Automatic Number Plate Recognition Techniques Performance on Zimbabwean Number Plates

Dzinaishe Mpini
Department of Computer Science
Midlands State University
Gweru, Zimbabwe
mpinid@staff.msu.ac.zw

Maxmillan Giyane
Department of Computer Science
Midlands State University
Gweru, Zimbabwe
giyanem@staff.msu.ac.zw

2023 2nd Zimbabwe Conference of Information and Communication Technologies (ZCICT) | 979-8-3503-1812-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ZCICT59466.2023.10528439

Abstract— **In this modern age, Automatic Number Plate Recognition systems are an important aspect to improve service efficiency (toll collection, traffic management, parking management), security (vehicle tracking, monitoring vehicle visits) and law enforcement when dealing with motor vehicles. However, many of the existing Automatic Number Plate Recognition systems were developed for countries with number plate standards and font that differ from the Zimbabwean number plates. This research compares different approaches to Automatic Number Plate Recognition techniques with a special focus on Zimbabwean number plates through experiments. The experiments were grouped into two main phases of Automatic Number Plate Recognition namely plate detection and plate recognition. The experiments for plate detection compared traditional image processing techniques using OpenCV and deep learning techniques using the YOLOv8 algorithm. The YOLOv8 algorithm proved to be the most accurate approach for plate detection. Experiments for plate recognition compared Optical Character Recognition engines and machine learning approaches. The compared machine learning algorithms were random forest classifier, and K-Nearest Neighbour. In this set of experiments, a K-Nearest Neighbour algorithm proved to be quicker and more accurate in detecting Zimbabwean number plates. This research recommends the development of an end-to-end system for Zimbabwean number plates recognition using deep learning.**

Keywords—Automatic Number Plate Recognition, Zimbabwean number plates, number plate detection, optical character recognition

I. Introduction

Automatic Number Plate Recognition (ANPR) is the ability of computer systems to detect vehicle number plates without human intervention [1]. In this modern age, ANPR systems are an important driver to more efficient, quicker and secure systems. Through ANPR, service time and efficiency can be improved in toll fees collection [2], traffic or congestion management [1,2] and parking management [2]. ANPR helps with vehicle tracking [3], monitoring vehicle visits [1] hence improving security of vehicles and premises. Finally, ANPR can be used to improve traffic laws enforcements through quick identification of offenders [4,5]. A lot of organisations can benefit from the use of ANPR systems. These organisations include the traffic unit of the police department, toll collection companies, parking management companies, security companies as well as several companies wishing to manage parking at their premises or to improve security at their premises. Despite the numerous benefits of ANPR, no ANPR research has been done in Zimbabwe. This research compares different approaches to plate detection and recognition in a bid to identify the best methods to employ for ANPR on Zimbabwean number plates. Generally, there are three generic types of number plates in Zimbabwe with the following distinct differences: black lettered yellow plates for private use vehicles, red lettered white plates for public service vehicles and black lettered white plates for Government vehicles. The characters are in a single line for rectangular number plates and two lines for square number plates. The main focus of this research will be on black lettered yellow number plates (both rectangular and square) for private use vehicles.

II. Research Objectives

The aim of the study is to compare ANPR techniques performance on Zimbabwean number plates. To achieve the main goal, the following are specific objectives of the study:

1. To determine a high-performance plate detection / plate localisation approach in detecting Zimbabwean number plates.
2. To investigate on the most suitable character recognition approach for Zimbabwean number plates.
3. To recommend the suitable methods that can be used in number plate detection and text recognition for developing a Zimbabwean vehicles ANPR platform.

III. Related Work

ANPR encompasses two main phases namely plate area detection and character recognition. Plate detection also known as plate localisation refers to identifying the license plate area from a given image or video stream whilst character recognition deals with identifying the actual characters or text on the plate area [6]. This review considered studies which have both plate detection and plate recognition steps.

A. Plate Detection Approaches

Traditional image processing is the most common approach to plate area detection and is executed using OpenCV; an open-source computer vision library developed by Intel [8]. Several recent studies and experiments [7, 9, 10, 11, 12, 13, 14, 15, 16] make use of image processing techniques present in OpenCV. It is used to detect features in an image which includes edge detection, mathematical morphology and contour detection. Other studies on the other hand incorporate Deep Learning (DL) into plate detection. DL is a type of Machine Learning (ML) which is based on Artificial Neural Networks (ANN) where multiple layers of processing are used to extract features from the provided data [17]. It is an efficient approach to plate detection as plate detection itself heavily relies on feature extraction. The majority of studies make use of the You Only Learn Once

(YOLO) algorithm for plate detection using deep learning. YOLO is a popular deep learning and efficient object detection algorithm which makes use of Convolutional Neural Network (CNN) to detect and recognise objects [18]. Versions of YOLO used for plate detection in previous studies include YOLOv1 [19, 20], YOLOv2 [21], YOLOv3 [22, 23], YOLOv4 [3, 19, 24, 25] and YOLOv5 [26]. Some techniques that can be used for plate detection compare Tensorflow and ImageAI [3].

B. Plate Recognition Approaches

For plate recognition, Optical Character Recognition (OCR) engines are commonly used. OCR is a procedure used to transform an image of text into a machine-readable text format [27]. With OCR a computer scans an image and decodes text written on the image. Tesseract and SimpleOCR are two OCR engines employed in number plate text recognition [2, 3, 22, 24, 27, 28, 29, 30, 31]. ML is another common approach to character recognition from number plates. ML refers to the usage and creation of computer systems that can adapt and learn without being given explicit instructions by analysing data patterns and making inferences using statistical models and algorithms [32]. It can suitably be applied in character recognition to detect letters and numbers in a number plate. The most common ML algorithms used for character recognition in number plates are Support Vector Machine (SVM) and Random Forest classifier [32]. Deep Learning can also be applied in recognition of characters on a number plate and not just detecting a number plate area on a supplied image. K-Nearest Neighbour (K-NN) [1, 7, 11, 13] and Regions with Convolutional Neural Networks (R-CNN) [12] are the most common algorithms applied in number plate recognition [2]. Both K-NN and R-CNN are CNN algorithms. Table I and Table II present findings from ANPR studies carried out in different countries.

TABLE I. SUMMARY OF PLATE DETECTION APPROACHES

Country	Detection Method	Plate Detection Results
India	OpenCv	[13] – 97.8% accuracy
Indonesia	OpenCv	[7, 31] - unspecified accuracy.
India	OpenCv	[5, 9, 10, 11, 14, 34] unspecified accuracy
Tunisia	OpenCV	[12] – unspecified accuracy.
India	YOLOv1	[3] compared YOLOv1and ImageAI.YOLOv1 attained 97% accuracy.
India	YOLOv1	[19] - 98% accuracy.
Indonesia	YOLOv1	[20] - 98.2% accuracy.
United Kingdom	YOLOv2	[21] - 90.3% Accuracy
Iran	YOLOv3	[22] – unspecified accuracy.
Malaysia	YOLOv3	[23] – 99% accuracy.
-	YOLOv4	[25] – unspecified accuracy.
India	YOLOv4	[36] – unspecified accuracy.
India	YOLOv4 (CSPDarknet53)	[24] – 98.33% accuracy.
India	YOLOv5	[26] – 90% accuracy.
Malaysia	CNN	[4] – 0.635s
India	CNN (VGG16)	[30] – unspecified accuracy.
Iraq	Single Shot Detector	[35] – 98% accuracy.
India	Tensorflow	[29] – unspecified accuracy.

TABLE II. SUMMARY OF CHARACTER RECOGNITION APPROACHES

Country	Recognition Method	Character Recognition Results
India	PyTesseract	[10] – 94.3% accuracy
India	Google Tesseract	[26] – 93% accuracy.
India	EasyOCR	[29] unspecified accuracy
India	EasyOCR	[24] – 95% accuracy.
India	EasyOCR	[29, 30] – unspecified accuracy.
India	Tesseract	[36] - unspecified accuracy.
-	Tesseract	[25] – unspecified accuracy.
Iran	Unspecified OCR engine	[22] – 95,05% Accuracy
India	Unspecified OCR engine	[14, 27] – unspecified accuracy.
-	RF Classifier	[32] – 90.9% accuracy.
Indonesia	KNN	[7] – 92.86 accuracy.
India	KNN	[1] – 87.2% (clear plates), 46.87% (blurred plates), 51.64% (skewed plates), 73.02% (average plates) accuracy.
India	KNN	[13] – 98% accuracy.
Iraq	KNN	[35] – 96% accuracy.
India	KNN	[11] – unspecified accuracy.
Tunisia	CNN	[12] – 95.84% accuracy.
India	CNN	[34] – 98.13% accuracy.
Indonesia	CNN	[31] – 96.23 accuracy.
India	CNN	[2, 5] - unspecified accuracy.
India	Template Matching	[9] – 67.98% accuracy.

From the surveyed literature, it was noted that for plate detection OpenCV is the commonly used basic approach to image processing and YOLO is the commonly used DL approach. OCR engines are the most commonly used basic approaches to text classification, RF classifiers are commonly used ML approaches and K-NN is the commonly used DL approach for plate character recognition. All the studies reviewed did not compare different approaches to plate detection and plate recognition. In the cases where comparisons were made [3], only two approaches would be compared to each other and it was only a comparison of accuracy of the approach and not the time it took or computational power to yield the results. Secondly, there is no end to end platform or service developed for use in various sectors which make use of number plate recognition. Experiments were carried out with satisfying accuracy scores but no end to end platform which can be used in different fields to address challenges with plate recognition in high volumes and density such as toll gates, traffic congestion and parking spaces. Lastly, no ANPR studies have been carried out in the Zimbabwean context. Therefore, this study will compare ANPR techniques performance on Zimbabwean number plates.

IV. METHODOLOGY

Experiments were used to compare the plate detection and character recognition approaches. A total of eighty-five Zimbabwean number plates were used in this research study. The research concentrated on black lettered yellow plates for private use vehicles due to their raising numbers in Zimbabwe and narrowing the scope of the research. The dataset preparation process as well as outlining the experiments carried out in a bid to make the comparisons are explained in sections that follow.

A. *Dataset Preparation*

For successful experiments, a dataset with Zimbabwean number plates was required. A total of eighty-five images showing Zimbabwean number plates were obtained. The eighty-five images included number plates already mounted on vehicles from the front, rear and at different angles. Images of number plates with characters in a single line and in two lines were obtained. The number plate images went through a pre-processing stage where all images were cropped to remove backgrounds which are not bodies of the motor vehicle. In addition, the images were resized to a size of 640*640px. The dataset was labelled using the RoboFlow platform which is an online service focused on helping to build and deploy computer vision experiments and products. Of the eighty-five images in the dataset; fifty-nine images (69.4%) were used for model trainings, eleven images (12.9%) for model validations in cases where model validations were important and nine images (10.6%) were left for models testing.

B. *Experiments*

Five experiments which fall under two main categories of ANPR namely plate detection (two experiments) and plate character recognition (three experiments) were conducted. The experiments were conducted to determine the most efficient and accurate method to employ for the Zimbabwean number plate detection and recognition.

1. *Plate Detection Experiments*

To detect number plates, OpenCV and YOLOv8 approaches were used. The output of the approach with the efficient and accurate results was used in plate character recognition experiments.

i. *Experiment 1 - Plate area detection using traditional image processing techniques*

In this research, the first experiment in plate detection was carried out using OpenCV in Python programming language. The experimental procedure for plate detection using OpenCV was as follows: **1.** Resize image from 640*640px to 300*300px, **2.** Gray scale image using cvtColor method, **3.** Reduce image noise using bilateral filter **4.** Find contours on image using contours method, **5.** Find edges on image using Canny function, **6.** Determine plate area through identifying contours with a length around 70px and with four sides joined together and **7.** Crop plate area by extracting the bounding points from the plate area found in step 6.

ii. *Experiment 2 - Plate area detection using deep learning*

The experiment for plate area detection using deep learning were conducted using YOLOv8 model. The steps carried out in this experiment were: **1.** Model training, **2.** Model validation and **3.** Model testing.

2. *Plate Recognition Experiments*

In order to perform character recognition, images should be detected first then recognition follows. Character recognition experiments were conducted through PyTesseract, Random forest and K-Nearest Neighbour (K-NN).

i. *Experiment 3 - Character recognition using an Optical Character Recognition (OCR) engine*

The character recognition experiment for this research was carried out using PyTesseract. An image showing a cropped number plate area was subjected to the following few lines of Python code to recognise and output the characters on the image.

```
import pytesseract
pytesseract.pytesseract.tesseract_cmd = '/opt/homebrew/Cellar/tesseract/5.3.1_1/bin/tesseract'
def recognise(image):
    plate = pytesseract.image_to_string(image, lang='eng')
    print("Detected license plate number: ", plate)
```

ii. *Experiment 4 - Character recognition using ML*

In this research, the experiment was carried out using Random Forest classifier. The steps carried out in this experiment were: **1.** Model training, **2.** Model validation and **3.** Model testing.

iii. *Experiment 5 - Character recognition using DL*

K-NN was used in this research for its simplicity and less demand for computing resources. The steps carried out in this experiment were: **1.** Model training, **2.** Model validation and **3.** Model testing.

V. RESULTS AND DISCUSSION

The performance of different approaches to Zimbabwean number plates detection and character recognition through five experiments was investigated. This section presents the results of the experiment carried out and further presents a discussion of results from the experiments.

A. *Experiments Results*

1. *Experiment 1 Results - Plate detection using OpenCV*

77.78% of the images were detected successfully. The average time taken to localise a number on an image was 12.36s. Figure 1 shows a successful plate area detection run. The cropped image shows the detected number plate area.

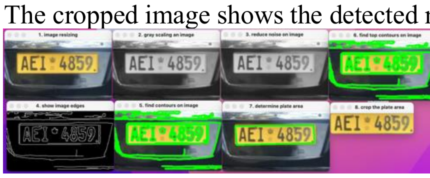

Fig. 1. Image with few edges and contours

Figure 2 shows an unsuccessful plate area detection run. No number plate was shown in the cropped number plate area.

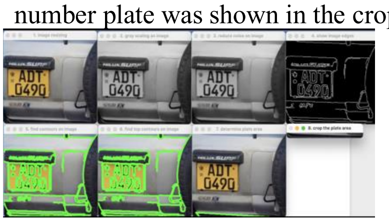

Fig. 2. Image with more edges and contours

Plate detection using OpenCV proved to be more efficient where there are few edges and contours on the

IEEE 979-8-3503-1812-8/23/$31.00 ©2023 IEEE

image. The approach mainly relies on edges and contours when detecting images. The back of most registered Zimbabwean vehicles has red reflectors and these were usually mistaken for the license plate area by this approach. Therefore, the approach yields no results if the image has more edges and contours. Most reviewed studies [5, 7, 10, 12, 14] used OpenCV and [10] quantified the findings. The approach was seen to be 92% in detecting Indian number plates. The variation of results might be due to differences in fonts and plate standards.

2. *Experiment 2 Results - Plate detection using YOLOv8*

100% of the images were detected successfully using YOLOv8 model. It took an average of 0.10s to detect an image successfully. Figure 3 shows a sample of the training batch and Figure 4 shows a sample of a detected plate area using the same model.

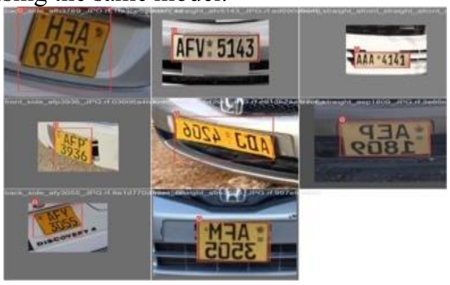

Fig. 3. YOLOv8 sample training batch

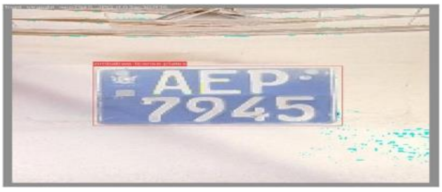

Fig. 4. YOLOv8 sample of detected plate area

Yolov8 model proved to be very accurate and took a fraction of a second to give the output. The model managed to distinguish between a number plate and the area surrounding the plate. The results corroborated with [19, 20, 21, 23, 24, 26], though different versions of YOLO were used in the aforementioned studies. Therefore, in general YOLO model is reliable in detecting Zimbabwean number plates regardless of the standards, fonts, and density of both edges and contours.

3. *Experiment 3 Results - Plate recognition using PyTesseract*

PyTesseract managed to accurately recognise number plate text from 55.56% (both square and rectangular number plates) of the testing sample provided. The approach took an average time of 1.50s to give output. Character recognition on rectangular number plates was 100% and 42% for square number plates. Figure 5 and Figure 6 shows successful and unsuccessful plate recognition experiment results respectively.

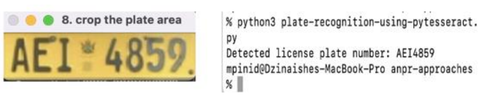

Fig. 5. Successful plate recognition experiment using PyTesseract

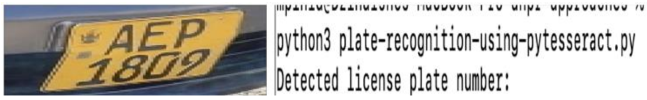

Fig. 6. Unsuccessful plate recognition experiment using PyTesseract

It was noted that PyTesseract works efficiently for rectangular number plates. In addition, the approach works efficiently when the number plate font is clear. For unclear and square plates, the approach fails to recognise some number plate characters. The results for rectangular plates character recognition corroborated with [10, 26], but for square number plates the results varied from the aforementioned studies which were done in India. The studies conducted used rectangular number plates.

4. *Experiment 4 Results - Plate recognition using K-NN*

K-NN model managed to accurately recognise 66.67% of the subjected number plate text images. The average time taken to give the output was 0.19s per image. The individual text on a number plate were recognised in no particular order. Figure 7 shows K-NN number plate recognition experiment results.

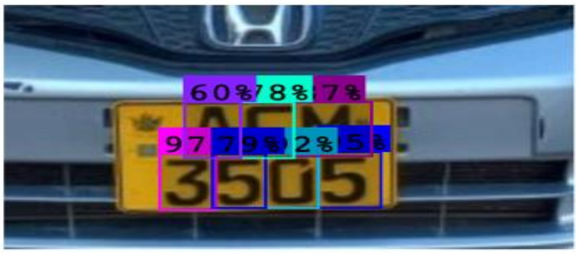

Fig. 7. Plate recognition experiment using K-NN

The maximum percentage accuracy from the classification report on an individual character was 97% whilst the minimum was 60%. K-NN model identifies individual characters on the number plate area. Results of the study were supported by [1, 7, 13, 35]. The studies presented the accuracy from the K-NN model classification report. In this study 66.67% were the actual number plates with characters that were correctly recognised. Since the K-NN model works through recognising individual characters on a plate area, it failed to classify some of the single characters. This resulted in 33.33% of the tested number plates to have output with missing characters.

5. *Experiment 5 Results - Plate recognition using RF Classifier*

Image recognition using RF classifier yielded a 0% accuracy for untrained images. An average time of 3.23s was taken to produce an output. 45% of the trained images were recognised accurately. It took an average of 1.34s per image to give the output. The algorithm managed to identify the number plates but could not recognise the text as shown in Figure 8.

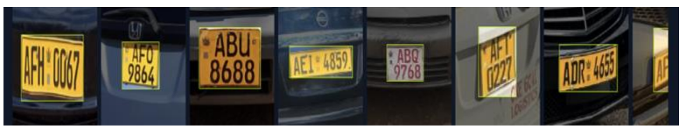

Fig. 8. Plate recognition using Random Forest classifier results

The trained RF classifier model worked through attempting to classify the supplied image into an already known image of a number plate as a whole. Results indicated that the model could not identify a number plate not included in a trained dataset, and it cannot read individual letters from a plate but rather the whole number plate. The results varied from [32] and this is because the dataset used for testing had number plates that differ from the ones used for training, yet the model could only identify already trained number plate areas and not individual characters on a plate area. This approach is not efficient since it requires training all number plates that should be identified in the future.

B. ANPR techniques performance comparison

There are several techniques that can be used for both plate detection and plate recognition. For plate detection, the performance of OpenCV and YOLOv8 approaches on Zimbabwean number plates was compared. PyTesseract, K-NN and RF classifier plate recognition approaches' performance on Zimbabwean number plates was also compared. Table III and Table IV give a comparison of the ANPR approaches.

TABLE III. COMPARISON OF PLATE DETECTION APPROACHES

Approach	OpenCV	YOLOv8
Accuracy	77.78%	100%
Average Time	12.36s	0.10s
Strengths	• No training is required. • It has a large community support base.	• Once model has been trained, detection is very quick. • It can be easily trained.
Weaknesses	• Poor accuracy for images with many contours and edges. • Takes more time to process a single image. • Requires more code to fine tune the detection.	• Model should be trained first and training takes time. • Requires high performance computing platforms.

TABLE IV. COMPARISON OF PLATE RECOGNITION APPROACHES

Approach	PyTesseract	K-NN	RF Classifier
Accuracy	• 55.56% for rectangular and square number plates. • 42.86% for square number plates. • 100% for rectangular number plates.	66.67%	• 0% for untrained dataset. • 45% for trained dataset.
Average Time	1.50s	0.19s	• 3.23s for untrained dataset. • 1.34s for trained dataset.
Strengths	• Few lines of code are required since it uses an existing library.	• Accurate in identifying individual text characters in an image. • Fast in giving the recognised characters.	-
Weaknesses	• Characters should be in a single line for optimal results. • The font on the image should be clear. • Slow in detecting characters.	• Identifies characters in no particular order, hence there is need to reorder the identified characters.	• Cannot identify individual characters, hence unable to recognise new and untrained images. • Takes time to give output.

VI. CONCLUSION AND RECOMMENDATIONS

To detect Zimbabwean number plates, a YOLOv8 model yielded 100% plate localisation accuracy which was achieved in an average of 0.10s per image. Deep learning, using K-NN character recognition technique produced an accuracy score of 66.67% which was achieved on an average of 0.17s per image. It is evident from Table 2 that YOLOv8 is the best approach in plate detection and K-NN is the most suitable character recognition approach. It is recommended to use YOLOv8 and K-NN for ANPR of Zimbabwean number plates. Even though the research proved Zimbabwean number plates can also join the ANPR bandwagon, it did not address how the different approaches would perform under different weather condition such as misty or rainy conditions. It is recommended to train YOLOv8 model with a large dataset including images subjected to different weather conditions and images with poor quality. As a future recommendation, the development of an Application Programming Interface (API) service or end to end system with records from the Central Vehicle Records (CVR) Department in Zimbabwe; for use in cases where the vehicle owner should be quickly identified will be beneficial. In addition, making use of newer approaches to ANPR such as You Only Learn One Representation (YOLOR) which are more efficient despite the picture quality might yield better results.

REFERENCES

[1] M. Arora, A. Jain, S. Rustagi, and T. Yadav, "Automatic number plate recognition system using optical character recognition," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 986–992, 2019. doi:10.32628/cseit1952280

[2] A. M. Pujar and P. B. Kulkarni, "Automatic number plate recognition using machine learning," *Advances in Computer Science Research*, pp. 149–156, 2023. doi:10.2991/978-94-6463-136-4_16

[3] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using Deep Learning," *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012027, 2021. doi:10.1088/1757-899x/1084/1/012027

[4] A. Gani S. F. *et al.*, "A live-video automatic number plate recognition (ANPR) system using Convolutional Neural Network (CNN) with data labelling on an Android smartphone," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 10, pp. 88–95, 2021. doi:10.46338/ijetae1021_11

[5] S. Roy, "Automatics number plate recognition using convolution neural network," *Azerbaijan Journal of High Performance Computing*, vol. 3, no. 2, pp. 234–244, 2020. doi:10.32010/26166127.2020.3.2.234.244

[6] G. Hajare, U. Kharche, P. Mahajan, and A. Shinde, "Automatic Number Plate Recognition System for Indian number plates using machine learning techniques," *ITM Web of Conferences*, vol. 44, p. 03044, 2022. doi:10.1051/itmconf/20224403044

[7] D. Gunawan, W. Rohimah, and R. F. Rahmat, "Automatic number plate recognition for Indonesian license plate by using K-nearest neighbor algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 648, no. 1, p. 012011, 2019. doi:10.1088/1757-899x/648/1/012011

[8] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Real-time computer vision with opencv," *Communications of the ACM*, vol. 55, no. 6, pp. 61–69, 2012. doi:10.1145/2184319.2184337

[9] P. Sharma *et al.*, "AUTOMATIC NUMBER PLATE RECOGNITION," *International Journal of Scientific and Technical Advancements*, vol. 5, no. 1, pp. 97–100, 2019.

[10] B. Baby, J. Jose, L. Thomas, and S. George, "Automatic Number Plate Recognition System Using OpenCV And Machine Learning," *National Conference in Emerging Computer Applications,* 2019.

[11] D. Narula, Dr. M. Mahajan, and K. Kaur., "Automated car number plate recognition system using K-nearest neighbor," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 1113–1117, 2019. doi:10.35940/ijitee.j8782.0881019

[12] T. Damak, O. Kriaa, A. Baccar, M. A. Ben Ayed, and N. Masmoudi, "Automatic Number Plate Recognition System Based on Deep Learning," *International Journal of Computer and Information Engineering*, vol. 14, no. 3, 2020.

[13] V. Rai and D. Kamthania, "Automatic Number Plate Recognition," *SSRN Electronic Journal*, 2021. doi:10.2139/ssrn.3879574

[14] S. Ranjithkumar and S. Chenthur Pandian, "Automatic License Plate Recognition System for vehicles using a CNN," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 35–50, 2022. doi:10.32604/cmc.2022.017681

[14] S. Tonge, B. Hemwani, G. Thawani, S. Katara, and N. Dhanwani, "AUTOMATIC NUMBER PLATE RECOGNITION," *International Research Journal of Modernization in Engineering Technology and Science,* vol. 4, no. 4, Apr. 2022.

[15] Y. Shambharkar, S. Salagrama, K. Sharma, O. Mishra, and D. Parashar, "An automatic framework for number plate detection using OCR and deep learning approach," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023. doi:10.14569/ijacsa.2023.0140402

[17] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, 2018. doi:10.1155/2018/7068349

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, 2016.

[19] A. Gattawar, S. Vanwadi, J. Pawar, P. Dhore, and H. Mhaske, "Automatic Number Plate Recognition using YOLO for Indian Conditions," *International Research Journal of Engineering and Technology*, vol. 8, no. 1, 2021.

[20] B. Setiyono, D. A. Amini, and D. R. Sulistyaningrum, "Number Plate recognition on vehicle using YOLO - Darknet," *Journal of Physics: Conference Series*, vol. 1821, no. 1, p. 012049, 2021. doi:10.1088/1742-6596/1821/1/012049

[21] R. Al-batat, A. Angelopoulou, S. Premkumar, J. Hemanth, and E. Kapetanios, "An end-to-end automated license plate recognition system using YOLO based vehicle and license plate detection with vehicle classification," *Sensors*, vol. 22, no. 23, p. 9477, 2022. doi:10.3390/s22239477

[22] A. Tourani, A. Shahbahrami, S. Soroori, S. Khazaee, and C. Y. Suen, "A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems," *IEEE Access*, vol. 8, pp. 201317–201330, 2020. doi:10.1109/access.2020.3035992

[23] Y. Y. Lee, Z. Abdul Halim, and M. N. Ab Wahab, "License plate detection using convolutional neural network–back to the basic with design of experiments," *IEEE Access*, vol. 10, pp. 22577–22585, 2022. doi:10.1109/access.2022.3153340

[24] S. Shivam *et al.*, *Intelligent Automatic Number Plate Recognition System*, 2023.

[25] G Poorani, B. R. K. Krishnna, T. Raahul, and P. P. Kumar, "Number plate detection using Yolov4 and TESSERACT OCR," *Journal of Pharmaceutical Negative Results*, vol. 13, no. SO3, 2022. doi:10.47750/pnr.2022.13.s03.021

26] K. Premaa, P. Arivubrakanb, and V. Suganyac, "Automatic Number Plate Recognition Using Deep Learning," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 9, 2021.

[27] V. Nayak, "Automatic Number Plate Recognition," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3783–3787, 2020. doi:10.30534/ijatcse/2020/195932020

[28] G. Hajare, U. Kharche, P. Mahajan, and A. Shinde, "Automatic Number Plate Recognition System for Indian number plates using machine learning techniques," *ITM Web of Conferences*, vol. 44, p. 03044, 2022. doi:10.1051/itmconf/20224403044

[29] V. Burkpalli, A. Joshi, A. B. Warad, and A. Patil, "AUTOMATIC NUMBER PLATE RECOGNITION USING TENSORFLOW AND EASYOCR," *International Research Journal of Modernization in Engineering Technology and Science*, 2022.

[30] D. Salunke, R. Joshi, R. Patil, and S. Patinge, "Automatic Vehicle Plate Recognition using Deep Learning," *Grenze International Journal of Engineering and Technology*, 2023.

[31] Y. Shambharkar, S. Salagrama, K. Sharma, O. Mishra, and D. Parashar, "An automatic framework for number plate detection using OCR and deep learning approach," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023. doi:10.14569/ijacsa.2023.0140402

[32] Z. Akhtar and R. Ali, "Automatic number plate recognition using random forest classifier," *SN Computer Science*, vol. 1, no. 3, 2020. doi:10.1007/s42979-020-00145-8

[33] U. Salimah, V. Maharani, and R. Nursyanti, "Automatic license plate recognition using optical character recognition," *IOP Conference Series: Materials Science and Engineering*, vol. 1115, no. 1, p. 012023, 2021. doi:10.1088/1757-899x/1115/1/012023

[34] P. Kaur *et al.*, "Automatic License Plate Recognition System for Vehicles Using a CNN," *Computers, Materials & Continua*, 2021.

[35] G. Abbass and A. Marhoon, "Iraqi license plate detection and segmentation based on Deep Learning," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 17, no. 2, pp. 102–107, 2021. doi:10.37917/ijeee.17.2.12

[36] D. Gondhalekar, O. Chalke, S. Bansal, and S. Banerjee, "Vehicle license plate recognition using neural networks," *SSRN Electronic Journal*, 2021. doi:10.2139/ssrn.3866116

IEEE 979-8-3503-1812-8/23/$31.00 ©2023 IEEE