Random Multi-Augmentation to Improve TensorFlow-Based Vehicle Plate Detection

Kartika Candra Kirana, Salah Abdullah Khalil Abdulrahman
Department of Electrical Engineering, Universitas Negeri Malang, Indonesia

ARTICLE INFORMATION

Article History:

Submitted 22 April 2024
Revised 31 May 2024
Accepted 04 June 2024

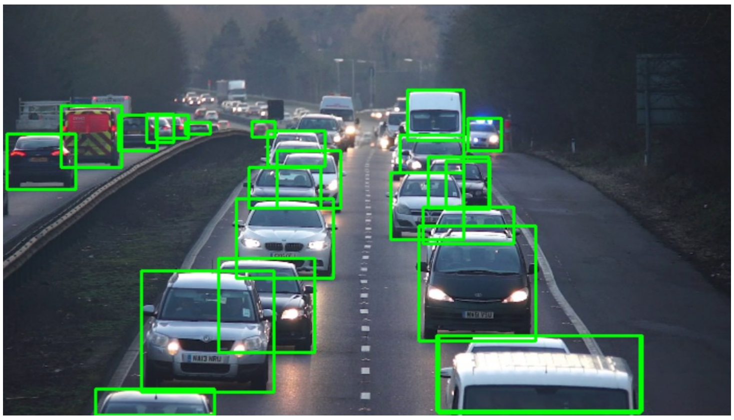
Keywords:

TensorFlow;
Augmentation;
Vehicle Number Plate;
Detection;
Edukit

Corresponding Author:

Kartika Candra Kirana,
Universitas Negeri Malang,
Indonesia.
Email:
kartika.candra.ft@um.ac.id

ABSTRACT



In the development of the "Machine Learning" education kit, vehicle plate recognition was created using TensorFlow with SSD MobileNetV2. The detection failure rate in the training process with varying distances and lighting from the camera is high if the training data is insufficient. Addressing that notable gap in research, we proposed Random Multi-Augmentation to Improve TensorFlow-Based Vehicle Plate Detection. Augmentation techniques are expected to train data that is manipulated at varying lighting and distance. The proposed method consists of two combining augmentation approaches, namely: position augmentation and lighting augmentation. Position augmentation which consists of Flip, Crop, Rotate, Shift, and Crop is used to enrich the visualization of distance and viewing angle, while Lighting augmentation which consists of Greyscale, Hue, Saturation, Brightness, Exposure, and Blur is used to enrich the visualization of lighting. Variations in values were determined randomly based on variations in values from several previous studies. The comparison of TensorFlow SSD MobileNetV2 and Augmentation were tested using one video Roboflow. TensorFlow without augmentation exhibited an accuracy of 60%, precision of 100%, recall of 60%, and an F1 score of 75%, whereas TensorFlow within augmentation achieved a higher accuracy of 70%, precision of 100%, recall of 70%, and an F1 score of 82.3%. Based on precision measurement, Tensorflow can be claimed to prevent false positives, which indicates that the algorithm did not detect non-plate objects as vehicle plates. Furthermore, a comparison of the use of augmentation shows an increase in plate detection capabilities when using augmentation as Tensorflow preprocessing, which is indicated by an increase in recall and accuracy values. These results emphasize that augmentation is the pre-processing optimizer for vehicle number plate systems.

Document Citation:

K. C. Kirana and S. A. K. Abdulrahman, “Random Multi-Augmentation to Improve TensorFlow-Based Vehicle Plate Detection,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 2, pp. 113-125, 2024, DOI: 10.12928/biste.v6i2.10542.

1. INTRODUCTION

License plate recognition is an essential stage for an automatic number plate recognition system (ANPR) [1][2]. These systems are widely used in security surveillance and law enforcement [3]. Developments of the ANPR technology are required as the need for effective traffic management systems rises in tandem with the growth of car traffic [4]. The benefits of ANPR have led to the development of vehicle plate detection education for "Machine Learning" courses. Despite their usefulness, ANPR systems have difficulties detecting vehicles with accuracy and dependability, especially in situations requiring real-time response [5].

Tensorflow is a well-known open source commonly used in machine learning. Tensorflow presents the MobileNetSSDv2 (MobileNet Single Shot Detector) detection model which presents linear bottleneck and shortcut connections features. Several studies show that TensorFlow SSD MobileNetV2 outperforms other methods such as YoloV3 [6]–[8]. However TensorFlow detect accuratelly when supported by rich learning data [9]. Ashrafee et.al. (2022) train a Bangla 130,000 image using TensorFlow SSD MobileNetV2 and test their pipeline on a video dataset with real-time processing speed (27.2 frames per second), achieving an 82.7% identification rate [6]. Gautam et al. (2023) used MobileNetV2 to train the 20,054 license plate images and Convolutional Neural Networks (CNNs) to find cornet points in license plate images after which plate correction was performed using perspective transformation [10]. Both of studies [6][10] strengthen the argument of Haghighat (2021) about the relationship between accuracy and the amount of training data.

On the other hand, the challenges of lighting and distance in plate detection are still issues. Burkpalli et.al. (2022) show that a traffic management system using TensorFlow and Easy OCR reaches 89% on 448 images CCPD dataset [11]. However, they cannot detect the blurred plate image that causes vehicle movement, and their proposed methods were not conducted using real-time data. Sharma et.al. (2022) modify detection using contour-based TensorFlow to repair the representation of vehicle plates in parking lots. Bayesian rule-three Bayesian rules are adapted to apply the adaptive contours [12]. Tote et al. (2023) show the contribution of red, green, and blue color distribution improves TensorFlow in 50 for license plate images up to 79% confidence rate [13].

In this study, we use the 1.126 Plate Number images in the Roboflow dataset The dataset we use in the real-time system is relatively small when compared with previous methods. Moreover, test data has more complex angle, distance, and lighting characteristics compared to training data. In other words, the training data supplied includes images of number plates in ideal settings, which do not accurately reflect the plates' state at various camera angles and color schemes in real-time. Based on the aforesaid evaluation, the investigations focus on method optimization or test data preprocessing. Those methods increase processing time, which is contrary to the real-time character, which necessitates a quick procedure. Our primary idea is to optimize the training data from numerous angles, hence varying illumination changes and camera viewing angles. This is expected to reduce computing time on real-time testing.

One method of enriching data is Augmentation which is manipulating objects at varying lighting and visualization. In various research, the augmentation procedure has been shown to significantly boost machine learning performance [14][15]. Data augmentation is a popular strategy for increasing the amount and variety of training data. Kumdakcı et al. found that augmentation boosts average precision by up to 25.2% and 25.7% [16]. Chen et al. (2022) determined that the mean average precision (mAP) of Yolov4 with augmentation increased by 1.93% when compared to the YOLOv4 model trained without data augmentation for vehicle recognition [17]. Wang et al (2023) solved augmentation-based fusion to classify a small number of labeled samples with rich features for vehicle classification. The accuracy of their proposed system reaches 92.28% accuracy on the BIT-Vehicle dataset and 82.43% accuracy on the car10 dataset [18]. On the Caltech Cars dataset, the use of augmentation in the recognition of letter characters on license plates resulted in 96% accuracy for segmentation and recognition [19]. Silva et al. (2020) investigated data augmentation for training deep CNNs on short datasets [20]. Khurram Khan et al. al (2020) applied HSI color space augmentation using a multiple number plate recognition method, for high-resolution images. The increase in accuracy when applying augmentation reaches 2% [7]. Orhan Bulan et al (2017) perform segmentation and optical character recognition (OCR) jointly by using a probabilistic inference method based on hidden Markov models (HMMs) for localizing the license plate that is too bright or too dark [21].

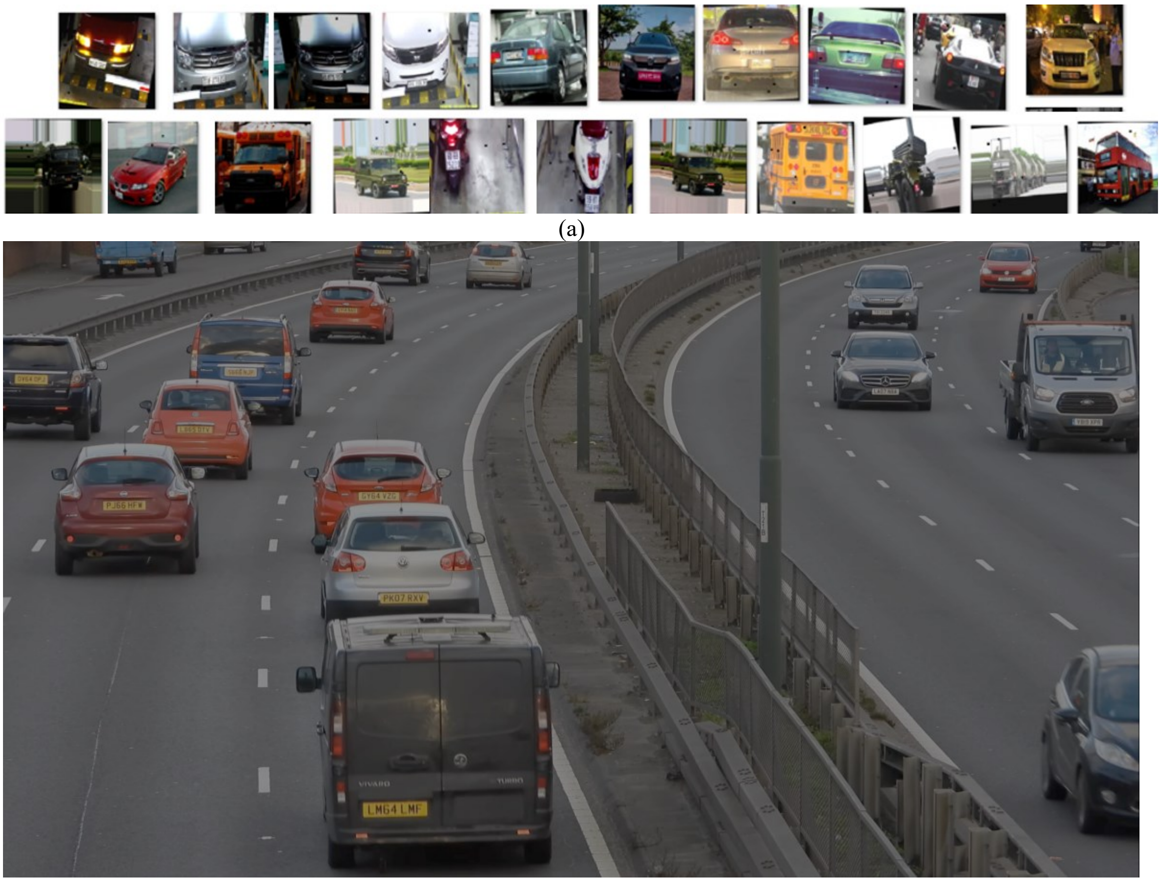
Based on the explanation above, previous research has demonstrated that TensorFlow's capabilities thrive in ANPR when supported by rich learning data [6]–[10]. However, license plate recognition which multiple cars traveling quickly are situated at varying distances and lighting from the camera remains a challenge [13]. Moreover, the detection failure rate in the training process with varying distances and lighting from the camera is high if the training data is insufficient [11]–[13]. Addressing that notable gap in training on small amounts of data with limited light and angle knowledge representation, we propose TensorFlow optimization with MobileNetV2 SSD using augmentation to enrich our limited training data representation. Augmentation techniques are expected to train data that is manipulated at varying lighting and distance. Thus the proposed method combines Tensorflow with SSD MobileNetV2 as machine learning and two augmentation approaches which aim to enrich the visualization of distance and viewing angles and enrich the visualization of lighting. An explanation of the proposed method can be seen in Chapter 2.

2. METHODS

2.1. Data Gathering

The dataset utilized in this study is the License Plate Recognition Image Dataset, which includes one class (license plate) and was obtained from the Roboflow website. It has a total of 1,126 images before augmentation. The photographs are all JPGs, resized to 640x640 with a 1:1 ratio. This dataset contains photos of license plates taken under various situations and environments, offering a broad sample for training and evaluation. Figure 1(a) depicts numerous examples of training data. The test data was provided by Roboflow. We may assess the model's generalizability and efficacy in practical scenarios by running it on previously unseen test data. The test data that will be tested consists of one MP4 video, as illustrated in Figure 1(b). Some of the recordings were shot manually with a smartphone camera, while others were captured with CCTV. Meanwhile, the videos to be tested have a 16:9 aspect ratio and various resolutions.

Based on the comparison of training data and test data in Figure 1 and Figure 2, it can be seen that there are significant differences in data characteristics. In the test data (Figure 2) the data tends to be darker, while the training data (figure 2) tends to be dominated by bright images. Thus, lighting augmentation is proposed in this case. Additionally, Figure 2 illustrates the variation in image acquisition angles, with vehicles facing away from the camera predominating. Compared to the training image displayed in Figure 1, the acquisition of cars facing the camera is more dominant. Thus, position augmentation has a significant impact on this case study.



(a)

(b)

Figure 1. Dataset (a) Training Data [8] (b)Test Data Set (Video)

2.2. The Proposed Method

Figure 2 serves the proposed system architecture, illustrating the integration of the chosen models and highlighting their respective functionalities in real-time ANPR scenarios:

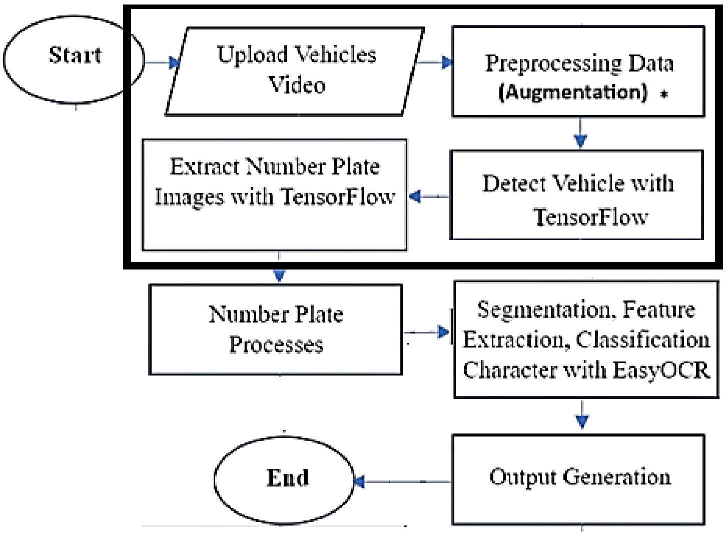
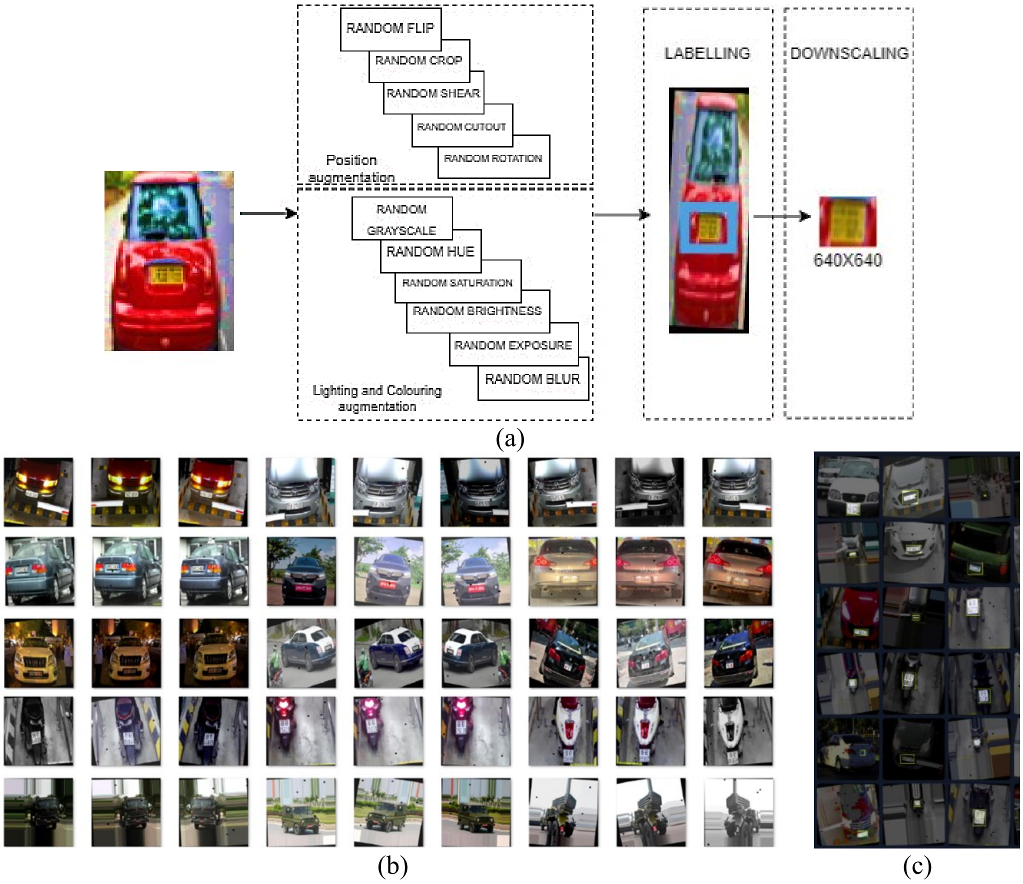

Start
Upload Vehicles Video
Preprocessing Data (Augmentation) *
Detect Vehicle with TensorFlow
Extract Number Plate Images with TensorFlow
Number Plate Processes
Segmentation, Feature Extraction, Classification Character with EasyOCR
Output Generation
End

Figure 2. The Proposed Method (*The contribution Research)

2.2.1. Pre-Processing Using Augmentation

Figure 3(a) shows that preprocessing consists of downscaling, labeling, and augmentations which was used to assure its suitability for training machine learning models. Augmentations add to the dataset, improving model generalization and robustness by introducing variability which is shown in Figure 3(b). These two approaches are implemented in eleven augmentation techniques. Labeling is an important asset in interpreting vehicle plates. Then, The training images are downsized to 640x640 pixels for standardized input size after labeling, allowing for consistent feature learning. The sample of labeling and downscaling is shown in Figure 3(c).



(a)

(b) (c)

Figure 3. Preprocessing (a) Flowchart (b) Augmentation (c) Labeling and Downscaling

Table 1 describes specific augmentation approaches aimed at diversifying the dataset and improving the model's ability to learn from various perspectives and conditions. There are two categories of augmentation used, namely position augmentation and lighting augmentation. These two categories were chosen referring to the data characteristics that were discussed in 2.1.

The eleven augmentation techniques are applied simultaneously with random techniques. Five techniques that constitute positional augmentation consist of Flip, Crop, Rotate, Shift, and Crop. Flip is used to augment with mirroring techniques. There are two possible values to be set, namely no flip and horizontal flip. The horizontal flip reflects the recorded vehicle in a different direction. The second augmentation technique is crop which is applied randomly with a choice of no crop or 15% crop. The third technique is random rotation which is used to achieve orientation variations. Horizontal and vertical shear is the fourth technique used to achieve perspective distortion. Five rectangular cutout regions were applied to simulate occlusions or noise.

There are six color and light augmentations applied, namely: grayscale, hue, saturation, brightness, exposure, and blur. The six combinations aim to increase variations in test images which are influenced by exposure to the sun, lights, and weather. Hue is applied randomly altering the color tone, while Saturation introduces variations in color intensity. Hue and Saturation are taken from the HSV feature to extract color degree (hue) and color purity (saturation). Grayscale introduces variations in color space to easier extract predominantly white vehicle plates. Exposure mimics the amount of light that falls on an object, while brightness adjusts the light of images. Both features were chosen because the outdoor object is influenced by the sun so light exposure needs to be observed. Blur presents motion blur or camera imperfections. Blur was chosen because there is a possibility of objects blurring due to vehicle movement. The selected values refer to previous research that applied a single augmentation technique which can be seen in the citation in Table 1.

Table 1. Augmentation Techniques

Augmentation Approach	Augmentation Techniques	Value
Position	Flip	[No Flip, Horizontal Flip][20]
	Crop	[0%,15%][17]
	Rotation	[-10°, +10°][23]
	Shear	[-2°,2°][22]
	Cutout	2% size from the center[17]
Lighting	Grayscale	[No Grayscale,10%] [19] [7]
	Hue	[-15°, 0, +15°] [7]
	Saturation	[-15°, 0, +15°] [13]
	Brightness	[-15°, 0, +15°] [13]
	Exposure	[-15°, 0, +15°][24]
	Blur	Gaussian blur[24]

2.3. Tensorflow Using SSD MobileNetV2

After the dataset has been augmented, TensorFlow uses the pre-trained SSD MobileNetV2 (Single Shot Multibox Detector) model for object detection. which is shown in Figure 4.

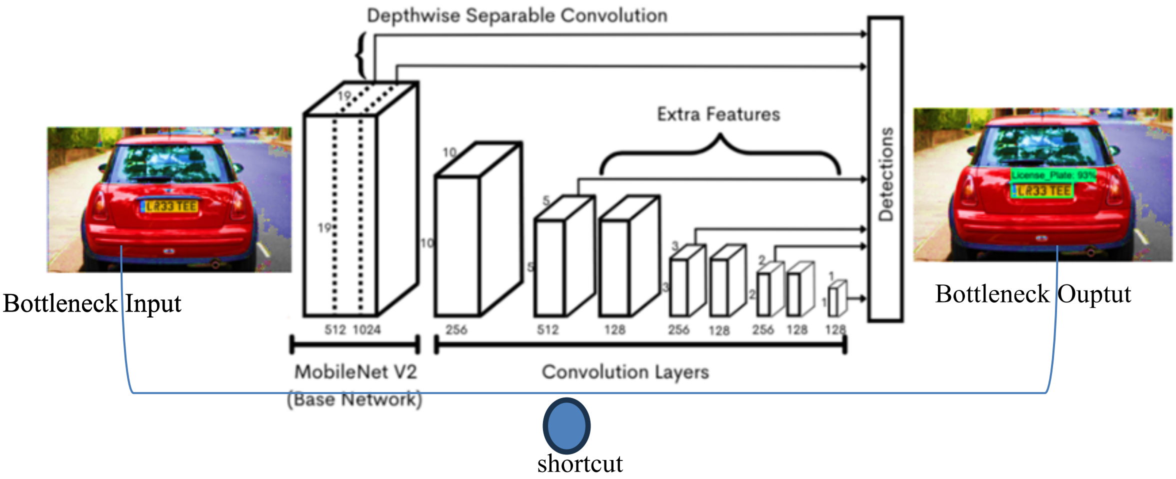


Figure 4. Tensorflow SSD MobileNetV2

Based on Figure 4, MobileNetV2 does depthwise and pointwise convolution. Moreover, MobileNetV2 introduces two new features, consist of: linear bottleneck and shortcut connections. Depthwise convolution uses a single convolutional filter for each input channel, whereas pointwise convolution creates a linear combination of the depthwise convolution outputs. The bottleneck part contains inputs and outputs from the model, whilst the inner layer or layers encapsulate the model's capacity to convert information from lower level concepts (pixels) to higher level descriptors (picture categories). Tensorflow combines MobileNetV2's with the SSD framework to enable real-time detection. The procedure begins with the model processing the full image, extracting information at various scales. SSD MobileNetV2 then localizes and classifies items, including license plates, while anticipating bounding boxes and class labels. Post-processing techniques refine predictions by deleting duplicates and false positives. The final result includes accurate bounding boxes around recognized items, as well as their labels and confidence scores. This streamlined approach offers rapid and dependable license plate detection in a variety of applications.

The optimal hyperparameters for the TensorFlow with SSD MobileNetV2 model are required to provide reasoning based on past research findings. The factors were developed based on their findings, as well as insights from other relevant. Several hyperparameters were useful in optimizing the performance of deep learning models, consisting of pre-trained weights, epoch numbers, optimizer functions, and batch size. They considered how these characteristics affected model performance and the risk of overfitting. This test specifically measures the performance of Pretrained Weight SSD MobileNetV2 [25]. Meanwhile, we have observed that the combination of the Number of Epochs, Optimizer Function, and Batch Size has been tested in previous research [26]-[28]. The setting value also refers to [25] By considering efficient time with good performance, thus the selected parameters are shown in Table 2. The selection of hyperparameters refers to previous research which is shown through citations in Table 2.

Table 2. Test Parameters

Tested Parameters	Test Value Variations
Pretrained Weight	SSD MobileNetV2 [25]
Number of Epochs	100 [26]
Optimizer Function	Adam [27]
Batch Size	32 [28]

2.3.1. Detect Vehicle Using Tensorflow

The Tensorflow model is used to identify automobiles in pre-processed video frames. Tensorflow is a cutting-edge object identification model noted for its speed and accuracy when recognizing things in photos or videos. By integrating Tensorflow, pretrained model on the COCO dataset, for comprehensive object representation, enabling accurate recognition of cars in video streams, as shown in Figure 5.

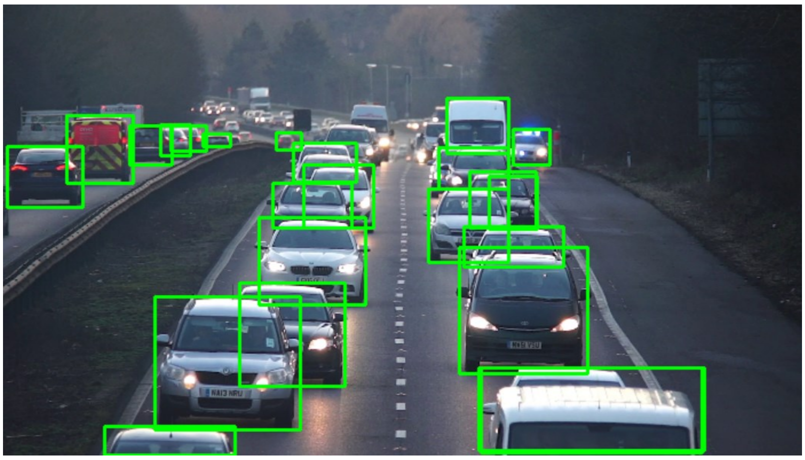


Figure 5. Detect Vehicles Using TensorFlow with MobileNetSSDV2

2.3.2. Extract the Number plate Image Using TensorFlow

Following vehicle detection, TensorFlow with SSD-MobileNet-v2 models also extract license plates from detected automobiles. These models use deep learning approaches to reliably detect and localize license plates within video frames, as seen in Figure 6.

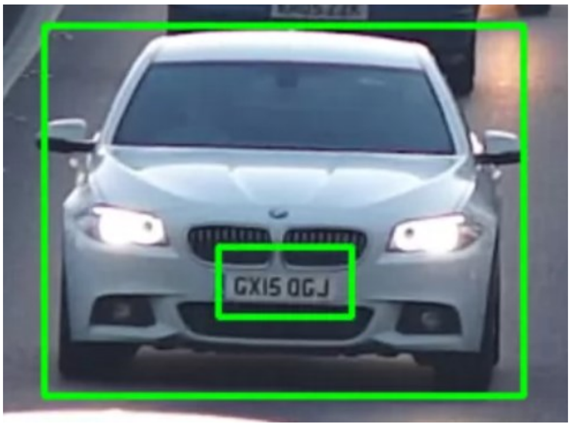
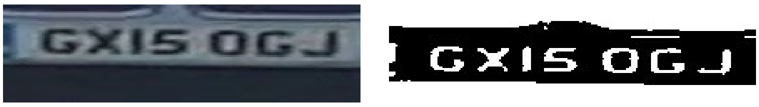


Figure 6. Extract Number Plate Image with TensorFlow with MobileNetSSDV2

2.4. Number Plate Processing using Adaptive Thresholding

After removing the license plates, several processing procedures are performed to prepare the license plate pictures for character recognition. This includes tasks such as converting the license plate crop to grayscale and using adaptive thresholding to generate the binary picture. The adaptive thresholding method [29] used here allows for successful binarization of the license plate picture, as illustrated in Figure 7, which is required for subsequent character segmentation and recognition.



(a) (b)

Figure 7. Number Plate Processing (Adaptive Threholding)

2.5. The Integration of EasyOCR

The incorporation of EasyOCR into the ANPR architecture to enable accurate text recognition on vehicle license plates. EasyOCR is an open-source Python program that provides an intuitive interface for Optical Character Recognition (OCR) activities. Its simplicity and efficiency make it an excellent addition to our ANPR system, allowing for seamless text extraction from license plate photos. In this study, license plate identification is achieved by combining TensorFlow using SSD MobileNetV2, and EasyOCR as the text recognition library. Figure 8 depicts the process of text recognition using the EasyOCR library. The first step of EASY OCR is feature extraction. The input data must be converted using VGG and Resnet to create a set of features. Resnet is a type of convolutional neural network (CNN) that uses skip connections or shortcuts to go around some layers, whereas Visual Geometry Group (VGG) has a homogeneous architecture consisting of very small (3x3) convolution filters. Then Sequence Labeling is carried out after feature extraction using a Long Short-Term Memory (LSTM) network. LSTMs are superior to traditional RNNs because have a unique design that helps avoid the vanishing gradient problem. Decodering is the last step which employed the Connectionist Temporal Classification (CTC). CTC adds a blank label to the list of current labels in order to produce a prediction with varying length. Because it computes loss by adding up all possible alignments of the input to the target sequences, it is then ideal for OCR decoding applications. The output result is shown in Figure 9.

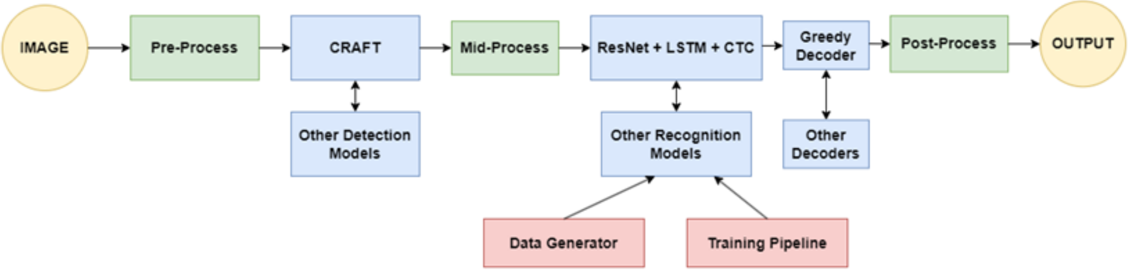


Figure 8. EasyOCR Workflow [30]

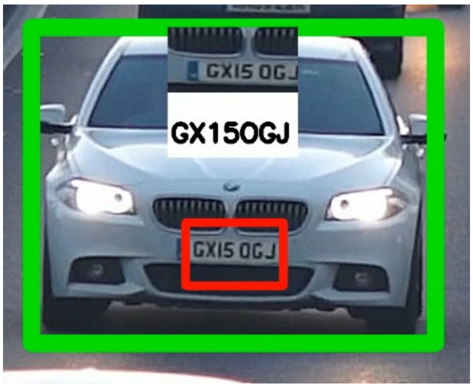


Figure 9. Output

2.6. Evaluation

Based on the proposed method, Tensorflow is used twice, namely for car detection and license plate detection. The limitation of this research is comparing the performance of TensorFlow SSD-MobileNet-v2 by the use of augmentation for vehicle plate detection. The influence of tensorflow on car detection is proposed in a different article. Apart from that, the limitations of this research also lie in testing the use of OCR that is applied, so that plate number testing can be continued in further research.

One video was tested on Windows using the NVIDIA GeForce RTX 3070 Ti 8GB GPU. Model performance will be evaluated using multiple metrics, including F1 Score, Accuracy, Precision, and Recall. Accuracy is a fundamental evaluation statistic that measures how accurate the model's predictions are across all classes and is based on the confusion matrix. Equation (1) shows how it is calculated as the ratio of successfully identified license plate to total occurrences in the dataset. The precision metric, which is calculated by applying Equation (2), compares the correct predictions to all plate prediction results produced. The recall metric identifies all true license plate cases and is determined using Equation (3). Equation (4) is used to generate the F1 Score metric, which is the harmonic average of the precision and recall values to show how well the two metrics are balanced.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

The confusion matrix shows the accuracy with which the system predicts the license plates. True Positives (*TP*) are occasions in which the system accurately detects the presence of a license plate. False positives (*FP*) arise when the system incorrectly predicts the presence of a license plate in the absence of one. False Negatives (*FN*) occur when the system fails to detect a number plate even though one exists. True Negatives (*TN*) represent accurate predictions of the absence of a number plate.

3. RESULT AND DISCUSSION

3.1. Validation Testing

In this section, we will compare TensorFlow and augmentation in validation testing which is shown in Table 3. The TensorFlow training process with a MobileNetV2 SSD took approximately 0.90 hours longer and resulted in a retrained model size that was 1.81 MB larger than without the augmentation approach. Although both models were trained with the same parameters of 100 epochs, Adam optimizer, and batch size of 32, the differences in training time and a retrained model size were not striking in hardware capability and efficiency. The insignificant differences in computing time and batch size would be advantageous if considering better performance. This is shown through improvements in mAP50 and Total Loss.

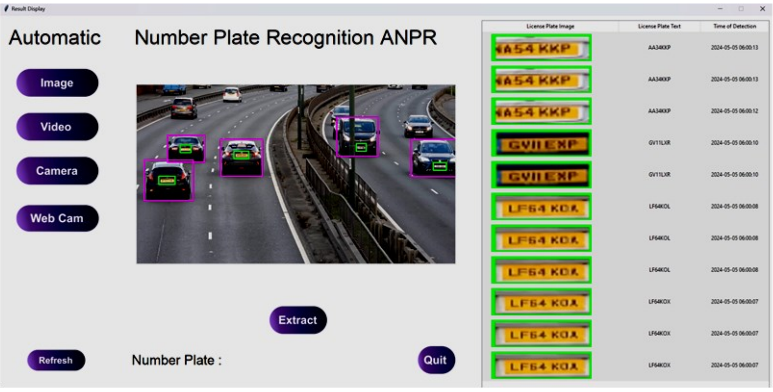
The combination of augmentation and Tensorflow yielded a mAP50 of 0.674, showing higher precision metrics than without augmentation. The combination of augmentation with Tensorflow resulted in an average localization loss of 0.615, a classification loss of 0.746, and a total loss of 1.363, which is better than Tensorflow without augmentation. While a smaller total loss is normally better, both models perform well across a variety of measures.

Table 3. Metric Evaluation of Training Results

Metric	The Result of TensorFlow	
	Without Augmentation	**With Augmentation**
mAP50	0.573	0.674
Localization Loss	1.01	0.615
Classification Loss	0.3945	0.746
Total Loss	1.404	1.363
Time	7.25 hours	8.15 hours
Retrained Size	21.4 MB	39.5 MB

3.2. Testing Result

The deployment of our Number Plate Recognition is shown in Figure 10. Figure 10 presents the ANPR test in one video. 20 license plate numbers were recorded while the video was playing.



License Plate Image	License Plate Text	Time of Detection
	AA34KKP	2024-05-05 06:00:13
	AA34KKP	2024-05-05 06:00:13
	AA34KKP	2024-05-05 06:00:12
	GV11LXR	2024-05-05 06:00:10
	GV11LXR	2024-05-05 06:00:10
	LF64KOL	2024-05-05 06:00:08
	LF64KOL	2024-05-05 06:00:08
	LF64KOL	2024-05-05 06:00:08
	LF64KOX	2024-05-05 06:00:07
	LF64KOX	2024-05-05 06:00:07
	LF64KOX	2024-05-05 06:00:07

Figure 10. Deployment

The analysis focuses on model predictions about the basic information provided by the confusion matrix. The confusion matrix of Tensorflow and Augmentation in (LPR) is presented in Table 4. Based on Table 4, The results of the augmentation and TensorFlow revealed 14 true positives, implying that 14 vehicles could be detected. Furthermore, false positives reached 6 plate, implying that 6 plate were not detected. In this case, there are no vehicles without a plate, resulting in a false negative of zero. Further, no false negatives were found.

Table 4. Confusion Matrix Result of Tensorflow and Augmentation in (LPR)

N=20		Predicted	
		Lisence Plat	**No Lisence Plat**
Actual	**Lisence Plat**	14 (TP)	6 (FN)
	No Lisence Plat	0 (FP)	0 (TN)

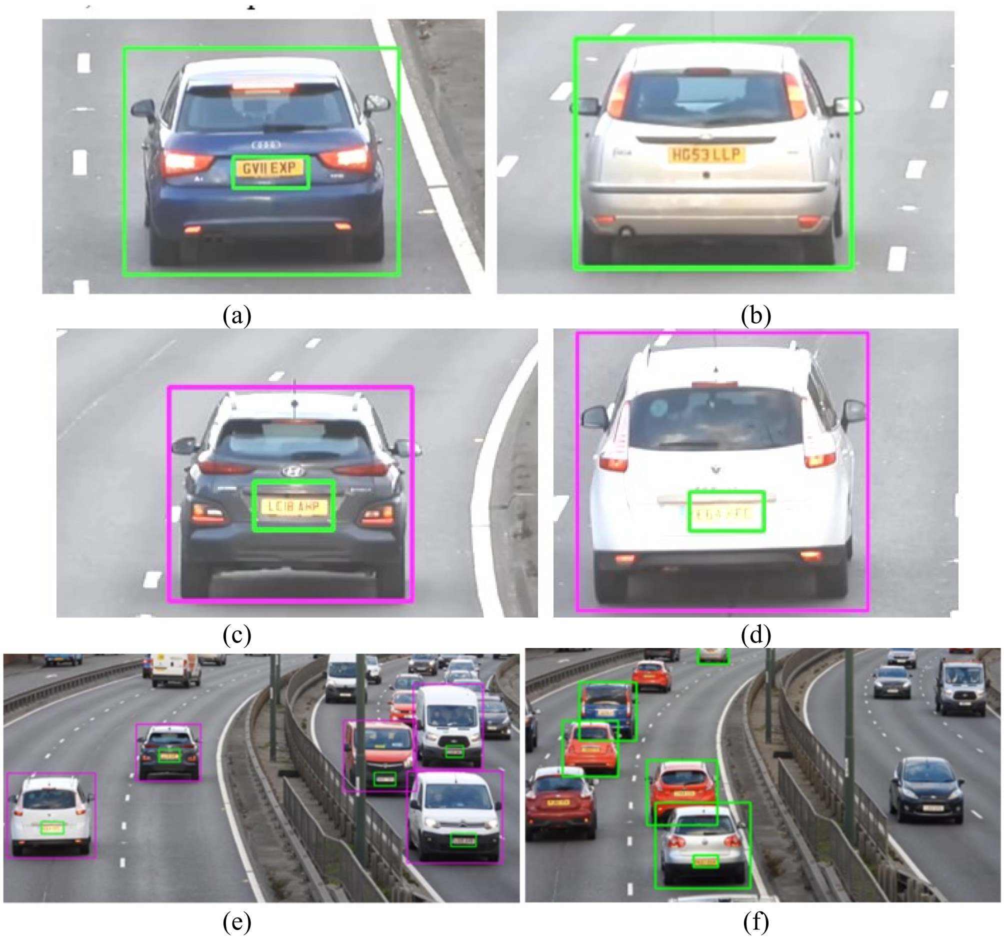
Table 5 shows the results of the TensorFlow without augmentation revealed 12 true positives, implying that 12 vehicles could be detected. This true positive value in methods that do not use augmentation is smaller than the true positive value in methods that use augmentation. Furthermore, false positives reached 8 plate, implying that 8 plate were not detected. In this case, there are no vehicles without a plate, resulting in a false negative value of zero. Both results have an impact on the values of accuracy, pressure, and recall that are displayed in Table 6.

Table 5. Confusion Matrix Result of Tensorflow without Augmentation in (LPR)

N=20		Predicted	
		Lisence Plat	**No Lisence Plat**
Actual	**Lisence Plat**	12 (TP)	8 (FN)
	No Lisence Plat	0 (FP)	0 (TN)

Table 6 shows a comparison of evaluation metrics between empowered Tensorflow and augmented-assisted Tensorflow without augmentation. Tensorflow SSD MobileNetV2 without augmentation exhibited an accuracy of 60% and recall of 60, whereas SSD MobileNetV2 within augmentation achieved a higher accuracy of 70% and recall of 70%. Accuracy and recall in the combined Augmentation and TensorFlow increased due to the increase in vehicle plate detection capability (true positive) which increased from 12 (Table 5) to 14 (Table 4). This comparison indicates the use of augmentation enriches knowledge for machine learning. Precision measurements show the same strength. Both methods, Tensorflow and augmented-assisted Tensorflow without augmentation exhibited a precision of 100%. These measurements show that both of them do not overclaim in detecting vehicle plates, which is equal to their ability to recognize non-vehicle plates. The F1-Score indicates that Tensorflow with augmentation outperforms the other technique in terms of recall and precision harmonizing because both methods have similarly strong recall but higher precision. It is shown by the F1-Score results for Tensorflow with Augmentation which is around 10% better than Tensorflow without Augmentation.

The hypothesis that augmentation improves Tensorflow performance is supported by Figure 11. There are vehicle plates that are not detected in Tensorflow without augmentation shown in Figure 11(b), but are detected in the combination of Tensorflow and augmentation shown in Figure 11(d). In a more specific domain, the use of augmentation can improve the False Negative into a True Positive shown in the comparison of Figure 11(b) and Figure 11(d). These improvements are supported by lighting augmentation. Training data manipulated through color and lighting augmentation can help extract license plates from cars that have identical colors. This support is helped by supporting a combination of hue, saturation, exposure, and brightness. However, a lot of vehicles are equally well detected by both compared methods which is shown in Figure 11(a) and Figure 11(c). A detailed comparison in one video frame can be seen in Figure 11(e) for Tensorflow with Augmentation and Figure 11(f) for Tensorflow without Augmentation.



(a) (b)

(c) (d)

(e) (f)

Figure 11. Comparison of Result (a), (b), (e) TensorFlow without Augmentation (c), (d), (f) TensorFlow with Augmentation (a), (c), (d) True Positive Image Fragments (b) False Negative Image Fragments (e), (f) Detection Results on One Video Frame

Table 6. Metric Evaluation of Testing Results

Metric	The result of TensorFlow	
	Without Augmentation	**With Augmentation**
Accuracy (%)	60%	70%
Recall (%)	60%	70%
Precision (%)	100%	100%
F1 Scores (%)	75%	82.3%

Comparing strengths and weaknesses based on confusion matrix analysis and model performance metrics, the augmented implementation shows a lower number of true negatives, indicating its ability to correctly identify events where a license plate is missing. Although MobileNetV2 SSD achieves a higher number of false positives and a lower number of true positives compared to the combination of MobileNetV2 SSD and augmentation, undetected plate conditions due to undetected vehicles are still an issue in Tensorflow.

3.3. Comparative Previous Study

We compare the proposed method with research by Asheefa (2020) where the test data used is also the license plate contained in the test video. Given the research limitations, the performance of the proposed method is superior. However, the comparative researchers were superior in the amount of test data where the license plates detected reached 87 units. For this reason, this research needs to test further in ongoing research on the use of TensorFlow on more comprehensive video data can be seen in Table 7.

Table 7. Comparative Previous Studies with This Study

Methodology	F1 Score
Cascade classifier and SSD MobileNetV2 [6][31]	60.8%
The Proposed method	82.3%

4. CONCLUSIONS

The proposed method was tested using hyperparameters, including the number of epochs, optimizer function, and batch size. This research employed the SSD MobileNetV2 using the comparison of augmentation. SSD MobileNetV2 without augmentation exhibited an accuracy of 60%, precision of 100%, recall of 60%, and an F1 score of 75%, whereas SSD MobileNetV2 within augmentation achieved a higher accuracy of 70%, precision of 100%, recall of 60%, and an F1 score of 82.3%. Based on test results, Tensorflow can independently prevent overclaim detection in non-plate images, which is demonstrated by a precision value that reaches 100%. Furthermore, a comparison of the use of augmentation shows an increase in plate detection capabilities when using augmentation as Tensorflow preprocessing, which is indicated by an increase in recall and accuracy values. These results emphasize that augmentation is the pre-processing optimizer for ANPR systems. In this study, it was found that Tensorflow without augmentation failed to detect vehicle plates when the plate and vehicle colors were identical. However, the combination of light augmentation techniques succeeded in detecting the vehicle plate in this case. It shows the usefulness of lighting augmentation. Although we have not seen a significant effect of using position augmentation for vehicle plate detection, but position augmentation can influence vehicle detection which tests can be investigated in further research. Thus adaptive augmentation methods for problem characteristics need to be developed as well. According to the limitations of this research, where this research only tests the performance of augmentation and TensorFlow on vehicle plate detection capabilities, several performances have not been tested, such as testing vehicle number detection performance, so the comparison of the use of OCR needs to be explored further. We also found vehicles that were not fully captured on video. Coincidentally, the plate is not visible, so it does not count towards this measurement. However, it is necessary to observe the vehicle detection performance based on TensorFlow because if the vehicle is not detected, the vehicle plate and number plate cannot be recognized.

ACKNOWLEDGEMENT

This research is the result of a machine learning course whose funding is funded by LPPM UM in 2024. The first author would like to thank the second author who plans to continue this research in his thesis.

REFERENCES

[1] A. Ait Ouallane, A. Bakali, A. Bahnasse, S. Broumi, and M. Talea, “Fusion of engineering insights and emerging trends: Intelligent urban traffic management system,” *Inf. Fusion*, vol. 88, pp. 218–248, 2022, https://doi.org/10.1016/j.inffus.2022.07.020.

[2] X. Yang and X. Wang, “Recognizing License Plates in Real-Time,” *arXiv preprint arXiv:1906.04376*, 2019, https://doi.org/10.48550/arXiv.1906.04376.

[3] G. Boquet, A. Morell, J. Serrano, and J. L. Vicario, "A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection," *Transp. Res. Part C Emerg. Technol.*, vol. 115, p. 102622, 2020, https://doi.org/10.1016/j.trc.2020.102622.

[4] M. Gupta, H. Miglani, P. Deo, and A. Barhatte, "Real-time traffic control and monitoring," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 5, p. 100211, 2023, https://doi.org/10.1016/j.prime.2023.100211.

[5] I. Taufiq, B. Purnomosidi, M. A. Nugroho, and A. Dahlan, "Real-time Vehicle License Plate Detection by Using Convolutional Neural Network Algorithm with Tensorflow," in *2018 2nd Borneo International Conference on Applied Mathematics and Engineering (BICAME)*, 2018, pp. 275–279, https://doi.org/10.1109/BICAME45512.2018.1570513024.

[6] A. Ashrafee, A. M. Khan, M. S. Irbaz, and M. A. Al Nasim, "Real-time Bangla License Plate Recognition System for Low Resource Video-based Applications," *Proc. - 2022 IEEE/CVF Winter Conf. Appl. Comput. Vis. Work. WACVW 2022*, pp. 479–488, 2022, https://doi.org/10.1109/WACVW54805.2022.00054.

[7] K. Khan, A. Imran, H. Z. U. Rehman, A. Fazil, M. Zakwan, and Z. Mahmood, "Performance enhancement method for multiple license plate recognition in challenging environments," *Eurasip J. Image Video Process.*, vol. 2021, no. 1, 2021, https://doi.org/10.1186/s13640-021-00572-4.

[8] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using deep learning," In *IOP Conference series: materials science and engineering*, vol. 1084, no. 1, p. 012027, 2021, https://doi.org/10.1088/1757-899X/1084/1/012027.

[9] E. Haghighat and R. Juanes, "SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks," *Comput. Methods Appl. Mech. Eng.*, vol. 373, p. 113552, 2021, https://doi.org/10.1016/j.cma.2020.113552.

[10] A. Gautam, D. Rana, S. Aggarwal, S. Bhosle, and H. Sharma, "Deep learning approach to automatically recognise license number plates," *Multimed. Tools Appl.*, vol. 82, no. 20, pp. 31487–31504, 2023, https://doi.org/10.1007/s11042-023-15020-w.

[11] A. Y. Sugiyono, K. Adrio, K. Tanuwijaya, and K. M. Suryaningrum, "Extracting Information from Vehicle Registration Plate using OCR Tesseract," *Procedia Computer Science*, vol. 227, pp. 932-938, 2023, https://doi.org/10.1016/j.procs.2023.10.600.

[12] P. Sharma, S. Gupta, P. Singh, K. Shejul, and D. Reddy, "Automatic Number Plate Recognition and Parking Management," in *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pp. 1–8, 2022, https://doi.org/10.1109/ACCAI53970.2022.9752632.

[13] A. S. Tote, S. S. Pardeshi, and A. D. Patange, "Automatic number plate detection using TensorFlow in Indian scenario: An optical character recognition approach," *Mater. Today Proc.*, vol. 72, pp. 1073–1078, 2023, https://doi.org/10.1016/j.matpr.2022.09.165.

[14] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Elsevier. 2006. https://books.google.co.id/books?hl=id&lr=&id=gAGRCmp8Sp8C.

[15] Q. Zhu, L. Fan, and N. Weng, "Advancements in point cloud data augmentation for deep learning: A survey," *Pattern Recognit.*, vol. 153, p. 110532, 2024, https://doi.org/10.1016/j.patcog.2024.110532.

[16] H. Kumdakcı, C. Öngün, and A. Temizel, "Generative data augmentation for vehicle detection in aerial images," In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10-15, 2021, Proceedings, Part VII*, pp. 19–31, 2021, https://doi.org/10.1007/978-3-030-68793-9_2.

[17] X.-Z. Chen, C.-P. Cheng, and Y.-L. Chen, "Data Augmentation Method for Improving Vehicle Detection and Recognition Performance," in *2022 IEEE International Conference on Consumer Electronics - Taiwan*, pp. 419–420, 2022, https://doi.org/10.1109/ICCE-Taiwan55306.2022.9869222.

[18] X. Wang, S. Yang, Y. Xiao, X. Zheng, S. Gao, and J. Zhou, "A vehicle classification model based on deep active learning," *Pattern Recognit. Lett.*, vol. 171, pp. 84–91, 2023, https://doi.org/10.1016/j.patrec.2023.05.009.

[19] R. D. Castro-Zunti, J. Yépez, and S. B. Ko, "License plate segmentation and recognition system using deep learning and OpenVINO," *IET Intell. Transp. Syst.*, vol. 14, no. 2, pp. 119–126, 2020, https://doi.org/10.1049/iet-its.2019.0481.

[20] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 71, p. 102773, 2020, https://doi.org/10.1016/j.jvcir.2020.102773.

[21] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and Annotation-Free License Plate Recognition With Deep Localization and Failure Identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2351–2363, 2017, https://doi.org/10.1109/TITS.2016.2639020.

[22] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," *arXiv preprint arXiv:2204.08610*, 2022, https://doi.org/10.48550/arXiv.2204.08610.

[23] W. Sapitri, Y. N. Kunang, I. Z. Yadi, and M. Mahmud, "The Impact of Data Augmentation Techniques on the Recognition of Script Images in Deep Learning Models," *J. Online Inform.*, vol. 8, no. 2, pp. 169–176, 2023, https://doi.org/10.15575/join.v8i2.1073.

[24] M. Momeny *et al.*, "Learning-to-augment strategy using noisy and denoised data: Improving generalizability of deep CNN for the detection of COVID-19 in X-ray images," *Comput. Biol. Med.*, vol. 136, p. 104704, 2021, https://doi.org/10.1016/j.compbiomed.2021.104704.

[25] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 13, no. 2, pp. 1–43, 2023, https://doi.org/10.1002/widm.1484.

[26] K. C. Kirana, S. Wibawanto, A. Hamdan, and W. N. Hidayat, "Optimization of 2D-CNN Setting for the Classification of Covid Disease Using Lung CT Scan," vol. 1, pp. 143–149, 2022, https://doi.org/10.21609/jiki.v15i2.1083.

[27] K. C. Kirana, S. Wibawanto, N. Hidayah, and G. P. Cahyono, "Improved Neural Network using Integral-RELU based Prevention Activation for Face Detection," *Int. Conf. Electr. Electron. Inf. Eng.*, vol. 6, pp. 260-263, 2019, https://doi.org/10.1109/ICEEIE47180.2019.8981443.

[28] K. C. Kirana, S. Wibawanto, N. Hidayah, and G. P. Cahyono, "The Improved Artificial Neural Network based on Cosine Similarity for Facial Emotion Recognition," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, pp. 45-48, 2019, https://doi.org/10.23919/EECSI48112.2019.8977006.

[29] S. Khazaee, A. Tourani, S. Soroori, A. Shahbahrami, and C. Y. Suen, "A real-time license plate detection method using a deep learning approach," In *International Conference on Pattern Recognition and Artificial Intelligence* (pp. 425-438, 2020, https://doi.org/10.1007/978-3-030-59830-3_37.

[30] K. C. Kirana, G. D. K. Ningrum, D. U. Soraya, F. Alqodri, A. A. Mubarak, and S. Wibawanto, "Synchronization of the Internship Registration System using a Study Plan and Acceptance Letter Based on Tesseract-OCR and Template Matching," in *2023 8th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, pp. 1–6, 2023, https://doi.org/10.1109/ICEEIE59078.2023.10334686.

[31] D. Talukder and F. Jahara, "Real-Time Bangla Sign Language Detection with Sentence and Speech Generation," in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, 2020, https://doi.org/10.1109/ICCIT51783.2020.9392693.

AUTHOR BIOGRAPHY

Kartika Candra Kirana is a lecturer at the State University of Malang. She received his S.Pd., M.Kom. degrees in Universitas Negeri Malang, and Institut Teknologi Sepuluh Nopember (ITS), respectively. Her research interests include computer vision, artificial intelligence, and machine learning. She received research funding on machine learning material in 2024.
Google Scholar : https://scholar.google.com/citations?hl=en&user=s8KMcNoAAAAJ
Scopus : https://www.scopus.com/authid/detail.uri?authorId=57205442175

Salah Abdullah Khalil Abdulrahman is the second author is an international student at the International State University of Malang who is involved in the First Author's research and learning.