IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 4, December 2024, pp. 3951~3961
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i4.pp3951-3961

Fog and rain augmentation for license plate recognition in tropical country environment

Vriza Wahyu Saputra, Nanik Suciati, Chastine Fatichah
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Feb 1, 2024
Revised Apr 24, 2024
Accepted Jun 1, 2024

Keywords:

License plate recognition
Permutated autoregressive sequence
Scene text recognition
TPS-ResNet-BiLSTM-attention
Weather augmentation

ABSTRACT

Automatic license plate recognition (ALPR) is a critical component in modern traffic management systems. However, ALPR systems often face challenges in accurately recognizing license plates under adverse weather conditions, such as fog and rain, prevalent in tropical regions. Deep learning ALPR models necessitate huge and diverse datasets for robustness, but data availability remains a concern since unpredictable fog and rain patterns hinder data collection. In this study, we address the issue of enhancing ALPR's robustness by introducing a novel augmentation strategy that combines traditional and weather augmentation techniques. By augmenting the dataset with weather-induced variations, we aim to improve the generalization capability of ALPR models, enabling them to handle a wider range of weather-related challenges. We also investigate the synergy between these weather augmentations and established scene text recognition (STR) methods, such as convolutional recurrent neural network (CRNN), TPS-ResNet-BiLSTM-attention (TRBA), autonomous bidirectional iterative scene text recognition (ABINet), vision transformer (ViTSTR), and permutated autoregressive sequence (PARSeq), to determine their impact on recognition accuracy. Experiments using different training data sets show that training data containing a combination of traditional and weather augmentation produces the best accuracy and 1-NED performance compared to training data without augmentation and traditional augmentation only. The average increase accuracy of all STR model is 1.13% with the best increase accuracy of 3.68% using TRBA.

Corresponding Author:

Nanik Suciati
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
Email: nanik@if.its.ac.id

1. INTRODUCTION

Automatic license plate recognition (ALPR) has been widely used in transportation violation management due to its efficiency in validating license plate numbers. Every vehicle must have number plate that aims to differentiate one vehicle from another based on location, possessions, and type of vehicle. ALPR is extracting a vehicle's number plate from a single image into a sequence of characters and numbers. However, the detection and recognition of license plates faces numerous challenges, including variations in illumination, angles of capture, size discrepancies, contrasting background colors, image sharpness, and adverse weather conditions [1], [2]. Particularly in tropical regions, weather elements such as rain and fog can cause the captured image of the license plate to be distorted, thereby impacting the accuracy of the recognition outcomes. As technology continues to advance, researchers and engineers are actively working to develop ALPR systems that can overcome these challenges and ensure reliable license plate recognition even under adverse environmental conditions.

Several studies on license plate detection and recognition from tropical country have had bad results with noise datasets [3]–[5]. This is because the total of training dataset is small and lack of diversity. Deep learning needs extensive data to achieve a robust model. In the other hand, collecting license plate images in foggy and rainy weather is tricky because we cannot determine when fog and rain are coming. Augmentation is a common technique used in deep learning to increase the size of training datasets by constructing new examples through various transformations of the existing data. Several augmentation methods have been used previously to improve the performance of the model in several domains, such as vehicle and construction detection. Traditional augmentation, such as coloring, rotating, flipping, and blurring, was proposed for detecting objects in weather conditions [6]–[8]. Using traditional augmentation can improve the model's accuracy under various weather conditions. However, to produce a more robust model, a specific augmentation technique is required to address weather conditions. Another research was proposed using weather augmentation to improve the model performance under five weather conditions: brightness, darkness, rain, snow, and fog [9]. The experimental result shows that the presented model with weather augmentation surpassed the baseline model without augmentation. Based on the previous augmentation strategy, we propose to use traditional and weather augmentation before recognizing license plates under weather conditions.

ALPR is typically separated into two stages: license plate (LP) detection and LP recognition. LP detection involves identifying the LP area within an input image, while LP recognition involves extracting the alphanumeric characters. For this particular study, our primary focus is on LP recognition. Several studies have been proposed using robust object detection called YOLO in ALPR. Those studies suggested two steps for recognizing license plates: character segmentation and character recognition [10], [11]. They all achieved a recall of more than 90% in LP detection. However, they reached an accuracy lower than 80% because of the character segmentation. Several scene text recognition (STR) methods such as convolutional recurrent neural network (CRNN) [12], thin-plate spline (TPS)-residual network (ResNet)-bidirectional long short-term memory (BiLSTM) attention (TRBA) [13], and vision transformer scene text recognition (ViTSTR) [14] are designed for end-to-end processes without character segmentation [12]–[14]. Those methods achieved more than 90% accuracy in some datasets [15], [16]. The latest state-of-the-art STR methods including autonomous bidirectional iterative (ABINet) [17] and permutated autoregressive sequence (PARSeq) [18] employ a combination of context-free vision and context-aware language models to achieve efficient results. Based on previous statements, we propose using the STR model widely used in recognizing characters. In summary, the main contribution of this paper is as follows: i) enriching the rainy and foggy dataset using the fog and rain layer augmentation in addition to traditional augmentations; ii) collecting datasets comprehensively according to conditions in Indonesia, including day, night, fog, and rain, with different angles and sizes; and iii) recognizing license plates using some STR models, including CRNN, TRBA, ViTSTR, ABINet, and PARSeq.

2. METHOD

In this section, we explain the detail of the proposed method. In general, our proposed method is shown in Figure 1. The input image is the license plate image, and it is discussed in more points in subsection dataset. After that, the augmentation will be processed in order to enrich the license plate image and discussed in more detail in subsection data augmentation. Then, license plate images will be trained with STR methods and discussed in more detail in subsection STR Finally, the output prediction will be shown after processing with STR methods. The evaluation methods we used are accuracy and 1-NED, discussed in subsection evaluation.

2.1. Dataset

Indonesian license plates contain the area code, registration serial number, and serial number. In this study, we use a primary dataset where the dataset was taken using a Samsung S10 camera. We also use secondary datasets where datasets are obtained via web crawling. We take a comprehensive dataset with several considerations, such as: i) environmental conditions include day, night, rain, and fog; ii) different shooting angles and degrees of closeness; and iii) the dataset includes all background colours on Indonesian license plates.

The total data that has been collected is 6,174 license plates. The data has been cropped manually. We divided the dataset into training and testing with a ratio of 80:20. So we got total training data of 4,712 and testing of 1,463. The minimum resolution is 41×35, while the maximum resolution is 3264×1686. For the specific dataset, we distribute the dataset into some categories:

- Table 1 shows the data distribution base on Indonesia's background colour license plate. The dataset obtained is primarily black and yellow, while the rarely obtained colours are green, white, and red. It is hard to found white background colour because that colour is a new rule in Indonesia. For the green background colour, we were also difficult to found that colour because license plates with a green background can only be found in free trade areas.

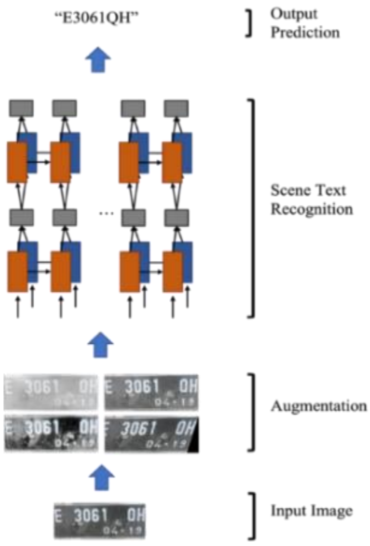
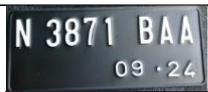
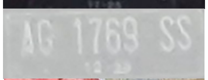
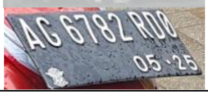


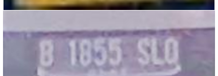
Figure 1. Our proposed method

Table 1. Environment dataset distribution

Environment	Image	Total dataset Train	Test
Day		3,938	985
Night		700	176
Fog		32	131
Rain		42	169

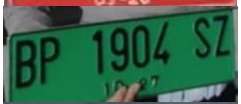
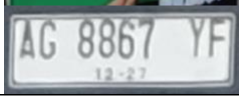
– Table 2 shows the data distribution based on the environment in Indonesia. For specific datasets with foggy and rainy conditions, we have allocated a larger portion of datasets for testing rather than training. This decision is rooted in our utilization of data augmentation techniques primarily for enhancing the training dataset. Furthermore, the scarcity of datasets featuring rainy and foggy conditions has prompted us to prioritize testing data allocation, considering the inherent uncertainty and rarity of these environmental circumstances.

Table 2. Resolution dataset distribution

Environment	Image	Total dataset Train	Test
Small (<128px)		711	171
Medium (128-700px)		3,763	1,218
Large (>700px)		238	173

– Table 3 displays the data distribution based on the dataset's resolution. The cropped image results of the license plates exhibit a wide range of dataset resolutions; hence the dataset distribution holds significant importance for analyzing the accuracy outcomes of the license plate recognition. We divided the dataset into three categories: small, medium, and hard. The small resolution encompasses datasets with resolutions <128px. The medium resolution encompasses datasets with resolutions ranging from 128px to 700px. Lastly, the large resolution pertains to datasets with resolutions >700px.

Table 3. Color dataset distribution

Environment	Image	Total dataset	
		Train	Test
Black		3413	1107
Yellow		1183	314
Red		85	17
Green		11	8
White		20	16

Even though our dataset has complied with all laws and environments in Indonesia, our dataset is still relatively small. Moreover, we recognize the inherent value of incorporating real-world public data that has amassed over time. To tackle this limitation, we draw inspiration from the methodology outlined by [19] and undertake the training of our model using a comprehensive real public dataset. The public datasets that we used including ArT [20], COCO [21], LSVT [22], and Uber [23].

2.2. Data augmentation

At times, models developed during the training process may exhibit inefficiencies. One prominent factor contributing to model inefficiency is overfitting—a circumstance where evaluation metrics during the training phase yield commendable results, yet those observed during the testing phase are comparatively bad. On the other hand, the training process in deep learning requires a substantial amount of data to make the model robust. One way to address overfitting is through augmentation. The objective of data augmentation is to enrich the variability of the training dataset so that the recognition model has more robustness from the dataset in different environments. Data augmentation is also used as an implicit regularization approach to avoid overfitting the standard dataset. We developed an augmentation method inspired by [24], [25]. We use augmentation techniques to enrich the training dataset, including auto contrast, histogram equalization, rotation, shear x, shear y, enhanced sharpness, color increasing, and brightness increasing. We also use weather augmentations such as rain and fog layers to suit the environment in Indonesia. Table 4 shows the result of the augmentation method.

$$Fog = \sum_{i=0}^{w}\sum_{j=0}^{h}\sum_{k=0}^{c}\left(1 - Alpha_{ij1}\right) \cdot Image_{ijk} + Alpha_{ij1} \cdot Intensity_{ij1} \tag{1}$$

$$Intensity = \sum_{i=0}^{w}\sum_{j=0}^{h}\left(\mathrm{I}_{\mu} + \mathrm{I}\_coarse_{ij}\right) + \left(\mathrm{I}_{\mu} * \left(2 \cdot \mathrm{I}\_fine_{ij} - 1\right)/5\right) \tag{2}$$

$$Alpha = \sum_{i=0}^{w}\sum_{j=0}^{h}\left(\alpha_{min} + \alpha_{mul} \cdot \alpha\_mask_{ij}\right)^{\gamma} \cdot \delta_{mul} \tag{3}$$

In (2), I_coarse is intensity coarse built by Gaussian distribution with constant width, height, I_{μ}, and I_{scale}. Width dan height is initialized by width dan height from the image, while I_{μ} is the mean intensity of the fog, and I_{scale} is the standard deviation of the Gaussian distribution used to add more localized power to the mean intensity. I_fine generated noise of varying frequencies with constant width, height, I_{μ} and I_{exp}. I_{exp} is the exponent of the frequency noise used to add fine intensity to the mean intensity. The result from Intensity is resized to produce one channel so fog layer in (1) can calculate it. In (3), Alpha generates noise of varying frequencies and then saves in variable α_mask. α_{min} is used to blend cloud noise with the image. High values will lead to fog being everywhere. α_{mul} is the multiplier for the sampled alpha values. High values will lead to denser fog wherever they are visible. γ or sparsity is the exponent applied late to the α_mask. Lower values will lead to coarser fog patterns, and higher values will lead to finer patterns. δ_{mul} or density multiplier is the late multiplier for the α_mask (similar to α_{mul}). The result of Alpha is resized by one channel so that fog layer in (1) can calculate it.

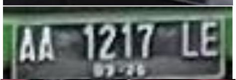
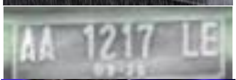
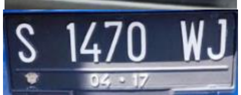
$$Rain = \sum_{i=0}^{w}\sum_{j=0}^{h}\sum_{k=0}^{c}\left(1 - MotionBlur_{ijk}\right) \cdot Image_{ijk} + MotionBlur_{ijk} \cdot \mu_{color} \tag{4}$$

$$MotionBlur = Noise \cdot Kernel \tag{5}$$

$$\mu_{color} = 110 + (240 - 110)\%(\sum_{i=0}^{1000} Flat(MotionBlur)_i) \tag{6}$$

In (4), rain layer is constructed using motion blur and convolved with noise. The motion blur is produced by adjusting the k, angle, and direction parameters. The k parameter is determined by the speed of the falling rain, while the angle represents the degree of motion blur applied to the rain. An angle of 0 produces motion blur that point's straight upwards. The direction parameter controls the forward or backward movement of the motion blur. Direction values towards -1.0 result in motion blur towards the back, while values towards 1.0 produce motion blur forward. μ_{color} or mean color in (6) is generated based on the noise and is a pseudo-random method that eliminates the need for a random state parameter.

Table 4. Image augmentation

Name	Original	Augmentation
Rain layer		
Fog layer		
Brightness increasing		
Color increasing		
Auto contrast		
Rotate		
Shear Y		
Shear X		
Enhance sharpness		
Histrogram equalization		

2.3. Scene text recognition

Within this section, we expound upon methodologies rooted in STR, including but not limited to CRNN, TRBA, ABINet, ViTSTR, and PARSeq. These STR-based methods have emerged as the model of advancement within the domain, having achieved the standard of excellence, often referred to as the state of the art (SOTA). CRNN [12] an end-to-end recognition system consisting of a convolutional neural network

(CNN), BiLSTM, and connectionist temporal classification (CTC) loss. The convolutional layer is constructed through convolution and max pooling operations, omitting the incorporation of a fully connected layer. This layer is dedicated to the extraction of sequential feature representations from input images. It is within this stratum that CNNs acquire insightful knowledge directly from the input images in a representative manner, thereby obviating the necessity for preprocessing steps such as binarization, segmentation, or component localization. The next layer is recurrent layer. The long short-term memory (LSTM) architecture is meticulously crafted to tackle the difficuly of vanishing gradients. This issue entails the constriction of the temporal context range that can be retained, consequently introducing a burden to the training process. LSTM, by design, addresses this concern by controlling the memory retention over extended sequences. LSTM is inherently directed in nature, primarily harnessing the contextual information from the past instances. Nevertheless, in sequences predicated upon imagery, insights from both past and future directions serve a meaningful purpose and mutually complement one another. To address this, the layer at hand employs the BiLSTM methodology, which adeptly incorporates insights from both temporal directions, enhancing the model's understanding of contextual intricacies. CTC loss is part of the transcription layer and translates per-frame predictions by the recurrent layer into a sequence of labels.

TRBA [13] divides the method into four stages: transformation, feature extraction, sequence modelling, and prediction. TPS transformation works to normalize the perspective or curved text into flat text. License plate images within real-world environments exhibit a range of forms, as evident from instances of curved and tilted text. When such input images are directly introduced, the ensuing phase of feature extraction necessitates the acquisition of an unvarying representation regarding these geometric variations. To alleviate this demand, a modification of the spatial transformation network (STN) named the TPS transformation has been employed. The TPS transformation offers a versatile solution to accommodate the various aspect ratios inherent to text lines. This technique leverages a seamless spline interpolation procedure between a defined set of reference points. Feature extraction using ResNet works for extracting visual feature representations using CNN. ResNet stands as a CNN incorporating residual connections, which in turn simplifies the training process of CNNs possessing a greater depth. Sequence modelling using BiLSTM captures contextual information within a character sequence, enhancing the subsequent prediction of each character by making it more robust, in contrast to treating character predictions as independent entities. Prediction layer using attention works to predict the character sequence from the contextual features.

ViTSTR [14] which is a simple single-stage model architecture in the STR model. The architecture of ViTSTR uses vision transformer with a data-effecient image transformer (DeiT) model weight to achieve simplicity and efficiency. ViTSTR converts the input image into patches. These patches are then converted into 1D vector embeddings, which are essentially flattened 2D patches. For the encoder's input, a patch embedding that can be learned is combined with a position encoding for each embedding. The resulting vector sum is the input to the encoder. The output vector corresponds to the [GO] token and extracts multiple feature vectors from the encoder. The layers of the encoder consist of layer normalization (LN), multi-head self-attention layer (MSA), and multilayer perceptron (MLP).

ABINet [17] the architecture ABINet is comprised of three phases. Firstly, it introduces an autonomous mechanism to prevent the flow of gradients between vision and language models, thereby emphasizing explicit language modeling. Secondly, it presents a new bidirectional cloze network (BCN) as the language model, utilizing bidirectional feature representation. Thirdly, it suggests an iterative correction approach for the language model's execution, effectively reducing the influence of noisy input. Moreover, by combining iterative predictions, ABINet presents a self-training technique that enables effective learning from unlabelled images.

PARSeq [18] is a unified STR model with a simple structure, and is capable of making inferences in both context-free and context-aware. PARSeq also uses iterative refinement using bidirectional (cloze) context. Context-free methods for STR directly generate character predictions based on image features. In contrast, context-aware methods leverage learned semantics from the data to enhance the recognition process. The architecture of PARSeq is encoder-decoder, a commonly used approach for sequence modeling tasks. The encoder has 12 layers, whereas the decoder is designed with a single layer. This deliberate deep-shallow configuration is chosen to reduce the computational demands of the model without significantly sacrificing performance.

2.4. Evaluation

We evaluate the proposed model using two evaluation methods. First, we use accuracy to measure the difference between the prediction result and the ground truth. Accuracy aims to determine what percentage of the predicted test data is correct. The second evaluation method is normalized edit distance (NED) [26]. NED comes from the Levenshtein distance, which went through a normalization process. 1-NED is the opposite of NED, in which a high score indicates that two words are similar. Our goal in using 1-NED is to determine the number of corrected predicted characters overall. The levenhstein distance and 1-NED formulas are shown in (8) and (9).

$$LD = \begin{cases} lev_{a,b}(i,j) \\ max(i,j) & min(i,j) = 0, \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1) \\ \quad +1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases} \tag{8}$$

$$1 - NED = 1 - \frac{1}{N}\sum_{i=1}^{N} lev_{a,b}/max(a,b) \tag{9}$$

where a and b are ground truth and prediction. i and j represent the position of character from a and b.

3. RESULTS AND DISCUSSION

In the experiment, we use a supercomputer with a specification of 128 GB of memory, VGA NVIDIA Geforce RTX 3080, and SSD 1TB. In the training phase, we train using STR methods such as CRNN [12], TRBA [13], ABINet [17], ViTSTR [14], and PARSeq [18]. We compose the dataset into three types: no aug, aug, and aug+weather. No Aug means that we train the model without augmentation. Aug is the traditional augmentation process, including brightness increasing, color increasing, auto contrast, rotation, shear X, shear Y, enhanced sharpness, and histogram equalization. Aug+weather is the traditional augmentation mentioned before, and weather augmentation includes a rain layer and a fog layer. Overall, fifteen training processes were carried out, each with a different combination of convolution layers and datasets. Our strategy in the training process is that in the first 20 epochs, we use a scene text public real dataset without a license plate dataset. Then, we train the model formed using the license plate dataset on epochs 21-100. The training process for each model takes about one day.

Table 5 shows the accuracy of each method. We can see that all of the augmentation + weather process techniques obtained the highest accuracy rather than no augmentation and traditional augmentation. PARSeq gets the best accuracy results with a value of 90.56%. Meanwhile, the CRNN method without augmentation dataset produces the lowest accuracy of 68.40%. In PARSeq, we can see that accuracy between dataset Aug and Aug+weather are and 90.56%, respectively. The increase accuracy is 0.27%. In comparison, TRBA obtains the accuracy of each dataset Aug and Aug+weather are 85.16% and 88.84%, respectively. The increase accuracy is 3.68%. The average of increase accuracy for all STR method is 1.32%. From the statement before, the weather augmentation process is impactful in making the model more robust.

Table 5. Accuracy for each method

Method	Accuracy (%)		
	No Aug	Aug	Aug+weather
CRNN	68.19	81.60	82.69
TRBA	83.65	85.16	88.84
ViTSTR	83.38	87.76	88.10
ABINet	86.11	90.01	90.29
PARSeq	88.99	90.29	90.56

In terms of 1-NED, we can see that Table 6 shows the best 1-NED is 98.41% in PARSeq. Meanwhile, the CRNN method without augmentation dataset produces the lowest 1-NED of 92.67%. The augmentation + weather process still has the best 1-NED rather than no augmentation and traditional augmentation. Nevertheless, the methods and augmentation process have more than 90% in 1-NED. It means that the 1-NED scores indicate that the models perform very well, with over 90% of the characters in the license plate test data can be recognized correctly.

Table 6. 1-NED for each method

Method	Accuracy (%)		
	No Aug	Aug	Aug+weather
CRNN	68.19	81.60	82.69
TRBA	83.65	85.16	88.84
ViTSTR	83.38	87.76	88.10
ABINet	86.11	90.01	90.29
PARSeq	88.99	90.29	90.56

Tables 7 and 8 present the accuracy and 1-NED scores on four different environments, day, night, fog, and rain, to show the performance improvement caused by weather augmentation. We use this model because TRBA obtain the best increase accuracy of 3.68%. Based on Table 7, the model's accuracy using no augmentation, augmentation, and augmentation + weather datasets in a fog environment are 84.85%, 93.18%, and 95.45%, respectively. Besides, in a rainy environment, the accuracies on no augmentation, augmentation, and augmentation+weather datasets are 92.90%, 95.86%, and 97.04%, respectively. The accuracy improvement from no augmentation dataset to an augmentation dataset is around 1-8%. The improvement is approximately 1-2% from augmentation to augmentation+weather datasets. It indicates that augmentation is a compelling process to increase accuracy significantly. Moreover, adding augmentation techniques to create license plate images with rain and fog effects to the traditional augmentation techniques also improves accuracy in foggy and rainy environments. In addition, other environments, such as day and night, also improve accuracy in both augmentation and augmentation+weather.

Table 7. Accuracy for each environment on TRBA

Method	Accuracy (%)		
	No aug	Aug	Aug + weather
Day	83.86	84.06	86.90
Night	72.73	75.00	81.82
Fog	84.85	93.18	95.45
Rain	92.90	95.86	97.04

Table 8. 1-NED for each environment on TRBA

Method	Accuracy (%)		
	No Aug	Aug	Aug + Weather
Day	96.97	97.25	97.78
Night	94.50	95.36	96.73
Fog	97.86	99.12	99.43
Rain	98.62	98.91	99.06

Tables 9 and 10 provide insights into the accuracy and 1-NED results for the dataset categorized by color. When observing the accuracy values, it becomes apparent that augmentation + weather consistently yields better results across almost all colors. However, an interesting pattern emerges in the case of the red background dataset, where the best performance is achieved without augmentation. This observation implies that augmentation coupled with weather conditions contributes significantly to enhancing license plate recognition, particularly when faced with diverse background colors. Shifting focus to the 1-NED metric, a parallel trend can be observed with the accuracy outcomes. The correlation between accuracy and 1-NED is consistent, reinforcing the notion that improvements in accuracy correspond to proportional enhancements in 1-NED values. This alignment between accuracy and 1-NED across the various color categories further underscores the reliability and effectiveness of the augmentation and weather-based techniques employed in the license plate recognition process.

Table 9. Accuracy for each background color on TRBA

Method	Accuracy (%)		
	No aug	Aug	Aug+weather
Black	85.20	87.17	89.33
Yellow	79.87	80.83	86.58
Red	70.59	47.06	52.94
Green	60.00	60.00	70.00
White	68.75	75.00	87.50

Table 10. 1-NED for each background color on TRBA

Method	Accuracy (%)		
	No aug	Aug	Aug+weather
Black	85.20	87.17	89.33
Yellow	79.87	80.83	86.58
Red	70.59	47.06	52.94
Green	60.00	60.00	70.00
White	68.75	75.00	87.50

We categorize the dataset into different resolutions, such as small, medium, and large. Tables 11 and 12 show that the augmentation + weather process is still the best in accuracy and 1-NED. The aim was to ascertain the impact of image resolution on accuracy levels and 1-NED. In the small category, with processes involving no augmentation, augmentation, and augmentation+weather, the accuracy results obtained were 77.19%, 73.86%, and 81.29%, respectively. Meanwhile, in the medium category, the accuracy results for processes without augmentation, with augmentation, and with augmentation+weather were 84.48%, 86.45%, and 89.33%, respectively. From these two categories, it can be observed that the augmentation process significantly influences accuracy results. In contrast, for the large category, there was a decrease in accuracy with the augmentation + weather process, specifically to 86.30%. The highest accuracy was found in the large category with the augmentation process, at 90.41%. From these results, it can be concluded that augmentation+weather has a limited effect on the results of the large category dataset, whereas the augmentation process consistently improves accuracy results in each category. This implies that the higher the image resolution, the more directly proportional the results are to its accuracy.

Table 11. Accuracy for each resolution on TRBA

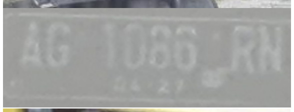
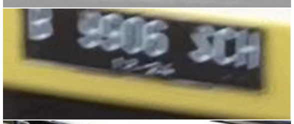
Method	Accuracy (%)		
	No Aug	Aug	Aug+weather
Small	77.19	73.86	81.29
Medium	84.48	86.45	89.33
Large	84.93	90.41	86.30

Table 12. 1-NED for each resolution on TRBA

Method	Accuracy (%)		
	No Aug	Aug	Aug+weather
Small	95.96	95.89	96.80
Medium	97.11	97.58	98.19
Large	96.51	97.60	96.67

The specific datasets in Table 13 have several challenges in predicting characters because the datasets are diverse and under the environment in Indonesia. For example, there is a dataset at night with minimal lighting conditions. The model is accurately predicted on the license plate "B7615TGD" with 100% 1-NED. In terms of weather condition, the license plate "AG2765RBS" and "AG1086RN" in rainy and foggy condition has successfully predicted with 100% 1-NED. The next challenge is a license plate with nails within the digits. For example, the model predicts only the license plate "E50780F" with 1-NED, which is 87.50%. Blurred license plates are also a challenge in this study. The model successfully predicted the blurred image on the license plate "B9906SCH" with 100% 1-NED. Lastly, images of license plates with a particular slope, such as "B1545CL," only produce 1-NED, which is 87.50%. The number plate has a truncated character at the end of the image. Nevertheless, the model instead predicts the truncated character with the C character.

Table 13. 1-NED for specific license plate in different angle, illumination, weather, and resolution

Image	Ground truth	TRBA pred	Image 1-NED (%)
	B7615TGD	B7615TGD	100
	AG2765RBS	AG2765RBS	100
	AG1086RN	AG1086RN	100
	B9906SCH	B9906SCH	100
	B1545CL	B1545CLC	87.50
	E5078QF	E6078QF	87.50

4. CONCLUSION

We propose a combination of weather and traditional augmentation to augment the Indonesian license plate images dataset and improve the performance of the license plate recognition model. We experimented using some widely used STR methods. Based on the experiment results, we draw several conclusions: i) the weather augmentation improves accuracy and 1-NED performance for all STR models. The TRBA method achieved the highest increase in accuracy of 3.68%; ii) the PARSeq method obtained the best accuracy and 1-NED in recognizing license plate characters, respectively 90.56% and 98.41%; and iii) the small and medium image resolution affects the weather augmentation process and, in turn, also influences the model performance. In future research, we suggest exploring an image enhancement process to reconstruct the small-resolution image and enhance the blurred image. The enhanced input image is expected to increase the recognition performance of the license plate model.

ACKNOWLEDGEMENTS

This paper is supported by a research grant from the Department of Informatics Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, 2023.

REFERENCES

[1] A. R. -Alvarez, J. D. A. -Suarez, M. G. -Rodriguez, D. F. -Lanvin, and B. L. Pérez, "Effects of challenging weather and illumination on learning-based license plate detection in noncontrolled environments," *Scientific Programming*, vol. 2019, no. 1, 2019, doi: 10.1155/2019/6897345.

[2] M. A. Ibrahim, K. El-Saady, and M. A. Kenk, "Challenging weather for license plate detection: impaired visibility," in *5th Novel Intelligent and Leading Emerging Sciences Conference*, 2023, pp. 285–290, doi: 10.1109/NILES59815.2023.10296738.

[3] H. Imaduddin, M. K. Anwar, M. I. Perdana, I. A. Sulistijono, and A. Risnumawan, "Indonesian vehicle license plate number detection using deep convolutional neural network," in *International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC*, 2018, pp. 158–163, doi: 10.1109/KCIC.2018.8628488.

[4] A. T. Musaddid, A. Bejo, and R. Hidayat, "Improvement of character segmentation for indonesian license plate recognition algorithm using CNN," in *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI*, 2019, pp. 279–283, doi: 10.1109/ISRITI48646.2019.9034614.

[5] I. W. Notonogoro, Jondri, and A. Arifianto, "Indonesian license plate recognition using convolutional neural network," in *2018 6th International Conference on Information and Communication Technology, ICoICT*, 2018, pp. 366–369, doi: 10.1109/ICoICT.2018.8528761.

[6] R. Walambe, A. Marathe, K. Kotecha, and G. Ghinea, "Lightweight object detection ensemble framework for autonomous vehicles in challenging weather conditions," *Computational Intelligence and Neuroscience*, vol. 2021, 2021, doi: 10.1155/2021/5278820.

[7] Q. V. Hoang, T. H. Le, and S. C. Huang, "Data augmentation for improving ssd performance in rainy weather conditions," in *2020 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-Taiwan*, 2020, pp. 1–2, doi: 10.1109/ICCE-Taiwan49838.2020.9258127.

[8] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 4, pp. 593–597, 2019, doi: 10.1109/LGRS.2018.2878773.

[9] K. S. Kang, Y. W. Cho, K. H. Jin, Y. B. Kim, and H. G. Ryu, "Application of one-stage instance segmentation with weather conditions in surveillance cameras at construction sites," *Automation in Construction*, vol. 133, 2022, doi: 10.1016/j.autcon.2021.104034.

[10] Hendry and R. C. Chen, "Automatic license plate recognition via sliding-window darknet-YOLO deep learning," *Image and Vision Computing*, vol. 87, pp. 47–56, 2019, doi: 10.1016/j.imavis.2019.04.007.

[11] R. Laroca *et al.*, "A robust real-time automatic license plate recognition based on the YOLO detector," in *Proceedings of the International Joint Conference on Neural Networks*, 2018, vol. 2018, pp. 1–10, doi: 10.1109/IJCNN.2018.8489629.

[12] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017, doi: 10.1109/TPAMI.2016.2646371.

[13] J. Baek *et al.*, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4714–4722, doi: 10.1109/ICCV.2019.00481.

[14] R. Atienza, "Vision transformer for fast and efficient scene text recognition," in *Document Analysis and Recognition – ICDAR 2021*, 2021, pp. 319–334.

[15] R. Laroca, E. V. Cardoso, D. R. Lucio, V. Estevam, and D. Menotti, "On the cross-dataset generalization in license plate recognition," in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2022, vol. 5, pp. 166–178, doi: 10.5220/0010846800003124.

[16] S. Agrawal and K. D. Joshi, "Indian commercial truck license plate detection and recognition for weighbridge automation," in *2022 28th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, Nanjing, China, 2022, doi: 10.1109/M2VIP55626.2022.10041077.

[17] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: autonomous, bidirectional, and iterative language modeling for scene text recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7094–7103, doi: 10.1109/CVPR46437.2021.00702.

[18] D. Bautista and R. Atienza, "Scene text recognition with permuted autoregressive sequence models," *Computer Vision – ECCV*, pp. 178–196, 2022, doi: 10.1007/978-3-031-19815-1_11.

[19] J. Baek, Y. Matsui, and K. Aizawa, "What if we only use real datasets for scene text recognition? Toward scene text recognition with fewer labels," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3112–3121, doi: 10.1109/CVPR46437.2021.00313.

[20] C. K. Chng *et al.*, “ICDAR2019 robust reading challenge on arbitrary-shaped text (RRC-ArT),” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2019, pp. 1571–1576, doi: 10.1109/ICDAR.2019.00252.
[21] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, “COCO-text: dataset and benchmark for text detection and recognition in natural images,” *arXiv-Computer Science,* 2016, doi: 10.48550/arXiv.1601.07140.
[22] Y. Sun *et al.*, “ICDAR 2019 competition on large-scale street view text with partial labeling-RRC-LSVT,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2019, pp. 1557–1562, doi: 10.1109/ICDAR.2019.00250.
[23] Y. Zhang, L. Gueguen, I. Zharkov, P. Zhang, K. Seifert, and B. Kadlec, “Uber-text: a large-scale dataset for optical character recognition from street-level imagery,” *SUNW: Scene Understanding Workshop-CVPR*, 2017.
[24] A. Jung, “IMG aug,” *GitHub*, 2020, Accessed: Feb. 15, 2023. [Online]. Available: https://github.com/aleju/imgaug
[25] U. Saxena, “Automold road augmentation library,” *GitHub*, 2018, Accessed: Feb. 15, 2023. [Online]. Available: https://github.com/UjjwalSaxena/Automold--Road-Augmentation-Library.
[26] A. Marzal and E. Vidal, “Computation of normalized edit distance and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 926–932, 1993, doi: 10.1109/34.232078.

BIOGRAPHIES OF AUTHORS

Vriza Wahyu Saputra received his Master degree at Informatics Engineering Master Program, Department of Informatis in Institut Teknologi Sepuluh Nopember (ITS) Indonesia in 2023. He also received Bachelor Degree from Brawijaya University, Indonesia in 2019. The proposed thesis is in computer vision field with the title "the influence of weather augmentation and image preprocessing on license plate detection and recognition using YOLOv7 and sequential based recognition". He became a teacher assistant while a master's student with a computational intelligence course. He can be contacted at email: vrizawahyu22@gmail.com.

Nanik Suciati held a doctorate in Information Engineering from Hiroshima University, Japan in 2010. She received her Bachelor's degree from Institut Teknologi Sepuluh Nopember (ITS) in 1994 and Master's degree from the University of Indonesia (UI) in 1998. She is currently a Professor in the Department of Informatics, ITS. She also served as Head of the Intelligent Computing and Vision Laboratory since 2015 to present. Her research interests are computer vision, computer graphics, computational intelligence, and machine learning. She can be contacted at email: nanik@if.its.ac.id.

Chastine Fatichah held Bachelor degree from Institut Teknologi Sepuluh Nopember (ITS) in 2000, Master degree from University of Indonesia in 2008, and Doctor degree from Tokyo Institute of Technology, Japan in 2012. She is currently Professor of Computer Science at Department of Informatics, ITS since 2023. She also served as Head of the Department of Informatics, ITS, Surabaya, Indonesia in 2019 to present period. Her research includes artificial intelligence, computational intelligence, data mining, and image processing. She can be contacted at email: chastine@if.its.ac.id.