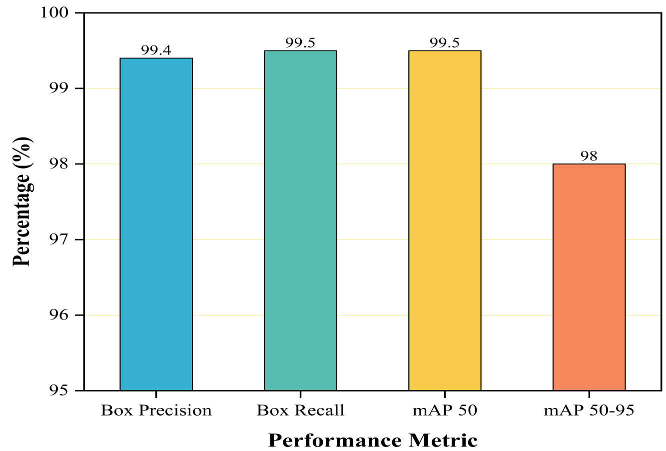

Fig. 4: Bounding Box Performance Metrics for License Plate Detection Model

$$\text{Precision} = \frac{\sum TP_c}{\sum TP_c + \sum FP_c} \tag{4}$$

$$\text{Recall} = \frac{\sum TP_c}{\sum TP_c + \sum FN_c} \tag{5}$$

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

Here, TP_c represents the instances where character 'c' is correctly identified. FP_c represents the instances where character 'c'is incorrectly identified as present. FN_c represents the instances where character 'c' is not recognized despite being present. These metrics on all the pretrained models for character recognition are given in Table I.

TABLE I: Performance metrics of various pre-trained CNN models

Models	Accuracy	Precision	Recall	F1 Score
MobileNet V2	96.39%	97.29	96.11	96.70
IncceptionV3	85.04	90.77	83.09	86.73
Resnet50	97.37	97.96	96.99	97.47
EfficientNet B0	97.32	98.17	96.57	97.57
EfficientNet B7	**98.22**	**98.40**	**98.11**	**98.25**

The evaluation of various pretrained CNN models determined that the EfficientNet B7 model exhibits superior performance, establishing it as the most suitable choice for license plate character recognition. Thus, we propose a two-stage ALPR system that combines YOLOv8 for license plate detection and EfficientNet B7 for character recognition from the detected license plate. To assess the system's robustness, an independent and novel Testing set was employed.This testing set consisted of 500 images, manually captured using a mobile phone with a resolution of 1080 x 1920 pixels. These images predominantly featured Indian vehicles commonly found in university premises, residential areas, and some traffic-congested regions. With 500 images, the system achieved 483 accurate predictions, with only 17 instances of misidentification, often due to minor errors in recognizing 1 or 2 characters on the plates. Consequently, the overall accuracy of the two-stage ALPR system reached 96.6%, affirming its efficacy. The model has also been evaluated on the image taken by a mobile phone, the processing of the image is shown in Figure 5. The accurate prediction determines the model's efficiency for real-world applications.

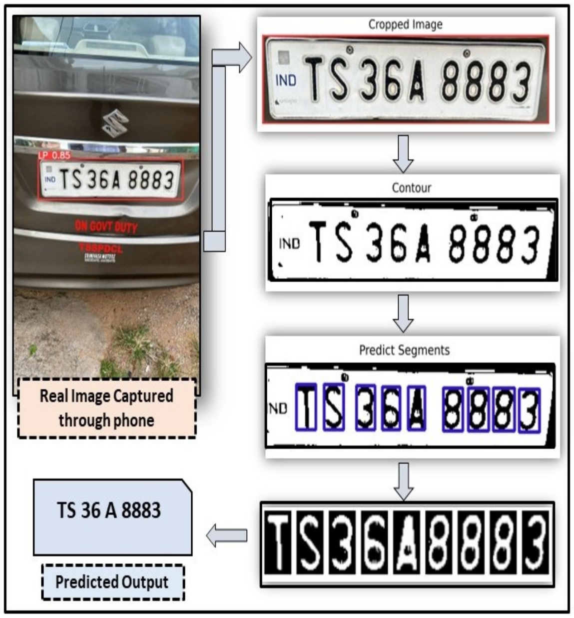

Fig. 5: Model Prediction on Image captured via Phone

VI. Conclusion

This work demonstrated an Enhanced Automatic License Plate Recognition system, which integrates YOLOv8 for precise object detection and EfficientNet B7 for character recognition, strategically catering to both accuracy and speed in the context of smart city transportation. Notably, this research is pioneering, as it is the first to leverage YOLOv8-based methods for License Plate Recognition, a choice driven by YOLOv8's speed and its distinct model architecture. The proposed system achieved 99.5% mean Average Precision in license plate detection and a robust 98.22% accuracy in recognizing the license numbers the detected plate, reaffirming the system's potential for optimizing traffic management. Notably, its adaptability shines through with a 96.6% accuracy on previously unseen data, solidifies its reliability in dynamic scenarios. The system's efficacy in real-world image analysis further solidifies its relevance in practical smart city implementations, offering a forward-looking solution to the ever-evolving realm of urban mobility challenges.

In future, the creation of a multifaceted dataset encompassing diverse number plate characteristics, including regional languages, varying font sizes, and license plate dimensions, along with an array of colors like yellow and green, , enhancing their resilience in the face of challenging real life conditions . Concurrently, exploration into the augmentation of ALPR system integration with other systems is essential, facilitating real-time data provision and supporting decision-making processes. Moreover, the imperative lies in augmenting processing speeds and identification efficiency. This strategic arrangement aligns with the envisioned future developments in ALPR technology, poised to enhance its adaptability and practical utility.

References

[1] F. Alegria and P. Girão, "Vehicle plate recognition for wireless traffic control and law enforcement system," pp. 1800 – 1804, 01 2007.

[2] G.-W. Chen, C.-M. Yang, and T.-U. İk, "Real-time license plate recognition and vehicle tracking system based on deep learning," pp. 378–381, 2021.

[3] D. Sarma, A. Bora, and A. Bhagat, "Automatic license plate detection and recognition system for security purposes," pp. 1–5, 2023.

[4] K. I. Kim, K. Jung, and J. H. Kim, "Color texture-based object detection: An application to license plate localization," pp. 293–309, 2002.

[5] P. Sagharichi Ha and M. Shakeri, "License plate automatic recognition based on edge detection," pp. 170–174, 04 2016.

[6] B. Hongliang and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," vol. 2, pp. 831–834 Vol.2, 2004.

[7] M. Khalil, "1car plate recognition using the template matching method," *International Journal of Computer Theory and Engineering*, vol. 2, pp. 683–687, 01 2010.

[8] C. Sinthia and M. H. Kabir, "Detection and recognition of bangladeshi vehicles' nameplates using yolov6 and blpnet," pp. 1–6, 2023.

[9] W. Puarungroj and N. Boonsirisumpun, "Thai license plate recognition based on deep learning," *Procedia Computer Science*, vol. 135, pp. 214–221, 2018, the 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life.

[10] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, 2023.

[11] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019.

[12] X. Zhang, P. Shen, J. Song, L. Zhang, W. Gong, W. Wei, Y. Tian, and G. Zhu, "An algorithm combined with color differential models for license-plate location," pp. 429–438, 09 2015.

[13] V. Pavani, K. Divya, V. V. Likhitha, G. S. Mounika, and K. S. Harshitha, "Image segmentation based imperative feature subset model for detection of vehicle number plate using k nearest neighbor model," pp. 704–709, 2023.

[14] N. Omar, A. Sengur, and S. Al-Ali, "Cascaded deep learning-based efficient approach for license plate detection and recognition," *Expert Systems with Applications*, vol. 149, p. 113280, 02 2020.

[15] N. Shatnawi, "Bees algorithm and support vector machine for malaysian license plate recognition," *International Journal of Business Information Systems*, vol. 28, 11 2016.

[16] K. Su, L. Zhang, and J. Wang, "License plate image recognition system based on firefly algorithm and bp neural network," pp. 153–157, 2023.

[17] A. Patil, J. Ingale, A. Upadhye, S. Jaiswal, S. Kakade, and A. Bhatt, "Deep learning-based approach for indian license plate recognition using optical character recognition," pp. 1–5, 2023.

[18] I. Slimani, A. Zaarane, W. Okaishi, A. Issam, and A. Hamdoun, “An automated license plate detection and recognition system based on wavelet decomposition and cnn,” *Array*, vol. 8, p. 100040, 09 2020.

[19] L. Dang, M. Yang, B. Si, and K. Yi, “A fuzzy license plate image processing technology based on matlab,” pp. 936–939, 2023.

[20] R. G. Nandhakumar, K. N. Devi, N. Krishnamoorthy, S. Shanthi, V. R. Pranesh, and S. Nikhalyaa, “Yolo based license plate detection of triple riders and violators,” pp. 1–6, 2023.

[21] R. Rattanakornphan, P. Keeratiwintakorn, and T. Ruttanapun-yagorn, “Thai license plate recognition based on yolo detector and cnn-classifier,” pp. 331–336, 2022.

[22] T. Mustafa and M. Karabatak, “Deep learning model for automatic number/license plate detection and recognition system in campus gates,” pp. 1–5, 2023.

[23] Q. M. Aljelawy and T. M. Salman, “Detecting license plate number using ocr technique and raspberry pi 4 with camera,” pp. 1–5, 2022.

[24] H. Shi and D. Zhao, “License plate recognition system based on improved yolov5 and gru,” *IEEE Access*, vol. 11, pp. 10 429–10 439, 2023.

[25] V. Ribeiro, V. Greati, A. Bezerra, G. Silvano, I. Silva, P. T. Endo, and T. Lynn, “Brazilian mercosur license plate detection: a deep learning approach relying on synthetic imagery,” pp. 1–8, 2019.

[26] A. Patil, J. Ingale, A. Upadhye, S. Jaiswal, S. Kakade, and A. Bhatt, “Deep learning-based approach for indian license plate recognition using optical character recognition,” pp. 1–5, 2023.

[27] A. Datta, O. K. Yadav, Y. Singh, S. S, K. M, and S. E, “Real-time face swapping system using opencv,” pp. 1081–1086, 2021.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” pp. 248–255, 2009.

[29] R. Tharsanee, R. Soundariya, A. S. Kumar, M. Karthiga, and S. Sountharrajan, “7 - deep convolutional neural network–based image classification for covid-19 diagnosis,” pp. 117–145, 2021.

[30] I. V. Srisurya and R. Prasanna Kumar, “Neural machine translation using adam optimised generative adversarial network,” pp. 383–387, 2023.