JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 8 Number 2 | December 2023: 169-176
DOI: 10.15575/join.v8i2.1073

The Impact of Data Augmentation Techniques on the Recognition of Script Images in Deep Learning Models

Wulan Sapitri[1], Yesi Novaria Kunang[2*], Ilman Zuhri Yadi[3], Mahmud[4]
[1,2,3] Intelligent Systems Research Group, Faculty of Sains Technology, Universitas Bina Darma
[1,2,3] Information Systems Department, Faculty of Science and Technology, Universitas Bina Darma
[4]Software Developer, OXY creative. Inc, Jakarta, Indonesia

Article Info

Article history:

Received May 23, 2023
Revised September 14, 2023
Accepted October 06, 2023
Published December 28, 2023

Keywords:

Augmentation
CNN
Deep Learning
Komering Script
Recognition

ABSTRACT

Deep learning technology is widely used for recognizing character images, including various regional characters and diverse ancient scripts. Deep learning models require large data sets to recognize images accurately. However, creating a dataset has limitations in terms of quantity, including the Komering script dataset used in this study. Data augmentation techniques can be applied to expand the dataset by modifying existing images to increase data diversity. This study aims to investigate the impact of augmentation techniques on the performance of deep learning models in the case of Komering script recognition. The dataset consists of 500 images for five classes of Komering script characters. Three augmentation techniques, namely random rotation, height shift, and width shift, were applied to the five characters, which were then used to test the model trained to recognize characters in the Komering dataset. This research contributes to providing insights into the effect of augmentation techniques on robust confidence prediction of deep learning models for recognizing newly augmented data. The results demonstrate that the deep learning model can recognize modified data using augmentation techniques with an average accuracy of 80.05%.

Corresponding Author:

Yesi Novaria Kunang,
Intelligent Systems Research Group, Faculty of Science and Technology, Universitas Bina Darma
Jl. A. Yani no. 3 Plaju Palembang
Email: yesinovariakunang@binadarma.ac.id

1. INTRODUCTION

In a local language, some characters are symbols used to write the language on various materials such as paper, leaves, and more. One example of a native regional script in South Sumatra is the Komering language, which has a Komering script. With time, the use of the Komering script and several other ancient scripts has become increasingly rare and has been replaced by the Latin script. As a result, most of the local population from the region of the script's origin no longer recognizes these ancient scripts. Therefore, efforts are needed to reintroduce knowledge about ancient regional scripts like the Komering script so that these regional scripts can be recognized and preserved from extinction.

To preserve ancient scripts like the Komering script some researchers have used technology to preserve the Komering script, such as deep learning [1], [2]. However, one of the challenges of a deep learning model is that it is trained to recognize images with a limited amount of data. In addition, other factors that need to be considered in image recognition, including characters, are when there is a shift or tilt in the image during capture [3], [4], requiring augmentation to modify diversity and increase the number of image data.

Data augmentation effectively increases the amount of data and its random variations [5]. The augmentation process enhances and expands the available data set. The advantage of this technique is that it can reduce the gap between the training, validation, and testing datasets, making the dataset more representative and resulting in a more accurate model while reducing overfitting [6]–[9]. The fundamental concept of data augmentation is to create a dataset that covers new variations in input while maintaining the correct labels [10]. Compared to other methods, data augmentation can modify input data without altering the network structure [11]. Furthermore, augmentation has been applied in various fields, such as image processing [6], [8], speech recognition [12], text recognition [13], and others.

Traditional data augmentation techniques are generally divided into three categories: spatial transformation or geometric transformation, color space augmentations, and information dropping or random erasing [6], [14]. Several augmentation techniques, including flipping, random rotation, translation (shifting images left, right, up, or down), and cropping, are commonly used in model training development as part of the geometric transformation. Color distortion or color space transformation modifies data by altering brightness, hue, and saturation to manipulate the color of RGB color matrices [15], [16]. On the other hand, the random erasing technique selects a rectangular region in an image, erases its pixels, and replaces them with random values [17]. The geometric transformation is easy to use and can address position bias issues in training data, such as a focus centered on the image frame, which is often a challenge in character images [6].

Several researchers have studied data augmentation in machine learning and deep learning. Wikarta et al. researched the influence of data augmentation [18]. They studied the impact of testing data size and data augmentation on driver detection in vehicles wearing masks. The results showed that adding augmentation techniques successfully improved the detection of the model. The augmentation techniques included zoom, rotation, shear, shifting, and horizontal flips. Sanjaya and Ayub [1[19]9] also conducted previous research on car image recognition using augmentation techniques such as Random Crop, Rotate, and Mixup. Their study showed that a CNN model with ResNet architecture achieved better accuracy. Tumewu et al. [20] also researched the classification of batik patterns using data augmentation techniques. They used techniques such as Scale, Random Erase, Rotation, and Flip to increase the variety of the dataset. The results showed that data augmentation, predominantly random erase and flip augmentation, significantly improved the accuracy when using CNN architectures such as ResNet-18 and ResNet-50.

Furthermore, Fadilah et al. [21] researched using data augmentation to overcome data limitations in Indonesian sign language translation. The augmentation techniques employed included image transformations such as flipping, rotating, and adding Gaussian noise to images. The results showed that augmentation improved the accuracy of the models, which were limited by the available data. Zhong et al. [17] researched the development of data augmentation techniques through random erasing. They modified commonly used data augmentation methods such as random flipping and random cropping. The results showed that the combined augmentation techniques for various architectures improved object detection and person re-identification. Zheng et al.[9] proposed full-stage data augmentation for natural image classification. The data augmentation process was applied to both the training and testing stages. The augmentation during the training process aimed to ensure that the CNN structure captured the information structure of the sample data. It also addressed issues related to the position of the images. On the other hand, data augmentation during the testing process aimed to reduce the dependency of ensemble models on a single network, ensuring consistency in ensemble models during the transfer of domain information. The results demonstrated that full-stage augmentation with techniques such as translation, horizontal flip, rotation, scale transformation, and noise disturbance in both the training and testing stages could improve accuracy on the tested dataset.

From these various studies, it can be observed that data augmentation can impact the accuracy of data in Deep Learning models. Therefore, this study aims to examine the effect of augmentation techniques on the accuracy of a deep learning model for recognizing script images. The challenge of image data augmentation, mainly using geometric transformation techniques, depends heavily on their "safety" context in the application domain [6]. "Safety" in augmentation methods refers to the resulting label after the transformation. Not all geometric transformations can be applied to the script images dataset. For example, flipping techniques may produce different character labels or significantly different images from the character labels. Similarly, the limitations of rotation and transformation values need to be evaluated to ensure robust confidence prediction. Therefore, this research will focus on considering the best augmentation techniques for the script images dataset and their impact on the model's ability to recognize augmented data.

This research is essential to provide insights into the most suitable and effective augmentation techniques for the case of image datasets containing characters. This study can contribute to knowledge in character image recognition and data augmentation techniques. The research findings can assist developers and other researchers in optimizing deep learning models specifically for character image recognition. A better understanding of how augmentation techniques affect deep learning models in the context of script image recognition can aid in developing more accurate and reliable solutions for various applications.

2. METHOD

This research focuses on the impact of augmentation techniques on the accuracy level of deep learning model recognition. The research workflow for studying the effect of augmentation techniques in recognizing ancient script images can be seen in Figure 1.

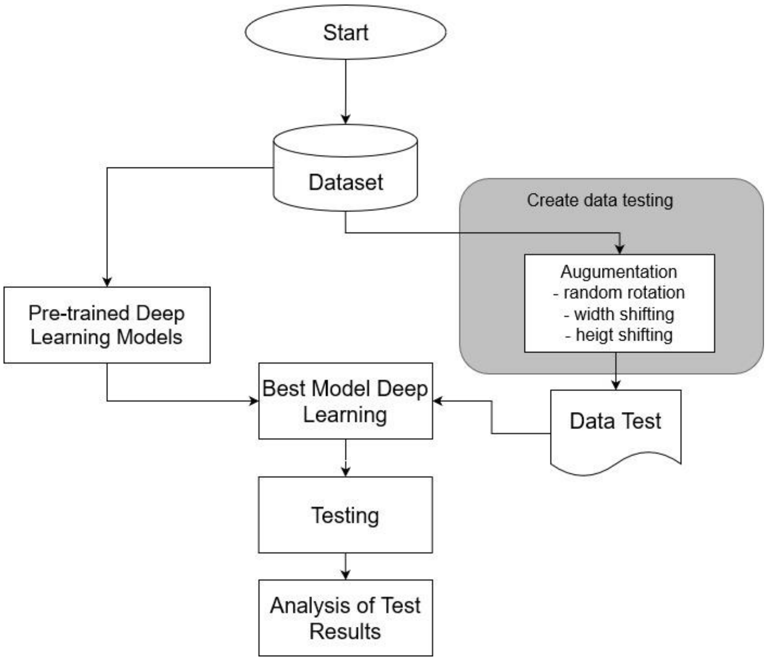

Figure 1. Process flow of augmentation

The stages can be explained as follows:

a. The first stage involves generating testing data using augmentation techniques to create various script image variations. For the dataset, samples of Komering characters were obtained from previous research [2], [22]. This dataset selected five characters (RHU-GHU, SU, TU, WU, and YU), as shown in Figure 2. Each character comprises 100 images, which will be generated to create new test data.
b. Augmentation techniques used to produce new images involve geometric augmentation suitable for a dataset containing characters, taking into account the augmentation process that will not affect label changes due to image distortion magnitude. The characteristics of Komering characters tend to be somewhat similar, mainly due to the use of diacritics. Therefore, basic augmentation techniques such as random rotation and shifting were chosen for creating the testing dataset. The random value limits for these techniques were set to ensure that they do not alter the character's shape. Techniques such as flipping and noise injection were not selected as they could alter the character label readings.
c. Data augmentation results yield a testing dataset consisting of modified script characters using Random rotation and Shift (height and width) augmentation techniques, which will subsequently be used to evaluate the base deep learning model.

d. The deep learning model used here is a CNN-based model previously trained with the Komering script dataset in a prior study [2]. The h5 model from the best model is imported and then tested with the newly generated dataset using augmentation techniques.
e. Lastly, the results are analyzed to determine the impact of Random rotation and Shift (height and width) augmentation techniques on accuracy, precision, recall, and F1 score values.

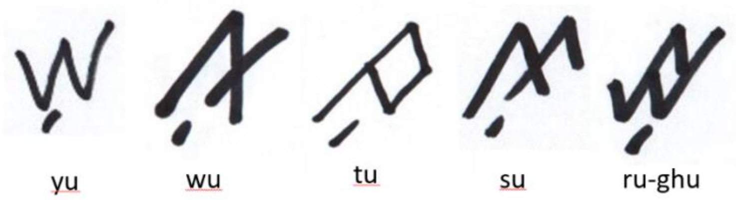

Figure 2. Example of augmented character images

3. RESULT AND DISCUSSION

3.1. Augmentation Process for Script Images

The augmentation process in this study utilizes the imageDataGenerator from the Keras library [23]. The evaluation process of the deep learning model to generate testing data involves three augmentation techniques: random rotation and random transformation shift (height and width), resulting in 14 model datasets for script images. The newly generated data consists of 5 characters: RHU-GHU, SU, TU, WU, and YU. Each of these characters has 100 image data in the dataset. From each character image, an additional 1000 images are generated. Therefore, after augmentation, each character has 10,000 images per model. Consequently, the total of the five augmented character samples in each model amounts to 50,000 images. The explanations for each of the three augmentation techniques are as follows:

1. Test A: Random Rotation for Script Images

In Test A, the random rotation technique generated 50,000 images for each testing data. The evaluation modifies the rotation size with eight different values while keeping the height and width constant. Random rotation is an image processing technique involving the image's random rotation at various angles or directions. This technique aims to increase the variation in the script image data within the testing dataset, allowing us to observe how well the model can recognize objects with different orientations or rotation positions. The random rotation values are modified by providing a range of values (-20, -15, -5, 0, 5, 10, 15, 20), where 0 represents the standard size (Table 1). Negative values indicate rotation to the left, while positive values indicate rotation to the right. The parameters brightness_range=(0.2,1.0), width_shift_range=0.1, and height_shift_range=0.1 are for the random rotation model.

Table 1. Test A: Evaluation of range values for random rotation technique

No	Trial Type	Rotation	Height	Widht
1	Test 1A	20	0.10	0.10
2	Test 2A	15	0.10	0.10
3	Test 3A	10	0.10	0.10
4	Test 4A	5	0.10	0.10
5	Test 5A	0	0.10	0.10
6	Test 6A	-5	0.10	0.10
7	Test 7A	-10	0.10	0.10
8	Test 8A	-15	0.10	0.10

Note: Negative values indicate rotation to the left, while positive values indicate rotation to the right.

2. Test B: Width Shift for Script Images

The second model, named Test B, evaluates the augmentation technique of transformation with width shift. This technique involves horizontally shifting the image to the left or right within a predetermined range. Moving the pixels of the image horizontally without changing its size performs the Shift. The range value is a float number <= 1, representing a percentage of the total width as the range. For the evaluation, the sizes for modifying the character samples with the width shift technique are (0.1, 0.15, 0.20). The range for height value and random rotation remains at 0.1 and 10, respectively, as shown in Table 2. The default brightness_range=(0.2, 1.0) is also maintained.

Table 2. Test B: Evaluation of range values for width shift technique

No	Trial Type	Widht	Height	Rotation
1	Test B	0.1	0.1	10
2	Test B	0.15	0.1	10
3	Test B	0.20	0.1	10

Note: The range value represents a percentage of the total width for the Shift.

3. Test C: Height Shift for Script Images

The following evaluated augmentation technique is the height shift technique (Test C). This technique involves changing the vertical or height position of the image in various sizes or directions. The given random size range for the height shift augmentation technique is (0.1, 0.15, 0.20), while the parameters for random width and random rotation remain the same (Table 3). The brightness range for this test is the same as in Tests A and B, using a range of values (0.2, 1.0).

Table 3. Test C: Evaluation of range values for height shift technique

No	Trial Type	Height	Widht	Rotation
1	Test C	0.1	0.1	10
2	Test C	0.15	0.1	10
3	Test C	0.20	0.1	10

The total number of models in Tests A, B, and C is 14, with each dataset consisting of 50,000 images. Figure 3 shows an example of the results of the augmentation techniques. By observing the image, one can perceive the changes that have occurred in the images after being modified using augmentation techniques.

Figure 3. Example images of applied augmentation techniques

3.2. Testing Deep Learning Model for Augmented Images

The researchers based the deep learning model used in this study on the model utilized in a previous study [2]. This deep learning model employs the Convolutional Neural Network (CNN) algorithm. The deep learning model this research uses consists of 4 CNN modules and 3 Neural Network (NN) layers. The best model from the abovementioned study has been trained to recognize 336 characters of the Komering script and achieved an accuracy of 99%. This best model was tested in the evaluation process to recognize 14 test datasets generated through the augmentation process: Test A, B, and C for the characters RHU-GHU, SU, TU, WU, and YU. Accuracy, precision, recall, and F1-Score were used for evaluation to assess the impact of each augmentation process on the performance of the deep learning model.

Table 4 presents the performance of the deep learning model in recognizing images of the 5 characters that underwent the augmentation process with modified rotation range values. By modifying the augmentation data with a small random rotation range to the left (-) and right (+), the deep learning model still performs very well in recognizing Komering script characters. A rotation of 5 degrees to the left yields better performances than non-rotated images. To highlight the best performance, we have emphasized the corresponding performance values. As the rotation angle increases, the deep learning model finds it slightly more challenging to recognize the script characters. Ideally, the deep learning model can tolerate script inclinations of up to 10 degrees to the left and right. We can observe the model's performance, particularly the F1-Score, which remains above 90%. These values demonstrate that the deep learning model tolerates classifying script characters at various inclinations.

Table 4. Performance of the deep learning model for testing dataset A

Model	Rotation	Accuracy	Precision	Recall	F1-Score
Test1A	20	0.74414	0.98992	0.74414	0.84842
Test2A	15	0.78427	0.99365	0.78427	0.87574
Test3A	10	0.83134	0.99280	0.83134	0.90435
Test4A	5	0.84642	0.98980	0.84642	0.91203
Test5A	0	0.86535	0.99127	0.86535	0.92322
Test6A	-5	**0.88178**	**0.99316**	**0.88178**	**0.93347**
Test7A	-10	0.82808	0.99312	0.82808	0.9024
Test8A	-15	0.8048	0.99427	0.80480	0.88880

Table 5 displays the results of Test B, which involved modifying the data using horizontal shifting techniques to the left and right. The range values from 0 to 0.2 specify the percentage of image shifting based on the image's width. The table illustrates that a maximum width shift of 0.10 or 10% of the image resulted in improved performance compared to no shifting. For the best performance, the images could be accurately recognized with a precision of 99.33%, an accuracy and recall of 83.02%, and an F1-score of 90.36%. As the shifting value increases, the performance decreases.

The results for vertical shifting in Test C can be seen in Table 6. This test involved modifying the height (Shift) range values from 0.1 to 0.2 for vertical shifting. The results demonstrate that the model's performance decreases as the height range increases. However, the precision value remains above 90%, indicating that the model can still recognize character images with a tolerance of up to 15% vertical shifting from the original image height. This indicates that larger height ranges for vertical shifting negatively impact the model's performance in correctly classifying the images. However, even with the largest height range (0.20), the precision remains relatively high (above 98%), indicating that the model can still identify positive predictions accurately.

Table 5. Performance of the deep learning model for testing dataset B.

Model	Widht	Accuracy	Precision	Recall	F1-Score
default	0	0.86535	0.99127	0.86535	0.92322
Test1B	0.10	0.83024	0.99330	0.83024	0.90361
Test2B	0.15	0.78573	0.99184	0.78573	0.87561
Test3B	0.20	0.71285	0.99885	0.71285	0.83055

Table 6. Performance of the deep learning model for testing dataset C.

Model	Height	Accuracy	Precision	Recall	F1-Score
default	0	0.86535	0.99127	0.86535	0.92322
Test1C	0.10	0.83479	0.99976	0.83479	0.90886
Test2C	0.15	0.80751	0.98925	0.80751	0.88817
Test3C	0.20	0.73472	0.98085	0.73472	0.83744

3.3. Discussions

To assess the impact of augmentation techniques on the overall accuracy of script image recognition in the deep learning model, Table 7 summarizes the average results of all testing on the dataset using the three augmentation techniques. The overall average value is calculated from each augmentation technique testing result.

Table 7. Average values performances of dataset testing for each augmentation technique

TEST	Accuracy	Precision	Recall	F1-Score
default	0.86535	0.99127	0.86535	0.92322
Test A	0.82327	0.99225	0.82327	0.89856
Test B	0.77627	0.99466	0.77627	0.86992
Test C	0.79234	0.98995	0.79234	0.87816
Average	**0.80045**	**0.99135**	**0.80045**	**0.88411**

This information provides a general overview of the model's performance for each tested augmentation technique. The results show that the random rotation augmentation technique provides a more stable performance than the script images dataset testing generated with vertical and horizontal shifting techniques. The trained deep learning model tolerates classifying characters at various angles, vertical and horizontal shifts. The overall average precision value reached 99.14%, indicating that the deep learning model exhibits a low false positive rate and can correctly recognize character images. Even in Test A and Test B, the precision values increased by 0.09% and 0.05%, respectively, compared to the default dataset. Precision measures the model's ability to identify characters from all the identified objects correctly. In image recognition, a precision value approaching 100% means that almost all the characters identified by the model are correct, with very few false positives (characters wrongly identified as certain characters). The value indicates that the model is excellent at accurately recognizing characters. On the other hand, the average recall value of 80.05% suggests that the deep learning model still has some difficulty recognizing specific character images of undetected characters. The average of both performance measures can be observed in the F1-Score, which approaches 88.41%.

The approach of testing augmented data with the trained deep learning model compared to testing with original images reveals a performance gap in character recognition. Therefore, for the implementation of real-time script image recognition applications, similar to the deep learning model developed in case [2], it is necessary to consider the accuracy of the image positioning. The existing deep learning model has the potential to be embedded into a model capable of reading documents, as in the example of Komering script recognition. Therefore, the geometric augmentation factors that have been applied can serve as a basis for determining the script boundaries during the document segmentation process.

In script image recognition, data augmentation can assist the deep learning model in understanding variations in handwriting, writing styles, sizes, orientations, and image lighting. Augmentation techniques can enhance the model's ability to recognize and accurately classify images. The approach taken by Zheng et al. [9], which involves full-stage augmentation in training and testing data, can be considered for future development. Data augmentation during training can help the deep learning model recognize a more comprehensive range of image variations due to rotation, image position, noise, and others. Meanwhile, augmentation during testing images is essential to evaluate the model's ability to provide more robust predictions.

4. CONCLUSION

Augmentation techniques in this study can be utilized to test the stability of the deep learning model. The test results demonstrate that the deep learning model performs better in recognizing images with random rotations than those modified with horizontal and vertical shifts. However, as the rotation angle increases, the performance of the Komering character recognition model becomes more challenging, as evidenced by the decline in performance. Overall, the deep learning model tested with Test A (Random rotation), Test B (Width), and Test C (Height) exhibits good tolerance in classifying characters at various angles of inclination and image shifts in both vertical and horizontal directions. The model can recognize modified characters in all datasets with an accuracy rate of over 77%, a precision rate of over 98%, and an F1 score of over 86%. For future development, the recognition of Komering characters will be enhanced by employing transfer learning approaches and implementing full-stage augmentation processes on the training and testing data.

REFERENCES

[1] M. Mahmud and Y. N. Kunang, "Pengembangan Aplikasi Pengenalan Aksara Komering Menggunakan Metode Deep Learning Berbasis Android," *J. Comput. Inf. Syst. Ampera*, vol. 1, no. 2, pp. 99–111, May 2020, doi: 10.51519/journalcisa.v1i2.38.

[2] Y. N. Kunang, I. Z. Yadi, Mahmud, and M. Husin, "A New Deep Learning-Based Mobile Application for Komering Character

Recognition," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, May 2022, pp. 294–299. doi: 10.1109/ISRITI56927.2022.10053072.

[3] Y. Sutal, R. K. Pingak, A. S. Ahab, and S. D. Baunsele, "Sistem Identifikasi Nominal Uang Logam Menggunakan Tensorflow dan Convolutional Neural Networkberbasis Raspberry Pi," *Kaji. Awal Ekstraksi Silika Dari Pasir Noeltoko Menggunakan X-Ray Fluoresc.*, no. June, pp. 278–289, 2019, [Online]. Available: https://www.conference.undana.ac.id/sainstek/article/view/84/70

[4] P. R. Nesbit and C. H. Hugenholtz, "Enhancing UAV-SfM 3D model accuracy in high-relief landscapes by incorporating oblique images," *Remote Sens.*, vol. 11, no. 3, pp. 1–24, 2019, doi: 10.3390/rs11030239.

[5] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Autoaugment," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-June, no. Section 3, pp. 113–123, 2019.

[6] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[7] Z. Pei, H. Xu, Y. Zhang, M. Guo, and Y. Yee-Hong, "Face recognition via deep learning using data augmentation based on orthogonal experiments," *Electron.*, vol. 8, no. 10, pp. 1–16, 2019, doi: 10.3390/electronics8101088.

[8] M. Elgendi *et al.*, "The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective," *Front. Med.*, vol. 8, no. March, pp. 1–12, 2021, doi: 10.3389/fmed.2021.629134.

[9] Q. Zheng, M. Yang, X. Tian, N. Jiang, and D. Wang, "A full stage data augmentation method in deep convolutional neural network for natural image classification," *Discret. Dyn. Nat. Soc.*, vol. 2020, 2020, doi: 10.1155/2020/4706576.

[10] Q. Wen *et al.*, "Time Series Data Augmentation for Deep Learning: A Survey," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 4653–4660, 2021, doi: 10.24963/ijcai.2021/631.

[11] M. Kim and C. Y. Jeong, "Label-preserving data augmentation for mobile sensor data," *Multidimens. Syst. Signal Process.*, vol. 32, no. 1, pp. 115–129, 2021, doi: 10.1007/s11045-020-00731-2.

[12] A. Cohen, I. Rimon, E. Aflalo, and H. H. Permuter, "A study on data augmentation in voice anti-spoofing," *Speech Commun.*, vol. 141, pp. 56–67, 2022, doi: 10.1016/j.specom.2022.04.005.

[13] C. Tang, K. Ma, B. Cui, K. Ji, and A. Abraham, "Long text feature extraction network with data augmentation," *Appl. Intell.*, vol. 52, no. 15, pp. 17652–17667, 2022, doi: 10.1007/s10489-022-03185-0.

[14] P. Chen, S. Liu, H. Zhao, and J. Jia, "GridMask Data Augmentation," 2020, [Online]. Available: http://arxiv.org/abs/2001.04086

[15] X. Guo and Q. Hu, "Low-light Image Enhancement via Breaking Down the Darkness," *Int. J. Comput. Vis.*, vol. 131, no. 1, pp. 48–66, 2023, doi: 10.1007/s11263-022-01667-9.

[16] W. Wang, X. Wu, X. Yuan, and Z. Gao, "An Experiment-Based Review of Low-Light Image Enhancement Methods," *IEEE Access*, vol. 8, pp. 87884–87917, 2020, doi: 10.1109/ACCESS.2020.2992749.

[17] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "2017(cites:1625)Random Erasing Data Augmentation," 2017.

[18] A. Wikarta, B. Suryo, and M. Khoirul Effendi, "Analisa Pengaruh Ukuran Testing Data dan Data Augmentation pada Tingkat Akurasi Deteksi Pemakaian Masker oleh Pengemudi Kendaraan menggunakan Deep Learning Influence," *Sent. 2020*, no. November 2020, pp. 20–24, 2020.

[19] J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[20] S. F. Tumewu, D. H. Setiabud, and I. Sugiarto, "Klasifikasi Motif Batik menggunakan metode Deep Convolutional Neural Network dengan Data Augmentation," *J. Infra*, vol. 8, no. 2, pp. 189–194, 2020.

[21] R. Z. Fadillah, A. Irawan, M. Susanty, and I. Artikel, "Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO)," *J. Inform.*, vol. 8, no. 2, pp. 208–214, 2021, [Online]. Available: https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/10768

[22] Y. N. Kunang, "Komering_(Ulu)_Characters_Datases." Intelligent System Research Group Bina Darma Dataset, 2022. [Online]. Available: https://www.kaggle.com/datasets/ykunang/aksara-komering

[23] "tf.keras.preprocessing.image.ImageDataGenerator." Keras, 2022. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator