Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Effects of Data Augmentation on a CNN Model for Baybayin Character Recognition

Luis William C. Meing, Percival N. Cael, Kris Michael D. Miana

Student, College of Information Technology and Computer Science, University of the Cordilleras - Baguio Colleges Foundation, Philippines
Email: louiemeing@gmail.com

In 2018, the Philippine government approved House Bill 1022 which declared the Baybayin script as the official national writing system of the Philippines. In accordance with the bill's mandate to promote, protect, preserve, and conserve the Baybayin script, this study takes an in depth look at the effects of data augmentation (DA) techniques on a Convolutional Neural Network (CNN) model for Baybayin character recognition. The dataset was preprocessed to balance the class distribution then we explored geometric and photometric DA methods to observe its effects on model performance using the YOLOv8 algorithm. The DA techniques were set to small uniform intervals when performing the experiments and then key metrics: precision, recall, F1-Score, and mAP are evaluated. The results demonstrate varying impacts of different DA techniques on model performance, with detailed analyses of rotations, shearing, and noise injections. The study contributes to understanding, promoting, and preserving the Baybayin script through machine learning advancements in learning and using the script.
CCS CONCEPTS
• Computing methodologies → Artificial intelligence → Natural language processing → Phonology/morphology.
Keywords: Baybayin, data augmentation, optical character recognition, computer vision, convolutional neural network, heritage conservation.

1. Introduction

Baybayin is an ancient script from the Philippines, utilized as a system of writing before the Spanish colonization of the archipelago [15]. It was mostly used by Filipinos of Tagalog ethnicity, the largest ethnic group in the Philippines. The script contains fourteen unmarked consonants and three vowels, with the unmarked consonants ending with the short "a" sound. Diacritical marks are used to indicate if the consonants are followed by vowels or not. For instance, when a bar or dot is placed on top of a consonant, this indicates that the consonant ends with the short "e" or "i" sound. At the bottom, a bar or dot would mean the consonant ends with the short "o" or "u" sound, while an "x" or cross mark indicates that the consonant does not have any ending vowel sounds [15][9]. In total, the Baybayin script contains 59 classes shown in Figure 1 where it contains the classic Baybayin script with all the consonants and vowels, their diacritic applied counterparts, as well as the English alphabet translation. In 2018, the Philippine "National Writing System Act" or House Bill 1022 was approved, making Baybayin the official Philippine national writing system to be used as a tool for cultural and economic development. This came about by the need for the preservation of the Philippine cultural legacy like its national writing system to promote awareness, appreciation, respect, as well as pride for its authentic identity [5, 14].

Figure 1: The Baybayin script (see Appendix A)

The technique of adding invariant samples to a dataset by label-preserving alterations is known as data augmentation (DA) [16]. DA can then be mapped as =ST where S represents the original dataset and T is the augmented form of S. The resulting addition of this artificial dataset T to S is then represented by S'=ST where S' contains both the original dataset and its respective transformations defined by [16]. DA is usually employed when there is a lack of readily available data or if there is a need to address the problem of overfitting [10]. The DA techniques employed by the study are geometric and photometric where the methods applied preserve the meaning of the Baybayin characters and not those that altered the meaning or label of the these characters. Geometric methods involve transformations that modify the image's geometry by mapping individual pixel values to new locations [11][16]. Photometric methods alter the color of the image thru the manipulation of the RGB channels by shifting the pixel values (r,g,b) to new pixel values (r',g',b') while preserving the geometry or spatial structure of the image [11][16]. One DA method is Rotation, which is a geometric method that performs image rotation around its center by applying a transformation that maps each pixel (x, y) in the image to proportional to the location of another pixel (x′, y′) defined by the equation below [16]:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Shearing is another geometric method employed for this study and it involves shifting the pixels of an image along a specific direction, leading to a distortion that creates a slanting effect. Noise Injection is the third DA technique employed which is a photometric approach that inhibits overfitting in neural network models [11]. This is achieved by adding salt-and-pepper noise which simply refers to randomly adding black and white dots to the image [8][10]. Noise injection can also be applied using the Gaussian noise method which is a statistical noise that has a probability density function equal to the normal distribution. Gaussian noise method is popular in data science because Gaussian random events are commonly observed [2]. According to Arslan et al. [2], Gaussian noise is applied in two steps: first is to calculate the random noise and then apply that generated noise to the dataset. In summary, the study employed the following DA techniques: two geometric methods (Rotation, Shearing) and one photometric method (Noise Injection).

The discipline of data science has witnessed the rise of technologies such as machine learning algorithms due to the swift advancement of Artificial Intelligence. Machines that are trained to perform these tasks in a human-like manner use these techniques. In machine learning, a convolutional neural network (CNN) is a deep learning algorithm that looks for patterns in photos, videos, or audio in order to identify and categorize specific targets (objects, sounds, humans, etc.). CNN models are among the best utilized for classifying image data [4], therefore using them for character recognition tasks in an image is ideal [12]. The base CNN model, Ultralytics’ You Only Look Once v8 (YOLOv8) is a state-of-the-art object detection model that further improved the previous YOLO versions by introducing new features and upgrades to boost model performance and flexibility. YOLOv8 is designed to be quick, precise, and simple to use, making it a good choice for applications such as object recognition and tracking, instance segmentation, image classification, and posture estimation [17]. Although there is a fair amount of studies tackling the Baybayin script in various fields and domains, it is worth noting that there is a limited number of studies covering the detection of Baybayin scripts using machine learning models. Among those noteworthy is one that detected word-level Baybayin text using the CNN-based real-time object identification model YOLOv3, together with data augmentation approaches. The experiment yielded an accuracy rate of 98.92%, where the researchers observed that illegible or distorted handwriting was the cause of the misclassifications [9]. Another study classified handwritten Baybayin script using Android mobile devices by employing a CNN model based on the MobileNetV2 algorithm. After experimenting with hyperparameter tuning and DA to obtain validation results of 96.02%, the researchers concluded that although the suggested model was functioning, MobileNetV2 was unable to identify the dataset's diacritics [5]. In a related study, recurrent neural network models based on the Long Short-Term Memory method with different configurations were employed that propagated the learning of the Baybayin script to identify Baybayin characters. The best model produced by the experiment had an accuracy of 92.9% [14]. A study evaluating four machine learning models used for Baybayin script recognition found that CNN models perform best when classifying image data [3]. A CNN model for transliterating Baybayin script on text pictures was developed in a different study, which highlighted the difficulties in identifying and comprehending the script. The experiment produced a transliteration accuracy of 92% for word-level Baybayin scripts using a CNN-based Baybayin Optical Character Recognition model that was built with an accuracy of 97.62% [15].

The focus of this study is to contribute to the proliferation and conservation of the Philippine Baybayin script using machine learning and the effects of data augmentation on the performance of the machine learning model. The study focused on answering the questions on what effects does DA have on a CNN model for the Baybayin script, which DA method has the most influence in negatively affecting the metrics of a CNN model for character recognition, and to what extent do DA technique(s) improve the performance of a CNN model for character recognition. There is a need to increase available studies focusing on the Baybayin script in the machine learning domain to propagate knowledge of the script as well as provide insights into the development of these models with data augmentation techniques combined.

2 METHODOLOGIES

To achieve and elicit greater interest in the use and understanding of the script; CNN; and DA, utilized is one of the widely used CNN models in object detection, YOLOv8. With YOLOv8 as the base model, the researchers observed the effects of various DA techniques to provide insights into the effects of these techniques to the performance of the model. The experiment applied DA techniques with varying configurations to a Baybayin script dataset and observed the efficacy of each technique to derive insightful information. To the researchers' knowledge, this study is the first to have examined the effectiveness of DA techniques in optical character recognition of the Baybayin script. Our work can be used as a basis of future experiments and research for those working on applications connected to cultural preservation of old scripts such as Baybayin and effects of data augmentation techniques.

The Baybayin script dataset is sourced from the Roboflow website (roboflow.com), a platform that provides tools and services related to computer vision and image processing. The dataset is titled "Baybayin OCR Computer Vision Project" [1] which has much fewer images than other publicly available Baybayin datasets making it suitable for DA experiments. Roboflow offers multiple versions of the dataset but chosen is "v29 cropped" which is best suited for the DA experiments done in this paper. The said dataset contains the original non-augmented version of the images that are preprocessed by cropping to separate individual characters into images alongside their annotations. Roboflow also offers various formats for this dataset in which YOLOv8 is the selected format. Figure 2 shows the dataset containing 912 preprocessed handwritten Baybayin characters alongside sample images of individual characters.

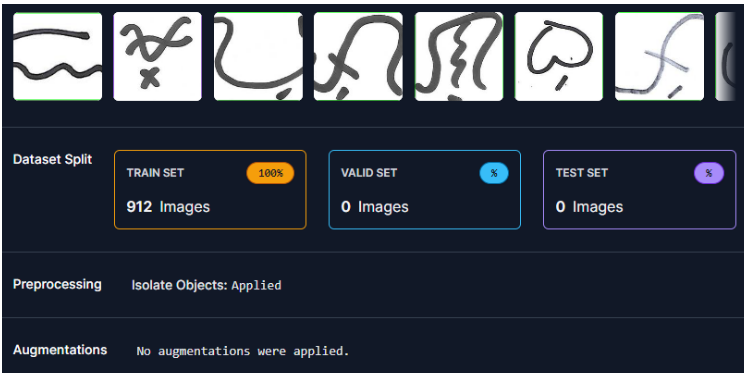

Figure 2: Partial screenshot of the information of the Baybayin OCR Computer Vision Project v29 dataset [1]

In Figure 3, it can be seen that the class distribution of the original dataset is not evenly distributed and needed some preprocessing to eliminate distribution bias. The researchers randomly removed characters from each Baybayin class until the remaining dataset was balanced to contain ten of each character. The resulting dataset is then used to build a training dataset for the experiments with the removed classes compiled and used as a test dataset for the model. To summarize, the initial dataset is split into two parts: a test dataset with 322 photos and a training dataset with 590 images. The training dataset is labeled “original” (Orig) dataset.

The study applies three DA techniques to the Orig dataset separately; rotate, shear, and noise injection. The data augmented datasets are each added to the Orig dataset to create new datasets for the model training. This means each new dataset contains double the images of the Orig dataset: 590 images from Orig plus 590 images of a data augmented copy of the Orig dataset. An example of the new datasets created can be seen in Figure 4 where the images contain another copy of themselves applied with counterclockwise rotation by 5°. The filenames of the Orig dataset are retained while the researchers created corresponding filenames for the transformed datasets by attaching a suffix to the original filename which states the DA technique and the value used. The same method is used for the labels of the images. Seen in Figure 4 is an example where the DA images have a suffix of da_cc_05 where cc means rotate counterclockwise and 05 indicates how much rotation is applied.

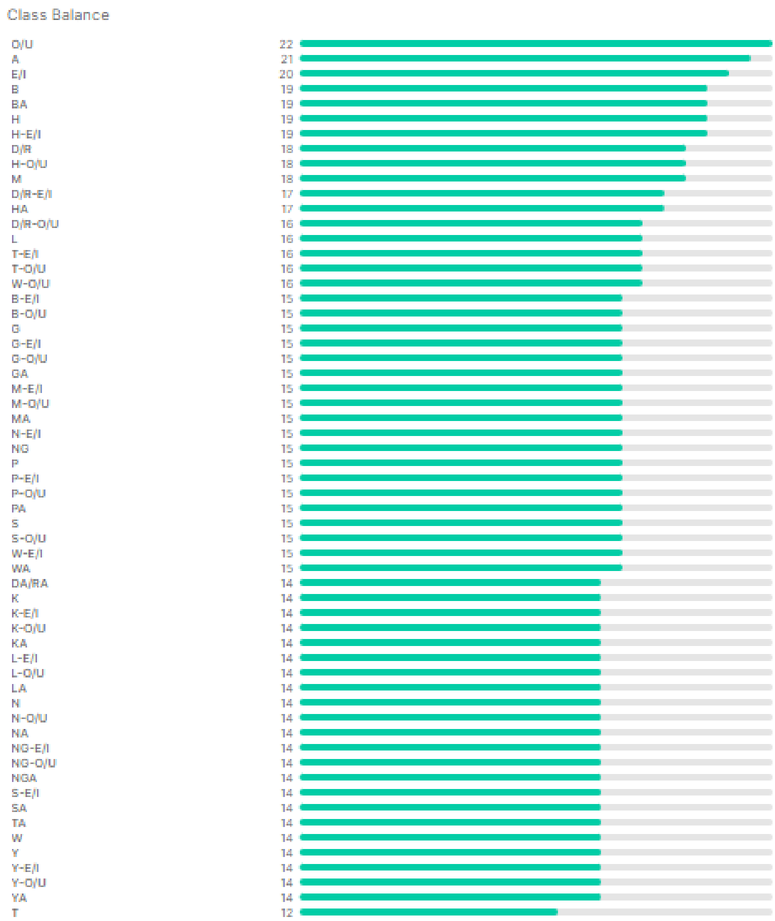

Figure 3: Class distribution of the original dataset [1]

Exploratory experiments on rotation indicated that setting the rotation value to +30° or -30° is large enough to generate new invariant samples [16] but the study performed by A. Mumuni and F. Mumuni [11] said that studies commonly use rotation values between -25° to +25°. To be able to properly observe the rotation effect, the researchers created twelve DA-applied dataset variants of rotation with increments of 5° from -30° to +30°. Studies that performed shearing commonly experimented with values between -20° to +20° [11]. For shearing, the researchers applied it to the images based on the horizontal plane and created eight datasets with increments of 5° from -20° to +20°. Noise injection is applied using two methods, Gaussian noise and salt-and-pepper noise. For Gaussian noise, created are eight datasets from two versions containing varying intensity of noise. The first version is color mode with standard deviation of 25 (weak), 50 (moderate), 75 (strong), and 100 (very strong). The second version of Gaussian noise injection is grayscale mode with similar standard deviations. The mean for all Gaussian noise applications is set to zero. The second noise injection method is salt-and-pepper which is the application of random black and white pixel corruption to an image with varying intensities. Ng and Ma [13] applied four models of salt-and-pepper noise in their study determined by varying density levels (low, medium, and high intensity). This paper applied salt-and-pepper randomly in four intensities, 5 (weak), 10 (moderate), 15 (strong), 20 (very strong). Lastly, the study also uses a special dataset derived from the Orig dataset wherein no DA is applied but instead have duplicated copies where no DA is applied to create observations of the difference from the other DA datasets to make the number of images equal per dataset used.

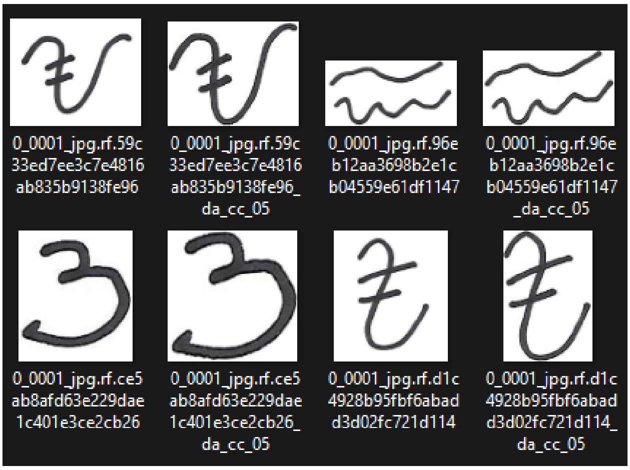

Figure 4: Partial screenshot of a 5° rotation applied dataset

Our model training utilized the YOLOv8 algorithm library by Ultralytics. The training configurations used for the YOLOv8 models can be seen in Table 1.

Table 1: The YOLOv8 configuration

Model Training Configuration	Specification
variant	nano
epochs	100
optimizer	auto
batch	-1 (autobatch)
device	0 (GPU)
other settings	default

YOLOv8 is a cutting-edge picture and object detection tool. Utilized is the fastest but least accurate variant, nano, for this study and left most configurations at their default values because the research concentrated on the effects of the DA techniques and not on improving the YOLOv8 algorithm. The number of training epochs was set to 100, which is a sufficient quantity for the purpose of this study. The optimizer and batch configurations were set to automatic to help with training speed. The device is also configured to use the video card or graphics processing unit (GPU), which cuts the computational cost of model training significantly in terms of time. Most of the model training took almost half an hour to complete.

A new model was trained separately for each DA dataset using the same YOLOv8 configuration. A summary of the separate datasets can be seen in Table 2 showing the specific DA technique and the range of values observed per technique.

Table 2: Summary of datasets

Dataset	Images	DA Applied	Value Range
Orig	590	none	
Origx2	1180	none	
cl	1180	rotate clockwise	5, 10, 15, 20, 25, 30
cc	1180	rotate counter clockwise	5, 10, 15, 20, 25, 30
sl	1180	shear left	horizontal 5, 10, 15, 20
sr	1180	shear right	horizontal 5, 10, 15, 20
gnc	1180	Gaussian noise color	25, 50, 75, 100
gng	1180	Gaussian noise gray	25, 50, 75, 100
snp	1180	salt and pepper	5, 10, 15, 20

To measure the effects of the DA techniques to the performance of a CNN model, several metrics were used. The first metric is precision which is a metric that measures the true positives amongst all positive predictions. Precision indicates how many detections were correct and is vital when minimizing false detections [6]. Precision is calculated using equation (1) [7]. The second metric, recall, calculates the ratio of true positives among actual positives. Recall is important when we want to detect all instances of an object [6]. Precision can be calculated using equation (2) [7].

$$\text{Equation (1): Precision} = \frac{T_p}{T_p + F_p}$$

$$\text{Equation (2): Recall} = \frac{T_p}{T_p + F_n}$$

Where Tp refers to the true positives, Tn indicates true negatives, while Fp is the false positives, and Fn is the false negatives. The next metric is F1-Score which indicates the harmonic mean between precision and recall [7] and is calculated by the equation (3).

$$\text{Equation (3): } F_1 - \text{Score} = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}}$$

The metric mean average precision (mAP) is also utilized. mAP is a comprehensive evaluation of a model's performance and comes in two metrics: mAP50 and mAP50-95 [6]. mAP50 means that it is the mAP calculated at the intersection over union threshold of 0.5 while mAP50-95 is the value calculated from multiple intersection over union thresholds between 0.5 and 0.95 with a step size of 0.05 [7]. These metrics are measured within the values of 0 and 1. The closer the values are to zero indicates that the model has a bad performance while the closer values are to one indicates a good performing model. Metric evaluation will be based on the 100th epoch of each DA technique. The focus of this evaluation is to do a comparison on equal settings and not necessarily to get the best metrics.

3. RESULTS AND DISCUSSIONS

The results of each DA technique can be seen in Table 3. For the clockwise rotation experiments, a clockwise rotation of 15° yielded slightly better results than a duplicated dataset beating the duplicated dataset in all metrics except Recall. A clockwise rotation of 10° is less accurate (50% Precision) but the Recall is much higher indicating that this DA technique might be best applied when character recognition is most important. Higher clockwise rotation values (20° and above) incrementally decrease the performance of the model. For the counterclockwise rotation, a value of 10° is most suitable for DA of Baybayin characters yielding better F1-score (64%) and Recall (79%), slightly lesser mAP values and 4% less Precision. A 5° counterclockwise rotation is also a good DA technique but values of 15° and above incrementally decreases the model performance.

For horizontal shearing, all shear left experiments decreased the performance of the model. Shearing to the left with 5° and 15° values yielded better model performance but notably the model decreased performance by as much as 4% compared to the duplicated set. It is also notable that shearing left by 20° improves Recall by 10% (79.7%). For shearing right experiments, a 5° value improves the model performance in all metrics better than the duplicate set by as much as 3%. Higher shearing right values (10° or higher) reduced the model performance but not by much (up to 4% reduction).

The Gaussian noise injection experiments reduced the model performance by as much as 12% (lowest mAP50-95 is 0.5971). All color versions reduce the accuracy by at least 10% (highest precision is 47.5% down from 57.53%). It is worth noting that all color and grayscale versions increased the Recall performance of the model. The color version with the least standard deviation (25 SD) is the most suitable DA technique to employ for Gaussian noise color. Grayscale versions are interesting in a way that a small application (gng_25) beats the color version (gnc_25) in all metrics but strong applications of the grayscale version (gng_100) yielded the worst metrics.

The Salt-and-Pepper noise injection drastically reduced the model's performance in most metrics except Recall. Application of Salt-and-Pepper noise by small amounts (5%/5%) reduced the model's overall performance but increased the Recall by as much as 13% (from 69% to 82%). Salt-and-Pepper noise injection negatively impacts a model performance more than a Gaussian noise injection does.

Table 3: Results at 100 epochs

100th epoch	DA Applied	Precision	Recall	F1 Score	mAP50	mAP50-95
Orig	none	0.2831	0.5862	0.3818	0.4264	0.4198
Origx2	none (duplicated Orig)	0.5753	0.6887	0.6269	0.7216	0.7210
cl_05	rotate clockwise 5°	0.5119	0.7386	0.6047	0.7110	0.7107
cl_10	rotate clockwise 10°	0.5019	0.8588	0.6335	0.7080	0.7073
cl_15	rotate clockwise 15°	0.5981	0.6878	0.6398	0.7279	0.7271
cl_20	rotate clockwise 20°	0.4772	0.7449	0.5817	0.6611	0.6603
cl_25	rotate clockwise 25°	0.4532	0.8088	0.5809	0.6509	0.6497
cl_30	rotate clockwise 30°	0.4432	0.6858	0.5385	0.5828	0.5822
cc_05	rotate counter clockwise 5°	0.5600	0.7145	0.6279	0.7050	0.7039
cc_10	rotate counter clockwise 10°	0.5381	0.7930	0.6411	0.7115	0.7098
cc_15	rotate counter clockwise 15°	0.4040	0.7929	0.5353	0.6442	0.6435
cc_20	rotate counter clockwise 20°	0.4408	0.8127	0.5715	0.6513	0.6495
cc_25	rotate counter clockwise 25°	0.4057	0.7448	0.5253	0.5970	0.5965
cc_30	rotate counter clockwise 30°	0.3909	0.7331	0.5099	0.5601	0.5597
sl_05	horizontal shear left 5°	0.5326	0.7538	0.6242	0.6887	0.6884
sl_10	horizontal shear left 10°	0.4030	0.7744	0.5301	0.6073	0.6064
sl_15	horizontal shear left 15°	0.5369	0.6728	0.5972	0.6806	0.6788
sl_20	horizontal shear left 20°	0.4491	0.7970	0.5745	0.6558	0.6539
sr_05	horizontal shear right 5°	0.5969	0.7562	0.6672	0.7504	0.7495
sr_10	horizontal shear right 10°	0.4955	0.7352	0.5920	0.6878	0.6869
sr_15	horizontal shear right 15°	0.4993	0.7885	0.6114	0.6959	0.6951

sr_20	horizontal shear right 20°	0.5151	0.6470	0.5736	0.6893	0.6883
gnc_25	Gaussian noise color 25 SD	0.4506	0.7617	0.5662	0.6492	0.6490
gnc_50	Gaussian noise color 50 SD	0.4750	0.7131	0.5702	0.6586	0.6585
gnc_75	Gaussian noise color 75 SD	0.4380	0.7138	0.5428	0.6275	0.6274
gnc_100	Gaussian noise color 100 SD	0.4159	0.6908	0.5192	0.6098	0.6098
gng_25	Gaussian noise gray 25 SD	0.5392	0.7642	0.6323	0.7126	0.7116
gng_50	Gaussian noise gray 50 SD	0.4265	0.7255	0.5372	0.6246	0.6239
gng_75	Gaussian noise gray 75 SD	0.3898	0.7741	0.5185	0.6045	0.6040
gng_100	Gaussian noise gray 100 SD	0.4045	0.7302	0.5206	0.5973	0.5971
snp_5_5	salt & pepper noise 5%/5%	0.4334	0.8179	0.5666	0.6610	0.6602
snp_10_10	salt & pepper noise 10%/10%	0.3216	0.6833	0.4374	0.4906	0.4894
snp_15_15	salt & pepper noise 15%/15%	0.3684	0.4106	0.3883	0.2659	0.2659
snp_20_20	salt & pepper noise 20%/20%	0.3657	0.3251	0.3442	0.2124	0.2124

SD = standard deviation (mean = 0 for all Gaussian noise)

4. CONCLUSION

In conclusion, the significance and usefulness of DA techniques on scarce datasets cannot be underestimated, and its implications on CNN model performance warrant careful consideration. Throughout this paper, an exploration into the ways in which DA techniques can be applied to a small dataset for Baybayin character recognition have been performed.

With the analysis and scrutinization of the results of the experiments, it became evident that choosing the best configuration of a DA technique is paramount in CNN model training as it directly affects the model's capabilities. Acknowledging the negative impacts of a DA technique such as noise injection does not imply a dismissal of the benefits; rather, it underscores the need for careful consideration of choosing the DA techniques which can best train the model to adapt to real world conditions when detecting objects such as, in this case, Baybayin characters.

For future studies, it is recommended that further research looks into more advanced DA techniques that can be applied to Baybayin character recognition. Additionally, the current increments (i.e. increments of 5° in rotation) can be expanded ideally to include all acceptable possible values.

It is the researchers' collective responsibility to propagate their cultural heritage, the Baybayin script, amongst all Filipinos and increase awareness and expertise of the script.

References

1. AMIKA (2021). Baybayin OCR Computer Vision Project. Accessed: Oct. 22, 2023. [Online]. Available: https://universe.roboflow.com/amika/baybayin-ocr
2. Arslan, M., Guzel, M., Demirci, M., & Ozdemir, S. (2019). SMOTE and Gaussian Noise Based Sensor Data Augmentation. 4th International Conference on Computer Science and Engineering, pp. 1–5. https://doi.org/10.1109/UBMK.2019.8907003
3. C. D. L. Dela Rosa, K. Z. L. G. Lagunilla, J. V. Ramos, A. K. V. San Pedro, G. A. R. Sampedro & R. L. Rodriguez (2022). Cross-Model Interpretation of Machine Learning-Based Automatic Baybayin Script Recognition Systems. 2nd International Conference in Information and Computing Research (iCORE), Cebu, Philippines, pp. 302-307.doi:

10.1109/iCORE58172.2022.00071

4. Chowdhury, R. R., Hossain, M. S., Ul Islam, R., Andersson, K., & Hossain, S. (2019). Bangla Handwritten Character Recognition using Convolutional Neural Network with Data Augmentation. Joint 8th International Conference on Informatics, Electronics & Vision and 3rd International Conference on Imaging, Vision & Pattern Recognition, pp. 318–323.a. https://doi.org/10.1109/ICIEV.2019.8858545
5. E. C. C. Hao, G. B. S. Lim, M. K. Cabatuan, E. Sybingco & A. Dulay (2022). CNN-based Baybayin Character Recognition on Android System, IEEE Region 10 Conference 2022, Hong Kong, Hong Kong, pp. 1-6. doi:10.1109/TENCON55691.2022.9977568
6. Jocher, G. & Abirami, V., (2023). 'YOLO Performance Metrics'. Accessed: Dec. 21, 2023. [Online]. Available:https://docs.ultralytics.com/guides/yolo-performance-metrics
7. Karna, N. B. A., Putra, M. A. P., Rachmawati, S. M., Abisado, M., & Sampedro, G. A. (2023). Toward Accurate Fused Deposition Modeling 3D Printer Fault Detection Using Improved YOLOv8 With Hyperparameter Optimization. In IEEE Access, vol. 11, pp. 74251–74262. https://doi.org/10.1109/ACCESS.2023.3293056
8. Khosla, C. & Saini, B. S. (2020). Enhancing Performance of Deep Learning Models with different Data Augmentation Techniques: A Survey. International Conference on Intelligent Engineering and Management (ICIEM), pp. 79–85.https://doi.org/10.1109/ICIEM48762.2020.9160048
9. A. M. P. Ligsay, J. B. Rivera & J. F. Villaverde (2022). Optical Character Recognition of Baybayin Writing System using YOLOv3 Algorithm. IEEE International Conference on Artificial Intelligence in Engineering and Technology, Kota Kinabalu, Malaysia, pp. 1-5.doi: 10.1109/IICAIET55139.2022.9936792
10. Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data Pre-processing and Data Augmentation Techniques. Global Transitions Proceedings, vol. 3(1), pp. 91–99. https://doi.org/10.1016/j.gltp.2022.04.020
11. Mumuni, A. & Mumuni, F. (2022). Data augmentation: A Comprehensive Survey of Modern Approaches. Array, vol. 16, 100258, ISSN 2590-0056. https://doi.org/10.1016/j.array.2022.100258
12. Nair, P. P., James, A., & Saravanan, C. (2017). Malayalam Handwritten Character Recognition Using Convolutional Neural Network. International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, pp. 278-281.doi:10.1109/icicct.2017.7975203
13. Ng, P. & Ma, K. (2006). A Switching Median Filter With Boundary Discriminative Noise Detection For Extremely Corrupted Images. IEEE Transactions on Image Processing, vol. 15(6), pp. 1506–1516. https://doi.org/10.1109/TIP.2005.871129
14. Nogra, J. A., Romana, C. L. S., & Maravillas, E. (2019). LSTM Neural Networks for Baybáyin Handwriting Recognition. IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, pp. 62-66.doi: 10.1109/CCOMS.2019.8821789
15. Oraño, J. F. V., Pahamotang, M. E. P., & Malangsa, R. D. (2022). Using Deep Learning and Adaptive Thresholding Approach for Image-based Baybayin to Tagalog Word Transliteration. IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Boracay Island, Philippines, pp. 1-6.doi: 10.1109/HNICEM57413.2022.10109553
16. Taylor, L. & Nitschke, G. (2018). Improving Deep Learning with Generic Data Augmentation. IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, pp. 1542-1547.doi: 10.1109/SSCI.2018.8628742
17. Ultralytics. (2023). YOLOv8. Accessed: Oct. 22, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics