Character Recognition of Handwritten Baybayin Symbols using EfficientNetV2 Convolutional Network

Kyle Francis Campit[1], Symon Alan Fontillas[1], Jian Carlo Mendoza[1], Kobi Angelo Rasing[1], Marielet Guillermo[1], Arvin Fernando[1], Neil Oliver Velasco[1,*]

[1] *Gokongwei College of Engineering, De La Salle University*
**Corresponding Author: neil.velasco@dlsu.edu.ph*

Abstract: This study aims to explore the use of EfficientNetV2 for Baybayin symbol character recognition. Existing studies use other architectures of CNNs for the same purpose. EfficientNetV2 has a faster training speed and parameter efficiency suited for lightweight applications. The dataset used is a publicly available Baybayin dataset, and the researchers made a custom dataset for testing containing 63 samples for each class. The different types of the EfficientNetV2 architectures are then trained and the B2 model resulted in the highest validation accuracy among 7 other architectures. The B2 model underwent hyperparameter tuning to determine the best configuration for training the model. The network resulted in 95.9% validation accuracy on the publicly available dataset, and 79.3% accuracy in a custom dataset made by the researchers. The performance of the network is comparable to existing research given the network has much less parameters and less training data. The performance and parameters of the network makes it a more viable option for lightweight applications.

Key Words: Baybayin recognition; EfficientNetV2; optical character recognition

1. INTRODUCTION

In 2018, the Philippine Congress approved House Bill 1022, also known as the "National Writing System Act" whose aim was to declare Baybayin as the national writing system of the Philippines (Congress of the Philippines, 2018). Baybayin is a pre-Spanish writing system of the Philippines that was primarily used by the Tagalog people residing in the northern part of the Philippines (Bayani Art, n.d.). In the paper of Pino et al. (2021), it was stated that ever since Baybayin was declared as the national writing system, multiple optical character recognition systems or techniques have been proposed and implemented. However, most of the studies were focused on classification and recognition at the character level. As of today, character recognition for Baybayin is not yet well-developed and fully explored as compared to other writing systems given that it is not mainly used by almost all Filipinos in their daily lives. Given the existing Baybayin OCRs, the goal of this research is to integrate EfficientNet V2 to further increase the accuracy of existing models in detecting or recognizing the Baybayin writing system.

The objective of this study is developing a complete character recognition model for the hand-written variations of Baybayin characters using the EfficientNetV2 recognition model. Another objective is improving previous implementations by exploring hyperparameter model tuning.

1.1 Review of Related Literature

Several optical character recognition approaches are made throughout the years. Most popular being different architectures of artificial neural networks and convolutional neural networks with multiple applications are being considered for the task. (Drobak & Linden, 2020), (Guillermo et al., 2023),

(Fernando et al., 2015). Aside from neural networks, genetic algorithms are also considered, which are iterative in searching for an optimal model equation (Kimura et al., 2009), (Velasco et al., 2019).

There have already been multiple studies that delve into Baybayin character recognition using convolutional neural networks. In 2020, Nogra implemented a Baybayin handwriting recognition system using an Inception network. Nogra's proposed model made use of the same dataset as the EfficientNet V2 model in the study but with only 59 classes rather than 63. Nogra was able to yield a 96.2% validation accuracy using the Inception network model.

Hao et al's (2022) implementation of a CNN-based Baybayin character recognition system made use of MobileNetV2 as its baseline. The study made use of a dataset with 63 classes and 45,833 total images, albeit with an imbalanced number of samples requiring duplication in certain classes. The study yielded a 96.02% validation accuracy.

Bague et al (2020) implemented their recognition model for Baybayin handwritten letters using the VGG16 deep convolutional neural network. Their study made use of a significantly larger dataset with 1,500 images for each of their 45 classes, totalling 67,500 images in total. Due to their substantial dataset, their study yielded a testing accuracy of 98.84%. The VGG16 network has a model size of 528 MB and consists of 138.4 million parameters.

2. METHODOLOGY

2.1 Dataset Preparation

The dataset used for the training and validating of the EfficientNet V2 model is James Nogra's publicly available Baybayin handwritten images dataset (Nogra, 2020). The dataset contains 9,845 images of Baybayin handwritten by students and teachers from the Cebu Institute of Technology. The dataset consists of 32-by-32 1-channel images of characters. There are a total of 63 classes, one for each possible Baybayin character, and each of those classes has 160 to 220 images.

For the use of the dataset in the model, the images are first preprocessed using TensorFlow's JPEG decoder. An 80-20 train-test split was employed for the training and validation of the models. The 80-20 split is applied to each of the classes rather than the dataset as a whole. This means that each class will be well represented both in the training data and the validation data.

2.2 Model Architecture

EfficientNetV2 is the convolutional network that is used for the Baybayin character recognition model. The main contribution of the model is its faster training speed and improved parameter efficiency (Tan & Le, 2021). Compared to networks such as EfficientNetV1, ResNet-RS, and DeiT.ViT, EfficientNetV2 accomplishes similar Top-1 accuracy on the ImageNet ILSVRC2012 dataset with much-improved training efficiency and a much lower number of parameters. There are seven versions of EfficientNetV2 available under the Keras library: B0, B1, B2, B3, S, M, and L (Keras Applications, n.d.). The variations have different combinations of accuracy, parameter size, and training time. Selecting the optimal convolutional network for the character recognition model depends on the acceptable trade-off between accuracy and time.

The EfficientNetV2 architecture builds on the original EfficientNet model. The major distinctions between the two models are the extensive use of MBConv and fused-MBConv for the earlier layers, the smaller expansion ratios, the smaller 3x3 kernel sizes with accompanying increase in layers, and the removal of the last stride-1 stage from the original model to reduce parameter size.

2.3 Hyperparameter Tuning

Using hyperparameter tuning, the most optimal optimizer, learning rate, number of epochs, and batch size is going to be determined. The search for the optimal hyperparameters will be done for five trials with an objective of focusing on the validation categorical accuracy since it is the goal of the hyperparameter tuning process to determine the hyperparameters that would give the best accuracy that the model can have. As for the optimizers, each will be utilized during the hyperparameter tuning process and will be compared to other optimizers based on the validation accuracy that they have produced. These are the initial hyperparameters that are to be tested in the model:

- Learning rate: 0.0001, 0.001, 0.01, 0.1
- Epochs: 20, ..., 100
- Batch size: 32, ..., 256
- Loss function: Categorical Cross-Entropy
- Optimizer: SGD, RMSprop, Adam, Adagrad
- EfficientNet V2 Version: B0, B1, B2, B3, S, M

2.4 Deep Learning Framework Utilization

EfficientNet V2 comes with seven variations available in the Keras CV library. While the architectures of the models are generally the same, there are slight differences from version to version. There are differences in the number of parameters, the number of convolutional blocks used, the number and type of filtered, the width coefficients, and the depth coefficients. The table below summarizes the differences between each of the EfficientNetV2 presents.

Table 1. Summary of EfficientNetV2 Versions

Preset Name	Parameters	Description
efficientnetv2_b0	5.92M	B-style architecture with 6 convolutional blocks. This has a width coefficient of 1.0 and depth coefficient of 1.0.
efficientnetv2_b1	6.93M	B-style architecture with 6 convolutional blocks. This has a width coefficient of 1.0 and depth coefficient of 1.1.
efficientnetv2_b2	8.77M	B-style architecture with 6 convolutional blocks. This has a width coefficient of 1.1 and depth coefficient of 1.2.
efficientnetv2_b3	12.93M	B-style architecture with 7 convolutional blocks. This has a width coefficient of 1.2 and depth coefficient of 1.4.
efficientnetv2_s	20.33M	EfficientNet architecture with 6 convolutional blocks.
efficientnetv2_m	53.15M	EfficientNet architecture with 7 convolutional blocks.
efficientnetv2_l	117.75M	EfficientNet architecture with 7 convolutional blocks, but more filters than efficientnetv2_m.

3. RESULTS AND DISCUSSION

3.1 EfficientNetV2 Architecture Performance

To determine the most optimal preset for the project, each of the models is tested using the training and validation datasets. In this initial testing phase, all models are initialized as an ImageClassifier model with EfficientNet V2 as its backbone. The optimizer used is Adam with a learning rate of 0.001 and the loss function used is categorical cross entropy. Each of the models runs for 20 epochs. The table below summarizes the performance of each model.

Table 2. Summary of Model Performance per Framework

Arch.	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
B0	0.1833	94.64%	0.2575	92.33%
B1	0.2600	92.19%	0.2956	91.62%
B2	**0.1987**	**93.73%**	**0.1880**	**94.21%**
B3	1.0816	66.92%	0.4419	86.34%
S	0.5288	83.49%	1.0103	78.52%
M	1.4181	53.47%	0.9529	66.58%
L	1.8997	40.85%	1.8974	39.69%

The primary observation of the performance of the different EfficientNet V2 models is that the B-type architectures performed significantly better than the non-B-type architectures. The models with the B-type architectures averaged a validation loss of 0.2957 and a validation accuracy of 91.13 % compared to the 1.2877 average validation loss and 61.60% average validation accuracy. That is 77.04% lower validation loss and 32.40% higher accuracy for the B-type architectures. This could signify that the lower number of parameters of the B-type models was more beneficial for the character recognition problem. The width and depth coefficients may also have contributed to significant differences in performance. Out of the four B-type models, the best performing is the B2 model with a validation loss of 0.1880 and a validation accuracy of 94.21%. With this, the EfficientNetV2 B2 model will be used for hyperparameter tuning to further improve performance.

3.2 Model Hyperparameter Tuning Results

Upon conducting hyperparameter tuning on each optimizer, it has been evident that RMSprop had the best performance as it was able to produce the best validation accuracy as shown in the table below. With this optimizer in use, the best learning rate was said to be 0.001 with the batch size set to 64 which produces a validation accuracy of approximately 0.96.

Table 3. Sample Hyperparameter Tuning and Validation Accuracy

Optimizer	Learning Rate	Batch Size	Validation Accuracy
Adam	0.001	32	93.33%
SGD	0.1	256	90.97%
RMSProp	**0.001**	**64**	**95.64%**
Adagrad	0.01	32	93.28%

Using the RMSprop optimizer with the given values for the learning rate and batch size, these were used to train and evaluate the performance of the model which was able to generate an accuracy of 95.85%. The amount of epochs used for training was limited since early stopping was utilized in training the model to prevent overfitting. As for the performance of the model in terms of the training and validation accuracy and losses, the trends for both areas followed the same pattern without overfitting the model which indicates that the model is well-trained for a much more general set of inputs. This trend is shown in Fig. 1.

3.3 Model Predictions and Confusion Matrix

A true test data set of 63 samples for each of the classes is created by the researchers to test the real-world application of the model. Sample characters are shown in Fig. 2. The dataset was created using a drawing tablet and initially had dimensions of 280 by 280. Each of the characters is loaded into the model in the JPEG file format before the appropriate decoding and preprocessing measures take place.

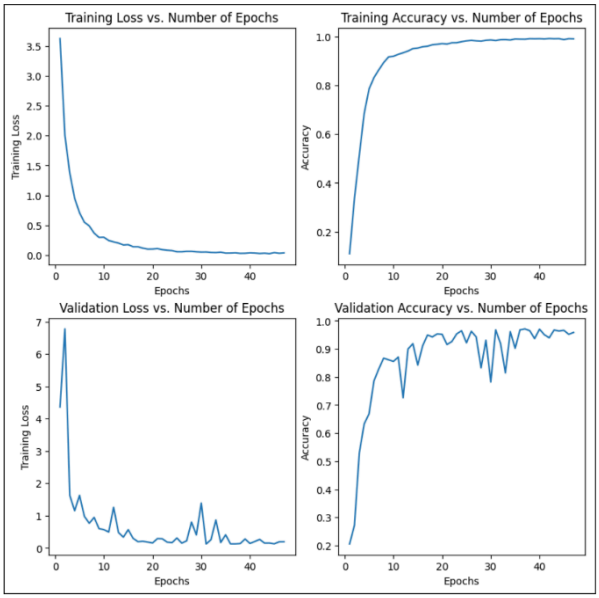

Fig. 1. Training Graph of the B2 Model

Sample characters are recognized by the model as shown in Fig. 3. The model yielded a 79.37% accuracy for the true test samples. This score could be better understood by analyzing the confusion matrix shown in Fig. 4 for the 15 consonants when disregarding the diacritic and the confusion matrix for the characters when only taking into account the vowel of the diacritic. When removing the diacritic from consideration, the model yielded a true test accuracy of 93.33%, with some misclassifications evident between the n and ng characters. This significantly higher performance without the diacritic implies that most of the misclassifications are due to variants of the characters.

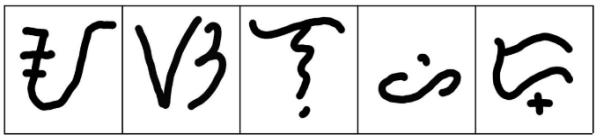

Fig. 2. Sample images from the true test dataset

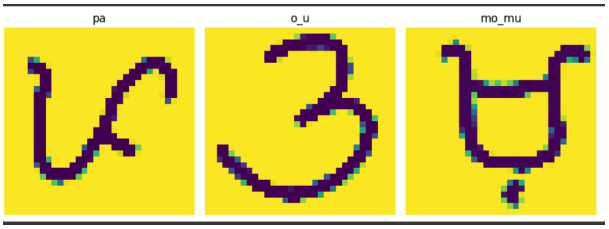

Fig. 3. Test characters with the model with labels

Exploring the confusion matrix shown in Fig. 5 for the dataset when classifying based on the vowel/diacritic, the biggest source of the misclassifications can be identified. The e/i diacritic is responsible for 77.92% of the 13 misclassifications of the model. This issue could be caused by a relatively small sample per class or the poor quality of some of the samples.

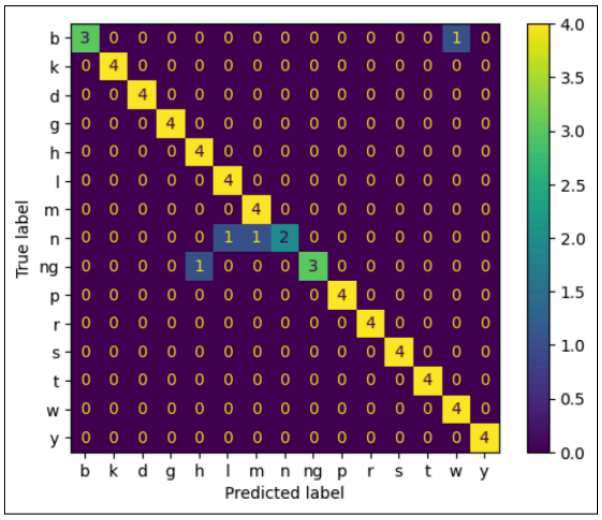

Fig. 4. Confusion matrix for Baybayin consonants

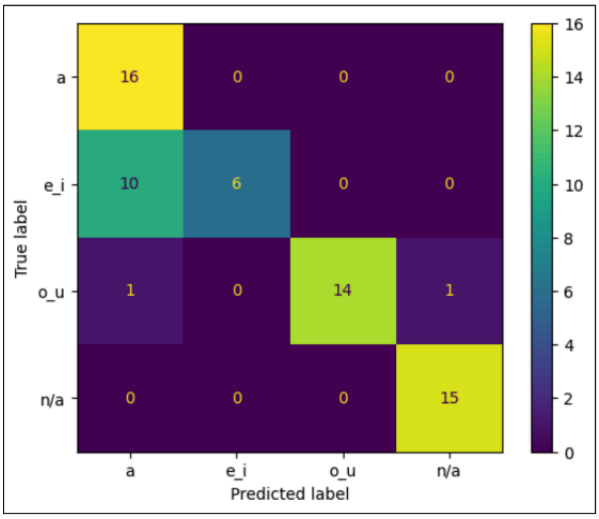

Fig. 5. Confusion matrix for vowel/diacritics of characters

3.4 Comparative Performance

From the implementation of Nogra (2020), the Inception network resulted in a 96.2% validation accuracy, compared to the 95.9% validation accuracy of the EfficientNetV2. The difference is very minimal and may be due to some variances in training.

From the implementation of Hao et al. (2022), the study yielded a comparable 96.02% compared to the 95.9% of the EfficientNetV2 implementation. The network may still increase with more samples.

From the implementation of Bague et al. (2020), their study yielded a testing accuracy of 98.84%. This can be attributed to the size of the VGG16 network used by their study. However, the B2 model of EfficientNet only has 42MB and 10.2 million parameters compared to the 528MB and 138.4 million parameters of VGG16. This study's implementation is a much lighter model while retaining a high accuracy. The EfficientNetV2 model may also improve given the access to a similar dataset used by Bague et al. (2020).

4. CONCLUSIONS

This project mainly focused on the development of a robust Baybayin character recognition using the EfficientNetV2 convolutional network. The goal of this project was to enhance existing optical character recognition (OCR) systems by integrating EfficientNetV2, known for its speed and efficiency in training. Given that EfficientNetV2 has different versions, each version was tested and plotted to see which is the most ideal for Baybayin detection. From here on, the chosen model was further improved through hyperparameter tuning to achieve the desired results. During the hyperparameter tuning, different optimizers were tested and it was found that RMSprop produced the best results. Through dataset preparation, model architecture selection, and an iterative process of hyperparameter tuning optimization, the model was able to get a validation categorical accuracy of approximately 95.85%. During the actual prediction with the test dataset, the model was able to recognize the characters and generated an accuracy result of 79.37%, indicating the capability of the model to recognize characters.

However, there are instances wherein the model misinterprets the character for another character due to the fact that some characters were similar in stroke. Results showed that removing diacritics led to a higher accuracy rate of 93.33%, indicating that misclassifications are mainly due to

character variants. Moreover, the e/i diacritic was found to be the culprit for 77.92% of misclassifications. The possible reason for this could be due to the limited training dataset for this specific diacritic. Another possible factor for the recognition errors is the size of the training dataset wherein using a larger dataset could have improved the generalization process. With that in mind, hyperparameter tuning must be performed once again to get a new set of learning rates and batch sizes so that even the smallest features of the test image will be taken into account.

In conclusion, this research represents a significant step forward in the development of robust Baybayin OCR systems. By utilizing the intricacies and capabilities of deep learning and embracing cultural diversity, it would be possible to pave the way for a more inclusive and technologically empowered future. The recommendation to further improve the research is the implementation of the network in mobile applications, and testing the time response of the network for real-time applications.

5. ACKNOWLEDGMENTS

Special thanks to Engr. Dino Dominic Ligutan and Engr. Neil Oliver Velasco for being our instructors and mentors in CPECOG2 - Neural Networks and LBYCPH3 - Neural Networks Laboratory. We are also grateful to De La Salle University for being instrumental for us to conduct our study.

6. REFERENCES

Bague, L. R., Jorda, R. J. L., Fortaleza, B. N., & Evanculla, D. (2020). Recognition of Baybayin (Ancient Philippine Character) handwritten letters using VGG16 deep convolutional neural network model. *International Journal of Emerging Trends in Engineering Research, 8(9).*

Bayani Art. (n.d.). Baybayin - ancient writing script of the Philippines - Bayani Art. *Bayani Art.* https://www.bayaniart.com/articles/baybayin/

Drobac, S., & Lindén, K. (2020). Optical character recognition with neural networks and post-correction with finite state methods. *International Journal on Document Analysis and Recognition, 23,* 279–295.

Fernando, A. H., Marfori, I. A. V., & Maglaya, A. B. (2015). A comparative study between artificial neural network and linear regression for optimizing a hinged blade cross axis turbine. In *2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)* (pp. 1-4). IEEE.

Guillermo, M., Francisco, K., Concepcion, R., Fernando, A., Bandala, A., Vicerra, R. R., & Dadios, E. (2023). A Comparative Study on Satellite Image Analysis for Road Traffic Detection using YOLOv3-SPP, Keras RetinaNet and Full Convolutional Network. In *2023 8th International Conference on Business and Industrial Research (ICBIR)* (pp. 578-584). IEEE.

Hao, E. C., Lim, G. B., Cabatuan, M. K., Sybingco, E., & Dulay, A. (2022). CNN-based Baybayin Character Recognition on Android System. *TENCON 2022 - 2022 IEEE Region 10 Conference (TENCON).*

House of Representatives Press Releases. (2018, April 23). Congress of the Philippines. https://www.congress.gov.ph/press/details.php?pressid=10642

Keras Applications. (n.d.). Keras.io. Retrieved February 28, 2024, from https://keras.io/api/applications/

Kimura, Y., Suzuki, A., & Odaka, K. (2009). Feature Selection for Character Recognition Using Genetic Algorithm. In *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)* (pp. 401-404).

Nogra, J. A. E. (2020). Inception Network for Baybayin Handwriting Recognition. *International Journal, 9*(1.3).

Pino, R., Mendoza, R., & Sambayan, R. (2021). A Baybayin word recognition system. *PeerJ Computer Science, 7*, e596. https://doi.org/10.7717/peerj-cs.596

Tan, M., & Le, Q. (2021, July). Efficientnetv2: Smaller models and faster training. In *International conference on machine learning* (pp. 10096-10106). PMLR.

Velasco, N. O. M., del Rosario, J. R. B., & Bandala, A. A. (2019). Solving 3D Coverage Problem using Genetic Algorithms in Wireless Camera-Based Sensor Network Modelling. In *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*