**A NOVEL METHOD FOR HAND WRITTEN DIGIT RECOGNITION**

**PROJECT REPORT**

*Submitted by*

**MUNIRAJ.M (612819104022)**

**RAMU.M (612819104032)**

**SENTHAMILSELVAN.S (612819104043)**

**SENTHIL.S (612819104044)**

***in partial fullfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**TABLE OF CONTENTS**

1. **INTRODUCTION** 
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING** 
   1. Feature 1
   2. Feature 2
8. **TESTING** 
   1. Test Cases
   2. User Acceptance Testing
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

13.1 Source Code

13.2 GitHub & Project Demo Link

**INTRODUCTION**

**1.1 PROJECT OVERVIEW**

Machine learning and deep learning plays 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 many more areas.

This article presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset, comparing classifiers like KNN, PSVM, NN and convolution neural network on basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with the classifiers.

To make machines more intelligent, the developers are diving into machine learning and deep learning techniques. A human learns to perform a task by practicing and repeating it again and again so that it memorizes how to perform the tasks. Then the neurons in his brain automatically trigger and they can quickly perform the task they have learned. Deep learning is also very similar to this. It uses different types of neural network architectures for different types of problems.

For example – object recognition, image and sound classification, object detection, image segmentation, etc. The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because 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 System:**

Digit recognition system is the working of a machine to train itself or 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 , numeric entries in forms filled up by hand and so on.

**1.2 PURPOSE**

The goal of this project is to create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual’s handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.

**LITERATURE SURVEY**

Anuj Dutt in his paper demonstrated that utilizing Deep Learning systems, he had the capacity to get an extremely high measure of accuracy. By utilizing the convolutional Neural Network with Keras and Theano as backend, he was getting a accuracy of 98.72%. In addition, execution of CNN utilizing Tensorflow gives a stunningly better consequence of 99.70%. Despite the fact that the complication of the procedure and codes appears to be more when contrasted with typical Machine Learning algorithms yet the accuracy he got is increasingly obvious. In a paper published by Saeed AL-Mansoori, Multilayer Perceptron (MLP) Neural Network was implemented to recognize and predict handwritten digits from 0 to 9.The proposed neural system was trained and tested on a dataset achieved from MNIST.

**2.1 EXISTING SYSTEM**

These days, an ever-increasing number of individuals use pictures to transmit data. It is additionally main stream to separate critical data from pictures. Image Recognition is an imperative research area for its generally used applications. In general, the field of pattern recognition, one of the difficult undertakings is the precise computerized recognition of human handwriting. Without a doubt, this is a very difficult issue because there is an extensive diversity in handwriting from an individual to another individual. In spite of the fact that, this difference does not make any issues to people, yet, anyway it is increasingly hard to instruct computers to interpret general handwriting. For the image recognition issue, for example, handwritten classification, it is essential to make out how information is depicted onto images.

Handwritten Recognition from the MNIST dataset is well known among scientists as by utilizing different classifiers for various parameters, the error rate has been decreased, for example, from linear classifier (1-layer NN) with 12% to 0.23% by a board of 35 convolution neural systems. The scope of this is to implement a Handwritten Digit Recognition framework and think about the diverse classifiers and different techniques by concentrating on how to accomplish close to human performance. For an undertaking of composing diverse digits (0-9) for various people the general issue confronted would be of digit order issue and the closeness between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 and so forth. Additionally, individuals compose a similar digit from various perspectives, the uniqueness and assortment in the handwriting of various people likewise impact the development and presence of the digits.

**2.2 REFERENCES**

[1]. M. Wu and Z. Zhang, Handwritten Digit Classification using the MNIST Dataset, 2010.

[2]. A. Dutta and A. Dutta, Handwritten digit recognition using deep learning, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 7, July 2017.

[3]. Al Maadeed, Somaya, and Abdelaali Hassaine, Automatic prediction of age, gender, and nationality in offline handwriting. EURASIP Journal on Image and Video Processing, no. 1 2014. [4]. Gaurav Jain, Jason Ko, Handwritten DigitsRecognition, Project Report, University of Toronto, 11/21/2008.

[5]. Hamid, Norhidayu Abdul, and NilamNur Amir Sjarif,Handwritten recognition using SVM, KNN and neural network, arXiv preprint arXiv:1702.00723 (2017).

[6]. R.G.Mihalyi, Handwritten digit classification using support vector machines, 2011.

[7]. Z. Dan, C. Xu, The Recognition of Handwritten Digits Based on BP Neural Networks and the Implementation on Android, In: 3rd International Conference on Intelligent System Design and Engineering Applications, pp. 1498-1509, 2013.

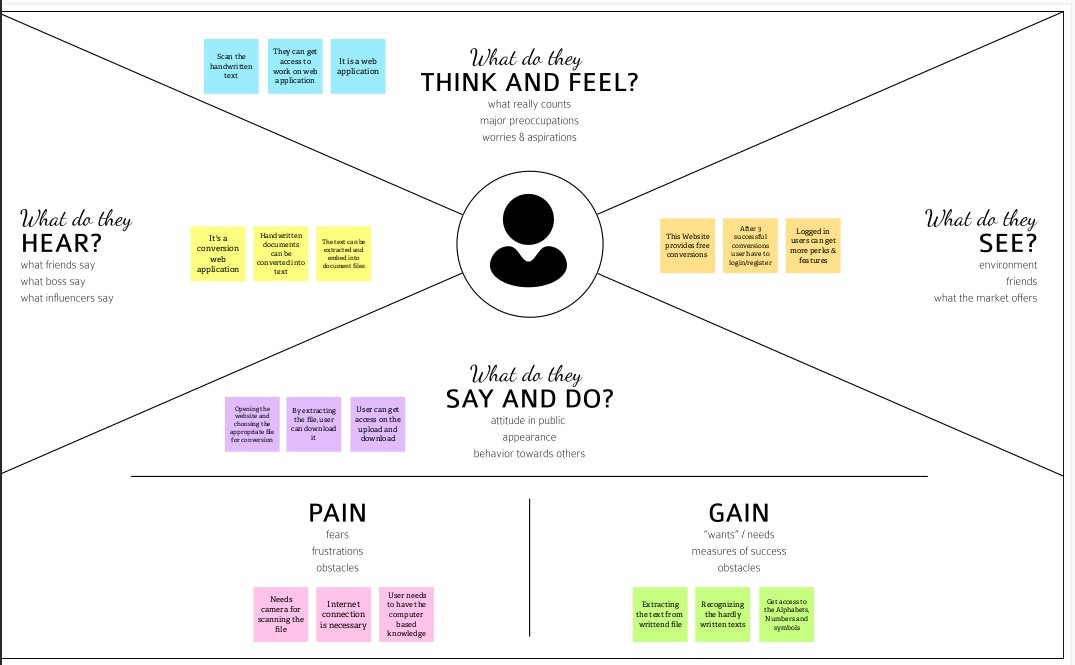
[8]. http://cvisioncentral.com/resourceswall/?resource.

**2.3 PROBLEM STATEMENT DEFINITION**

The aim of this project is to implement a classification algorithm to recognize the handwritten digits. The after effects of probably the most broadly utilized Machine Learning Algorithms like SVM, KNN and RFC and with Deep Learning calculation like multilayer CNN utilizing Keras with Theano and Tensorflow. Utilizing these, the accuracy of 98.70% utilizing CNN (Keras + Theano) when contrasted with 97.91% utilizing SVM, 96.67% utilizing KNN, 96.89% utilizing RFC was obtained.

**IDEATION & PROPOSED SOLUTION**

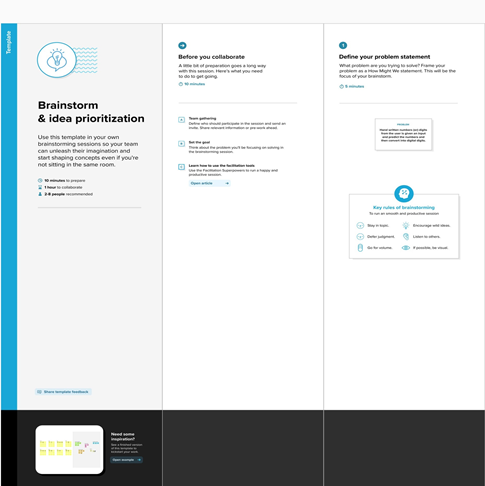
**3.1 Empathy Map Canvas**



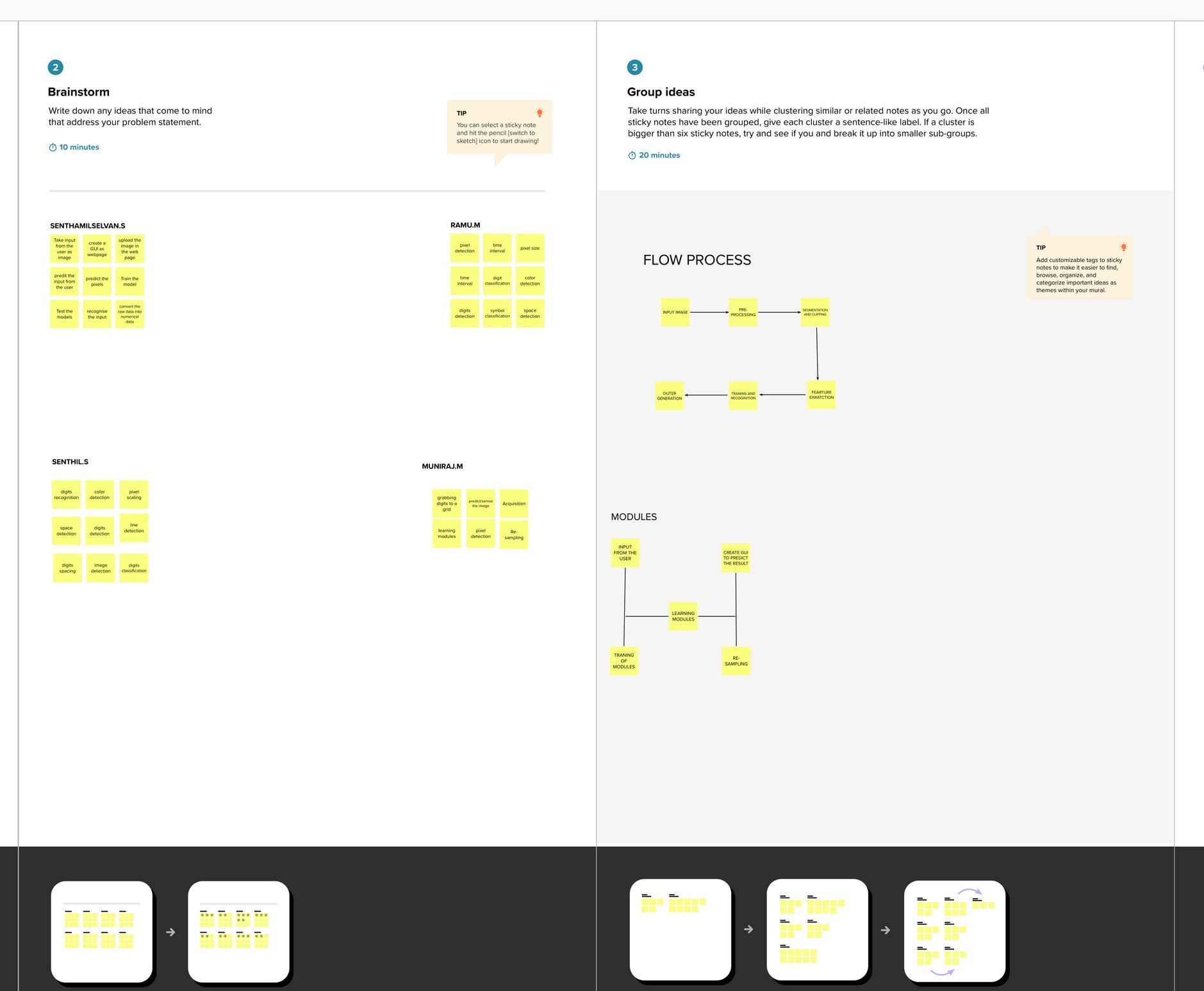
* An empathy map is a template that organizes a user’s behaviours and feelings to create a sense of empathy between the user and your team. The empathy map represents a principal user and helps teams better understand their motivations, concerns, and user experience.
* Empathy mapping is a simple yet effective [workshop](https://frescopad.com/virtual-workshop-definition/) that can be conducted with a variety of different users in mind, anywhere from stakeholders, individual use cases, or entire teams of people. It can be conducted by many different teams such as design teams, sales, product development or [customer service](https://www.questionpro.com/blog/customer-service/). Essentially, it is an exercise that seeks to get inside the head of the customer as they interact with your product/service.
* There are **four quadrants to a traditional empathy map**. These are: Does, Thinks, Says, and Feels. These quadrants will all ask unique questions about how you can analyse the perspective of the user and what they accomplish in their daily use.
* Based on that we define the Problem statement as template by the use of empathy map.

**3.2**  **IDEATION & BRAINSTORMING**

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

****

# STEP-2: BRAINSTORM, IDEA LISTING AND GROUPING



# Step-3: Idea Prioritization



* 1. **Proposed Solution**
* Problem Statement (Problem to be Computer programmes' ability to detect solved) human-written numbers is known as handwritten digit recognition. Because handwritten figures are not always accurate and can take many various forms and sizes, it is a difficult work for the machine.
* Idea / Solution description Using data from various sources, including images, documents, and touch defences, a computer is able to celebrate the mortal handwritten numbers. It permits users to convert all of their handwritten notes and signatures into text documents in electronic form, using much less physical space than would be needed to store the physical copies of those documents.
* Novelty / Uniqueness Recognize the digits precisely rather than all the characters like OCR.
* Social Impact / Customer The Handwritten Digit Recognizer software Satisfaction was made using artificial intelligence. It approximates the printed word digitally by identifying letters using sophisticated algorithms before producing a digital approximation.
* Business Model (Revenue Model) For efficient traffic control, this technology can be connected with traffic surveillance cameras to read licence plates. Pin-code details can be easily identified and recognised by integrating with the postal system.
* Scalability of the Solution The capacity to recognise numbers in more distracting circumstances. The maximum number of digits that can be recognised is unlimited.

**3.4 Problem Solution Fit**

1. CUSTOMER SEGMENT(S) -A officer in a post office receiving letters and couriers in a written format.
2. JOBS-TO-BE-DONE / PROBLEMS -He/She Wants To Store The Pincode Or Mobile Number Etc… Into A Storage Space . So A Hand Written Digit Recognition System Is Needed To Solve Those Problems.
3. TRIGGERS - To do the work in a efficient manner. So that the officer getting satisfied.
4. EMOTIONS: BEFORE / AFTER- BEFORE: They eagerly wants to finish is work quickly and easily .AFTER: If its working fine they feels better
5. AVAILABLE SOLUTIONS - Various people’s handwriting should be used for train the AI. That should improve the accuracy. Training the model in a proper way to get the better outcome.
6. Customer Constraints- it is a difficult because most person’s handwriting will not similar, scanner or camera work perfect in perfect light condition, It took to much time to process.
7. BEHAVIOUR -Before Processing The Image Application Should Verify The Photo Was Taken In Correct Angle And Correct Lighting. User Should Completely Aware Of Instruction Of Application.
8. CHANNELS of BEHAVIOUR -**ONLINE** Online handwriting recognition involves the automatic conversion of text as it is written on a special digitizer where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. **OFFLINE** -K-NN combined with preprocessing methods can achieve great performance apart from Neural Network when used as a classification algorithm in offline handwritten digit recognition.
9. PROBLEM ROOT CAUSE - A small recognition error may cause the big difference in the end result.
10. The handwritten recognition model takes an image as an input and compare the preprocessed digits with the trained datasets and give the output of digits as a text well

**REQUIREMENT ANALYSIS**

**4.1 Functional Requirement**

Following are the functional requirements of the proposed solution.

|  |  |
| --- | --- |
| **FR No.** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | Image Data: Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorise them into ten established classifications (0-9). In the realm of deep learning, this has been the subject of countless studies. |
| FR-2 | Website: Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties. |
| FR-3 | Digit Classifier Model: To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training and validation data first. |
| FR-4 | Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual platform that enables unlimited storage and access to your data over the internet. |
| FR-5 | Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9. |

**4.2 Non-Functional Requirements**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| FR No. | Non-Functional Requirement | Description |
| NFR-1 | Usability | One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail |
| NFR-2 | Security | 1) The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. 2) The generative models are capable of segmentation driven by recognition. 3) The procedure uses a relatively |
| NFR-3 | Reliability | The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances. Numerous techniques and algorithms, such as Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers |
| NFR-4 | Accuracy | With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification. |

**PROJECT DESIGN**

**1.1User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement**  **(Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Mobile user) | Home | USN-1 | As a user, I can view the guide and awareness to use this application | I can view the awareness to use this application and its limitations. | Low | Sprint-1 |
|  |  | USN-2 | As a user, I’m allowed to view the guided video to use the interface of this application | I can gain knowledge to use this application by a practical method | Low | Sprint-1 |
|  |  | USN-3 | As a user, I can read the instructions to use this application. | I can read instructions also to use it in a user- friendly method. | Low | Sprint-2 |
|  | Recognize | USN-4 | As a user, In this prediction page I get to choose the image. | I can choose the image from our local system and predict the output. | High | Sprint-2 |
|  | Predict | USN-5 | As a user, I’m Allowed to upload and choose the image to be uploaded | As a user, I’m Allowed to upload and choose the image to be uploaded. | Medium | Sprint-3 |
|  |  | USN-6 | As a user, I will train and test the input to get the maximum accuracy of output. | I can able to train and test the application until it gets maximum accuracy of the result. | High | Sprint-4 |
|  |  | USN-7 | As a user, I can access the MNIST data set | I can access the MNIST data set to produce the accurate result. | Medium | Sprint-3 |
| Customer (Web user) | Home | USN-8 | As a user, I can view the guide to use the web app. | I can view the awareness of this application and its limitations. | Low | Sprint-1 |

**PROJECT PLANNING & SCHEDULING**

* 1. **Sprint Planning & Estimation**

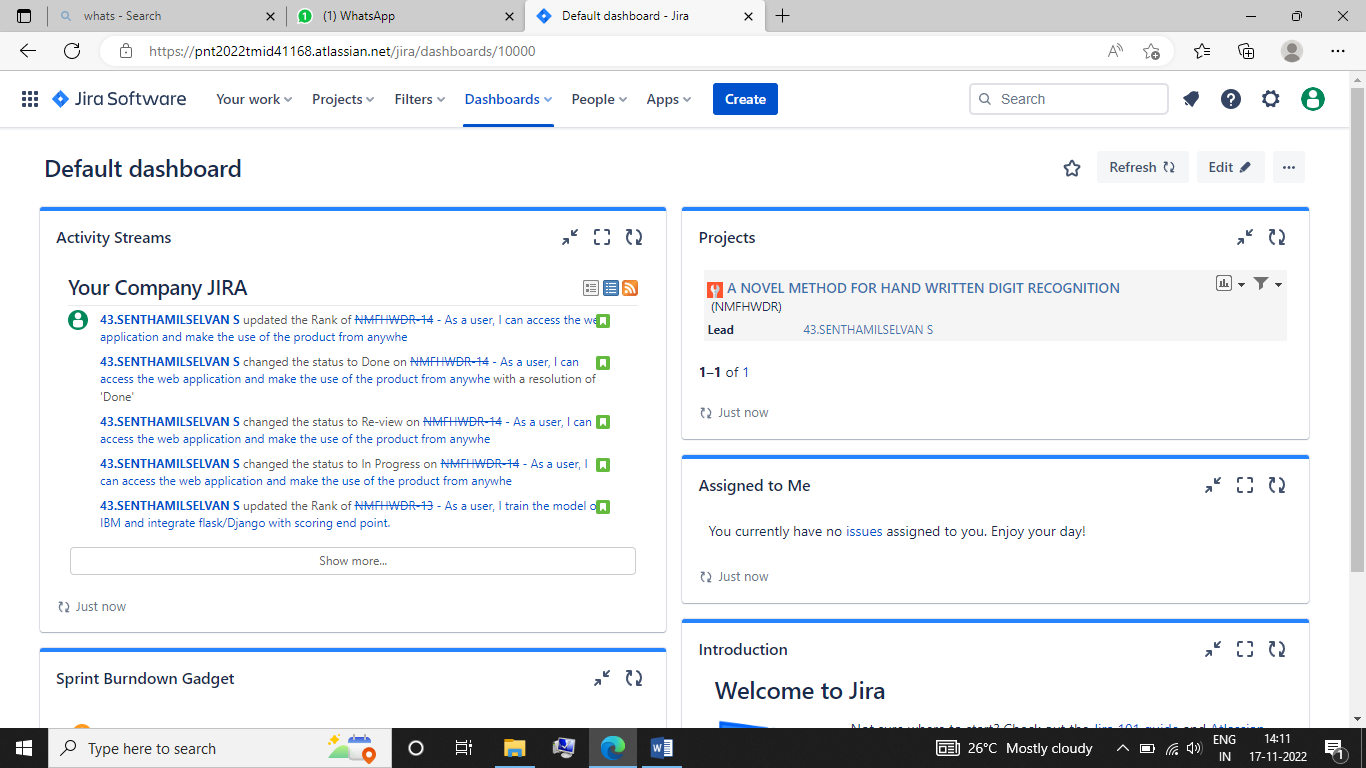
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Data Collection | USN-1 | As a user, I can collect the dataset from various resources with different handwritings. | 10 | Low | Senthamilselvan S |
| Sprint-1 | Data Preprocessing | USN-2 | As a user, I can load the dataset, handling the missing data, scaling and split data into train and test. | 10 | Medium | Ramu M |
| Sprint-2 | Model Building | USN-3 | As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit. | 5 | High | Muniraj M |
| Sprint-2 | Add CNN layers | USN-4 | Creating the model and adding the input, hidden, and output layers to it. | 5 | High | Senthil S |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-2 | Compiling the model | USN-5 | With both the training data defined and model defined, it's time to configure the learning process. | 2 | Medium | Senthil S |
| Sprint-2 | Train & test the model | USN-6 | As a user, let us train our model with our image dataset. | 6 | Medium | Ramu M |
| Sprint-2 | Save the model | USN-7 | As a user, the model is saved & integrated with an android application or web application in order to predict something. | 2 | Low | Muniraj M |
| Sprint-3 | Building UI Application | USN-8 | As a user, I will upload the handwritten digit image to the application by clicking a upload button. | 5 | High | Senthamilselvan S |
| Sprint-3 |  | USN-9 | As a user, I can know the details of the fundamental usage of the application. | 5 | Low | Ramu M |
| Sprint-3 |  | USN-10 | As a user, I can see the predicted / recognized digits in the application. | 5 | Medium | Senthil S |
| Sprint-4 | Train the model on IBM | USN-11 | As a user, I train the model on IBM and integrate flask/Django with scoring end point. | 10 | High | Senthamilselvan S |
| Sprint-4 | Cloud Deployment | USN-12 | As a user, I can access the web application  and make the use of the productfrom anywhere. | 10 | High | Senthamilselvan S |

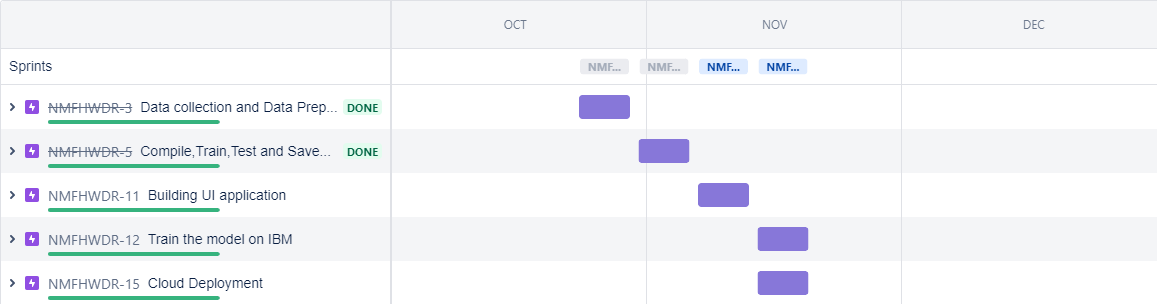
* 1. **Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

* 1. **Reports From JIRA**

****

**Roadmap of JIRA**

****

Jira Software can be configured to fit any type of project. Teams can start with a project template or create their own custom workflow. Jira issues, also known as tasks, track each piece of work that needs to pass through the workflow steps to completion. Customizable permissions enable admins to determine who can see and perform which actions. With all project information in place, reports can be generated to track progress, productivity, and ensure nothing slips.

* Issue / Task management
* [Roadmaps](https://www.atlassian.com/software/jira/features/roadmaps)
* [Report & Analytics](https://www.atlassian.com/software/jira/features/reports)
* [Integrated mobile app](https://www.atlassian.com/software/jira/mobile-app)
* Project backlogs
* Project & issue customization
* Granular user permissions
* [Workflow customization](https://www.atlassian.com/software/jira/features/workflows)

Based on the reference of the JIRA software we can easily collobrate with team members and do the project in a easy manner.

Jira Software can be configured to fit any type of project. Teams can start with a project template or create their own custom workflow. Jira issues, also known as tasks, track each piece of work that needs to pass through the workflow steps to completion. Customizable permissions enable admins to determine who can see and perform which actions. With all project information in place, reports can be generated to track progress, productivity, and ensure nothing slips.

* [Roadmaps](https://www.atlassian.com/software/jira/features/roadmaps)
* [Report & Analytics](https://www.atlassian.com/software/jira/features/reports)
* [Integrated mobile app](https://www.atlassian.com/software/jira/mobile-app)
* Project backlogs
* Project & issue customization
* Granular user permissions
* [Workflow customization](https://www.atlassian.com/software/jira/features/workflows)
* Release planning
* Sprint planning
* [CI/CD integrations](https://www.atlassian.com/software/jira/guides/developers/ci-cd#use-cases)
* Issue management
* Project Backlog
* [Jira Service Management integration](https://www.atlassian.com/software/jira/service-management/jira-integration)
* [Feature flagging](https://www.atlassian.com/devops/feature-flagging-tutorials)
* [Developer tool integrations](https://www.atlassian.com/software/jira/guides/expand-jira/jira-integrations)

**CODING & SOLUTIONING**

**7.1 Feature 1**

* By using POST method for uploading a image.

**@app.route('/predict',methods = ['POST'])**

* Define a function to perform upload action.

**def upload\_image\_file():**

* If the request method == POST then the requested image file is gathered and then resize the image into ((28,28)).

**if request.method == 'POST':**

**img = Image.open(request.files['file'].stream).convert("L")**

**img = img.resize((28,28))**

**7.2 Feature 2**

* For the User Identification and For the Prediction the uploaded image is converted into Array.

**im2arr = np.array(img)**

* After that the image again reshaped into (1,28,28,1) because the arrays have those kind of Dimensions.

**im2arr = im2arr.reshape(1,28,28,1)**

* By using the model we can predict the image and check the accuracy of the model.

**y\_pred = model.predict\_classes(im2arr)**

**print(y\_pred)**

**TESTING**

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements

**8.1 Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Parameter** | **Values** | **Screenshot** |
|  | Model Summary | Fitting the Model-   * loss: 0.0181 * val\_loss: 0.1560 | C:\Users\muniraj\Pictures\Screenshots\Screenshot (83).png |
|  | Accuracy | Training Accuracy – 0.9957  Validation Accuracy -0.9797 | C:\Users\muniraj\Pictures\Screenshots\Screenshot (83).png |

By the above table we can know the accuracy and the loss of the model.

**8.2 USER ACCEPTANCE TESTING**

This is a type of system or software testing where a system has been tested for availability.The purpose of this test is to check the business requirements and assess whether it will be accepted for delivery.In this part ADRIAN of pyrimagsearch has been referred to, who worked with the same platform and to check this project accepted by the delivery partner or not.

# Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of **A Novel Method For Hand Written Recognition project** at the time of the release to User Acceptance Testing (UAT).

# Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 10 | 5 | 2 | 3 | 20 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 5 | 0 | 1 | 8 |
| Fixed | 10 | 3 | 2 | 18 | 33 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 23 | 18 | 11 | 24 | 76 |

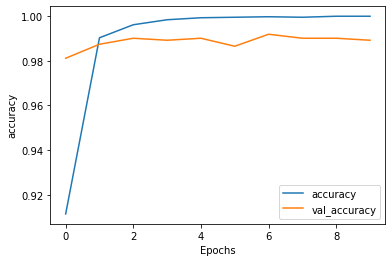
# 3.Test Case Analysis

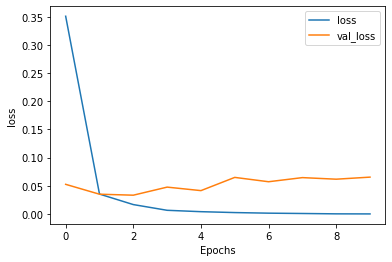
This report shows the number of test cases that have passed, failed, and untested

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| Print Engine | 6 | 0 | 0 | 6 |
| Client Application | 62 | 0 | 0 | 58 |
| Security | 4 | 0 | 0 | 3 |
| Outsource Shipping | 2 | 0 | 0 | 2 |
| Exception Reporting | 7 | 0 | 0 | 7 |
| Final Report Output | 3 | 0 | 0 | 3 |
| Version Control | 6 | 0 | 0 | 5 |

**RESULTS**

**9.1 Performance metrics**

****

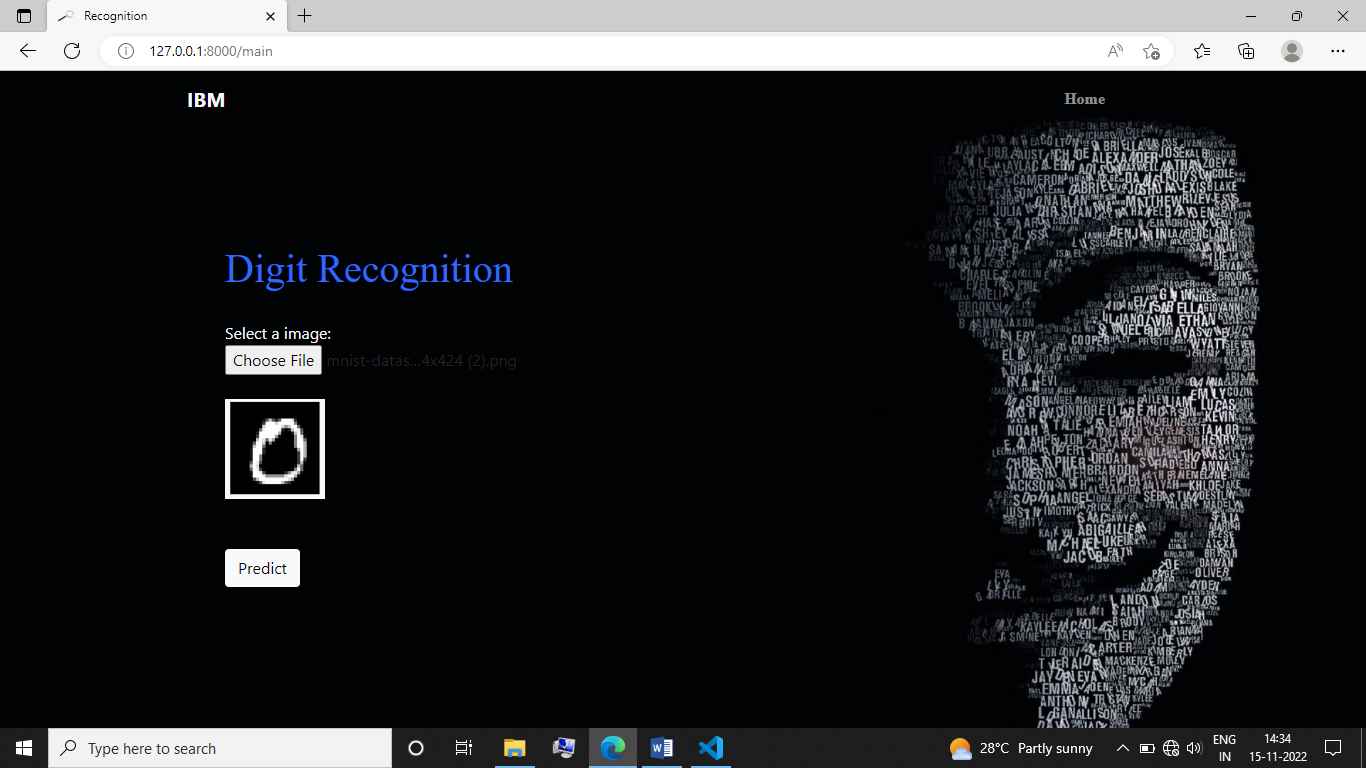
****

**OUTPUT & SCREENSHOTS**

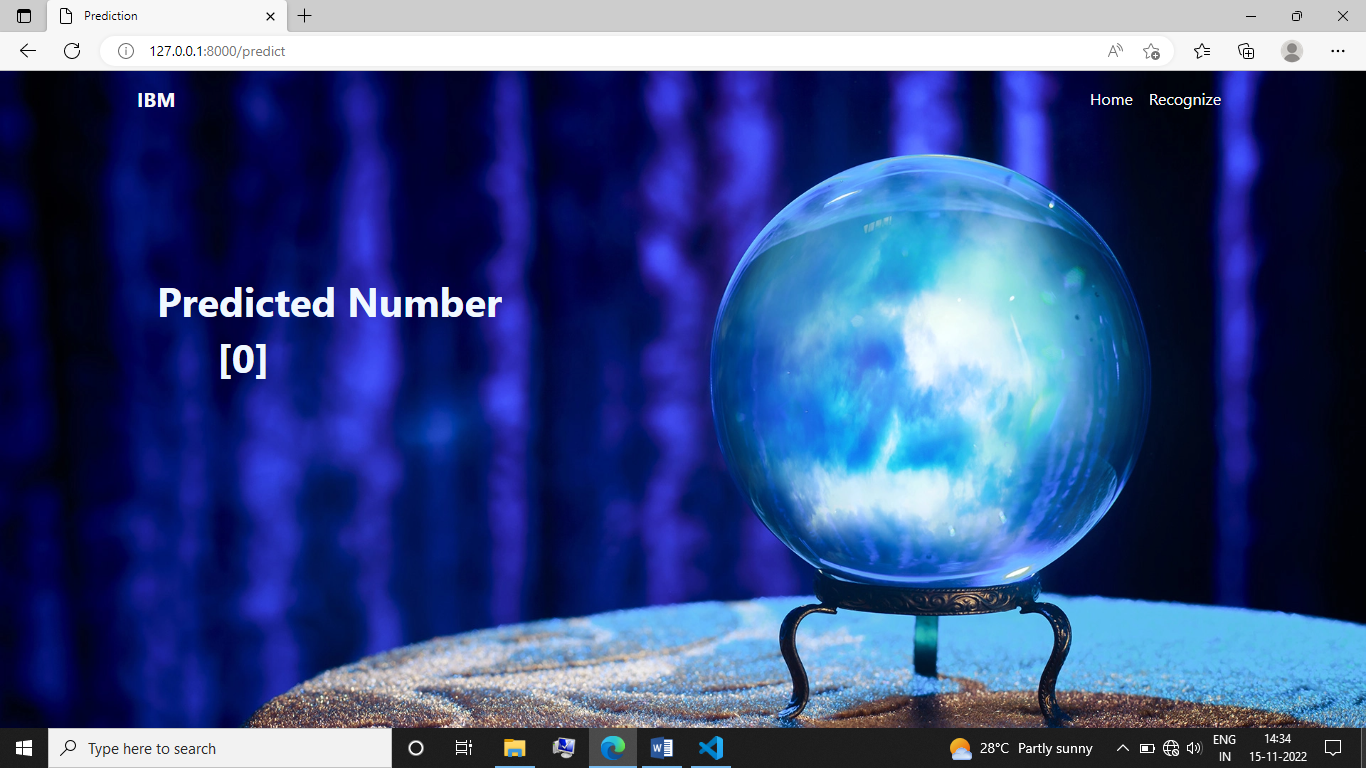
**Index page:**



**Main page:**



**Prediction Page:**



**ADVANTAGES & DISADVANTAGES**

**Advantages:**

* **Easy to Recognize Hand written digits in a simple way.**
* **Accuracy of the model is very high.**
* **We can get the predicted output as much we expected.**
* **More organized fields.**
* **Historical preservation.**
* **Can be easier form of recognition.**
* **It makes the interaction between the human and the system.**

**Disadvantages:**

* **It cannot predict a sequence of numbers in a single process.**
* **It can predict only a single digit.**
* **It is a sample project .**
* **Need to improve the Model in order to get better outcomes.**

**CONCLUSION**

In this study, interdisciplinary research has been carried out for Handwritten Digit Prediction through interaction of 60,000 sample dataset. A robust multilayer prediction model is generated in combination with the computation of maximum obtainable seismic features. 36 seismic features are consider for this problem. The features are employed for training of an earthquake prediction model for better result. The prediction model consists of Random forests classifier (RFC) method. RFC provides an initial estimation for Handwritten Digit prediction. Thus RFC based prediction model is trained and tested successfully with encouraging and improved result by selecting important features and training model. Performance of a network depends on many factors like low memory requirements, low run time and better accuracy, although in this paper it is primarily focused on getting better accuracy rate for classification. Before Artificial neurons had better accuracy but now the branch of computer vision mainly depends on deep learning features like convolutional neural networks. The branch of computer vision in artificial intelligence primary motive is to develop a network which is better to every performance measure and provide results for all kinds of datasets which can be trained and trained and recognized.

**FUTURE SCOPE**

.

The future steps that to go for would be having a closer look at the results of all the versions in order to find new rules. By extracting and implementing them, we will be able to enhance the performance of these versions. Moreover, it would be good if we could make some modifications to both the reference set and the rules in order to make our program more general and able to identify both typed and handwritten digits. Furthermore, in the future, we could make a great use of the matrices that indicate the first maximum overlap of each test image with the reference images, along with the number of pixels left out from both. These matrices could be used with some clustering algorithms to build a program able to recognize handwritten digits with a very high efficiency. Last but not least, we thought about using linear or high level regression in the versions we have developed in order to create more rules. As regression could be used for binary classification and is not very suitable to classify a digit out of ten, this technique could be used in order to tell which digit is the most suitable, the first maximum or second maximum, which will enable us to generate more rules; thus, reach a higher efficiency.

**APPENDIX**

**13.1 SOURCE CODE**

**MODEL BUILDING**

**Understanding the Data**

#importing the required libraries

import numpy as np

import tensorflow

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Flatten

from tensorflow.keras.layers import Conv2D

from keras.optimizers import Adam

from keras.utils import np\_utils

#loading data

(X\_train,y\_train) , (X\_test,y\_test)=mnist.load\_data()

print(X\_train.shape)

print(X\_test.shape)

**Analyzing the data**

X\_train[0]

y\_train[0]

import matplotlib.pyplot as plt

plt.imshow(X\_train[0])

**Reshaping the data**

X\_train = X\_train.reshape(60000, 28, 28, 1).astype('float32')

X\_test = X\_test.reshape(10000, 28, 28, 1).astype('float32')

**Apply one-Hot Encoding**

number\_of\_classes = 10

y\_train = np\_utils.to\_categorical(y\_train, number\_of\_classes)

y\_test = np\_utils.to\_categorical(y\_test, number\_of\_classes)

y\_train[0]

**Model Building**

**Add CNN Layers**

**Creating the Model**

model = Sequential()

model.add(Conv2D(64, (3, 3), input\_shape = (28, 28, 1), activation='relu'))

model.add(Conv2D(32, (3,3), activation = 'relu'))

model.add(Flatten())

model.add(Dense(number\_of\_classes, activation = 'softmax'))

**compiling the Model**

model.compile(loss='categorical\_crossentropy',optimizer="Adam",metrics=['accuracy'])

number\_of\_classes = 10

**Train the model**

#Fitting the model

model.fit(X\_train, y\_train, validation\_data = (X\_test, y\_test), epochs=12, batch\_size=128 )

**Observing the metrics**

metrics =model.evaluate(X\_test,y\_test,verbose=0)

print("Metrics(Test loss & Test Acurracy):")

print(metrics)

**Test the Model**

**Predicting the output**

prediction=model.predict(X\_test[:4])

print(prediction)

print(np.argmax(prediction,axis=1))

print(y\_test[:4])

**Save The model**

model.save('models/mnistCNN.h5')

**Test With saved model**

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import numpy as np

model = load\_model('mnistCNN.h5')

img = image.load\_img('9.png',target\_size=(28,28),grayscale=True)

img = image.img\_to\_array(img)

print(img.shape)

x = np.expand\_dims(img,axis=0)

print(x.shape)

print('\*'\*20)

print(model.predict(x))

print('\*'\*20)

print(np.round\_(model.predict(x)))

**FLASK APP**

from flask import Flask, render\_template, request

from PIL import Image

import numpy as np

from tensorflow.keras.models import load\_model

import tensorflow as tf

from flask import Flask

#You need to use following line [app Flask(\_\_name\_\_)]

app = Flask(\_\_name\_\_,template\_folder="templates")

model = load\_model("models\mnistCNN.h5")

@app.route('/')

def upload\_file():

return render\_template('index.html')

@app.route('/main')

def upload\_file1():

return render\_template('main.html')

@app.route('/predict',methods = ['POST'])

def upload\_image\_file():

if request.method == 'POST':

img = Image.open(request.files['file'].stream).convert("L")

img = img.resize((28,28))

im2arr = np.array(img)

im2arr = im2arr.reshape(1,28,28,1)

y\_pred = model.predict\_classes(im2arr)

print(y\_pred)

if(y\_pred == 0) :

return render\_template("0.html", showcase = str(y\_pred))

elif(y\_pred == 1) :

return render\_template("1.html ", showcase = str(y\_pred))

elif(y\_pred == 2) :

return render\_template("2.html", showcase = str(y\_pred))

elif(y\_pred == 3) :

return render\_template("3.html", showcase = str(y\_pred))

elif(y\_pred == 4) :

return render\_template("4.html", showcase = str(y\_pred))

elif(y\_pred == 5) :

return render\_template("5.html", showcase = str(y\_pred))

elif(y\_pred == 6) :

return render\_template("6.html", showcase = str(y\_pred))

elif(y\_pred == 7) :

return render\_template("7.html", showcase = str(y\_pred))

elif(y\_pred == 8) :

return render\_template("8.html", showcase = str(y\_pred))

else :

return render\_template("9.html", showcase = str(y\_pred))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0',port=8000,debug=True,threaded=False)

**13.2 GITHUB AND PROJECT DEMO**

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-45981-1660733831

PROJECT DEMO LINK: