

Brahmi Script Classification using VGG16 Architecture Convolutional Neural Network

Vincen Vincen, Samsuryadi Samsuryadi

*Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya
vincen320@gmail.com

ABSTRACT

Many Indonesians have difficulty reading and learning the Brahmi script. Solving these problems can be done by developing software. Previous research has classified the Brahmi script but has not had an output that matches the letter. Therefore, letter classification is carried out as part of the process of recognizing Brahmi script. This study uses the Convolutional Neural Network (CNN) method with the VGG16 architecture for classifying Brahmi script writing. Training results from various amounts of image data. Smooth model. The requested image data is a 224x224 binary image. This study has the highest quality, accuracy is 96%, highest recall is 98% and highest precision is 98%.

Keywords: Brahmi Script, Deep Learning, Convolutional Neural Network, VGG16

1. INTRODUCTION

Based on an article written by (Utama, 2017) [1] that many Indonesian people are not fluent in reading the Brahmi script at a productive age. Overcoming these problems through the introduction of Scripts must be done as early as possible. Brahmi script is a language that originated in Ancient India. This script can still be found in religious books on Buddhist and Japanese culture [2]. In this study, the Brahmi script was chosen as a supporting motif in order to introduce one of the oldest scripts in Indonesia because the Javanese and Balinese scripts are derivatives of the Indian Brahmi script through the Kawi script intermediary and are closely related to the Balinese script [3,4]. One of the Pali languages is written using the Brahmi script, Devanagari and so on. Pali is studied to study Buddhist texts and is often sung by Buddhists as well. Pali language itself shows that this language is used as a liturgical language or for teaching Buddhism [5]. In previous studies, the Brahmi script was also classified but did not have output that corresponded to the letters but only numbers. Therefore, by classifying by providing output in the form of letters, it will be useful for the community. This paper consists of elements such as points, lines, curves, corners and edges, which can be known as feature extraction [6] which can be performed on the convolution layer [7].

Deep Learning is a subset of Machine Learning that allows computers to understand and experience the world based on data representations used for learning. Many people have used deep learning for text classification and recognition. All Deep learning methods in text recognition utilize Convolutional Neural Networks in extracting features in images [8]. Many networks have been developed with deep learning algorithms. Many problems have been solved such as classifying using a deep network architecture [9]. Deep Learning has been used by several big technology companies such as NVIDIA, Google, Microsoft, Facebook, Apple,

Adobe, etc. Deep Learning has also helped researchers a lot [10]. Deep learning techniques improve system vision in identifying and classifying based on computer vision applications [11]. There have been many studies that have used a deep learning approach as a method for classifying letters [12,13].

CNN is a method that has been widely used for the image classification process because CNN was built to process multiple array type data such as images, videos and signals. Convolutional Neural 'fully connected layers and also this method has high accuracy, because CNN requires convolution and pooling processes based on the information received [14].

VGG16 architecture was discovered by Zisserman and Simonyan in 2014 which is a deeper network based on the AlexNet network and can accurately provide dataset characteristics when searching and classifying images [15]. The use of VGG16 architecture has been used for character recognition in Tamil script and can produce a higher level of accuracy than the previous method, which is 94.52% with the same data. Tamil script is a language originating from India with the Brahmi script writing system [13]. This architecture can also run better when classifying large and complex datasets [15]. VGGnet got 2nd place in the ImageNet competition in 2014 with an error value of 7.4%. VGG16 is one of VGGnet whose architectural model uses 16 layers.

2. MATERIAL AND METHODS

2.1. DATASET

The type of data used in this study is secondary data taken from the website of the dataset provider, namely www.kaggle.com which was uploaded by Neha Gautama which consists of 4 vowels, 27 consonants and the rest are composite characters. Each letter has 25 train data and 5 test data, which means it has a total of 30 data per letter.

The data obtained will be processed to provide zero padding, grayscaling for binarization then resize and perform data augmentation. So that the training process can be carried out then the label of the data is changed according to the corresponding letter because the data label obtained is only a sequence of dataset numbers, then unites the train and test folders. The data is added again as many as 20 images per letter by drawing the letters themselves with Paint application.

The data will be tested using accuracy, recall and precision matrices. Testing the level of accuracy, recall and precision is done by dividing by the number of datasets. The first test was carried out using a comparison of datasets that are 70% training data and 30% test data, 60% training data and 40% test data, 50% training data and 50% test data.

Tests are also carried out by looking at the effect of the amount of training data on the level of accuracy, recall and precision of testing using the same amount of test data, namely 12 images for each class. The number of training data tested is 12, 18, 24, 30, and 25 images for each class. Tests are carried out so that there are no other factors that affect the value of accuracy, recall or precision.

2.2. METHODS

2.2.1. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network has a very high efficiency and is also the most commonly used in different computer vision applications [16]. CNNs are inspired by biological and have a multilayer class of deep learning models that use in a single neural network that has been trained from image pixel values into a classification output. CNN has become one of the methods that dominate all text recognition such as Handwriting, scene text and license plate recognition and has become the de facto standard for these tasks [17].

CNN has an architecture that is divided into two major parts, that are Feature Extraction Layer and the Fully connected layer [18].

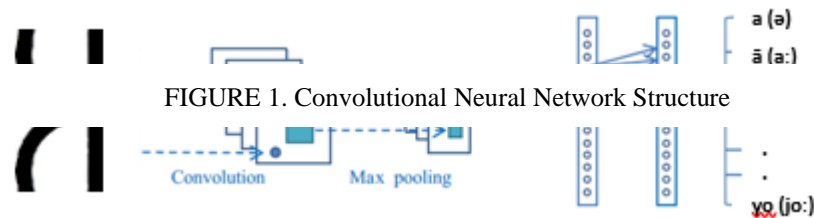


FIGURE 1. The architecture of CNN

a. Feature Extraction Layer

This architecture is encoding from an image into a feature that is in the form of pixel numbers from the image. In this architecture there is a convolutional layer that is arranged to form a filter that has a length x height (pixels) which will convolute on each shift to all parts of the image pixel so that the output of the filter will produce an activation map or feature map [18].

b. Fully Connected Layer

The feature map generated from the feature extraction layer is still in the form of a multidimensional array, so it must be converted to vector form so that it is flattened so that it can be used as input from the fully connected layer. In this architecture is the same as the Multi-Layer Perceptron concept which has neurons that are interconnected as a whole to process data so that it can be classified [18].

In the Feature Extraction Layer, there are three layers that have an important role in CNN:

a. Convolutional Layer

A layer in which each neuron will dot the product between the weights and a small region of the previous input. Which aims to identify features from the previous layer and map its appearance into a feature map. The calculation formula for the Convolutional Layer [19]:

$$Z_i^l = B^l + \sum_{j=1}^{k-1} W_{i,j}^l * Z_j^{l-1}, \text{ where } i \in [1, k^l], B^l \quad (1)$$

b. Non-Linearity Layer

Perform a non-linear function performed on feature map components. Calculation formula on Non-Linearity Layer [19]:

$$a^l = f(z^l) \quad (2)$$

c. Pooling Layer

Used to minimize output via convolution of the pooling function. In this research used the Max Pooling operation. Max Pooling calculation formula on Pooling Layer [19]:

$$p^l = \max_{i \in R} a_i^l \quad (3)$$

2.2.2. VGG16

The VGG16 architecture was discovered by Zisserman and Simonyan in 2014 [15]. VGG16 (D) uses 16 hidden layers consisting of convolution layer, max pooling layer and fully connected layer [20].

In the training process, the input layer has a size of 224 x 224 RGB images. The Convolution Layer uses a 3 x 3 filter, has a stride value of 1px, 1px padding for 3 x 3 convolution layers, 5 times the max pooling size of 2 x 2 with 2 strides. Each hidden layer has a ReLU activation function. In Fully Connected layers consist of three layers, first and second layers has 4096 channels, and third layer has 1000 channels with softmax layer [21].

2.2.3. PERFORMANCE METRICS

Confusion matrix displays information about the classification with the actual result being a 2-dimensional matrix. One dimension shows the actual result and one dimension shows the predicted result [22].

The metrics that are often used to describe how the classification level is in detecting a class are as follows [23].

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (4)$$

$$\text{Recall/Sensitivity} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (7)$$

where are TP = *True Positive*, TN = *True Negative*, FP = *False Positive*, FN = *False Negative*.

2.2.4. FLOWCHART

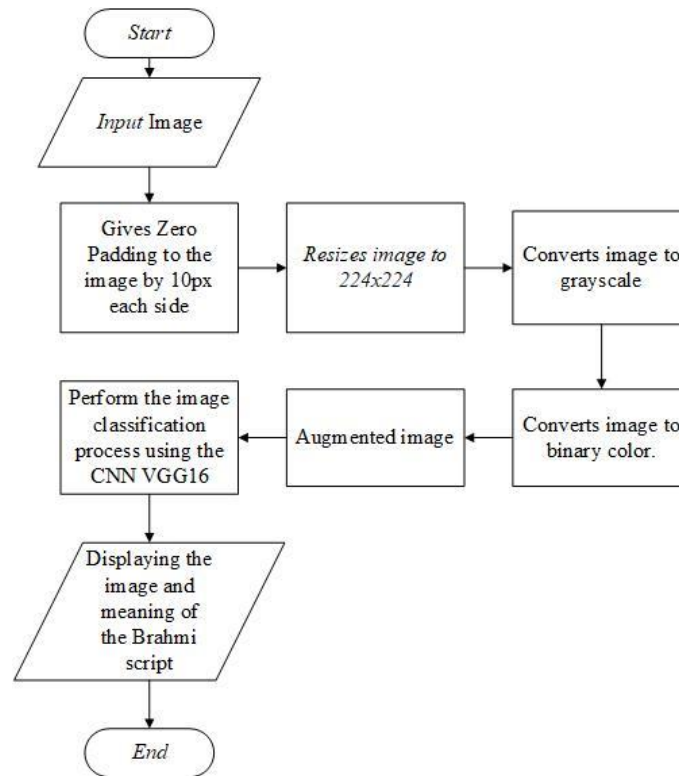


FIGURE 2. Flowchart

- a. Input Image, in the form of image input from the dataset obtained so that it is immediately processed to the next stage.
- b. Giving Zero Padding to the image by 10px each side, at this stage the input data is carried out the first pre-processing, which is to provide zero padding of 10px each side so that the image is in the middle.
- c. Resizes image to 224x224, at this stage it is done to change the image size to 224x224 so that it can be used in CNN input.
- d. Converts image to grayscale, at this stage the image that has been resized will be convert to grayscale so that the next process can be done, namely binarization.
- e. Converts image to binary color, at this stage the image is converted to black and white with white depicting letters because white has a pixel value of 255 which means that the computer has a value.

f. Augmented image, data Augmentation is a method that performs data transformation on a set of training data so as to produce new sample data. Data augmentation can improve accuracy and reduce bias in the data by changing the training data by transforming the data [24,25]. The image is augmented by the process of horizontal shifting, vertical shifting of 10px and zooming out of 10%.

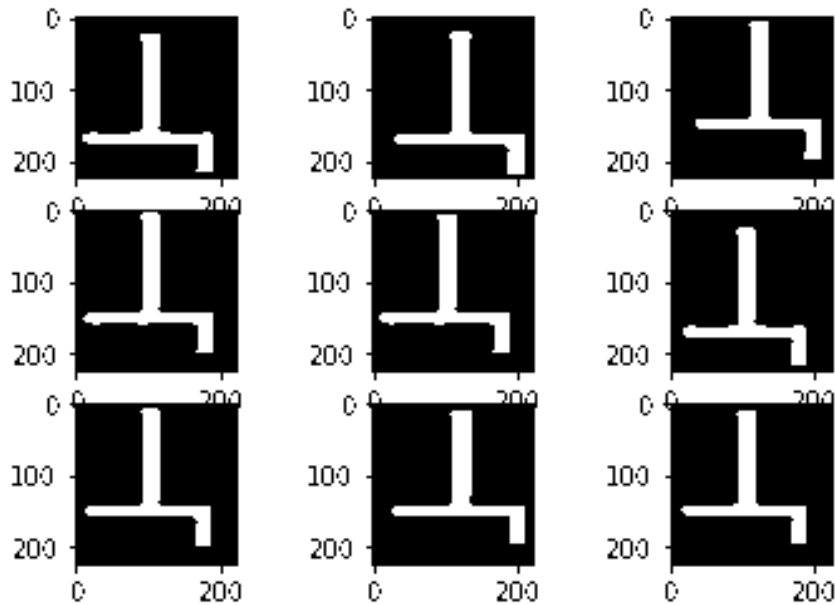


FIGURE 3. Augmented Images

g. Perform the image classification process using the CNN VGG16, the images carried out by a series of processes will then be classified using a model that has been trained by CNN VGG16. Model trained with 10 epochs, 0.00001 learning rate and 25 batch size hyperparameters.

h. Displaying the image and meaning of the Brahmi script, the classification results will display the image and meaning of the Brahmi script.

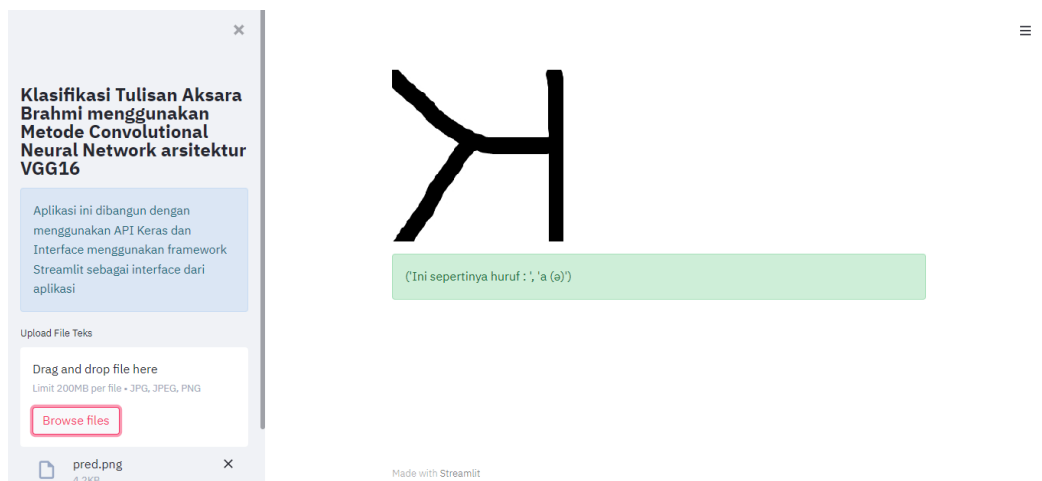


FIGURE 4. Testing output

3. RESULTS AND DISCUSSION

Testing result using test data from each number of dataset comparisons shown on Table 1.

TABLE 1.
Testing with each number of datasets companions

Model	Data = 8500		Accuracy	Recall	Precision
	train	test			
1	70%	30%	98%	98%	95%
2	60%	40%	96%	97%	94%
3	50%	50%	94%	95%	92%

Testing result using the same test data amount but with different amount of training for each class data shown on Table 2.

TABLE 2.
Testing with different amounts of training data

Model	Each data per class Class = 170		Accuracy	Recall	Precision
	train	test			
A	12	12	78%	84%	87%
B	18	12	86%	91%	92%
C	24	12	88%	92%	94%
D	30	12	92%	95%	96%
E	35	12	95%	97%	98%

The following Figure 5 is an example of the result of classifying the Brahmi script correctly and incorrectly classifying letters. The writing in red indicates the wrong classification result, while the writing in green indicates the correct classification result.



FIGURE 5. Classification Result

The test results are seen by looking at the comparison of the levels of accuracy, recall and precision in each comparison of the number of datasets. The test results show that the more data used for training, the higher the accuracy, recall and precision, as can be seen in Table 1. In Table 2 it can be seen that the use of the amount of training data also affects the level of accuracy, recall and precision on the model, it can be seen that in Model A and Model B the accuracy level decreases further, testing is also carried out using the same test data, namely as many as 12 image data for each class so that there are no other factors that affect the level of accuracy, recall or precision. Thus, the use of a larger amount of training data results in a higher level of accuracy, recall and precision.

4. CONCLUSION

After some testing classification done, the results shown the testing receive the highest accuracy rate of 96%, the highest recall of 98% and the highest precision of 98%. The more data used as training data, the higher the level of accuracy, recall and precision will be. For further research, it is hoped that can do writing recognition in the form of sentences, add sound for the spelling of Brahmi script and add a greater number of datasets such as handwritten data, or writing on monuments.

REFERENCES

- [1]. F. Utama, "Pengenalan Aksara Melalui Media," *Iqra' J. Kaji. Ilmu Pendidik.*, vol. 2, no. 2, pp. 433–457, 2017
- [2]. D. K. Sharma, V. Emilia, B. Le, and H. Son, "Lecture Notes in Networks and Systems 106 Micro-Electronics and Telecommunication Engineering," *Micro-Electronics Telecommun. Eng.*, pp. 693–705, 2019.
- [3]. K. T. Y. Aditya, M. W. A. Kesiman, and G. A. Pradnyana, "Pengembangan Game Edukasi Tematik Aksara dan Bahasa Bali," *Kumpul. Artik. Mhs. Pendidik. Tek. Inform.*, vol. 8, no. 3, pp. 522–533, 2019, [Online]. Available: <https://ejournal.undiksha.ac.id/index.php/KP/article/viewFile/522/14044>.
- [4]. G. S. Damayanti, "Transformasi Cerita Prabu Watugunung dalam Motif Batik pada Kebaya," 2020, [Online]. Available: <http://digilib.isi.ac.id/id/eprint/7165>.
- [5]. S. Maulidan, "TRADISI SEMBAHYANG UMAT BUDDHA," *Fak. USHULUDDIN DAN FILSAFAT Univ. Islam NEGERI AR-RANIRY*, 2016.
- [6]. N. Gautam and S. S. Chai, "Optical character recognition for Brahmi script using geometric method," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3–11, pp. 131–136, 2017.

- [7]. R. A. Pangestu, B. Rahmat, and F. T. Anggraeny, "Implementasi Algoritma CNN untuk Klasifikasi Citra Lahan dan Perhitungan Luas," *Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 166–174, 2020.
- [8]. C. Zhang, W. Ding, G. Peng, F. Fu, and W. Wang, "Street View Text Recognition with Deep Learning for Urban Scene Understanding in Intelligent Transportation Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4727–4743, 2021, doi: 10.1109/TITS.2020.3017632.
- [9]. Y. Zheng, Y. Cai, G. Zhong, Y. Chherawala, Y. Shi, and J. Dong, "Stretching deep architectures for text recognition," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2015-November, pp. 236–240, 2015, doi: 10.1109/ICDAR.2015.7333759.
- [10]. I. G. and Y. B. and A. Courville, *Deep learning*, vol. 29, no. 7553. 2016.
- [11]. A. Shrivastava, J. Amudha, D. Gupta, and K. Sharma, "Deep Learning Model for Text Recognition in Images," *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, pp. 1–6, 2019, doi: 10.1109/ICCCNT45670.2019.8944593.
- [12]. N. Gautam, S. S. Chai, and J. Jose, "Recognition of Brahmi words by Using Deep Convolutional Neural Network," no. May, 2020, doi: 10.20944/preprints202005.0455.v1.
- [13]. M. A. Pragathi, K. Priyadarshini, S. Saveetha, A. S. Banu, and K. O. Mohammed Aarif, "Handwritten Tamil Character Recognition Using Deep Learning," *Proc. - Int. Conf. Vis. Towar. Emerg. Trends Commun. Networking, ViTECoN 2019*, pp. 1–5, 2019, doi: 10.1109/ViTECoN.2019.8899614.
- [14]. K. Chellapilla et al., "High Performance Convolutional Neural Networks for Document Processing To cite this version : High Performance Convolutional Neural Networks for Document Processing," 2006.
- [15]. H. Wang, "Garbage recognition and classification system based on convolutional neural network vgg16," *Proc. - 2020 3rd Int. Conf. Adv. Electron. Mater. Comput. Softw. Eng. AEMCSE 2020*, pp. 252–255, 2020, doi: 10.1109/AEMCSE50948.2020.00061
- [16]. Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016, doi: 10.1016/j.neucom.2015.09.116.
- [17]. M. Yousef, K. F. Hussain, and U. S. Mohammed, "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks," *Pattern Recognit.*, vol. 108, p. 107482, 2020, doi: 10.1016/j.patcog.2020.107482.
- [18]. K. P. Danukusmo, "Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU," *Univ. Atma Jaya Yogyakarta*, 2017.
- [19]. M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 120–147, 2018, doi: 10.1016/j.isprsjprs.2017.11.021.
- [20]. Y. Gultom, A. M. Arymurthy, and R. J. Masikome, "Batik Classification using Deep Convolutional Network Transfer Learning," *J. Ilmu Komput. dan Inf.*, vol. 11, no. 2, p. 59, 2018, doi: 10.21609/jiki.v11i2.507.

- [21]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.
- [22]. X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci. (Ny)*, vol. 340–341, pp. 250–261, 2016, doi: 10.1016/j.ins.2016.01.033.
- [23]. S. Ruuska, W. Hämäläinen, S. Kajava, M. Mughal, P. Matilainen, and J. Mononen, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behav. Processes*, vol. 148, pp. 56–62, 2018, doi: 10.1016/j.beproc.2018.01.004.
- [24]. E. Lashgari, D. Liang, and U. Maoz, "Data augmentation for deep-learning-based electroencephalography," *J. Neurosci. Methods*, vol. 346, no. July, p. 108885, 2020, doi: 10.1016/j.jneumeth.2020.108885.
- [25]. L. Taylor and G. Nitschke, "Improving Deep Learning with Generic Data Augmentation," *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 1542–1547, 2019, doi: 10.1109/SSCI.2018.8628742.