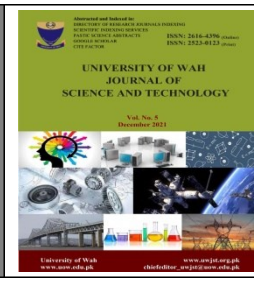
UW Journal of Science and Technology Vol. 5 (2021) 28-33
ISSN: 2523-0123 (Print) 2616-4396 (Online)

University of Wah
Journal of Science and Technology

www.uow.edu.pk



Improved Recognition of Hand-Written Digits Using Convolutional Neural Network (CNN)

Huma Hafeez, Ch Asad Abbas and Muhammad Usman Khan

***Abstract* — Traditional handwriting recognition systems depend on various features and prior knowledge. This making traditional hand writing practical implementation more challenging. Handwriting recognition system has focused deep learning algorithms in recent times and achieved good performance accuracy but still needs improvement in terms of recognition accuracy due to rapid growth of data and massive computations. A wide range of applications uses deep learning. In this paper, we have employed CNN (Convolutional Neural Network) with two different optimizers and five different epoch number to observe the variations of accuracy for classifying different handwritten digits. We have trained seven-layered CNN with Adam and SGDM optimizer on MNIST (Modified National Institute of Standards and Technology) dataset and found that for maximum number of iterations, Adam optimizer outperform the SGDM optimizer in terms of accuracy which is found to be 99.50%, respectively.**

***Index Terms*— Natural Language Processing, Hand-Written Digit Recognition, Convolutional Neural Network, Adam Optimizer, SGDM Optimizer.**

I. INTRODUCTION

HAND-WRITTEN digit recognition plays a significant role in information processing. It aims to transform hand-written digits into the ASCII format readable by machines. Handwriting recognition system being enormously used in vehicle number-plate recognition, bank check truncation system, ID card recognition, Postal address checking and zip code recognition etc.

Manuscript received: September 16,2021; Accepted: December 27, 2021

Huma Hafeez (email: huma.asad918@yahoo.com), Ch Asad Abbas (email: chasad573@yahoo.com) are affiliated with School of Mechanical Engineering, Shandong University, Jinan, China.
Muhammad Usman Khan (email: mukhan.mts19ceme@mts.ceme.edu.pk) is affiliated with (NUST), Islamabad, Pakistan.

Corresponding author email: huma.asad918@yahoo.com

All of them utilizes large datasets and thus, demands higher recognition accuracy with comparatively less computational complexity, achieved using deep learning methods. Deep neural architectures out-performs the shallow neural architectures [1] as it consists of multiple hidden layers, automatically detect the objects' features and can deal with complicated problems of pattern recognition. Deep learning mostly uses CNN [2] due to its applications such as natural language processing, robotics, object detection, regression analysis, pattern recognition, face recognition, image classification, segmentation, fault detection & classification and many more.

Hand-written recognition systems are focus of research nowadays. In research, its proven that CNN with keras, tensorflow etc. give higher accuracy than machine learning algorithms i.e. KNN, SVM etc. form different datasets. Different handwriting datasets are available publically out of which MNIST dataset is considered as benchmark in this domain [3]. It is quite easy to train CNN, extract required features in less computation time and map input to output with distortion, certain degree of rotation and shift invariance [2]. CNN has provided us with good accuracy in offline handwritten digit recognition in different languages i.e. Chinese [4], English [5], Indic Script [6], Arabic [7], Tamil [8] and Urdu [9].

Tavanaei et al. [10] proposed the combination of classical image processing with CNN to unblur images in MNIST dataset and achieved 98% accuracy with 0.1 to 8.5. Loss range and loss functions were derived by Jin et al. [11], implemented in 2D CNN but the accuracies obtained was 93%. Tabil et al. [12] implemented different transformations on dataset and analyzed accuracy for three nets of CNN with LeNeT, DropConnect and Network3. They have found that the elastic and rotation transform combination improved the accuracy upto 0.71% and Ensembles with preprocessing improved it upto 0.74%, respectively. Siddique et al. [13] implemented CNN along with stochastic gradient descent method and back propagation method to observe the improvement in accuracy by changing the number of hidden layers with varying batch-size. Maximum accuracy of 97.32% was achieved with 4 hidden layers and 50 epochs. Enriquez et al. [14] compared traditional methods with CNN and concluded that CNN perform better than traditional methods with the accuracy of 98%.

Lopez et al. [15] implemented four different neural network configurations i.e. CNN-6, CNN-7-BN, VGG-16 and ResNet-18, and found the highest accuracy of 97.53 with ResNet-18 configuration. Nguyen et al. [16] employed multi-scale convolutional neural network and observed that by using the combination of max pooling and attentive-pooling for training CNN, classification needs to be improved. Zhao et al. [17] proposed a system comprising of CNN and multiple classifiers for digit recognition from MNIST dataset and come up with an accuracy of ≥ 98%. Ge et al. [18] described and trained the CNN with gradient descent algorithm for 500 epochs and obtained the accuracy of 95.7%. Younis et al. [19] implemented the DNN on MNIST dataset without any pre-processing step but still, the system gave the accuracy upto 98.46%. Jana et al. [20] proposed the digit recognition system consisting of CNN with two convolutional layers with filter size of 32 and 64, respectively, to improve the accuracy of system upto 98.85%. Hossain et al. [21] described the implementation of CNN with MatConvNet to obtain 99.15% accuracy. Ahlawat et al. [22] evaluated different SGD optimizers to improve recognition accuracy of hand written digit recognition system and suggested Adam optimizer to be applicable in further research as it gives better results as compared to other SGD optimization algorithms.

In this paper, we have implemented convolutional neural network (CNN) along with Adam and SGDM optimizer for different epochs to visualize their effect on recognition accuracy of Hand-written digit recognition system. For training and testing, the proposed network MNIST dataset was employed. CNN classification employed in this paper out-performs other methods suggested in the literature in terms of complexity and accuracy. The training and validation speed was found to be quite efficient making this system applicable in real-time implementation.

II. PROPOSED METHODOLOGY

A. Dataset Collection

We have used publically available MNIST dataset in .csv format available on Kaggle [23]. It consists of 60,000 labelled images in its training dataset and 10,000 labelled images in its testing dataset, each image having the resolution of 28x28 pixels. All the images are in greyscale consisting of 28 x 28 pixels intensity at the centre, so all of these images were flattened into a single dimensional vector of 28x28=784 consisting of binary values. The training dataset file consists of 60,000 rows and 784 columns and its label file consists of 60,000 rows and 10 columns containing the values from 0 to 9. Similarly, the testing dataset file consists of 10,000 rows and 784 columns and its label file consists of 10,000 rows and 10 columns containing the values from 0 to 9. Sample of images in MNIST dataset are presented in Fig. 1.

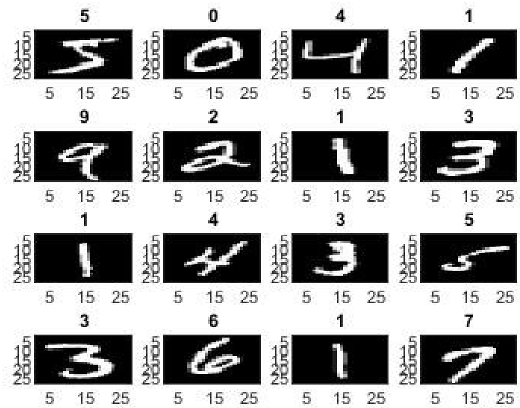


Fig. 1. Samples of MNIST Dataset

B. Convolutional Neural Network (CNN) Architecture

CNN architecture, basically, consists of two main unites i.e. feature extraction unit where each network layer collect output from previous layer and forward the output to next layer as input and feature classification unit for generation of predicted outputs. Overall CNN architecture employed in this research article is displayed in Fig. 2.

The explanation of CNN architecture consists of:

- Layer 1 shows the input image from the dataset consisting of digit having size of 28x28 pixels.
- Layer 2 shows the convolutional layer gets one image 28x28 as input and provide 20 feature maps of size 24x24 as an output and input layer via 5x5 neighborhoods.
- Layer 3 is the ReLU layer where ReLU is an activation function. Input for this layer will be layer 2 and give the output consisting of 20 feature maps of size 24x24. ReLU (r) enhances the performance of model. Let A_j^{x-1} be the input from x-1st layer, $k_{j,l}^{x-1}$ be the kernel and B_l^x be the bias value for 1st layer, then the convolution will be performed as:

$$A_k^x = r\left(A_j^{x-1} * k_{j,l}^{x-1}\right) + B_l^x \quad (1)$$

- Layer 4 is the max-pooling layer, which will sub sample the input as expressed in eq. 2 using the pool size of 2x2 along with stride 2 to reduce the size of output to 20 feature maps of size 12x12.

$$A_k^x = down(A_j^{x-1}) \quad (2)$$

- Layer 5 is fully connected layer having 2880 output nodes. It consists of 10 layers connecting each neuron from previous layer to the next layer.
- Layer 6 is the softmax layer, which is also an activation function. It will enhance the model performance and will classify the output from 0 to 9

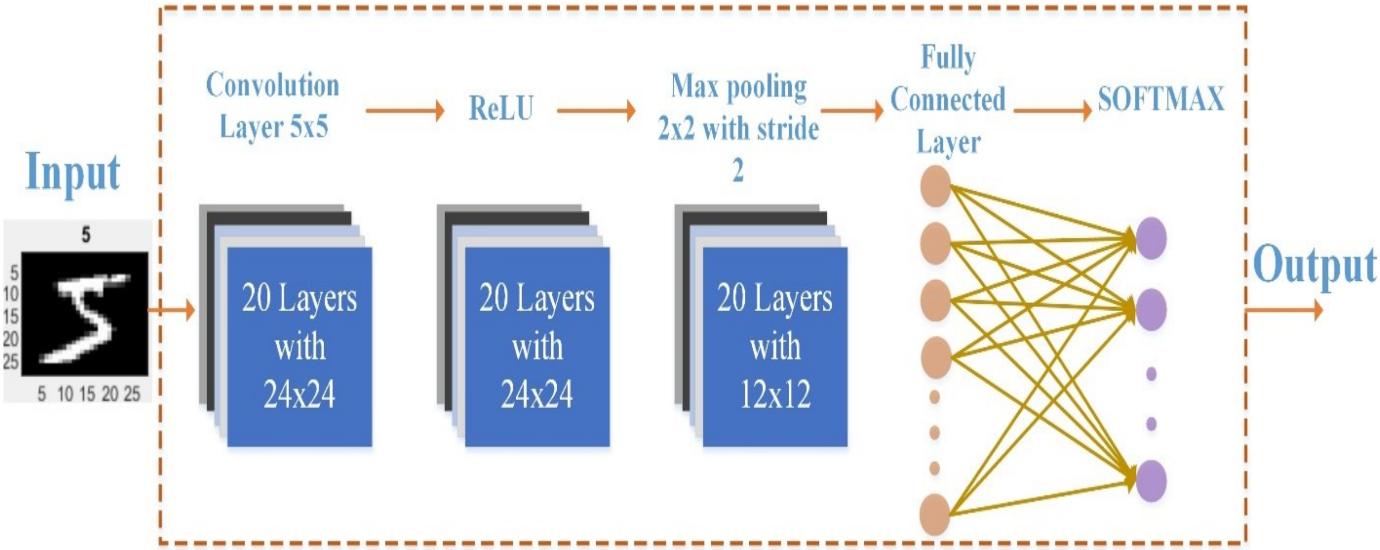


Fig. 2. Proposed CNN Architecture

and convert it into probability that input belongs to which class. Softmax expression for ith class will be:

$$S(b = j|a) = \frac{\exp(a^T w_j)}{\sum_{l=1}^{l} \exp(a^T w_l)} \tag{3}$$

Where a is the ith class input value, w is the weight vector and l represent distinct linear function, respectively.

- Layer 7 is the output layer consisting of 10 neurons, which will give the predicted results. Topology employed in the proposed methodology is explained in Table 1.

TABLE 1
TOPOLOGY OF PROPOSED CNN ARCHITECTURE

Layer	Description	Activations	Learnable
imageinput	28x28x1 images along with 'zerocenter' normalization	28x28x1	-
conv	20 5x5 convolutions along with stride [1 1]	24x24x20	Weights: 5x5x1x20 Bias: 1x1x20
relu	ReLU	24x24x20	-
maxpool	2x2 max pooling along with stride [2 2]	12x12x20	-
fc	10 fully connected layer	1x1x10	Weights: 10x2880 Bias: 10x1
softmax	softmax	1x1x10	-
classoutput	crossentropyex	-	-

C. Convolutional Neural Network (CNN) Training

The CNN was trained by applying optimizers i.e. SGDM (Stochastic Gradient Descent with Momentum) and Adam (Adaptive Moment Estimation). In SGDM [2], to reduce the oscillations and for faster convergence of the network, the momentum parameter is updated. The updated equation with momentum (m) will be:

$$v(t) = mv(t-1) + \eta \nabla E(\theta) \tag{4}$$

$$\theta = \theta - v(t) \tag{5}$$

Adam [3] optimizer update the learning weight and average the parameters of gradient and squared gradient values. The updated equation will be:

$$\vartheta_{t+1,k} = \vartheta_{t,k} - \frac{\eta}{\sqrt{u_t + \varepsilon}} . m_t \tag{6}$$

Where m_t represents the first-moment value, given as:

$$m_t = \frac{m_t}{1-\beta_1^t} \tag{7}$$

Moreover, u_t represents the gradient variance value, given by:

$$u_t = \frac{u_t}{1-\beta_1^t} \tag{8}$$

ε denotes smoothing parameter, respectively. The parameters chosen for training CNN using Adam and SGDM optimizer are expressed in Table 2.

TABLE 2
CONFIGURATION OF PROPOSED OPTIMIZER MODELS

Detector Optimizer	Batch Size	Epoch	Learning Rate
Adam	100	25	0.001
	100	50	0.001
	100	75	0.001
	100	100	0.001
	100	125	0.001
SGDM	100	25	0.001
	100	50	0.001
	100	75	0.001
	100	100	0.001
	100	125	0.001

III. EXPERIMENTAL RESULTS

CNN trained on the system consisting of Windows 10 Pro, Intel® Core™ i5-6200U CPU @ 2.30GHz processor with 8GB RAM and MATLAB 2020b. MNIST dataset is used for training, validation and testing of proposed methodology. 60,000 images dataset is used for training

TABLE 3
PERFORMANCE OF CNN FOR DIFFERENT OPTIMIZER METHODS

EPOCH	**No. of Iterations**	**Learning Rate**	**Optimizers Performances**							
			ADAM				***SGDM***			
			Mini-Batch Accuracy	*Validation Accuracy*	*Mini-Batch Loss*	*Validation Loss*	*Mini-Batch Accuracy*	*Validation Accuracy*	*Mini-Batch Loss*	*Validation Loss*
25	15000	0.001	100	97.60	0.0213	0.0986	95	93.10	0.2394	0.3133
50	30000	0.001	100	99.0	0.0056	0.0455	100	95.70	0.0740	0.1705
75	45000	0.001	100	99.10	0.0044	0.0425	100	97.70	0.0280	0.1149
100	60000	0.001	100	99.40	0.0043	0.0298	100	97.70	0.0253	0.1011
125	75000	0.001	100	**99.50**	0.0032	0.0309	100	98.80	0.0048	0.0414

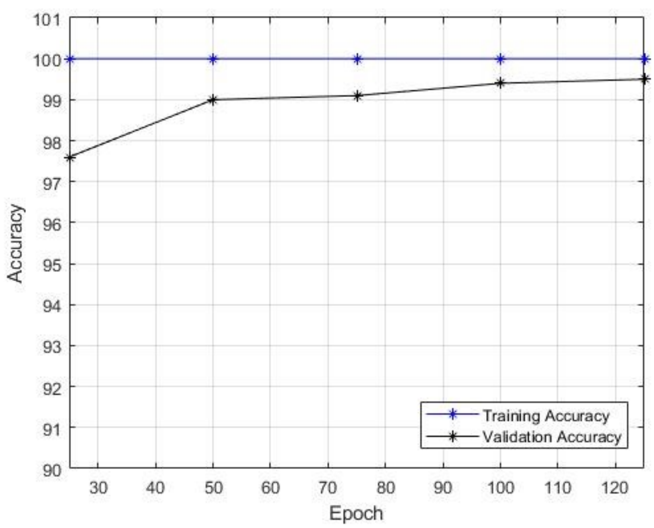


Fig. 3. Adam Optimizer Training & Validation Accuracy

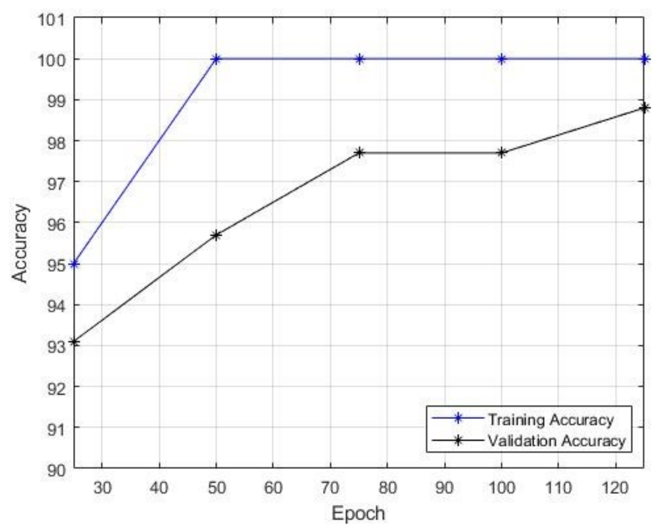


Fig. 4. SGDM Optimizer Training & ValidationAccuracy

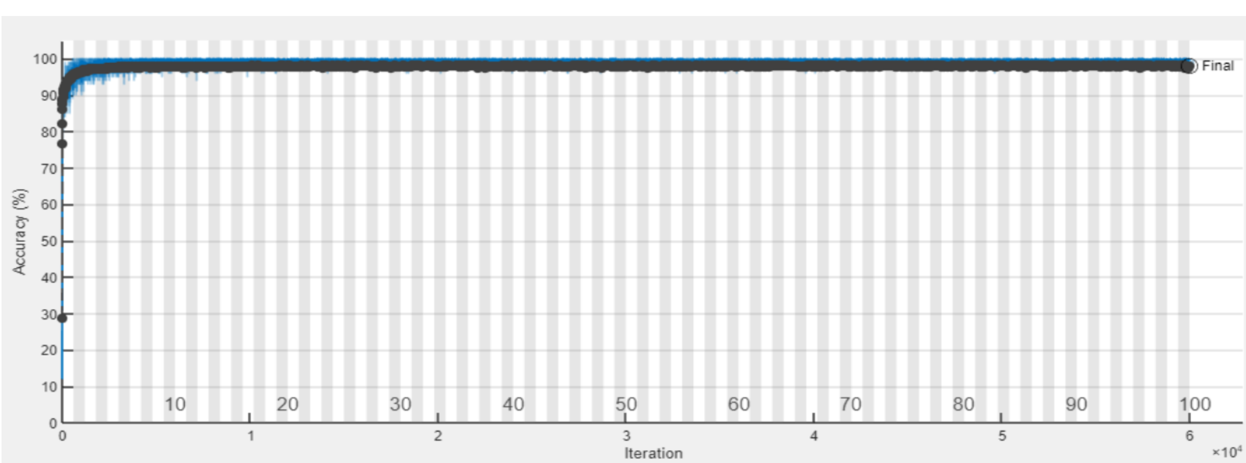


Fig. 5. Accuracy Curve for Taining and Validation using Adam Optimizer

an 10,000 images dataset is used for testing purpose. CNN, specifically trained for different parameters of Adam and SGDM optimizers and the results obtained after training Adam and SGDM optimizers for same learning rate and five different epoch number are expressed in Table 3. Table 3 represented that for same epoch number and for same learning rate, accuracy of Adam optimizer is greater than that of SGDM optimizer and as the number of epoch is increased, validation accuracy is also increased, but still, the accuracy of Adam optimizer is higher than that of SGDM optimizer in each case. Training loss as well as validation loss also decreases as the number of iteration is increased. Training loss for Adam optimizer with 125 epochs and 75000 iterations with 0.001 learning rate is found to be 0.0032 which is quite less than that for SGDM optimizer=0.0048. Similarly, Validation loss of Adam is equal to 0.0309, which is also less than that of SGDM validation loss of

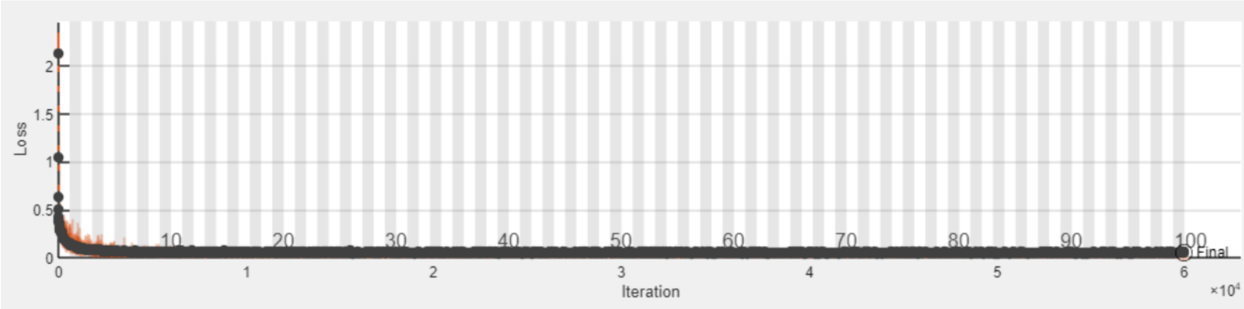


Fig. 6. SGDM Curve for Taining and Validation using Adam Optimizer

0.0414. Hence, Adam optimizer results are selected for further validation and testing process.

Fig. 3 and Fig. 4 show the Adam optimizer training and testing accuracy along with SGDM optimizer training and testing accuracy. Fig. 5 and Fig. 6 describe accuracy curve and loss curve for Adam optimizer training for 100 epochs. As a result, the proposed scheme achieved the higher accuracy of 99.50% with Adam optimizer and 98.80% with SGDM optimizer for MNIST dataset by using seven-layered CNN. Accuracy of digit recognition depends upon size of dataset, shape and writing style, number of layers and optimizer selected for training. Table 4 shows the comparison of obtained accuracy to the previously proposed methodologies in literature.

TABLE 4
CONFIGURATION OF PROPOSED OPTIMIZER MODELS

Authors	**Methodology**	**Accuracy (%)**	**Reference**
Siddique et al.	Artificial Neural Network (ANN)	97.32	[4]
Lopez et al.	ResNet-18	97.53	[5]
Enriquez et al.	LeNet-5	98	[6]
Hui-huang Zhao et al.	CNN with multiple classifiers	≥ 98	[7]
Younis et al.	CNN	98.46	[8]
Jana et al.	CNN+Keras	98.85	[9]
Hossain et al.	CNN	99.15	[10]
Saqib Ali et al.	CNN+DL4J	99.21	[11]
Proposed work	CNN with SGDM	98.80	--
Proposed work	CNN with Adam	**99.50**	--

IV. CONCLUSION

In this paper, it is observed that the network performance not only depend upon the image data but also on the network architecture being used. Variation in accuracy of network with seven-layered CNN and different optimizers is also observed for different number of iterations. The accuracy curves are generated for two different cases i.e. CNN with Adam optimizer and CNN with SGDM optimizer for MNIST dataset. Both cases performed differently because of different optimizer implementation. Epoch are selected randomly in periodic manner to observe improve in accuracy for both cases. Maximum accuracy observed for CNN with SGDM optimizer is 98.80% with cross entropy of 0.0048. Similarly, maximum accuracy observed for CNN with Adam optimizer is 99.50% with cross entropy of 0.0032 at 125 epochs, 75000 numbers of iterations and 0.001 learning rate. This lesser value of cross-entropy make proposed method outperforms the state-of-art methods. Our proposed method can recognize the handwritten digit with accuracy of 99.50%. In future, we will vary the number of layers to observe its effect on overall accuracy of system.

REFERENCES

[1] Xiao, J., Zhu, X., Huang, C., Yang, X., Wen, F., & Zhong, M. (2019). A new approach for stock price analysis and prediction based on SSA and SVM. International Journal of Information Technology & Decision Making, 18(01), 287-310.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[3] Lee, S., Son, K., Kim, H., & Park, J. (2017, July). Car plate recognition based on CNN using embedded system with GPU. In 2017 10th International Conference on Human System Interactions (HSI) (pp. 239-241). IEEE.

[4] Wu, Y. C., Yin, F., & Liu, C. L. (2017). Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models. Pattern Recognition, 65, 251-264.

[5] Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017, May). EMNIST: Extending MNIST to handwritten letters. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2921-2926). IEEE.

[6] Sarkhel, R., Das, N., Das, A., Kundu, M., & Nasipuri, M. (2017). A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. Pattern Recognition, 71, 78-93.

[7] Boufenar, C., Kerboua, A., & Batouche, M. (2018). Investigation on deep learning for off-line handwritten Arabic character recognition. Cognitive Systems Research, 50, 180-195.

[8] Kavitha, B. R., & Srimathi, C. (2019). Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks. Journal of King Saud University-Computer and Information Sciences.

[9] Husnain, M., Saad Missen, M. M., Mumtaz, S., Jhanidr, M. Z., Coustaty, M., Muzzamil Luqman, M., ... & Sang Choi, G. (2019). Recognition of Urdu handwritten characters using convolutional neural network. Applied Sciences, 9(13), 2758.

[10] Tavanaei, A., & Maida, A. S. (2017, May). Multi-layer unsupervised learning in a spiking convolutional neural network. In 2017 international joint conference on neural networks (IJCNN) (pp. 2023-2030). IEEE.

[11] Jin, J., Fu, K., & Zhang, C. (2014). Traffic sign recognition with hinge loss trained convolutional neural networks. IEEE Transactions on Intelligent Transportation Systems, 15(5), 1991-2000.

[12] Tabik, S., Peralta, D., Herrera-Poyatos, A., & Herrera, F. (2017). A snapshot of image pre-processing for convolutional neural networks: case study of MNIST. International Journal of Computational Intelligence Systems, 10(1), 555-568.

[13] Siddique, M. A. B., Khan, M. M. R., Arif, R. B., & Ashrafi, Z. (2018, September). Study and observation of the variations of accuracies for handwritten digits recognition with various hidden layers and epochs using neural network algorithm. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT) (pp. 118-123). IEEE.

[14] Enriquez, E. A., Gordillo, N., Bergasa, L. M., Romera, E., & Huélamo, C. G. (2018, November). Convolutional neural network vs traditional methods for offline recognition of handwritten digits. In Workshop of Physical Agents (pp. 87-99). Springer, Cham.

[15] Lopez, B., Nguyen, M. A., & Walia, A. (2019). Modified MNIST.

[16] Nguyen, C. T., Khuong, V. T. M., Nguyen, H. T., & Nakagawa, M. (2020). CNN based spatial classification features for clustering offline handwritten mathematical expressions. Pattern Recognition Letters, 131, 113-120.

[17] Zhao, H. H., & Liu, H. (2020). Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. Granular Computing, 5(3), 411-418.

[18] Ge, D. Y., Yao, X. F., Xiang, W. J., Wen, X. J., & Liu, E. C. (2019, October). Design of high accuracy detector for MNIST handwritten digit recognition based on convolutional neural network. In 2019 12th International Conference on Intelligent Computation Technology and Automation (ICICTA) (pp. 658-662). IEEE.

[19] Younis, K. S., & Alkhateeb, A. A. (2017). A new implementation of deep neural networks for optical character recognition and face recognition. Proceedings of the new trends in information technology, 157-162.

[20] Jana, R., & Bhattacharyya, S. (2019). Character recognition from handwritten image using convolutional neural networks. In Recent Trends in Signal and Image Processing (pp. 23-30). Springer, Singapore.

[21] Hossain, M. A., & Ali, M. M. (2019). Recognition of handwritten digit using convolutional neural network (CNN). Global Journal of Computer Science and Technology.

[22] Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). Sensors, 20(12), 3344.

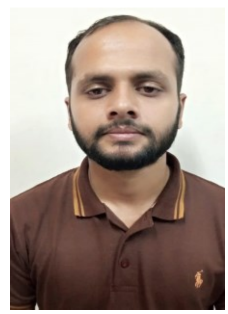
[23] https://www.kaggle.com/oddrationale/mnist-in-csv

[24] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

[25] Ali, S., Shaukat, Z., Azeem, M., Sakhawat, Z., Mahmood, T., & ur Rehman, K. (2019). An efficient and improved scheme for handwritten digit recognition based on convolutional neural network. SN Applied Sciences, 1(9), 1-9.

Huma Hafeez did her bachelor degree in Mechatronics Engineering with Gold Medal in 2018 from UET Taxila, Pakistan. She completed her Master's degree in Mechatronics Engineering from Shandong University, Jinan, China in 2021. Her research interest includes Image Processing, Deep Learning, Machine Vision and Artificial Intelligence.

Ch Asad Abbas has obtained his bachelor degree in Mechatronics Engineering from UET Taxila, Pakistan in 2014. He got his Master's degree in Mechatronics Engineering from Shandong University, China in 2018. Currently, he is doing his PhD from Shandong University, Jinan, China.



Muhammad Usman Khan completed his bachelor degree from UET Taxila, Pakistan in 2018. Currently, he is doing his Master's degree in Mechatronics Engineering from College of Electrical & Mechanical Engineering (CEME), National University of Science and Technology (NUST), Pakistan.