ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

Deep Learning Based Automatic License Plate Detection and Extraction

[1] Rashmita Pandey, [2] Uday Joshi

[1] [2] K. J. Somaiya College of Engineering/Computer, Mumbai, India
Corresponding Author Email: [1] rashmita.pandey@somaiya.edu, [2] udayjoshi@somaiya.edu

***Abstract*— License plate detection and extraction are essential tasks in various applications, such as traffic monitoring, parking management, and law enforcement. In this research, we have explored the use of YOLOv3, YOLOv4, and YOLOv5 models for license plate detection and evaluated their performance. We found that the YOLOv5 model outperforms the other models in terms of accuracy and speed. For license plate number extraction, we have used the Easy OCR model, which provides state-of-the-art performance in recognizing license plate characters. The proposed system was trained on the Open Image dataset and tested on the dataset captured by us, achieving an overall mAP50 of 88% for license plate detection and 87% for Vehicle detection. Our results demonstrate the potential of using deep learning models for license plate detection and recognition, and we believe that our findings will be useful for future research in this field.**

***Index Terms*—automatic license plate detection, easyocr, mAP50, precision, recall, yolov3, yolov4, and yolov5.**

I. INTRODUCTION

India faces a significant challenge in monitoring and controlling traffic violations due to the country's large population, increasing commuter numbers, poor traffic signal management, and a disregard for traffic laws. According to the National Crime Records Bureau (NCRB) report of 2021, two-wheelers were responsible for the majority of fatalities, accounting for nearly 70,000 deaths or 44.5% of total accidents. Cars followed with 23,531 deaths or 15.1%, while trucks accounted for 14,622 deaths or 9.4%. Given the high volume of traffic, relying solely on traffic monitoring officer is inadequate to monitor and identify offenders. As a result, many violators go undetected, leading to more severe accidents that endanger both the offender's life and the lives of others. To overcome this problem, Artificial Intelligence based solutions is required to replace human intervention in identifying and capturing violators.

Given its numerous uses in traffic management, security, and surveillance systems, Automatic License Plate Detection (ALPD) technology has attracted a lot of attention recently. ALPD typically comprises two primary components: License Plate Detection and License Plate Number Extraction. License Plate Detection involves locating license plates in digital images or videos, whereas License Plate Recognition involves identifying the characters on the license plate and converting them into text.

In this research paper, we present a novel method for automated license plates detection and extraction based on the YOLO object detection model and the EasyOCR optical character recognition library.

The remaining sections of this paper are structured as follows. In Section 2, we discuss prior work on the detection and recognition of license plates using deep learning. In Section 3, we give a detailed description of YOLOv3, YOLOv4, and YOLOv5 along with their architecture. In section 4, we describe the proposed method in detail, including the dataset, model architecture, and experimental setup. Section 5 presents the experimental results and compares our method with other approaches.

II. LITERATURE REVIEW

Reference [1] came up with an innovative, time-saving automatic license plate recognition system based on YOLOv5 model. The solution comprises of two stages: an LSTM-based OCR engine for plate recognition and a customized transfer learning model for plate detection. The authors created their dataset using the Indian License Plate dataset and the Google Open Images collection. In order to compare size and performance, they also trained YOLOv4 models on the same dataset. The proposed ALPR system generates a 14-megabyte model with an average mean precision of 87.2% when tested with an Nvidia T4 GPU on still images.

Reference [2] proposes a system for integrated vehicle type and license plate recognition utilizing YOLOv4. The system has features for detecting the type of vehicle, license plate, and license plate character. The proposed solution employs one to four multilane images to identify licence plates and six distinct vehicle types. The system's mAP values were 98.0%, 94.0%, 97.1%, and 84.6%, respectively, for vehicle type detection, licence plate detection, and licence plate character detection. The suggested technique yielded mAP values of 99.3 and 99.4 percent for licence plates detected using the author's dataset and a publicly accessible dataset, respectively. The system can detect licence plates as narrow as 100 pixels wide from high-resolution 4K images.

Reference [3] proposes a rapid and precise automatic license plate recognition system designed specifically for processing small regions of interest (ROIs). By enabling efficient detection within the image, this approach alleviates the burden of manually setting the ROI area while reducing computational requirements. This study revealed that employing the YOLOv4 detector significantly enhances performance. Specifically, the 256x256 network achieved a mAP@50 of 98.36%, outperforming the YOLOv3 detector trained on a 416x416 network, which attained a mAP@50 of 71.53%.

Reference [4] discusses the machine learning models and approaches used in the development of autonomous driving systems. The effectiveness of these models is compared based on the time it takes them to predict a single image. The paper covers deep learning models for movement prediction, research gaps in existing studies, and proposed solutions. The author begins by discussing a basic net architecture for object detection before moving on to more advanced models such as YOLO and Fast RCNN. The author concludes that YOLO is the most accurate and fastest algorithm for object recognition.

Reference [5] proposes a three-step method for license plate detection, multiple nation License plate layout detection and unitary character recognition, which is predominately based on you only look once networks. YOLOv3-SPP, which contains the spatial pyramid pooling (SPP) block, is utilized in the second phase. YOLOv3-SPP is used for character recognition, yielding the expected character bounding frames but omitting the license plate number sequence information. An incorrect license plate number sequence would ensure accuracy. A layout identification technique is proposed in this paper in order to retrieve the precise sequence of license plate characters from foreign license plate. KarPlate is a dataset of Korean license plates that the authors made available to the public. The proposed method is evaluated using license plate datasets from five countries: South Korea, Taiwan, Greece, the United States, and Croatia. In addition, a small dataset of license plates from seventeen countries is collected to evaluate the performance of the global license plate layout detection method.

Reference [6] proposes a new approach for object detection, called the Improved Warped Planar Object Detection Network (IWPOD-NET). This method can warp a license plate to a front-parallel view and correct any distortions related to perspective. The detector can identify a license plate's four corners in many different kinds of circumstances. The study also assessed two optical character recognition methods based on corrected license plate object detection. Results show that even with a relatively limited training set, the suggested detector performs on par with cutting-edge methods. In terms of the overall Automatic License Plate Recognition (ALPR) results, this method yields the highest-scoring results for a variety of datasets that encompass various capture conditions and vehicle kinds, especially motorcyclists.

The system proposed in Reference [7] is designed to detect licence plates on vehicles in challenging conditions, including distorted, dim or bright light, and polluted environments. The method detects a moving vehicle's licence plate using Faster R-CNN and a surveillance camera situated in a high-traffic area or another suitable location. The system extracts the licence plate from the video using frame segmentation and picture interpolation techniques in order to produce superior results. Using an optical character recognition (OCR) technique, the plate number is then extracted from the image. The proposed system accurately identified a vehicle's licence plate 99.1% of the time.

Reference [8] proposes a method for vehicle tracking, recording, and verification that utilizes automatic license plate recognition. The system authenticates automobiles against records in a database to provide information about the vehicle. To accomplish this, the system uses CNNs and GRUs in sequence modeling. In addition, the system incorporates a GSM module that can notify the car owner of the vehicle's location via SMS upon request. Instead of using picture segmentation, the proposed approach trained on a dataset of 400 photos with various font types. The approach achieved an accuracy of 98% in recognizing characters and 88% in recognizing whole license plates.

Reference [9] introduces an Automatic License Plate Recognition (ALPR) system that relies on image processing for license plate detection. In Myanmar, license plates are characterized by a white border, and thus the proposed system adopts an edge-based approach that can work with plates of any color. By detecting edges, the system can identify the white-bordered plate region. Morphological operations are then applied to the image to refine the object and extract the license plate more precisely. Finally, the use of bounding box technology allows for accurate extraction of the license plate region.

Reference [10] recommends implementing an Automatic License Plate Recognition system based on the you only look once object detection algorithm, which has proved to be a reliable and efficient method for license plate recognition. The authors trained the ALPR system's Convolutional Neural Networks under several irrelevant conditions, including variations in camera, illumination, and background. The authors also developed a two-step method for character segmentation and recognition using low-tech data augmentation techniques such as inverted characters and license plates (LPs) in reverse. With recognition rates of 89.80% and 93.03%, respectively, the proposed ALPR system outperformed the commercial systems Sighthound and OpenALPR. With a recognition rate of 93.53%, it outperformed both systems in the SSIG dataset, which contains 2,000 frames from 101 vehicle videos. In addition, a two-stage strategy for character segmentation and recognition was implemented in the study using low-tech

IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

data augmentation techniques, such as license plate (LP) inversions and reversed characters.

The Automatic License Plate Recognition system proposed in Reference [11] consists of four phases: data collection and data processing, feature extraction, region of interest (ROI) selection, and classification. Grayscale scaling, median filtering, thresholding, and masking were used as preprocessing techniques. Following CNN training, a CNN masking layer was applied to enhance image quality. CNN finally recognized the information on the license plate. Precision, recall rate, and recognition rate were used to evaluate the performance of the MATLAB-created system. With a recognition rate of 98.13 percent, the ALPR system successfully identified 160 images, including multi-line and crooked license plates. However, it was observed that the EasyOCR model occasionally produced inaccurate results when applied to areas of interest.

III. DETECTION MODELS

A. YOLOv3:

YOLOv3 or You Only Look Once version 3 is a deep learning based object detection model that has gained popularity due to its efficiency. It is an improved version of its predecessors and is capable of real-time object detection even in complex and diverse settings.

YOLOv3 utilizes a multi-scale prediction method, where the input image is processed at multiple scales, and the results are combined to produce the final output. This allows YOLOv3 to detect objects of various sizes within the same image and handle scenarios where objects are taking up a small or large portion of the image.

Figure 1 depicts the architecture of YOLOv3, which consists of numerous convolutional and fully connected layers that comprise a deep convolutional neural network (CNN). The key characteristics of the YOLOv3 architecture are described below.

- Backbone Network: The YOLOv3 model utilizes a backbone network, like Darknet-53, to act as its feature extractor. The purpose of this network is to generate feature maps from the input image, which are then utilized by the remaining sections of the model for object detection.
- Detection Layers: The detection layers in YOLOv3 are its main building blocks because they predict objects' existence and features in the input image. Three detecting layers are used by YOLOv3, each of which runs at a different resolution to accommodate objects of different sizes. To calculate item predictions, these layers use anchor boxes, non-maximum suppression (NMS), and intersection over union (IoU) approaches.
- Skip Connections: The YOLOv3 model utilizes skip connections to facilitate the direct transfer of information from the backbone network to the detection layers. This strategy improves the accuracy of the predictions and allows the network to handle complex and dynamic environments.
- Output Layer: The last layer in the YOLOv3 model is tasked with generating the predictions for the input image. This layer produces multiple bounding boxes for each detected object, along with a corresponding probability for each box. The output of the model consists of bounding boxes, class probabilities, and confidence scores, which together identify the objects present in the input image.

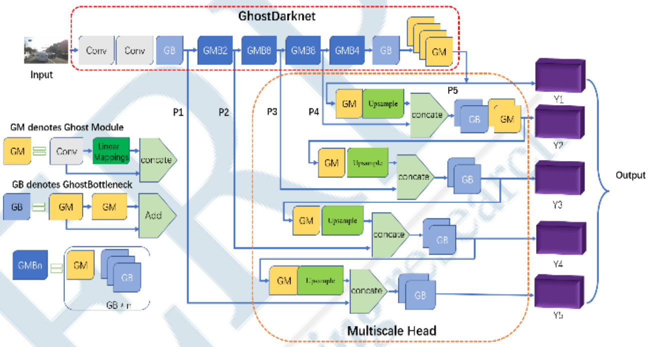

Figure 1: YOLOv3 Architecture [18]

B. YOLOv4:

You Only Look Once version 4, or YOLOv4, is a real-time object detection algorithm. It is a more advanced version of YOLO that offers superior performance and a more complex architecture. Figure 2 illustrates the architecture of YOLOv4, which is described in detail below.

- Backbone: The architecture of YOLOv4 starts with a backbone network that processes the input image and extracts features at various scales. Typically, this backbone network is a deep convolutional neural network such as Darknet-53, which consists of 53 convolutional layers.
- Neck: In order to extract features that are best for object detection, the YOLOv4 model uses a neck component that takes the output from the backbone network and processes it. This component is made up of many modules, including path aggregation network (PAN) and spatial pyramid pooling (SPP), which allow to capture features at various scales.
- Head: The head's main responsibility is to predict the object classes and bounding boxes present in the input image. It consists of a fully connected layer that generates a set of bounding boxes, where each box corresponds to an object in the image. The model associates a confidence score with each bounding box, which reflects its confidence in detecting an object in that location.
- Output: YOLOv4 produces a collection of bounding boxes along with corresponding confidence scores

IFERP
connecting engineers... developing research
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 7, July 2023

and probabilities for each class. The confidence scores indicate the model's level of certainty in the accuracy of each bounding box, while the class probabilities express the likelihood of the object in the box belonging to a specific class, such as "car," "person," or "traffic light.

The YOLOv4 architecture also includes several other components, including:

- Cross-stage partial connections (CSP): A technique that connects the backbone network and neck to improve feature extraction.
- Mish activation function: An activation function that has been shown to improve model performance compared to other activation functions.
- Spatial attention modules: Modules that assist in directing the model's attention towards significant characteristics within the input image.

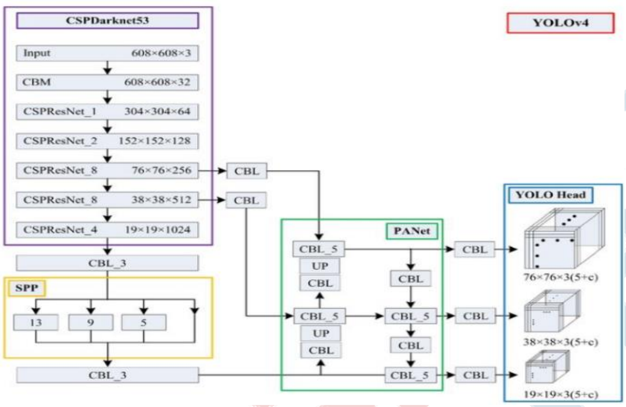

Figure 2: YOLOv4 Architecture [19]

C. YOLOv5:

YOLOv5 is a object detection framework that follows the You Only Look Once (YOLO) approach. This deep learning-based system is specifically designed to quickly and accurately detect objects in real time, making it ideal for a wide range of applications.

Figure 3 illustrates the architecture of YOLOv5, while the key components of the framework are discussed below:

- Backbone Network: The YOLOv5 model utilizes a backbone network, such as ResNet, to extract features from the input images in order to produce feature maps that the rest of the network uses for object detection.
- Neck: The YOLOv5 neck module is responsible for gathering the feature maps from the backbone network and transferring them to the head. It consists of several convolutional layers and aims to decrease a measure of the spatial resolution of feature maps and simultaneously enhance the channel depth.
- Head: The head's primary function is to predict the existence and characteristics of objects in the image. The model employs anchor boxes, Intersection over Union (IoU), and non-maximum suppression (NMS) to achieve this goal.
- Output Layer: The YOLOv5 output layer makes the final predictions for the input image. For each detected item, this layer outputs a number of bounding boxes together with the probabilities for the class of each box. This results in a collection of bounding boxes, class probabilities, and confidence ratings for each object in the image.

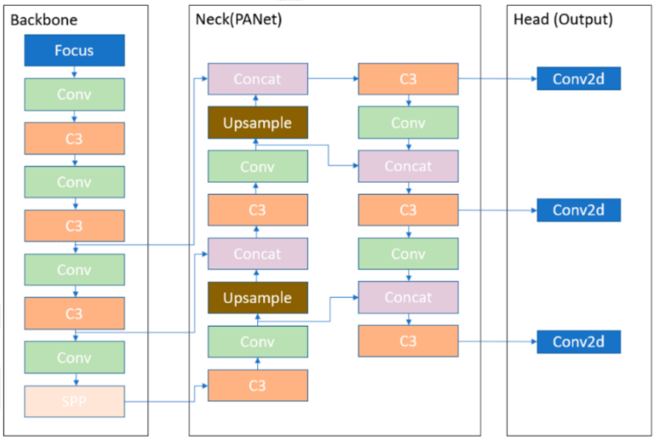

Figure 3: YOLOv5 Architecture [20]

IV. METHODOLOGY

The aim of this research to evaluate the performance of an object detection model for detecting vehicles and licence plates from pre-captured traffic frames, as well as to extract license plate numbers from the detected plates using the EasyOCR tool.

The flowchart presented in Fig. 4 outlines the sequence of steps taken to evaluate the object detection model for automatic license plate and vehicle detection.

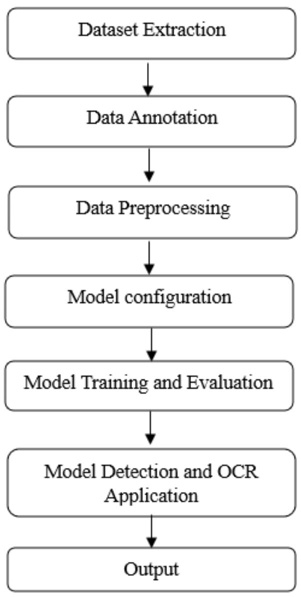

Figure 4: Flow of Implementation

IFERP connecting engineers... developing research
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 7, July 2023

A. Dataset Extraction:

To train the YOLO model for detecting vehicles and license plates, we utilized public dataset images from the Google OpenImage dataset. This dataset is known for its object detection, classification, and visual relationship recognition capabilities.

Using the OIDv4 toolkit, we were able to extract a total of 2320 vehicles with license plate images on the road in .jpg format.

B. Data Annotation:

Data annotation is an essential part of YOLO model training. It involves labeling objects present in images using bounding boxes that specify the object's position and size in the image. We have labeled the vehicles and license plates using this method. Class labels are assigned to these bounding boxes to identify the type of object present in them. We chose to annotate our datasets in COCO format using the makesense.ai tool, which is compatible with the YOLO format.

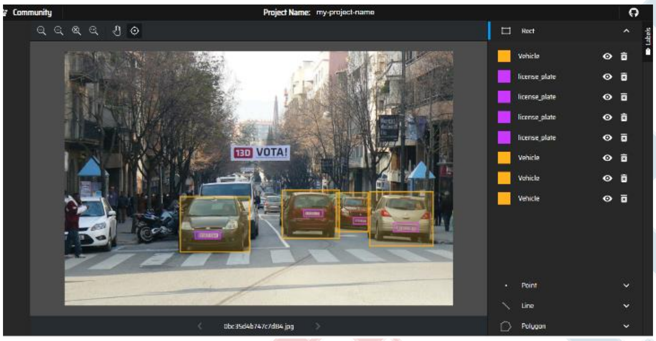

Figure 5: Data Annotation

C. Data Preprocessing:

After data has been annotated, it is typically split into two sets: a training set and a testing set. The training set is used to train the YOLO model, while the testing set is used to evaluate the model's performance on new, unobserved data.

Our dataset statistics are presented in detail in Table I, and it has been split into a 75:25 ratio for training and validation. The dataset consists of 2320 files, with 1740 files used for training and 580 files for testing.

Table I: Dataset Statistics

Classes	Car	Two-wheeler	Other (Bus, Truck, Van)	Total
Train	598	450	692	1740
Validation	200	150	230	580
total	798	600	922	2320

D. Model Configuration:

In this experiment, we used three object detection models: YOLOv3, YOLOv4, and YOLOv5. Before training these models, we configured the network based on our dataset. In the YOLOv5 model, all we need to change is the number of classes of custom datasets. In our research, we changed the class count to 2 as we have two classes to detect i.e., vehicle and license plate.

To train the YOLOv3 and YOLOv4 models we changed batch_size to 64, sub-division to 16, height and width to 416×416 pixels, and MaxBatchsize to 6000 as it is necessary to have minimum MaxBatchsize as 6000 if several classes are less than 3 else MaxBatchsize can be computed as equation (1), steps in range 4800 to 5400 as mentioned in equation(2), Classes to 2 in all yolo section and Filter to 28 in convolution layer preceding yolo layers which is computed using equation (3)

$$\text{MaxBatchsize} = \text{Total No. of Classes} \times 2000 \quad (1)$$

$$\text{Steps} = (80\% \text{ of Max\_BatchSize}), (90\% \text{ of Max\_BatchSize}) \quad (2)$$

$$\text{Filter} = (\text{Number of Classes} + 5) \times 3 \quad (3)$$

E. Model Training and Evaluation:

After data preprocessing and configuring the model, the annotated dataset is utilized to train the YOLO model, including YOLOv3, YOLOv4, and YOLOv5, for detecting objects of interest in new images or videos that have not been previously seen. During the training phase, the model analyses the patterns and features in the training data to recognize vehicles and license plates within an image.

After training, the model is assessed using a variety of metrics on the distinct validation dataset, such as precision, recall, and mAP50 i.e., mean average precision at an IoU threshold of 0.5. The performance measures for each class are shown in Fig. 6, Fig. 7, and Fig. 8, which show how the YOLOv3, YOLOv4, and YOLOv5 models performed for each class.

```
 detections_count = 882, unique_truth_count = 309
class_id = 0, name = Vehicle, ap = 83.49%        (TP = 101, FP = 47)
class_id = 1, name = license_plates, ap = 81.21%         (TP = 143, FP = 39)

 for conf_thresh = 0.25, precision = 0.74, recall = 0.79, F1-score = 0.76
 for conf_thresh = 0.25, TP = 244, FP = 86, FN = 65, average IoU = 58.25 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.823504, or 82.35 %
Total Detection Time: 3 Seconds

Set -points flag:
 `-points 101` for MS COCO
 `-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
 `-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```

Figure 6: YOLOv3 Performance Metrics

IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 7, July 2023

```
detections_count = 962, unique_truth_count = 309
class_id = 0, name = Vehicle, ap = 76.50%        (TP = 105, FP = 34)
class_id = 1, name = license_plate, ap = 84.73%        (TP = 155, FP = 54)

 for conf_thresh = 0.25, precision = 0.79, recall = 0.84, F1-score = 0.81
 for conf_thresh = 0.25, TP = 250, FP = 88, FN = 44, average IoU = 63.37 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.853159, or 85.32 %
Total Detection Time: 84 Seconds

Set -points flag:
 `-points 101` for MS COCO
 `-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
 `-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```

Figure 7: YOLOv4 Performance Metrics

```
Validating runs/train/yolov5m_results_/weights/best.pt...
Fusing layers...
custom_YOLOv5m summary: 182 layers, 7249215 parameters, 0 gradients
                 Class     Images  Instances          P          R      mAP50
                   all        104        309      0.899      0.804      0.884
               Vehicle        104        117      0.916      0.829      0.887
         license_plate        104        192      0.882      0.779       0.88
Results saved to runs/train/yolov5m_results_
CPU times: user 49 s, sys: 5.79 s, total: 54.8 s
Wall time: 1h 19min 45s
```

Figure 8: YOLOv5 Performance Metrics

F. Model Detection and OCR Application:

The methodology proposed in this paper detects licence plates using the YOLO model, followed by character recognition using EasyOCR.

EasyOCR is an open-source optical character recognition tool designed for extracting text from images. The OCR technology enables machines to recognize and interpret printed or handwritten text characters from an image or document. It employs deep learning algorithms for efficient and accurate character recognition, which makes it a suitable tool for several applications, including license plate recognition, image-to-text conversion, and document scanning

G. Output:

Figure 9 shows the output of model detection while Fig. 10 shows the output of EasyOCR.

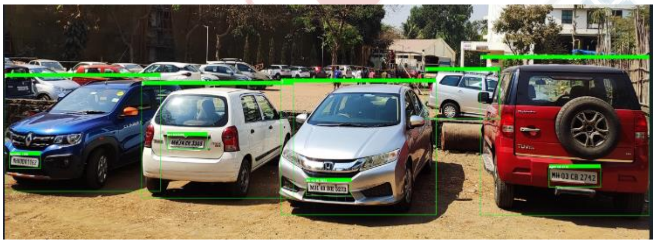

Figure 9: Detection Model Output

```
"[INFO] Extracting BBox coordinates. . . \n",
"[([[11, 8], [176, 8], [176, 64], [11, 64]], \"'WHO3DA 1162\", 0.23448404458579625)]\n",
"'WHO3DA 1162\n",
"[INFO] Extracting BBox coordinates. . . \n",
"[([[8, 16], [299, 16], [299, 88], [8, 88]], 'M#03 CB 2712', 0.4184527525818701)]\n",
"M#03 CB 2712\n",
"[INFO] Extracting BBox coordinates. . . \n",
"[([[20, 16], [226, 16], [226, 68], [20, 68]], 'Mh14Ck3305', 0.240687115878984)]\n",
"MH14CK3305\n",
"[INFO] Extracting BBox coordinates. . . \n",
"[([[20, 18], [254, 18], [254, 66], [20, 66]], 'MH 43 BE 5273', 0.4954056129380337)]\n",
"MH 43 BE 5273\n",
"[INFO] Exiting. . . \n"
```

Figure 10: EasyOCR Output

V. RESULTS

Table II: Comparative Analysis of YOLOv3, YOLOv4 and YOLOv5 Detection Model

Model	Precision	Recall	mAP50
YOLOv3	0.74	0.79	0.82
YOLOv4	0.79	**0.82**	0.85
YOLOv5	**0.88**	0.80	**0.88**

The Table II. shows the study of YOLOv3, YOLOv4, and YOLOv5 model in comparison for all classes. From the table, we observe that the YOLOv5 model gives better results with a precision of 0.88, and mAP@50 0.88 as compared to YOLOv3 and YOLOv4 while YOLOv4 gives slightly better recall of 0.82.

The YOLOv5 model features enhanced training methods and a more flexible and modular architecture compared to its predecessors, YOLOv3 and YOLOv4. This results in increased computational efficiency and faster processing times, while also improving accuracy and reducing the risk of overfitting. The modular design of YOLOv5 enables customization and adaptation to meet specific object detection requirements, making it a versatile and flexible solution for various applications.

VI. CONCLUSION

In conclusion, our study demonstrated the efficiency of deep learning techniques for the automatic detection and extraction of licence plates. Our research indicates that in terms of accuracy, YOLOv5 model outperformed both YOLOv3 and YOLOv4, making it the best option for this application.

However, it is important to note that the performance of the EasyOCR model utilized for license plate extraction was not consistently accurate when applied to the area of interest.

Our findings indicate that methods using deep learning for automatic license plate detection and extraction have the potential to significantly enhance traffic management and public safety. By improving the accuracy and speed of license plate detection, we would able to effectively monitor traffic patterns and enforce traffic laws, leading to safer, more efficient roadways.

REFERENCES

[1] Batra Piyush, et al. "A novel memory and time-efficient ALPR system based on YOLOv5." Sensors 22.14: 5283, June 2022.

[2] Park S. H., Yu S. B., Kim J. A., and Yoon H., "An all-in-one vehicle type and license plate recognition system using YOLOv4," *Sensors* 22.3: 921, Jan. 2022.

[3] Sung Ju-Yeong, and Saet-Byeol Yu., "Real-time automatic license plate recognition system using YOLOv4," IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia). IEEE, June 2020.

[4] Pandey, Radha, and Aruna Malik, "Object detection and movement prediction for autonomous vehicle: a review," 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC). IEEE, May 2021.

[5] Henry, Chris, Sung Yoon Ahn, and Sang-Woong Lee. "Multinational license plate recognition using generalized character sequence detection." IEEE Access 8 (2020): 35185-35199, Feb. 2020.

[6] Silva Sergio M., and Cláudio Rosito Jung, "A flexible approach for automatic license plate recognition in unconstrained scenarios." IEEE Transactions on Intelligent Transportation Systems 23.6: 5693-5703, Feb. 2021.

[7] N.Palanivel Ap., T. Vigneshwaran. M. Sriv Arappradhan and R. Madhanraj, "Automatic number plate detection in vehicles using faster R-CNN," International conference on system, computation, automation, and networking (ICSCAN). IEEE, July 2020.

[8] Amonkar, Saidatt, Anikumar Naik, and Amogh A. Sanzgiri, "ALPR System Using Sequence Modelling: A real-time system for vehicle authentication," International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE, Nov. 2019.

[9] Aung Khin Pa, Khin Htar Nwe, and Atsuo Yoshitaka, "Automatic license plate detection system for Myanmar vehicle license plates," 2019 International Conference on Advanced Information Technologies (ICAIT). IEEE, Nov. 2019.

[10] Laroca, Rayson, et al. "A robust real-time automatic license plate recognition based on the YOLO detector," 2018 international joint conference on neural networks (ijcnn). IEEE, July 2018.

[11] Parneet Kaur, Yogesh Kumar, Shakeel Ahmed, Abdulaziz Alhumam, Ruchi Singla and Muhammad Fazal Ijaz, "Automatic license plate recognition system for vehicles using a cnn," Computers, Materials & Continua 71.1:35-50, April 2021.

[12] A. Tonge, S. Chandak, R. Khiste, U. Khan and L. A. Bewoor, "Traffic Rules Violation Detection using Deep Learning," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 1250-1257, Nov. 2020.

[13] S.R. Ibadov, B.Y. Kalmykov, R.R. Ibadov, and R.A. Sizyakin, "Method of Automated Detection of Traffic Violation with a Convolutional Neural Network," EPJ Web Conf. IV International Conference-Modeling of Nonlinear Processes and Systems, Volume 224: 04004, Dec. 2019.

[14] Sriram Narayana Cummaragunta, Srinandan K S and Jyoti Shetty “Wrong Side Driving Detection,” International Journal of Technology and Emerging Sciences (IJTES), July 2022.

[15] D. X. Ascar, D. Judson and S. K. Pittala, "SegNet Approach for Vehicle License Plate Localization," 2020 Seventh International Conference on Information Technology Trends (ITT), Abu Dhabi, United Arab Emirates, pp. 113-117, Nov. 2020.

[16] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," in IEEE Access, vol. 9, pp. 11203-11225, Dec. 2020.

[17] Yousri Kessentini, Mohamed Dhia Besbes, Sourour Ammar and Achraf Chabbouh, "A two-stage deep neural network for multi-norm license plate detection and recognition," Expert Systems with Applications, Volume 136, Pages 159-170, ISSN 0957-4174, Dec. 2019.

[18] Huanqia Cai, Lele Xu, and Lili Guo, “Enhanced Efficient YOLOv3-tiny for Object,” CSAI '21: Proceedings of the 2021 5th International Conference on Computer Science and Artificial Intelligence, Pages 14–20, Dec. 2021.

[19] Wang, Shengying, et al. "A real-time deep learning forest fire monitoring algorithm based on an improved Pruned+ KD model," Journal of Real-Time Image Processing 18.6 (2021): 2319-2329, May 2021.

[20] Nepal Upesh, and Hossein Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," Sensors 22.2 (2022): 464, Jan. 2022.

[21] Ravirathinam, Praveen, and Arihant Patawari. "Automatic license plate recognition for Indian roads using faster-RCNN." 2019 11th international conference on advanced computing (ICoAC), IEEE, Dec. 2019.

[22] Saif Nazmus, et al. "Automatic license plate recognition system for Bangla license plates using convolutional neural network," TENCON 2019-2019 IEEE Region 10 Conference (TENCON), Oct. 2019.

[23] Saha Satadal, "A review on automatic license plate recognition system," arXiv preprint arXiv:1902.09385, Feb. 2019.

[24] Izidio, Diogo MF, et al. "An embedded automatic license plate recognition system using deep learning," Design Automation for Embedded Systems 24.1 (2020): 23-43, Nov 2019.

[25] Chen Rung-Ching, "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning," Image and Vision Computing 87 (2019): 47-56.

[26] Riaz, Waqar, et al. "YOLO-based recognition method for automatic license plate recognition." 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Aug. 2020.