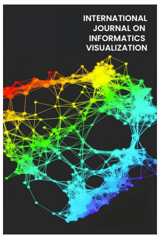
INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



Handwritten Character Recognition Using Deep Learning Algorithm with Machine Learning Classifier

Muhamad Arief Liman [a], Antonio Josef [a], Gede Putra Kusuma [a,*]

[a] *Computer Science Department, Bina Nusantara University, Palmerah, Jakarta, 11480, Indonesia,*
Corresponding author: [*]inegara@binus.edu

Abstract— **Handwritten character recognition is a problem that has been worked on for many mainstream languages. Handwritten letter recognition has been proven to achieve promising results. Several studies using deep learning models have been conducted to achieve better accuracies. In this paper, the authors conducted two experiments on the EMNIST Letters dataset: Wavemix-Lite and CoAtNet. The Wavemix-Lite model uses Two-Dimensional Discrete Wavelet Transform Level 1 to reduce the parameters and speed up the runtime. The CoAtNet is a combined model of CNN and Visual Transformer where the image is broken down into fixed-size patches. The feature extraction part of the model is used to embed the input image into a feature vector. From those two models, the authors hooked the value of the features of the Global Average Pool layer using EMNIST Letters data. The features hooked from the training results of the two models, such as SVM, Random Forest, and XGBoost models, were used to train the machine learning classifier. The experiments conducted by the authors show that the best machine-learning model is the Random Forest, with 96.03% accuracy using the Wavemix-Lite model and 97.90% accuracy using the CoAtNet model. These results showcased the benefit of using a machine learning model for classifying image features that are extracted using a deep learning model.**

Keywords— **Deep learning; ensemble model; handwritten character recognition; image embedding; machine learning models.**

Manuscript received 13 Mar. 2023; revised 9 Jul. 2023; accepted 8 Sep. 2023. Date of publication 31 Mar. 2024.

I. Introduction

Writing is one of the activities used by humans to communicate, convey, and record information. Everyone has a different way of writing; for example, when writing the letter "a," everyone is different, or the letter "i" or "l" is sometimes the same. The difference in a person's handwriting is one of the difficulties in research on handwriting recognition. Handwritten character recognition is a process by which handwriting can be recognized in any language. Handwritten character recognition is a nearly solved problem for many mainstream languages [1]. According to Khandokar, "handwritten character recognition" is a mechanism to process handwriting into data that can be analyzed, edited, and searched. The primary purpose of using handwritten character recognition is so that a machine can imitate the human ability to read, change, and communicate in a short time [2]. Handwritten character recognition is divided into two types: online and offline handwritten character recognition, where offline handwritten character recognition is recognition through an image, and online handwritten character recognition is direct recognition using electronic tools [3].

Fig. 1 shows the processes in a handwritten character recognition model consisting of preprocessing, feature extraction, and classification. Differences in a person's handwriting are one of the most severe difficulties in handwritten character recognition; besides differences in handwriting, the other most considerable difficulty is the small, labeled dataset [4]. The presence of numerous similar characters and a wide range of character categories is discussed [2]. Deep learning is one method that can be used to model handwritten character recognition. Deep learning is a development of machine learning where the machine automatically performs feature extraction [5]. CNN is one of the algorithms that is often used in deep learning. Moreover, CNN is a state-of-the-art neural network with significant applications in computer vision [6].

Research on handwriting character recognition is essential, considering that we have entered the digitalization era, which requires technology to convert handwritten documents into digital format. With the existence of research in this field, it can be constructive to create such technology that is more efficient in saving time and resources. The research that Jeevan [7] has conducted produces the most excellent accuracy in making the handwritten character recognition model using the MNIST letter dataset, which is 95.96% accurate. Furthermore, Jeevan proposed a model called WaveMix-Lite combined with a 2-dimensional Discrete Wavelet Transform (2D-DWT). In this model, Jeevan uses Level 1 2-Dimensional Discrete Wavelet Transform, which makes computations when training this model faster and reduces the parameters compared to using Level 4 2-Dimensional Discrete Wavelet Transform [7]. WaveMix-Lite is a reasonably good model in terms of computation and the resulting level of accuracy. However, the drawback of this model is the classifier, which only consists of one output layer. Changing the classifier and using a machine learning classifier like SVM, Random Forest, or XGBoost can improve Jeevan's proposed model. Apart from conducting experiments on the Wavemix-Lite model, improvements were made to one of the Vision Transformer models, CoAtNet, where CoAtNet combines the CNN model and the Vision Transformer itself, proposed by Google Research [8].

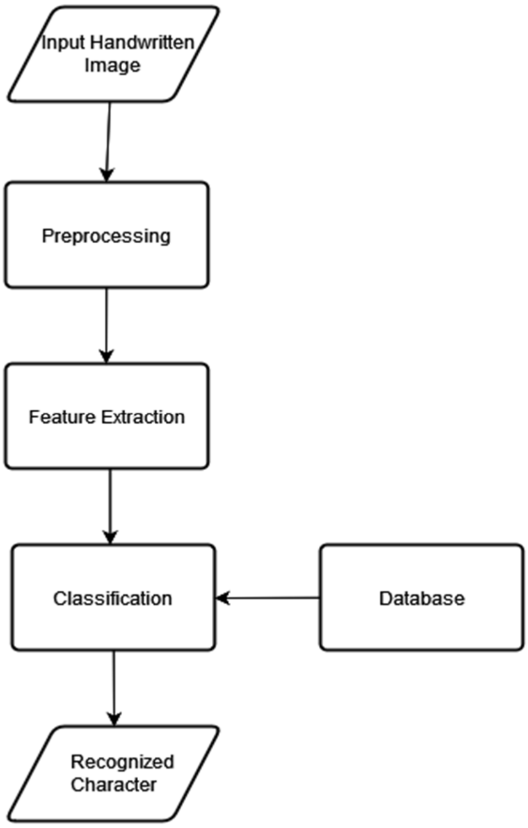


Fig. 1 Handwritten character recognition process

II. Materials and Method

A. Related Works

EMNIST Letters is a handwritten dataset of alphabetic letters (uppercase and lowercase) obtained from the NIST particular database 19 and converted to a 28x28 pixel image format. This dataset is used as one of the benchmarks in the work on handwriting recognition research. Furthermore, several studies using deep learning methods have been conducted to achieve the best accuracy using the proposed models. In 2018, research by [9] tried applying a deep neural network (DNN) for letter classification in the EMNIST Letters dataset. The stage begins with preprocessing the input image, which includes image thresholding, character thinning using morphological operations, slant correction, and image segmentation. After that, the images that have undergone the preprocessing process enter the feature extraction and classifier stages using the DNN model. The DNN model uses a stacked autoencoder to train its many layers. It has three hidden layers: two hidden layers and one SoftMax layer on top of them, each with 300, 50, and 27 neurons. The results of this study achieved an accuracy of 88.8%.

Experiment with an extensive optical convolution that uses logarithmic activation as an image sensor before feeding the image into a perceptron network with 1600 input vectors, three hidden layers with a total of 256 neurons, each with a rectified linear unit (ReLu) activation function, and one fully connected layer with linear activation for output units performed by [10]. From the experiments, the test results on the EMNIST Letters dataset obtained an accuracy of 93.65%.

Researchers from Google [11] conducted experiments by introducing the " μ2net" method, which involved a knowledge transfer system. The proposed method can produce a dynamic multitask ML system that is initialized with one root model, ViT-L/16. From the root model, the evolution process will continue to look for the best model, which is then stored in the active population to be reinitialized with the following model in the next iteration. The active population is iteratively expanded by taking a sample from the parent model and applying a series of sample models to the parent model to generate a child model by going through a training and validation process to score the child model. The score is intended to cut the active population by removing models with worse scores than the primary model, and the better models are appointed as the primary model. From the method offered, testing on the EMNIST Letters dataset obtained an accuracy of 93.68%.

Continuing research [11], [12] carried out the extensions of the μ2net method known as the μ2Net+ method, which produces more parameters shared by the new model with a reduction in computation and size without affecting a decrease in quality. The introduction of mutation actions accomplishes this by lowering the number of transformer layers and increasing the hyperparameter search space, which enables choosing a lower image resolution, identifying the size, and calculating the cost factor. Learning the (μ) function increases the possibility of sampling models with lower costs. Lower, taking into account the chance of different types of mutations occurring. Of the methods offered, experiments on the EMNIST Letters dataset obtained an accuracy of 95.03%.

In 2019, research by [4] used the TextCaps (Capsule Network) model, in which the input image is processed through three convolutional layers: one primary capsule layer and one fully connected capsule, or character capsule layer. Dynamic routing with three routing iterations connects the primary capsule and character capsule. Based on the results of testing this model on the EMNIST Letters dataset, training with the full train dataset gave an accuracy of 95.36%, and training with 200 data samples for each class gave an accuracy of 92.79%.

Research by [13] tested the deep convolutional neural networks (DCNN) model, which has autonomous and continuous learning (ACL) capabilities so that it can automatically generate a DCNN architecture for a specific vision task. [13] tested the deep convolutional neural networks (DCNN) model, which has autonomous and continuous learning (ACL) capabilities to generate a DCNN architecture automatically for a given vision task. The model being tested begins by partitioning the DCNN model into several stacks of meta-convolutional blocks and fully connected blocks. Then, genetic evolutionary operations are used to develop the population of the DCNN architecture, which consists of selection, mutation, and crossover. Based on how the DCNN genetic model performed on the EMNIST Letters dataset, the DCNN architecture comprises three fully connected blocks and five convolutional blocks. The first convolutional block consists of 437 filters with a size of 3x3, average pooling, batch normalization, ReLU activation function, and 15% dropout. The second convolutional block consists of 238 3x3 filters, no pooling layer, batch normalization, LeakyReLU activation function, and 20% dropout. The third convolutional block consists of 133 3x3 filters, no pooling layer, batch normalization, a ReLU activation function, and a 10% dropout. The fourth convolutional block consists of 387 3x3 filters, no pooling layer, batch normalization, TReLU activation function, and 10% dropout. The fifth convolutional block comprises 187 5x5 filters without a pooling layer, batch normalization, ELU activation function, or 50% dropout. The first fully connected block consists of 313 neurons with a ReLU activation function, batch normalization, and 20% dropout. The second fully connected block consists of 252 neurons with the functions of ELU activation, batch normalization, and 20% dropout. The last fully connected block is the output layer, with 26 neurons and the SoftMax activation function. The optimizer used is RMSprop. From the DCNN model formed, an accuracy of 95.58% is obtained.

The research used VGG-5 with a fully connected spinal cord [14]. The size of the fully connected layer proposed to have four hidden layers with several neurons for each layer is 128 neurons. The model used in this study was obtained from training using several existing networks and variations from SpinalNet with random initialization, which were trained ten times to obtain the best model. Researchers use the transfer learning method to get the best model. In the experiment, the best model obtained using the EMNIST Letters dataset obtained an accuracy of 95.88%.

In 2022, research by [7] conducted an experiment using the WaveMix-Lite architecture with one level of 2D-DWT to reduce parameters and computations. The offered architecture has several layers, including the convolutional layer, the WaveMix-Lite block, an MLP head, a global average pooling layer, and a SoftMax layer to generate possible classes. In the WaveMix-Lite block, there are several steps. First, the incoming input is processed at the convolutional layer to reduce the embedding dimension. Next, the input is processed at the 2D-DWT layer, which creates four output images that are half the size of the input. These output images are combined and sent to the MLP layer, which comprises two 1x1 convolutional layers separated by GELU non-linearity. The MLP flows information unidirectionally from the input to the output layer through the hidden layers in the multilayer feedforward neural network [15]. After that, the image size is reconciled to the input size using the transposed convolutional layer. Then, it proceeds to the batch normalization layer, and the output results are continued to the image classification process layer. In the experiment using the method offered on the EMNIST Letters dataset, an accuracy of 95.96% was obtained.

From the results of previous research, the use of deep learning on the EMNIST Letters dataset for handwritten recognition has been proven to achieve high accuracy. According to the author's literature review, the WaveMix-Lite architecture has the highest accuracy. As a result, the authors will conduct experiments using the WaveMix-Lite architecture to obtain features from the EMNIST Letters dataset and use the features obtained to train on various machine learning models. The authors also saw potential in using vision transformers, which can achieve high accuracy, as [11] and [12] have done. So, the authors will use the same method, namely extracting features from the CoatNet vision transformer model [8]. Our research mainly aims to use Wavemix-Lite and CoatNet as feature extractors and a machine-learning model based on classification as a classification layer.

B. Proposed Method

1) Preprocessing: Preprocessing is carried out to maximize the data used. In this process, the normalization of data is carried out using the torch vision function, which uses the mean and standard deviation of the dataset used. The mean and standard deviation values are obtained using the formula in equation (1), where E[X2] is the mean of quadratic data and (E[X]) 2 is the square of the mean data. Furthermore, get an average value of 0.1722 and a standard deviation value of 0.3309. In addition to normalization, resizing the dataset used in the model will be tried, where the initial size of the 28x28 image is changed to 32x32. In this process, the resize function is used to change the image size in the dataset that will be used.

$$\sigma = \sqrt{E[X^2] - (E[X])^2} \qquad (1)$$

σ : Standard deviation

$E[X^2]$: Mean of Quadratic Data

$(E[X])^2$: Square of The Mean Data

2) Model Development: Two models—the first, WavemixLite, named after [7], and the second, CoAtNet, named after [8] —were used to conduct this study. Both of these models are deep learning models that are used and trained using datasets in the form of images.

The model proposed by Jeevan [7] is a model that uses a 2-dimensional Discrete Wavelet Transform in its architecture. This model uses a Level 1 2-dimensional Discrete Wavelet Transform to reduce parameters and speed up run time compared to a Level 4 2-dimensional Discrete Wavelet Transform, which is the highest level. According to Huang [16], going through a 2D DWT process, an image can be divided into four parts: 3 images with high-frequency sub-bands and one with low-frequency sub-bands.

Figure 2 shows the Wavemix-Lite architecture, which is made up of a Convolutional Layer, Wavemix-Lite Blocks (Level 1 2D-DWT), MLP Head, Global Average Pooling, and SoftMax (Output Layers). Convolutional neural networks

(CNNs) frequently employ the Global Average Pooling (GAP) technique for spatial dimension reduction in the last layers before the fully connected layers, which are typically utilized for classification or regression problems [17]. Most of the time, stochastic gradient descent (SGD) or other gradient-based optimization techniques are used to train SoftMax, updating the model's parameters based on the gradients of the loss function [18].

Wavemix-Lite Block, as shown in Fig. 4, produces four outputs for each input; the four outputs have the same value. According to the channel value on an input, this wavemix-lite block also reduces half the input's resolution; if the input is 28x28, then the output will be 14x14.

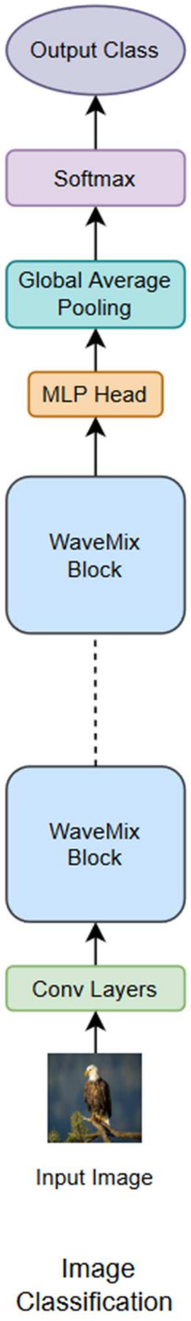


Fig. 2 Wave Mix-Lite Image Classification Architecture

CoAtNet is a deep learning model proposed by Dai [8] from Google Research, where the model is a combination of the convolutional neural network and the vision transformer. A convolutional neural network is one of the algorithms in deep learning that is often used to handle image analysis processes. A CNN has three main types of neural layers: convolutional layers, pooling layers, and fully connected layers [19]. A vision transformer is an image classification process in which the image is broken up into fixed-size patches, each embedded linearly, and a classification token is added to the sequence used [20]. Self-attention is one of the mechanisms of ViTs, which captures an extended range of token dependencies in a global context, the same as traditional recurrent neural networks [21]. The CoAtNet architecture is shown in Fig. 4.

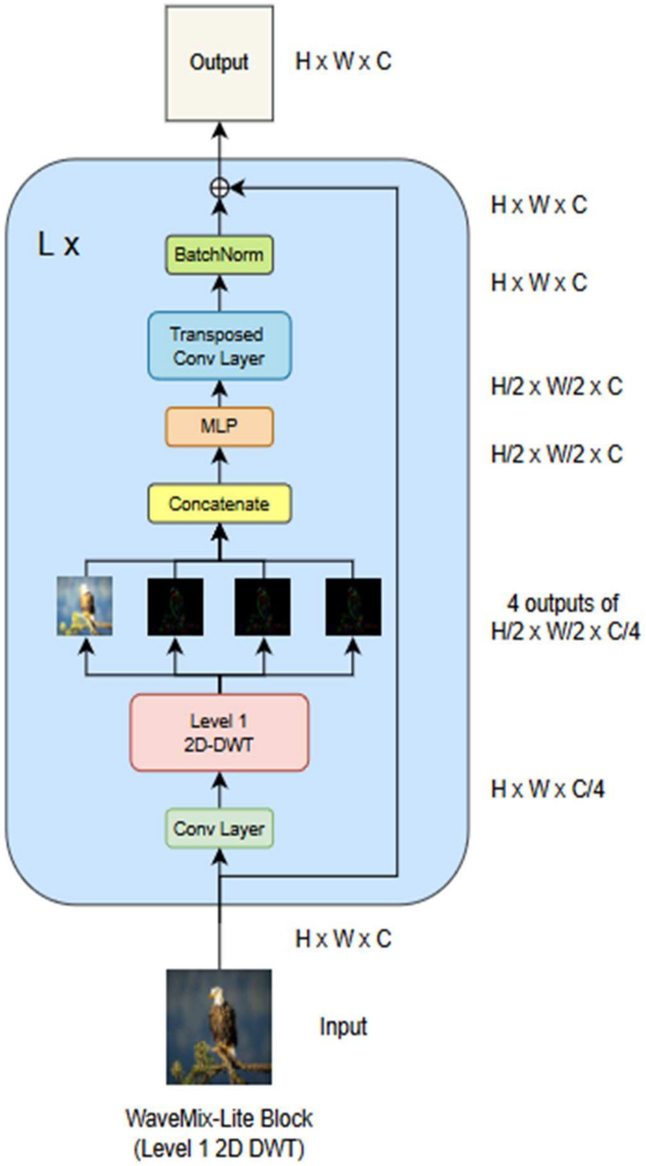


Fig. 3 WaveMix-Lite Image Classification Architecture

The two models proposed in previous research were used; both are deep learning models trained using a dataset containing images. Both models have reasonably good accuracy, but one of the drawbacks is that there is only one output layer. Another classifier model may take the place of the output layer.

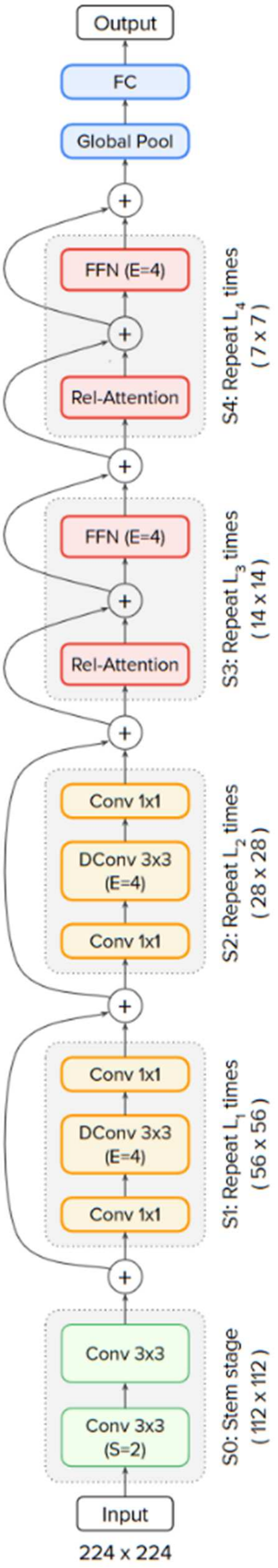


Fig. 4 CoatNet Architecture

3) Classifier Model: As previously mentioned, the two models used for this research have only one output layer. Wavemix-Lite uses a SoftMax activation output layer, while CoAtNet uses linear activation. SoftMax, also known as Multinomial Logistic Regression, is mostly utilized in mathematics, particularly probability theory and related subjects [22]. The SoftMax classifier is based on the Logistic Regression classification in statistics. The primary principle of logical regression is to apply the classification approach to judge the input data and then output a single discrete result [23].

In this research, the output layer in each model will be replaced by a machine learning classifier model such as SVM, Random Forest, or XGBoost. In the proposed method shown in Fig. 5, the first thing to do is do a feature extraction hook [24] on the global average pool, which is the feature value in the model that has been created, so that later, the feature value will be used to train the machine learning classifier model that will be combined. Global Average Layer transforms a (M x M x N) feature map into a (1 x N) feature map, where (M x M) is the size of the image and N is the number of filters [25]. In doing the hooking, it uses the function from PyTorch to fetch values from a particular layer in both models. Finally, the feature value is resized to train the machine learning classifier model.

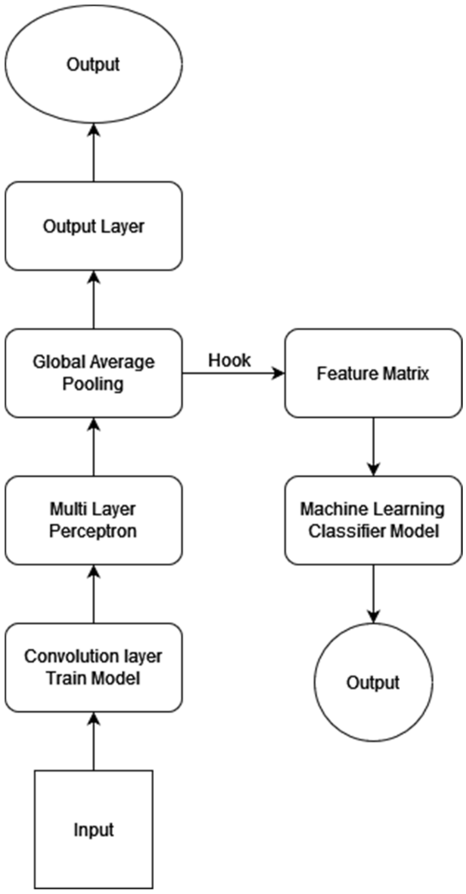


Fig. 5 Proposed method

This experiment used three classifier models: SVM (Support Vector Machine). This machine learning algorithm performs classification based on optimal margins developed for nonlinear data using kernel and multi-class data [26]. Furthermore, a random forest is a combination of tree predictors that depends on the value of a random vector sample [27]. Finally, XGBoost is an algorithm that can scale learning systems for tree boosting [28].

SVM is a powerful supervised learning algorithm widely used in engineering. Solve nonlinear and high-dimensional classification problems using VC dimensionality and structural risk minimization theories. The SVM algorithm aims to find the optimal hyperplane to separate the feature space[29]. SVM uses data items represented as points in space of size N (where N = the number of features), with characteristic values serving as coordinates. Identify the hyperplane separating the two groups. In scikit-learn, SVM provides three categories (SVC, NuSVC, and LinearSVC) to distinguish multiple classes. LinearSVC is suitable for classifying the MNIST dataset because of its flexible choice of penalties and loss functions and its suitability for large sample sizes [30].

Random forest is a supervised machine-learning technique that uses decision trees for tasks such as classification and regression. It works as an ensemble method to create multiple decision tree classifiers with different subsets of input features. Combining predictions from these trees based on separate sets of random vectors yields robust and accurate results [31]. Each tree contributes to a class prediction, and the final model's prediction is based on the majority vote (for classification) or mean prediction (for regression) of each tree. It is commonly used for classification and regression tasks and has shown effective results in predicting stroke and other life-threatening risks [32]. Combine multiple classifier systems (MCS) to improve reliability compared to individual classifiers. The four approaches proposed to build MCPs are design level, classifier level, feature level, and data level. The last two approaches, incorporating bagging techniques, boosting techniques, and random subspace principles, have been used and have proven very successful. This algorithm works with two parameters, L and K. L is the number of trees in the forest, and features K are preselected for the splitting process [33].

Xgboost is one implementation of Gradient Boosting Machines (GBM), considered one of the most powerful supervised learning algorithms. It can be used for both regression and classification problems. Xgboost is preferred by data scientists due to its high execution speed outside of core computing [34]. Most existing GBM models consistently outperform other machine learning algorithms. It shows excellent performance on various machine learning reference datasets. XGBoost is an ensemble method that builds new models to correct the residuals and errors of previous models and combines them to make final predictions. The effectiveness of the XGBoost algorithm has been widely recognized in many machine learning and data mining challenges, making it a more widely used and popular tool in the data scientist industry [35].

III. Result and Discussion

A. Dataset

In this research, the EMNIST Dataset was used where EMNIST is a dataset consisting of a collection of Handwritten Character letters originating from the NIST Special Database 19, which have been converted to a 28x28 image format and done in grayscale and have a structure that matches the MNIST dataset [36]. An example of an image in the dataset is shown in Fig. 6. EMNIST consists of 6 datasets: ByClass, ByMerge, Balanced, Digits, Letters, and MNIST. In this research, EMNIST Letters is the primary dataset used. The EMNIST Letters consist of 26 Balanced Classes, with A-Z for each Class.

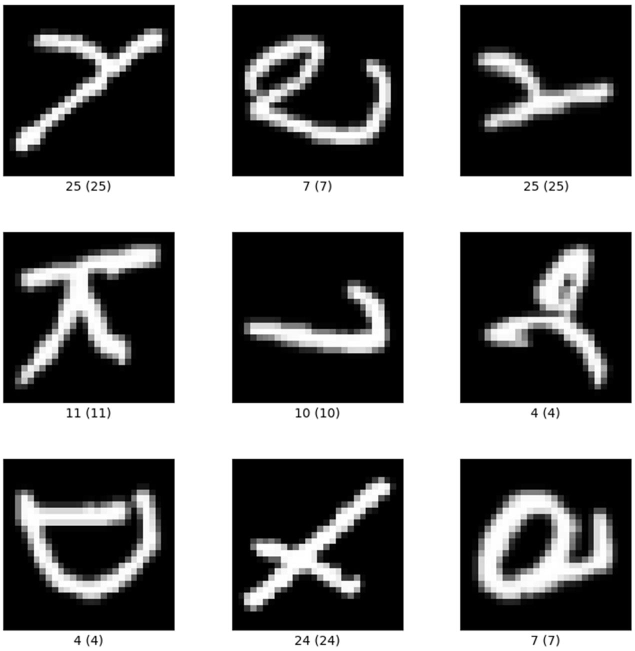


Fig. 6 Example of EMNIST Letters dataset

B. Experimental Design

PyTorch provided the dataset used in this research by normalizing it according to previously mentioned values. The EMNIST Letters dataset consists of 145600 image data consisting of 124800 training data, and 20800 test data, which are constantly shuffled. Each data has been done in grayscale and has a 28x28 image format. There are two optimizers used, namely Adam and SGD, where the optimizer is a process of reducing the value of the cost function by changing the parameter values step by step. SGD or Stochastic Gradient Descent is an optimization technique to update the parameter θ at each time step t [37]. SGD is called "stochastic" because it uses a random mini batch of training examples for computing the gradients and updating the parameters. At the same time, Adam is an adaptive learning rate method, where Adam calculates individual learning rates for different parameters [38] based on the average of the past squared gradients (similar to RMSprop) and the average of the past gradients (similar to momentum) [39]. This experiment used a service from Google Colab Pro with GPU Hardware Accelerator, Premium GPU Class, and High-RAM Runtime mode.

1) Experiment – I: In the first experiment, the Wavemix-lite model was adjusted using 112 dimensions and 16 channels according to what was proposed earlier [7]. In this experiment, model training was carried out using the Adam optimizer and SGD optimizer in the last 15 out of 50 epochs with a momentum of 0.9. The time used to train the Wavemix-Lite model is around 4 hours. A straightforward method that, when a triggering condition is met, switches from Adam to SGD. The projection of Adam's steps on the gradient subspace relates to our suggested condition [40].

2) Experiment – II: In the second experiment, the CoAtNet model was used by adjusting the model using coatnet 4, which has five dimensions consisting of [2, 2, 12, 28, 2] and five channels consisting of [192, 192, 384, 768, 1536] which is by the research proposed previously [8]. In this research, model training was carried out using the Adam optimizer with a total of 30 epochs and using a resized dataset with an image format of 32x32. The time used to train the CoAtNet model is around 3 hours. After the two models have been successfully trained, the hooks feature process is carried out on the Global Average Pooling. These hooks are carried out using the functions in PyTorch, which were explained earlier. The resulting features are then reshaped and used for implementation into machine learning classifier models, namely SVM, Random Forest, and XGBoost, with a total time of about 1 hour for the three classifier models.

C. Experimental Result

From the experiments on the two models, Wavemix-Lite and CoAtNet obtained improved accuracy results from the base model that each author had proposed. Due to the dissimilarity of the existing datasets, the base model's results are slightly different. However, they are still around the results of the accuracy of the models that were previously proposed. The results are shown in the following Table I.

TABLE I
BASE WAVEMIX-LITE AND COATNET MODE

	Wavemix-Lite	CoAtNet
Proposed Model	Wavemix-Lite 112/16	Coatnet-4
Dimension	112	5 | [2, 2, 12, 28, 2]
Channels	16	5 | [192, 192, 384, 768, 1536]
Epochs	50	30
Optimizer	Adam & SGD (Last 15 epoch)	Adam

Table II below shows the results of the WaveMix-Lite model with the Model classifier, with the results for SVM getting an accuracy of 95.82% with Precision 0.96, Recall 0.96, and F1-Score 0.96. Furthermore, the results for Random Forest get an accuracy of 96.03% with a Precision of 0.96, Recall 0.96, and F1-Score 0.96. Again, XGBoost brings an accuracy of 95.42% with a precision of 0.95, a recall of 0.95, and an F1-Score of 0.95.

TABLE II
RESULT OF WAVEMIX-LITE WITH CLASSIFIER MODEL

Classifier	Accuracy	Precision	Recall	F1-Score
Wavemix-Lite 112/16	95.35	0.95	0.95	0.95
SVM	95.82	0.96	0.96	0.96
Random Forest	96.03	0.96	0.96	0.96
XGBoost	95.42	0.95	0.95	0.95

Table III below shows the results of the CoAtNet model with the Model classifier, with the results for SVM getting an accuracy of 96.11% with Precision 0.96, Recall 0.96, and F1-Score 0.96. Furthermore, the results for Random Forest get an accuracy of 97.90% with Precision 0.98, Recall 0.98, and F1-Score 0.98. Again, XGBoost brings an accuracy of 96.73% with a Precision of 0.97, Recall of 0.97, and F1-Score 0.97.

TABLE III
RESULT OF COATNET WITH CLASSIFIER MODEL

Classifier	Accuracy	Precision	Recall	F1-Score
CoAtNet 4	91.06	0.95	0.95	0.95
SVM	96.11	0.96	0.96	0.96
Random Forest	97.90	0.98	0.98	0.98
XGBoost	96.73	0.97	0.97	0.97

IV. CONCLUSION

The authors did two experiments with the EMNIST Letters dataset in this paper. They used the Wavemix-Lite and CoAtNet models to get features from the training results using EMNIST Letters data. The features successfully formed from the training results of the two models were used to train the machine learning classifier SVM, Random Forest, and XGBoost models. From this experiment, the authors compared the accuracy obtained by the model without using the machine learning classifier model and by using the machine learning classifier model, where using the machine learning classifier model can increase the accuracy of up to 0.68% of the features obtained from the training results using the WaveMix-Lite and 6.84% uses the CoAtNet model. In addition, the experiments conducted by the authors show that the best machine-learning model for increasing accuracy is the Random Forest model.

The proposed method is proven to increase accuracy in handwritten letter recognition, especially on the emnist letters dataset. The author anticipates that this research can expand once more by using additional machine learning or deep learning models to extract features and applying the model to another dataset of handwritten letters. In addition, this research can also be applied to other research fields.

ACKNOWLEDGMENT

The authors thank Cohen *et al.* from Western Sydney University for the EMNIST Dataset.

REFERENCES

[1] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep, Big, Simple Neural Nets for Handwritten Digit Recognition,” Neural Computation, vol. 22, no. 12, pp. 3207–3220, Dec. 2010, doi:10.1162/neco_a_00052.

[2] I. Khandokar, M. Hasan, F. Ernawan, S. Islam, and M. N. Kabir, “Handwritten character recognition using convolutional neural network,” Journal of Physics: Conference Series, vol. 1918, no. 4, p. 042152, Jun. 2021, doi: 10.1088/1742-6596/1918/4/042152.

[3] A. Priya, S. Mishra, S. Raj, S. Mandal, and S. Datta, “Online and offline character recognition: A survey,” 2016 International Conference on Communication and Signal Processing (ICCSP), Apr. 2016, doi: 10.1109/iccsp.2016.7754291.

[4] V. Jayasundara, S. Jayasekara, H. Jayasekara, J. Rajasegaran, S. Seneviratne, and R. Rodrigo, “TextCaps: Handwritten Character Recognition With Very Small Datasets,” 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Jan. 2019, doi: 10.1109/wacv.2019.00033.

[5] H. Ansaf, H. Najm, J. M. Atiyah, and O. A. Hassen, “Improved Approach for Identification of Real and Fake Smile using Chaos Theory and Principal Component Analysis,” Journal of Southwest Jiaotong University, vol. 54, no. 5, 2019, doi: 10.35741/issn.0258-2724.54.5.20.

[6] R. Vaidya, D. Trivedi, S. Satra, and Prof. M. Pimpale, “Handwritten Character Recognition Using Deep-Learning,” 2018 Second International Conference on Inventive Communication and

Computational Technologies (ICICCT), Apr. 2018, doi:10.1109/icicct.2018.8473291.

[7] P. Jeevan, K. Viswanathan, and A. Sethi, "WaveMix-Lite: A Resource-efficient Neural Network for Image Analysis," May 2022.

[8] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoAtNet: Marrying Convolution and Attention for All Data Sizes," Jun. 2021.

[9] T. S. Gunawan, A. F. R. Mohd Noor, and M. Kartiwi, "Development of English Handwritten Recognition Using Deep Neural Network," Indonesian Journal of Electrical Engineering and Computer Science, vol. 10, no. 2, p. 562, May 2018, doi: 10.11591/ijeecs.v10.i2.pp562-568.

[10] P. Pad, S. Narduzzi, C. Kundig, E. Turetken, S. A. Bigdeli, and L. A. Dunbar, "Efficient Neural Vision Systems Based on Convolutional Image Acquisition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[11] A. Gesmundo and J. Dean, "An Evolutionary Approach to Dynamic Introduction of Tasks in Large-scale Multitask Learning Systems," May 2022.

[12] A. Gesmundo, "A Continual Development Methodology for Large-scale Multitask Dynamic ML Systems," Sep. 2022.

[13] B. Ma, X. Li, Y. Xia, and Y. Zhang, "Autonomous deep learning: A genetic DCNN designer for image classification," Neurocomputing, vol. 379, pp. 152–161, Feb. 2020, doi: 10.1016/j.neucom.2019.10.007.

[14] H. M. D. Kabir *et al.*, "SpinalNet: Deep Neural Network with Gradual Input," Jul. 2020.

[15] H. Taud and J. F. Mas, "Multilayer Perceptron (MLP)," Lecture Notes in Geoinformation and Cartography, pp. 451–455, Oct. 2017, doi:10.1007/978-3-319-60801-3_27.

[16] Z.-H. Huang, W.-J. Li, J. Shang, J. Wang, and T. Zhang, "Non-uniform patch based face recognition via 2D-DWT," Image and Vision Computing, vol. 37, pp. 12–19, May 2015, doi:10.1016/j.imavis.2014.12.005.

[17] A. Ghosh, B. Bhattacharya, and S. Basu Roy Chowdhury, "AdGAP: Advanced Global Average Pooling," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, Apr. 2018, doi:10.1609/aaai.v32i1.12154.

[18] J. Ul Rahman, A. Ali, M. Ur Rehman, and R. Kazmi, "A Unit Softmax with Laplacian Smoothing Stochastic Gradient Descent for Deep Convolutional Neural Networks," Intelligent Technologies and Applications, pp. 162–174, 2020, doi: 10.1007/978-981-15-5232-8_14.

[19] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," Computational Intelligence and Neuroscience, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.

[20] K. Han et al., "A Survey on Vision Transformer," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/tpami.2022.3152247.

[21] K. Islam, "Recent Advances in Vision Transformer: A Survey and Outlook of Recent Work," Mar. 2022.

[22] J. Wolfe, X. Jin, T. Bahr, and N. Holzer, "Application of Softmax Regression and Its Validation for Spectral-Based Land Cover Mapping," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-1/W1, pp. 455–459, May 2017, doi: 10.5194/isprs-archives-xlii-1-w1-455-2017.

[23] X. Qi, T. Wang, and J. Liu, "Comparison of Support Vector Machine and Softmax Classifiers in Computer Vision," 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Dec. 2017, doi: 10.1109/icmcce.2017.49.

[24] Y. Cheng, W. Liu, P. Duan, J. Liu, and T. Mei, "PyAnomaly: A Pytorch-based Toolkit for Video Anomaly Detection," Proceedings of the 28th ACM International Conference on Multimedia, Oct. 2020, doi: 10.1145/3394171.3414540.

[25] R. L. Kumar, J. Kakarla, B. V. Isunuri, and M. Singh, "Multi-class brain tumor classification using residual network and global average pooling," Multimedia Tools and Applications, vol. 80, no. 9, pp. 13429–13438, Jan. 2021, doi: 10.1007/s11042-020-10335-4.

[26] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: a review," Artificial Intelligence Review, vol. 52, no. 2, pp. 803–855, Jan. 2018, doi: 10.1007/s10462-018-9614-6.

[27] L. Breiman, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi:10.1023/a:1010933404324.

[28] T. Chen and C. Guestrin, "XGBoost," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016, doi: 10.1145/2939672.2939785.

[29] W. Liu, J. Wei, and Q. Meng, "Comparisions on KNN, SVM, BP and the CNN for Handwritten Digit Recognition," 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA), Aug. 2020, doi:10.1109/aeeca49918.2020.9213482.

[30] P. C. Vashist, A. Pandey, and A. Tripathi, "A Comparative Study of Handwriting Recognition Techniques," 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM), Jan. 2020, doi: 10.1109/iccakm46823.2020.9051464.

[31] B. K. Nyaupane, R. K. Sah, and K. C. Dahal, "SVM, KNN, Random Forest, and Neural Network based Handwritten Nepali Barnamala Recognition," Journal of Innovations in Engineering Education, vol. 4, no. 2, pp. 64–70, Dec. 2021, doi: 10.3126/jiee.v4i2.38254.

[32] B. Vamsi, D. Bhattacharyya, and D. Midhunchakkaravarthy, "Detection of Brain Stroke Based on the Family History Using Machine Learning Techniques," Lecture Notes in Networks and Systems, pp. 17–31, 2021, doi: 10.1007/978-981-16-1773-7_2.

[33] O. M. Khandy and S. Dadvandipour, "Analysis of machine learning algorithms for character recognition: a case study on handwritten digit recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, pp. 574–581, 2021.

[34] A. Ibrahem Ahmed Osman, A. Najah Ahmed, M. F. Chow, Y. Feng Huang, and A. El-Shafie, "Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia," Ain Shams Engineering Journal, vol. 12, no. 2, pp. 1545–1556, Jun. 2021, doi:10.1016/j.asej.2020.11.011.

[35] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," Interactive Learning Environments, vol. 31, no. 6, pp. 3360–3379, May 2021, doi: 10.1080/10494820.2021.1928235.

[36] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," 2017 International Joint Conference on Neural Networks (IJCNN), May 2017, doi:10.1109/ijcnn.2017.7966217.

[37] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, "SGD: General Analysis and Improved Rates." PMLR, pp. 5200–5209, May 24, 2019. Accessed: Apr. 16, 2023. [Online]. Available: https://proceedings.mlr.press/v97/qian19b.html

[38] S. Bock and M. Weis, "A Proof of Local Convergence for the Adam Optimizer," 2019 International Joint Conference on Neural Networks (IJCNN), Jul. 2019, doi: 10.1109/ijcnn.2019.8852239.

[39] T.-T.-H. Le, J. Kim, and H. Kim, "An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization," 2017 International Conference on Platform Technology and Service (PlatCon), Feb. 2017, doi:10.1109/platcon.2017.7883684.

[40] N. S. Keskar and R. Socher, "Improving Generalization Performance by Switching from Adam to SGD," Dec. 2017.