

Improvement Analysis of Hand Written Digit Recognition using Convolutional Neural Network by Varying Values of CNN Parameters

Akanksha Gupta¹, Dr. Ravindra Pratap Narwaria² and Prof. Madhav Singh³

¹⁻³MITS/Department of Electronics Engineering, Gwalior, India

Email: akankshagupta318@gmail.com, {rpnarwaria, madhav.singh1}@mitsgwalior.in

Abstract—Hand written digit recognition is the demand of this era. Everybody sharing and storing documents in digital format. Hand written digital documents needs to be processed to make it understandable for everyone. Researchers are writing CNN based codes to achieve perfection in HDR (Handwritten digit recognition). Although CNN has achieved excellent perfection but still there is chance of improvements in terms of speed and complexity. In order make more improvements in the results of HDR, improvement analysis has been done on CNN parameters. There are many positive points which came out of this analysis, like the optimized value of epoch, minibatch and alpha. It is also seen that every handwritten number digit has better recognition at certain values of the parameters, it concludes that if those values are used while verifying the results then HDR final results can be improved in terms of speed and accuracy.

Index Terms— digit recognition (HDR), Deep learning, Convolutional Neural Networking (CNN), Artificial Neural Network (ANN), Modified National Institute of Standards and Technology (MNIST).

I. INTRODUCTION

The recently popular algorithm in machine learning is Convolutional Neural Networks (CNN), it has improved the results of many research fields by giving quality in the fields like object detection algorithms, human face recognition, image segmentation etc. this makes CNN to be introduced in the sectors like banks (for reading cheque), post office (for sorting posted letters area wise) and lots of other fields. Therefore CNN is now highly recommended classifier for any kind of HDR applications. It is used in academic systems for examination purpose as well as use in high value business transactions [3].

The major advantage of using the CNN model is that it exploits the topological information present in the input and is invariant to basic transformations like rotation, translation etc. On the other hand, MLP models never consider detailed topology information of input and are not suitable for complex problems. The organization of the rest of the paper is as follow [2]. English Handwriting datasets like Modified National Institute of Standards and Technology (MNIST) are used to calculate the performance of HDR technique [5].

In this paper, improvement analysis has been done on the parameters of CNN. It is found that HDR results can be improved in terms of speed and accuracy by varying CNN parameters.

II. CONVOLUTIONAL NEURAL NETWORKING (CNN)

The CNN classifier is very popular to perform HDR. CNN is a 7 layer convolutional network where there is one input layer then five layers which are hidden and at last one output layer as shown in Fig. 5. The size of input layer is 28 by 28 pixel that is 784 neurons can be sent to its input [1]. All the input images are grayscale in nature where intensity value vary from 0 to 255, here 0 represents black and 1 represents white.

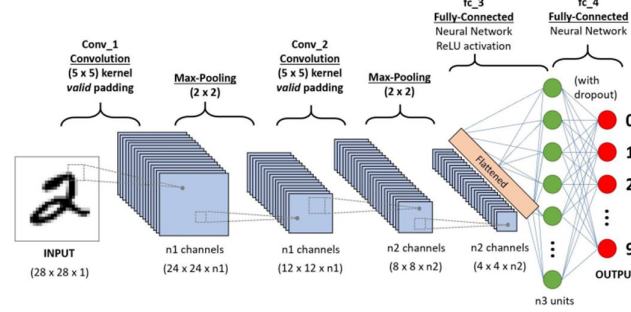


Fig. 1: CNN steps

Grayscale images with standard size of 28x28 pixels are used in CNN for training as well as testing purpose. Training input is represented by x . Here x is a vector with dimension 784 because the input image is 28x28 pixels. The output is represented by $y(x)$, here y is a vector of dimension 10. When this image is passed through the network, then this network does the calculations to find correct weights and biases. This comes to $y(x)$ for all training inputs x , in this way, the output is derived from these weight and bias values. The cost function is defined for calculation of performance of network as shown in equation 1 [1].

$$C(w, b) = \left(\frac{1}{2n}\right) \sum_{x=0}^n [y(x) - a]^2 \quad (1)$$

Where

w = cumulative weights in the network,

b = biases,

n = number of training inputs,

a = output.

Output is a function of x , w & b . here, $C(w, b)$ is non-negative because it's a combination of all non-negative terms. While sometimes $C(w, b) = 0$ when output $y(x)$ is almost equal to the actual output that is for all the values of training inputs n . The cost $C(w, b)$ can be reduced to a smaller degree in terms of weight and biases. For good results, cost must be smaller and for that it is required to find proper weights and bias. This can be achieved by using Stochastic Gradient Descent (SDG) algorithm. In SDG, effective solutions can be found by a small number of iteration. Also suitable solution can be achieved by a small number of iteration. The Stochastic Gradient Descent algorithm uses the below equations:

$$w^{new} = w^{old} - \frac{n}{m} \frac{dC_{xj}}{dw^{old}} \quad (2)$$

$$b^{new} = b^{old} - \frac{n}{m} \frac{dC_{xj}}{db^{old}} \quad (3)$$

The output can be represented as:

$$a = f(z) = f(wa + b) \quad (4)$$

CNN is kind of ANN with feed forward method. Connectivity in CNN is inspired by the organization of the animal visual cortex. CNN has many neurons that carry two parameters which are learnable weights & biases. Some input is provided to each neuron and it performs operations like dot product and performs it with non-linearity [6] [9].

A. Layers of CNN

There are many layers in CNN. Deep learning is a use of all these layers in iteration. The 3 category of CNN layers can be explained as:

- Input Layer: the raw pixel values of digital image are carried by input layer.
- Convolutional Layer: A result of input block neuron layer is supplied into convolution layer. There are many filters defines for this layer by user. The filter with window size of 5x5 is used for input pixels and it gives out the highest intensity pixels in output [4].
- Rectified Linear Unit Layer: the function of this layer to pass image pixel into activation function element wise. Back propagation is applied in CNN, this leads to change in values of image pixel, in order to rectify it ReLU function is applied.
- Pooling Layer: down sampling is performed on spatial dimensions like width and height by this layer. The output comes in the form of volume.
- Fully Connected Layer: computation of score of classes is performed at this layer. It computes maximum of score secured by input digits[8].

B. Stages for HDR

There are following stages to perform HDR using CNN. It has 4 stages as given below.

- Pre-Processing: in this stage, multiple operations are performed on input image. This stage is to make image ready for segmentation. The operations like noise filtering, standardization and image smoothening is performed on the input image. All the images like gray scale or colored image is first converted into binary image. This operation of binarization is performed to limit the quantity of data by performing threshold operation on it [10]. The images will look like as shown in Fig. 4 MNIST dataset.
- Segmentation: after the completion of pre-processing process, some sub-images are created from the original image data. Now this pre-processed image is divided into small sub images where each sub-image represents a single digit. All the singular digit images are resized into pixels. Edge detection method is applied to perform segmentation process [11].
- Feature Extraction: at the end of segmentation process, each output image is shown as a separate matrix where each value represents a single pixel. These matrix representing individual single digit is an easiest way of representation for further stages. This operation of representation each sub-image into the form of matrix is termed as feature extraction stage. This stage is used to remove the data redundancy from the image data.
- Classification & Recognition: in this stage, the formed matrix is taken as input to classifiers to recognize the digit present in the image. The output of extraction is input into classifiers like KNN, CNN, SVM. These classifiers used trained data to find the possible digit present in the image [7].

III. MNIST DATASET FOR HDR

The standard database used for HDR is MNIST database (Modified National Institute of Standards and Technology database). It contains images of handwritten digits which are very frequently used to train classifiers for image processing applications. Machine learning algorithms also use this data base very frequently. This database has training image database of 60,000 images and testing image database of 10,000 images. Some of them are shown in Fig. 2 below, where 50% of the images are taken from NIST's database. Many researches have been done on it to achieve good accuracy [12].

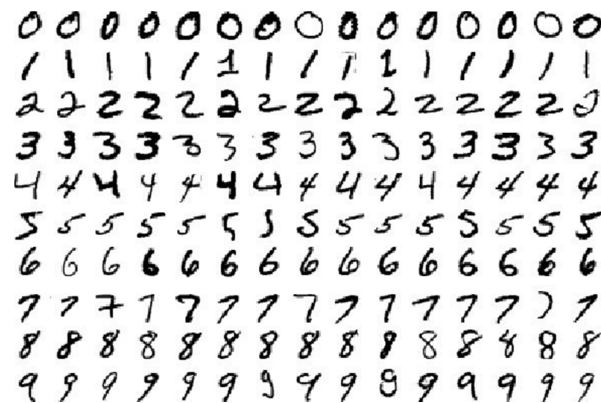


Fig. 2: some images of MNIST

IV. QUALITY PARAMETER

Accuracy can be defined as the closeness of the measured value to a standard or true value. In HDR, the number of true predictions and false predictions are counted and then formula shown in the equation 5 is used to calculate accuracy:

$$ACC = TP / (TP + FP) \quad (5)$$

Here

ACC = accuracy

TP = True predictions

FP = False predictions

V. EXPERIMENTAL RESULTS

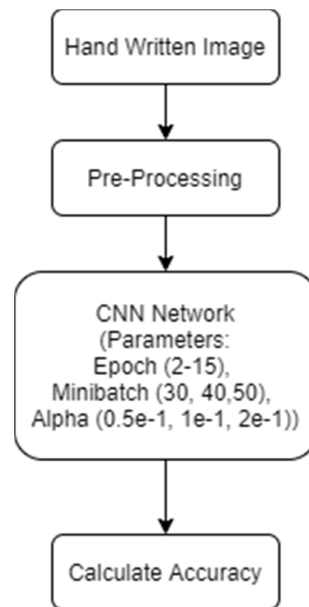


Fig. 3: Flowchart of Algorithm

Flowchart of working algorithm is shown in the Fig. 3 above. In order to detect the real time hand written number, we have written some number on the page as shown in Fig. 4(a), and then these numbers are required to be segmented individual image as shown in Fig. 4(b) and then converted into standard size of 28x28 pixels. This image of handwritten digit of size 28x28 pixels is used as an input to see the prediction.

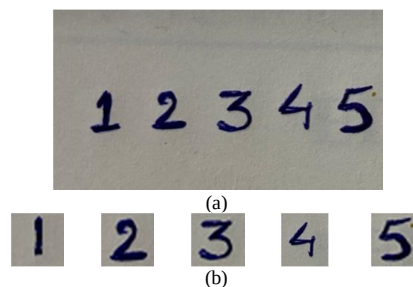


Fig. 4: (a) Hand written digit Image of number 1 to 5

(b) Segmentation of Hand written digit Images into individual image of number 1 to 5

This image of 28x28 is then browse into MATLAB using MATLAB codes as shown in the Fig. 5(a). then this image is first pre-processed this image as shown as a black & white image in the Fig. 5(b), then this black &

white image is processed by CNN to predict the digit in the image and result will be displayed on the MATLAB window as shown in the Fig. 5(b), here it is predicting the hand written digit of 1. In this way, any hand written digit can be predicted.

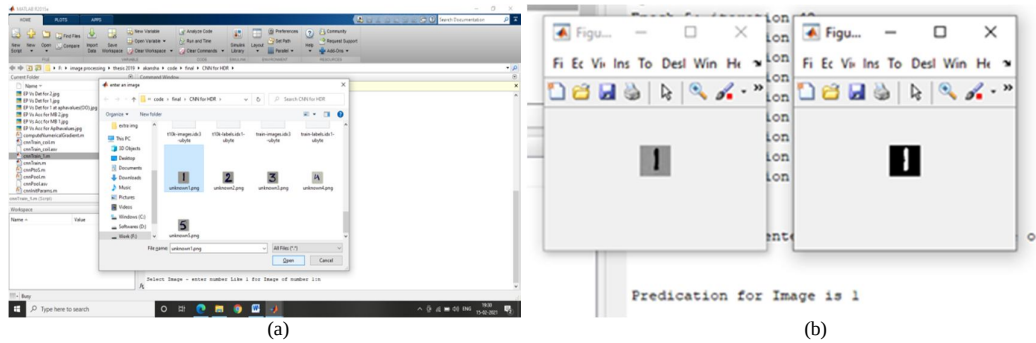


Fig. 5: (a) browsing image of hand written digit for real time prediction,(b) Prediction of handwritten image of digit 1

TABLE I: NUMBER OF EPOCH VS ACCURACY FOR HAND WRITTEN IMAGES OF DIGITS 1 TO 5 FROM MNIST DATABASE AT ALPHA = 1E-1, MINIBATCH = 50

Alpha = 1e-1, minibatch = 50						
Epoch	1	2	3	4	5	Accuracy
2	10	4	5	10	7	0.72
3	8	5	8	10	5	0.72
4	9	8	8	10	8	0.86
5	9	8	10	10	4	0.82
6	9	8	8	10	8	0.86
7	9	8	8	10	6	0.82
8	9	9	5	10	10	0.86
9	9	10	7	10	10	0.92
10	9	10	8	10	8	0.90
11	9	10	7	10	10	0.92
12	8	10	6	10	10	0.88
13	9	8	4	10	10	0.82
14	9	10	8	10	9	0.92
15	8	8	5	10	10	0.82

MNIST database is used for our experimentation, total 500 images of digits from 1 to 5 are used for training the CNN network and then 50 images of digits 1 to 5 are used for testing purpose. The accuracy is calculated by keeping CNN parameters Alpha = 1e-1, minibatch = 50 and by changing the values of epoch from 2 to 15, these values are tabulated in TABLE I.

TABLE II: NUMBER OF EPOCH VS ACCURACY FOR HAND WRITTEN IMAGES OF DIGITS 1 TO 5 FROM MNIST DATABASE AT ALPHA = 1E-1, MINIBATCH = 40

Alpha = 1e-1, minibatch = 40						
Epoch	1	2	3	4	5	Accuracy
2	9	8	8	10	4	0.78
3	9	7	2	10	9	0.74
4	9	5	6	10	9	0.78
5	9	10	7	10	9	0.90
6	9	10	3	10	10	0.84
7	8	10	8	10	9	0.90
8	8	10	8	10	9	0.82
9	9	10	7	10	10	0.92
10	9	10	6	9	10	0.88
11	9	10	6	10	9	0.88
12	9	10	9	10	10	0.96
13	9	8	8	10	9	0.88
14	9	7	7	10	10	0.86
15	9	10	8	10	10	0.94

Similarly the accuracy is calculated by keeping CNN parameters Alpha = 1e-1, minibatch = 40 and by changing the values of epoch from 2 to 15, these values are tabulated in TABLE II.

TABLE III: NUMBER OF EPOCH VS ACCURACY FOR HAND WRITTEN IMAGES OF DIGITS 1 TO 5 FROM MNIST DATABASE AT ALPHA = 1E-1, MINIBATCH = 30

Alpha = 1e-1, minibatch = 30						
Epoch	1	2	3	4	5	Accuracy
2	9	9	3	10	9	0.80
3	9	10	6	10	0	0.88
4	8	8	9	10	10	0.90
5	9	10	9	10	9	0.94
6	9	10	9	9	9	0.92
7	9	10	7	9	9	0.88
8	9	10	6	9	9	0.86
9	9	10	9	9	9	0.92
10	9	10	9	10	9	0.94
11	9	10	9	9	9	0.92
12	9	10	7	9	10	0.90
13	9	10	8	10	10	0.94
14	9	10	7	10	10	0.92
15	9	10	8	9	9	0.90

Similarly the accuracy is calculated by keeping CNN parameters Alpha = 1e-1, minibatch = 30 and by changing the values of epoch from 2 to 15, these values are tabulated in TABLE III.

TABLE IV: NUMBER OF EPOCH VS ACCURACY FOR HAND WRITTEN IMAGES OF DIGITS 1 TO 5 FROM MNIST DATABASE AT ALPHA = 0.5E-1, MINIBATCH = 50

Alpha = 0.5e-1, minibatch = 50						
Epoch	1	2	3	4	5	Accuracy
2	0	3	10	3	5	0.42
3	6	10	8	10	0	0.68
4	8	10	5	10	8	0.82
5	9	6	5	10	8	0.76
6	9	10	4	10	8	0.82
7	8	4	7	10	10	0.78
8	7	7	3	10	9	0.72
9	9	5	9	10	8	0.82
10	8	10	9	9	7	0.86
11	7	10	7	9	8	0.82
12	9	8	7	10	7	0.82
13	7	9	7	10	9	0.84
14	8	10	5	9	9	0.82
15	9	10	9	10	7	0.90

Similarly the accuracy is calculated by keeping CNN parameters Alpha = 0.5e-1, minibatch = 50 and by changing the values of epoch from 2 to 15, these values are tabulated in TABLE IV.

TABLE V: NUMBER OF EPOCH VS ACCURACY FOR HAND WRITTEN IMAGES OF DIGITS 1 TO 5 FROM MNIST DATABASE AT ALPHA = 2E-1, MINIBATCH = 50

Alpha = 2e-1, minibatch = 50						
Epoch	1	2	3	4	5	Accuracy
2	9	5	8	10	3	0.70
3	10	6	7	9	5	0.74
4	9	8	9	10	5	0.82
5	10	7	4	10	5	0.72
6	9	8	3	9	5	0.68
7	9	6	6	10	4	0.70
8	9	7	7	10	6	0.78
9	9	7	7	9	5	0.74
10	9	8	8	10	7	0.84
11	9	6	5	10	5	0.70
12	9	8	4	9	6	0.72

13	9	7	5	10	5	0.72
14	9	8	8	10	4	0.78
15	9	6	4	10	7	0.72

Similarly the accuracy is calculated by keeping CNN parameters Alpha = $2e-1$, minibatch = 40 and by changing the values of epoch from 2 to 15, these values are tabulated in TABLE V.

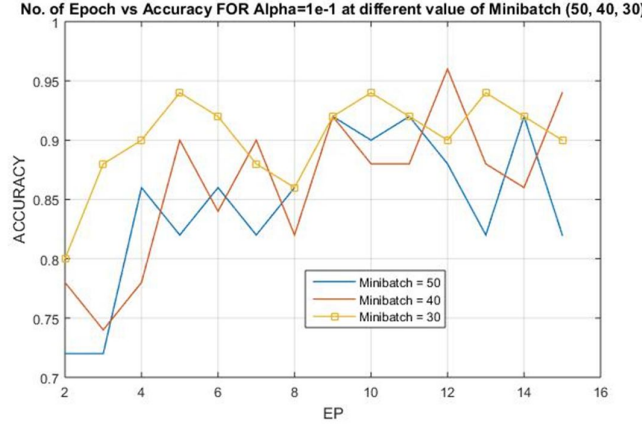


Fig. 6: Plot of Epoch Vs Accuracy for hand written images of digits 1 to 5 from MNIST database at Alpha = $1e-1$ and multiple values of minibatch (50, 40, 30)

A plot is created by using data from above tables from TABLE I TO TABLE III as shown in Fig. 6. This plot is created in between number of epoch and accuracy by varying the values of minibatch (50,40,30). It is found that minibatch values below 30 is not applicable for all case as size will be very less while minibatch values more than 50 shows a reduction in accuracy value. The plot shows that for lesser epoch, lesser minibatch value is giving better accuracy. But consistent and better results are achieved by using minibatch value of 30 while highest accuracy is achieved at minibatch value of 40. The conclusion of the plot is that minibatch 30 should be used for consistent results and for applications which required maximum accuracy, minibatch 40 can be used but if we have an application where we need better speed then minibatch 30 is better because it is giving maximum accuracy at lower epoch.

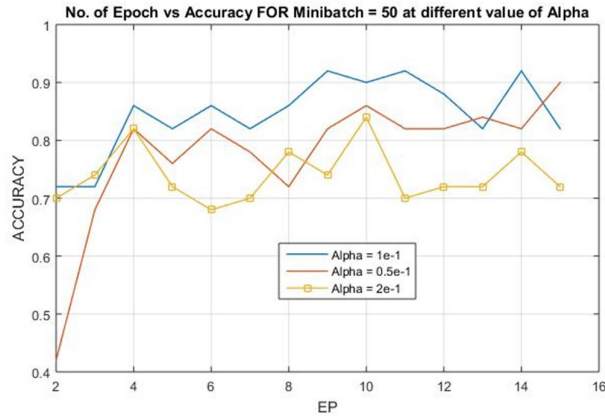


Fig. 7: Plot of Epoch Vs Accuracy for hand written images of digits 1 to 5 from MNIST database at minibatch = 50 and multiple values of Alpha ($1e-1$, $0.5e-1$, $2e-1$)

Another plot is created by using data from above tables from TABLE I, TABLE IV& TABLE V as shown in Fig. 7. This plot is created in between number of epoch and accuracy by varying the values of alpha ($1e-1$, $0.5e-1$, $2e-1$). It is found from the plot that values of accuracy are much better at alpha = $1e-1$, values more than $1e-1$ that is $2e-1$ or less than $1e-1$ that is $0.5e-1$ are providing lower accuracy. Even the highest accuracy is achieved at

$\alpha = 1e-1$. So it is concluded from this plot that in order to get the best results means better accuracy α value must be at $1e-1$ at any number of epoch.

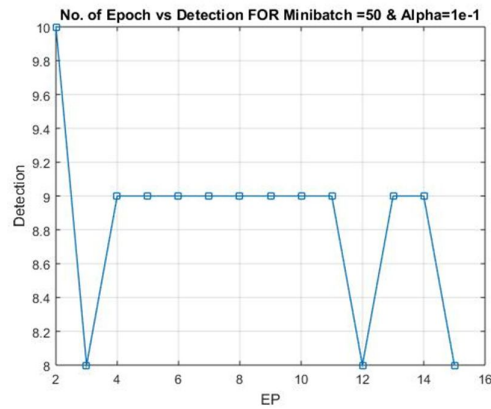


Fig. 8: Plot of Epoch Vs number of detection at $\alpha = 1e-1$ and minibatch = 50 for hand written images of digits 1

For the handwritten images of digit 1, a plot is created between number of epoch and number of true detection as shown in Fig. 8. It is found from the plot that the accuracy is almost consistent in detection of digit 1 at every value of epoch. So it can be concluded that digit 1 can be predicted accuracy irrespective of number of epoch.

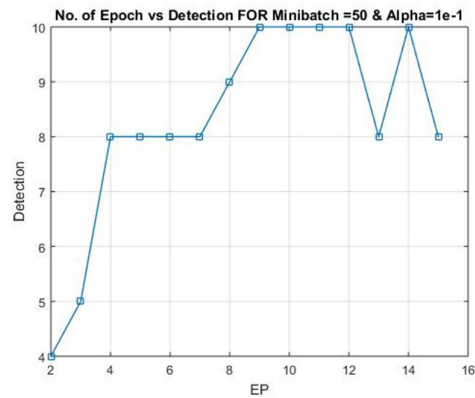


Fig. 9: Plot of Epoch Vs number of detection at $\alpha = 1e-1$ and minibatch = 50 for hand written images of digits 2

For the handwritten images of digit 2, a plot is created between number of epoch and number of true detection as shown in Fig. 9. It is found from the plot that higher value of epoch mainly more than 9 are giving better predication in case of digit 2. So it can be concluded that for detection digit 2, epoch value of 9 or more are better for correct prediction.

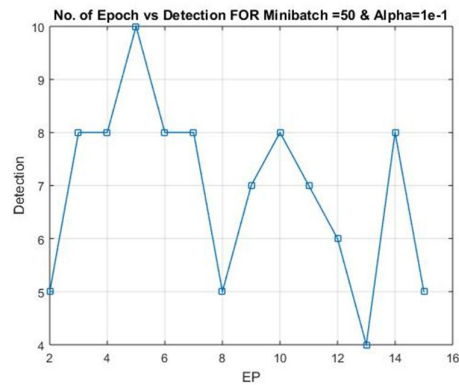


Fig. 10: Plot of Epoch Vs number of detection at $\alpha = 1e-1$ and minibatch = 50 for hand written images of digits 3

For the handwritten images of digit 3, a plot is created between number of epoch and number of true detection as shown in Fig. 10. It is found from the plot that lower value of epoch mainly 5 is giving better predication in case of digit 3. So it can be concluded that for detection digit 3, epoch value of 5 is better for correct prediction.

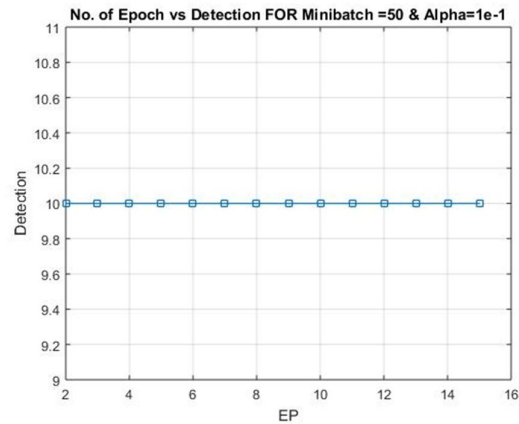


Fig. 11: Plot of Epoch Vs number of detection at Alpha = 1e-1 and minibatch = 50 for hand written images of digits 4

For the handwritten images of digit 4, a plot is created between number of epoch and number of true detection as shown in Fig. 11. It is found from the plot that the accuracy is almost consistent in detection of digit 4 at every value of epoch. So it can be concluded that digit 4 can be predicted accuracy irrespective of number of epoch.

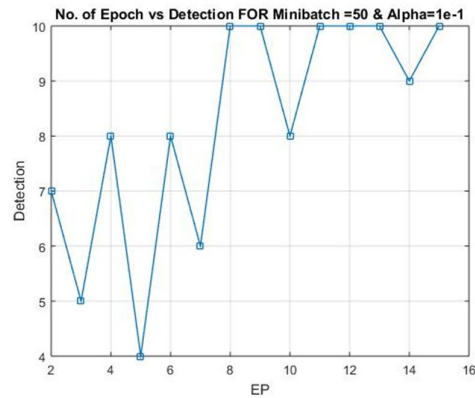


Fig. 12: Plot of Epoch Vs number of detection at Alpha = 1e-1 and minibatch = 50 for hand written images of digits 5

For the handwritten images of digit 5, a plot is created between number of epoch and number of true detection as shown in Fig. 12. It is found from the plot that higher value of epoch mainly more than 8 are giving better predication in case of digit 5 and even epoch value of 12 is better than any other value. So it can be concluded that for detection digit 5, epoch value of 11 is better for correct prediction.

TABLE VI. OPTIMIZED EPOCH VALUES FROM PROPOSED WORK

DIGITS	OPTIMIZED EPOCH VALUES
1	4
2	9
3	5
4	2
5	11

TABLE VII. COMPARISON OF TIME TAKEN BY CNN PREVIOUS WORK (FOR ALL DIGITS, EPOCH =15) AND CNN PROPOSED WORK (DIGIT 1=5 EPOCH, DIGIT 2=9 EPOCH, DIGIT 3=5 EPOCH, DIGIT 4=2 EPOCH, DIGIT 5=11 EPOCH)

Parameter	CNN previous work	CNN proposed work
Time (Training) approx.	35 min	23 min
Time (Testing) approx.	20 min	14 min

From the Table VII, it can be seen that when we use optimized values of epoch which are comes out of our analysis to get maximum accuracy then time taken by the training and testing is lesser as compare to the previous work result where value is taken as 15 for all digits. It is concluded from the table that for fast applications, in order to get maximum efficiency and minimum time optimized epoch values can be used as given in table VI.

VI. CONCLUSION

CNN classification is very popular in Handwritten digit recognition. The hand written image is first pre-processed into a black & white 28x28 image. MNIST database is used for training and testing images. Self-written images are also used to find the prediction in real time. Experimentation has been done on the parameters of CNN to improve the performance of the HDR. It is found that minibatchvalue equals to 30 should be used for consistent results and minibatch value equals to 40 for maximum accuracy but for faster applications,minibatch value equals to 30 is better.While studying alpha value, it is found that the best accuracy can be achieved at alpha value equals to 1e-1 at any number of epoch. Atlast each digit is studied for number of true detection against number of epoch. It is seen that digit 1& 4 can be predicted accuracy irrespective of number of epoch. For digit 2, epoch value of 9 or more are better for correct prediction. For digit 3, epoch value of 5 is better for correct prediction. For digit 5, epoch value of 11 is better for correct prediction. Since 2, 3 & 5 are little complex image pattern that's why from the results it is concluded that changing epoch value as per above analysisfor different digits can give better prediction, means in future work, after prediction a verification phase can be implemented with adaptive epoch value for better prediction.

REFERENCES

- [1] Mayank Singh, Rahul (2020). Handwritten Digit Recognition using Machine Learning. International Research Journal of Engineering and Technology, 7(7), 921-925.
- [2] FathmaSiddique, ShadmanSakib, Md. Abu BakrSiddique (2019). Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers. 5th International Conference on Advances in Electrical Engineering (ICAEE).
- [3] Mahmoud M. Abu Ghosh; Ashraf Y. Maghari(2017), "A Comparative Study on Handwriting Digit Recognition Using Neural Networks", IEEE,International Conference on Promising Electronic Technologies (ICPET), 16-17 Oct. 2017.
- [4] SavitaAhlwata, AmitChoudhary (2019). Hybrid CNN-SVM Classifier for Handwritten Digit Recognition. Elsevier B.V, International Conference on Computational Intelligence and Data Science, Procedia Computer Science 167, 2554-2560.
- [5] Md. Anwar Hossain& Md. Mohon Ali, (2019), "Recognition of Handwritten Digit using Convolutional Neural Network (CNN)", Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence, Volume 19, Issue 2.
- [6] Yellapragada SS Bharadwaj, Rajaram P, Sriram V.P, Sudhakar S, KollaBhanuPrakash (2020, April). Effective Handwritten Digit Recognition using Deep Convolution Neural Network. International Journal of Advanced Trends in Computer Science and Engineering, 9(2), 1335-1339.
- [7] Vijayalaxmi R Rudraswamimath, Bhavanishankar K (2019, June). Handwritten Digit Recognition using CNN. International Journal of Innovative Science and Research Technology, 4(6), 182-187.
- [8] AnujDutt, AashiDutt (2017, July). Handwritten Digit Recognition Using Deep Learning", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 6(7), 990-997.
- [9] M. M. A. Ghosh and A. Y. Maghari (2017). A Comparative Study on Handwriting Digit Recognition Using Neural Networks. International Conference on Promising Electronic Technologies.
- [10] Haider A. Alwzawy, Hayder M. Albehadili, Younes S. Alwan, Naz E. Islam (2016, February). Handwritten Digit Recognition Using Convolutional Neural Networks. International Journal of Innovative Research in Computer and Communication Engineering, 4(2), 1101-1106.
- [11] . R. Alhajj and A. Elnagar, "Multiagents to separating handwritten connected digits," in IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 35, no. 5, pp. 593-602, Sept. 2005.
- [12] Fabien Lauer, Ching Y. Suen, G´erard Bloch (2007). A trainable feature extractor for handwritten digit recognition. Journal Pattern Recognition, Elsevier, 40(6), 1816-1824.