**PROJECT REPORT**

**A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION**

**submitted by PNT2022TMID25375**

|  |
| --- |
| Premkumar G |
| Ramani S |
| Simon S |
| Sankar U |
| Stebin S |

**TABLE OF CONTENTS**

1. **INTRODUCTION 4**
   1. [PROJECT OVERVIEW 4](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.gjdgxs)
   2. [PURPOSE](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.30j0zll)  4
2. **LITERATURE SURVEY 5**
   1. [EXISTING PROBLEM 5](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.1fob9te)
   2. [REFERENCES 5](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.3znysh7)
   3. [PROBLEM STATEMENT DEFINITION 9](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.2et92p0)
3. **IDEATION AND PROPOSED SOLUTION 10**
   1. [EMPATHY MAP 10](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.tyjcwt)
   2. [IDEATION & BRAINSTORMING](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.3dy6vkm) 11
   3. [PROPOSED SOLUTION](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.1t3h5sf)  12
   4. [PROBLEM SOLUTION FIT](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.4d34og8) 13
4. **REQUIREMENT ANALYSIS 19**
   1. [FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.2s8eyo1) 19
5. **PROJECT DESIGN 21**
   1. [DATA FLOW DIAGRAM 21](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.3rdcrjn)
   2. [SOLUTION & TECHNICAL ARCHITECTURE](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.26in1rg)  22
   3. [USER STORIES](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.lnxbz9)  24
6. **PROJECT PLANNING AND SCHEDULING 25**
   1. [SPRINT PLANNING AND ESTIMATION 25](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.35nkun2)
   2. [SPRINT DELIVERY SCHEDULE](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.1ksv4uv) 26
   3. REPORT FROM JIRA 27
7. **CODING & SOLUTIONING 28**
   1. FEATURE 1 28
   2. FEATURE 2 29
8. **TESTING 30**
   1. [TEST CASES](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.44sinio) 30
   2. [USER ACCEPTANCE TESTING 34](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.2jxsxqh)
9. **RESULTS 36**
   1. [PERFORMANCE METRICS 36](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.1y810tw)
10. **ADVANTAGES & DISADVANTAGES 37**
11. **CONCLUSION 38**
12. **FUTURE SCOPE 3**
13. [**APPENDIX 28**](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.1ci93xb)

SOURCE CODE 28

GITHUB & [PROJECT DEMO](https://docs.google.com/document/d/1EprNi6aR2EWVb1Pn_LtUSx5PQA6FLJ8r/edit#heading%3Dh.2bn6wsx) LINK 37

**CHAPTER 1 INTRODUCTION**

1. **PROJECT OVERVIEW**

Machine learning and deep learning play an important role in computer technology and artiﬁcial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

Handwritten Digit Recognition is the ability of computer systems to recognise handwritten digits from various sources, such as images, documents, and so on. This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

1. **PURPOSE**

Handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

Digit recognition systems are capable of recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms ﬁlled up by hand (tax forms) and so on.

**CHAPTER 2 LITERATURE SURVEY**

1. **EXISTING PROBLEM**

The issue is that there’s a wide range of handwriting – good and bad. This makes it tricky for programmers to provide enough examples of how every character might look. Sometimes, characters look very similar, making it hard for a computer to recognise accurately.

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting inﬂuence the structure and appearance of the digits.

1. **REFERENCES**

**Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers(2019)**

# F. Siddique, S. Sakib and M. A. B. Siddique

The increase of Artificial Neural Network (ANN), deep learning has brought a dramatic twist in the field of machine learning by making it more artificially intelligent. Deep learning is remarkably used in vast ranges of fields because of its diverse range of applications such as surveillance, health, medicine, sports, robotics, drones, etc. In deep learning, Convolutional Neural Network (CNN) is at the center of spectacular advances that mixes Artificial Neural Network (ANN) and up to date deep learning strategies. It has been used broadly in pattern recognition, sentence classification, speech

recognition, face recognition, text categorization, document analysis, scene, and handwritten digit recognition. The goal of this paper is to observe the variation of accuracies of CNN to classify handwritten digits using various numbers of hidden layers and epochs and to make the comparison between the accuracies. For this performance evaluation of CNN, we performed our experiment using Modified National Institute of Standards and Technology (MNIST) dataset. Further, the network is trained using

stochastic gradient descent and the back-propagation algorithm.

**An Efﬁcient And Improved Scheme For Handwritten Digit Recognition Based**

**On Convolutional Neural Network (2019)**

# Ali, Saqib and Shaukat, Zeeshan and Azeem, Muhammad and Sakhawat, Zareen and Mahmood, Tariq and others

This study uses rectiﬁed linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deeplearning4j (DL4J) architecture to recognize handwritten digits. The proposed CNN framework has all the necessary parameters for a high level of MNIST digit classiﬁcation accuracy. The system's training takes into account the time factor as well. The system is also tested by altering the number of CNN layers for additional accuracy veriﬁcation. It is important to note that the CNN architecture consists of two convolutional layers, the ﬁrst with 32 ﬁlters and a 5x5 window size and the second with 64 ﬁlters and a 7x7 window size. In comparison to earlier proposed systems, the experimental ﬁndings show that the proposed CNN architecture for the MNIST dataset demonstrates great performance in terms of time and accuracy. As a result, handwritten numbers are detected with a recognition rate of 99.89% and high precision (99.21%) in a short amount of time.

**Handwritten Digit Recognition using Convolutional Neural Network in Python with Tensorflow and Observe the Variation of Accuracies for Various Hidden Layers (2019)**

Fathma Siddique, Shadman Sakib, Md. Abu Bakr Siddique

In recent times, with the increase of Artificial Neural Network (ANN), deep learning has brought a dramatic twist in the field of machine learning by making it more Artificial Intelligence (AI). Deep learning is remarkably used in vast ranges of fields because of its diverse range of applications such as surveillance, health, medicine, sports, robotics, drones etc. In deep learning, Convolutional Neural Network (CNN) is at the center of spectacular advances that mixes Artificial Neural Network (ANN) and up to date deep learning strategies. It has been used broadly in pattern recognition, sentence classification, speech recognition, face recognition, text categorization, document analysis, scene, and handwritten digit recognition. The goal of this paper is to observe the variation of accuracies of CNN to classify handwritten digits using various numbers of hidden layers and epochs and to make the comparison between the accuracies. For this performance evaluation of CNN, we performed our experiment using the Modified National Institute of Standards and Technology (MNIST) dataset. Further, the network is trained using stochastic gradient descent and the backpropagation algorithm.

**Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021)**

# Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. To determine which model was the most accurate, they compared them based on their individual properties.

Support vector machines are among the simplest classiﬁers, making them faster than other algorithms and providing the highest training accuracy rate in this situation. However, due to their simplicity, SVMs cannot categorize

complicated and ambiguous images as accurately as MLP and CNN algorithms can. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective

solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the conﬁguration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over- ﬁtting the dataset and provides biased predictions.

**An Enhanced Handwritten Digit Recognition Using Convolutional Neural Network(2021)**

# M. S, C. N. Vanitha, N. Narayan, R. Kumar and G. R

Handwritten digit recognition has a great impact in the applications of deep learning. Convolutional Neural Network in deep learning has become one of the major methods and one of the important factors in the various success in recent times and deep learning is used majorly in the area of object recognition. In the paper work, the speech output feature is integrated along with the text output. Convolutional Neural Network model is applied in the image classification. The dataset used to train and test is the MNIST dataset. There are various applications of handwritten digit recognition in real time. It is applied in detection of vehicle number, reading of bank cheques, the arrangement of letters in the post office.

1. **PROBLEM STATEMENT DEFINITION**

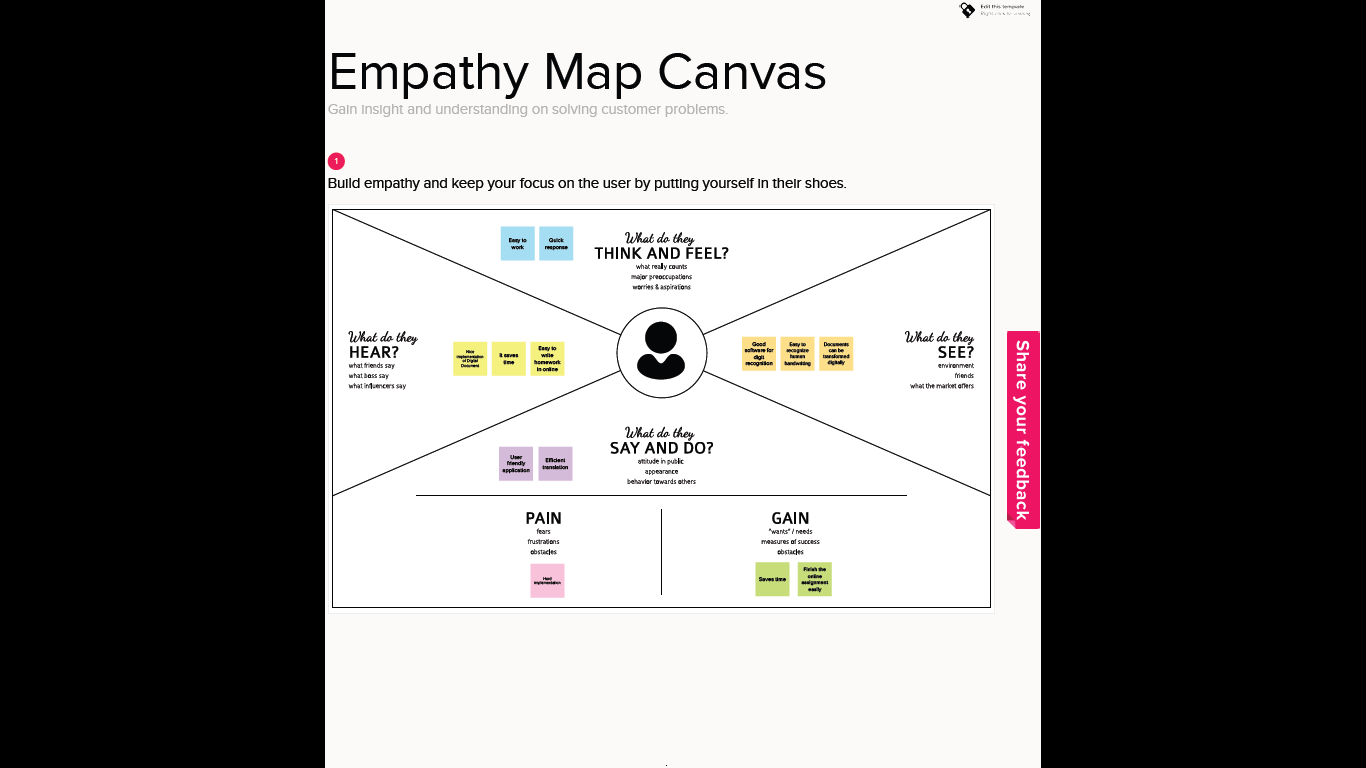
Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitized to reduce

human effort. Hence, there comes a need for handwritten digit recognition in many real- time applications. The MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. This image is analyzed by the model and the detected result is returned to the UI. [MNIST](http://yann.lecun.com/exdb/mnist/) (“Modified National Institute of Standards and Technology”) is considered an unofficial computer vision “hello-world” dataset. This is a collection of thousands of handwritten pictures used to train classification models using Machine Learning techniques.

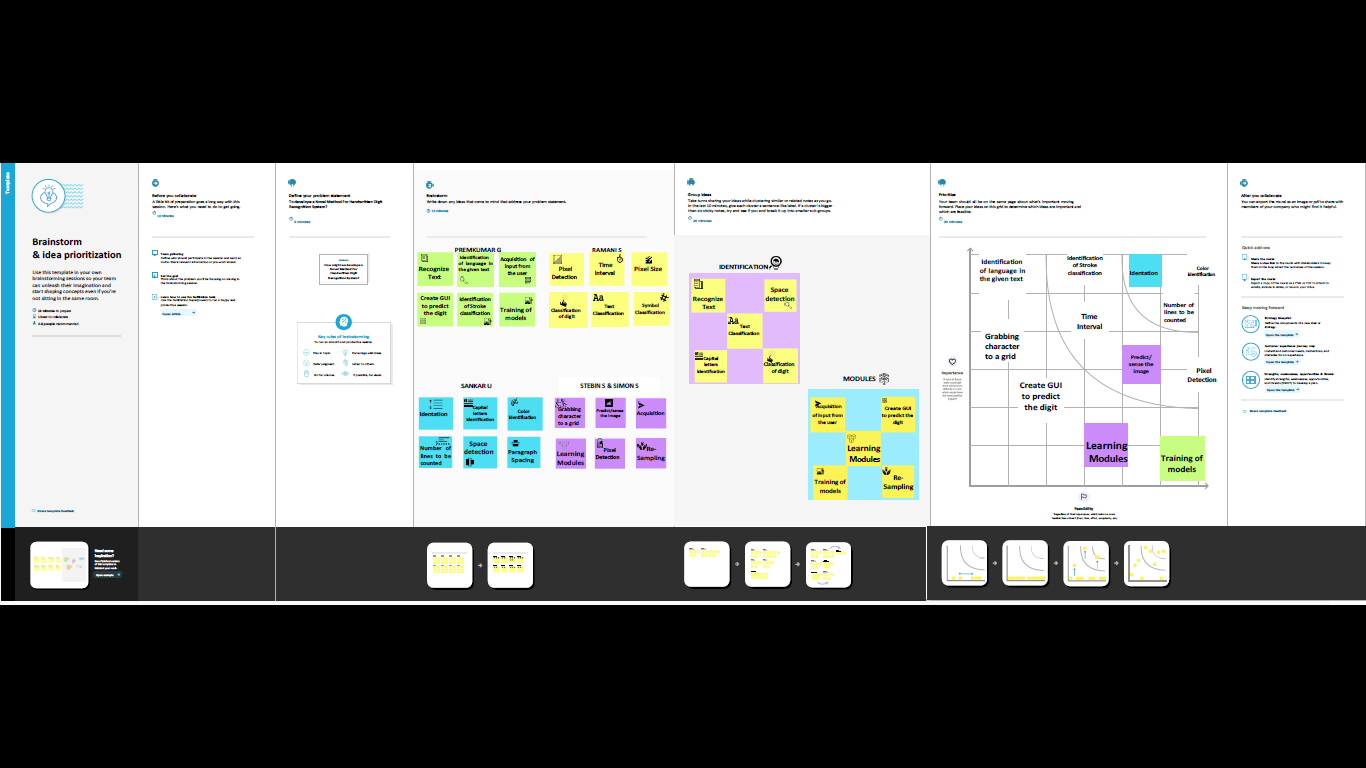
**CHAPTER 3**

**IDEATION AND PROPOSED SOLUTION**

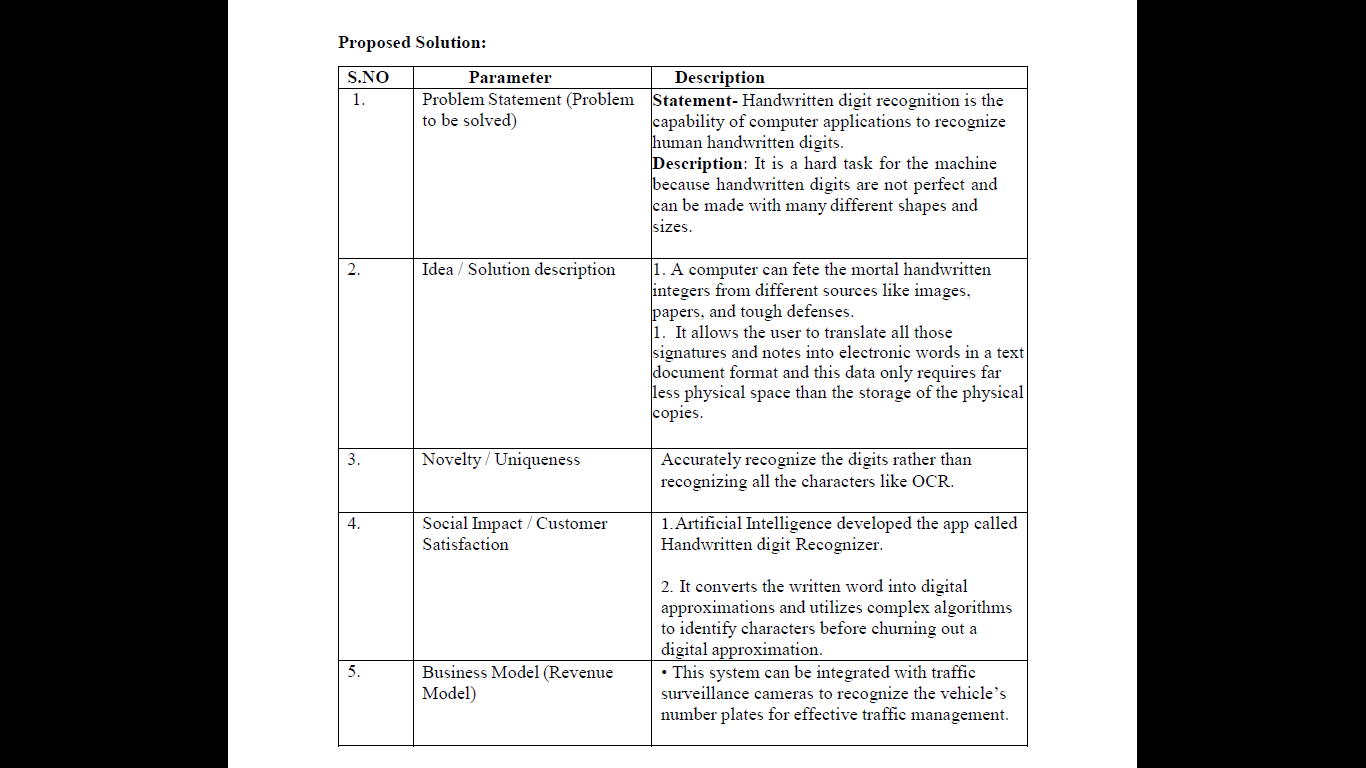
1. **EMPATHY MAP**

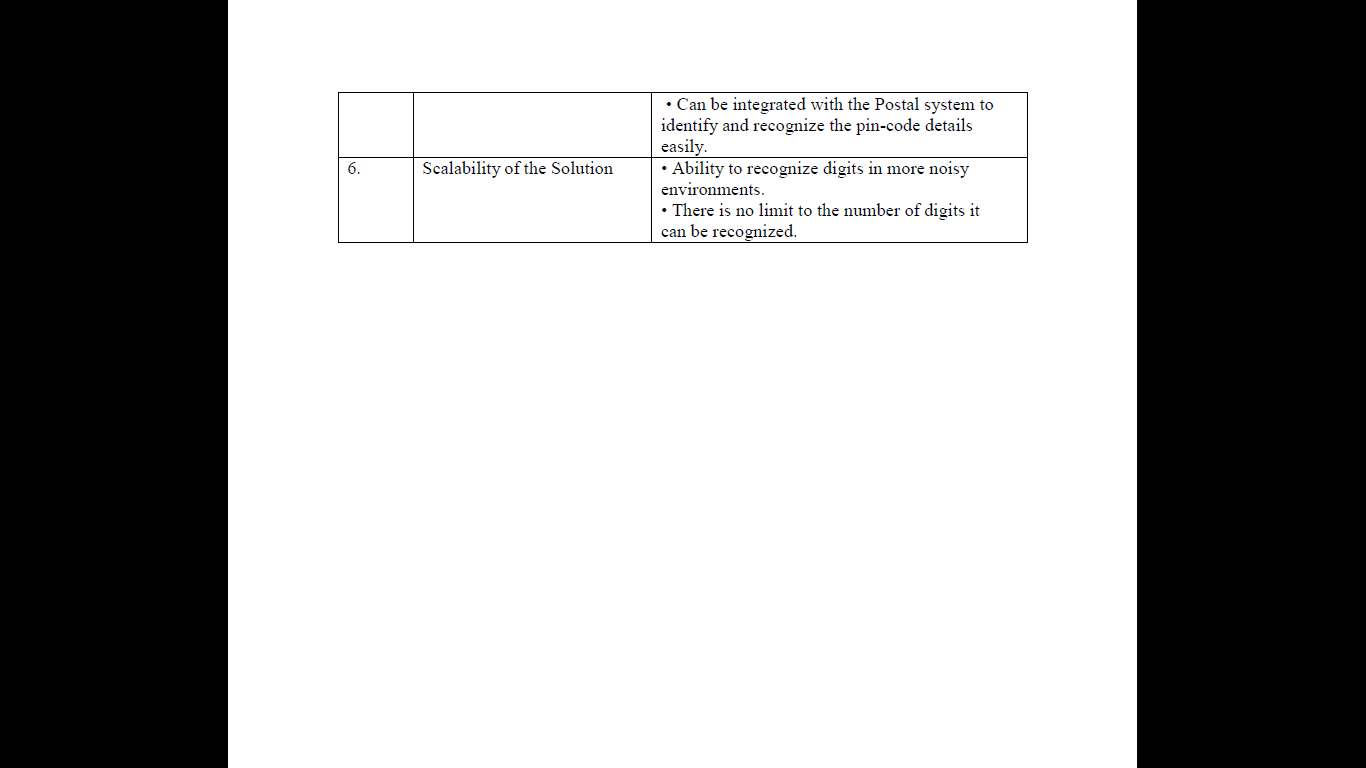
****

1. **IDEATION & BRAINSTORMING**

****

1. **PROPOSED SOLUTION**

****



1. **PROBLEM SOLUTION FIT**

**MNIST Dataset Description**

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real- time applications. The MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI

The [MNIST Handwritten Digit Recognition Dataset](http://yann.lecun.com/exdb/mnist/) contains 60,000 training and 10,000 testing labeled handwritten digit pictures.

Each picture is 28 pixels in height and 28 pixels wide, for a total of 784 (28×28) pixels. Each pixel has a single pixel value associated with it. It indicates how bright or dark that pixel is (larger numbers indicate darker pixels). This pixel value is an integer ranging from 0 to 255.

**PROCEDURE**

* + Install the latest TensorFlow library.
  + Prepare the dataset for the model.
  + Develop Single Layer Perceptron model for classifying the handwritten digits.
  + Plot the change in accuracy per epochs.
  + Evaluate the model on the testing data.
  + Analyze the model summary.
  + Add a hidden layer to the model to make it a Multi-Layer Perceptron.
  + Add Dropout to prevent overfitting and check its effect on accuracy.
  + Increasing the number of Hidden Layer neurons and checking its effect on accuracy.
  + Use different optimizers and check its effect on accuracy.
  + Increase the hidden layers and check its effect on accuracy.
  + Manipulate the batch size and epochs and check its effect on accuracy.

MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consists of 60,000 training images and 10,000 test images. The artificial neural networks can almost mimic the human brain and are a key ingredient in the image processing field.Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits. We have taken this a step further where our handwritten digit recognition system not only detects scanned images of handwritten digits but also allows writing digits on the screen with the help of an integrated GUI for recognition.

**APPROACH:**

We will approach this project by using a three-layered Neural Network.

* + - The input layer: It distributes the features of our examples to the next layer for calculation of activations of the next layer.
    - The hidden layer: They are made of hidden units called activations providing nonlinear ties for the network. A number of hidden layers can vary according to our requirements.
    - The output layer: The nodes here are called output units. It provides us with the final prediction of the Neural Network on the basis of which final predictions can be made.

A neural network is a model inspired by how the brain works. It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters in a set of data which could help to recognize the underlying relationships. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

**METHODOLOGY:**

We have implemented a Neural Network with 1 hidden layer having 100 activation units (excluding bias units). The data is loaded from a .mat file, features(X) and labels(y) were extracted. Then features are divided by 255 to rescale them into a range of [0,1] to avoid overflow during computation. Data is split up into 60,000 training and 10,000 testing examples. Feedforward is performed with the training set for calculating the hypothesis and then backpropagation is done in order to reduce the error between the layers. The regularization parameter lambda is set to 0.1 to address the problem of overfitting. Optimizer is run for 70 iterations to find the best fit model

**ALGORITHM:**

**Forward Propagation Architecture:**

It is a small workflow of how CNN module will extract the features and classify the image based on it. The architecture shows the input layer, hidden layers and output layer of the

network. There are many layers involved in the feature extraction phase of the network which involves convolution and subsampling .

**EXPLANATION OF THE PROPOSED SYSTEM**

* The first layer of the architecture is the User layer. User layer will comprise of the people who interacts with the app and for the required results.
* The next three layers is the frontend architecture of the application.

The application will be developed using which is the open-source platform for HTML, CSS and JavaScript. The application is deployed in the localhost which is shown on the browser. Through the app, the user will be able to upload pictures of the handwritten digits and convert it into the digitalized form. • The one in between the database and view layer is the business layer which is the logical calculations on the basis of the request from the client side. It also has a service interface. • The backend layer consists of two datasets: Training Data and Test Data. The MNIST database has been used for that which is already divided into a training set of 60,000 examples and a test of 10,000 examples. • The training algorithm used is Convolution Neural Network. This will prepare the trained model which will be used to classify the digits present in the test data. Thus, we can classify the digits present in the images as: Class 0,1,2,3,4,5,6,7,8,9.

**WORKING:**

* Neural Networks receive an input and transform it through a series of hidden layers.
* Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer.
* Neurons in a single layer function completely independently. • The last fully connected layer is called the "output layer“.

**Convolution Layer:** The Convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume.

During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2- dimensional activation map of that filter.

As a result, the network learns filters that activate when they see some specific type of feature at some spatial position in the input..

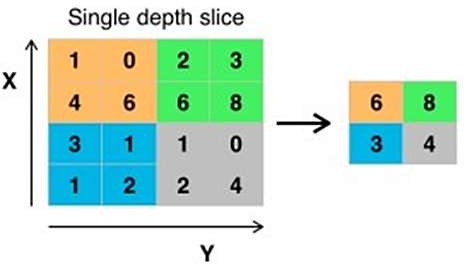
**Feature Extraction:**

All neurons in a feature share the same weights .In this way all neurons detect the same feature at different positions in the input image. Reduce the number of free parameters.

**Subsampling Layer:** Subsampling, or down sampling, refers to reducing the overall size of a signal .The subsampling layers reduce the spatial resolution of each feature map. Reduce the effect of noises and shift or distortion invariance is achieved.

**Pooling layer:** It is common to periodically insert a Pooling layer in-between successive Conv layer in a Convent architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.

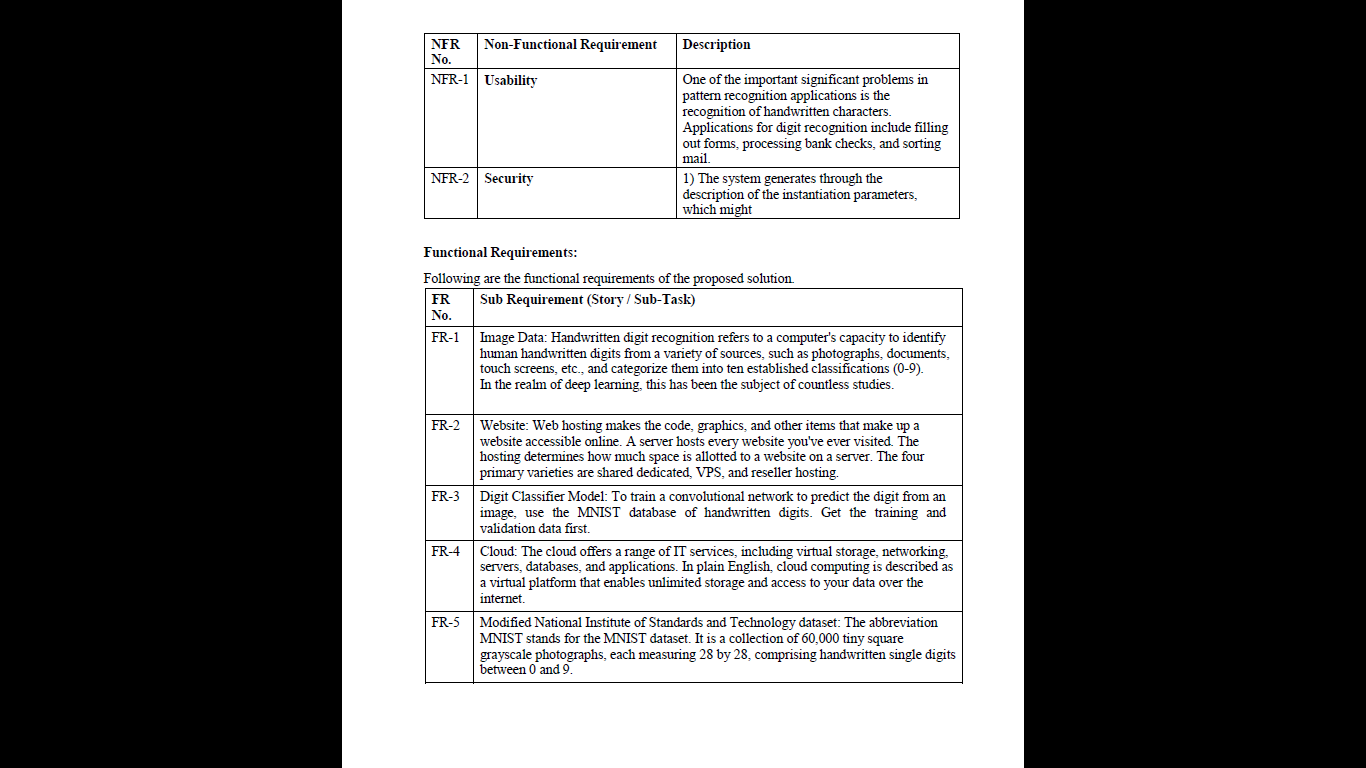
**TensorFlow:** TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop,

mobile, web, and cloud. See the sections below to get started. By scanning the numerical digit and convert into png format using python3 command in terminal we can get text output and sound output.

**Pooling layer**

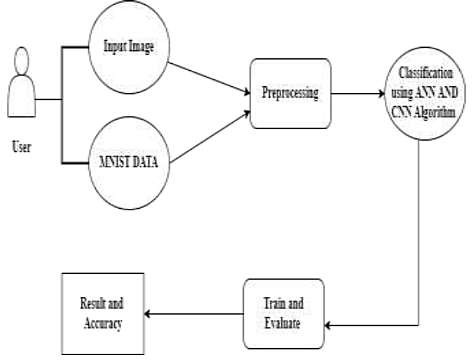
**CHAPTER 4 REQUIREMENT ANALYSIS**

1. **FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

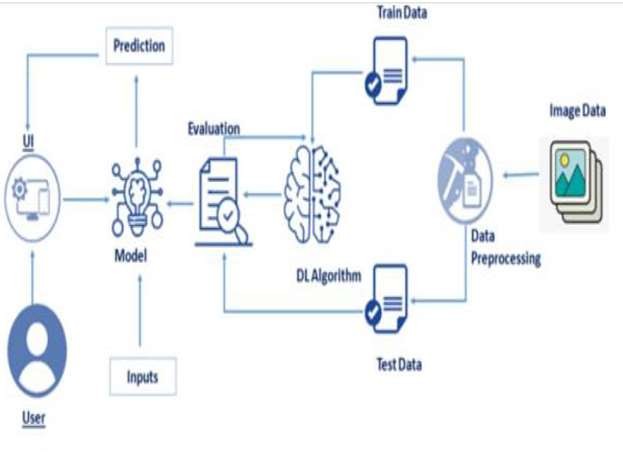
****

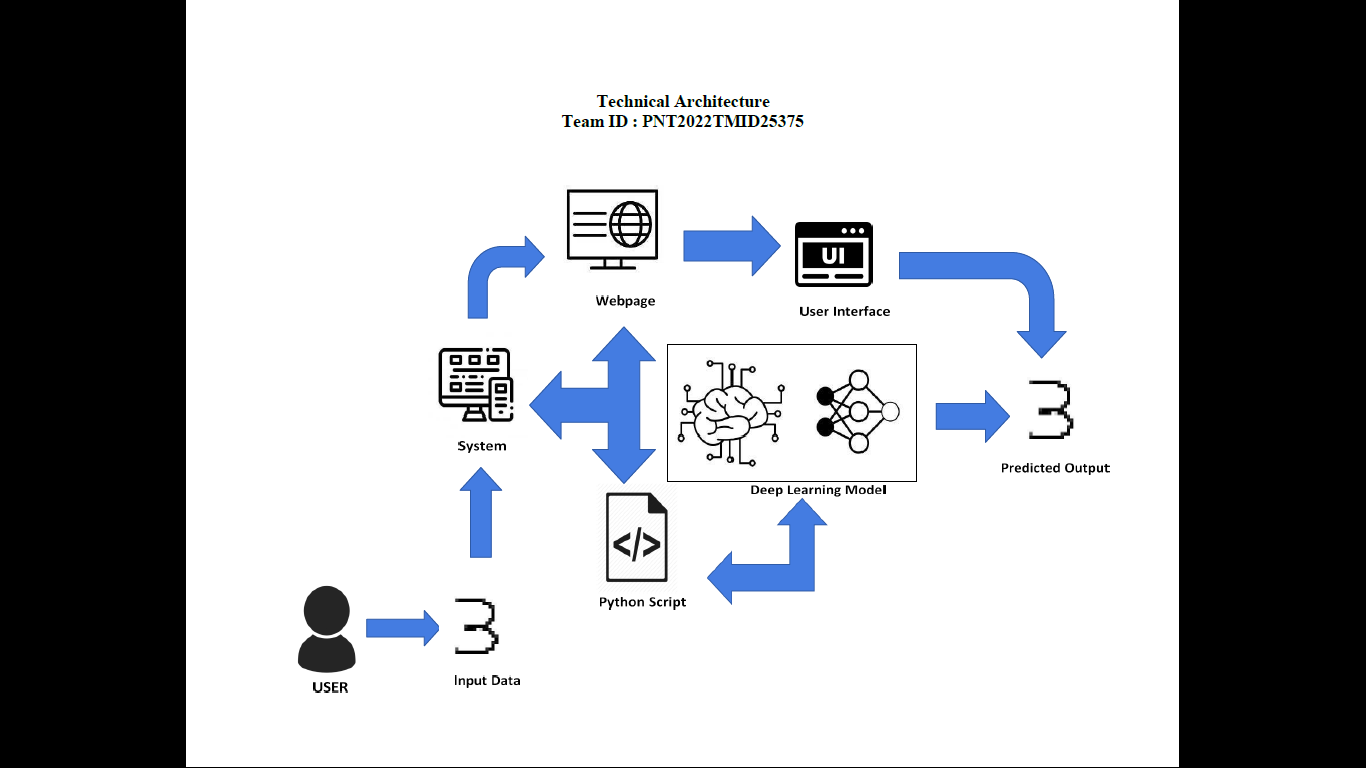
**CHAPTER 5 PROJECT DESIGN**

1. **DATA FLOW DIAGRAM**

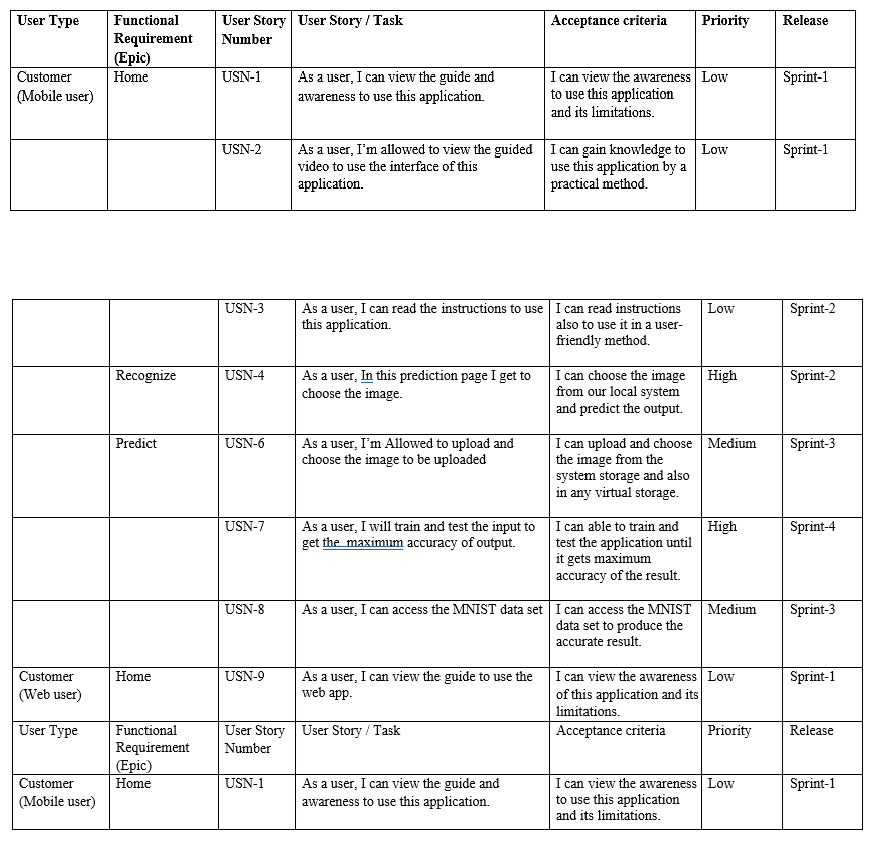


1. **SOLUTION & TECHNICAL ARCHITECTURE**





1. **USER STORIES**

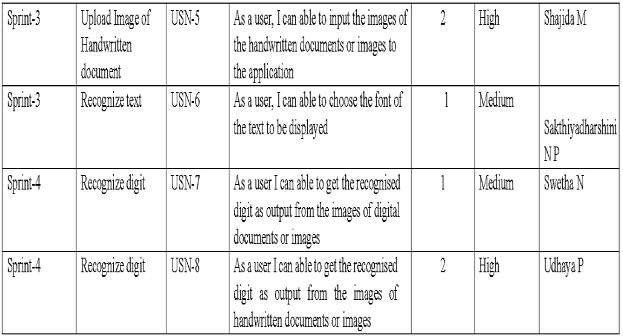


**CHAPTER 6**

**PROJECT PLANNING AND SCHEDULING**

1. **SPRINT PLANNING AND ESTIMATION**





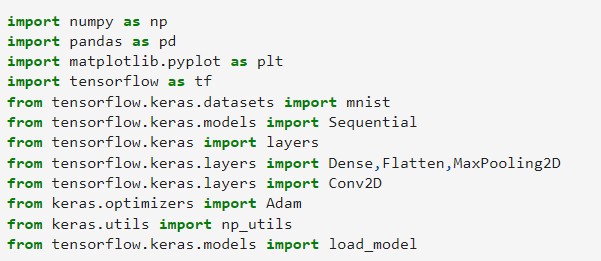
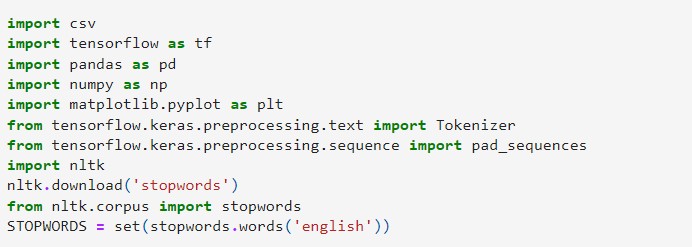
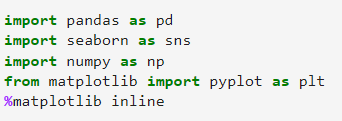
1. **SPRINT DELIVERY SCHEDULE**

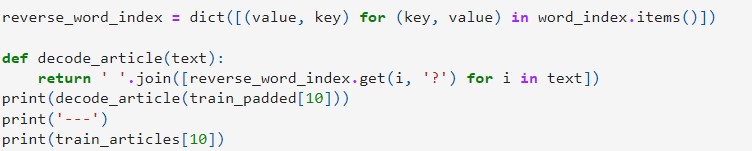


**CHAPTER 7**

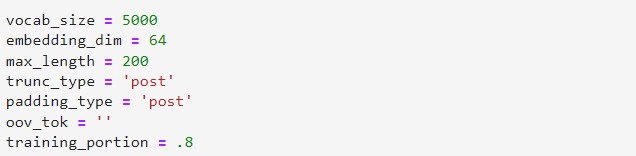
**CODING & SOLUTIONING**

**FEATURE 1**

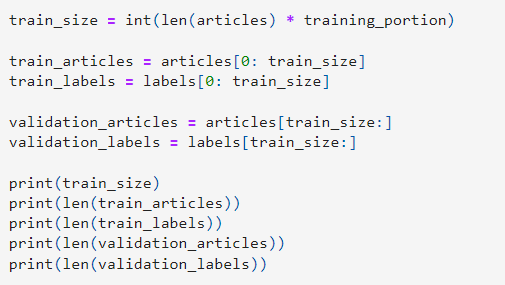


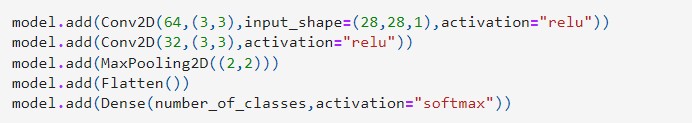
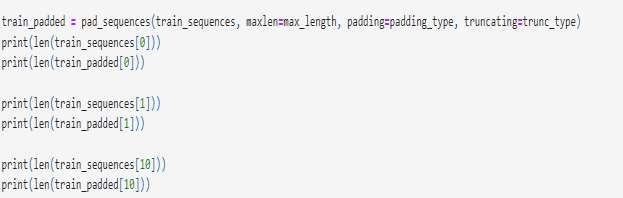


**FEATURE 2**









**CHAPTER 8 TESTING**

* 1. **TEST CASES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **st case ID** | **Feature Type** | **Componen t** | **Test**  **Scenari o** | **Expected Result** | **Actual Result** | **Statu s** |
| HP\_TC\_00 1 | UI | Home Page | Verify UI elements in the Home Page | The Home page must be displayed properly | Working as expected | PASS |
| HP\_TC\_00 2 | UI | Home Page | Check if the UI elements are displayed properly in different screen sizes | The Home page must be displayed properly in all sizes | The UI is not displayed properly in screen size 2353 x 1651  and 758 x  630 | FAIL |
| HP\_TC\_00 3 | Functiona l | Home Page | Check if user can upload their ﬁle | The input image should be uploaded to the application successfully | Working as expected | PASS |
| HP\_TC\_00 4 | Functiona l | Home Page | Check if user cannot upload unsupported ﬁles | The application should not allow user to select a non image ﬁle | User is able to upload any ﬁle | FAIL |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HP\_TC\_00 5 | Functiona l | Home Page | Check if the page redirects to the result page once the input is given | The page should redirect to the results page | Working as expected | PASS |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| M\_TC\_001 | Functional | Model | Check if the model can handle various image sizes | The model should rescale the image and predict the results | Working as expected | PASS |
| M\_TC\_002 | Functional | Model | Check if the model predicts the digit | The model should predict the number | Working as expected | PASS |
| M\_TC\_003 | Functional | Model | Check if the model can handle complex input image | The model should predict the number in the complex image | The model fails to identify the digit since the model is not built to handle such data | FAIL |
| AC\_TC\_001 | Functional | Accur acy | check if the model can provide the output with accuracy | The model should predict the image with accuracy from  the dataset. | working as expected | PASS |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RP\_TC\_00 1 | UI | Result Page | Verify UI elements in the Result Page | The Result page must be displayed properly | Working as expected | PASS |
| RP\_TC\_00 2 | UI | Result Page | Check if the input image is displayed properly | The input image should be displayed properly | The size of the input image exceeds the display container | FAIL |
| RP\_TC\_00 3 | UI | Result Page | Check if the result is displayed properly | The result should be displayed properly | Working as expected | PASS |
| RP\_TC\_00 4 | UI | Result Page | Check if the other predictions are displayed properly | The other predictions should be displayed properly | Working as expected | PASS |

* 1. **USER ACCEPTANCE TESTING**
     1. **DEFECT ANALYSIS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Total |
| By Design | 1 | 0 | 1 | 0 | 2 |
| Duplicate | 0 | 0 | 0 | 0 | 0 |
| External | 0 | 0 | 2 | 0 | 2 |
| Fixed | 4 | 1 | 0 | 1 | 6 |

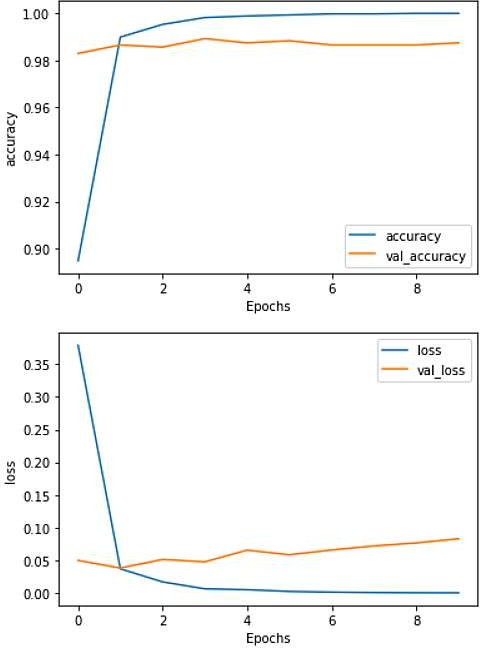
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Not Reproduced | 0 | 0 | 0 | 1 | 1 |
| Skipped | 0 | 0 | 0 | 1 | 1 |
| Won’t Fix | 1 | 0 | 1 | 0 | 2 |
| Total | 6 | 1 | 4 | 3 | 14 |

* + 1. **TEST CASE ANALYSIS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section | Total Cases | Not Tested | Fail | Pass |
| Client Application | 10 | 0 | 3 | 7 |
| Security | 2 | 0 | 1 | 1 |
| Performance | 3 | 0 | 1 | 2 |
| Exception Reporting | 2 | 0 | 0 | 2 |

**CHAPTER 9 RESULTS**

**1. PERFORMANCE METRICS**



**CHAPTER 10 ADVANTAGES & DISADVANTAGES**

**ADVANTAGES**

* Reduces manual work
* Backups
* More accurate than average human
* Capable of handling a lot of data
* Can be used anywhere from any device

**DISADVANTAGES**

* Cannot handle complex data
* Low retention
* All the data must be in digital format
* Requires a high performance server for faster predictions
* Prone to occasional errors

**CHAPTER 11**

**CONCLUSION**

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in

real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms ﬁlled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

**CHAPTER 12 FUTURE SCOPE**

This project is far from complete and there is a lot of room for improvement.

Some of the improvements that can be made to this project are as follows:

* Add support to detect from digits multiple images and save the results
* Add support to detect multiple digits
* Improve model to detect digits from complex images
* Add support to different languages to help users from all over the world

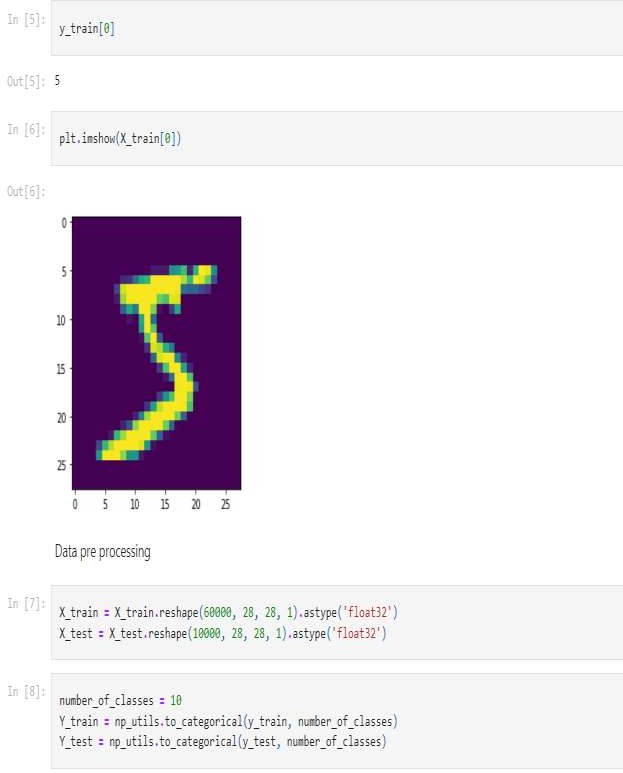
This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will beneﬁt several industries and reduce the workload on many workers, enhancing overall work efﬁciency.

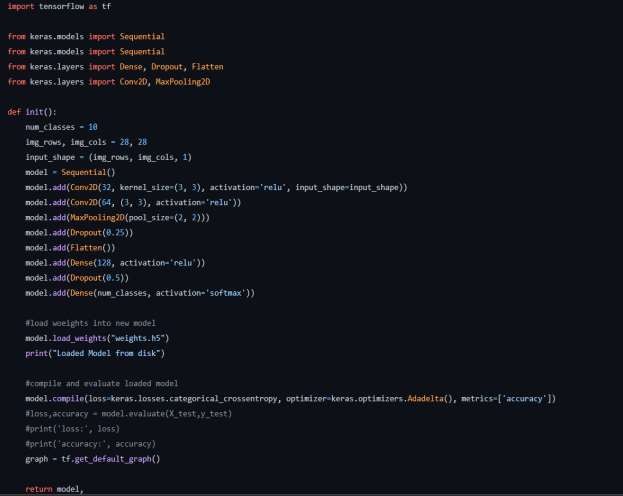
**APPENDIX**

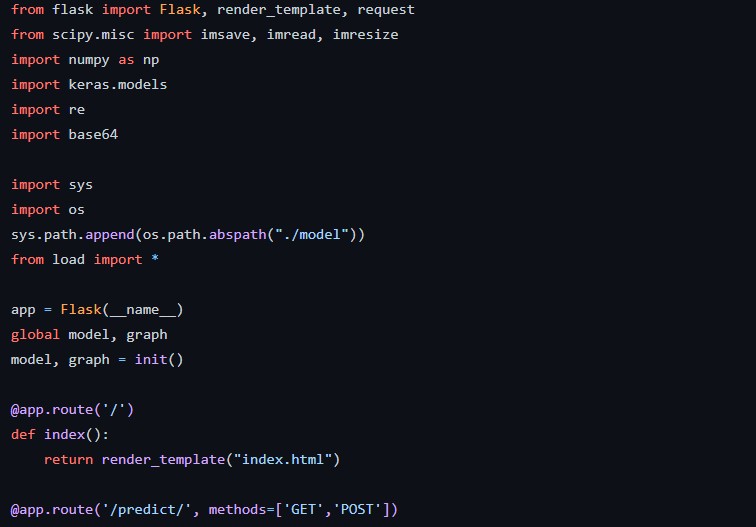
**SOURCE CODE**

MODEL CREATION





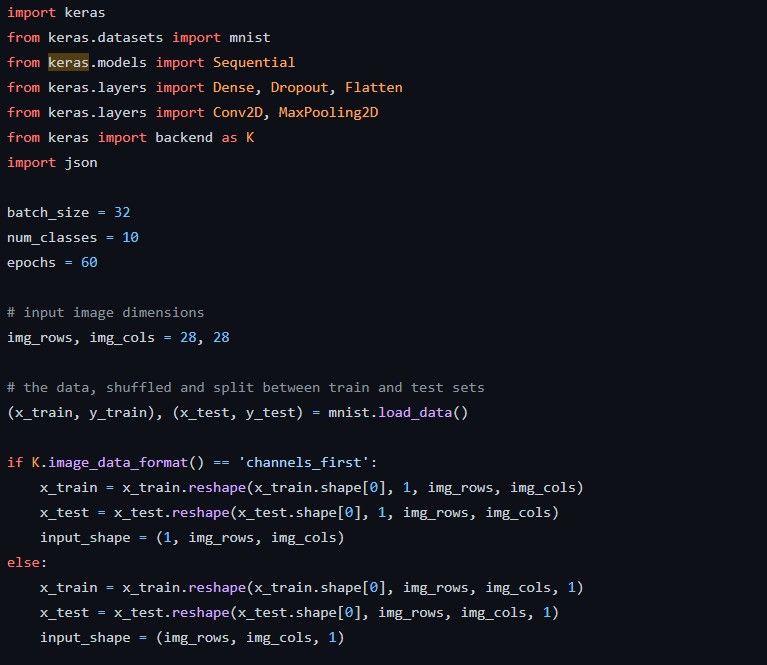


FLASK APP

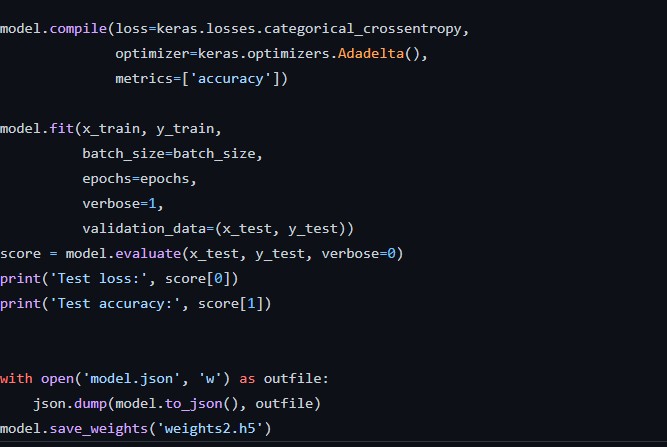
HOME PAGE (HTML)



TRAIN THE MODEL







PREDICT PAGE (HTML)



**GITHUB**

https://github.com/IBM-EPBL/IBM-Project-42867-1660710548

**PROJECT DEMO**