BAYBAYIN SCRIPT WORD RECOGNITION AND TRANSLITERATION USING A CONVOLUTIONAL NEURAL NETWORK

Ric Andrei Vilvar[a], Daniel Shawn C. Hammond[b], Francis Mark Santos[c], Hernan S. Alar[d]

City of Makati, Philippines

University of Makati

[a]*rvilvar.k11617264@umak.edu.ph*,[b]*dhammond.a61827562@umak.edu.ph*,
[c]*fsantos.k11614538@umak.edu.ph*, [d]*hernan.alar@umak.edu.ph*

Abstract

In the Philippines, the sudden interest in the Baybayin script through the advocacy of institutions to revive its use leads to the need for it to be preserved and taught for future generations. A learning solution is needed for an easier and faster understanding of the script. This paper presents a Convolutional Neural Network (CNN) based transliteration model that converts Baybayin symbols into a coherent Filipino word. The model was trained using datasets that are specific for Baybayin character classification, word detection, and transliteration and was then evaluated in these individual categories. A VGG16 based classification model was used as a benchmark for the Baybayin character classification. Results show that the proposed and benchmark model has the same accuracy of 97.4% and 97.26% but the models have a significant difference between their predicting and training time. The assessment of word detection and transliteration using a novel dataset has an accuracy score of 96.15% and 91.54% respectively.

Keywords: Baybayin, Character Detection, Convolutional Neural Network, Deep Learning, OpenCV, Optical Character Recognition, Transliteration

1. Introduction

Baybayin is an ancient script of the Philippines used before the Spanish colonial period [1][2]. This ancient writing system consists of 59 unique characters in which 3 are for vowels and the others are for the syllables. There are only 14 consonants in this script but the symbol for these consonants has four variations, with each variation corresponding to 3 syllables or a consonant. During pre-colonial times, Baybayin was used in Tagalog-speaking regions before it was replaced by the Spaniards with the Latin alphabet. The use of Baybayin, especially in lowland areas, began to disappear as Filipinos began to learn the Latin alphabet from the Spaniards. In 2018, the House Committee on Basic Education and Culture approved House Bill 1022 or the "National Writing System Act" which aims to declare Baybayin as our national writing system. The bill also aims to promote awareness of the ancient writing system and to encourage the use of it for the wider appreciation of the script [6]. As the resurgence of interest in Baybayin grew, few Filipino researchers have been implementing new technologies to help this ancient script integrate to the modern world.

In this study, we proposed a Convolutional Neural Network (CNN) classification model to identify the Baybayin written words and transliterate them into the Filipino language written in the Latin alphabet.

2. Related Studies

2.1 The Baybayin Script

The Baybayin script is considered as an Abugida, a writing system where symbols can represent a vowel, consonant, or syllables. The Baybayin as seen on Table 1 has 3 symbols for vowels (A, E/I, O/U) and 14 symbols for syllables that starts with a consonant (B, C/K, D/R, G, H, L, M, N, NG, P, S, T, W, and Y) and ends with a vowel with the default being A and can be change into an E/I or O/U if the symbol is written with a dash or mark called *kudlit* above or below it, respectively [1][3]. The Spaniards modified the 14 symbols of syllables to be a symbol for consonants by adding a cross mark on the bottom.

Table 1 Baybayin Symbols

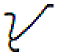
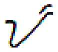
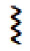
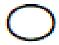
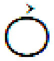
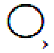
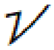
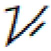
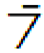
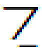
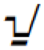
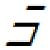
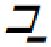
		A	E / I	O / U
		ᜀ	ᜁ	ᜂ
B	ᜊ᜔	ᜊ	ᜊᜒ	ᜊᜓ
C / K	ᜃ᜔	ᜃ	ᜃᜒ	ᜃᜓ
D / R	ᜇ᜔	ᜇ	ᜇᜒ	ᜇᜓ
G	ᜄ᜔	ᜄ	ᜄᜒ	ᜄᜓ
H	ᜑ᜔	ᜑ	ᜑᜒ	ᜑᜓ
L	ᜎ᜔	ᜎ	ᜎᜒ	ᜎᜓ
M	ᜋ᜔	ᜋ	ᜋᜒ	ᜋᜒ
N	ᜈ᜔	ᜈ	ᜈᜒ	ᜈᜓ
NG	ᜅ᜔	ᜅ	ᜅᜒ	ᜅᜓ
P	ᜉ᜔	ᜉ	ᜉᜒ	ᜉᜓ
S	ᜐ᜔	ᜐ	ᜐᜒ	ᜐᜓ

T	ᜆ᜔	ᜆ	ᜆᜒ	ᜆᜓ
W	ᜏ᜔	ᜏ	ᜏᜒ	ᜏᜓ
Y	ᜌ᜔	ᜌ	ᜌᜒ	ᜌᜓ

During the Spanish colonization, the Spaniards introduced the Latin alphabet to the Filipinos as part of the assimilation of Spanish culture and beliefs [4]. The Baybayin, even though modified and used by the Spaniards in some publications like the Doctrina Christiana in 1593 , has been relegated as a secondary writing system and only a few ethnic groups from remote islands and mountain ranges around the Philippines use it as their writing system [2]. At the end of the Spanish occupation in 1898, the Baybayin evolved into 3 distinct forms (see Table 2) The first form is the Tagbanwa script and is used on the island of Palawan by the ethnic group of the same name. The second form is the Hanunoo that is used by the Hanunuo Mangyans in the southern part of Mindoro. The third one is the Buhid that is used by the Buhid Mangyans of Mindoro.

Table 2 Comparison of Selected Baybayin, Tagbanwa, Hanunuo, and Buhid Symbols

	A	**E** **I**	**O** **U**	**Ba**	**Be** **Bi**	**Bo** **Bu**
Baybayin	ᜀ	ᜁ	ᜂ	ᜊ	ᜊᜒ	ᜊᜓ

Tagbanwa						
Hanunuo						
Buhid						

As the distinct forms of Baybayin are still being used by ethnic groups, the version of the Baybayin script which the Spaniards have modified is the subject of interest of many Filipinos today. It is the version that is being considered as our National Writing System and the subject of debate among Filipinos who propose and oppose this proposition [6][7]. In this paper, we used the modified version of the Baybayin script and created a CNN-based classifier that can identify the 59 symbols of the script as seen on table 1.

2.2 Baybayin Character Recognition

There have been a small number of studies that focused on the use of character recognition for the Baybayin script. In a paper by Nogra, J. et al., they have created a Long Short-Term Memory (LSTM) Neural Network for the conversion of handwritten Baybayin characters into their equivalent English characters [8]. 9,700 handwritten characters are collected with each of the 63 unique characters having 150-200 sample images. The images are scaled down into 28x28 pixels and 8,500 of these images have been used in training 5 LTSM models. In their testing and validation, the LSTM model B with 512 units in the first hidden layer, 256 units in the second layer, and 128 units in the dense layer have achieved an accuracy of 95.6% in training and 92.9% in validation. The authors have created another study on the character recognition of Baybayin a year after their previous study where they used a LSTM Neural Network for Baybayin character conversion [9]. In this study, they focused on a new type of model which is based on CNN architecture but their aim, limitation, and methodology stays relatively the same. The result of this study yields an accuracy rate of 94% for the CNN model that has channel size of 32 for Convolution 1, 64 for Convolution 2, 128 for Convolution 3, and a 256 Fully Connected layer with 3×3 filter size. Daday, M. et al. introduces two Neural Network models for the recognition of Baybayin symbols; the Feed-Forward Neural Network (FFNN) and the CNN both emphasizing that the models are used with a dropout method [10]. They have gathered a total of 36,000 sample images that are resized to be 28x28 pixels for their dataset with 17 unique characters to classify. The result of their study implies that the FFNN-DM model with an accuracy score of 92.4% and a loss rate of 0.25% with an error rate of 7.55% is better than the CNN-DM with an accuracy score of 91.69% and a loss of rate of 0.31% with an error rate of 8.31%. The work done by Bague, L. et al. uses a VGG16 Deep Learning Convolutional Neural Network (DL-CNN) model for the recognition of Baybayin characters and it is the most recent study that focuses on Baybayin character recognition [11]. Their goal is to create a model that can translate the original set of Baybayin characters, the version in which there is no variation of symbol for consonants. In their study, they have created a dataset of 108,000 images that were resized to 50x50 pixels and have 45 unique characters to classify the Baybayin symbol. They trained their system using 80% of their dataset and the result is that their system had an accuracy of 99.54% and 98.84% for the training and testing phase, respectively. Their proposed system can gather user input in two forms and that is through uploading an image file into the system or a real-time translation using a web camera. Real-time translation can also detect and classify multiple Baybayin characters that are put together to form a word and translate it into its Tagalog equivalent given that the characters are 0.5cm apart and the characters are horizontally aligned. During the real-time translation, the researchers reported that their system can recognize all of the 45 Baybayin characters.

A summary of the studies about character recognition for Baybayin is presented on table 3. In this table, 3 out of 4 studies for Baybayin character recognition are limited to the identification and translation of Baybayin symbols and not full words or sentences. The study done by Bague, L. et al. is the only study that translates the Baybayin symbol into Tagalog, a language in which the Baybayin script was made to interpret [11]. The study of Daday, M. et al. has produced the 2 models that have the lowest accuracy compared to the classification models from the other studies [10].

Table 3 Comparison of models developed that focuses on character recognition of Baybayin

Author	**Classification Model used**	**No. of Images for Testing**	**Image Size**	**No. of Output Characters**	**Accuracy Score**
Nogra, J., Romana, C., and Maravillas, E. (2019)	LSTM Neural Network	8,500	28x28	63 (3 Vowels, 15 Consonants, 45 Syllables)	Testing 95.6% Validation 92.9%
Nogra, J., Romana, C., and Maravillas, E. (2020)	Convolutional Neural Network (CNN)	8,500	28x28	63 (3 Vowels, 15 Consonants, 45 Syllables)	94%

Daday,M., Fajardo, A., and Medina, R. (2020)	Feed-Forward Neural Network with Dropout Method (FFNN-DM) CNN with Dropout Method (CNN-DM)	36,000	28x28	17 (3 Vowels, 14 Consonants)	92.4% 91.69%
Bague, R., Jorda Jr., R., Fortaleza, B., Evanculla, A., Paez, M., and Velasco, J. (2020)	VGG16 Deep Learning Convolutional Neural Network	67,500	50x50	45 (3 Vowels, 42 Syllables)	Training 99.54% Testing 98.84%

In 2021, two new studies regarding the Baybayin script were published by the same authors with the first one being the same as the previous studies where their scope focuses on the individual characters of the Baybayin script. The study done by Pino et al. uses a Support Vector Machine (SVM) classifier to differentiate and classify Latin and Baybayin characters [12]. This study also proposed to have a separate classifier for the Baybayin *kudlit.* The result of their study in terms of the Baybayin character classification yielded an accuracy score of 96.51% after being trained and tested with 9,000, 56x56 character images. The second (and most recent) study published by these authors closely resembles the scope of our study as they proposed to transliterate Baybayin words into its Latin alphabetic equivalent. This study by Pino et al. uses their previously made SVM classifier and builds upon its architecture to add the transliteration of the Baybayin characters to form a Latin alphabetic word [13]. The corpus that they have used for this study comprises 74,990 words. Their proposed system is trained and tested using 1000 images of Baybayin words under specific restrictions and achieved an accuracy score of 97.9%. This study is the first to use OCR in the Baybayin script to detect, recognize, and transliterate words. The studies done by the researchers we have mentioned have provided key insights for the development of systems that can be used for Baybayin character recognition.

3. Methodology

3.1 Model Framework

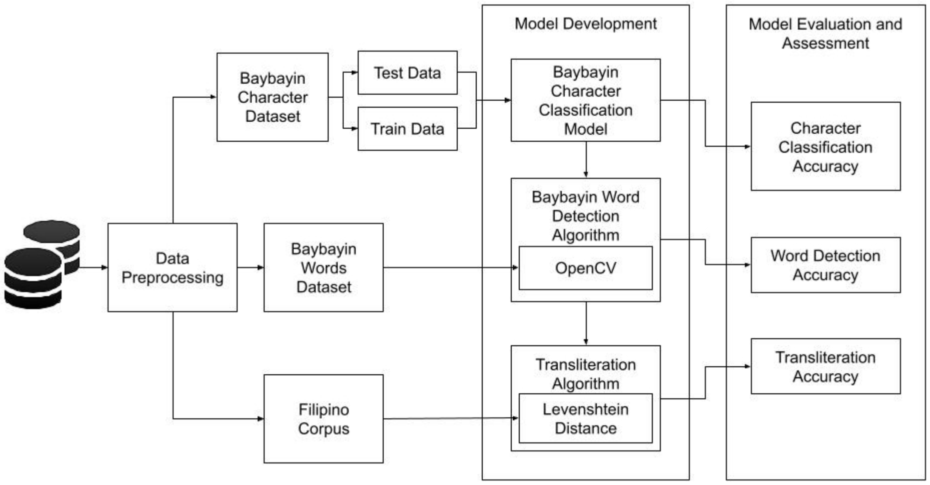

Model Development
Model Evaluation and Assessment
Baybayin Character Dataset
Test Data
Train Data
Baybayin Character Classification Model
Character Classification Accuracy
Data Preprocessing
Baybayin Words Dataset
Baybayin Word Detection Algorithm
OpenCV
Word Detection Accuracy
Filipino Corpus
Transliteration Algorithm
Levenshtein Distance
Transliteration Accuracy

Figure 1 Baybayin Transliteration Model Framework

Figure 1 shows that the model framework is separated into three categories. These are the Baybayin classification model that will classify single Baybayin characters, the Baybayin word detection algorithm that will identify the Baybayin word in the image using OpenCV, and the Baybayin transliteration algorithm which uses Levenshtein distance to transliterate the Baybayin word into a coherent Filipino word based on the Filipino corpus.

These categories were used in the model development that were trained, tested, and evaluated.

3.2 Data Resource

3.2.1 Baybayin Character Dataset

The dataset that we used in this study came from three different sources; The first dataset is a public dataset that we got from an online repository which is a collection of handwritten Baybayin characters [14]. The second dataset was from the benchmark model that we used from the research conducted by Bague et al. concerning image recognition of handwritten Baybayin characters using VGG16 CNN model, and the third dataset is from the crowdsourced data from various people aging from 15 – 55 years old [11].

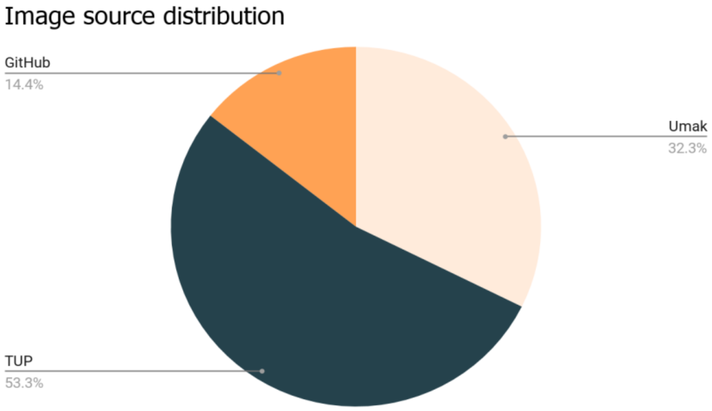


Figure 2: Distribution of the Dataset Sources

Figure 2 shows the number of images that have been collected from the 3 sources. A total of 59,000 images were used for this study with each unique symbol of Baybayin equally having 1,000 images.

3.1.2 Baybayin Word Dataset

The Internet-sourced data that the researchers obtained and used for the project is novel and distinctive. For the dataset that was used, 130 total number of photos were acquired. It was also ensured that the images gathered were correct in terms of their Baybayin spelling. The criteria for selecting the dataset were determined by the scope and limitations of our research.

3.1.3 Filipino Corpus

In the model development, Filipino corpus will be used in correcting the transliteration of the extracted Baybayin text to Filipino, specifically the correction of words. The Filipino corpus that was used from this study was obtained from the collection of datasets from different studies totaling 34,850 Filipino words [15][16][17] (Borra et al., 2010; Dita et al., 2009; Oco et al. 2016).

3.2 Data Pre-processing

3.2.1 Baybayin Character Dataset

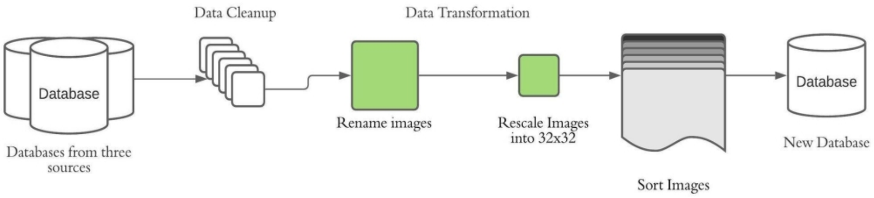


Figure 3. Flow of data pre-processing

Figure 3 shows the flow of data pre-processing and cleanup. Mislabeled characters, wrongly written characters, as well as duplicate characters were manually and carefully renamed, removed, and replaced from the dataset. We renamed all images to X (Y).jpg where X is the character and Y is the character count. The images are then rescaled to 32x32 which is enough for the features to be visible and differentiable. During the model development we relabeled all characters with vowels 'e/i' and 'o/u' into a single consonant. 'be' instead of 'be/bi' and 'bo' instead of 'bo/bu' as an example. This enables the translation model to form the predicted characters into a coherent word.

3.2.2 Baybayin Word Dataset

With the creation of the Baybayin word dataset, the photos were then edited using Adobe Photoshop to eliminate Latin characters in the image. Image augmentation was applied using ImageDataGenerator from Keras to make different versions of the images which are either slanted, repositioned, zoomed in, or zoomed out to make the dataset larger and diversified resulting in 650 Baybayin word images.

3.2.3 Filipino Corpus

The Filipino words that were gathered from 3 different sources were compiled into one dataset for the Filipino corpus. Words that contain non-alphabetic characters were replaced or removed from the corpus as well as;

- English characters and symbols
- English and Filipino proper nouns of non-native names, characters and objects
- Filipino and English phrases

Duplicate words were removed and all remaining words were changed into lowercase characters since Levenshtein distance distinguishes lowercase letters and uppercase letters as different characters, resulting in having a distance of one which will affect the transliteration.

3.3 Model Development

Model development is done on a desktop computer with Intel Core i5 4690k, NVIDIA GTX 1050 and 8 GB RAM specifications, using TensorFlow GPU optimization to speed up the training and testing process. 50 epochs were used to train the model. The performance of the model was evaluated using classification accuracy.

3.3.1 Baybayin Classification Model

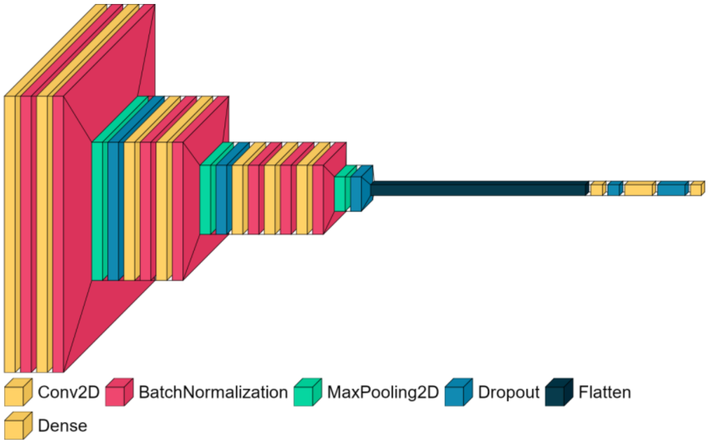
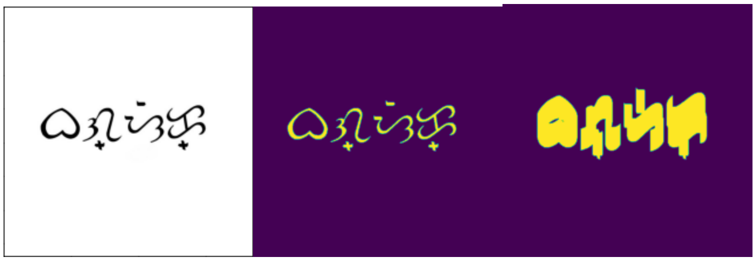


Figure 4: (A),(B) Baybayin recognition model architecture

We used a CNN model to recognize or classify the Baybayin characters. As presented on Figure 4, we used batch normalization and dropouts to optimize the model and to reduce overfitting [18]. This CNN model is capable of classifying all 59 Baybayin characters compared to other existing models in Baybayin character recognition that classifies some of the Baybayin characters only or alters the classification for the characters [8][9][10][12]. The CNN model that we developed will predict the characters after the detection algorithm individually crops the detected characters.

3.3.2 Baybayin Word Detection Algorithm

We used OpenCV in our proposed word detection algorithm to detect the Baybayin characters in the image. Our detection algorithm is highly dependent on the contour of the image. Hence, we used binarization to remove unnecessary noise in the image. We also used dilation to stretch out the detected contour vertically so that the *kudlit* would look like it is a part of the character. After detecting the characters, the detection algorithm crops the detected characters into separate images then the Baybayin classification model will predict the individual characters. Finally, the Baybayin word detection algorithm will form the predicted characters into a single word.



(A) (B) ©

Figure 5 Flow of the pre-processing method of dilation (A) Input Baybayin image (B) Binarized image (C) Image is stretched vertically using dilate

3.3.3 Baybayin Transliteration Algorithm

The Baybayin transliteration algorithm that we developed uses Levenshtein distance to look up the predicted word to the Filipino corpus following the Levenshtein distance formula (see Figure 5). The output word will depend on the distance between the predicted word and the words that are in the Filipino corpus. If the predicted word is in the Filipino corpus or if the distance is zero, the output word will be the predicted word. If the predicted word is not in the Filipino corpus, then the words with shortest distance will be the output. The model will then display the text "*The word is not in the dictionary. The possible translations are:"* to indicate that the detected word is not present in the Filipino corpus. The output words then would be properly capitalized.

3.3.4 Baybayin Transliteration Model

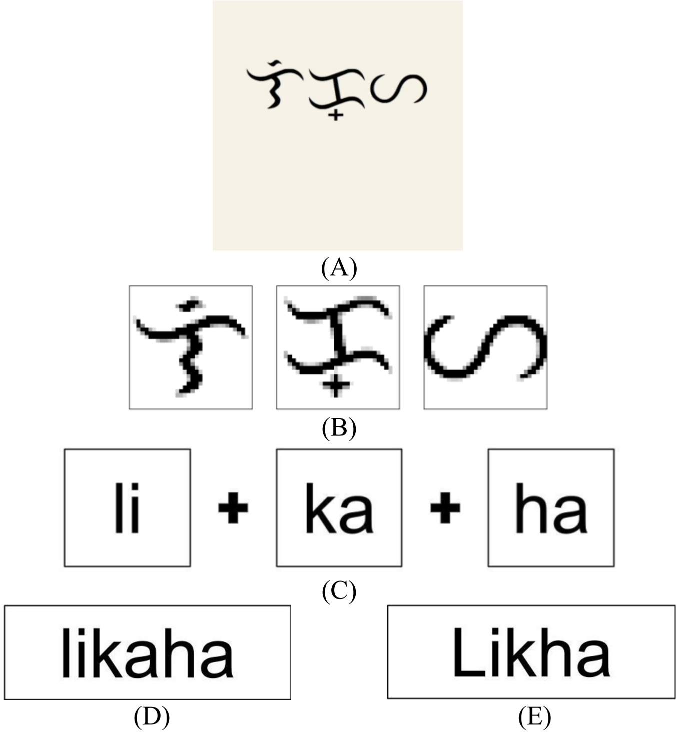


Figure 6 Workflow of the transliteration model (A) Input Baybayin image (B) Detected and cropped characters (C) Predicted Baybayin characters (D) A word created from the predicted characters (E) The Tagalog word found in the Filipino corpus with the shortest distance to the created word

Figure 6 shows the overall process of the application will start by detecting and identifying the Baybayin word in the image and as mentioned earlier, the given input images should follow these requirements or prerequisites:

- The text is written in black with a white background.
- A character should be written in thick and continuous strokes.
- A character's *kudlit* is written in the center top or center bottom and must not be wider or go across the boundaries of its character.

- The Baybayin characters that form a word must not overlap with each other.

These requirements are a set of rules established in the model to correctly extract the characters of the given image. The word will be placed in a bounding box adjusted based on the contour of the word. The contour analysis method allows for the recognition of items based solely on their exterior contours, with the inside contours of the objects being ignored [19]. After the model detects the individual characters inside the bounding box, the detected word will be cropped from the image and will be divided into individual characters. The number of the divided individual characters will be determined and the characters will then be predicted by the model. Next, the predicted characters will be appended into a word and the model will search for the word in the Filipino corpus using Levenshtein distance, and eventually selecting the word with the minimum distance as the best result. The Levenshtein distance works by calculating the difference between two words and the degree of resemblance between two words is determined by the distance between them [20].

4. Result and Discussion

4.1 Baybayin Classification Model

The study done by Bague et al. proposed a VGG16 based classification model for Baybayin character recognition and this model will be considered as the benchmark model to be compared to as it has the highest accuracy in its predictions and one of the latest studies in this topic [11]. The classification accuracy, prediction time, and training time of our model and the benchmark model were recorded to compare their performance. A stratified K-fold validation test was used to evaluate the performance of these models.

Table 4: Prediction accuracy of the Benchmark and Proposed Model

Fold #	Benchmark Model	Proposed Model
K1	97.90	97.60

K2	96.60	97.00
K3	97.60	97.80
K4	97.50	97.80
K5	96.70	96.80
Average	**97.26**	**97.40**
P-Value	**0.5823**	

Table 4 shows that the accuracy of the proposed model (**97.40%)** is relatively the same as the accuracy of the benchmark model (**97.26%),** with a hairline difference of **0.14%** and a p-value of **0.5823** indicating that there are no significant difference between the two models at **α = 0.05**.

Table 5: Predicting Time (sec/s) of Benchmark and Proposed Models

Fold #	**Benchmark Model**	**Proposed Model**
K1	2.07	1.52
K2	2.04	1.55
K3	2.13	1.53
K4	2.18	1.51
K5	2.19	1.52
Average	**2.12**	**1.53**

P-Value	**0.000061**

Table 5 shows that the proposed model has an average predicting time of **1.53** seconds. This is faster than the benchmark model's predicting time of **2.12** second. A paired T-test was used on the predicting time records of the two models. The paired T-test resulted with a p-value of **0.000061** indicating that there is a significant difference between the two models' predicting time at **α = 0.05**.

Table 6: Training Time (min/s) of Benchmark and Proposed Model

Fold #	**Benchmark Model**	**Proposed Model**
K1	33.18	15.45
K2	37.40	15.51
K3	37.22	15.30
K4	36.56	15.35
K5	38.48	15.28
Average	**36.57**	**15.38**
P-Value	**0.000021**	

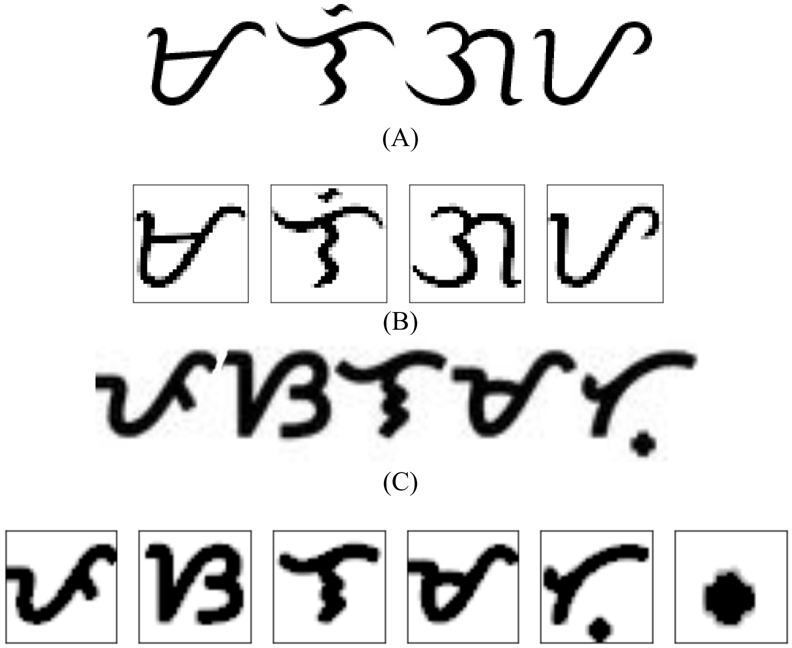
Table 6 shows that the proposed model is twice faster in average training time with **15.38** minutes outperforming the average training time of the benchmark model with **36.57** minutes. A p-value of **0.000021** indicates that there is a significant difference between the two models at α = 0.05

4.2 Baybayin Word Detection Algorithm

The detection accuracy is computed by using the formula:

$$Baybayin\ Word\ Detection\ Accuracy\ = \frac{Number\ of\ Correctly\ Detected\ Baybayin\ Word}{Total\ Number\ of\ Test\ Baybayin\ Word} * 100\%$$

A test image is considered correct if the detection algorithm correctly detects the number of Baybayin characters present in the image. The test image is considered incorrect if it detects the *kudlit* as a separate character or if the number of characters detected does not match the Baybayin characters present in the image.



(A)

(B)

(C)

(D)

Figure 7 Example of correct and incorrect detection (A) Test input image 1 (B) Example of correct detection (C) Test input image 2 (D) Example of incorrect detection

A total of 650 Baybayin word images were tested and 625 of these images were correctly detected. This interprets to a competitive 96.15% detection accuracy.

4.3 Baybayin Transliteration Algorithm

The transliteration accuracy is computed by using the formula

$$Baybayin\ Transliteration\ Accuracy = \frac{Number\ of\ Correctly\ Transliterated\ Word}{Total\ Number\ of\ Test\ Baybayin\ Word} * 100\%$$

The transliterated word is considered correct if the word is found in the Filipino corpus and that the transliterated word matches the actual translation of the input image. The transliterated word would be considered incorrect if the word was not found in the Filipino corpus, if the transliterated word is found in the Filipino corpus but outputs a different word, and if the transliterated word does not match the actual translation of the input image. 595 out of 650 words were correctly transliterated resulting in 91.54% transliteration accuracy.

The test results of Baybayin word detection accuracy and Baybayin transliteration accuracy can be accessed in a Google spreadsheet created by the authors [21].

5. Conclusion

In this paper, we introduced a CNN-based transliteration model to identify the Baybayin written words and transliterate them into the corresponding Filipino word. The transliteration model was evaluated in three categories being; Baybayin character classification, Baybayin word detection, and Baybayin word transliteration.

The Baybayin character classification model can identify the 59 characters of the Baybayin Script. The proposed model had an accuracy of **97.4%** which is close to the accuracy of the benchmark model at **97.26%**. Although the accuracy of both models is relatively the same, the proposed model outperformed the benchmark model in terms of the time it took in training and predicting the Baybayin characters. The proposed model has an average training time of **15.38 minutes** compared to the **36.57 minutes** average training time of the benchmark model. The proposed model has an average prediction time of **1.53 seconds** opposed to the benchmark model's prediction time of **2.12 seconds**.

In the Baybayin word detection algorithm, our test proves that the detection algorithm we developed is capable of detecting Baybayin characters with 96.15% detection accuracy as long as the image is within the criteria of our scope and limitation.

In our proposed Baybayin transliteration algorithm, the test proves that the transliteration algorithm is capable of transliterating Baybayin words into Filipino with its 91.54% transliteration accuracy. Although the result was not as competitive as Pino et. al Baybayin word recognition system using SVM with 97.6% accuracy, we still have proven that using Levenshtein distance is still a good transliteration model to use in single word recognition [13]. Using the study of Holley as a basis for measuring Optical Character Recognition (OCR) accuracy, our transliteration model with the word accuracy of 91.54% would be considered as 'average accuracy' based on their metrics of acceptable accuracy rates for OCR [22].

The few studies done on this topic by recent researchers shows that there are still further improvements that can be applied and worked upon. Our transliteration algorithm could still be further improved by cleaning and expanding on the Filipino corpus as well as tuning the Baybayin character classification model. We hope that this work will be helpful to future studies about Baybayin computer vision and other classification and transliteration models.

Acknowledgements

We would like to express our gratitude to the authors of the research entitled "Recognition of Baybayin (Ancient Philippine Character) Handwritten Letters Using VGG16 Deep Convolutional Neural Network Model" who let us use their dataset in our study. This work was supervised by the University of Makati under the College of Computer Science.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal ties that would have influenced the work presented in this study.

Funding

This study received no specific support from public, private, or non-profit funding bodies.

References

[1] Cabuay, C. (2012) "An Introduction to Baybayin" (pp.7-11). Retrieved from: https://books.google.com.ph/books?id=VVfGAQAAQBAJ&dq=baybayin&lr=&source=gbs_navlinks_s

[2] Potet, J. P. G (2019) "Baybayin, the Syllabic Alphabet of the Tagalogs". Lulu.com. Retrieved from: https://books.google.com.ph/books/about/Baybayin_the_Syllabic_Alphabet_of_the_Ta.html?id=rHGADwAAQBAJ&redir_esc=y

[3] Rodríguez, F. R. (2013) "Early writing and printing in the Philippines". HIPHILANGSCI. Retrieved from: https://hiphilangsci.net/2013/07/10/early-writing-and-printing-in-the-philippines/

[4] Kawahara, T. (2016) "A study of literacy in Pre-Hispanic Philippines" (pp. 24-25). Retrieved from: https://www.izumi-syuppan.co.jp/LLO_PDF/vol_08/16-02Kawahara.pdf

[5] Plasencia, J. (1593) "Doctrina Christiana,"

[6] Morallo, A. (2018). "House panel approves use of Baybayin as country's national writing system" in PhilStar.com. Retrieved from: https://www.philstar.com/headlines/2018/04/23/1808717/house-panel-approves-use-baybayin-countrys-national-writing-system

[7] Garcia, P. (2019). "Beyond ABCs: Ancient Philippine script revival spells debate" in mb.com.ph. Retrieved from: https://mb.com.ph/2019/07/31/beyond-abcs-ancient-philippine-script-revival-spells-debate

[8] Nogra, J., Romana, C., and Maravillas, E. (2019) "LSTM Neural Networks for Baybáyin Handwriting Recognition" in 2019 IEEE 4th International Conference on Computer and

Communication Systems (pp. 62-66) Retrieved from: https://sci-hub.do/10.1109/CCOMS.2019.8821789

[9] Nogra, J., Romana, C., and Maravillas, E. (2020) “Baybáyin Character Recognition Using Convolutional Neural Network” in International Journal of Machine Learning and Computing Vol. 10 No. 2 (pp. 265-270) Retrieved from: doi: 10.18178/ijmlc.2020.10.2.930

[10] Daday,M., Fajardo, A., and Medina, R. (2020) “Recognition of Baybayin Symbols (Ancient Pre-Colonial Philippine Writing System) using Image Processing” in International Journal of Advanced Trends in Computer Science and Engineering. Vol. 9 No. 1 (pp. 594-598) retrieved from: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse83912020.pdf

[11] Bague, R., Jorda Jr., R., Fortaleza, B., Evanculla, A., Paez, M., and Velasco, J. (2020) “Recognition of Baybayin (Ancient Philippine Character) Handwritten Letters Using VGG16 Deep Convolutional Neural Network Model” in International Journal of Emerging Trends in Engineering Research. Vol. 8 No. 9 (pp. 5233 – 5237) retrieved from: http://www.warse.org/IJETER/static/pdf/file/ijeter55892020.pdf

[12] Pino, R., Mendoza, R., & Sambayan, R. (2021a). “Optical character recognition system for Baybayin scripts using support vector machine”. PeerJ Computer Science, 7, e360. Retrieved from: http://dx.doi.org/10.7717/peerj-cs.360

[13] Pino, R., Mendoza, R., & Sambayan, R. (2021b). A Baybayin word recognition system. PeerJ Computer Science, 7, e596. Retrieved from: http://dx.doi.org/10.7717/peerj-cs.596

[14] ”Baybayin-Handwritten-Character-Dataset,” (May 1, 2019). Retrieved from: https://github.com/jmbantay/Baybayin-Handwritten-Character-Dataset (accessed Dec. 2 2020)

[15] Borra, A., Pease, A., Roxas, R., & Dita, S. (2010). Introducing Filipino WordNet. In Principles, Construction and Application of Multilingual Wordnets: Proceedings of the 5th Global WordNet Conference.

[16] Dita, S., Roxas, R. E., & Inventado, P. (2009). “Building online corpora of philippine languages.” In Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2 (pp. 646-653).

[17] Oco, N., Syliongka, L. R., Allman, T., & Roxas, R. E. (2016). Resources for Philippine Languages: Collection, Annotation, and Modeling. In Proceedings of the 30th Pacific Asia Conference on Language, Information, and Computation (pp. 433-438).

[18] Srivastava, N., Hinton, G., Krizhevsky, A., and Sutskeve, I. (2014) “Dropout: A simple

way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, issue 1, pp. 1929-1958. Retrieved from: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

[19] Lubis, A. H., Ikhwan, A., & Kan, P. L. E. (2018). Combination of levenshtein distance and rabin-karp to improve the accuracy of document equivalence level. International Journal of Engineering & Technology, 7(2.27), 17-21.

[20] Pervej, M., Das, S., Hossain, M. P., Atikuzzaman, M., Mahin, M., & Rahaman, M. A. (2021). Real-Time Computer Vision-Based Bangla Vehicle License Plate Recognition using Contour Analysis and Prediction Algorithm. International Journal of Image and Graphics, 2150042.

[21] Hammond, D. et. al. (2021). Appendix: Baybayin Accuracy. Available at https://docs.google.com/spreadsheets/d/1NLSNtTUeIkZ51s4Y5yPK28AzTjgwxnBYBhEEB5zOHU4/edit?usp=sharing. [Accessed on December 30, 2021]

[22] Holley, R. (2009). How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. D-Lib Magazine, 15(3/4).