2021 International Conference on Emerging Smart Computing and Informatics (ESCI)
AISSMS Institute of Information Technology, Pune, India. Mar 5-7, 2021

2021 International Conference on Emerging Smart Computing and Informatics (ESCI) | 978-1-7281-8519-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/ESCI50559.2021.9396870

Handwritten Digit Recognition of MNIST dataset using Deep Learning state-of-the-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN)

Drishti Beohar
Computer Science Department
Maulana Azad National Institute of Technology,
Bhopal, India
beohardrishti25@gmail.com

Akhtar Rasool
Computer Science Department
Maulana Azad National Institute of Technology,
Bhopal, India
akki262@gmail.com

***Abstract* -- Handwritten digit recognition is an intricate assignment that is vital for developing applications, in computer vision digit recognition is one of the major applications. There has been a copious exploration done in the Handwritten Character Recognition utilizing different deep learning models. Deep learning is rapidly increasing in demand due to its resemblance to the human brain. The two major Deep learning algorithms Artificial Neural Network and Convolutional Neural Network which have been compared in this paper considering their feature extraction and classification stages of recognition. The models were trained using categorical cross-entropy loss and ADAM optimizer on the MNIST dataset. Backpropagation along with Gradient Descent is being used to train the networks along with reLU activations in the network which do automatic feature extraction. In neural networks, Convolution Neural Network (ConvNets or Convolutional neural networks) is one of the primary classifiers to do image recognition, image classification tasks in Computer Vision.**

Keywords—Handwritten digit recognition, Convolutional Neural Network (CNN), Deep learning, MNIST dataset, Epochs, Hidden Layers, Stochastic Gradient Descent, Backpropagation

I. INTRODUCTION

At whatever point we have heard the word Handwritten Character Recognition, the initial term that rings a bell is OCR, termed for Optical Character Recognition, it is a system of reading characters and manipulating them into a structure that a machine can interpret. This was the solitary innovation utilized by the scientists for 3-4 decades to change over any actual record into machine editable structure. OCR is a pipeline of 5 phases i.e stage one is **Image Acquisition**, after which stage two of **Pre-processing** is cone stage three is **Segmentation**. After segmentation, the **Feature Extraction** of the data is done and the final stage i.e. **Classification**. OCR follows the pipeline idea, progressive paces of each stage rely on the success pace of the past stage. With the headway of innovation, we need the machine to perform the most extreme undertakings. There are various uses of computer vision like

a) interpretation of the report,

b) language translation,

c) arranging mail,

d) perusing checks and so forth.

That is the reason OCR is a region of enthusiasm for a large number of analysts as of late. Fundamentally, writings being in various dialects and contents, the efficiency of the recognition system to a great extent relies on the classification strategies utilized. Consequently, the improvement of such a framework was an outrageous task. Artificial intelligence has opened many doors before this AI era. Anyone who wants to design a classifier has to manually do the feature extraction and feel the machine gives the decision based on the features feed and the programming done. Earlier the algorithms like SVM gave great results but that also came with the cost of mathematics and consumed a lot of time. Now with the enhancement to technology, the Deep learning algorithms in machine learning like Artificial Neural Network, Convolutional Neural Network, and Recurrent Neural Network gives an outstanding performance. The character recognition framework utilizing deep neural networks (DNNs) for different scripts have been created with practically ideal outcomes.

II. LITERATURE REVIEW

Hand Digits Recognition turns out to be progressively significant in the advanced world because of its actual implementation in our every day life [1]. Recently , various recognition frameworks have been presented inside numerous applications where higher order effectiveness is required. It causes us to take care of increasingly complex issues and is simpler. [2] Programmed preparing of bank checks, the postal location is a general utilization of hand-written digit recognition [3]. In this particular paper, we prepared both Artificial neural network and Convolutional neural network model to recognize written by hand digits from 0 to 9. A node in a neural system can be comprehended as a neuron in the brain. Every node is associated with different nodes through weights (which are basically the edges between the nodes) which are balanced in the algorithm. A value is determined for every node dependent on the feature and methods of previous node. This procedure is called forward propagation [6]. The

last output of the system is related with the objective output, at that point weights are changed according to the loss function to depicting whether the system is speculated effectively [7]. This procedure is called back propagation [8]. To include complexity and correctness in the neural network, the systems have different Layers. In the middle of a fully connected neural system, there are various layers that exist, in particular information, output and hidden layers. Suppose we have features x1, x2, x3….. xn. The edges from one node to the other node of the network have weights that play the most important role in both forward and backward propagation. In forward propagation, there are two types of operation that happens in the hidden layer with the feature and the weights being passed to the neuron or node. The sum of the product of feature and weights and then applying an activation function. Whenever we have a Neural Network which is very deep at that time you will understand there are many weights and bais parameters. In backward propagation we have to change the values of the previous epoch weights, this reduces the loss value. In a completely associated neural system nodes in each particular layer are associated with the nodes and the layers preceding and succeeding them[9].

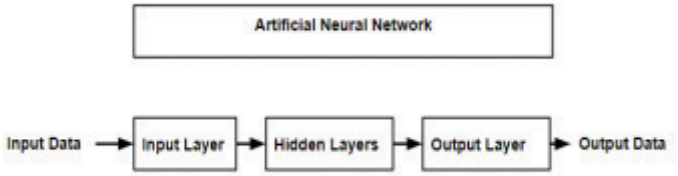

Fig. 1. Artificial Neural Network

In 1980-2000 researchers were not able to create a deep neural network in an Artificial neural network. The reason is the use of sigmoid function in every neuron (in 1980-2000 the ReLU was not invented). This is termed a vanishing gradient problem in a Neural network. The Activation function(sigmoid), when applied to the summation of the product of weights and the features, is always ranging between 0-1 and the derivative of the activation function ranges between 0-0.25 which gets smaller when the layers of the neural network become deeper. To deal with the vanishing gradient problem the use of ReLU or other activation function which does not lead to the collapse of the derivative is used.

$$w_{new} = w_{old} - \eta * \delta(Loss)/\delta(w_{old}) \quad (1)$$

Equation-1 Gradient Decent

When the weights assigned are large numbers then the expected number of the derivative $\delta(Loss)/\delta(w_{old})$ will be a very large number which will result in a large variation in the new and old values when backpropagating. Then new weight will jump on large values over the epochs and the weights will vary a lot with the value never converging at a point. So the weight initialization in a Neural network is a very crucial point otherwise this can lead to an Exploding Gradient problem[10].

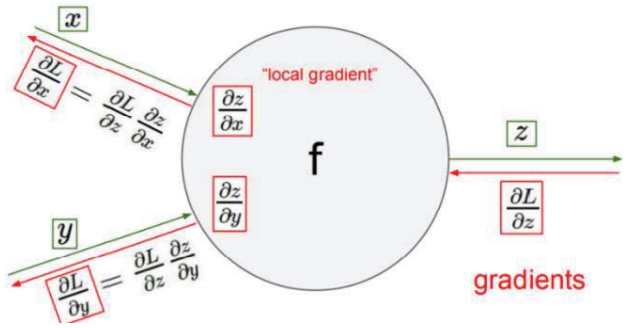

Fig. 2. Forward and Back propagation

Whenever we have a deep Neural network or a network with a huge number of layers then we have a huge number of weights and bias parameters as well which leads to overfitting the dataset problem or a particular data. In a multilevel Neural Network, underfitting will never happen because we will be having multiple levels that try to fit the training data perfectly. High variance is a problem with increasing levels in the network. We can apply regularization(L1 or L2) or Dropout layer to decrease the overfitting problem. In a Random forest multiple decision trees are created. Every Decision tree is created to its depth which also leads to an overfitting problem. Similarly, like the decision tree, we will be using a subset of features which is regularization which improves the accuracy of the whole model.

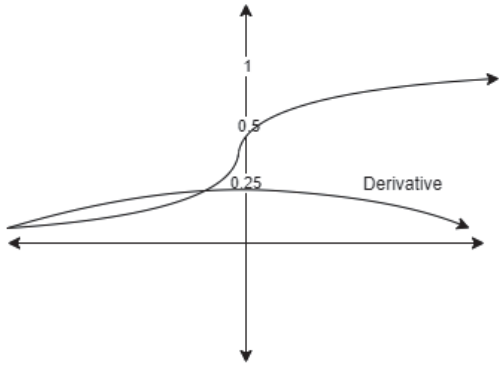

Fig. 3. Graph of derivative $\delta(Loss)/\delta(w_{old})$ for vanishing and exploding gradient problem

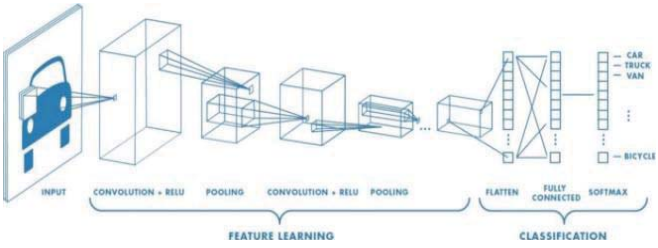

Fig. 4. CNN Model Architecture

In the Neural Network, we select a subset of features from the input layer and select a subset of hidden neurons. The other neurons which are not selected in the subset are deactivated [11]. The number of nodes in a subset count is calculated by the use of the dropout ratio. In image classification, object detection, and many other data augmentation Convolutional Neural Network(Convolutional neural network) plays a very major role. In the Convolutional neural network, the input data is in the form of a matrix which is having values in each cell ranging from 0-255 and either one or 3 artificial neural networks depending on grayscale and RGB scale respectively.[12]

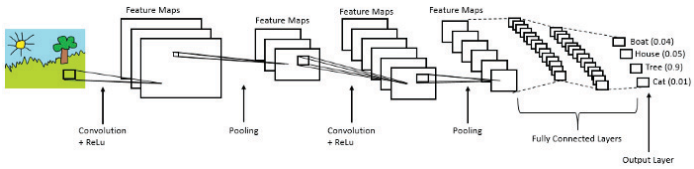

Fig. 5. Operations on an image using CNN model

The filters are applied to images and the output is also a matrix in a particular operation. The images go through a pipeline of operations of convolution layers with the filter, pooling, fully connected layer, and applying softmax function. The beneath figure is the complete architecture of a convolutional neural network to process an input picture and classify it based on values.[13]

III. SCRIPTS AND DATASETS

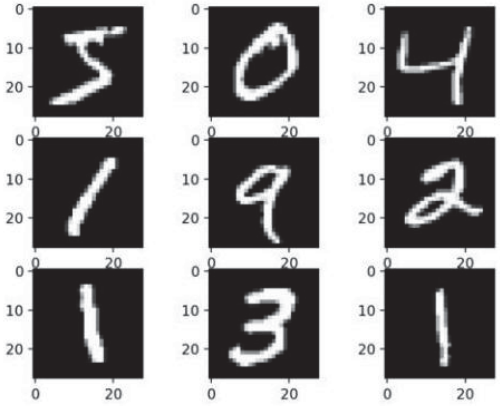

Fig. 6. MNIST Dataset

The MNIST is a great dataset for the handwritten digit classification problem. The MNIST dataset is a very authenticated and great dataset for the students and researchers. It has 60000 images with 10 classes (0-9) which is enormous in itself. Each image in the MNIST dataset is of 28 height and 28 weight which make the image of 784-dimensional vectors. The MNIST dataset is available easily on the internet. Each image in MNIST is a grey-scale image and the range is 0-255 which indicates the brightness and the darkness of that particular pixel. The MNIST dataset was created by the National Institute of Standards and Technology(NIST). To estimate the performance of a model, we split the preparation set into a training and testing dataset. Execution on the train and testing dataset would then be able to be plotted to give expectations to learn and adapt knowledge into how well a model is learning the issue.

IV. THE RECOGNITION MODEL

Optical character recognition(OCR) is a recognition system that has various stages. Each stage plays a very important role in the model. The stages are pipelined one after other Fig 7 shows the stages in the Recognition model.

A. Image Acquisition

Image acquisition is the first stage of all the recognition models. In this stage, the images are gathered, filtered, and cleaned before any preprocessing is done on the images.

B. Pre-Processing

Preprocessing is a very vital operation in the image. In Pre-processing major operations that are carried are image cleaning to reduce the noise in the image and removing the garbage. The optimization of the image is also done in this stage by filling the voids or holes, straightening curved lines. Different algorithms are also performed for skew correction. The output of this particular stage is a binary image which is done by binarization and texture filtering.

C. Segmentation

Decomposition of an image into sub-images is segmentation. Segmentation is of three types line, word, and character segmentation. When the input is an image with multiple lines breaking that image into a single line is line segmentation. When the input image is an image with a single line but multiple words and words have to be segmentation is word segmentation. Similarly, in character segmentation, the words are segmented into words.

D. Feature Extraction

Feature Extraction is a very important stage in the recognition model. It is a part of dimensionality reduction techniques. In Dimensionality reduction, the input data is converted into more simple and easy operation data. Large datasets like MNIST are great for this step as this particular stage optimizes the whole process of recognition. This stage removes the redundant data by retaining the originality of the dataset. In image processing, the feature extraction stage helps in edge detection and many other operations. Without the feature extraction stage, the classification of the image is a bit more complex and time-consuming. PCA and Image pixel vector are some techniques for feature extraction.

E. Classification

Classification is the decision-making stage in the pipeline of image recognition. The input to this stage is the output of the feature extraction stage. For classification nowadays many classifiers are present like Logistic regression, random forest, K nearest neighbors (KNN), Support vector machine (SVM) algorithm, Artificial Neural Network (Artificial neural network), Convolutional Neural Network (Convolutional neural network), and many more. For image classification, Deep neural network classifiers give great results i.e. Artificial Neural Network and Convolutional Neural Network. The MNIST dataset is huge and classifiers like Artificial Neural Network (Artificial neural network) and

Convolutional Neural Network (Convolutional neural network) give great accuracy on training 80% of the dataset and testing 20% of the dataset.

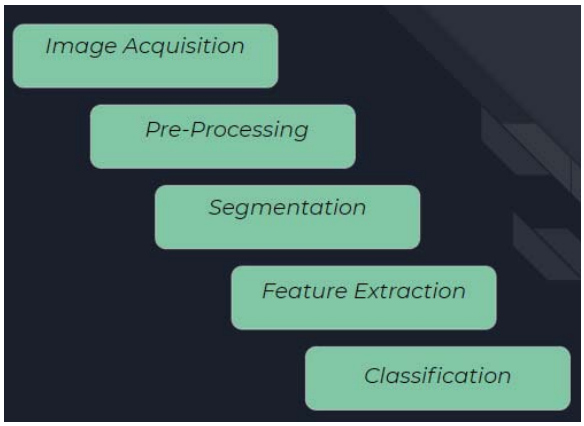

Fig. 7. Character Recognition Stages

V. CLASSIFIERS

A. Artificial Neural Network:

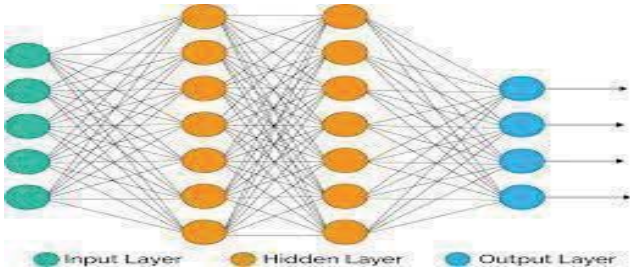

Fig. 8. ANN Network

The artificial neural network speaks to Artificial Neural Networks. It's a computational model. That relies upon structures and components of normal neural frameworks. Regardless of the way that, the structure of the Artificial neural network is impacted by a movement of information. Therefore, the neural framework changes relied upon data and output. It imitates the human cerebrum's capacity to recognize designs. Neural networks are fundamentally utilized for arrangement issues. The feed-forward technique got popular with the Artificial neural network model also called multilayer perceptron. The node or neuron in the network or the Artificial Neural Network model is connected with the edge which has their respective weights. The weights are the key to classifying the class of the given input. Figure 8 shows the network is broadly classified into three classes of input layers which take the input i.e. features from the fourth stage of the recognition model and then passes it onto the hidden layer which may have 1 or more layers and each layer depending upon the complexity of the dataset. The number of layers and neurons in each layer artificial neural network be too high or too low. Too large several layers and neurons can lead to an overfitting problem which can be handled using normalization or dropout. Backpropagation is a great technique by Jeffrey Hinton which opens a lot of doors in the field of deep learning. It lets us have a deep neural network. The Gradient descent technique is used to update the weight of the edges and bais when backpropagating.

B. Convolutional Neural Network

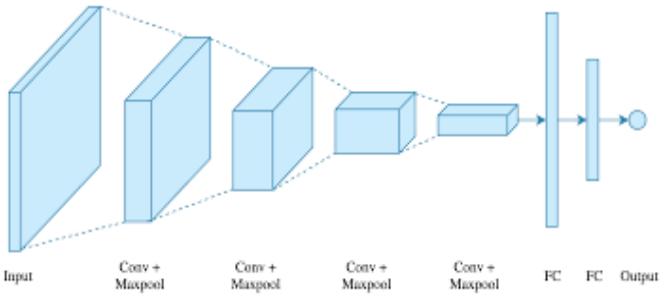

Fig. 9. CNN Dimensionality reduction in each stage

Convolutional neural networks are one of the best models in deep learning algorithms. Convolutional neural networks are spectacular for image classification in computer vision. The main components in Convolutional neural networks are feature extraction and classification. Deep learning algorithms are designed in a way that has similarities with the human brain. As we need several pictures to learn a particular object in the same way the Convolutional neural network is trained with the images. Unlike the human brain, their world only understands the numbers so to feed the Convolutional neural networks with those training images an image is represented as a 2D matrix (also pixels). Like an Artificial neural network, the Convolutional neural network doesn't have similar architecture. In the Convolutional neural network, the layers of the model have architecture such that they are in # dimensions i.e. width, height, and depth. Like an Artificial neural network in a Convolutional neural network in the convolution layer, all the layers are not connected with the ones in the next layer. The final result in the Convolutional neural network is a vector with dimensions that has a value of the probability scores. The first layer is the convolutional layer following the pooling layer and the last Fully-connected layer. In a Convolutional neural network the feature extraction is followed as a Feature map is produced with the use of two components the first one is a filter and the second is the kernel in the first layer. In the Convolutional layer, the method of convolution is done by multiplication at every cell and then summing up the result onto the feature map. To make our output nonlinear we use an activation function(ReLU, Sigmoid). Stride is the size of the step the convolution filter moves each time. To prevent the feature map from shrinking as its size tend to be smaller than the input we use padding. As the input in the Convolutional neural network is mostly large and the result is a vector that shows a need which dimensionality reduction. In a convolutional neural network, it is handled by adding a pooling layer in between the layers. The pooling layers is a great way to control overfilling in a Convolutional neural network model. There is two type of pooling - max, min. The classification stage of the Convolutional neural network model is handled by the Fully-Connected layer. The features being extracted by the convolution layer and pooling layer are passed on to the Fully-Connectedwhich performs classification. Fully-connected usually accepts 1-D data. The last layer of the Convolutional neural network is very similar to the Artificial

neural network. The last layer is strongly connected and is called a fully-connected layer.

VI. RESULTS

A. Implementation using Artificial Neural Network:

The Digit recognition of the MNIST dataset consists of 0-9 digits which act as classes in classification. The PyCharm IDE (Integrated Development Environment) has built-in developer tools and is a customizable and cross-platform IDE. PyCharm is used with the latest stable version of Python3.7. As discussed above in the recognition model we have 5 stages: the Data acquisition is already implemented as MNIST is a very reliable dataset. In the Image processing phase in the Artificial Neural Network to make all the images uniform for reducing the complexity of the dataset. The loading of the data is done by the python library Numpy which is a fundamental package for scientific computing in python). As mentioned earlier the Model of Artificial Neural Network has 3 layers Input, hidden, and output layer. The input to the next layer is the output of the previous one. In the Neural Network, the size of the input image is equal to the number of neurons in the input layer. In the dataset description, we mentioned it as 28x28 which is 784 pixels. The output of every layer is calculated with the help of the activation function which in our model is the ReLU activation function. The number of neurons in the hidden layer is kept the same as that of the input layer. The number of classes in the MNIST dataset is 0-9 which is 10 classes so the output layer consists of 10 neurons for 10 classes.

TABLE I. RESULTS OF ARTIFICIAL NEURAL NETWORK WITH LEARNING RATE-0.2 AND BAIS-0.5

epoch	Learn correct	Learn wrong	Test correct	Test wrong
1	0.96403	0.03592	0.95740	0.0426
2	0.96710	0.03290	0.96000	0.04000
3	0.97073	0.02927	0.96250	0.03750
4	0.97388	0.02612	0.96370	0.03630
5	0.97395	0.02605	0.96150	0.03850
6	0.97526	0.02473	0.96290	0.03710
7	0.97413	0.02587	0.96080	0.03920
8	0.97853	0.02147	0.96340	0.03660
9	0.97616	0.02383	0.96160	0.03840
10	0.9802	0.01980	0.96380	0.03620

We used the softmax activation function to get the probabilistic values of the output, which makes it easier to choose the maximum value from the given output of the classes. The epochs value is chosen as 10 with the batch size value of 200 for 60000 images being trained. For calculating the loss the categorical cross-entropy which is a logarithmic function is used and optimization is done by the ADAM i.e. Adaptive Moment Estimation algorithm for modifying the values of the weights and bias in the backpropagation. And the value of Baseline error was achieved as 1.31%.

B. Implementation using Convolutional Neural Network (Convolutional neural network or ConvNet):

The Digit recognition of the MNIST dataset consists of 0-9 digits which act as classes in classification. The PyCharm IDE (Integrated Development Environment) has built-in developer tools and is a customizable and cross-platform IDE. PyCharm is used with the latest stable version of Python3.7. As discussed above in the recognition model we have 5 stages: the Data acquisition is already implemented as MNIST is a very reliable dataset. The Convolutional Neural Network is Not as simple as the Artificial Neural Network to be trained. Like Artificial Neural Network had the number of neurons in the input layers as the number of pixels in the image(i.e. image size) here we have a 2-D matrix of the network.

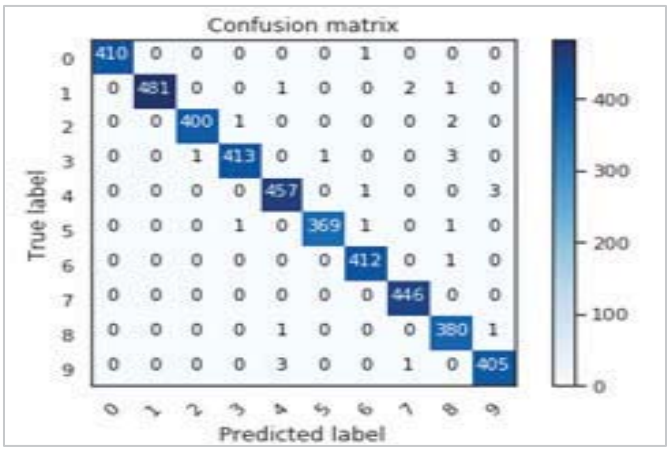

Fig. 10. Confusion matrix depicting true labels and predicted labels

As described earlier the convolutional neural network has 3 layers the convolutional layer the pooling layer and the fully-connected layer. The preprocessing part is done by the convolutional layer by applying numerous filters which enhances the image for the next layer for better segmentation and feature extraction. The input to this layer is in the form of a matrix with 3 dimensions are the height of the image, width of the image, and height of the image which can have binary values i.e 0 or 1. The size of the image here is passed as a parameter. These are hyper-parameters so usually we take the number of filters as 32,64 and so on and sizes of filters as 3x3, 5x5, etc. The deciding factor of the number of parameters being learned is by filter size and the number of filters and the model learns the value of the filter itself.

The next layer which handles the segmentation and the feature extraction is the MaxPooling2D. Each stride is taken and applied with the Max Pooling function to evaluate the max as the name suggests. This layer takes care of the overfitting problem in the model by regularization or Dropout

ratio. Here we used the dropout ratio of 20%(here the value of drop out ratio p) of some particular randomly selected neurons in the layer to reduce the overfitting problem (regularization is another way to avoid overfitting). The learning rate is minimal as it will help us get the global minimum. The preparation dataset is coordinated as a 3-dimensional exhibit of occurrence, picture width, and picture tallness. For a multi-layer perceptron model, we ought to decrease the photos down into a vector of pixels. For the present circumstance, the 28×28 measured pictures will be 784-pixel input esteems For this situation the 28×28 sized pictures will be 784-pixel input values.

TABLE II. RESULTS OF CONVOLUTIONAL NEURAL NETWORK WITH BASELINE ERROR OF 0.91%

Epoch	Loss	Acc	Val-loss	Val-acc
1/10	0.2754	0.9231	0.1339	0.9600
2/10	0.1089	0.9684	0.0935	0.9717
3/10	0.0710	0.9794	0.0866	0.9743
4/10	0.0496	0.9854	0.0732	0.9766
5/10	0.0358	0.9900	0.0634	0.9798
6/10	0.0257	0.9933	0.0597	0.9826
7/10	0.0192	0.9955	0.0626	0.9802
8/10	0.0142	0.9969	0.0612	0.9817
9/10	0.0107	0.9981	0.0573	0.9825
10/10	0.0081	0.9985	0.0558	0.9829

The discussion above in the classifiers section about Convolutional Neural Network that the dimensionality reduction is done the input layer was a 3D matrix and with Max Pooling layer it is converted to a 2D matrix and the final layer or the output layer is feed with the 1D matrix. To convert the 2D matrix to 1D form we use the Flatten function. The output layer is fed with 128 neurons after flattening and an activation function which in our case is the ReLU function. The output layer has 10 neurons for 10 classes with the softmax activation function for the probabilistic value of the output. The logarithmic capacity which is likewise called unmitigated cross-entropy in Keras is utilized as the misfortune capacity and ADAM (Adaptive Moment Estimation) advancement calculation is performed to get familiar with the various loads and their inclinations. For assessment of the model, the standard error is determined utilizing the precision metric on the test pictures. Furthermore, the estimation of Baseline error was accomplished as 0.91%. It is seen that the gauge mistake on account of convolutional neural organizations is not exactly the benchmark blunder in a convolutional neural network. The vital contrast in both is the number of layers and learnable boundaries. On account of a fake neural organization, all various layers are emphatically associated, and learnable boundaries are all the more; however execution is less, and in the convolutional neural organization, the underlying two layers for example. The convolution layer and max-pooling layer are handling the picture with the assistance of an artificial neural network pipelined by the unequivocally associated layers and several learnable boundaries are less yet the execution is better. Fit and evaluate the model can be a choice. The particular model fits more than 10 epochs which revive every 200 pictures. In the approval, dataset test data is used. Input to one line for each preparation epoch a verbose estimation of 2 is used.

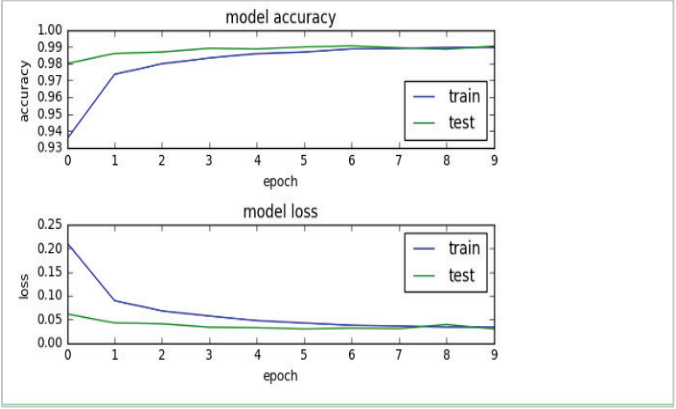

Fig. 11. Accuracy and Loss graph

VII. CONCLUSION

Convolutional Neural Network and Artificial Neural Network both are trained and tested with the MNIST dataset. Both the models were trained with an 80% dataset and 20% was tested. The models used the ReLU activation function for the backpropagation algorithm. And softmax activation function for the probabilistic values of the output. The training was done with the categorical cross-entropy loss. For further optimization, the ADAM optimizer was used for decreasing the loss. The critical point noticed during the training in hardware purpose was that the average error of Convolutional neural networks is less than artificial neural networks on the CPU. Convolutional Neural networks gave better performance for image classification. The average baseline error for the Artificial Network was 1.31% and for Convolutional Neural Network was 0.91%. This clearly shows the advantage of the Convolutional Neural Network over the Artificial Neural Network. The disadvantage of the Convolutional Neural Network over the Artificial Neural Network is that Convolutional Neural Network takes more time and CPU power. For reducing the time we can use GPU over a CPU for better performance. Since Convolutional neural networks could be successfully utilized in more impressive and cutting edge computers. This above-examined strategy could be probably the best element for all the bodies, working in the recognition of manually written digits. This leads us to believe that the Convolutional neural network's strategy is significantly better than other techniques.

ACKNOWLEDGEMENT

I want to pay extraordinary appreciation and warmth to my Research Work Guide, Dr. Akhtar Rasool, who allowed

me this opportunity to do this research paper. His crucial help made it possible to accomplish the objective. I might also want to thank different writers of reference materials referenced in the reference segment for their commendable research.

REFERENCE

[1] M .Nagu, N. V.Shankar, and K.Artificial neural network apurna, "A novel method for Handwritten Digit Recognition with Neural Networks," 2011.

[2] Y .LeCun, B .E .Boser, J .S .Denker, D .Henderson, R .E .Howard, W .E .Hubbard, et al., "Handwritten digit recognition with a backpropagation network," in Advances in neural information processing systems, 1990, pp .396-404 .

[3] A .Ashworth, Q .Vuong, B .Rossion, M .Tarr, Q .Vuong, M .Tarr, et al., "Object Recognition in Man, Monkey, and Machine," Visual Cognition, vol .5, pp .365-366, 2017 .

[4] J . Janai, F . Güney, A . Behl, and A . Geiger, "Computer Vision for Autonomous Vehicles: Problems, Datasets, and State-of-the-Art," arXiv preprint arXiv:1704.05519, 2017.

[5] K .Islam and R .Raj, "Real-Time (Vision-Based) Road Sign Recognition Using an Artificial Neural Network," Sensors, vol .17, p .853, 2017 .

[6] D .Arpit, Y.Zhou, B.Kota, and V.Govindaraju, "Normalization propagation: A parametric technique for removing internal covariate shift in deep networks," in International Conference on Machine Learning, 2016.

[7] I .Patel, V.Jagtap, and O.Kale, "A Survey on Feature Extraction Methods for Handwritten Digits Recognition," International Journal of Computer Applications, vol .107, 2014.

[8] I . H . Witten, E . Frank, M.A.Hall, and C . J . Pal, Data Mining: Practical machine learning tools and techniques: Morgan Kauf Artificial neural network, 2016.

[9] Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Neural Network Algorithm Md.Abu Bakr Siddique; Mohammad Mahmudur Rahman Khan; Rezoana Bente Arif; Zahidun Ashrafi,2018

[10] Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with VariousHidden Layers and Epochs using Convolutional Neural Network Rezoana Bente Arif; Md Abu Bakr Siddique; Mohammad.Mahmudur.Rahman… Khan.; Mahjabin Rahman. O,2019

[11] Beyond human. Recognition: A.Convolutional neural network-based framework for .handwritten character recognition Li Chen; Song.Wang; Wei Fan; Jun Sun; Satoshi Naoi,2020

[12] Dhanya Sudarsan; Shelbi Joseph, “A Novel Approach for Handwriting Recognition in Malayalam Manuscripts using Contour Detection and Convolutional Neural Nets”,2020.

[13] Nanehkaran, Y.A., Zhang, D., Salimi, S. et al. “Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits”. J Supercomput (2020).

[14] S. Oktaviani, C. A. Sari, E. Hari Rachmawanto and D. R. Ignatius Moses Setiadi, "Optical Character Recognition for Hangul Character using Artificial Neural Network," 2020 International Seminar on Application for Technology of Information and Communication (semantic), Semarang, Indonesia, 2020.

[15] R. Sharma, B. Kaushik, and N. Gandhi, "Character Recognition using Machine Learning and Deep Learning - A Survey," 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2020.

[16] P. Gupta, S. Deshmukh, S. Pandey, K. Tonge, V. Urkunde and S. Kide, "Convolutional Neural Network-based Handwritten Devanagari Character Recognition," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, 2020.

[17] P. Dhande and R. Kharat "Recognition of cursive English handwritten characters" 2017 International Conference on Trends in Electronics and Informatics (ICEI) pp. 199-203 2017.

[18] Shalini Puri and Satya Prakash Singh "An efficient Devanagari character classification in printed and handwritten documents using SVM" Procedia Computer Science vol. 152 pp. 111-121 2019.

[19] J. Schmidhuber "Deep learning in neural networks: an overview" Neural Networks vol. 61 pp. 85-117 2015.