



A Comparative Study of Javanese Script Classification with GoogleNet, DenseNet, ResNet, VGG16 and VGG19

Ajib Susanto^{1*}, Christy Atika Sari², Eko Hari Rachmawanto³, Ibnu Utomo Wahyu Mulyono⁴,
Noorayisahbe Mohd Yaacob⁵

^{1,2,3,4}Study Program of Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia

⁵Center of Software Technology and Management (SOFTAM), Faculty of Information Science and Technology, University Kebangsaan Malaysia, Selangor, Malaysia

Abstract.

Purpose: Javanese script is a legacy of heritage or heritage in Indonesia originating from the island of Java needs to be preserved. Therefore, in this study, the classification and identification process of Javanese script letters will be carried out using the CNN method. The purpose of this research is to be able to build a model which can properly classify Javanese script, it can help in the process of recognizing letters in Javanese script easily.

Methods: In this study, the Javanese script classification process has been used the transfer learning process of Convolutional Neural Network, namely GoogleNet, DenseNet, ResNet, VGG16 and VGG19. The purpose of using transfer learning is to improve the sequential CNN model, processing can be better and optimal because it utilizes a previously trained model.

Result: The results obtained after testing in this study are using the transfer learning method, the GoogleNet model gets an accuracy of 88.75%, the DenseNet model gets an accuracy of 92%, the ResNet model gets an accuracy of 82.75%, the VGG16 model gets an accuracy of 99.25% and the VGG19 model gets an accuracy of 99.50%.

Novelty: In previous studies, it is still very rare to discuss the Javanese script classification process using the CNN transfer learning method and which method is the most optimal for performing the Javanese script classification process. In this study, it had been resulted find an effective method to be able to carry out the Javanese script classification process properly and optimally.

Keywords: Javanese script, GoogleNet, DenseNet, ResNet, VGG, CNN

Received September 2023 / **Revised** November 2023 / **Accepted** December 2023

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



INTRODUCTION

Javanese language is one of the oldest heritage languages in Indonesia [1]. Javanese script is a character in the form of ancient script or letters used on the island of Java [2] Because of this, Javanese script is a relic or cultural heritage is very valuable and priceless [3]. Therefore, Javanese Script is important to preserve, one way is to make Javanese as a local content subject in elementary to high school [4]. In the Javanese language lessons, there are lessons about Javanese script, but because of the complexity and diversity of Javanese script, students are still not optimal in mastering Javanese script both in reading and writing [5]. This makes students quickly bored in learning Javanese script [6]. Because of this, technological assistance is needed, students can easily understand and learn about Javanese characters [7], and the coordination and important role of education personnel is needed to be able to provide fun and effective learning for students [8].

Deep learning is a form of development of machine learning [9]. Sarker [10], argues in the current era, deep learning is one of the popular topics to be discussed, especially in the fields of Artificial Intelligence, Machine Learning and data analysis. This can happen, because deep learning is very good at processing big data [11], which is now widely used. Deep learning is also widely used now because it is increasingly facilitated by the performance of CPU and GPU computers which are getting faster which can speed up the process of both training and testing models [12]. In its implementation, deep learning uses the concept of artificial neural network [13] which is inspired by the workings of the biological neural system of the human

*Corresponding author.

Email addresses: ajib.susanto@dsn.dinus.ac.id

DOI: [10.15294/sji.v11i1.47305](https://doi.org/10.15294/sji.v11i1.47305)

brain [14], [15]. Therefore, deep learning is widely used in image processing, sound processing [16], automation in industry, automatic control systems, and others [15]. Deep learning is widely used in various fields, because in deep learning, it can automatically perform the process of learning features from data [17], without having to process the data first.

Therefore, when compared with machine learning methods, deep learning methods can provide good accuracy and speed performance [18], [19]. The method can be used to perform deep learning is Convolutional Neural Network (CNN). This method can perform object recognition and detection in a digital image [20]. Yadav et. al [21], suggested CNN is widely used and produces good accuracy in the image classification process since 2012. Because of this, CNN is known as the best solution for solving complex problems such as image processing and IOT [22], which can search for patterns from data well [23]. In its implementation, building a CNN model requires layers. The layers are usually used are input layer, convolution layer, pooling layer, fully connected layer and output layer [24]. Input layer is a layer reads data [25]. Convolution layer is a layer used to extract features [26]. Pooling layer is a layer used to improve the classification process by reducing the dimensions of the image generated from the previous process, the next process can be faster [27]. Fully connected layer is a layer used to perform the classification process on an image based on existing classes [28]. And the output layer is a layer used to display the classification results of the process on the previous layer [29].

Research on the identification and classification of Javanese script letters is an interesting research to do. There have been several studies conducted to classify Javanese alphabets. However, the research conducted still does not get perfect accuracy. As conducted in research by Rasyidi et. al [1], [30] in this study, the Javanese script letter classification process obtained an accuracy, precision and recall value of 97.7%. This value is not a bad value, but it can still be improvised, the value is close to perfect, therefore, this research is carried out, it is expected to increase the accuracy of the tests carried out. In research conducted by Das et. al [31] in 2019, discussed about classification of brain tumors using the CNN method. The purpose from this research is to build model can accurately classify brain tumors. The results obtained from this research is getting a brain tumor prediction accuracy of 94.39%.

Research conducted by Chaganti et. al [32], discusses the classification process of Dalmatian, Pizza, Dollar Ball, Soccer Ball and Sunflower images using CNN. The purpose conducting this research is to compare the machine learning process SVM and deep learning CNN to classify images. The results obtained from this study are after the classification process using SVM, getting a test accuracy of 82% while when the classification process using CNN, getting an accuracy of 93.57%. In research conducted by Jasim et al [33], discuss about process of classification and detection of diseases in leaves using deep learning. The purpose of this research is to build a model can accurately predict and classify leaf diseases. The results obtained from this study are by using deep learning get the prediction test accuracy of 98%. Research conducted by Badza et. al [34], discuss about process of classifying brain tumors based on MRI images using CNN. The purpose of this research is to build a model can accurately classify brain tumors using MRI images. The results obtained in this study are after testing with the model built to classify brain tumors, the test accuracy is 97.15% with an average precision of 97.15%, an average recall of 97.82% and an average f1-score of 97.47%.

Here, the classification process of Javanese script letters has been used CNN method. The CNN method has been used is the transfer learning method where in the transfer learning process, a previously trained model would be used again to perform new classification tasks [29]. The CNN transfer learning method used in this study is GoogleNet, ResNet, DenseNet, VGG16 and VGG19 [35] architectures. The purpose of using all of transfer learning model is because these models can be used to capture hierarchical features inside in the images. The purpose of using the transfer learning process is so also for model training process can be faster and accurate for performing the calcification process. Then, purpose of using the five transfer learning architectures is because the five architectures have similarities and good performance to be able to classify images which are explained in the research methods chapter. The purpose of this research is to build a model can be used to classify Javanese letters accurately which can help in the process of identifying Javanese letter.

METHODS

In this research, we will use the publicly available Javanese script dataset which obtained from the kaggle.com website. In the dataset would be used, there are 20 Javanese script letter classes consisting of ba, ca, da, dha, ga, ha, ja, ka, la, ma, na, nga, nya, pa, ra, sa, ta, tha, wa, ya. The dataset used is obtained from the kaggle.com website and after the dataset is obtained, the dataset is divided into 60% training data, 10% validation data and 30% testing data using the train test split module from sklearn, later the data can be used for deep learning model building. Figure 1 shows the Javanese script letters had been used in this study. The purpose of dividing into training data and testing data, the trained model does not experience overfitting because the data used from training and testing data is a different type of data, by using testing data, the model can carry out the classification process with new data. Each letter shown in Figure 1 represents 1 class, so there is a representative visualization of 20 classes had been used in this study. In this research, used a total of 2000 images, each class had 100 images. The 2000 classes are divided into 80% train data and 20% test data. The distribution of data becomes 1600 train images and 400 test images. Which in the train data will be divided again, it becomes 70% train data and 10% valid data. The distribution of data becomes as much as 1400 train data, 200 valid data and 400 test data.



Figure 1. Javanese script letter

In this research, the Javanese script classification process uses the CNN method. The advantage of using CNN is because CNN can perform feature extraction automatically [36], making it easier to compute. The CNN method had been used in this research is the CNN transfer learning method. Transfer learning is a method to use the model had already train to used again for new process [37]. Rahman et. al [38], suggested transfer learning is used so the training process not require a large dataset for train because the model has previously been trained and can cut training time more efficiently, as implemented in this article using GoogleNet, DenseNet, ResNet, VGG16 and VGG19 all model is trained under the same dataset is Imagenet dataset [39]. GoogleNet is model developed using a high feature representation trained using millions of images of everyday objects contained in the imaginet dataset [40]. Therefore, this model already had lot of knowledge to able to perform recognition on images with various levels of complexity and various scales and resolutions. DenseNet is a model developed by utilizing very strong layer connections, all layers have direct connections to each other [41]. ResNet is a model avoids gradient vanishing in the process by using the block residual method [42]. In the block, the input will be taken in a skip connection or shortcut connection, where in the skip connection, the returned gradient will not be lost because part of the input will be added to the output. This also keeps the ResNet model from overfitting because it is possible for the model to remember the features in the input. VGG16 is a model built using the overfit dataset consisting of 1.2 million images with 1000 classes used during model training [30], which consists of 13 convolutional layers, 3 fully connected layers for classification [43].

$$P = \frac{TrPos}{TrPos + FalPos} \quad (1)$$

$$R = \frac{TrPos}{TrPos + FalNeg} \quad (2)$$

$$F1 = 2 * \frac{P * R}{P + R} \quad (3)$$

The Javanese script classification process using the CNN transfer learning method is divided into 3 processing, namely the training, validation, and testing process. The training process is to make the model able to recognize patterns from the data. The validation process is useful for checking the model at the same time as the training process, so model not overfitting. While the testing process is a process to evaluate the performance of the model to be able to make predictions on new data other than training data and testing data. Figure 2 shown the visualization of the flow of the process carried out to be able to perform the classification process of Javanese script letters with CNN transfer learning. For an explanation of the process per stage is given below.

1. First read the image will be used for the training and testing process of the model.
2. After the image data to be used is read, the data division process is carried out so data is split into 70% training data, 10% validation data and 20% testing data. Where each data had been divided will be used for the training process, validation and testing of the model built.
3. Then after the data division process is carried out, call or initialize the pre-trained model used for the transfer learning process. Pre-trained models used are GoogleNet, DenseNet, ResNet, VGG16 and VGG19 models. In this process, the model used does not use the top layer of the previous model, so the top layer in each pre-trained model is set false.
4. After initializing the model to be used, the next layer will be built which will be used as the top layer of the CNN pre-trained model. The top layer of the pre-trained model used can be seen in Figure 3.
5. Then after the model testing process is carried out, the performance calculation process of the model can be carried out using the confusion matrix to get the accuracy, precision, recall and f1-score values from the model test results. It can be seen how well the model performs to be able to carry out the classification process.

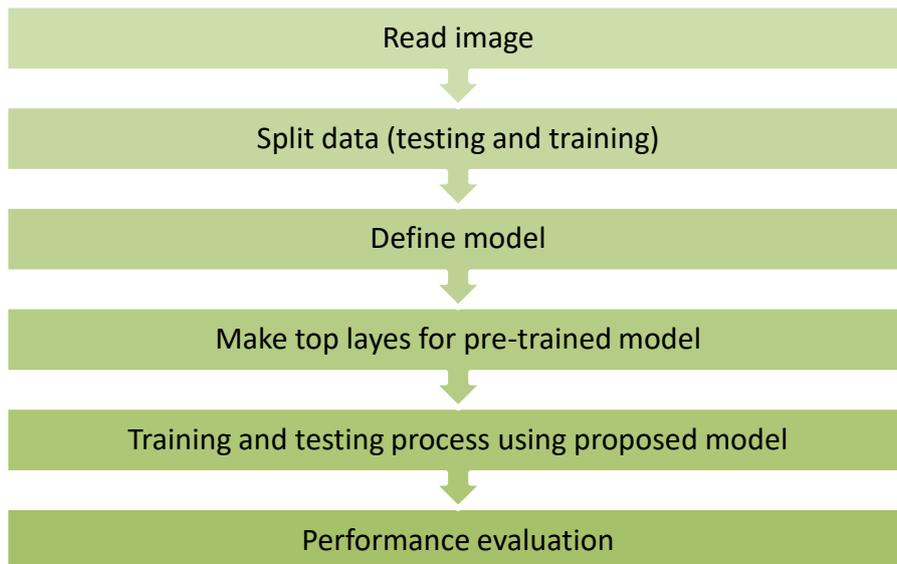


Figure 1. Proposed scheme of classification

Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
inception_v3 (Functional)	(None, 1, 1, 2048)	21802784	densenet121 (Functional)	(None, 3, 3, 1024)	7037504
flatten (Flatten)	(None, 2048)	0	flatten_1 (Flatten)	(None, 9216)	0
dense (Dense)	(None, 1024)	2098176	dense_5 (Dense)	(None, 128)	1179776
dense_1 (Dense)	(None, 128)	131200	dense_6 (Dense)	(None, 256)	33024
dense_2 (Dense)	(None, 256)	33024	dense_7 (Dense)	(None, 512)	131584
dense_3 (Dense)	(None, 512)	131584	dense_8 (Dense)	(None, 1024)	525312
dense_4 (Dense)	(None, 20)	10260	dense_9 (Dense)	(None, 20)	20500

=====
 Total params: 24207028 (92.34 MB)
 Trainable params: 2404244 (9.17 MB)
 Non-trainable params: 21802784 (83.17 MB)

Top Layer for GoogleNet

Top Layer for DenseNet

Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 3, 3, 512)	20024384	vgg16 (Functional)	(None, 3, 3, 512)	14714688
flatten_4 (Flatten)	(None, 4608)	0	flatten_3 (Flatten)	(None, 4608)	0
dense_19 (Dense)	(None, 512)	2359808	dense_15 (Dense)	(None, 512)	2359808
dense_20 (Dense)	(None, 256)	131328	dense_16 (Dense)	(None, 256)	131328
dense_21 (Dense)	(None, 128)	32896	dense_17 (Dense)	(None, 128)	32896
dense_22 (Dense)	(None, 20)	2580	dense_18 (Dense)	(None, 20)	2580

=====
 Total params: 22550996 (86.03 MB)
 Trainable params: 2526612 (9.64 MB)
 Non-trainable params: 20024384 (76.39 MB)

Top Layer for VGG19

Top Layer for VGG16

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 4, 4, 2048)	23587712
flatten_2 (Flatten)	(None, 32768)	0
dense_10 (Dense)	(None, 512)	16777728
dense_11 (Dense)	(None, 1024)	525312
dense_12 (Dense)	(None, 256)	262400
dense_13 (Dense)	(None, 128)	32896
dense_14 (Dense)	(None, 20)	2580
activation_94 (Activation)	(None, 20)	0

=====
 Total params: 41188628 (157.12 MB)
 Trainable params: 17600916 (67.14 MB)
 Non-trainable params: 23587712 (89.98 MB)

TopLayer for ResNet

Figure 2. Top layer for CNN pre-trained model

RESULTS AND DISCUSSIONS

Here, the Javanese script classification process had been used the python programming language to implement the system and jupyter notebook IDE as a tool used to write and run programs. After the program is made to read and share data, programming is done to initialize the pre-trained model will be used for transfer learning and the top layer used in each pre-trained model. Finally, the process for model training can be carried out and the validation process is carried out simultaneously. The results of the training and validation process given in the form of accuracy and loss graphs contained in Table 1. Table 1 shows the results of training and validation on the model in the process of classifying Javanese script letters. These results were obtained using 10 epochs on each model and with layers had been built before.

In Table 1, the best test and validation values are obtained when using the pre-trained CNN VGG16 model. With a test accuracy of 100% and a validation accuracy of 99.37%. This shows VGG16 can properly perform the classification process on training data. Meanwhile, the best loss value is also obtained when validating with the VGG16 model, it had been seen when the validation process is carried out with VGG16 it has good performance to be able to accurately classify the validation data. However, it had been seen in the graph of loss results from the training and validation process does not show a significant gap when the training and validation process is carried out. The model built in this study does not experience overfitting and can perform the classification process well. After training and validation of the model, a model training process will be carried out to see how the classification performance of the model has been trained and validated on new data other than train and valid data.

Table 1. Results from training and validation process

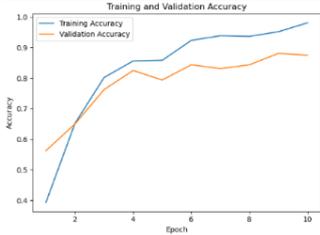
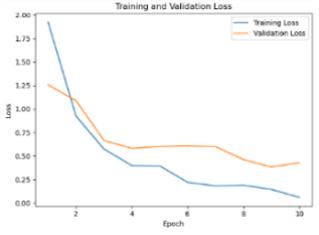
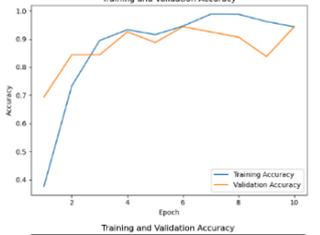
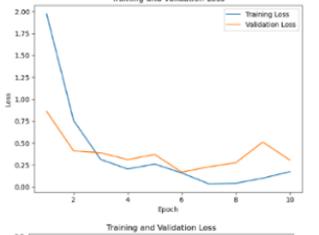
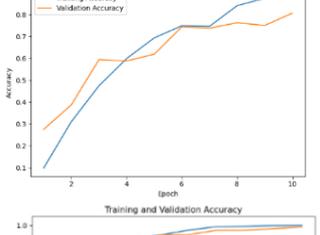
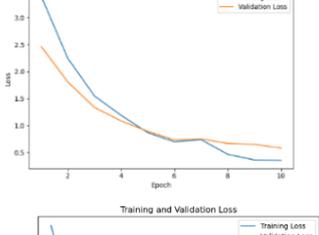
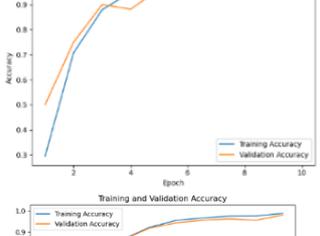
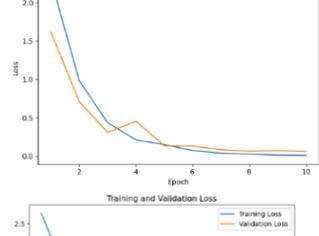
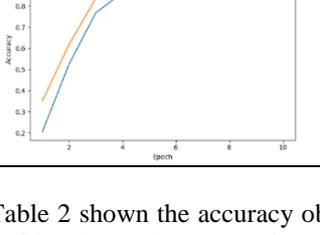
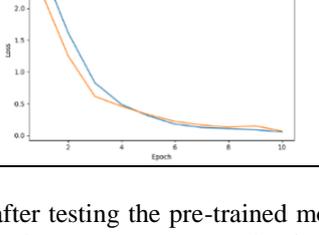
Accuracy Graph	Loss Graph	Model and Result
		Model: GoogleNet Train accuracy: 98.12% Train loss: 6% Validation accuracy: 87.50% Validation loss: 42.77%
		Model: DenseNet Train accuracy: 94.31% Train loss: 17.41% Validation accuracy: 94.38% Validation loss: 30.72%
		Model: ResNet Train accuracy: 87.85% Train loss: 35.26% Validation accuracy: 80.62% Validation Loss: 57.93%
		Model: VGG16 Train accuracy: 100% Train loss: 1.01% Validation accuracy: 99.37% Validation loss: 6.11%
		Model: VGG19 Train accuracy: 98.89% Train loss: 5.66% Validation accuracy: 98.12% Validation loss: 6.77%

Table 2 shown the accuracy obtained after testing the pre-trained model has previously been trained and validated. Can be seen in table 2, the best test accuracy value is obtained when using the pre-trained model VGG16 which is 99.50%. Because the testing process uses new data has not previously been given to the model either in the training or validation process, it shows using the VGG16 pre-trained model with a previously built top layer can be very good and accurate in performing the Javanese script letter classification process. After testing the model, then the performance calculation process can be carried out

using the confusion matrix. The values used in calculating the performance of model testing are the precision value, recall value, and the f1-score value previously described. For the results of these value after the testing process, given in Table 3. The best average precision value obtained after classifying using VGG19 model which is 100%. This shows the VGG19 model can accurately predict all existing classes. Meanwhile, the best average value of recall and f1-score obtained when using VGG16 and VGG19 models which is 99%. This shows the VGG16 and VGG19 models built had a good performance in predicting all classes and had a good balance value between precision, and recall. Based on the research results obtained by [44] regarding the Javanese script classification process using the model results of the combination of CNN and SVM, the test accuracy result is 98.35%, this proves the model built in this study can perform the classification process very well, especially with models use transfer learning VGG16 and VGG19. As for the support value in all models is 400, because the support value is the total amount of data used to test the model. Javanese script letter classification is interesting research to discuss. Therefore, there are several studies have been conducted to classify Javanese characters. Based on Table 4, the accuracy obtained in this study can provide an increase, especially the accuracy better than the previous research. It was concluded the pre-trained model built and implemented in this study can be more effective and accurate in carrying out the classification process of Javanese script letters.

Table 2. A Comparison result based on our proposed method

Model	Testing Accuracy	Avg Precision	Avg Recall	Avg F1-Score	Support
GoogleNet	88.75%	89%	89%	89%	400
DenseNet	92.00%	95%	92%	92%	400
ResNet	82.74%	87%	82%	82%	400
VGG16	99.25%	99%	99%	99%	400
VGG19	99.50%	100%	99%	99%	400

Table 4. A Comparison result based on our previous research

Research by	Method	Accuracy Obtained
Muhdalifah [45]	CNN with Pooling Layer Costumization	CNN with average pooling layer 5x5 is 93%, then 3x3 is 91%. with max pooling layer 5x5 is 92%, then 3x3 is 90%
Diqi [46]	CNN	Accuracy is 99.69%
Our Proposed	Pre-Trained Model CNN	GoogleNet is 88.75%, DenseNet is 92%, ResNet is 82.7499%, VGG16 is 99.25%, VGG19 is 99.50%

CONCLUSION

After conducting research and testing to classify Javanese script letters using the CNN pre-trained model, getting the best test accuracy is when classifying using the VGG19 pre-trained model which is 99.50%. The average value of precision, recall, and f1 score also obtained by the VGG19 pre-trained model, namely 100%, 99% and 99%. It is concluded using the VGG19 pre-trained model can be good and accurate in the process of classifying Javanese script letters. We hope in the next research, it is expected to be able to carry out the image enhancement process to further prevent overfitting and also further expected to be able to use enhancement on CNN either by using other pre-trained models such as SqueezeNet, MobileNet, EfficientNet, and others or increasing parameters on the CNN model. Future research is also expected to improve the accuracy of GoogleNet, DenseNet, and ResNet transfer learning models by adding data augmentation or using parameter optimization using Particle Swarm Optimization or Genetic Algorithm or other methods.

REFERENCES

- [1] M. A. Rasyidi, T. Bariyah, Y. I. Riskajaya, dan A. D. Septyani, "Classification of handwritten Javanese script using random forest algorithm," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, hal. 1308–1315, 2021, doi: 10.11591/eei.v10i3.3036.
- [2] A. R. G *et al.*, "Implementation of of Optical Optical Character Character Recognition Recognition using using Tesseract Tesseract with the Javanese Script Target in Android Application with the Javanese Script Target in Android Application," *Procedia Comput. Sci.*, vol. 157, hal. 499–505, 2019, doi: 10.1016/j.procs.2019.09.006.
- [3] I. Prihandi, I. Ranggadara, S. Dwiasnati, Y. S. Sari, dan Suhendra, "Implementation of Backpropagation Method for Identified Javanese Scripts," *J. Phys. Conf. Ser.*, 2020, doi:

- 10.1088/1742-6596/1477/3/032020.
- [4] U. D. Choiro, K. Qismiyah, dan D. P. Yuniar, "Development of Learning Media with Javanese," *Proc. Int. Conf. Educ. Manag. Technol. (ICEMT 2022)*, vol. 1, hal. 172–179, 2023, doi: 10.2991/978-2-494069-95-4.
 - [5] D. Iskandar, S. Hidayat, U. Jamaludin, dan S. M. Leksono, "Javanese script digitalization and its utilization as learning media: an etnopedagogical approach," *Int. J. Math. Sci. Educ.*, vol. 1, hal. 21–30, 2023, doi: 10.59965/ijmsed.v1i1.24.
 - [6] A. A. Nggofur dan S. Dwijonagoro, "Improving Javanese Letter Reading Skill through the Iqro Script Method," *Int. J. Corner Educ. Res.*, vol. 1, no. 2, hal. 69–77, 2022, doi: 10.54012/ijcer.v1i2.95.
 - [7] A. Budiman, A. Fadlil, dan R. Umar, "Identification of Learning Javanese Script Handwriting Using Histogram Chain Code," *Edumaspul J. Pendidik.*, vol. 7, no. 1, hal. 147–153, 2023, doi: <https://ummaspul.e-journal.id/maspuljr/article/view/5413>.
 - [8] B. Prayoga, H. Mita, A. Sari, L. Sri, dan J. Asharani, "Development of javanese script learning media based on uno stacko game and penta helix approach at sdn 2 sidowarno," *Int. J. Javanese Stud.*, vol. 6, no. 2, hal. 1286–1297, 2023, doi: <https://dx.doi.org/10.20961/javanologi.v6i2.75696> DEVELOPMENT.
 - [9] K. L. Kohsasih dan B. H. Hayadi, "Classification SARS-CoV-2 Disease based on CT-Scan Image Using Convolutional Neural Network," *Sci. J. Informatics*, vol. 9, no. 2, hal. 197–204, 2022, doi: 10.15294/sji.v9i2.36583.
 - [10] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 6, hal. 1–20, 2021, doi: 10.1007/s42979-021-00815-1.
 - [11] H. Liu dan B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," *Appl. Sci.*, vol. 9, no. 20, 2019, doi: 10.3390/app9204396.
 - [12] I. Rizwan, I. Haque, dan J. Neubert, "Deep learning approaches to biomedical image segmentation," *Informatics Med. Unlocked*, vol. 18, hal. 100297, 2020, doi: 10.1016/j.imu.2020.100297.
 - [13] C. Janiesch dan K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, hal. 685–695, 2021.
 - [14] N. Kasabov, "Springer Series on Bio- and Neurosystems," in *Springer Series on Bio-and Neurosystems 7 Time-Space*, 2019.
 - [15] P. Wang, E. Fan, dan P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognit. Lett.*, vol. 141, hal. 197–204, 2021, doi: <https://doi.org/10.1016/j.patrec.2020.07.042>.
 - [16] D. A. Bashar, "Survey on evolving deep learning neural network architectures," *J. Artif. Intell. Capsul. Networks*, vol. 1, hal. 73–82, 2019, doi: 10.36548/jaicn.2019.2.003.
 - [17] X. Liu, L. Song, S. Liu, dan Y. Zhang, "A Review of Deep-Learning-Based Medical Image Segmentation Methods," *Sustain.*, hal. 1–29, 2021, doi: <https://doi.org/10.3390/su13031224>.
 - [18] N. F. Nissa, A. Janiati, N. Cahya, dan P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method for Women ' s Skin Classification," *Sci. J. Informatics*, vol. 8, no. 1, hal. 144–153, 2021, doi: 10.15294/sji.v8i1.26888.
 - [19] M. R. Ningsih, K. Aalifian, H. Wibowo, dan A. U. Dullah, "Global recession sentiment analysis utilizing VADER and ensemble learning method with word embedding," *J. Soft Comput. Explor.*, vol. 4, no. 3, hal. 142–151, 2023, doi: <https://doi.org/10.52465/jossex.v4i3.193>.
 - [20] S. S. Yadav dan S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *J. Big Data*, 2019, doi: 10.1186/s40537-019-0276-2.
 - [21] N. Zhang, Y.-X. Cai, Y.-Y. Wang, Y.-T. Tian, X.-L. Wang, dan B. Badami, "Skin cancer diagnosis based on optimized convolutional neural network," *Artif. Intell. Med.*, vol. 102, 2020, doi: <https://doi.org/10.1016/j.artmed.2019.101756>.
 - [22] S. A. Alanazi *et al.*, "Boosting Breast Cancer Detection Using Convolutional Neural Network," *J. Healthc. Eng.*, vol. 2021, 2021, doi: <https://doi.org/10.1155/2021/5528622> Research.
 - [23] A. Çinar dan M. Yildirim, "Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture," *Med. Hypotheses*, vol. 139, 2020, doi: <https://doi.org/10.1016/j.mehy.2020.109684>.
 - [24] G. Shrestha, Deepshikha, M. Das, dan N. Dey, "Plant Disease Detection Using CNN," *Proc. 2020 IEEE Appl. Signal Process. Conf.*, 2020, doi: 10.1109/ASPCON49795.2020.9276722.
 - [25] M. Desai dan M. Shah, "Clinical eHealth An anatomization on breast cancer detection and

- diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN),” *Clin. eHealth*, vol. 4, no. 2021, hal. 1–11, 2022, doi: 10.1016/j.ceh.2020.11.002.
- [26] F. Demir, A. M. Ismael, dan A. Sengur, “Classification of Lung Sounds With CNN Model Using Parallel Pooling Structure,” *IEEE Access*, vol. 8, hal. 105376–105383, 2020, doi: 10.1109/ACCESS.2020.3000111.
- [27] A. Rafiei, A. Rezaee, F. Hajati, S. Gheisari, dan M. Golzan, “SSP: Early prediction of sepsis using fully connected LSTM-CNN model,” *Comput. Biol. Med.*, vol. 128, 2021, doi: <https://doi.org/10.1016/j.combiomed.2020.104110>.
- [28] M. Cnn, M. Wang, S. Lu, C. Hsieh, dan C. Hung, “Fault Detection of Wind Turbine Blades Using,” *Sustain.*, vol. 14, 2022, doi: <https://doi.org/10.3390/su14031781>.
- [29] T. Approach dan Z. Leonowicz, “Identification of Plant-Leaf Diseases Using CNN and,” *Electron.*, vol. 10, no. 12, 2021, doi: <https://doi.org/10.3390/electronics10121388>.
- [30] D. I. S. H. T. C. Fathicah, “Classification of Tobacco Leaf Pests Using VGG16 Transfer Learning,” *Dwiretno Istiyadi Swasono; Handayani Tjandrasa; Chastine Fathicah*, hal. 176–181, 2019, doi: 10.1109/ICTS.2019.8850946.
- [31] S. Das, O. F. M. R. R. Aranya, dan N. N. Labiba, “Brain Tumor Classification Using Convolutional Neural Network,” *2019 1st Int. Conf. Adv. Sci. Eng. Robot. Technol.*, 2019, doi: 10.1109/ICASERT.2019.8934603.
- [32] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. N. R. S. N. Prudhvith, dan N. Kumar, “Image Classification using SVM and CNN,” *Int. Conf. Comput. Sci. Eng. Appl.*, 2020, doi: 10.1109/ICCSEA49143.2020.9132851.
- [33] M. A. Jasim dan J. M. AL-Tuwaijari, “Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques,” *Marwan Adnan Jasim; Jamal Mustafa AL-Tuwaijari*, 2020, doi: 10.1109/CSASE48920.2020.9142097.
- [34] M. M. Badža, “applied sciences Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network,” *Appl. Sci.*, 2020, doi: <https://doi.org/10.3390/app10061999>.
- [35] J. Unjung dan M. R. Ningsih, “Optimized Handwriting-based Parkinson ’ s Disease Classification Using Ensemble Modeling and VGG19 Feature Extraction,” *Sci. J. Informatics*, vol. 10, no. 4, hal. 489–498, 2023, doi: 10.15294/sji.v10i4.47108.
- [36] N. Dua, S. Nand, S. Vijay, dan B. Semwal, “Multi - input CNN - GRU based human activity recognition using wearable sensors,” *Computing*, vol. 103, no. 7, hal. 1461–1478, 2021, doi: 10.1007/s00607-021-00928-8.
- [37] A. W. Salehi, S. Khan, G. Gupta, B. I. Alabduallah, dan A. Almjally, “A Study of CNN and Transfer Learning in Medical Imaging : Advantages , Challenges , Future Scope,” *Sustain.*, vol. 15, no. 7, 2023, doi: <https://doi.org/10.3390/su15075930>.
- [38] T. Rahman *et al.*, “Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using,” *Appl. Sci.*, vol. 10, no. 9, 2020, doi: <https://doi.org/10.3390/app10093233>.
- [39] M. Fernando *et al.*, “Melanoma diagnosis using deep learning techniques on dermatoscopic images,” *BMC Med. Imaging*, hal. 1–11, 2021, doi: 10.1186/s12880-020-00534-8.
- [40] M. S. Al-huseiny dan A. S. Sajit, “Transfer learning with GoogLeNet for detection of lung cancer,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 2, hal. 1078–1086, 2021, doi: 10.11591/ijeecs.v22.i2.pp1078-1086.
- [41] M. A. Wakili *et al.*, “Classification of Breast Cancer Histopathological Images Using DenseNet and Transfer Learning,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: <https://doi.org/10.1155/2022/8904768>.
- [42] Y. Zhang, J. Li, S. Wei, F. Zhou, dan D. Li, “Heartbeats Classification Using Hybrid Time-Frequency Analysis and Transfer Learning Based on ResNet,” *IEEE J. Biomed. Heal. Informatics*, vol. 25, no. 11, 2021, doi: 10.1109/JBHI.2021.3085318.
- [43] N. L. Tun, A. Gavrilov, N. M. Tun, D. M. Trieu, dan H. Aung, “Remote Sensing Data Classification Using A Hybrid Pre-Trained VGG16 CNN- SVM Classifier,” *IEEE Conf. Russ. Young Res. Electr. Electron. Eng.*, 2021, doi: 10.1109/EIConRus51938.2021.9396706.
- [44] D. U. K. Putri, D. N. Pratomo, dan Azhari, “Hybrid convolutional neural networks-support vector machine classifier with dropout for Javanese character recognition,” *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 21, no. 2, hal. 346–353, 2023, doi: 10.12928/TELKOMNIKA.v21i2.24266.
- [45] M. Diqi, R. N. Wijaya, dan M. F. M, “Pooling Comparison in CNN Architecture for Javanese Script Classification,” *Int. J. Informatics Comput.*, vol. 3, no. 2, 2021, doi: 10.35842/ijicom.

- [46] M. Diqi dan M. F. Muhdalifah, "Design and Building Javanese Script Classification in The State Museum of Sonobudoyo Yogyakarta," *Int. J. Informatics Comput.*, vol. 1, no. 2, hal. 34–35, 2019, doi: <https://doi.org/10.35842/ijicom.v1i2.18>.