Comparative Analysis of Handwritten Digit Recognition Using Logistic Regression, SVM, KNN and CNN Algorithms

Dr. R. Pradeep Kumar Reddy[1], Prof. C. Naga Raju[2]
[1](Department of Computer Science and Engineering, Y.S.R Engineering College of YVU, India)
[2](Department of Computer Science and Engineering, Y.S.R Engineering College of YVU, India)
Corresponding Author: pradeepmadhavi@gmail.com

To Cite this Article
Dr. R. Pradeep Kumar Reddy and Dr. C. Naga Raju, "Comparative Analysis of Hand written Digit Recognition Using Logistic Regression, SVM, KNN and CNN Algorithms", Journal of Science and Technology, Vol. 06, Issue 06, Nov-Dec 2021.

Article Info
Received: 12-12-2021 Revised: 10-12-2021 Accepted: 14-12-2021 Published: 18-12-2021

Abstract: *The style of handwriting varies from person to person. Handwritten numbers are not always the same size, orientation and width. To develop a system to understand this, the machine recognizes handwritten digit images and classifies them into 10 digits (from 0 to 9).Handwritten digit recognition is a technology which is used for automatic recognizing and detecting handwritten digital data through various machine learning models. This paper uses a different machine learning algorithms to improve productivity and a variety of models to reduce complexity. Machine Learning is an artificial intelligence application which learns from previous experiences and it automatically improves with the previous experiences. This paper is about recognizing handwritten digits from 0 to 9 from the well-known Modified National Institute of Standards and Technology(MNIST) dataset, then comparison takes place between machine learning algorithms like Support Vector Machine(SVM), logistic regression, K-Nearest Neighbor (KNN) and deep learning algorithm like CNN.*

Key Words: *Support Vector Machine, KNN, Logistic Regression, CNN, Handwritten digit images.*

I. Introduction

A handwritten digit recognition system is how a machine works to recognize handwritten digits from a variety of sources such as documents and images of handwritten digits. Handwritten digit recognition is a feature of a computer or machine that can recognize numbers/digits handwritten by humans. In order to develop a system for understanding handwritten numbers, a machine for recognizing and classifying images of handwritten numbers into 10 digits (0-9) is included. Handwritten numbers are not always the same size, width, orientation and justified at the edge, the individuality and difference of the handwriting of different people also influence the structure and appearance of the digits.

The idea of reading handwritten numbers, letters, and words through a computer system is often referred to as imitating humans. Artificial intelligence is used to read handwritten digits from any handwriting source or images. Artificial intelligence (AI) is one of the subfields of computer science that deals with simulating the intelligent behavior of the computers. Machine learning is one of the applications of Artificial Intelligence, it provides the ability for a system to automatically learn from the environment and use that learning to make better decisions.

In this paper, we compare machine learning algorithms such as logistic regression, SVM, K-nearest neighbor and deep learning algorithm CNN with the dataset MNIST, test the efficiency of these algorithms and find precision, recall, f1-score and accuracy are found for all these algorithms and compared.

II. Literature Review

Wang et.al[1] – focuses on Study of the efficiency of quantum computing by using Grover Algorithm and KNearest Algorithm. Khanet.al[9] - In this paper, This multicell size (MCS) uses a histogram of directed gradient (HOG) using the Support Vector Machine (SVM) algorithm to identify digits from the MNIST digital dataset. In the experiment, the accuracy was determined using cross validation with 10 fold and an independent test set for verification, and both accuracy was found to be 99.26% and 99.36%, respectively.

D. Ge et.al[4] (Development of a high precision handwritten digit recognition detector based on a Convolutional-Neural Network) –focuses on determining accuracy and efficiency using a Convolutional neural network of two 32 image layers and another 64 images with some neurons in each layer. After completing the training with the dataset, found that the neural network accuracy was 92.6% for the training set and 90.1% for the test set. Convolutional neural networks are highly accurate in deep learning models that provide excellent performance.

Al-Wzwazy et.al[5] - We have optimized various SGD algorithms to improve the efficiency of handwritten digit recognition. This paper discusses various options such as stride size, kernel, padding, number of layers, and receptive. The MNIST dataset is used for testing. With the CNN algorithm, the training parameter has a learning rate of 0.01 and a maximum epoch number of 4. The accuracy of the 3rd layer Convolutional neural network is 99.76%, and the accuracy of the 4th layer (CNN) detection is the highest(99.76%). For the MINST dataset, the CNN 3-layer model with the optimizer achieved the highest accuracy of 99.89%.

III. Methodology

Procedure

- We have chosen a MNIST dataset from Kaggle website for Handwritten Digit Recognition.
- Preprocessing of data.
- We compared machine learning algorithms such as logistic regression, Support Vector Machine, K-nearest neighbor and deep learning algorithm CNN for MNIST dataset.
- Test the efficiency of these algorithms and find precision, recall, f1-score and accuracy.

Dataset

The dataset used in the Handwritten digit Recognition is the MNIST dataset. The Modified National Institute of Standards and Technology (MNIST) is a Database of the handwritten digits. This is a subset of the larger set which is available from National Institute of Standards and Technology (NIST).

The MNIST database provides researchers with a simple static classification task to help analyze machine learning and pattern recognition techniques. The number of digit classes and a sample Handwritten digit images from MNIST test dataset are as shown in Fig. 1 and Fig. 2 [2].

It consists of 60,000 examples as a train set and 10,000 examples as a test set. The dataset is based on grayscale images of handwritten numbers, each image having a height and width of 28 x 28 pixels. Each pixel is assigned a value, where 0 represents a dark pixel and 255 represent a white pixel.

Both the train and test datasets have 785 columns, the first column is the "label" for handwritten digits (numbers from 0 to 9), and the remaining 784 values are for pixel values. In the test, the data is individually labeled to predict the value.

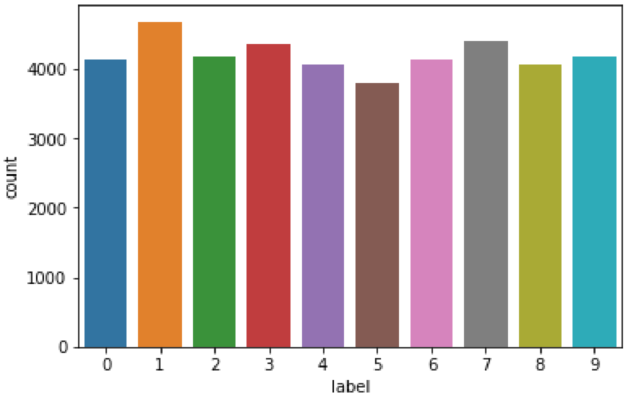


Fig1: Number of digit classes

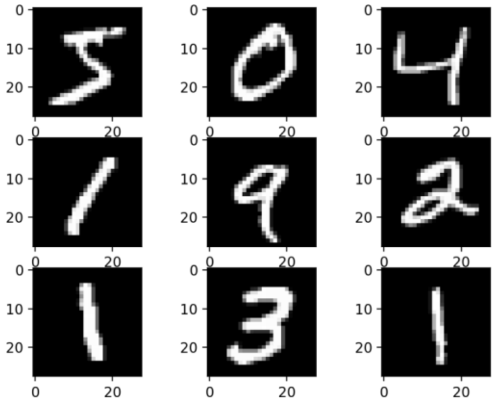
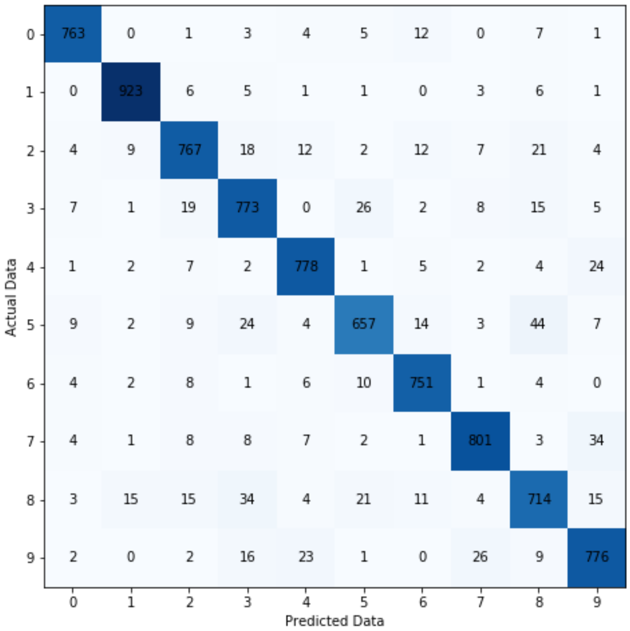


Fig 2: Handwritten digit images from MNIST test dataset.

Logistic Regression

Logistic Regression is primarily used to handle with the classification problems. This is one of the most popular machine learning algorithms and is a supervised learning technique. Given a set of independent variables, this algorithm can be used to predict categorical dependent variables. It can be used to classify observations based on different data types, making it easy to determine the most effective variables to use for classification. Logistic regression include some benefits such as Easy implementation, efficiency in terms of training, simple regularization and computational efficiency [6].

The preprocessed training dataset is supplied as input to the logistic regression model to obtain the training accuracy of the classifier. After training, the test dataset is passed to the classifier to determine the accuracy of the test. The confusion matrix for test data is represented in Fig. 3. Further, Classification reports are used to display metrics such as precision, recall, and f1 score for each class (0-9). The algorithm obtained a test accuracy of approximately 91.7%.



	0	1	2	3	4	5	6	7	8	9
0	763	0	1	3	4	5	12	0	7	1
1	0	923	6	5	1	1	0	3	6	1
2	4	9	767	18	12	2	12	7	21	4
3	7	1	19	773	0	26	2	8	15	5
4	1	2	7	2	778	1	5	2	4	24
5	9	2	9	24	4	657	14	3	44	7
6	4	2	8	1	6	10	751	1	4	0
7	4	1	8	8	7	2	1	801	3	34
8	3	15	15	34	4	21	11	4	714	15
9	2	0	2	16	23	1	0	26	9	776

Fig 3: Confusion matrix using logistic regression of test dataset

Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm. The aim of the SVM algorithm is to create optimal lines or decision boundaries that can divide n-dimensional space into classes so that new data points can be easily placed in the correct category. This best decision boundary is called the hyperplane represented in Fig. 4. In this way, we usually plot the data items in n dimensional space where n is the number of objects, a particular coordinate represents the value of the object, we perform the classification by finding the discriminant hyperplane double layer. It will select the hyperplane that separates the layers correctly. SVM chooses extreme vectors to create hyperplane. These extreme cases are called support vectors and hence the algorithm is called a support vector machine. There are mainly two types of SVM, linear and nonlinear SVM. In this article, we have used Linear SVM to recognize handwritten digits.

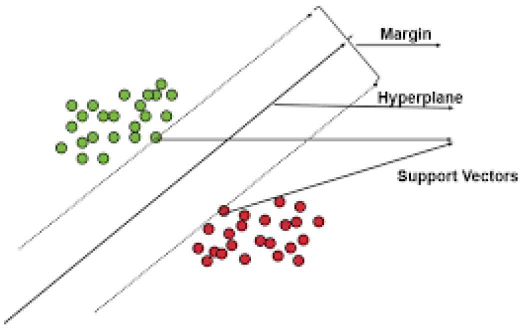
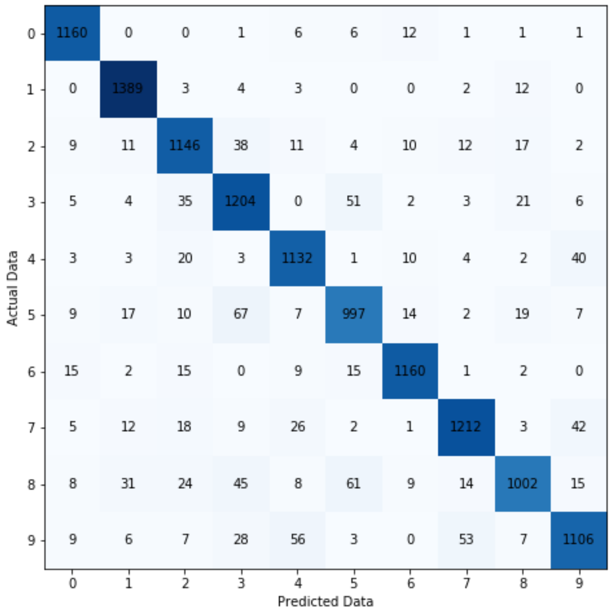


Fig 4: Classification of SVM with hyperplanes and supporting vectors

A preprocessed training dataset is provided as input to the SVM model to train the classifier. After training, the test dataset is passed to the classifier to predict the label and determine the accuracy of the test. The probability between the original data and the predicted data is expressed using a confusion matrix represented inn Fig. 5. The accuracy, precision, recall, and f1 score for each class (0-9) are determined using a classification matrix. The test dataset achieved an accuracy of approximately 91.3% using the decision tree classifier.



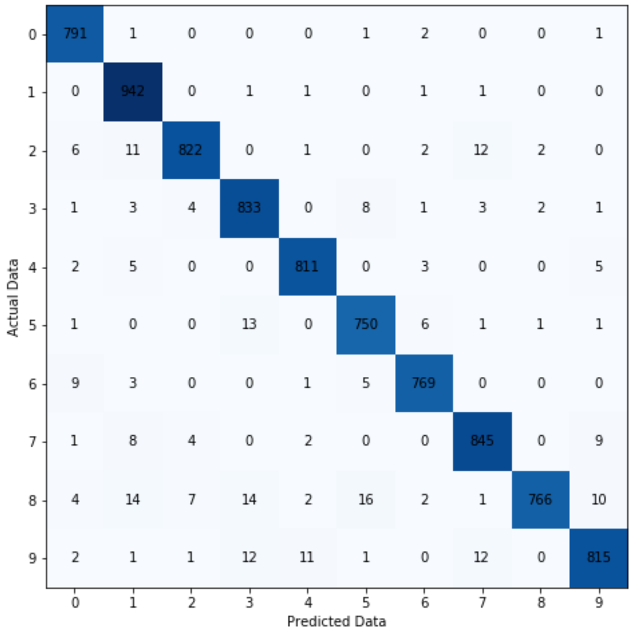
	0	1	2	3	4	5	6	7	8	9
0	1160	0	0	1	6	6	12	1	1	1
1	0	1389	3	4	3	0	0	2	12	0
2	9	11	1146	38	11	4	10	12	17	2
3	5	4	35	1204	0	51	2	3	21	6
4	3	3	20	3	1132	1	10	4	2	40
5	9	17	10	67	7	997	14	2	19	7
6	15	2	15	0	9	15	1160	1	2	0
7	5	12	18	9	26	2	1	1212	3	42
8	8	31	24	45	8	61	9	14	1002	15
9	9	6	7	28	56	3	0	53	7	1106

Fig 5: Confusion matrix using SVM of test dataset

K- Nearest Neighbor (KNN)

K-Nearest Neighbor is simple Machine Learning Algorithm that assumes the similarity between new data and available data then it keeps the new data in the category which is closest to the available categories. The training phase saves the dataset and, as new data is added, categorizes the data into categories that closely resemble the new data. There are two main advantages to using the KNN algorithm. That is, it is robust against noisy training data and very efficient when the data is very large [7]. KNN is a Lazy learning algorithm i.e. there is no special training of data in the training phase and all the data available for training is used. It is a non-parametric learning algorithm because it does not assume anything about the data.

The preprocessed training dataset is supplied as input to the KNN model to obtain the training accuracy of the classifier by using confusion matrix represented in Fig. 6. After training, the test dataset is passed to the classifier to determine the accuracy of the test. Classification reports are used to display metrics such as precision, recall, and f1 score for each class (0-9). The algorithm obtained a test accuracy of approximately 96.9%.



Actual Data \ Predicted Data	0	1	2	3	4	5	6	7	8	9
0	791	1	0	0	0	1	2	0	0	1
1	0	942	0	1	1	0	1	1	0	0
2	6	11	822	0	1	0	2	12	2	0
3	1	3	4	833	0	8	1	3	2	1
4	2	5	0	0	811	0	3	0	0	5
5	1	0	0	13	0	750	6	1	1	1
6	9	3	0	0	1	5	769	0	0	0
7	1	8	4	0	2	0	0	845	0	9
8	4	14	7	14	2	16	2	1	766	10
9	2	1	1	12	11	1	0	12	0	815

Fig. 6: Confusion matrix using KNN of test dataset

Convolutional Neural Networks

A convolutional neural network (CNN) is a special type of neural network model designed to process 2D image data, but it can also be used with 1D and 3D data. The main idea of a convolutional neural network is to alternate between convolutional and sub sampling layers at the output, using a multi-layer perceptron [8]. Takes the image as input and categorizes it into a specific category for processing. A CNN consists of an input layer, an output layer and a hidden layer. Hidden layers include convolution layers, high density layers, and so on, the flow of CNN layer architecture design is as shown in Fig. 7.

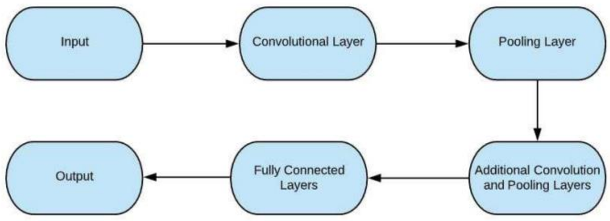
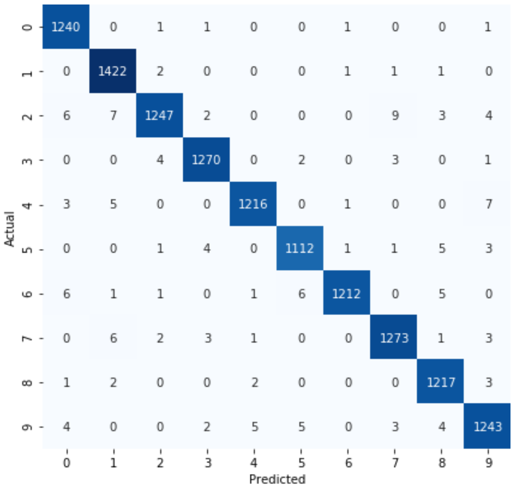


Fig 7: Flowchart representation of CNN layer architecture design.

We used about 8 hidden layers. These include convolution, pooling, flattening, and dropout layer. Two layers of containment are used to reduce over fitting of data. We use a lot of sizes 128 and 12 epochs. The first epoch reached approximately 81.48% training accuracy, and then increased in the next epoch. The model achieves a training accuracy of approximately 99.9% and a testing accuracy of 99% obtained by using confusion matrix, it is as shown in Fig. 8.



Actual \ Predicted	0	1	2	3	4	5	6	7	8	9
0	1240	0	1	1	0	0	1	0	0	1
1	0	1422	2	0	0	0	1	1	1	0
2	6	7	1247	2	0	0	0	9	3	4
3	0	0	4	1270	0	2	0	3	0	1
4	3	5	0	0	1216	0	1	0	0	7
5	0	0	1	4	0	1112	1	1	5	3
6	6	1	1	0	1	6	1212	0	5	0
7	0	6	2	3	1	0	0	1273	1	3
8	1	2	0	0	2	0	0	0	1217	3
9	4	0	0	2	5	5	0	3	4	1243

Fig 8: Confusion matrix using CNN of test dataset

IV. Results

In this paper, we trained and tested the handwritten digits from the MNIST database by using various machine learning algorithms such as Decision Trees, logistic regression, K-Nearest Neighbor, and deep learning CNN algorithm. 60000 samples will be used to train the model and 10000 samples will be used to test the model. Models are compared based on accuracy, precision, recall, and f1 score, Fig. 9 shows the graphical representations of different algorithm based accuracy.

For CNN we run about 10 epochs for training and in epoch 1 the training accuracy was found to be 95.3%, increasing to 98.6% in epoch 10. People are concerned observed that the CNN model took longer to train than the machine learning models, but gave 99% better test accuracy than other models. With the value predictions of all (0-9) classes of the confusion matrix, CNN predicts the value more accurately than other models. In machine learning algorithms, K Nearest Neighbor is found to provide better model accuracy i.e. 96.8% and support vector machine algorithms.

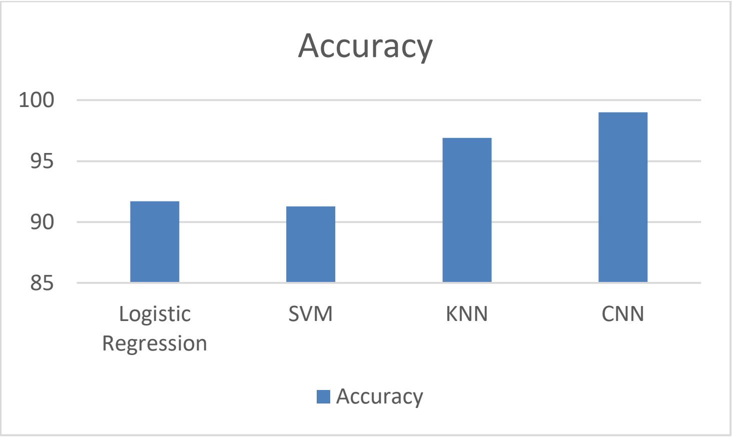


Fig. 9: Graphical representation of different algorithms based on Accuracy.

Table 1: weighted average of Precision, Recall, f1 score & Accuracy of Logistic Regression, Decision Tree, KNN and CNN Algorithms.

Algorithm	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Logistic Regression	0.92	0.92	0.92	91.7%
SVM	0.91	0.91	0.91	91.3%
KNN	0.97	0.97	0.97	96.9%
CNN	0.99	0.99	0.99	99%

V. Conclusion

In this Handwritten Digit Recognition by Using Comparative Analysis of Machine Learning Algorithms such as SVM, Logistic Regression, KNN and a Deep Learning Algorithm like CNN, we implemented these four models for handwritten digit recognition using MNIST dataset based on Deep Learning algorithm and Machine Learning algorithms. We compared them based on their characteristics and ranked the most accurate model among the four models. CNN has been found to provide the most accurate results for handwritten digit recognition. From this, we conclude that CNN is best suited for all types of prediction problems, including image data as a input then by comparing the execution times of the algorithms, we come to the conclusion that it makes no sense to increase the number of epochs without changing the configuration of the algorithm due to the limitations of a particular model, and over after a certain number of epochs. Start the dataset with a over fitting and make a biased prediction**.**

References

[1] Wang, Y., Wang, R., Li, D. et al. Improved Handwritten Digit Recognition using Quantum K-Nearest Neighbour Algorithm. Int J Theor Phys 58, 2331–2340 (2019).

[2] I. J. S. G. D. G. Anchit Shrivastava, "Handwritten Digit Recognition Using Machine Learning : A Review," pp. 322-326, 2019.

[3] Assegie, Tsehay & Nair, Pramod. (2019). Handwritten digits recognition with decision tree classification: a machine learning approach. International Journal of Electrical and Computer Engineering (IJECE). 9. 4446. 10.11591/ijece.v9i5.pp4446-4451. K. Elissa, "Title of paper if known," unpublished.

[4] D. Ge, X. Yao, W. Xiang, X. Wen and E. Liu, "Design of High Accuracy Detector for MNIST Handwritten Digit Recognition Based on Convolutional Neural Network," 2019 12th International Conference on Intelligent Computation Technology and Automation (ICICTA), Xiangtan, China, 2019, pp. 658-662, doi: 10.1109/ICICTA49267.2019.00145.

[5] Al-Wzwazy, Haider. (2016). Handwritten Digit Recognition Using Convolutional Neural Networks. International Journal of Innovative Research in Computer and Communication Engineering.

[6] S. Ray, "A Quick Review of Machine Learning Algorithms," IEEE, pp. 35 - 39, 2019.

[7] B. K. Vijayalaxmi R Rudraswamimath, "Handwritten Digit Recognition using CNN," International Journal of Innovative Science and Research Technology (IJISRT), vol. 4, no. 6, pp. 182-187, June 2019.

[8] D. T. Zufar Kayumov, "Convolution Neural Network Learning Features for Handwritten Digit Recognition," IEEE, 2020.

[9] Khan, H. (2017) MCS HOG Features and SVM Based Handwritten Digit Recognition System. Journal of Intelligent Learning Systems and Applications, 9, 21-33.

[10] S. S. M. A. B. S. Fathima Siddique, "Recognition of Handwritten Digit using Convolutional Neural Network in python with Tensorflow and Comparison of Performance for Various Hidden Layers," IEEE , pp. 541-546, 2019.

[11] E. ,. S. J.Pradeep, "Neural Network based Handwritten Character Recognition system without feature extraction," IEEE,International Conference on Computer, Communication and Electrical Technology – ICCCET, pp. 40-44, 2011.

[12] P. H. ,. a. D. L. Junyi Tang, "Adhesive Handwritten Digit Recognition Algorithm Based on Improved Convolutional Neural Network," IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), pp. 388-392, 2020.

[13] P. a. P. S. M.Rajalakshmi, "Pattern Recognition - Recognition of Handwritten Document Using Convolutional Neural Networks," IEEE, 2019.

[14] B. Mahesh, "Machine Learning Algorithms - A Review," International Journal of Science and Research (IJSR), vol. 9, no. 1, pp. 381-386, 2018.

[15] A. M. Z. ,. A. M. Z. Mustafa Ali Abuzaraida, "Online Recognition System for Handwritten Hindi Digits Based On Matching Alignment Algorithm," IEEE, 2015.

[16] B. K. K. M. A. A. Nurseitov Daniyar, "Classification of handwritten names of cities using various deep learning models," IEEE ,15th International Conference on Electronics Computer and Computation (ICECCO), 2019.