# Handwritten Digits Recognition using Machine Learning Algorithms

**Harshit Mishra[1], Suraj Kumar Pathak[2], Aastik Srivastava[3]**

*[1]ABES Engineering College*
*[2]ABES Engineering College*
*[3]ABES Engineering College*
*[4]Mr. Manish Srivastava: Professor, Dept. of Information Technology, ABES Engineering College, Uttar Pradesh, India*

***

**Abstract -** Composed by hand Digit Recognition is working in the field of digit and model portrayal. This has an open issue here in the field. A couple of examinations have found that AI has a mind-boggling execution in data requests. The essential purpose of this assessment paper is to give various frameworks to the affirmation of physically composed request models. This assessment paper takes a gander at the presentation of three AI classifier models, specifically, K-Nearest Neighbor (K-NN), Support Vector Machine (SVM) and Random Forest with tendency assistance. These various methodologies give full results yield with improved computational profitability. In this paper, we are using the accompanying three techniques classifiers, for instance, K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), and Random Forest. After this, we get to the course of action of getting ready data which further forms the accuracy of the strategy with respected classifiers. We mark directly off the bat with K-NN classifiers; its results show up with a better than average display consequence of 96.60%. By then, we check with Random - Forest classifiers it in like manner shows the outcome with a conventional display consequence of 96.84% and better than past classifiers we used, for instance, K-NN one. What's more, a short time later, we check with classifiers SVM, and this classifier of this paper shows that SVM has correspondingly high exactness of 97.75 % appeared differently in relation to other people, in any case, for our circumstance SVM achieves a getting ready speed snappier than others. The examination presented in this paper suggests that the SVM got together with preprocessing systems is fit for achieving unimaginable execution isolated from various classifiers when used as a request count in separated physically composed digit affirmation.

***Key Words***: Handwritten digit recognition, K-Nearest Neighbour, Support Vector Machine, Random Forest Classifier, Machine Learning, MNIST etc

## 1. INTRODUCTION

The issue of interpreted numerals affirmation has been commonly focused starting late and the tremendous measure of preprocessing techniques and gathering counts have been made. (Wang and Yan, June 2000). In any case, physically composed numerals affirmation is up 'til now a test for us. The rule inconvenience of physically composed numerals affirmation is the certifiable contrast in size, translation, stroke thickness, turn, and contorting of the numeral picture considering the way that interpreted digits are made by different customers and their making style isn't exactly equivalent to one customer to another. Methods of the International Conference on Industrial Engineering and Operations Management Washington DC, USA, September 27-29, 2018. Composed by hand affirmation is the limit of a PC to get and grasp conceivable interpreted commitment from sources, for instance, paper chronicles, customer input contact screens and various contraptions. The image of the created substance may be distinguished from a touch of paper by optical inspecting (optical character affirmation) or savvy word affirmation or by customer input. On the other hand, the advancements of the pen tip may be identified "on line", for example by a pen-based PC screen surface, an all things considered more straightforward endeavor as there are more snippets of data available This paper presents seeing the translated digits (0 to 9) from the prestigious MNIST dataset using TensorFlow framework(library) and python as a language and its libraries as the customer enters the individual digit the machine would see and show the results with exactness rate.

### 1.1 Problem with handwritten digits

The physically composed digits are not for the most part of the proportional size, domains. For example, digit affirmation has been used ordinarily by the mail station for the explanations behind portraying road numbers using AI. Different people have entirely unexpected forming styles; even digits of a comparative individual written in different events are not unclear. How mans caused mental aptitude to deal with the boundless possibility of different conditions of digits, given only an image. Width , heading and upheld to edges as they yield from making out of individual to individual, so the a general issue would describe the digits due to the comparability between digits, for instance, 1 and 7,5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. This issue is stood up to more when various people create a single digit with a grouping of different handwriting styles.

## 1.2 Various type of handwritten digit

Interpreted digit affirmation accepts a significant activity in various customer approval applications in a propelled world. The composed by hand digits are not of comparable size, thickness, style and course, thus, these troubles are to be digitized to decide this issue. The composed by hand digits are not for the most part of comparable size, width, course, and bolstered to edges as they surrender from creating from individual to individual, so the general issue would orchestrate the digits on account of the closeness between digits.

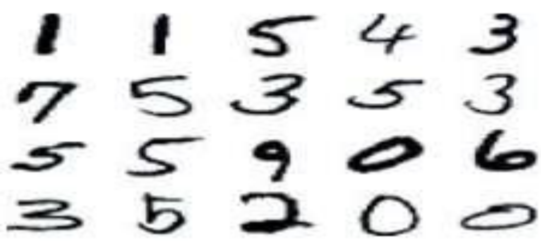

**Fig -1**: Handwritten digits

## 2. LITERATURE SURVEY

In 1959, Grimsdale put forth an attempt in the territory of character acknowledgment. Later in 1968 Eden recommended a methodology named as examination-by-amalgamation technique to carry on the exploration work. Eden demonstrated that every manually written character has some schematic highlights. Parveen Kumar, Nitin Sharma and Arun Rana [8] made an endeavor to perceive a transcribed character utilizing SVM classifier and MLP Neural Network. Diverse portion based SVM like the direct piece, polynomial part, and quadratic bit based SVM classifiers are utilized. In the SVM classifier model, there are two periods of preparing and testing. From each character, around 25 highlights are separated with the assistance of which SVM is prepared. Among the three pieces utilized the straight part gives a precision of 94.8%. Reena Bajaj, Lipika Dey and Santanu Chaudhury [7] in 2002 chose to introduce an all around fabricated and deliberate approach for transcribed numeral acknowledgment. In this paper, connectionist systems are utilized for building acknowledgment engineering. Patrice Y. Simard, Dave Steinkraus, John C. Platt [1] portrayed that the convolutional neural systems are better for visual record investigation like penmanship acknowledgment tasks. The fundamental basic practices for this assignment are the development of the informational collections utilizing versatile bends and the utilization of convolutional neural systems.

T. Siva Ajay [2] likewise suggested that the higher pace of exactness in manually written digit acknowledgment assignment can be accomplished by the utilization of convolutional neural systems. The usage of CNN is made simple and basic by the utilization of LeNet building. Because of this exactness more prominent than 98% is gotten in this paper. Ming Wu and Zhen Zhang [3] in 2010 made an examination between various classifiers to close which gives better execution in the acknowledgment task. The examination is done between the six classifiers specifically LDA (Linear Discriminant Analysis), GMM (Gaussian Mixture Models), QDA(Quadratic Discriminant Analysis), SVML (SVM with direct part work), SVMR (SVM with spiral premise portion capacity) and k-NN. Out of all the classifiers,k-NN (k=3) gives the most minimal blunder rate. Haider A Alwzwazy, Haider M Albehadili, Younes S.Alwan, and NazE. Islam [4] began a difficult errand of acknowledgment assignment of Arabic written by hand digits. For this, they chose to carry on the exploration utilizing the Deep Convolutional Neural Networks. The exactness of 95.7% is accomplished because of this work. In 2015, Saeed AL Mansoori [5] applied the Multi-Layer Perceptron model to distinguish manually written digits. The examples from the informational index are prepared by utilizing inclination drop backpropagation calculation and later feedforward calculation. From the obtained outcomes it tends to be seen that the digit 5 has the most elevated exactness of 99.8% while digit 2 has the least precision of 99.04%. What's more, the proposed framework accomplished a general exactness of 99.32%. Shobhit Srivastava, Sanjana Kalani, Umme Hani and Sayak Chakraborty [6] represented the manually written acknowledgment utilizing Template Matching, Support Vector Machine and Artificial Neural Networks. Among the techniques utilized, Artificial Neural Networks ended up giving progressively precise outcomes. Shashank Mishra, D.Malathi and K.Senthil Kumar [7] endeavored the manually written acknowledgment utilizing Deep Learning. They utilized Convolutional Neural Network because of which they inferred that exactness is expanded and there is a decrease in the calculation time. The exactness of 99.2% is acquired.

## 3. IMPLEMENTATION

### 3.1 SUPPORT VECTOR MACHINE (SVM)

Classifier for digit acknowledgment Support Vector Machine is a regulated AI method that is applied for characterization and relapse. It is only the portrayal of the information focused in the space and mapped in this manner ordering them into classes. The SVM arranges or isolates the classes utilizing the hyper-plane idea. The division edge ought to be equidistant from the classes.

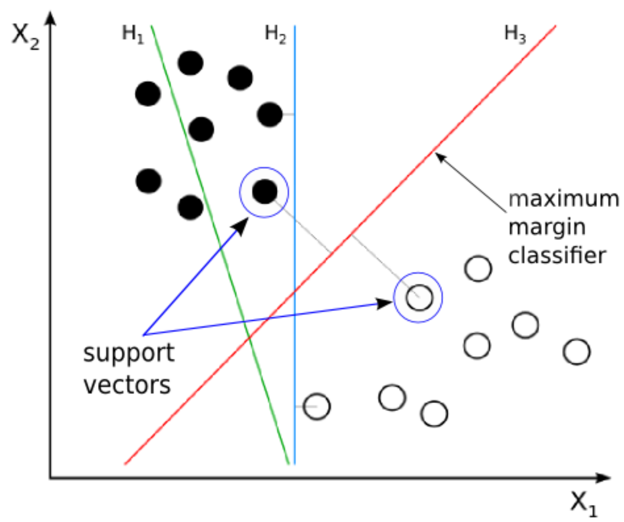


**Fig -2** :Support Vector Machine Classifier

It is basically used for two-class gathering issues. However, it might be used for multi-class issues by a one-against-rest approach. SVM is remarkable since it offers charming features to manage the request issue. For digit requests, the MNIST dataset is stacked into our program from the start. After that readiness set data is dealt with as commitment to the SVM classifier with the objective that the classifier gets readied. SVM polynomial piece is used. The hypothesized name is facilitated with the first to get the exactness of the readied classifier.

At the point when the arrangement is done, the testing data is given to the classifier to anticipate the imprints and the testing exactness is obtained. The chaos system is made which gives the probability between the genuine data and the foreseen data. Using the confusion lattice, the introduction gauges like exactness, audit, and f1 score can be resolved.

$$Precision = TP/(TP + FP) \text{---------------------} \quad (i)$$

where TP = True Positive, FP = False Positive

$$Recall = TP/(TP + FN) \text{--------------------------} \quad (ii)$$

where FN = False Negative

$$F1\ Score = (2 * Precision * Recall)/(Precision + Recall) \text{---------------} \quad (iii)$$

The confusion matrix obtained here is in the form of matrix M 10x10 because it is a multi-class classification (0-9) and as a result the formulae for the precision and recall are as follows

$$Precision_i = M_{ii}/\sum_j M_{ji} \text{-----------------------} (iv)$$

$$Recall_i = M_{ii}/\sum_j M_{ji} \text{-----------------------} (v)$$

The below is the confusion matrix obtained for the trained data set using SVM classifier.

**Table 1: Confusion matrix for Trained Data Set using SVM classifier**

| Digit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 602 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 1 |
| **1** | 0 | 692 | 1 | 1 | 0 | 0 | 0 | 3 | 3 | 0 |
| **2** | 2 | 2 | 538 | 1 | 0 | 0 | 0 | 3 | 3 | 0 |
| **3** | 0 | 0 | 4 | 597 | 0 | 5 | 0 | 4 | 6 | 2 |
| **4** | 1 | 1 | 2 | 0 | 538 | 0 | 0 | 2 | 1 | 7 |
| **5** | 1 | 0 | 3 | 6 | 0 | 547 | 3 | 1 | 3 | 2 |
| **6** | 1 | 1 | 1 | 0 | 0 | 0 | 573 | 0 | 0 | 0 |
| **7** | 0 | 2 | 3 | 2 | 2 | 1 | 0 | 584 | 2 | 5 |
| **8** | 1 | 3 | 2 | 4 | 0 | 4 | 0 | 3 | 601 | 3 |
| **9** | 0 | 3 | 1 | 5 | 5 | 0 | 0 | 4 | 0 | 593 |

The precision, recall and f1 score values for the trained data set are calculated using the above confusion matrix. These values are as mentioned in table 2.

**Table 2: Precision, Recall and F1 score using SVM classifier**

| Digit | Precision | Recall | F1 Score |
|---|---|---|---|
| **0** | 0.97 | 0.99 | 0.98 |
| **1** | 0.98 | 0.99 | 0.98 |
| **2** | 0.98 | 0.97 | 0.98 |
| **3** | 0.97 | 0.98 | 0.98 |
| **4** | 0.98 | 0.98 | 0.98 |
| **5** | 0.98 | 0.97 | 0.97 |
| **6** | 0.98 | 0.98 | 0.98 |
| **7** | 0.98 | 0.98 | 0.98 |
| **8** | 0.97 | 0.97 | 0.97 |
| **9** | 0.98 | 0.97 | 0.97 |
| **Average** | 0.98 | 0.98 | 0.98 |

The Trained Data Set obtained accuracy of 97.75% and the Test Data Set obtained accuracy of 97.74% using the SVM classifier on MNIST data set.

## 3.2 K-NEAREST NEIGHBOR (KNN) CLASSIFIER FOR DIGIT RECOGNITION

KNN is also one of the machine learning techniques which can be used for both classification and regression. Depending on the weight of the nearest neighbors among the classes the new data is classified.

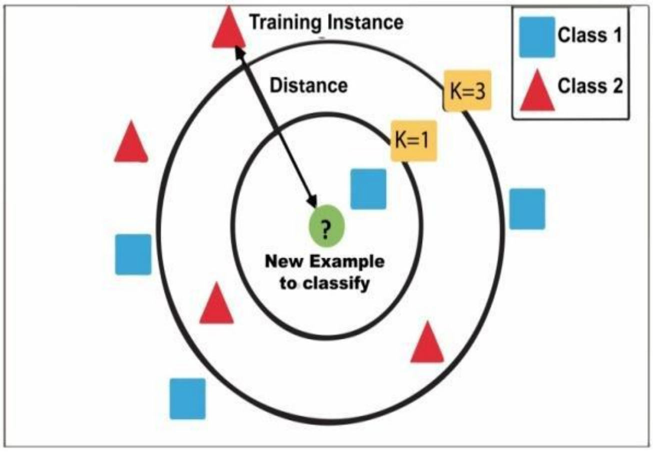


**Fig -2**: Example of KNN Classifier

For the digit acknowledgment process, the MNIST informational index is first stacked into our program. Presently KNN (k=5) classifier is to be prepared utilizing the preparation information. The classifier figures the separation between the first focuses and the focuses given to grouping the digit picture. When the digit name is distinguished, it is contrasted with the first names with get the preparation precision. Similarly, test precision is likewise acquired.

Here also, the confusion matrix is obtained and precision and recall values are computed using the formulas (iv) and (v).

**Table 3: Confusion matrix for Trained Data set using KNN classifier**

| Digit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 600 | 1 | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 2 |
| **1** | 0 | 650 | 2 | 0 | 1 | 0 | 0 | 2 | 0 | 0 |
| **2** | 5 | 1 | 583 | 1 | 0 | 0 | 1 | 2 | 2 | 2 |
| **3** | 2 | 1 | 4 | 555 | 0 | 6 | 0 | 4 | 3 | 3 |
| **4** | 0 | 5 | 0 | 0 | 577 | 0 | 3 | 1 | 0 | 5 |
| **5** | 2 | 1 | 0 | 2 | 2 | 519 | 5 | 0 | 0 | 2 |
| **6** | 5 | 1 | 0 | 0 | 2 | 3 | 565 | 0 | 0 | 0 |
| **7** | 0 | 4 | 0 | 0 | 1 | 0 | 0 | 608 | 1 | 7 |
| **8** | 1 | 7 | 1 | 7 | 1 | 13 | 3 | 2 | 557 | 8 |
| **9** | 0 | 2 | 2 | 6 | 4 | 1 | 0 | 5 | 3 | 619 |

The above Table 3 shows the confusion matrix obtained for the trained data set using the KNN classifier. The precision, recall and f1 score values are calculated using this confusion matrix.

The below table 4 shows the precision, recall and f1 score values obtained for the trained data set using the KNN classifier. In the same way, test data set accuracy and precision, recall and f1 score are obtained.

**Table 4: Precision, Recall and F1 score for KNN on trained data set**

| Digit | Precision | Recall | F1 Score |
|---|---|---|---|
| **0** | 0.96 | 0.99 | 0.98 |
| **1** | 0.95 | 1.00 | 0.97 |
| **2** | 0.98 | 0.96 | 0.97 |
| **3** | 0.97 | 0.97 | 0.97 |
| **4** | 0.97 | 0.96 | 0.97 |
| **5** | 0.97 | 0.97 | 0.97 |
| **6** | 0.98 | 0.99 | 0.98 |
| **7** | 0.96 | 0.96 | 0.96 |
| **8** | 0.99 | 0.94 | 0.96 |
| **9** | 0.95 | 0.95 | 0.95 |
| **Average** | 0.97 | 0.97 | 0.97 |

The Trained Data Set obtained accuracy of 97.21% and the Test Data Set obtained accuracy of 96.82% using the KNN classifier on MNIST data set.

## 3.3 RANDOM FOREST CLASSIFIER

Irregular woodland, similar to its name infers, comprises an enormous number of individual choice trees that work as a group. Every individual tree in the arbitrary woodland lets out a class forecast and the class with the most votes turns into our model's expectation. The crucial idea driving irregular woods is a straightforward yet amazing one — the shrewdness of groups. In information science talk, the explanation that the irregular backwoods model works so well is: countless generally uncorrelated models (trees) working as a board of trustees will outflank any of the individual constituent models.

The low relationship between models is the key. Much the same as how speculations with low relationships (like stocks and securities) meet up to shape a portfolio that is more prominent than the aggregate of its parts, uncorrelated models can create group expectations that are more exact than any of the individual forecasts. The purpose behind this magnificent impact is that the trees shield each other from their individual mistakes (as long as they don't continually all fail a similar way). While a few trees might not be right, numerous different trees will be correct, so as a gathering the trees can move in the right heading. So the requirements for Random Forest to perform well are:

There should be some real sign in our highlights with the goal that models fabricated utilizing those highlights show improvement over arbitrary speculating. The expectations (and subsequently the blunders) made by the individual trees need to have low relationships with one another.

Here likewise, the disarray framework is gotten and accuracy and review esteem are figured utilizing the recipes (iv) and (v).

**Table 5: Confusion matrix for Trained Data set using Random Forest classifier**

| Digit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 971 | 0 | 0 | 0 | 0 | 2 | 3 | 1 | 3 | 0 |
| **1** | 0 | 1122 | 3 | 4 | 0 | 2 | 2 | 0 | 1 | 1 |
| **2** | 6 | 0 | 997 | 6 | 3 | 0 | 4 | 9 | 7 | 0 |
| **3** | 1 | 0 | 10 | 974 | 0 | 7 | 0 | 9 | 6 | 3 |
| **4** | 1 | 0 | 1 | 0 | 950 | 0 | 6 | 0 | 3 | 21 |
| **5** | 3 | 0 | 1 | 14 | 3 | 856 | 4 | 2 | 6 | 3 |
| **6** | 8 | 3 | 0 | 0 | 3 | 3 | 940 | 0 | 1 | 0 |
| **7** | 1 | 3 | 23 | 3 | 0 | 0 | 0 | 985 | 3 | 10 |
| **8** | 5 | 0 | 5 | 8 | 3 | 5 | 2 | 4 | 932 | 10 |
| **9** | 5 | 5 | 4 | 8 | 15 | 5 | 1 | 5 | 4 | 957 |

The above Table 5 shows the confusion matrix obtained for the trained data set using the Random forest classifier. The precision, recall and f1 score values are calculated using this confusion matrix.

The below table 6 shows the precision, recall and f1 score values obtained for the trained data set using the Random forest classifier. In the same way, test data set accuracy and precision, recall and f1 score are obtained.

**Table 6: Precision, Recall and F1 score for random forest classifier on trained data set**

| Digit | Precision | Recall | F1 Score |
|---|---|---|---|
| **0** | 0.97 | 0.99 | 0.98 |
| **1** | 0.99 | 0.99 | 0.99 |
| **2** | 0.95 | 0.97 | 0.96 |
| **3** | 0.96 | 0.96 | 0.96 |
| **4** | 0.97 | 0.97 | 0.97 |
| **5** | 0.97 | 0.96 | 0.97 |
| **6** | 0.98 | 0.98 | 0.98 |
| **7** | 0.97 | 0.96 | 0.96 |
| **8** | 0.96 | 0.96 | 0.96 |
| **9** | 0.95 | 0.95 | 0.95 |
| **Average** | 0.97 | 0.97 | 0.97 |

The Trained Data Set obtained accuracy of 97.08% and the Test Data Set obtained accuracy of 96.84% using the Random forest classifier on MNIST data set.

## 4. RESULTS AND DISCUSSION

The accuracies of the algorithms SVM, KNN and Random Forest are tabulated below in table 7.

**Table 7: Accuracy Percentage of Algorithms**

| Algorithm | Trained Data Accuracy | Test Data Accuracy |
|---|---|---|
| **SVM** | 97.75% | 97.74% |
| **KNN** | 97.21% | 96.82% |
| **Random Forest** | 97.08% | 96.84% |

It can be clearly observed that SVM has more accuracy compared to Random Forest and KNN Classifiers for both the trained data set and test data. Moreover, Random forest is giving less accuracy in comparison with the remaining. The below image figure 4 shows the comparison of the accuracies of the algorithms on both trained and test data sets.

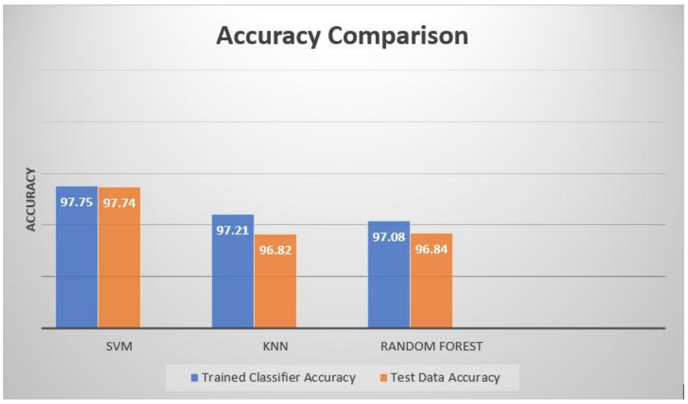


**Chart -1**: Accuracy Comparison of algorithms

The error rate for KNN classifier is 3.18% on the test data which is higher when compared to both SVM classifier and Random Forest. It is clear that SVM has given the least error rate of about 2.25% on the trained data set, yet it's not optimal and this error has to be reduced in the future.

## 5. CONCLUSION AND FUTURE SCOPE

As AI calculations are utilized like K-NN, SVM, and RF alongside various parameters and highlight scaling vectors. We likewise observed an alternate examination among the strategies classifiers regarding highlights of exactness and timing. Exactness can adjust as it relies upon the utilization of specific preparing and testing information, and this can additionally be improved if the quantity of preparing and testing information is given. Each classifier has its own exactness and time utilization. In this field, we can find the way that if the intensity of CPU changes to GPU, the classifier can perform with better exactness and less time and better outcomes can be observed. However, not all manually written digit sets are standardized in size, or focused and put away successively as 28x28 pixel pictures in grayscale in the genuine cases.

To additionally improve the exhibition, conceivable future works are as per the following:

(1) despite the fact that the second put together highlights perform greatly with respect to the entire, corresponding

highlights like concavity investigation may help in segregating confounding numerals. For instance, Indo-Arabic numerals "2" and "3" can all the more likely be isolated by thinking about the first size before standardization.

(2) For classifier structure, it is smarter to choose model parameters by cross approval instead of experimentally as done in our analyses.

(3) Combining different classifiers can improve acknowledgement precision. The exhibition of the classifier can be estimated as far as the capacity to recognize a condition appropriately, exactness, positive forecasts, and capacity to get condition accurately. As such, we closed a brief to the classifiers of Machine learning.

## REFERENCES

[1] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., Edinburgh, UK, 2003, pp. 958-963. T Siva Ajay, “Handwritten Digit Recognition Using Convolutional Neural Networks”,

[2] International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 07 | July -2017.

[3] Wu, Ming & Zhang, Zhen. (2019). Handwritten Digit Classification using the MNIST Data Set.

[4] Al-Wzwazy, Haider& M Albehadili, Hayder&Alwan, Younes& Islam, Naz& E Student, M &, Usa. (2016). Handwritten Digit Recognition Using Convolutional Neural Networks. International Journal of Innovative Research in Computer and Communication Engineering. 4. 1101-1106.

[5] AL-Mansoori, Saeed. (2015). Intelligent Handwritten Digit Recognition using Artificial Neural Network. 10.13140/RG.2.1.2466.0649.

[6] Srivastava, Shobhit & Kalani, Sanjana& Hani, Umme & Chakraborty, Sayak. (2017). Recognition of Handwritten Digits using Machine Learning Techniques. International Journal of Engineering Research and. V6. 10.17577/IJERTV6IS050456.

[7] Mishra, Shashank & Malathi, D & Senthil Kumar, K. (2018). DIGIT RECOGNITION USING DEEP LEARNING.