A Classification Javanese Letters Model using a Convolutional Neural Network with KERAS Framework

Yulius Harjoseputro
Department of Informatics
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia

***Abstract*—One of the essential things in research engaged in the field of Computer Vision is image classification, wherein previous studies models were used to classify an image. Javanese Letters, in this case, is a basis of a sentence that uses the Javanese language. The problem is that Javanese sentences are often found in Yogyakarta, especially the use of name tourist attractions, making it difficult for tourists to translate these Javanese sentences. Therefore, in this study, we try to create a Javanese character classification model hoping that this model will later be used as a basis for developing research into the next stage. One of the most popular methods lately for dealing with image classification problems is to use Deep Learning techniques, namely using the Convolutional Neural Network (CNN) method using the KERAS framework. The simplicity of the training model and dataset used in this work brings the advantage of computation weight and time. The model has an accuracy of 86.68% using 1000 datasets and conducted for 50 epochs based on the results. The average inference time with the same specification mentioned above is 0.57 seconds, and again the fast inference time is because of the simplicity of the model and dataset toolbar. This model's advantages with fast and light computation time bring the possibility to use this model on devices with limited computation resources such as mobile devices, familiar web server interface, and internet-of-things devices.**

***Keywords*—Javanese letters; deep learning; convolutional neural network; epoch; framework KERAS**

I. Introduction

Indonesia is a country that has a variety of cultures; one of the cultures heritages that must preserve is the Javanese letters [1], [2]. Javanese letter is an ancient Javanese character, used since the 17th century [3]. Because the Javanese letters are considered a Javanese style, some researchers raised this topic for their research [1], [4]–[7]. One of the researchers who raised the Javanese letters' issue was the one conducted by Widiarti and Wastu in 2009. The Javanese letters' application was widely available in Indonesia, especially in the province of Yogyakarta Special Region. For example, use in street name, menu names in restaurants around the Sultan's Palace, electronic media [8], and many others. The city of Yogyakarta is also known as one of the towns of Tourism, and not a few tourists who visit Yogyakarta every year. Even Javanese letters' application is often found in tourist destinations that tourists usually visit, for example, in the Palace, North and South Square, Malioboro, and many others. Therefore, in this study, we try to create a Javanese character classification model hoping that this model will later be used as a basis for developing research into the next stage.

This research is significantly related to digital images, and image classification [9], especially in this case, is the Javanese letters' image. Digital painting is now a necessity for many people for various purposes. It can see from the importance of digital images in various fields of science and entertainment. One function of the image, in this case, is to obtain important information contained in the picture, where the extraction of data from an image is the primary goal of Computer Vision [10]. The extraction results can be processed again to get the classification results from the image. The idea of image classification is to give a computer input from a collection of numbers processed from the image. It will produce a number that represents the category of the image.

Image classification alone is not enough, in this case, to get the processing results from that image. There needs to be a technology that can support making the image classification results better, seen from its accuracy and computation time. Therefore, several studies have emerged that use techniques from Deep Learning. Deep Learning is one of the sub-areas of Machine Learning, where the algorithm used is inspired by how the human brain works. In recent years Deep Learning has shown exceptional performance, mostly influenced by more reliable computational factors, large datasets, and techniques for training deeper networks.

One method of deep learning is the CNN method, inspired by the human brain [11] and uses image input as an assumption. Convolutional Neural Network (CNN), in the next few decades, will have many breakthroughs in various fields related to pattern recognition; from image processing to sound detection, the most useful aspect of CNN is reducing the number of parameters in artificial neural networks. This led the researchers to complete complex tasks, which could not be solved using artificial neural network methods [12]. CNN is an algorithm of deep learning, development of Multi-Layer Perception (MLP), designed to manage data in the form of grids, two-dimensional images such as images sounds [13].

This network has a particular layer called the convolution layer, wherein this layer, an inserted image will be processed based on a predetermined filter. Each of these layers will produce a pattern of several parts of the image, which will be easier to classify later. This technique can make the image learning function more efficient to implement. Several annual competitions carried out to enhance research in the field of Computer Vision, including each year, state of the art methods have emerged to win this competition, for example, GoogLeNet [14], AlexNet [15], and ResNet [16]. Besides, many studies have attempted to overcome this problem using a Convolution Neural Network (CNN) because of its ability to provide high character detection [6]. Some uses of the CNN method in its application, especially to recognize the character carried out by Harjoseputro in 2018. His research has succeeded in classifying the Javanese letters with an accuracy of 85% [9]. As for research on the character of Arabic handwriting[17], this study has successfully implemented deep learning and produced accuracy for each AIA9k and AHCD database of 94.8% and 97.6%.

Therefore, the writer thinks that how the Javanese letters are to decorate in tourist attractions, where the visitors, mostly tourists, do not understand the Javanese letters' meaning. In this case, the author would like to conduct research to create Javanese letters using the Convolutional Neural Network method using the Hard Framework. In this case, hard is the high-level neural network API developed in Python, focusing on the goal of accelerating the research or trial process. In this case, the KERAS framework is that there is a built-in function suitable for this Convolutional Neural Network method. Also, with this Hard framework can run the same source code using the CPU or GPU smoothly.

II. The Material and Method

Before discussing the methods used in this research, it is better to know any theory or material about the research topic carried out.

A. Deep Learning

Deep Learning is one area of Machine Learning [9], [17], which utilizes artificial neural networks to implement problems with large datasets. Also, it is a technology that is inspired by the functioning of the human brain. Besides, artificial neuron networks automatically analyze large data sets to find underlying patterns without human intervention. Deep learning identifies unstructured data patterns, such as pictures, sounds, videos, and text [11]. Deep Learning techniques provide a robust architecture for Supervised Learning. By adding more layers, the learning model can better represent labeled image data. Three fundamental reasons underlying the popularity of deep learning today are increased hardware capabilities for computational processing, increased size of data used to train networks, and the growing development of research related to machine learning and data processing. Some deep learning areas include computer vision, speech recognition, natural language processing, and other related fields. In Fig. 1, the following is an overview of the scope of deep learning.

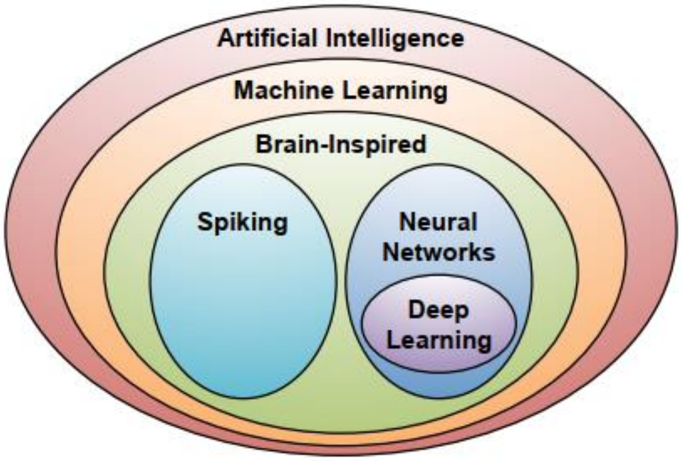

Fig. 1. Deep Learning Scope [18].

B. Convolutional Neural Network

In this Deep Learning technique, one of them is the CNN or Convolutional Neural Network method. This method is a development of the Multilayer Perceptron (MLP), designed to process two-dimensional data. CNN is included in the type of Deep Neural Network because of the high network depth and widely applied to image data [19]. Besides, this CNN method can be excellent in finding useful features in the image to the next layer to form a non-linear hypothesis that can increase a model's complexity. However, completing this method requires a reasonably elaborate model, so it will certainly require quite a long training time. Therefore, when using this CNN method, GPU is very much recommended to speed up the training process [7]. In Fig. 2, the following is the architecture in the Multilayer Perceptron, which has several x layers, shown in red and blue squares with each layer containing y neurons shown in white circles. MLP accepts one-dimensional input data and propagates the data on the network to produce output. Each relationship between neurons on two adjacent layers has a one-dimensional weight parameter that determines the model's quality. Every input data in the segment is carried out linearly with the actual weight value. Then the computational results will be transformed using a non-linear operation called the activation function [19].

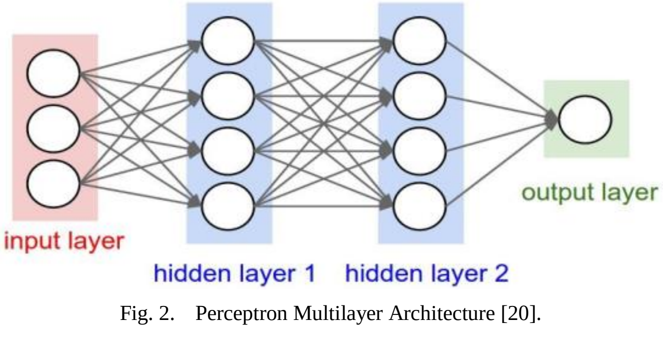

Fig. 2. Perceptron Multilayer Architecture [20].

Convolutional Neural Network (CNN) has several architecture layers, including the convolution layer, pooling layer, full connected layer, and loss layer. Each of these layers has a role in doing the process of managing two-dimensional data. The convolution layer is to carry out convolution operations at the output of the previous layer. This layer is the primary process that underlies a CNN. Convolution is a mathematical term used to apply a function to the output of other functions. The purpose of convolution in an image is to extract features from an input image. The convolution will produce a linear transformation of the input data according to the input data's information. The convolution layer is a system that studies the workings of the visual cortex of the brain and to study filters of the inserted images [13].

C. Tensorflow and Keras Framework

TensorFlow is a software library that has an open-source source for performing numerical calculations using data flow graphs. TensorFlow is created and developed by the Google Brain team and researches Google's intelligence engine for Machine Learning and Deep Learning. Also, TensorFlow is designed for large scale distributed training and inference. The distributed TensorFlow architecture contains a master service with kernel implementation. Tensorflow is designed to be used both in research, development, and production systems. Besides, TensorFlow can run on a single CPU and GPU system and an extensive distributed system with hundreds of nodes used; in its development, TensorFlow has been used to support various algorithms, including in the training process and inference for various Deep Neural Network models. Tensorflow is also the most famous library of data science with many community development and support [21].

Keras is a High-Level Neural Network API that can be run on machine learning frameworks such as Tensorflow, CNTK, or Theano. Hard is built using the python programming language. Keras also provides an API [22] that makes it easy for users to build ANN architecture. Hard has several applications, which are deep learning models that are used together with pre-trained weights. These models can be used to predict, perform feature extraction, or fine-tuning. These models include *Xception, VGG16, VGG19, ResNet50, InceptionV3, InceptionResNetV2, MobileNet, DenseNet & NasNet* [21].

D. Javanese Letters Dataset

This dataset contains images of all existing Javanese characters, namely, 20 Javanese characters, including HA, NA, CA, RA, KA, DA, TA, SA, WA, LA, PA, DHA, JA, YES, YES, MA, GA, BA, THA, NGA, each measuring 32 X 32 pixels taken from various repositories available on the website. Javanese script that is meant not included with his partner. The dataset used is not noisy, which means that it does not require this study's data set's cleansing process. The overall dataset has 1000 training images and 100 test images collected through a repository on the website. The following is an illustration of the image in the Javanese script dataset that will be used can be seen in Fig. 3.

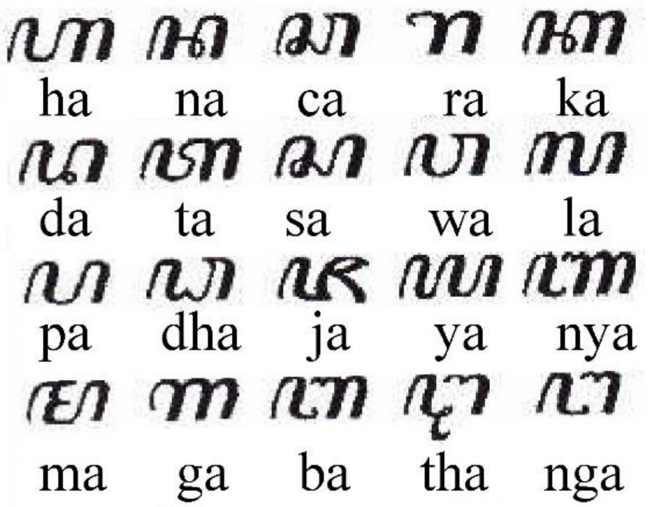

Fig. 3. Sample of Javanese Letter (Source from Google.com).

E. Proposed Javanese Letters Training Model

This work's primary goal is to create the Javanese language alphabet classifier based on the image dataset. The Javanese letters consist of 20 main characters known as Ha-Na Ca-Ra-Ka. The name Ha-Na-Ca-Ra-Ka comes from the first five letters of the Javanese letters [23]. The original data used to train the model uses the set data of the Javanese language alphabet image shown in Fig. 4, with most is a photo of the handwritten alphabet. The image's uses as the primary data in this work will lead to selecting the method to classify the Javanese language alphabet to purposes of the algorithm that suits it. Many machine learning algorithms work for image data types classification with many different approaches such as Support Vector Machine, Bayesian method, and a various number of Neural Network method like Deep Learning.

The Deep Learning approach currently is the most algorithm used for image recognition and classifier. Deep learning is just like the standard Neural Network method but with many layers on its network that makes Deep Learning powerful because of its feature extraction approach that can deal with various object patterns on the image. The current state-of-the-art Deep Learning layers model for image recognition and classifier is called Convolutional Neural Network (CNN). CNN uses some convolution layer stack instead of a fully connected layer to extract the images' feature. Convolution layer on CNN will perform better for feature extraction from the picture because it will use a filter that will learn and adapt to minimize training loss using the Back Propagation or Stochastic Gradient Descent (SGD) technique. The advance of CNN to work with image data objects will be used and adapted to train the Javanese language alphabet dataset.

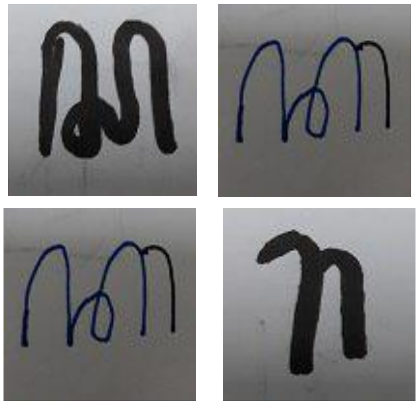

Fig. 4. Sample of Javanese Language Image used in the Dataset.

F. The Convolutional Model

One of the CNN model keys is using the right number and position of the convolution layer stacked to do the feature extraction from the image. Complex image objects like animals, people, or vehicles usually need more than 20 layers of convolution to get the right feature, pattern, and edge detection from the object. Although CNN has an excellent feature extraction result, the uses of so many convolution layers will slowly cost the computation process and time to train the model; also, many image datasets for training are needed. Some standard CNN model with 10 class objects with a lot of complexity usually needs many hours to feed the dataset into the model, even using parallel computation method with GPU or multi-threaded server CPU.

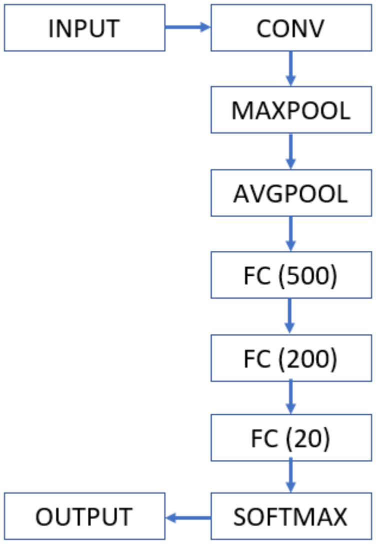

Fig. 5. Architectural Research Model.

The architectural model developed in this study can see in Fig. 5, which includes input with an image size of 32x32 and then continued with the usual convolution layer and the MaxPooling layer. The results of the operation of all these layers then added to the AvgPooling layer. Then the results of this pooling layer will be finalized with the fully connected layer with the number of neurons 500, 200, and 20. ReLU activates each of these Fully Connected layers. After that we use Softmax, which is a function to take vector input and distribute the output probability.

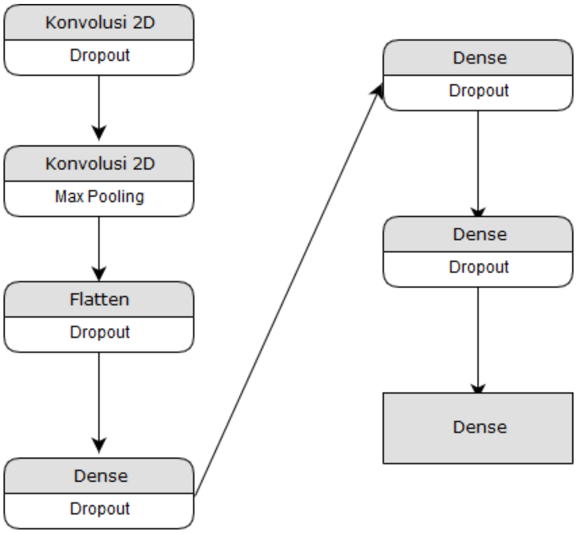

Fig. 6. The Layers in the Training Process.

The Javanese script image will be processed using seven network layers. Each operation will be carried out, such as the convolution process, the max-pooling process, the dense process, the flatten process, and the dropout process depicted in the diagram in Fig. 6.

Cases in this work are different from standard image classification—the difference in this work caused by the image data used on the dataset. Every image on the dataset only contains written Javanese alphabet objects with a simple edge or image feature, not as complex as the animal image object. The alphabet object class also only had a more straightforward form and pattern. As shown in Fig. 4, the image dataset of the Javanese language used in this work is a simple image containing one alphabet for each image class with a mostly white and clean background. The difference for each class is only the writing style and some lighting noise. The size and quality of the dataset also will be small. Applying the complex convolution layer used in public image classifying wouldn't suit the alphabet data image used in this work. It will cause failed feature extraction and overfitting models because of the burdensome extraction process. The solution to the problem, the small custom size of Convolutional Neural Network with sequential layer, will be used in this work; this approach also benefits from computational cost and time. Finally, other activation layers and different coats will be combined to get the right learning and fine-tuned models.

After selecting the final sequential layer, we decided to use CNN based deep learning network with a combination of convolution layer, dense layer, and some activation function and downsampling layer to train the dataset. The final system will consist of two convolution layers and two Dense layers with Rectified Linear Unit (ReLU) activation function, Flatten layer between them, Dropout process after each segment, and final dense layer a Softmax activation function. The whole sequential layer of the model is shown in Table I.

TABLE I. MODEL SUMMARY

Layer [(a)]	Output Shape [(b)]	Parameter
Convolution 2D (ReLU activation)	(None, 32, 32, 32)	320
Dropout	(None, 32, 32, 32)	0
Convolution 2D (ReLU activation)	(None, 32, 32, 32)	9248
Max Pooling	(None, 32, 16, 16)	0
Flatten	(None, 8192)	0
Dropout	(None, 8192)	0
Dense (ReLU activation)	(None, 500)	4096500
Dropout	(None, 500)	0
Dense (ReLU activation)	(None, 200)	100200
Dropout	(None, 200)	0
Dense (Softmax activation)	(None, 20)	4020
Total Parameter :		4210288

Layer listed on a sequential order

"None" on the first column shape means the batch size

Based on Table I, there will be 11 layers with 4210288 parameters used for training to get the final output model for the inferences process. The first two convolutional layers function as many other CNN model worked as the primary feature extraction. Due to the type of object on the image is a simple alphabet pattern, so we only use two convolutional layers with the ReLU activation function applied for each output of the layers. The next Flatten layer is to reshape the convolutional layers' output to have a more simplified linear shape to ease the classification. The following dense layers with the ReLU activation function will work like a fully connected layer and merge the output shape. The final dense layers with Softmax activation function used to convert the final output shape result to match the number of the class used in this classification (20 classes). The Dense layers in this model are like other traditional non-fully convolutional neural network layers, which serve the purpose of doing actual classification from the result convolution layer.

III. RESULT AND DISCUSSION

The CNN model adapted for classifying the Javanese language image in this work was evaluated and tested with 20 classes of Javanese language alphabet. The dataset contains 50 image data, with 80% of it used for training the model and 20% for testing the model. The training and testing process was done only with 2.7 GHz Intel i5-6198DU CPU, 8 GB of memory. The specification mentioned for the training and testing process is just a regular notebook specification without GPU acceleration. Using that specification proves that the adapted model did not need a vast resource of computation and capable of being adjusted on the devices with limited computation resources. Training and testing process of the adapted CNN layers are running with Keras framework with Theano backend on Python programming language.

As shown in Fig. 7, the proposed model evaluated the provided training and testing dataset; some metrics were recorded during the training and testing process. We record the time needed to finish the training per sequent of the epoch and the practice's final loss result during the training process. One epoch means feeding the whole image dataset to the network once. The right number of epochs will be needed to get the optimum model accuracy. The model accuracy also will be recorded during the testing process; this approach is needed to decide which final model will have the best accuracy.

```
def create_model(ep_model):
 epochs = ep_model
 lrate = 0.01
 decay = lrate / epochs
 model_javanese = Sequential()
 model_javanese.add(Conv2D(32, (3, 3), input_shape=(1, 32, 32),
activation='relu', padding='same'))
 model_javanese.add(Dropout(0.2))
 model_javanese.add(Conv2D(32, (3, 3), activation='relu',
padding='same'))
 model_javanese.add(MaxPooling2D(pool_size=(2, 2)))
 model_javanese.add(Flatten())
 model_javanese.add(Dropout(0.2))
 model_javanese.add(Dense(500, activation='relu',
kernel_constraint=maxnorm(3)))
 model_javanese.add(Dropout(0.2))
 model_javanese.add(Dense(200, activation='relu',
kernel_constraint=maxnorm(3)))
 model_javanese.add(Dropout(0.2))
 model_javanese.add(Dense(20, activation='softmax'))
 model_javanese.compile(loss='categorical_crossentropy',
optimizer='adadelta', metrics=['accuracy'])
 return model_javanese
```

Fig. 7. Sample Code for Create Model.

The final training and testing metrics show in Table II and Table III. The training process's total time in Table II shows linear results based on the number of epochs. The training process time gets longer to finish, and the loss value gets smaller for every more significant number of epochs. The training time gets more prolonged because of the more computation process needed for each more significant epoch. The average time needed for training the dataset per epoch is around 12.194 seconds. Based on the result, the training process time with only 12.194 seconds average per epoch is pretty much fast than other many CNN training processes, even with no accelerated hardware used. The light computation process mainly happens because of the simple model, with only 11 layers of computation assigned in this model. Although only using a simple model for training the model, the final result shows a fair number of accuracies. Table II shows various accuracy results for each model with different epoch numbers of training. The more significant number of epochs will result in better accuracy, shown on the resulting model with 5 to 50 epochs. After that, the 75 epoch results show that the accuracy gets the saturation point reduced. Overly high epoch on training may cause overfitting, which leads to stagnant or decreasing accuracy like what happens on these experimental results, so finding the epoch's right number is essential. Our model from the training also shows fast inference time when doing the classification. The average inference time with the same specification mentioned above is 0.57 seconds, and again the fast inference time is because of the simplicity of the model and dataset toolbar.

TABLE II. TIME REQUIRED FOR MODEL TRAINING

Epoch	**Time (s)**	**Loss**
5	65.04	2.566
10	123.13	1.4490
25	301.51	0.5596
50	639.85	0.3237
75	949.68	0.2882

TABLE III. TESTING THE ACCURACY FOR MODEL TRAINING

Epoch	Accuracy (c)
5	26.74%
10	57.44%
25	75.60%
50	86.68%
75	82.12%

Accuracy based on the selected 1000 images testing dataset

As shown in Table III, the overall result shows that the Javanese language image classifier can perform reasonably well with the proposed model based on the Convolutional Neural Network layers. The simplicity of the training model and dataset used in this work brings the advantage of computation weight and time. Although it had a simple model and dataset, the model's accuracy result is fair, with around 86.68% accuracy on a 50 training epoch model. This model's advantages with fast and light computation time bring the possibility to use this model on devices with limited computation resources such as mobile devices, familiar web server interface, and internet-of-things devices.

IV. CONCLUSIONS

In this study, five trials have been carried out, starting from at least five epochs, and the most are 75 epochs. From the five trials for the training time, it can be concluded that the more number of epochs conducted, the time needed for training is also longer with an increase in time by two times per trial. While the accuracy also has varying results, in epochs 5 to 50, the accuracy level increases. Whereas at 75 epochs, the level of accuracy has decreased by approximately 4%.

This study's results are a model with the best accuracy and time of 86.68% and 639.85 seconds using 1000 datasets and 50 epochs. Meanwhile, the smallest accuracy obtained in this study was 26.74% with five epochs.

This model's advantage is that the computation time is fast. The accuracy is relatively high, making it suitable for further development, especially implementing this model to a system that can recognize Javanese characters using a mobile or web platform.

REFERENCES

[1] A. Setiawan, A. S. Prabowo, dan E. Y. Puspaningrum, “Handwriting Character Recognition Javanese Letters Based on Artificial Neural Network,” Int. J. Comput. Netw. Secur. Inf. Syst. Vol, vol. 1, no. 1, hal. 39–42, 2019.

[2] G. H. Wibowo, R. Sigit, dan A. Barakbah, “Javanese Character Feature Extraction Based on Shape Energy,” Emit. Int. J. Eng. Technol., vol. 5, no. 1, hal. 154–169, 2017.

[3] M. L. Afakh, A. Risnumawan, M. E. Anggraeni, M. N. Tamara, dan E. S. Ningrum, “Aksara jawa text detection in scene images using convolutional neural network,” in Proceedings - International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2017, 2017, hal. 77–82.

[4] A. R. Widiarti dan P. N. Wastu, “Javanese character recognition using Hidden Markov Model,” World Acad. Sci. Eng. Technol., vol. 57, no. September 2009, hal. 261–264, 2009.

[5] G. S. Budhi dan R. Adipranata, “Handwritten Javanese Character Recognition Using Several Artificial Neural Network Methods,” J. ICT Res. Appl., vol. 8, no. 3, hal. 195–212, 2015.

[6] Y. Harjoseputro, Y. D. Handarkho, dan H. T. R. Adie, “The Javanese Letters Classifier With Mobile Client-Server Architecture And Convolution Neural Network Method,” Int. J. Interact. Mob. Technol., vol. 13, no. 12, hal. 67–80, 2019.

[7] C. K. Dewa, A. L. Fadhilah, dan A. Afiahayati, “Convolutional Neural Networks for Handwritten Javanese Character Recognition,” IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 12, no. 1, hal. 83–94, 2018.

[8] A. P. Ardhana, D. E. Cahyani, dan Winarno, “Classification of Javanese Language Level on Articles Using Multinomial Naive Bayes and N-Gram Methods,” J. Phys. Conf. Ser., vol. 1306, no. 1, hal. 1–9, 2019.

[9] Y. Harjoseputro, “Classifying Javanese Letters with Convolutional Neural Network (CNN) Method,” 2018, no. 1, hal. 52–59.

[10] S. J. D. Prince, “Computer vision: models, learning and inference,” in Cambridge University Press, 2012.

[11] K. Chauhan dan S. Ram, “Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures using Tensorflow and Keras,” Int. J. Adv. Eng. Res. Dev., vol. 5, no. 02, hal. 533–538, 2018.

[12] S. Albawi, T. A. Mohammed, dan S. Al-Zawi, “Understanding of a convolutional neural network,” Proc. 2017 Int. Conf. Eng. Technol. ICET 2017, vol. 2018-Janua, no. April 2018, hal. 1–6, 2018.

[13] Tutut Furi Kusumaningrum, “Implementasi Convolution Neural Network (CNN) Untuk Klasifikasi Jamur Konsumsi Di Indonesia Menggunakan Keras,” Univ. Islam Indones. Yogyakarta, vol. 10, no. 2, hal. 1–15, 2018.

[14] C. Szegedy et al., “Going Deeper with Convolutions,” in Designing and Tracking Knowledge Management Metrics, 2015, hal. 1–9.

[15] A. Krizhevsky, I. Sutskever, dan G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Adv. neural Inf. Process. Syst., hal. 1097–1105, 2012.

[16] K. He, X. Zhang, S. Ren, dan J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-Decem, hal. 770–778.

[17] K. Younis dan A. Khateeb, “Arabic Hand-Written Character Recognition Based on Deep Convolutional Neural Networks,” Jordanian J. Comput. Inf. Technol., vol. 3, no. 3, hal. 186, 2017.

[18] V. Sze, Y.-H. Chen, T.-J. Yang, dan J. S. Emer, “Efficient Processing of Deep Neural Networks: A Tutorial and Survey,” Proc. IEEE, vol. 105, no. 12, hal. 2295–2329, 2017.

[19] W. S. E. Putra, A. Y. Wijaya, dan R. Soelaiman, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” J. Tek. ITS, vol. 5, no. 1, hal. A65–A69, 2016.

[20] A. Dutt dan A. Dutt, “Handwritten Digit Recognition Using Deep Learning,” vol. 6, no. 7, hal. 990–997, 2017.

[21] G. Nguyen et al., “Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey,” Artif. Intell. Rev., vol. 52, no. 1, hal. 77–124, 2019.

[22] A. A. Kristanto, Y. Harjoseputro, dan J. E. Samodra, “Implementasi Golang dan New Simple Queue pada Sistem Sandbox Pihak Ketiga Berbasis REST API,” J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 4, no. 4, hal. 745–750, 2020.

[23] Y. Sugianela dan N. Suciati, “Javanese Document Image Recognition Using Multiclass Support Vector Machine,” CommIT (Communication Inf. Technol. J., vol. 13, no. 1, hal. 25, 2019.