Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 3): pp. 3432-3441
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.03.358

Hand Digit Recognition Using Cnn & Ann

[1]Upma Jain, [2]Vipashi Kansal, [3]Tanusha Mittal, [4] Ms Sonali Gupta

[1,2,3]Department of Computer Science and Engineering, Graphic Era Deemed to be University.

[4] Assistant Professor, Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun.

ABSTRACT

With the use of machine & deep learning algorithms, tasks ranging from object recognition in images to adding sound to silent films may now be completed with greater ease than ever before. Similar to this, Recognition of handwritten text is a key field for advancement and research with many different possible outcomes. The capability of a computer to accept and interpret understandable handwritten input from sources such as pictures, touch-screens, paper documents, etc. is known as handwriting recognition (HWR). Evidently, utilising MNIST datasets, artificial neural networks (MLP), convolution neural networks (CNN) we conducted handwritten digit recognition in this research. To find the most effective model for digit recognition, our major goal is to compare the training and validation accuracy and loss of the models mentioned above.

Keywords: Machine Learning, Deep Learning, Handwritten Digit Recognition, Convolution Neural Network (CNN), Artificial Neural Network (Multi-Layer Perceptron), and MNIST datasets, and

1. INTRODUCTION:

The capability of a computer to comprehend human handwritten number (digit) from a variety of sources, such as photographs, documents, contact sheets, and more, and classify them into ten specific categories (0-9) is known as handwritten digit recognition (HDR). Numerous uses for digit recognition include bank check processing, mail sorting, number plate identification, and more. Because handwritten digit recognition is not optical character recognition, there are numerous difficulties due to the wide variety of writing styles used by different cultures. This study focuses on Multi-layer perceptron model (MLP), and convolution neural networks (CNN), for hand digit recognition. The training accuracy, errors, and validation accuracy of these models are compared for the performance evaluation.

Convolutional neural networks (CNNs) have recently emerged as one of the most alluring methods, and they have played a crucial role in many recent successful and difficult machine learning applications, such as object detection [1], image segmentation [2], and face recognition [3]. So, for our difficult picture classification tasks, we select CNN. It can be used for high-level academic and commercial transactions such the recognition of handwriting digits. Handwriting digit recognition has a wide range of practical uses. In particular, it can be used in banks to read checks, post offices to sort mail, and many other relevant settings. Any model's correctness is crucial since more accurate models produce better results. Low precision models are unsuitable for use in practical situations. It is not ideal for the system to wrongly identify a digit because this could result in serious harm. In these real-world applications, a high accuracy algorithm is necessary. So that one can use the most promising algorithm with the lowest likelihood of errors. We are comparing several machine learning based methods based on their accuracy for HDR.

Next section describes the related work done by researchers in this field, IIIrd section describes the technique and implementation of the proposed methods. Section IV constitutes the results and discussion. Section V concludes the work.

2. RELATED WORK:

Machine learning, Deep learning, and artificial intelligence have all benefited from a significant amount of advancement related to the humanization of machines. The sophistication of machines is increasing over time; from performing simple math operations to performing retinal identification, they have improved the safety and control of human lives. Similar to facial recognition, handwritten text recognition is a crucial accomplishment of machine & deep learning that aids in spotting forgeries. A significant amount of research has already been conducted that includes a thorough analysis and application of several well-known algorithms, such as the works, Anuj Dutt [4], by S M Shamim [3], Hongkai Wang [8] and Norhidayu binti [5], to compare the various models of CNN algorithms on various datasets. [3] observed that the Multilayer Perceptron classifier produced results with the lowest error rate and the highest degree of accuracy, followed by Random Forest Algorithm, Support Vector Machine, Bayes Net, Random Tree, and Naive Bayes respectively. [4] compared SVM, KNN, CNN, and RFC and have been capable of acquiring the best accuracy of 98.72 percent by the use of CNN (which required the maximum execution time) and the lowest accuracy by the use of RFC. In order to classify handwritten text, [5] conducted a thorough study-comparison of KNN, SVM and MLP models. They observe that KNN and SVM predicted all of the dataset's classes efficaciously with 99.26 percent accuracy, however the procedure got a bit tricky with MLP when it struggled to classify number 9, for which they advocated using CNN with Keras to enhance the classification. [8] concentrated on contrasting deep learning approaches with machine learning approaches and contrasting their traits to determine which one is better for categorizing mediastinal lymph node. It was found that while categorizing mediastinal lymph node metastases of NSCLC, CNN finished about in addition to the first-class conventional techniques and human clinicians. The imported diagnostic features, that have been proved to be more reliable compared to the features for categorizing small-sized lymph nodes, are not used by CNN. As a consequence, integrating the diagnostic characteristics into CNN is a promising line of inquiry for the future. The concept of a convolution is "looking at functions nearby to produce a precise prediction of its outcome". Many researchers [6, 7, 8, 9, 10, 11] has done the HDR by using a CNN using MNIST datasets.

A neural network requires huge amount of data and information for training. A maximum accuracy of 99.2 percent was achieved in [6] by utilizing a 7-layered CNN model, while in [7], they mentioned the distinctive factors of CNN, its improvement from LeNet-5 to “Squeeze-and-Excitation network” (SENet), and comparisons with other models like, DenseNet, AlexNet and ResNet.

The accuracy and architecture rate of AlexNet are the same as LeNet-5 but much larger with around 4096000 parameters, and (SENet) have been named the champion of ILSVRC-2017 because they were able to decrease the error rate upto 2.25%.

3. PROPOSED METHOD:

This section constitutes the specifics details of Dataset and models which are used in this paper. Deep Learning has become a key tool for issues, like interpreting visuals, human voices, and robots investigating the environment. The suggested approach aims to comprehend CNN and apply it to the HDR system.

I. **DATASET:** In-depth implementation strategies, such as significant learning datasets, well-known algorithms, features scaling, and feature extraction techniques, are currently available for the vast study field of handwritten character recognition. NIST dataset is the superset of the Modified National Institute of Standards and Technology database (MNIST dataset, which is made up of Special Database 1 and Special Database 3 from NIST. The numbers in Special Databases 1 and 3 were typed in by high school students and Census Bureau staff members, respectively. MNIST has 70,000 handwritten digit pictures, each with a bounding box of 28x28 pixels with anti-aliasing. Each of these photos corresponds to a Y value that indicates what the digit is.

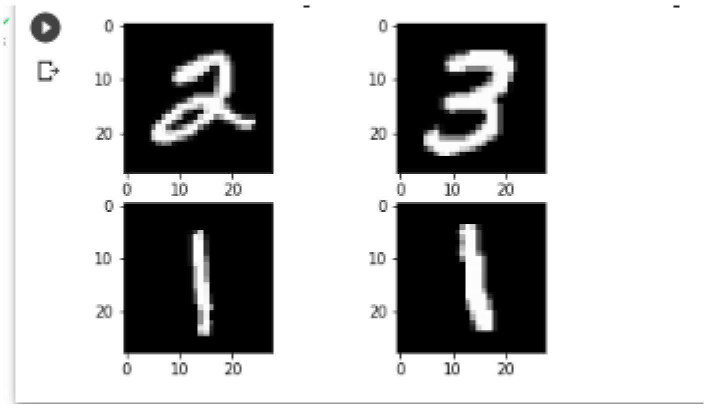
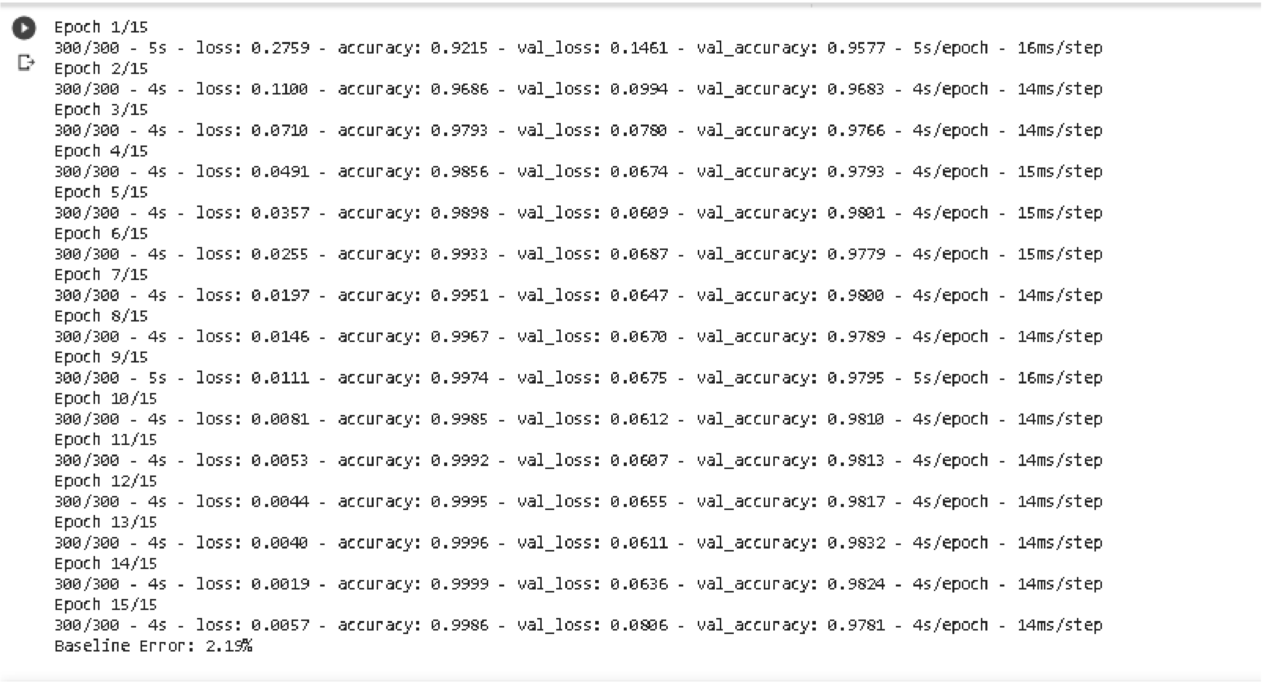


Fig 1. Showing a few arbitrary MNIST Handwritten digits.

II. **Multi-layer Perceptron (Base Model):** This model consists three different types of layers: an input, output, and a hidden layer. The input layer receives the signal for processing. The required tasks, such as prediction and classification are completed by the output layer. The principle computational mechanism of the MLP, constitutes an arbitrary variety of hidden layers between the input and output layers. Just like a feed-forward network, data moves ahead from an MLP's input layer to its output layer. The MLP's neurons are skilled using the back propagation learning algorithm.

- **Details of the model:**

The model is a straightforward neural network including a single hidden layer and exactly as many neurons as inputs (784) in input layer. The neurons in the layer are activated via a rectifier. A softmax activation feature is used in the output layer to transform the outputs into probabilistic values, allowing the model to select one category (class) out of 10 as its output prediction. The loss function, categorical cross entropy in Keras, is based on logarithmic loss, and the weights are learned using the effective ADAM gradient descent algorithm. The model is fitted and assessed. With updates every 200 photos, the model is fitted across 15 epochs. It is possible to see the model's skill as it trains because the test data is utilized as the validation dataset. The output is condensed to a single line for each training period using a verbose parameter of 2. The model is then assessed using the test dataset, and the classification error rate is displayed in figure 2.



```
Epoch 1/15
300/300 - 5s - loss: 0.2759 - accuracy: 0.9215 - val_loss: 0.1461 - val_accuracy: 0.9577 - 5s/epoch - 16ms/step
Epoch 2/15
300/300 - 4s - loss: 0.1100 - accuracy: 0.9686 - val_loss: 0.0994 - val_accuracy: 0.9683 - 4s/epoch - 14ms/step
Epoch 3/15
300/300 - 4s - loss: 0.0710 - accuracy: 0.9793 - val_loss: 0.0780 - val_accuracy: 0.9766 - 4s/epoch - 14ms/step
Epoch 4/15
300/300 - 4s - loss: 0.0491 - accuracy: 0.9856 - val_loss: 0.0674 - val_accuracy: 0.9793 - 4s/epoch - 15ms/step
Epoch 5/15
300/300 - 4s - loss: 0.0357 - accuracy: 0.9898 - val_loss: 0.0609 - val_accuracy: 0.9801 - 4s/epoch - 15ms/step
Epoch 6/15
300/300 - 4s - loss: 0.0255 - accuracy: 0.9933 - val_loss: 0.0687 - val_accuracy: 0.9779 - 4s/epoch - 15ms/step
Epoch 7/15
300/300 - 4s - loss: 0.0197 - accuracy: 0.9951 - val_loss: 0.0647 - val_accuracy: 0.9800 - 4s/epoch - 14ms/step
Epoch 8/15
300/300 - 4s - loss: 0.0146 - accuracy: 0.9967 - val_loss: 0.0670 - val_accuracy: 0.9789 - 4s/epoch - 14ms/step
Epoch 9/15
300/300 - 5s - loss: 0.0111 - accuracy: 0.9974 - val_loss: 0.0675 - val_accuracy: 0.9795 - 5s/epoch - 16ms/step
Epoch 10/15
300/300 - 4s - loss: 0.0081 - accuracy: 0.9985 - val_loss: 0.0612 - val_accuracy: 0.9810 - 4s/epoch - 14ms/step
Epoch 11/15
300/300 - 4s - loss: 0.0053 - accuracy: 0.9992 - val_loss: 0.0607 - val_accuracy: 0.9813 - 4s/epoch - 14ms/step
Epoch 12/15
300/300 - 4s - loss: 0.0044 - accuracy: 0.9995 - val_loss: 0.0655 - val_accuracy: 0.9817 - 4s/epoch - 14ms/step
Epoch 13/15
300/300 - 4s - loss: 0.0040 - accuracy: 0.9996 - val_loss: 0.0611 - val_accuracy: 0.9832 - 4s/epoch - 14ms/step
Epoch 14/15
300/300 - 4s - loss: 0.0019 - accuracy: 0.9999 - val_loss: 0.0636 - val_accuracy: 0.9824 - 4s/epoch - 14ms/step
Epoch 15/15
300/300 - 4s - loss: 0.0057 - accuracy: 0.9986 - val_loss: 0.0806 - val_accuracy: 0.9781 - 4s/epoch - 14ms/step
Baseline Error: 2.19%
```

Fig. 2: Epoch of Multilayer Perceptron Model

III. Convolutional Neural Network:

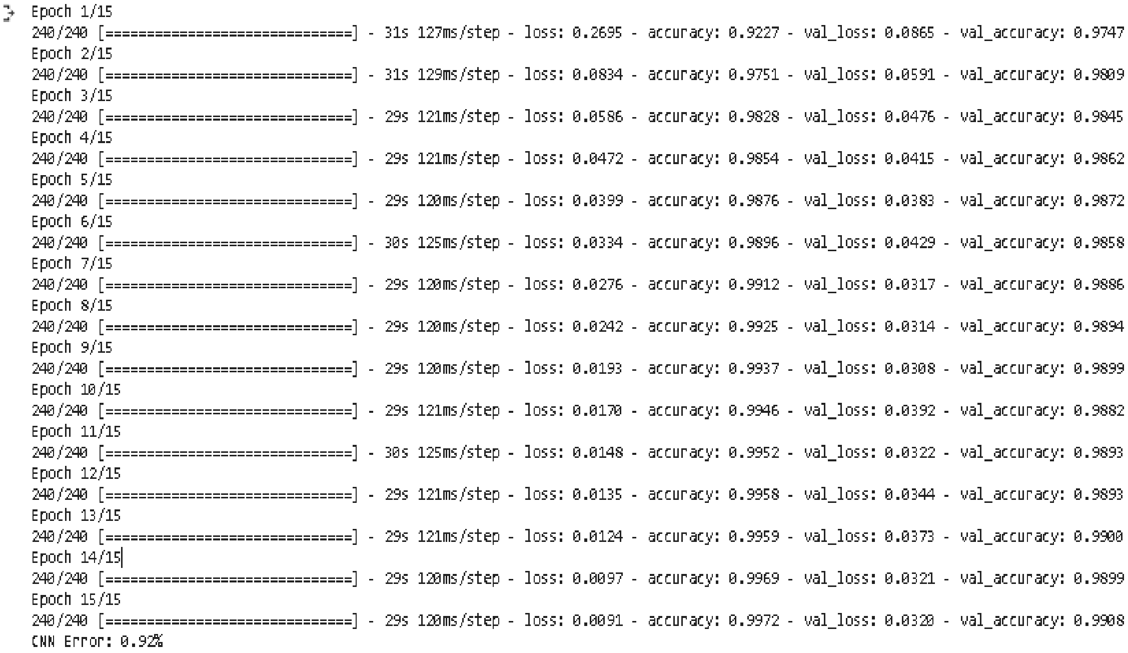
When using a neural network to learn deep learning, one quickly recognizes how crucial CNNs are, for image classification. A specific category of multi-layer neural network known as convolutional neural network is made to discover visual patterns straight from images with little to no preparation. The same essential design principles are followed by almost all CNN designs, including applying convolutional layers to the input one at a time, occasionally increasing the number of feature and downsampling the spatial dimensions (using Max pooling). In addition, activation functions, completely connected layers, and loss functions exist (e.g., softmax or cross entropy). Pooling layers, convolutional layers, and fully linked layers, however, are the most crucial CNN operations. So, before outlining our suggested model, let's quickly introduce those layers.

- **Convolution layer:** The convolutional layer is the first that extract features from an image. Convolution enables us to maintain the relationship between various components of a picture due to the fact pixels are only associated with their immediate neighbors and close neighbors. Convolution is a way for reducing the scale of an image without sacrificing the relationship between its pixels.

- **Pooling Layer:** In order to condense the spatial size of the feature maps in a CNN, pooling layers are frequently added after each convolution layer. The overfitting issue is also helped by pooling layers. By choosing the highest, common, or total values inside these pixels, a pooling length is selected to reduce the wide variety of parameters. Max Pooling that can be demonstrated as follows:

The term "max pooling" refers to a pooling technique that chooses the biggest element from the feature map region that the filter covers. Therefore, a feature map with the most major function from the prior function map will be the result after the max-pooling layer. We believe that it is now appropriate to offer an overview of our suggested CNN. Although it shares similarities with existing HDR architectures [1,6,8,10,11], improvements in performance have been made to a variety of filters, neurons, and activation functions. Two different models of CNN named Simple CNN and Complex CNN are proposed and analysed.

a. **Detailed architecture of Simple CNN:**
   - A convolutional layer known as Convolution2D is the first hidden layer. The layer includes 32 feature maps with a rectifier activation function, each measuring 5 by 5, in size. This layer serves as the input and anticipates photos with a structural outline that is higher than.
   - Subsequently, a pooling layer named MaxPooling2D is defined that accepts the maximum. It is set up with a pool that is 2 by 2.
   - The third layer, dubbed Dropout, is an integrated layer which uses dropout. To prevent overfitting, it arbitrarily excludes 20% of the layer's neurons.
   - The fourth layer, called Flatten, transforms the data from the 2D matrix into a vector. It enables the processing of the output by typical fully connected layers.
   - The following layer has a rectifier activation function and 128 completely linked neurons.
   - The output layer, which has 10 neurons for every 10 classes and a softmax activation function, produces predictions that resemble probabilities for each class.



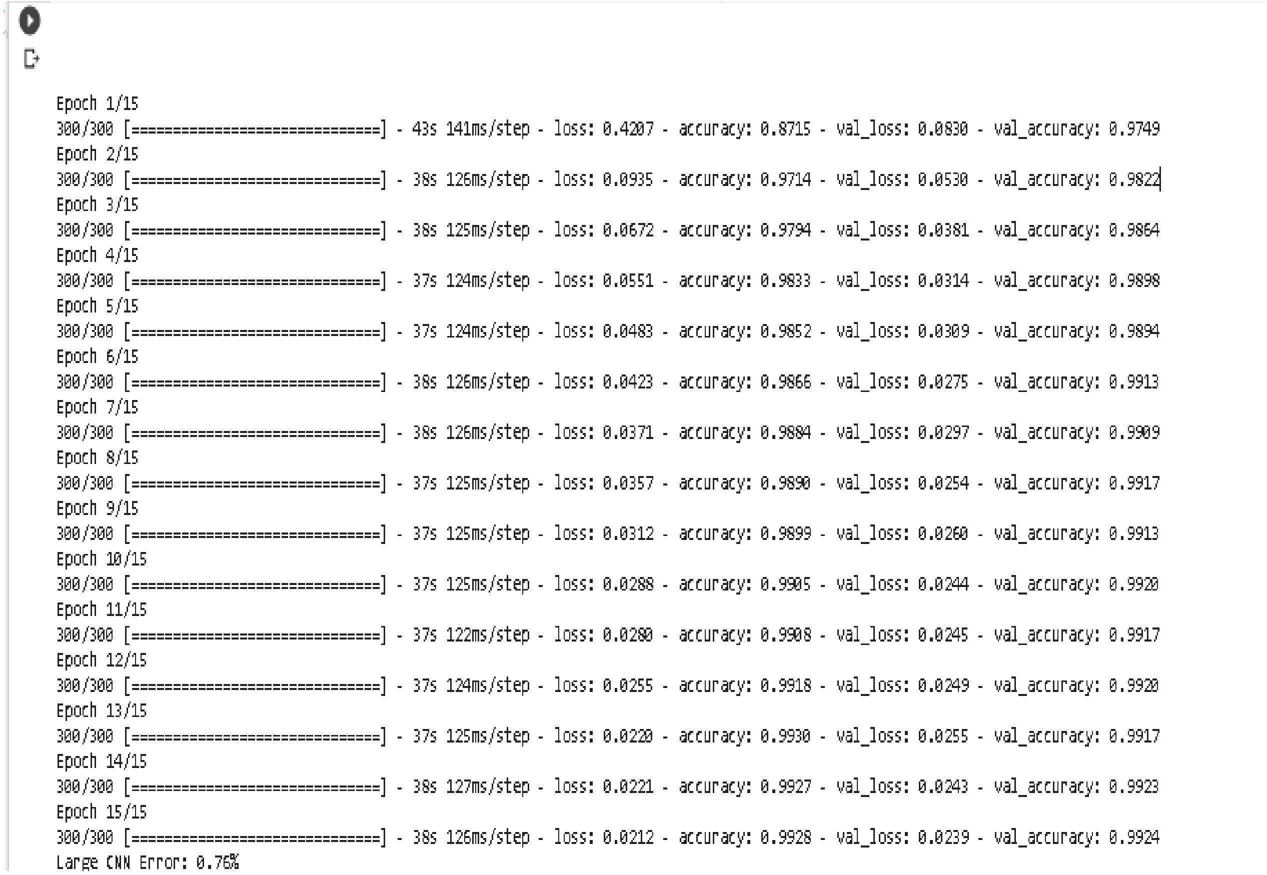
```
Epoch 1/15
240/240 [==============================] - 31s 127ms/step - loss: 0.2695 - accuracy: 0.9227 - val_loss: 0.0865 - val_accuracy: 0.9747
Epoch 2/15
240/240 [==============================] - 31s 129ms/step - loss: 0.0834 - accuracy: 0.9751 - val_loss: 0.0591 - val_accuracy: 0.9809
Epoch 3/15
240/240 [==============================] - 29s 121ms/step - loss: 0.0586 - accuracy: 0.9828 - val_loss: 0.0476 - val_accuracy: 0.9845
Epoch 4/15
240/240 [==============================] - 29s 121ms/step - loss: 0.0472 - accuracy: 0.9854 - val_loss: 0.0415 - val_accuracy: 0.9862
Epoch 5/15
240/240 [==============================] - 29s 120ms/step - loss: 0.0399 - accuracy: 0.9876 - val_loss: 0.0383 - val_accuracy: 0.9872
Epoch 6/15
240/240 [==============================] - 30s 125ms/step - loss: 0.0334 - accuracy: 0.9896 - val_loss: 0.0429 - val_accuracy: 0.9858
Epoch 7/15
240/240 [==============================] - 29s 120ms/step - loss: 0.0276 - accuracy: 0.9912 - val_loss: 0.0317 - val_accuracy: 0.9886
Epoch 8/15
240/240 [==============================] - 29s 120ms/step - loss: 0.0242 - accuracy: 0.9925 - val_loss: 0.0314 - val_accuracy: 0.9894
Epoch 9/15
240/240 [==============================] - 29s 120ms/step - loss: 0.0193 - accuracy: 0.9937 - val_loss: 0.0308 - val_accuracy: 0.9899
Epoch 10/15
240/240 [==============================] - 29s 121ms/step - loss: 0.0170 - accuracy: 0.9946 - val_loss: 0.0392 - val_accuracy: 0.9882
Epoch 11/15
240/240 [==============================] - 30s 125ms/step - loss: 0.0148 - accuracy: 0.9952 - val_loss: 0.0322 - val_accuracy: 0.9893
Epoch 12/15
240/240 [==============================] - 29s 121ms/step - loss: 0.0135 - accuracy: 0.9958 - val_loss: 0.0344 - val_accuracy: 0.9893
Epoch 13/15
240/240 [==============================] - 29s 121ms/step - loss: 0.0124 - accuracy: 0.9959 - val_loss: 0.0373 - val_accuracy: 0.9900
Epoch 14/15
240/240 [==============================] - 29s 120ms/step - loss: 0.0097 - accuracy: 0.9969 - val_loss: 0.0321 - val_accuracy: 0.9899
Epoch 15/15
240/240 [==============================] - 29s 120ms/step - loss: 0.0091 - accuracy: 0.9972 - val_loss: 0.0320 - val_accuracy: 0.9908
CNN Error: 0.92%
```

Fig 3: Epoch of Convolution Neural Network

b. Detail architecture of Complex CNN:

In this, a massive CNN architecture with more fully related, max pooling, and convolutional layers is defined. The following diagram summarizes the network topology.

1. A convolutional layer of thirty, 55 feature maps.
2. The pooling layer uses the maximum number of 2x2 patches.
3. A convolutional layer with 15, 3-by-3-inch feature maps.
4. The pooling layer uses a maximum of 2*2 patches.
5. A dropout layer with a 20 percent chance.
6. Level the layer.
7. Fully linked layer with activation of the rectifier and 128 neurons.
8. A layer with 50 neurons and activated rectifiers that is fully linked.
9. The output layer.



```
Epoch 1/15
300/300 [==============================] - 43s 141ms/step - loss: 0.4207 - accuracy: 0.8715 - val_loss: 0.0830 - val_accuracy: 0.9749
Epoch 2/15
300/300 [==============================] - 38s 126ms/step - loss: 0.0935 - accuracy: 0.9714 - val_loss: 0.0530 - val_accuracy: 0.9822
Epoch 3/15
300/300 [==============================] - 38s 125ms/step - loss: 0.0672 - accuracy: 0.9794 - val_loss: 0.0381 - val_accuracy: 0.9864
Epoch 4/15
300/300 [==============================] - 37s 124ms/step - loss: 0.0551 - accuracy: 0.9833 - val_loss: 0.0314 - val_accuracy: 0.9898
Epoch 5/15
300/300 [==============================] - 37s 124ms/step - loss: 0.0483 - accuracy: 0.9852 - val_loss: 0.0309 - val_accuracy: 0.9894
Epoch 6/15
300/300 [==============================] - 38s 126ms/step - loss: 0.0423 - accuracy: 0.9866 - val_loss: 0.0275 - val_accuracy: 0.9913
Epoch 7/15
300/300 [==============================] - 38s 126ms/step - loss: 0.0371 - accuracy: 0.9884 - val_loss: 0.0297 - val_accuracy: 0.9909
Epoch 8/15
300/300 [==============================] - 37s 125ms/step - loss: 0.0357 - accuracy: 0.9890 - val_loss: 0.0254 - val_accuracy: 0.9917
Epoch 9/15
300/300 [==============================] - 37s 125ms/step - loss: 0.0312 - accuracy: 0.9899 - val_loss: 0.0260 - val_accuracy: 0.9913
Epoch 10/15
300/300 [==============================] - 37s 125ms/step - loss: 0.0288 - accuracy: 0.9905 - val_loss: 0.0244 - val_accuracy: 0.9920
Epoch 11/15
300/300 [==============================] - 37s 122ms/step - loss: 0.0280 - accuracy: 0.9908 - val_loss: 0.0245 - val_accuracy: 0.9917
Epoch 12/15
300/300 [==============================] - 37s 124ms/step - loss: 0.0255 - accuracy: 0.9918 - val_loss: 0.0249 - val_accuracy: 0.9920
Epoch 13/15
300/300 [==============================] - 37s 125ms/step - loss: 0.0220 - accuracy: 0.9930 - val_loss: 0.0255 - val_accuracy: 0.9917
Epoch 14/15
300/300 [==============================] - 38s 127ms/step - loss: 0.0221 - accuracy: 0.9927 - val_loss: 0.0243 - val_accuracy: 0.9923
Epoch 15/15
300/300 [==============================] - 38s 126ms/step - loss: 0.0212 - accuracy: 0.9928 - val_loss: 0.0239 - val_accuracy: 0.9924
Large CNN Error: 0.76%
```

Fig. 4: Epoch of Complex Convolution Neural Network

4. RESULTS AND DISCUSSION:

We examined the accuracy and loss of MLP, Simple CNN and Complex CNN after implementing each method for clear comprehension. All of the models mentioned above have had their Training and validation Accuracy taken into consideration. After running all the three models, we discovered that Simple CNN achieves the highest training accuracy while Complex CNN achieves the maximum validation accuracy. Additionally, in order to better understand how the algorithms function, we have compared the loss. It can be observed from the table that the loss with Multi-layer perceptron is higher as compared to Simple CNN and Complex CNN. The loss with complex CNN is minimum and validation accuracy is maximum.

Table 1: Accuracy table of models

Model	Training Accuracy	Validation Accuracy	Loss
Multi-Layer Perceptron	99.66	97.81	2.19

Simple CNN	99.72	99.08	0.92
Complex CNN	99.28	99.24	0.76

5. CONCLUSION:

In this study, we developed three deep and machine learning-based models for handwritten digit recognition utilizing MNIST datasets. To determine which model was the most accurate, we analyzed them based on their individual properties. Simple CNN provides the highest training accuracy rate as this situation. In our research, we discovered that CNN produced the most precise outcomes for HDR as compared to MLP. This leads us to the conclusion that Complex CNN is the most effective solution for all types of prediction issues, including those using picture data. Future development of applications using deep and machine learning techniques is virtually unlimited. In future, we will concentrate on a hybrid algorithm with a wider range of data to find answers to a number of problems.

REFERENCES:

1. Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., & Ouyang, W. (2017). T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology, 28(10), 2896-2907.
2. Liu, F., Lin, G., & Shen, C. (2015). CRF learning with CNN features for image segmentation. Pattern Recognition, 48(10), 2983-2992.
3. Shamim, S. M., Miah, M. B. A., Angona Sarker, M. R., & Al Jobair, A. (2018). Handwritten digit recognition using machine learning algorithms. Global Journal of Computer Science and Technology.
4. Dutt, A., & Dutt, A. (2017). Handwritten digit recognition using deep learning. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 6(7), 990-997.
5. Siddique, F., Sakib, S., & Siddique, M. A. B. (2019, September). Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers. In 2019 5th International Conference on Advances in Electrical Engineering (ICAEE) (pp. 541-546). IEEE.
6. Sultana, F., Sufian, A., & Dutta, P. (2018, November). Advancements in image classification using convolutional neural network. In 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) (pp. 122-129). IEEE.
7. Wang, H., Zhou, Z., Li, Y., Chen, Z., Lu, P., Wang, W., & Yu, L. (2017). Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18F-FDG PET/CT images. EJNMMI research, 7(1), 1-11.

8. Nimisha Jain, Kumar Rahul, Ipshita Khamaru. AnishKumar Jha, Anupam Ghosh (2017). "Hand Written Digit Recognition using Convolutional Neural Network (CNN)", International Journal of Innovations & Advancement in Computer Science, IJIACS,ISSN 2347 – 8616,Volume 6, Issue 5.
9. Mishra, A., & Singh, D. (2017). Handwritten digit recognition using combined feature extraction technique and neural network. Computer Modelling & New Technologies, 21(2), 80-88.
10. Saeed, A. M. (2015). Intelligent handwritten digit recognition using artificial neural network. Int. Journal of Engineering Research and Applications, 5(5), 46-51.
11. Alwzwazy, H. A., Albehadili, H. M., Alwan, Y. S., & Islam, N. E. (2016). Handwritten digit recognition using convolutional neural networks. International Journal of Innovative Research in Computer and Communication Engineering, 4(2), 1101-1106.
12. Kussul, E., & Baidyk, T. (2004). Improved method of handwritten digit recognition tested on MNIST database. Image and Vision Computing, 22(12), 971-981.