A Project Report

On

Handwritten Character Recognition

Submitted by

BELLA BABU (20BCE0558)

AADHARSH S(20BCE0562)

ZAMAN SALEEL (20BCE2025)

For

Image Processing

CSE 4019

Slot: G2, J Component

B.Tech. in Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

November 2022

TABLE OF CONTENTS

AIM

This project's goal is to create a classification algorithm that can scan handwritten alphabets and digits and convert them into machine-readable representations. The major goal of this effort is to guarantee efficient and trustworthy methods for handwritten character recognition. The purpose of a handwritten character recognition system is to collect and decipher handwritten data from images or paper documents. Convolutional neural networks (CNNs) are the most successful method for resolving handwriting recognition issues because they are very good at understanding the structure of handwritten characters and words in ways that facilitate the automatic extraction of distinctive features and the aim is to implement CNN to achieve character recognition.

ABSTRACT

PROBLEM STATEMENT

Handwriting varies from person to person and is not always consistent in terms of size, width, orientation, and justification to the margins, the general issue would arise when categorizing the characters due to the similarity between characters like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. When multiple people write the same character in a range of different handwritings, the problem is exacerbated. Last but not least, the distinctiveness and variation of each person's handwriting has an impact on how the characters are formed and appear.

The problem of handwritten character identification, employing a classifier, has amazing significance and utility, such as online digit recognition on PC tablets, recognizing zip codes on mail, process bank check amounts, and processing numeric sections of structures filled out by hand (for example tax forms). Not only does handwritten digit recognition have professional and commercial uses, but it also has everyday used that can be quite helpful to people who are blind. Additionally, it facilitates the simple resolution of challenging issues, making our life easier

EXISITNG METHODS:

Supervised Learning:

A machine learning model is trained under supervision using labelled data. It indicates that the appropriate classification is already attached to the data you have. Using supervised learning to assist with value prediction for fresh data is a popular use.

When using supervised learning, you must continually update your models in order to ensure that the predictions you receive are still accurate.

Unsupervised Learning:

When you train a model using unlabeled data, this is known as unsupervised learning. As a result, the model will need to develop its own features in order to classify the data and generate predictions.

1. **Support Vector Machine (SVM)** is a supervised machine learning method that can be applied to problems involving classification or regression. However, classification issues are where it is most frequently utilised. With the SVM algorithm, each data point is represented as a point in an n-dimensional space (where n is the number of features), with each feature's value being the value of a certain coordinate. Then, classification is performed by identifying the hyper-plane that distinguishes the two classes.

   Support vector classifiers can be used to identify single handwritten digits that have been optically scanned. SVM can be used for information retrieval and subsequent label-based data categorization. When there are a lot of characteristics, SVMs are effective.

2. **Artificial Neural Networks (ANN)** are algorithms that simulate complex patterns and foresee problems, and they are based on brain activity. The idea of biological neural networks in the human brain gave rise to the Artificial Neural Network (ANN), a deep learning technique. An effort to simulate how the human brain functions led to the creation of ANN.

   A computational model known as an artificial neural network aims to replicate the composition and operation of biological brain networks [13]. An artificial neural network (ANN) is used to train the network with features taken from handwritten characters, and then the network is tested using some character data. In this case, the backpropagation technique is utilised. A supervised learning method used to train Multi-Layer Perceptrons is the BP algorithm.

3. **K-nearest neighbors (KNN) algorithm** is an example of a supervised machine learning algorithm that may be applied to classification and regression predicting issues. The K-nearest neighbours (KNN) method predicts the values of new datapoints based on "feature similarity," which further indicates that the new data point will be given a value depending on how closely it resembles the points in the training set.

   In order to recognise handwriting, this method compares the input character to the dataset's elements in order to determine how similar they are. This similarity determines the approximate character that is returned.

4. **Naive Bayes** techniques are a collection of supervised learning algorithms built on the "naive" assumption that all feature pairs are conditionally independent given the value of the class variable.

   The classifier is the Nave Bayes (NB) classification algorithm. We split the image into numerous parts for the classification procedure. We create a vector of descriptors for each block. Then, we cluster the low-level characteristics, such as Zernike and Hu moments, using K-means. Finally, we use a Bayesian networks classifier to categorise

the entire word picture. The use of both handwritten and machine-printed character pictures was tested.

5. **Random forests** are a supervised learning algorithm. It can be used both for classification and regression. Random forests create decision trees on randomly selected data samples, get forecasts from each tree and choose the most excellent arrangement by implies of voting. It works in four steps: 1. Select random samples from a given dataset. 2. Construct a decision tree for each test and get an expected result from each decision tree. 3. Perform a vote for each predicted result. 4. Select the prediction result with the most votes as the final prediction.

INTRODUCTION

Recognition is the process of recognizing something or someone based on prior knowledge or experiences. In a similar vein, character recognition only entails identifying or recognizing the characters in any given document. The ability of a computer to recognize a human handwritten character from various sources, such as photographs, papers, touch displays, etc., is known as handwritten character recognition. A significant neural network-based project uses the MNIST dataset to recognize handwritten characters. In essence, it recognizes handwritten characters in scanned photographs.

The primary goal was to implement a method for pattern characterization to recognize the handwritten characters offered in the MINIST data collection of handwritten character images. In many recent decades, the MNIST database's handwritten characters have gained widespread recognition for its ability to reduce mistake rates when used with various classifiers and parameters.

Machine learning offers several ways to recognize manually written digits with less effort from humans. Deep Learning is a machine learning technique that teaches computers to do what comes naturally to people: learn by doing. Human efforts in seeing, learning, recognizing, and many other areas can be reduced with the use of deep learning techniques. The computer learns to perform categorization tasks from images or content from any document using deep learning.

LITERATURE SURVEY:

S.NO	TITLE	FINDINGS	REFERENCE
1.	MNIST Handwritten Digit Recognition using Machine Learning	They use machine learning algorithm that is Convolutional Neural Network (CNN). MNIST database consist of nine digits that is 0 to 9. It can be used to change over manually written digits into machine lucid arrangements. The primary target of this work is to guarantee successful and dependable methodologies for acknowledgment of written by hand digits and make banking tasks more straightforward and mistake free. The MNIST database will be identified by the machine very accurately.	E. R. G, S. M, A. R. G, S. D, T. Keerthi and R. S. R, "MNIST Handwritten Digit Recognition using Machine Learning," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022, pp. 768-772, doi: 10.1109/ICACITE53722. 2022.9823806.
2.	Handwritten Digit Recognition Using Neural Network	This paper focuses on Neural Network (NN) approaches. Among the three famous NN approaches: deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN), the specialization of CNN as compared to other NN of being able to detect pattern is what makes it so useful for recognizing handwritten digits in this paper. Our goal is to implement a CNN based handwritten digit recognition model that uses the image of a digit and recognizes the digit present in the image. The performance is tested on MNIST dataset. The network was trained on 60,000 and tested on 10,000 numeral samples. We carried out extensive	https://ijcrt.org/papers/IJCRT2107214.pdf Penan Rajput, Jayesh Nahar, Sanjeev Kumar, Aniket Pathak in International Journal of Creative Research Thoughts (IJCRT)

		experiments and achieved a recognition accuracy of 99.87%.	
3.	Handwritten Digit Recognition Using Machine Learning	Handwriting recognition systems are also stand out on this field. In this study, handwriting digit recognition process has been done with algorithms having different working methods. These algorithms are Support Vector Machine (SVM), Decision Tree, Random Forest, Artificial Neural Networks (ANN), K-Nearest Neighbor (KNN) and K- Means Algorithm. The working logic of the handwriting digit recognition process was examined, and the efficiency of different algorithms on the same database was measured. A report was presented by making comparisons on the accuracy.	Authors: Rabia KARAKAYA, Serap KAZAN in Sakarya University Journal of Science Link: https://www.researchgate.net/publication/347943261_Handwritten_Digit_Recognition_Using_Machine_Learning/link/6023b27d92851c4ed55f1661/
4	Handwritten Digit Recognition Using Machine Learning: A Review	A number of 60,000 images were used as training sets of images with pixel size of 28x28. The images/training sets were matched with original image. It was found out after complete analysis and review that classifier ensemble system has the least error rate of just 0.32%. In this paper, review of different methods handwritten digit recognition were observed and analyzed.	A. Shrivastava, I. Jaggi, S. Gupta and D. Gupta, "Handwritten Digit Recognition Using Machine Learning: A Review," 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), 2019, pp. 322-326, doi: 10.1109/PEEIC47157.2019.8976601.
5.	Handwritten Text Recognition: With Deep Learning and Android	This research paper offers a new solution to traditional handwriting recognition techniques using concepts of Deep learning and computer vision. An extension of MNIST	Shubham Sanjay Mor, Shivam Solanki, Saransh Gupta, Sayam Dhingra, Monika Jain, Rahul Saxena in International

		digits dataset called the Emnist dataset has been used. It contains 62 classes with 0-9 digits and A-Z characters in both uppercase and lowercase. An application for Android, to detect handwritten text and convert it into digital form using Convolutional Neural Networks, abbreviated as CNN, for text classification and detection, has been created. Prior to that we pre-processed the dataset and applied various filters over it. We designed an android application using Android Studio and linked our handwriting text recognition program using tensorflow libraries. The layout of the application has been kept simple for demonstration purpose. It uses a protobuf file and tensorflow interface to use the trained keras graph to predict alphanumeric characters drawn using a finger.	Journal of Engineering and Advanced Technology. Link: https://www.ijeat.org/wp-content/uploads/papers/v8i2s2/B10370182S219.pdf

Handwriting Recognition using Machine Learning - Anil Chandra Naidu Matcha

The initial approaches of solving handwriting recognition involved Machine Learning methods like Hidden Markov Models (HMM), SVM etc. Once the initial text is pre-processed, feature extraction is performed to identify key information such as loops, inflection points, aspect ratio etc. of an individual character. These generated features are now fed to a classifier say HMM to get the results. The performance of machine learning models is pretty limited due to manual feature extraction phase and their limited capacity of learning. Feature extraction step varies for every individual language and hence is not scalable. With the advent of deep learning came tremendous improvements in accuracy of handwriting recognition.

RNN/LSTM as we know can deal with sequential data to identify temporal patterns and generate results. But they are limited to dealing with 1D data and hence won't be directly applicable to image data. To solve this problem, the authors in this paper proposed a multi-dimensional RNN/LSTM structure as can be seen in the figure below

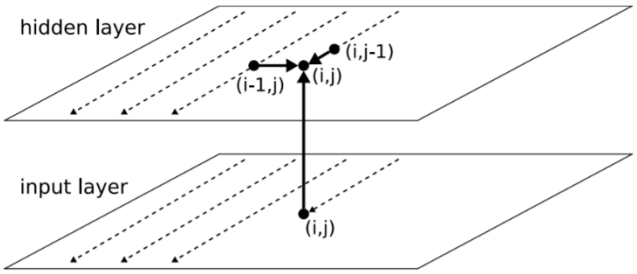


Figure 1: **Two dimensional MDRNN**. The thick lines show connections to the current point (i, j). The connections within the hidden layer plane are recurrent. The dashed lines show the scanning strips along which previous points were visited, starting at the top left corner.

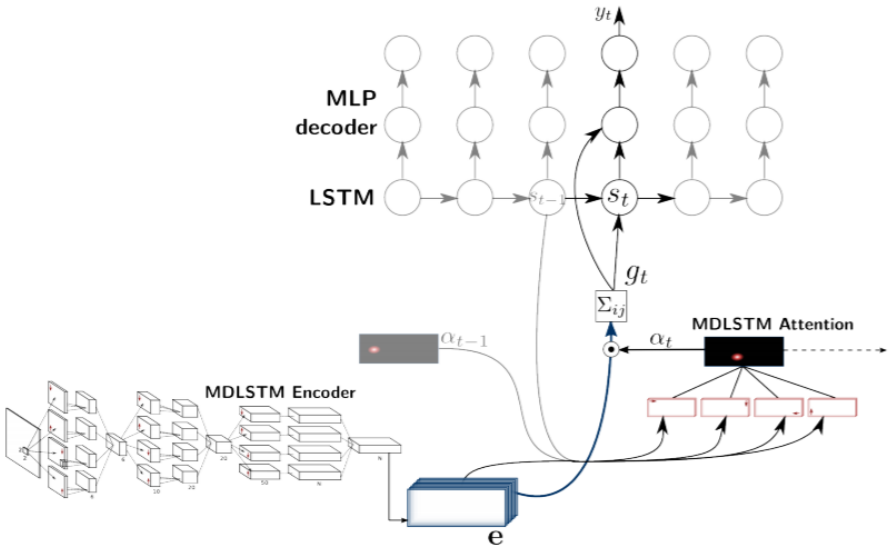


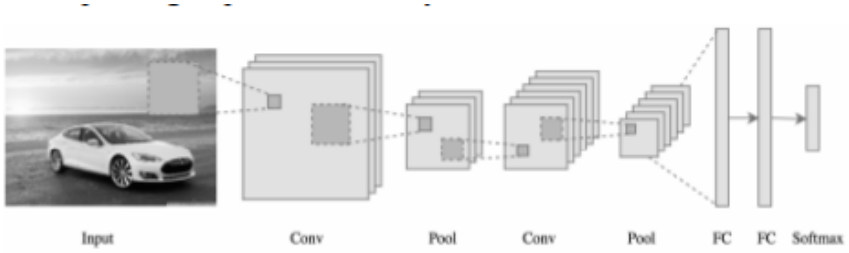
Figure 2: Proposed architecture. The encoder network has the same architecture as the standard network of Figure 1, except for the collapse and softmax layers. At each timestep, the feature maps, along with the previous attention map and state features are fed to an MDLSTM network which outputs new attention weights at each position. The weighted sum of the encoder features is computed and given to the state LSTM, and to the decoder. The decoder also considers the new state features and outputs character probabilities.

Although there have been significant developments in technology which help in better recognition of handwritten text, HTR (Handwritten Text Recognition) is a far from a solved problem compared to OCR (Optical Character Recognition) and hence is not yet extensively

employed in industry. Nevertheless, with the pace of technology evolution and with the introduction of models like transformers, we can expect HTR models to become a commonplace soon.

Handwritten Text Recognition: With Deep Learning and Android - Shubham Sanjay Mor, Shivam Solanki, Saransh Gupta, Sayam Dhingra, Monika Jain, Rahul Saxena

This paper uses EMNIST dataset, there have been several accomplishments that has been achieved using this dataset. Even before using Deep learning, Handwritten text recognition has been made possible, however their accuracies were really low or they had a relatively small dataset as said by Line Eikvil. In this paper, usage of OCR has been discussed such as in Speech Recognition, Radio Frequency, Vision systems, Magnetic Stripes, Bar Code and Optical Mark Reading. A popular machine learning task is classifying the MNIST dataset, which is dataset of numbers. This research paper offers a new solution to traditional handwriting recognition techniques using concepts of Deep learning and computer vision. An extension of MNIST digits dataset called the Emnist dataset has been used. It contains 62 classes with 0-9 digits and A-Z characters in both uppercase and lowercase. An application for Android, to detect handwritten text and convert it into digital form using Convolutional Neural Networks, abbreviated as CNN, for text classification and detection, has been created. Prior to that we pre-processed the dataset and applied various filters over it. An android application using Android Studio has been designed and linked the handwriting text recognition program using tensorflow libraries. The layout of the application has been kept simple for demonstration purpose. It uses a protobuf file and tensorflow interface to use the trained keras graph to predict alphanumeric characters drawn using a finger.

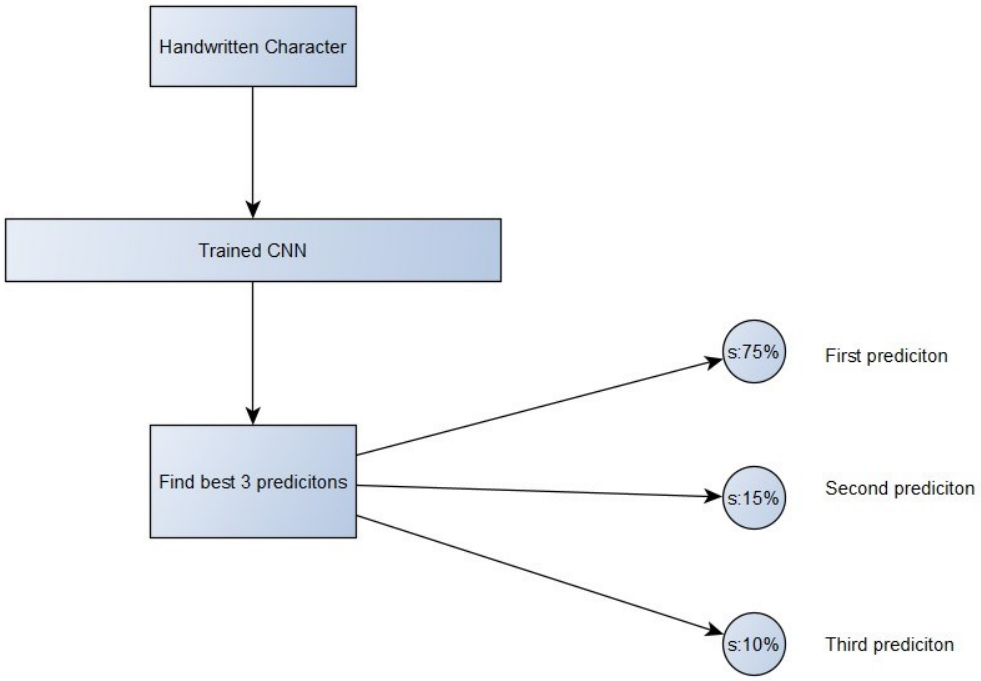


CNN image classifications take an input image, process it and classify it under certain categories (E.g., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels. Based on the image resolution, it will see h x w x d (h = Height, w = Width, d = Dimension). E.g., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image where 3 and 1 are the number of color values required to represent an image pixel.

Other Works:

- For word recognition, a Paper by Pham et al., used a 2-layer CNN which fed into a bidirectional recurrent neural network (RNN) with Long Short-Term Memory (LSTM) cells.
- The best model implemented, is by Graves and Schmiduber with a multidimensional RNN structure.
- Another paper on 'Handwritten Text Recognition' by M.J. Castro-Bleda dealt with dataset with slanted words as well and corrected them during pre-processing.
- Development of English Handwritten Recognition Using Deep Neural Network by Teddy Surya and Ahmad Fakhrur uses a Deep Neural Network model having two Encoding layer and one SoftMax layer on the EMNIST dataset. Their accuracy using DNN was way better than the earlier proposed patternnet and feedforwardnet ANN (Artificial Neural Networks).
- Handwritten text recognition can also be achieved on basis of Relaxation Convolutional Neural Networks(R-CNN) and alternatively trained relaxation convolutional neural networks (ATRCNN) as done by ChunPeng Wu and Wei Fan.

PROPOSED METHOD:



Workflow of the model

DATA PREPROCESSING:

For characters A to Z the A_Z Handwritten Data dataset is used and for numbers 0 to 9 the data is retrieved from the MNIST dataset of keras library. The A_Z Handwritten dataset has 785 columns of which the first column is the label and the other 784 columns are the features (pixels) of each image. The MNIST dataset consist of test and train data which are combined into a single dataset. The MNIST consists of around 70000 images of digits of 28*28-pixel size. The A_Z Handwritten dataset is then reshaped to the form similar to MNIST dataset. Each image is reshaped to 28*28 size. Then these 2 datasets are concatenated to form a merged dataset and 100 randomly picked images from this combined dataset is displayed to verify the correctness of the process.

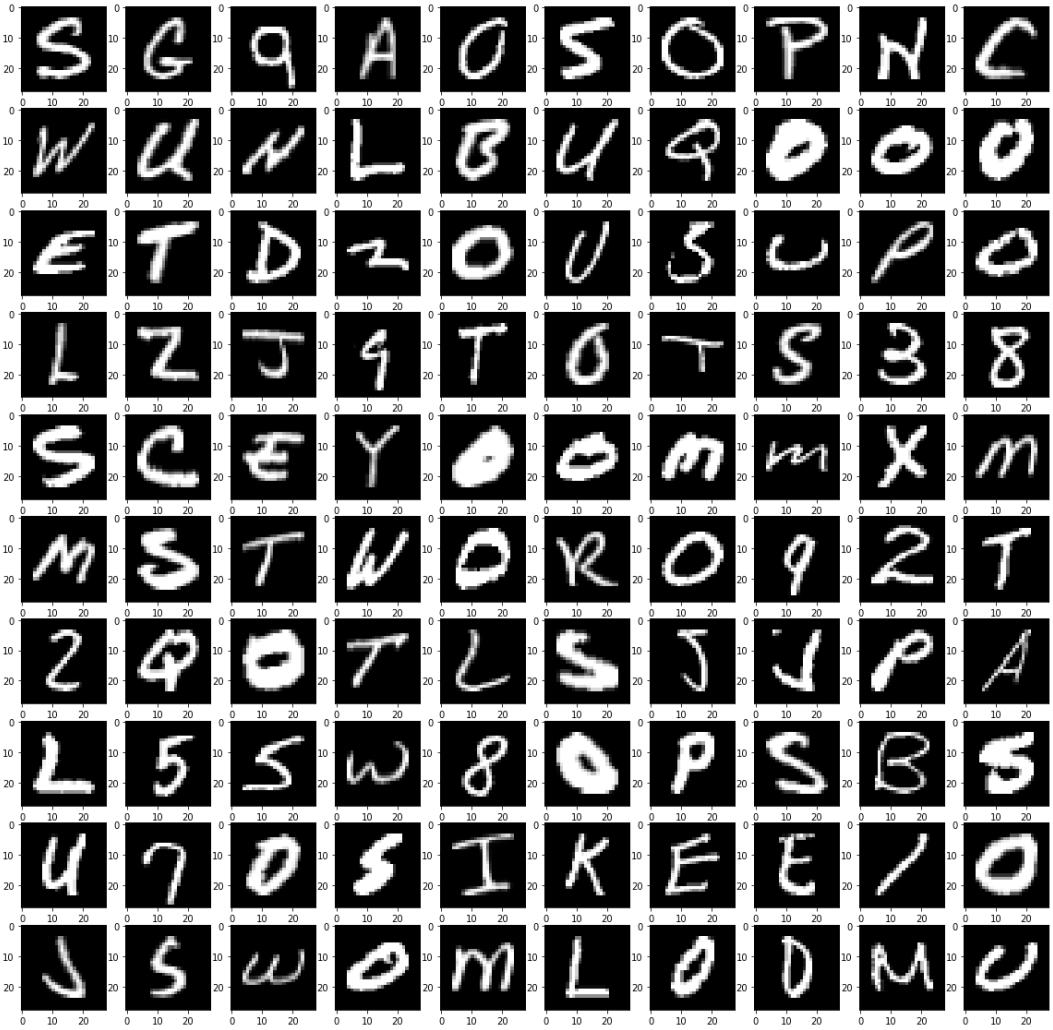
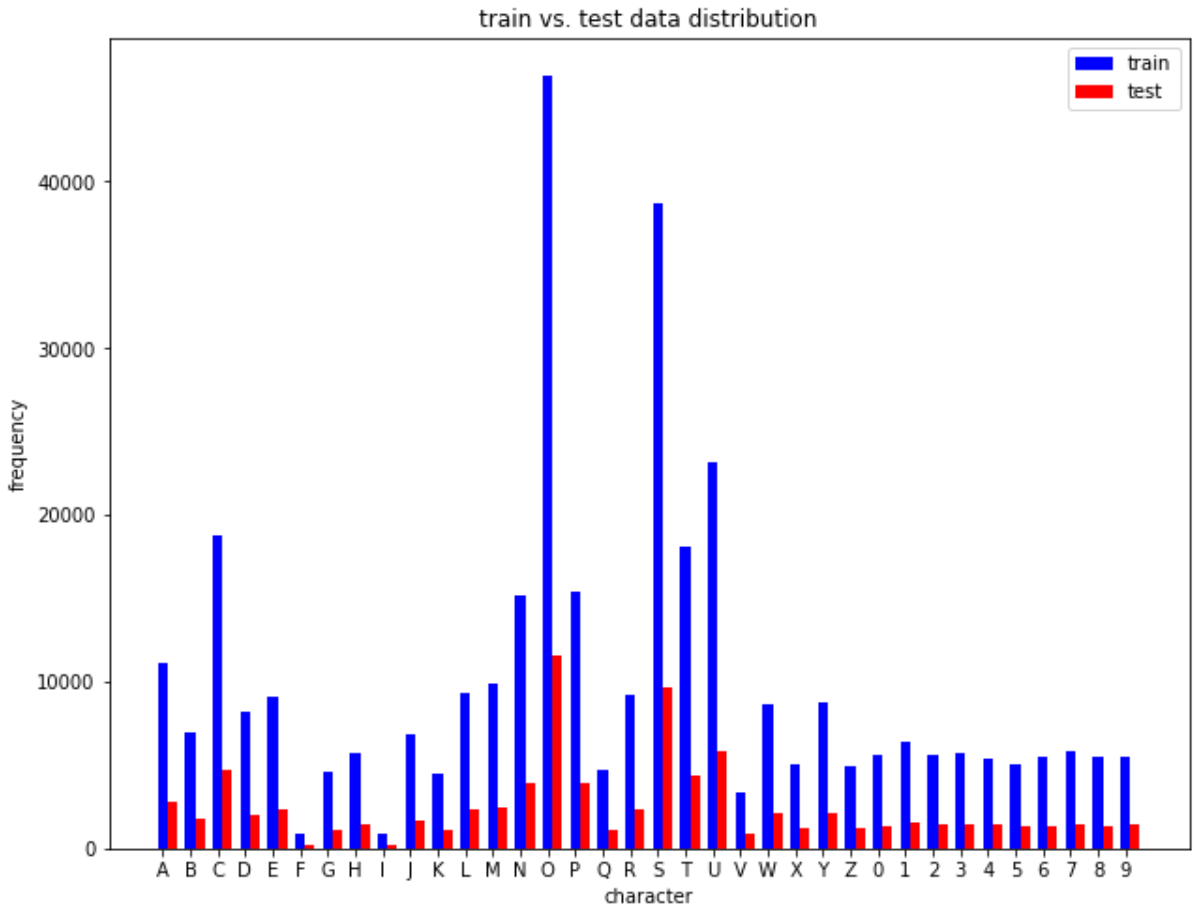


Figure shows random 100 images from the concatenated dataset

Next the dataset is split into train data and test data. 20% percentage of the entire dataset is assigned as test dataset and the rest as train dataset. Both these datasets contain the labels of the characters and their associated images. The train and test labels are then converted to categorical format as CNN accepts label only in categorical format. This leads to the formation of 36 categorical labels, one denoting each of the 26 letters and 10 digits. The train and test data are then converted to a 4D array.

The train and test data distribution are then represented in a plot using the matplotlib library. The train data, test data and labels are then stored in the local machine as numpy files using numpy save function.



Plot shows the frequency of each character in the train and test dataset respectively

CNN ARCHITECTURE:

Convolutional Neural Networks (CNN) are Deep Learning algorithms that take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and differentiate them. It is one of the most well-known deep learning architectures based on human brain's natural visual perception technique. When compared with other classification algorithms, CNN requires much less preprocessing.

The CNN architecture mimics the communication pattern of the neurons in the human brain, boosted by the arrangement of the visual cortex. A CNN can capture the spatial dependencies inside an image to relate with the content of the image for recognition purpose.

CNN consists of the following fundamental parts:

Convolution:
It is a process where a single matrix of numbers (known a kernel or filter), is passed over an image to reshape the image with the filter. Each selected pixel value of the image submatrix needs to be multiplied with the corresponding pixel value from the kernel and then sum up the result to formulate a single pixel value of the filtered image. This micro-scale operation is continued for the whole image. The convolution process captures all low-level features such as borders, colors, gradients, and orientations. Furthermore, the design adjusts for high-level usability with additional layers, allowing a network to understand images almost as well as humans do. Convolution removes high-level features from the source image such as edges. The method reduces the dimensionality of the reshaped image (characteristic) as compared to the input.

Pooling:
A second key component of CNN is pooling. The purpose is to slowly raise the spatial scale of the image description to reduce the computational complexity within the network. In each application, pooling can be of different types - maximum pool, minimum pool, average pool and adaptive pool. Maximum pool (Max-Pooling) is a popular approach for CNN method. It reduces the computing power by reducing the dimensionality by keeping the dominant features that are rotational and translational variants that preserve the model's efficient training cycle. Max-Pooling gives maximum value from its image part covered within the kernel. But on the other side, average pooling provides better noise suppression. It discards the noisy activations and also conducts de-noise along with reduction of the dimensionality.

Fully-connected neural network:
Fully connected neural network are used at the last part of CNN. They are simply artificial neural network which are fully connected. Weights which are involved with the network are computed during the training session. Fully connected neural network receives the end result of convolution/pooling operation and computes the most-matched label that represents the image. This part makes the connection of the image feature vector with the class of the image. The outputs from the convolution/pooling operation are multiplied by the weights associated with the network connection path. The result is then passed through an activation function.

Characters are recognized by CNN by comparing their forms and contrasting their features. The sequential model of CNN is used for character recognition. The CNN design provides a strong fit of the source image to find the distinguishing features in order to be classified. In the training stage, weights, parameters and biases for the transformations from original image to feature vector are determined to get a better understanding of the image's nature.

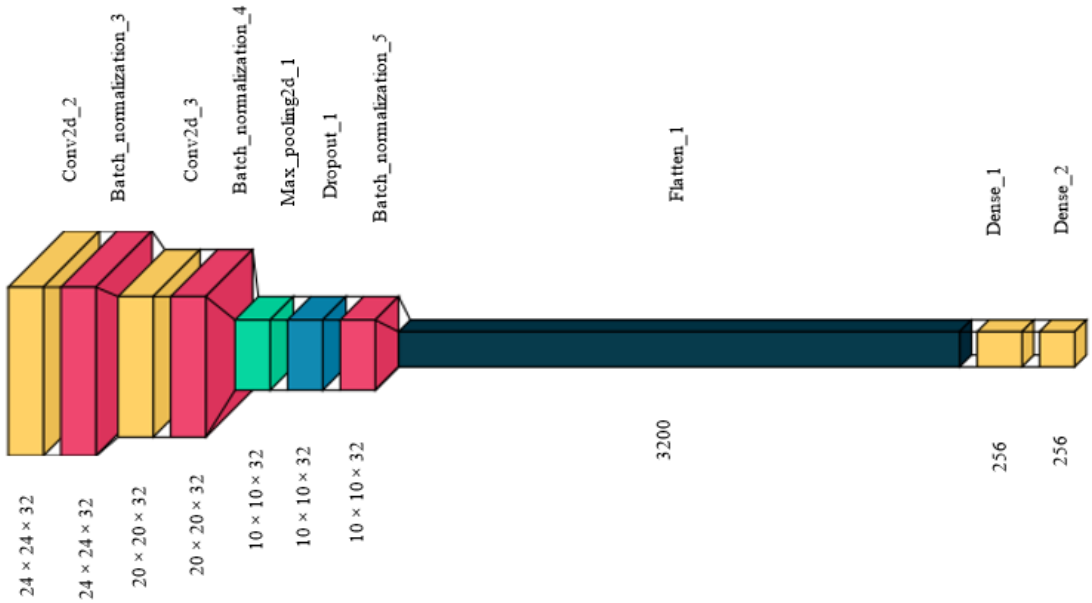


Figure shows the layers of the CNN model

The train, test and the label datasets are loaded and the sequential model is used to frame the CNN model. The CNN architecture consists of 10 layers of which first 8 layers are for feature extraction and the other 2 for classification.

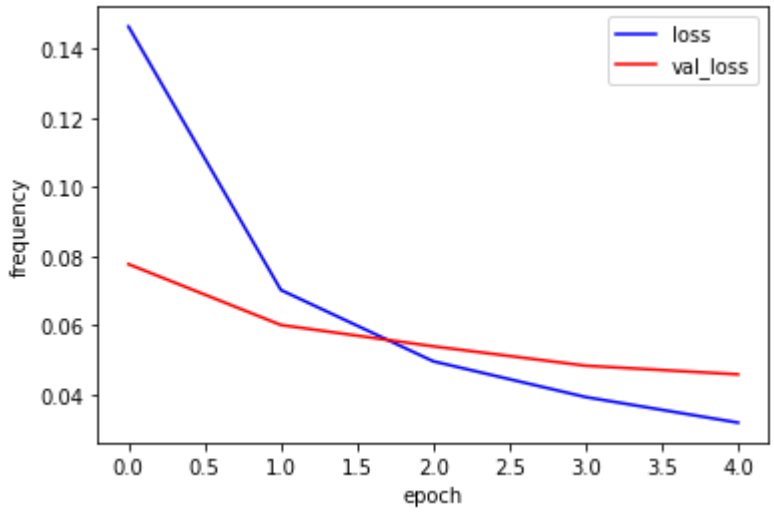
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 24, 24, 32)        832

batch_normalization_3 (Batc  (None, 24, 24, 32)        128
hNormalization)

conv2d_3 (Conv2D)            (None, 20, 20, 32)        25632

batch_normalization_4 (Batc  (None, 20, 20, 32)        128
hNormalization)

max_pooling2d_1 (MaxPooling  (None, 10, 10, 32)        0
2D)

dropout_1 (Dropout)          (None, 10, 10, 32)        0

batch_normalization_5 (Batc  (None, 10, 10, 32)        128
hNormalization)

flatten_1 (Flatten)          (None, 3200)              0

dense_2 (Dense)              (None, 256)               819456
...
Total params: 855,556
Trainable params: 855,364
Non-trainable params: 192
_________________________________________________________________
```

Next the model is compiled and trained using the train dataset and tested using 200 random test data for 5 sets and the model with best validation accuracy and best accuracy are stored in the local machine. The accuracy versus the test set is plotted using plot function to check for the ideal set to be used for the model. Next using the test dataset predictions are derived and these predictions are represented in a confusion matrix format. Using the confusion matrix, we can determine that the distribution between predicted and actual labels are linear and thus the predicted model offers good fit.

APPLICATION:

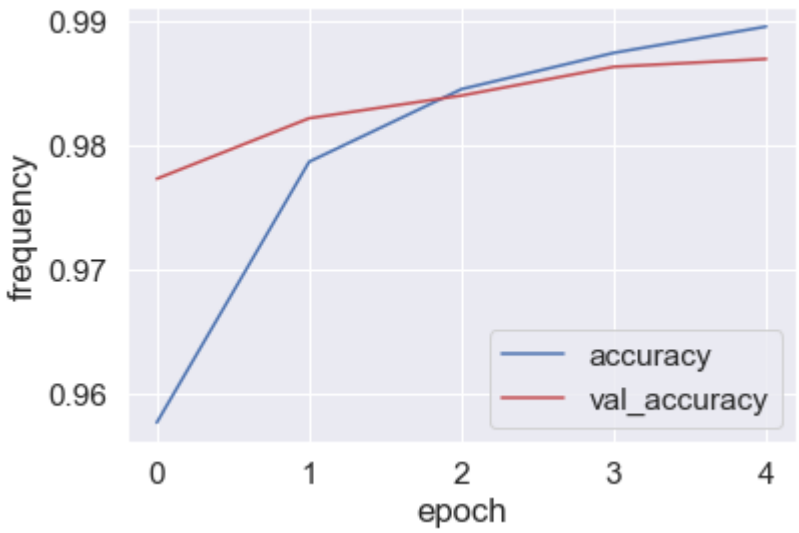
User input is acquired using a whiteboard which allows the user to save the whiteboard file with the character as test.png. Once the user has entered the input, the image is compressed and inverted and then sent to the model to predict the character. The output displays the 3 best predictions with their accuracy.

RESULT AND EVALUATION:



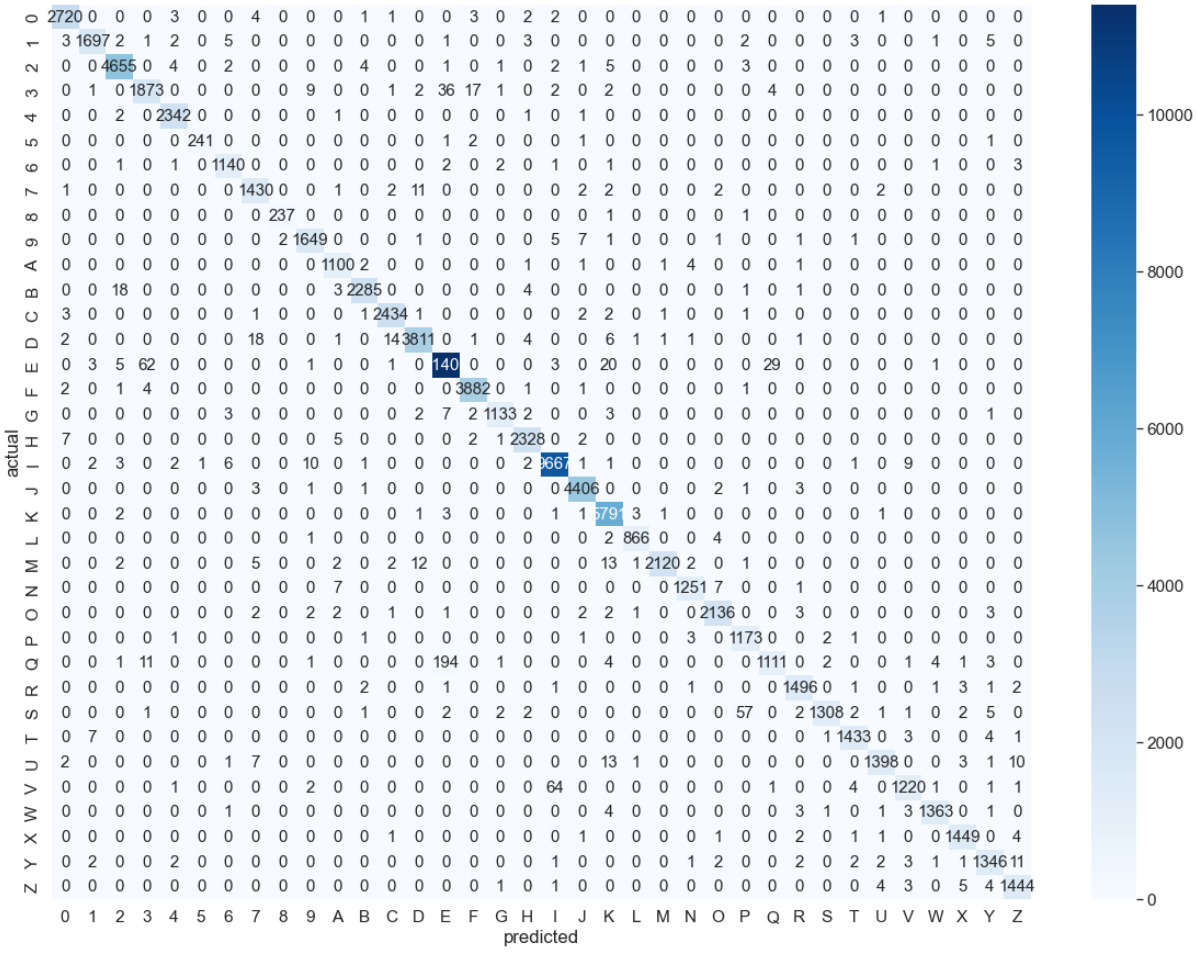
Dataloss vs Test Dataset Batch

From the graph it can be concluded that Loss and validating loss of the model decreases exponentially with test dataset batch.



Accuracy vs Test Dataset Batch

From the graph it can be concluded that Accuracy and validating accuracy of the model increases exponentially with test dataset batch.



Confusion Matrix

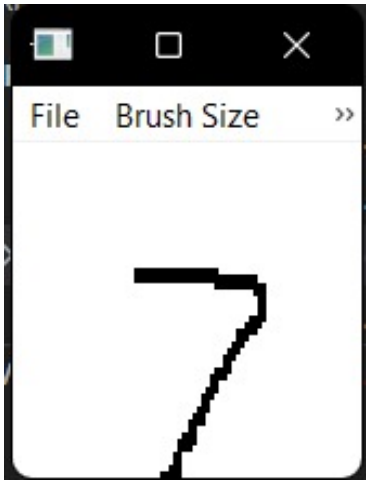
The predicted labels of the test dataset match the actual labels to a high degree. The true positive, true negative, false negative and false positive values that are detected by the model for each character is shown in the confusion matrix.

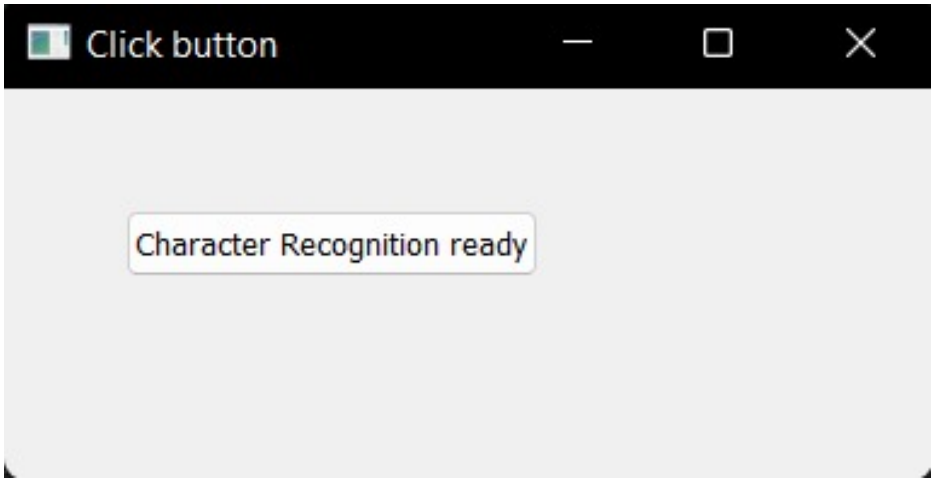
$$\text{ACCURACY} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

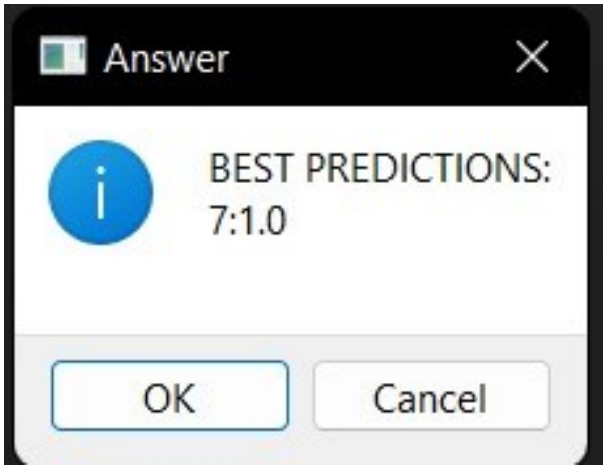
$$\text{ACCURACY} = \frac{76075}{76984} = 98.81$$

Using the confusion matrix and the above accuracy formula we get the accuracy of the proposed model to be **98.81%.**

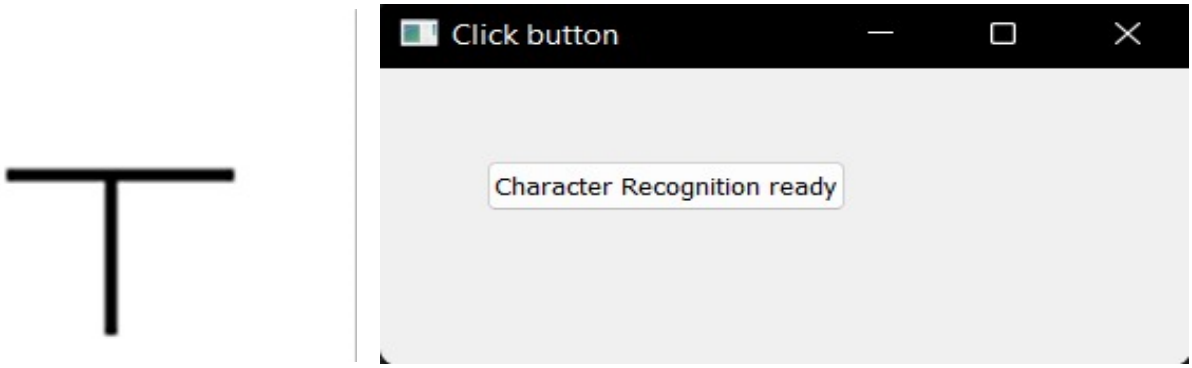
CHARACTER RECOGNITION

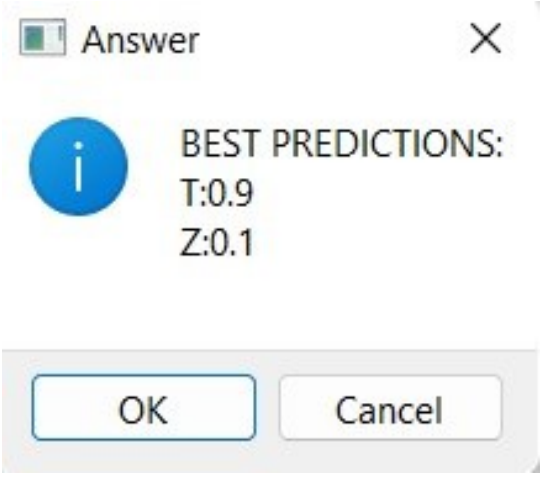






Case 1: The number 7 is getting predicted by the model as 7 with 100% probabiltiy.





Case 2: The character T is getting predicted as T and Z with 90% and 10% probability.

FUTURE WORK EVALUATION MATRIX

Character recognition can be extended to recognize words and sentences. This will prove beneficial for converting written answer sheets and written documents into a digital format which could be read easily by users. This system could be used by teachers while correcting answer sheets, which could help them to recognize words easily. Further the elderly can use it to read handwritten documents, and since digital documents can offer zoom feature it would reduce the strain on the eyes of people as compared to handwritten documents.

CODE (DRIVE LINK)

https://drive.google.com/drive/folders/141IRPTCnPK47pwgqjRY3o70Iua0LreOa?usp=sharing

CONCLUSION

To answer simple problems that can be completed by any person in a flash of an eye, such as text recognition, deep learning techniques like neural networks are being used.

We have only begun to explore machine learning's potential. This technology has countless applications and possibilities. OCR methods from the past resembled biometric technology. The physical attribute match points were collected using photo sensor technology, and they were afterwards converted into a database of recognised categories. However, thanks to contemporary methods like convolution neural networks, we are now able to scan and comprehend words with a level of precision that has never before been possible.

REFERENCES:

1. https://ijcrt.org/papers/IJCRT2107214.pdf
2. https://www.researchgate.net/publication/347943261_Handwritten_Digit_Recognition_Using_Machine_Learning/link/6023b27d92851c4ed55f1661/
3. https://www.ijeat.org/wp-content/uploads/papers/v8i2s2/B10370182S219.pdf
4. https://ieeexplore.ieee.org.egateway.vit.ac.in/stamp/stamp.jsp?tp=&arnumber=9750675
5. https://github.com/githubharald/WordDetector
6. https://ieeexplore.ieee.org.egateway.vit.ac.in/stamp/stamp.jsp?tp=&arnumber=9824910
7. https://ieeexplore.ieee.org.egateway.vit.ac.in/stampPDF/getPDF.jsp?tp=&arnumber=9823806&ref=aHR0cHM6Ly9pZWVleHBsb3JlLmllZWUub3JnL2RvY3VtZW50Lzk4MjM4MDY=
8. https://ieeexplore.ieee.org.egateway.vit.ac.in/stampPDF/getPDF.jsp?tp=&arnumber=8976601&ref=aHR0cHM6Ly9pZWVleHBsb3JlLmllZWUub3JnL2RvY3VtZW50Lzg5NzY2MDE=
9. https://ieeexplore.ieee.org.egateway.vit.ac.in/stamp/stamp.jsp?tp=&arnumber=9456118
10. https://content.ebscohost.com.egateway.vit.ac.in/ContentServer.asp?T=P&P=AN&K=148826411&S=R&D=bth&EbscoContent=dGJyMNXb4kSeqK840dvuOLCmsEqep7ZSrqm4TbKWxWXS&ContentCustomer=dGJyMPGrtVCzqrFNuePfgeyx44Dt6fIA
11. https://data-flair.training/blogs/python-deep-learning-project-handwritten-digit-recognition/

12. https://nanonets.com/blog/handwritten-character-recognition/

13. G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “Emnist: an extension of mnist to handwritten letters,” arXiv preprint arXiv:1702.05373, 2017.

14. L. Eikvil, “Optical character recognition,” citeseer. ist. psu. edu/142042. html, 1993.

15. A. Graves and J. Schmidhuber, “Offline handwriting recognition with multidimensional recurrent neural networks,” in Advances in neural information processing systems, 2009, pp. 545–552.

16. V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, “Dropout improves recurrent neural networks for handwriting recognition,” in Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. IEEE, 2014, pp. 285–290.

17. S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, “Improving offline handwritten text recognition with hybrid hmm/ann models,” IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 4, pp. 767–779, 2011.

18. T. S. Gunawan, A. F. R. M. Noor, and M. Kartiwi, “Development of english handwritten recognition using deep neural network,” 2018.

19. C. Wu, W. Fan, Y. He, J. Sun, and S. Naoi, “Handwritten character recognition by alternately trained relaxation convolutional neural network,” 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 291– 296, 2014.

20. F. Chollet et al., “Keras,” https://github.com/fchollet/keras, 2015.

21. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man´e, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vi´egas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

PLAGIARISM REPORT