TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 21, No. 2, April 2023, pp. 346~353
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v21i2.24266

Hybrid convolutional neural networks-support vector machine classifier with dropout for Javanese character recognition

Diyah Utami Kusumaning Putri[1], Dinar Nugroho Pratomo[2], Azhari[1]
[1]Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia
[2]Department of Electrical Engineering and Informatics, Vocational School, Universitas Gadjah Mada, Yogyakarta, Indonesia

Article Info

Article history:

Received Sep 24, 2021
Revised Mar 16, 2022
Accepted Jun 02, 2022

Keywords:

Convolutional neural networks
Dropout
Javanese character recognition
Multilayer perceptron
Support vector machine

ABSTRACT

This research paper explores the hybrid models for Javanese character recognition using 15600 characters gathered from digital and handwritten sources. The hybrid model combines the merit of deep learning using convolutional neural networks (CNN) to involve feature extraction and a machine learning classifier using support vector machine (SVM). The dropout layer also manages overfitting problems and enhances training accuracy. For evaluation purposes, we also compared CNN models with three different architectures with multilayer perceptron (MLP) models with one and two hidden layer(s). In this research, we evaluated three variants of CNN architectures and the hybrid CNN-SVM models on both the accuracy of classification and training time. The experimental outcomes showed that the classification performances of all CNN models outperform the classification performances of both MLP models. The highest testing accuracy for basic CNN is 94.2% when using model 3 CNN. The increment of hidden layers to the MLP model just slightly enhances the accuracy. Furthermore, the hybrid model gained the highest accuracy result of 98.35% for classifying the testing data when combining model 3 CNN with the SVM classifier. We get that the hybrid CNN-SVM model can enhance the accuracy results in the Javanese characters recognition.

Corresponding Author:

Diyah Utami Kusumaning Putri
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada, Yogyakarta, Indonesia
Email: diyah.utami.k@ugm.ac.id

1. INTRODUCTION

The Indonesian archipelago has many ethnic groups and a diversity of languages. Javanese tribble is the largest ethnic group and Javanese is the most widely spoken regional language in Indonesia. The Javanese language has its own traditional letter called Aksara Jawa or Javanese script [1]. Javanese script is a historic Javanese character that has been used by the Mataram Kingdom since the 17th century. Nowadays, the use of Javanese script just can be found in historical relics or wall cravings. Sometimes, it also can be used in place name signboards, street signboards, or decorations as the transcription for the Roman alphabet [2].

Recognizing Javanese script is difficult, this is due to the writing of each character being complex and some characters are almost similar, so it is more difficult to recognize. Furthermore, if the manuscript that will be recognized is hand-written because it was written by many different writers who have different writing styles [3]. Some Javanese are not able to write and read Javanese scripts, particularly adolescents. It is going to erase the presence of Javanese characters and have an effect on Javanese culture in general.

Therefore, to contribute to the preservation of Javanese script, we need a tool that has the ability to automatically recognize Javanese characters [4].

Machine learning has been used before as a solution to the recognition of handwritten characters. Popli *et al.* [5] recognized handwritten alphabets samples and classified them into one of the alphabet classes using machine learning models, i.e. ensemble learning, ensemble bagged trees, k-nearest neighbor (KNN), support vector machine (SVM) and Naïve Bayes. This research proposed a simplified methodology based on engineered features that are verified using the MatLab tool, then achieved the highest accuracy of 89.3% using ensemble subspace model 1.

There have been several kinds of research for recognizing Javanese characters using artificial neural networks (ANN) and deep learning techniques in order to get better accuracy. Dewa *et al.* [6] developed software that used the digital convolutional neural networks (CNN) method for classifying the segmented image of offline handwritten Javanese characters into 20 classes. In this research, CNN is compared to the multilayer perceptron (MLP). The results of experiments show that the CNN model outperforms MLP which achieves the highest accuracy of 89%. Fauziah *et al.* [7] also used the CNN method to classify 48 classes including Javanese script types, namely basic letters (*carakan*) and voice-modifying scripts (*sandhangan*). The CNN architecture consists of three convolution layers with max-pooling operations. This research used hyperparameters including the number of filters for each convolution layer 32, 64, or 128 filters, with a learning rate of 0.0001, a dropout value is 0.5, and the number of neurons in the fully-connected layer is 1,024 neurons. The average accuracy performance value was 87.65%, the average precision value was 88.01%, and the average recall value was 87.70%. Rismiyati *et al.* [8] performed CNN and deep neural network (DNN) for classifying 20 handwritten Javanese characters. The experiment used 2.470 images with an input image size is 32×32 pixels. The accuracy result with k-fold cross-validation obtained is 70.22% for CNN and 64.65% for DNN. Wibowo *et al.* [9] used the CNN method with two different numbers of layers and the dataset contains 11500 characters. The experimental results obtained that CNN has ensured to recognition of simple Javanese characters with a 94.57% accuracy score. Currently, the CNN model is a deep learning technique that is very powerful in solving classification problems with the input image. CNN model takes pixel neighbour information using extraction of feature task with convolution and pooling operation between a combination of many layers. Then, the features obtained are used to determine its class using the softmax activation.

Several other studies have been conducted to improve the performance of CNN, such as developing a hybrid model that integrates a CNN and support SVM [10]–[15]. In the hybrid model, CNN takes as a feature extractor and SVM performs as a classifier. This hybrid approach automatically extracts features from the input raw images using CNN and yields the predictions using SVM. Niu and Suen [10] experimented with the Modified National Institute of Standards and Technology (MNIST) digit database and compared the hybrid model with different studies on the equal database. The results imply that this hybrid model has accomplished better results. Ahlawat and Choudhary [11] additionally confirmed the effectiveness of hybrid CNN-SVM by producing an accuracy score of 99.28% on the recognition task using the MNIST handwritten digits dataset. Elleuch *et al.* [12] explored a new hybrid model CNN-SVM and applied the dropout technique for offline Arabic handwriting recognition. Simulation results proved that the novel CNN-SVM model with dropout shows extensively and efficiently better than the CNN-SVM architecture without dropout and the fundamental CNN classifier.

In this study, an architecture model using hybrid CNN-SVM with dropout for Javanese characters recognition has been offered to improve the accuracy score of character recognition. The focus of this research is to learn and extract the features from the raw images of Javanese characters using CNN. Then, these learned patterns are continued to the SVM classifier for executing the Javanese characters recognition. Dropout training is one of the effective approaches to manage overfitting issues via randomly ignoring subsets of features at each iteration of a training stage. The dropout layer will make the convergence speed different, weaken the effect of the initial parameters on the model, and enhance the training accuracy [16], [17]. For evaluation purposes, we additionally evaluated CNN models with three different architectures with MLP on the performance of accuracy and training time.

The rest of this paper is organized as: the basic concepts of CNN, SVM and the hybrid CNN-SVM model designed for Javanese characters recognition explained in section 2. Furthermore, section 3 presents our experimental method. Then, the experimental results are given and analyzed in section 4. Lastly, section 4 concludes some remarks and explains the future scope.

2. RESEARCH METHODS

The research methods employed in this work are outlined in this section. We provide an overview of the CNN and SVM models. Then, we explain our proposed model, the hybrid CNN-SVM.

2.1. Convolutional neural networks

A convolutional neural network model proposed by LeCun *et al.* [18] can be considered as an elaboration for conventional ANN models, such as MLP. A CNN model is arranged of certain layers which are called convolution and pooling layers to learn and extract features from raw image input and a fully-connected neural network (FCN) which is actually an MLP model to predict the output class. The features obtained from those special layers are called feature maps and become inputs for a fully-connected layer (FCL) [6].

Figure 1 represents an example of CNN architecture that consists of a set of many layers. To begin, the input is convoluted with a set of filters (with C hidden layers) in the convolution layer to get the feature maps value. Next, the dimensionality (with S hidden layers) of the spatial resolution of the feature maps is reduced, every convolution layer is continued to a subsampling (or pooling) layer. Convolutional layers alternate subsampling layers denote as the feature extractor to retrieve discriminating features from the input images. In the end, a flattened function is implemented to transform each feature map into a one-dimensional matrix. Then, these matrices will be further passed into the output layer which is the FCL or MLP with a softmax activation function that generates possibilities of each class of the input image [6], [10], [12]. The various combinations of the numbers of hidden layers, epochs and architectures of CNN may produce different performances [19]–[21].

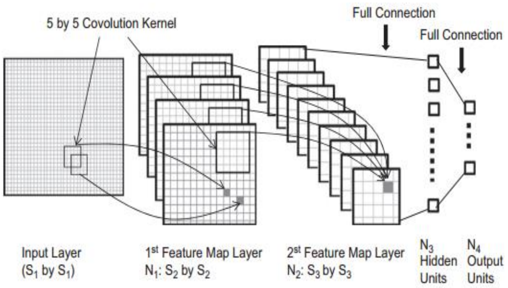

Figure 1. An example of CNN model architecture [10]

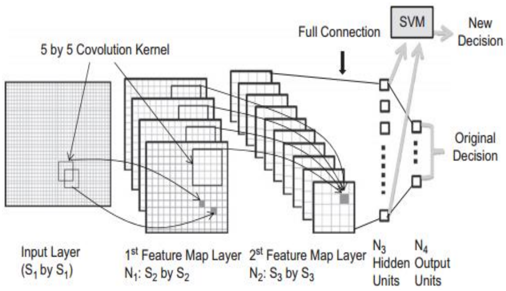

Figure 2. An example of hybrid CNN-SVM model architecture [10]

2.2. Support vector machine

Support vector machine, which has been proposed by Vapnik [22], Cortes and Vapnik [23] is a robust discriminative classifier. SVM is assumed to be a sophisticated tool to accomplish linear and also non-linear classification issues with flexibility, stinginess, prediction ability and the global optimum solution. Difference from ANN that minimizes the empirical risk, the foundation of SVM formulation is the minimization of structural risk [22].

Kernel functions in the SVM model can convert a nonlinear into a linear problem by transforming data into high-dimensional feature spaces and finding the best hyperplane to separate the features. The modification was conducted using various kernel functions such as sigmoid, linear, polynomial, and radial basis function (RBF) kernels. The best hyperplane is reached by solving a quadratic programming problem that depends on parameters of regularization [22], [23].

2.3. Hybrid CNN-SVM

The hybrid CNN–SVM architecture combines the CNN model and SVM classifier. A CNN has a supervised learning mechanism that includes convolution, subsampling (pooling), and fully connected layers. CNN can learn invariant local features conveniently and extract the most discriminating features from pixel image patterns. Furthermore, the SVM classifier can turn down the generalization error on invisible data. SVM intends to represent the dataset features into multi-dimensional feature spaces where an optimal hyperplane splits the features of image data belonging to variant classes. This model works by replacing the latest output layer with the SVM classifier. In this model, CNN becomes a feature extractor and SVM as a classifier and substitutes the softmax layer of CNN. Thus, the output of the hidden layer result can be assumed as the input features for the SVM classifier [12].

Figure 2 shows an example of the hybrid CNN–SVM model. First, the raw images are continued to the input layer and are pursued by the CNN model is trained until the training process converges with several iterations or epochs. Next, the SVM model with kernel function substitutes the output layer of the CNN model. The SVM uses the outputs from the last hidden layer of CNN as a new feature vector for the input training process. After that, the SVM classifier has been trained well, it performs the recognition task and produces new determinations to predict the classes on testing image datasets [10].

3. EXPERIMENTS

The objective of this study is to accomplish the task of Javanese character recognition and improve its performance. In order to achieve the objective, we investigated three CNN architectural variations and a hybrid CNN-SVM model with dropout. The focus of this research is to extract the patterns from the raw images of handwritten Javanese characters using CNN. Then, these learned features proceeded to the SVM classifier to perform the Javanese characters recognition. Dropout is also used to control overfitting. We also compared CNN models with three different architectures with MLP models with one and two hidden layer(s) for evaluation purposes. We evaluated all models on both the accuracy of classification and training time.

The details of experiments conducted in this paper are described in this section. The experiments section includes data acquisition for digital fonts and handwritten Javanese characters, the architecture of CNN models and compared MLP models, the SVM hyperparameters, and also experiment scenarios for Javanese characters recognition. Each part of the experiment will be presented in the subsection parts.

3.1. Data acquisition

The data is Javanese characters acquired from digital fonts and handwritten texts which are scanned into documents. The Javanese digital fonts were gathered from 10 different Javanese fonts with normal, bold, and italic text converted into a document file. Furthermore, the collection of handwriting Javanese script data was carried out using different pen thicknesses written by Javanese people, then we scanned the handwritten texts and converted them into a document file. After that, all Javanese documents are segmented into characters. A total of 100 sets of Javanese handwriting scripts had been acquired yielding 12000 characters (120 characters per set), besides, we also collected 30 sets of digital Javanese text from digital Javanese fonts, resulting in 3600 characters.

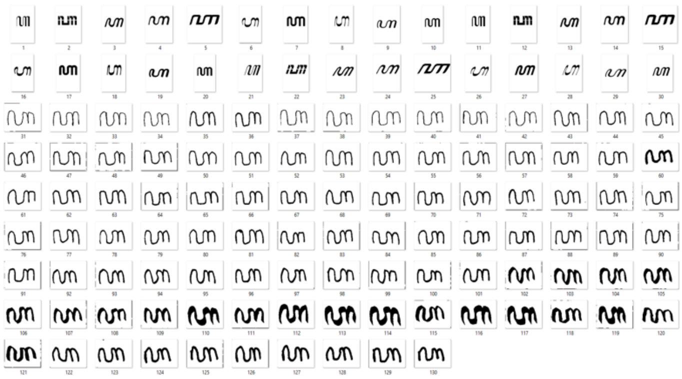

Figure 3. Sample of digital (from 1 until 30) and handwritten (from 31 until 130) "HA" Javanese character after preprocessing methods applied

A total of 15600 Javanese characters obtained from digital (30 sets × 120 characters) and handwritten (100 sets × 120 characters) resources. The characters obtained from the hand-writing process can be noisy, not aligned, slightly blurry (because of the pen inks), and so on must be enhanced. Hence, some image enhancement techniques have been adjusted to all the original Javanese characters images in the pre-processing step. Thus, the handwritten dataset gathered is clean and can be used robustly. Every data in the dataset is converted and normalized into an 8-bit grayscale image which has a fixed image of the size of 28×28 pixels and is positioned at the center. Figure 3 displays samples of digital (from 1 until 30) and handwritten (from 31 until 130) "HA" Javanese characters after preprocessing methods were applied. This dataset will be split into training, validation and testing datasets.

3.2. The architecture of CNN models

The CNN model consists of several layers with convolution and subsampling layers and the FCN layer as an output layer with a softmax function. Xavier weight initialization, also known as Glorot uniform initializer was used to initialize weight neurons in the CNN model. Adam optimizer was used in this work to obtain optimized performance. We adjusted dropout regularization and also the rectified linear unit (ReLU) activation function to entire layers in the CNN model. The implementation of the CNN model using a deep learning library in python language, tensorflow keras library [24], [25]. For evaluation objectives, we also compared CNN models with three different architectures with MLP models with one and two hidden layer(s). The details of different CNN architectures utilized in this research are presented in Table 1.

Table 1. The details of different CNN model architectures

Model architecture	Layer	Size	Output shape
Architecture of model 1 CNN	Input	(1, 28, 28)	-
	Conv + ReLU	32 (3×3) filters	(28, 28, 32)
	MaxPooling + Dropout (0.2)	(2×2)	(14, 14, 32)
	Conv + ReLU	64 (2×2) filters	(14, 14, 64)
	MaxPooling + Dropout (0.2)	(2×2)	(7, 7, 64)
	Conv + ReLU	128 (3 × 3) filters	(7, 7, 128)
	MaxPooling + Dropout (0.2)	(2×2)	(4, 4, 128)
	FullyConnected + ReLu + Dropout (0.2)	1000 neurons	1000
	FullyConnected	120 neurons	120
	Softmax	120 way	120
Architecture of model 2 CNN	Input	(1, 28, 28)	-
	Conv + ReLU	32 (5×5) filters	(28, 28, 32)
	MaxPooling + Dropout (0.2)	(2×2)	(14, 14, 32)
	Conv + ReLU	64 (5×5) filters	(14, 14, 64)
	MaxPooling + Dropout (0.2)	(2×2)	(7, 7, 64)
	Conv + ReLU	128 (5×5) filters	(7, 7, 128)
	MaxPooling + Dropout (0.2)	(2×2)	(4, 4, 128)
	Conv + ReLU	64 (5×5) filters	(4, 4, 64)
	MaxPooling + Dropout (0.2)	(2×2)	(2, 2, 64)
	Conv + ReLU	32 (5×5) filters	(2, 2, 32)
	MaxPooling + Dropout (0.2)	(2×2)	(1, 1, 32)
	FullyConnected + ReLu + Dropout (0.2)	1024 neurons	1024
	FullyConnected	120 neurons	120
	Softmax	120 way	120
Architecture of model 3 CNN	Input	(1, 28, 28)	-
	Conv + ReLU	64 (3×3) filters	(28, 28, 64)
	Conv + ReLU	64 (3×3) filters	(28, 28, 64)
	MaxPooling + Dropout (0.2)	(2×2)	(14, 14, 64)
	Conv + ReLU	128 (3×3) filters	(14, 14, 128)
	Conv + ReLU	128 (3×3) filters	(14, 14, 128)
	MaxPooling + Dropout (0.2)	(2×2)	(7, 7, 128)
	Conv + ReLU	256 (3×3) filters	(7, 7, 256)
	Conv + ReLU	256 (3×3) filters	(7, 7, 256)
	Conv + ReLU	256 (3×3) filters	(7, 7, 256)
	MaxPooling + Dropout (0.2)	(2×2)	(4, 4, 256)
	Conv + ReLU	512 (3×3) filters	(4, 4, 512)
	Conv + ReLU	512 (3×3) filters	(4, 4, 512)
	Conv + ReLU	512 (3×3) filters	(4, 4, 512)
	MaxPooling + Dropout (0.2)	(2×2)	(2, 2, 512)
	Conv + ReLU	512 (3×3) filters	(2, 2, 512)
	Conv + ReLU	512 (3×3) filters	(2, 2, 512)
	Conv + ReLU	512 (3×3) filters	(2, 2, 512)
	MaxPooling + Dropout (0.2)	(2×2)	(1, 1, 512)
	FullyConnected + ReLu + Dropout (0.2)	4096 neurons	4096
	FullyConnected + ReLu + Dropout (0.2)	4096 neurons	4096
	FullyConnected	120 neurons	120
	Softmax	120 way	120

In this experiment, we also trained MLP models with single and two hidden layer(s) with the same Javanese characters dataset. The detailed architecture of MLP models is given in Table 2. In this research, we utilized pixel values of the Javanese character images as inputs for MLP without the feature extraction approach previously.

The MLP architecture with a single hidden layer has 784 neurons in the input layer as Table 2. Each neuron will accept a single vector as the result of extracted features from a character image that has a size of 28×28 pixels. The MLP architecture with a single hidden layer consists of one hidden layer with 1,000 neurons. Then, the MLP model will generate 120 probability values to which the input image class may belong. Furthermore, the two hidden layers MLP model utilized 1,000 neurons and 2,000 neurons. MLP models used the rectified linear unit (ReLU) in hidden layers. Moreover, the softmax activation functions are also applied in the output layer, respectively.

3.3. The SVM hyperparameters

The generated features from the CNN model are continued to the SVM classifier for training, validating and then testing the Javanese images. One hundred and twenty values from the last layer of the trained CNN model were used as a new feature vector to denote the input matrix and were passed to the SVM for learning, validation, and testing. The parameters of SVM like kernel function, C parameter (regularization parameter) and gamma parameter are tuned accurately because they are the affecting parameters during SVM classification.

The kernel functions are observed in SVM using linear, sigmoid, polynomial, and RBF kernels. The values of the gamma parameter explored are 0.01, 0.001, 0.0001, 0.00001. Furthermore, the values of the C parameter observed are 1, 10, 100, 1000. The parameters used to apply the SVM method are optimized by a 5-fold cross-validated grid-search over a parameter grid.

3.4. Experiment scenario

A CNN model uses a supervised learning scenario to update internal weight value matrices during the training process, the model uses a lost function that calculates the difference between the predicted class and the actual class. CNN models utilize the lost (or cost) function as the cross-entropy error formula. The scenario of our experiment was started with the dataset which was separated into 80% for training and the rest for testing to perform the model development and evaluation. Then, the training dataset was split into 80% for training and the rest for the validation dataset. In our experiment, we use 128 as the batch size, the maximum number of epochs 1000 and a learning rate of 0.0001. We used 4 × NVIDIA Tesla V100 DGXS GPU for training the model.

Table 2. The details of different multilayer perceptron model architectures

Model architecture	Layer	Size	Output shape
The MLP model architecture with single hidden layer	Input	784 neurons	-
	FullyConnected + ReLu	1000 neurons	1000
	FullyConnected	120 neurons	120
	Softmax	120 way	120
The MLP model architecture with two hidden layers	Input	784 neurons	-
	FullyConnected + ReLu	1000 neurons	1000
	FullyConnected + ReLu	2000 neurons	2000
	FullyConnected	120 neurons	120
	Softmax	120 way	120

4. RESULTS AND DISCUSSION

Three different CNN architectures are compared with the MLP model with single and two hidden layer(s). We used three different variables to evaluate the performances: validation accuracy, testing accuracy and training time results. Table 3 presents the experiment results of those five different models.

Table 3 shows that MLP with a single hidden layer takes minimal training time among others. Moreover, model 3 CNN needs more training time than other CNN models. It is caused by the complex architectures of convolution and subsampling layers, thus increasing the computation and training time. Overall, the classification accuracies of all CNN models exceed the classification accuracies of both MLP models for validation and testing datasets. Using of convolutional and pooling layers in the CNN model can effectively learn the features of the Javanese characters dataset. The highest validation and testing accuracies from this experiment were gathered when we used model 3 CNN, which are 97.14% and 98.06%, respectively. We can also learn that the increment of hidden layers to the MLP model slightly enhances its performance.

In the next experiment, we built and trained a hybrid CNN-SVM model. An SVM classifier changed the last fully connected layer of CNN to produce classes of the characters. We used different kernel functions and determined the optimal value of gamma and C parameters to build the SVM in the hybrid model by applying the 5-fold cross-validation scenario on the training dataset using grid-search. Table 4 provides the results of the hybrid CNN-SVM model using three different CNN architectures.

Table 3. The results of CNN and MLP models

Model	Validation accuracy	Testing accuracy	Training time
Model 1 CNN	89.9%	90.1%	98.979
Model 2 CNN	94.75%	95.45%	251.456
Model 3 CNN	97.14%	98.06%	379.195
MLP with one hidden layer	80.3%	81.2%	24.987
MLP with two hidden layers	81.7%	82.3%	32.017

Table 4. The results of the hybrid CNN-SVM models

Model	Validation accuracy	Testing accuracy	Training time
Model 1 CNN + SVM	90.99%	91.86%	445.263
Model 2 CNN + SVM	95.22%	95.57%	600.764
Model 3 CNN + SVM	97.38%	98.35%	827.896

After examining the results presented in Table 3 and Table 4, the accuracy of the hybrid CNN-SVM model for recognition of Javanese characters is outperformed by the accuracy of the basic CNN model. The highest testing accuracy is 98.35% when combining model 3 CNN with SVM classifier. The training time of the hybrid CNN-SVM increases because searching the best parameters of SVM using grid search requires more time.

5. CONCLUSION

This study aims to accomplish the task of recognising Javanese characters and improving their performance. A total of 15600 Javanese characters were gathered from digital and handwritten sources. In order to achieve the objective, we investigated three variants of CNN architectures and a model of hybrid CNN-SVM with dropout. The focus of this research is to involve feature extraction using CNN and then predict the output using an SVM classifier. The model combines the advantage of deep learning CNN and a machine learning classifier using SVM in recognizing Javanese characters. Dropout training with the dropout layer is one of the powerful ways to manage overfitting problems and enhance training accuracy. We also compare CNN models with three different architectures with multilayer perceptron MLP models with one and two hidden layer(s). In this research, we assessed all models on both classification performance and training time.

The experimental outcomes showed that the classification performances of all CNN models outperform the classification performances of both MLP models for validation and testing datasets. The highest testing accuracy using basic CNN is 98.06% which used model 3 CNN. The increment of hidden layers to the MLP model slightly enhances the accuracy. Furthermore, the proposed model achieved the highest accuracy of 98.35% for testing data when combining model 3 CNN with SVM classifier. The CNN-based-SVM model is a promising classification method in character recognition research. For the training time, MLP with a single hidden layer needs minimal training time among other models. Moreover, CNN models require more significant time for training compared to MLP. The training time of hybrid CNN-SVM also increases because searching the best parameters of SVM using grid search requires more time.

The character recognition research using a hybrid CNN-SVM model can improve further. In future research, our proposed model can be enhanced to recognise digital and handwritten characters in different languages such as Japanese, Korean, Bengali, Hindi, and so on. The other optimizing techniques can also be explored to elevate the overall performance of classification. Different architectures of hybrid CNN such as CNN-RNN, CNN-HMM can be explored. Evolutionary algorithms also can be investigated for enhancing CNN learning parameters, i.e. the number of layers and/or neurons, learning rate, kernel size of convolution filters.

ACKNOWLEDGEMENTS

This work was supported in part by a grant from Direktorat Penelitian Universitas Gadjah Mada [grants number: 2403/UN1.P.III/DIT-LIT/PT/2020].

REFERENCES

[1] M. L. Afakh, A. Risnumawan, M. E. Anggraeni, M. N. Tamara, and E. S. Ningrum, "Aksara jawa text detection in scene images using convolutional neural network," in *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2017, pp. 77–82, doi: 10.1109/KCIC.2017.8228567.

[2] G. S. Budhi and R. Adipranata, "Comparison of bidirectional associative memory, counterpropagation and evolutionary neural network for Java characters recognition," in *2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA)*, 2014, pp. 7–10, doi: 10.1109/ICAICTA.2014.7005906.

[3] A. R. Widiarti, A. Harjoko, Marsono, and S. Hartati, "Line segmentation of javanese images of manuscripts in javanese scripts," *International Journal of Engineering Innovation and Research*, vol. 2, no. 3, pp. 239–244, 2013. [Online]. Available: https://core.ac.uk/download/pdf/45364081.pdf

[4] C. A. Sari, M. W. Kuncoro, D. R. I. M. Setiadi, and E. H. Rachmawanto, "Roundness and eccentricity feature extraction for javanese handwritten character recognition based on k-nearest neighbor," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2018, pp. 5–10, doi: 10.1109/ISRITI.2018.8864252.

[5] R. Popli, I. Kansal, A. Garg, N. Goyal, and K. Garg, "Classification and recognition of online hand-written alphabets using Machine Learning Methods," in *IOP Conference Series: Materials Science and Engineering*, 2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012111/meta

[6] C. K. Dewa, A. L. Fadhilah, and A. Afiahayati, "Convolutional neural networks for handwritten javanese character recognition," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 12, no. 1, 2018, doi: 10.22146/ijccs.31144.

[7] Y. Fauziah, K. Aprilianta, and H. C. Rustamaji, "Convolutional Neural Network Method for Classification of Syllabes in Javanese Script," *International Journal of Artificial Intelligence & Robotics (IJAIR)*, vol. 3, no. 2, pp. 80-89, doi: 10.25139/ijair.v3i2.4395.

[8] Rismiyati, Khadijah, and A. Nurhadiyatna, "Deep learning for handwritten Javanese character recognition," in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, 2017, pp. 59–64, doi: 10.1109/ICICOS.2017.8276338.

[9] M. A. Wibowo, M. Soleh, W. Pradani, A. N. Hidayanto, and A. M. Arymurthy, "Handwritten javanese character recognition using descriminative deep learning technique," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2017, pp. 325–330, doi: 10.1109/ICITISEE.2017.8285521.

[10] X. -X. Niu and C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012, doi: 10.1016/j.patcog.2011.09.021.

[11] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM classifier for handwritten digit recognition," *Procedia Computer Science*, vol. 167, pp. 2554–2560, 2020, doi: 10.1016/j.procs.2020.03.309.

[12] M. Elleuch, R. Maalej, and M. Kherallah, "A new design based-SVM of the CNN classifier architecture with dropout for offline arabic handwritten recognition," *Procedia Computer Science*, vol. 80, pp. 1712–1723, 2016, doi: 10.1016/j.procs.2016.05.512.

[13] Y. Wang *et al.*, "Improvement of MNIST image recognition based on CNN," *IOP Conference Series: Earth and Environmental Science*, vol. 428, 2020, doi: 10.1088/1755-1315/428/1/012097.

[14] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved handwritten digit recognition using convolutional neural networks (CNN)," *Sensors*, vol. 20, no. 12, 2020, doi: 10.3390/s20123344.
[15] A. A. A. Ali and S. Mallaiah, "Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3294–3300, 2022, doi: 10.1016/j.jksuci.2021.01.012.
[16] A. F. M. Agarap, "An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification," *arXiv*, 2017. [Online]. Available: https://arxiv.org/pdf/1712.03541.pdf
[17] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. N. R. S. N. Prudhvith, and N. Kumar, "Image classification using SVM and CNN," in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, 2020, pp. 1–5, doi: 10.1109/ICCSEA49143.2020.9132851.
[18] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," *Shape, Contour and Grouping in Computer Vision*, Berlin, Heidelberg: Springer, 1999, vol. 1681, pp. 319–345, doi: 10.1007/3-540-46805-6_19.
[19] F. Siddique, S. Sakib, and M. A. B. Siddique, "Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, 2019, pp. 541–546, doi: 10.1109/ICAEE48663.2019.8975496.
[20] M. A. Hossain and M. M. Ali, "Recognition of Handwritten Digit using Convolutional Neural Network (CNN)," *Global Journal of Computer Science and Technology:(D) Neural & Artificial Intelligence*, vol. 19, no. 2, 2019, doi: 10.34257/GJCSTDVOL19IS2PG27.
[21] H. A. Alwzwazy, H. M. Albehadili, Y. S. Alwan, and N. E. Islam, "Handwritten digit recognition using convolutional neural networks," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 2, pp. 1101–1106, 2016. [Online]. Available: https://www.researchgate.net/publication/316938847_Handwritten_Digit_Recognition_Using_Convolutional_Neural_Networks
[22] V. N. Vapnik, *Statistical learn theory*. New York: John Wiley, 1998. [Online] Available: https://www.wiley.com/en-us/Statistical+Learning+Theory-p-9780471030034
[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.
[24] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017. [Online]. Available: https://www.perlego.com/book/527043/deep-learning-with-keras-pdf
[25] Chollet F., *Keras*. Github, 2015. [Online]. Available: https://github.com/fchollet/keras

BIOGRAPHIES OF AUTHORS

Diyah Utami Kusumaning Putri received B.Sc. and M.Cs. degress in Computer Science from Universitas Gadjah Mada, Yogyakarta, Indonesia in 2016 and 2019, respectively, and received the M.Sc. degree in Computer Science according to the dual degree program between Universitas Gadjah Mada, Yogyakarta, Indonesian and National Central University, Taiwan, in 2019. She became lecturer in Department of Computer Science and Electronics, Universitas Gadjah Mada in 2020 and joined Intelligent System Laboratory. Her research interests are deep learning, computer vision, data science and natural language processing. email: diyah.utami.k@ugm.ac.id.

Dinar Nugroho Pratomo received B.Sc. degree in Informatics Engineering from Universitas Jenderal Soedirman, Purwokerto, Indonesia in 2016, the M.Sc. degree in Computer Science from Universitas Gadjah Mada, Yogyakarta, Indonesia in 2019, and received the M.IM. degree in Information Management according to the dual degree program between Universitas Gadjah Mada, Yogyakarta, Indonesian and National Taiwan University of Science and Technology, Taiwan, in 2019. He became a lecturer in Department of Electrical Engineering and Informatics, Vocational School, Universitas Gadjah Mada in 2020. His research interests include machine learning, software engineering, and data science.
email: dinar.nugroho.p@ugm.ac.id.

Azhari received B.Sc. degree in Statistics from Universitas Gadjah Mada, Yogyakarta, Indonesia, the M.T. degree in Software Engineering from Institut Teknologi Bandung, Bandung, Indonesia, and the Ph.D. degree in Computer Science from Universitas Gadjah Mada, Yogyakarta, Indonesia. He is currently Associate Profressor and Head of Computer Science Bachelor Program in Department of Computer Science and Electronics, Universitas Gadjah Mada. His research interests are software engineering, intelligent agent, and information technology project management. He can be contacted at email: arisn@ugm.ac.id.