A NOVEL METHOD FOR HAND WRITTEN DIGIT RECOGNITION

PROJECT REPORT

Submitted by

MUNIRAJ.M	**(612819104022)**
RAMU.M	**(612819104032)**
SENTHAMILSELVAN.S	**(612819104043)**
SENTHIL.S	**(612819104044)**

TEAMID:PNT2022TMID41168

in partial fullfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

TABLE OF CONTENTS

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.

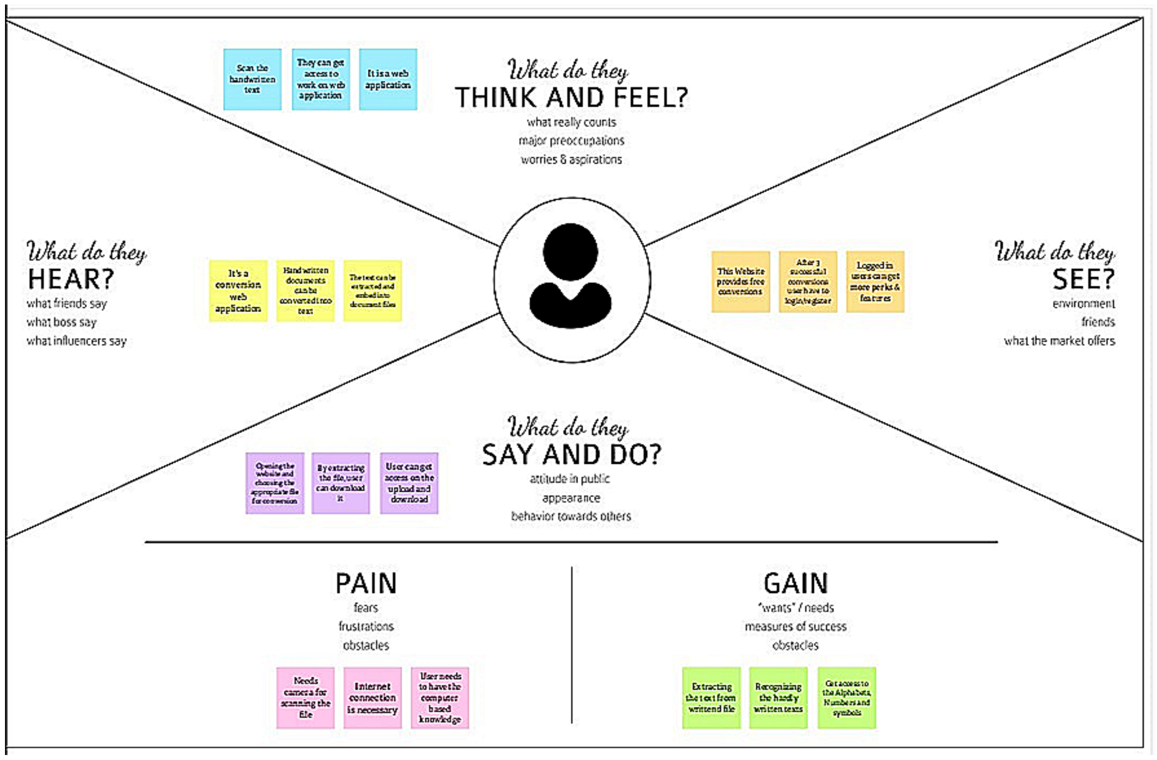
7, July 2017.
[3]. Al Maadeed, Somaya, and Abdelaali Hassaine, Automatic prediction of age, gender, and nationality in offline handwriting. EURASIP Journal on Image and Video Processing, no. 1 2014. [4]. Gaurav Jain, Jason Ko, Handwritten DigitsRecognition, Project Report, University of Toronto, 11/21/2008.
[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


Scan the handwritten text
They can get access to work on web application
It is a web application
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they HEAR?
what friends say
what boss say
what influencers say
It's a conversion web application
Hand written documents can be converted into text
The text can be extracted and embed into document files
This Website provides free conversions
After 3 successful conversions user have to login/register
Logged in users can get more perks & features
What do they SEE?
environment
friends
what the market offers
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Opening the website and choosing the appropriate file for conversion
By extracting the file, user can download it
User can get access on the upload and download
PAIN
fears
frustrations
obstacles
Needs camera for scanning the file
Internet connection is necessary
User needs to have the computer based knowledge
GAIN
"wants" / needs
measures of success
obstacles
Extracting the text from writtend file
Recognizing the handly written texts
Get access to the Alphabets, Numbers and symbols

1. 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.

2. Empathy mapping is a simple yet effective workshop 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. Essentially, it is an exercise that seeks to get inside the head of the customer as they interact with

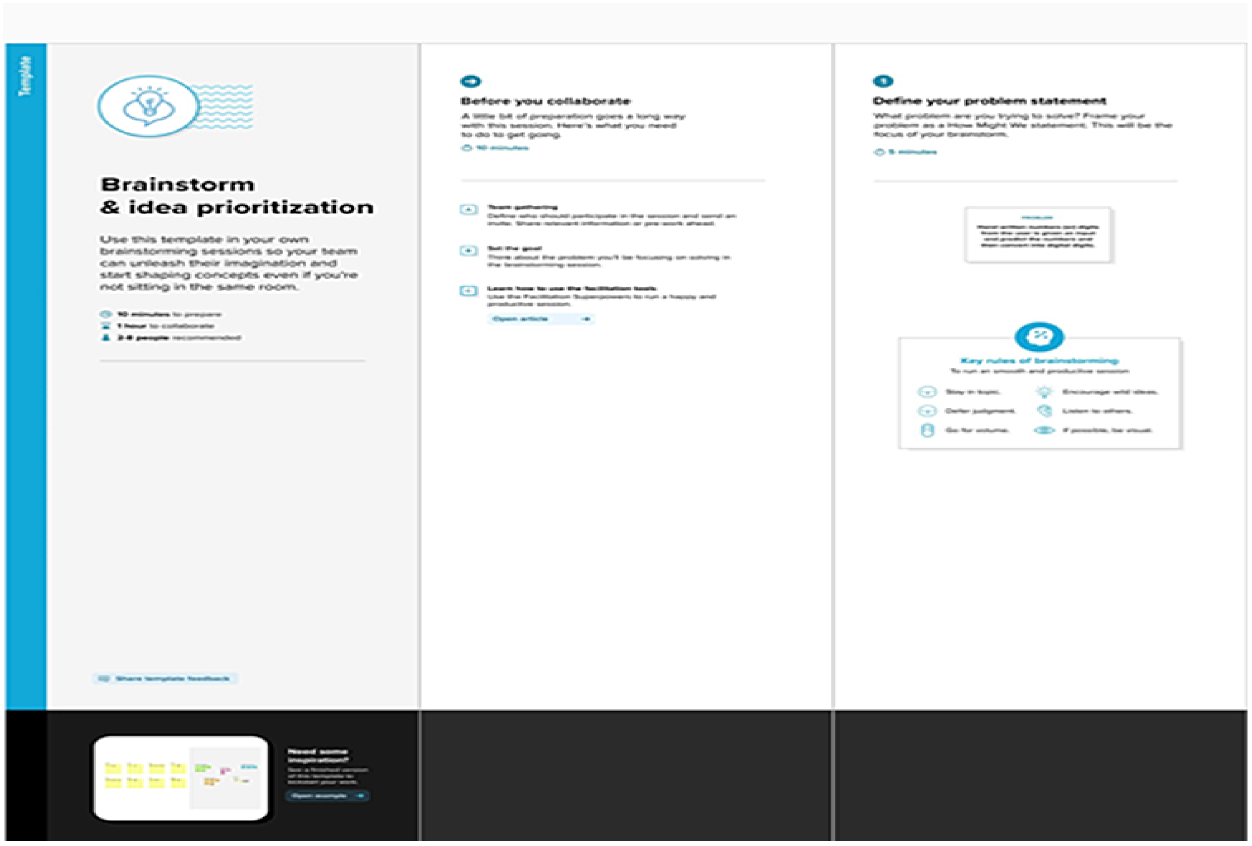
your product/service.

3. 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.

4. 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



Template

Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Need some inspiration?
See a finished version of this template to kickstart your work.
Open example

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.
Open article

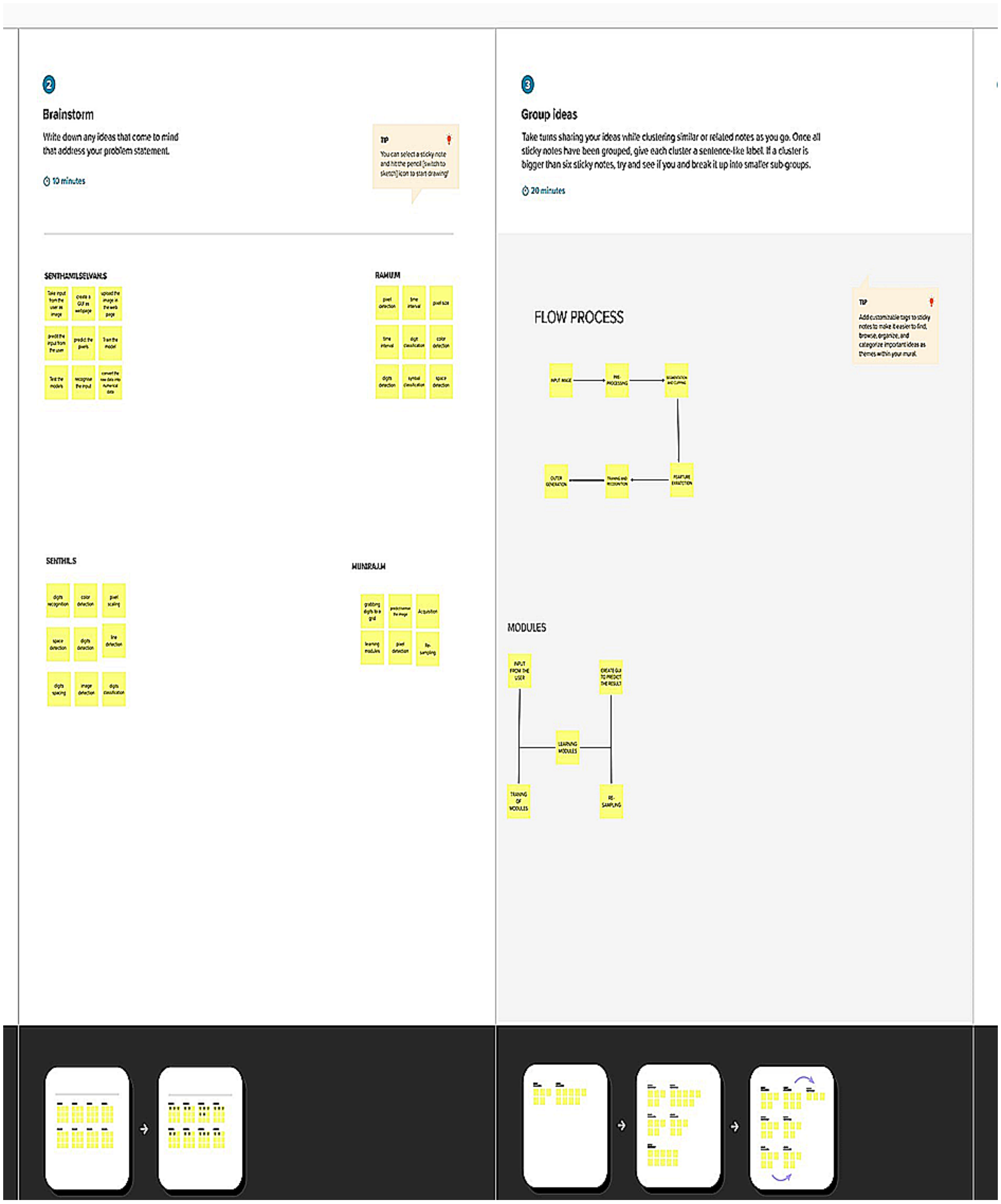
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes

PROBLEM
[illegible]

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

STEP-2: BRAINSTORM, IDEA LISTING AND GROUPING



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group ideas

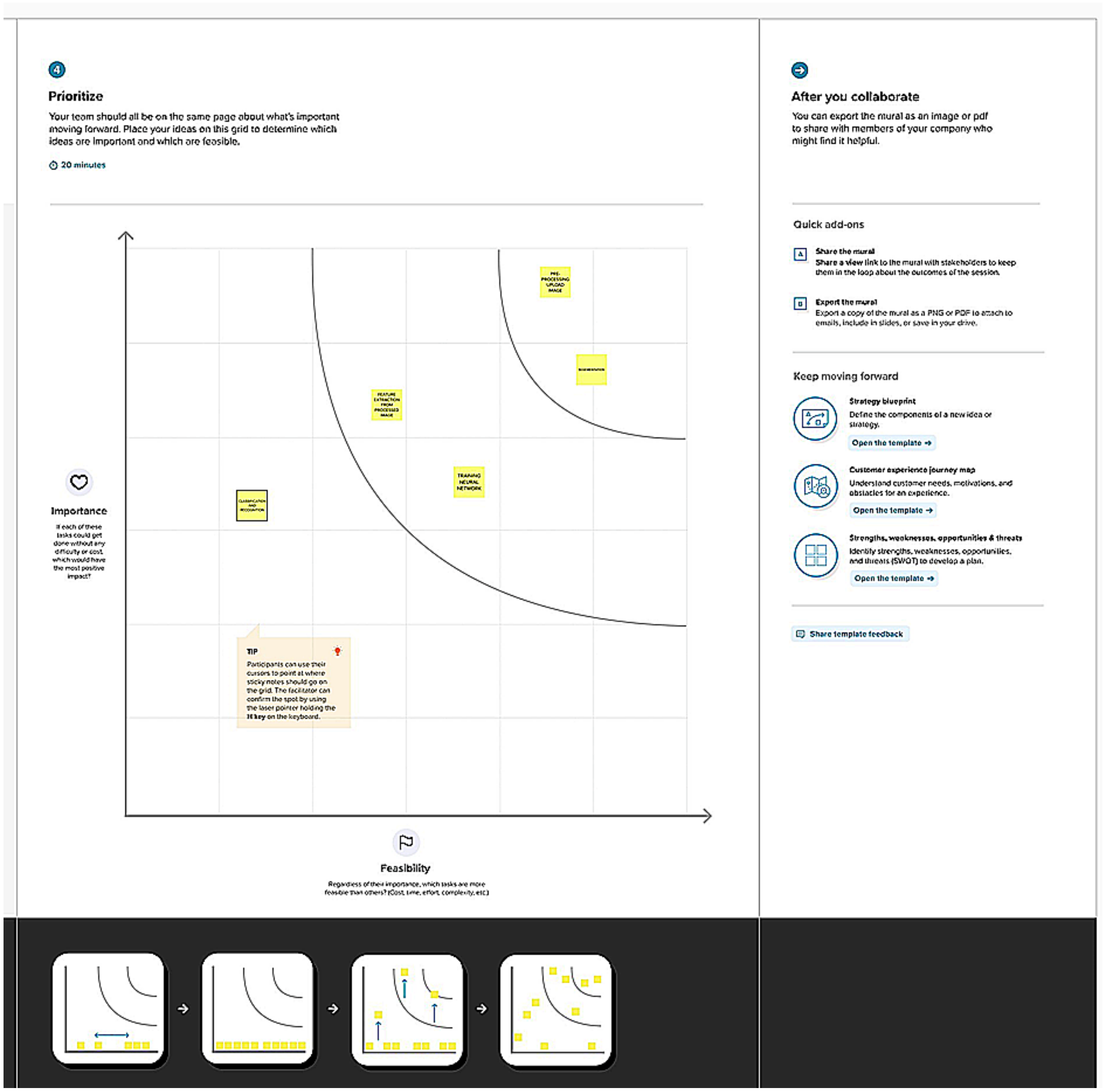
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Step-3: Idea Prioritization



Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

3.3 Proposed Solution

1. 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.

2. 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.

3. Novelty / Uniqueness Recognize the digits precisely rather than all the characters like OCR.

4. 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.

5. 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.

6. 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

A. CUSTOMER SEGMENT(S) -A officer in a post office receiving letters and couriers in a written format.

B. 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.

C. TRIGGERS - To do the work in a efficient manner. So that the officer getting satisfied.

D. EMOTIONS: BEFORE / AFTER- BEFORE: They eagerly wants to finish is work quickly and easily .AFTER: If its working fine they feels better

E. 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.

F. 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.

G. 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.

H. 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.

I. PROBLEM ROOT CAUSE - A small recognition error may cause the big difference in the end result.

J. 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

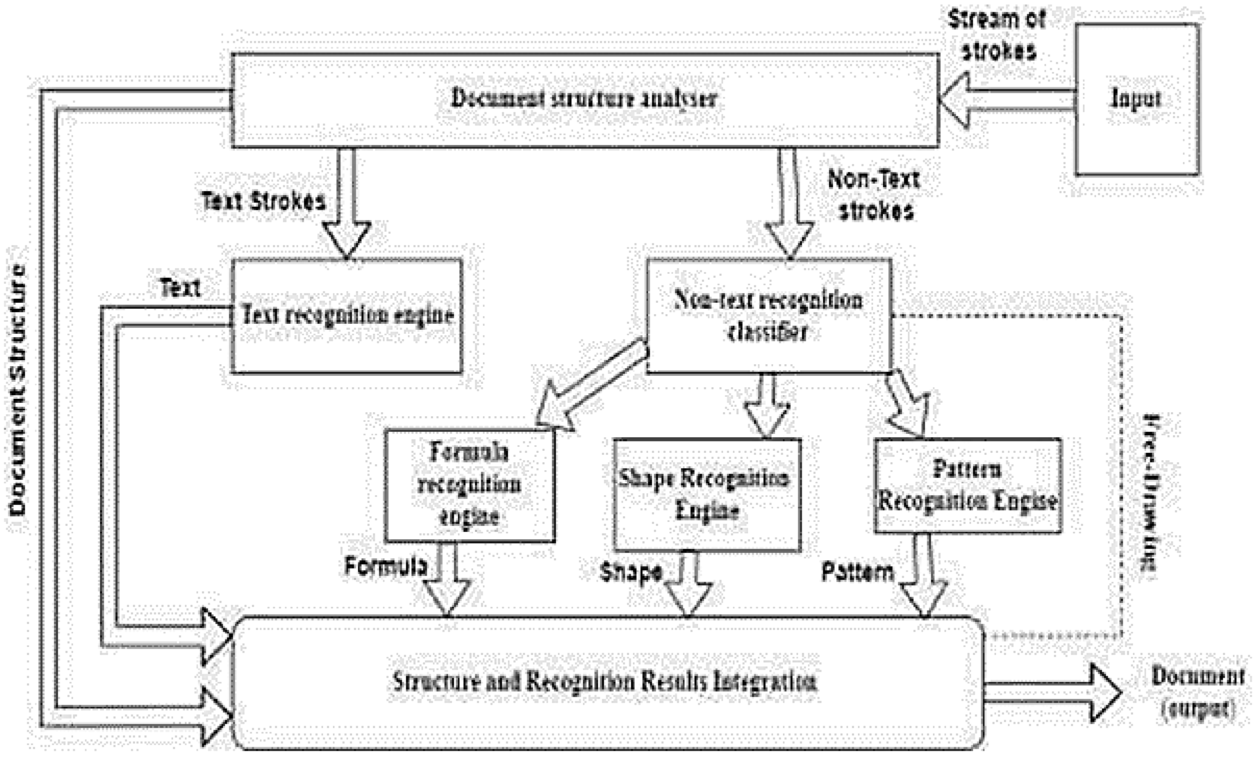
Following are the non-functional requirements of the proposed solution.

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

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

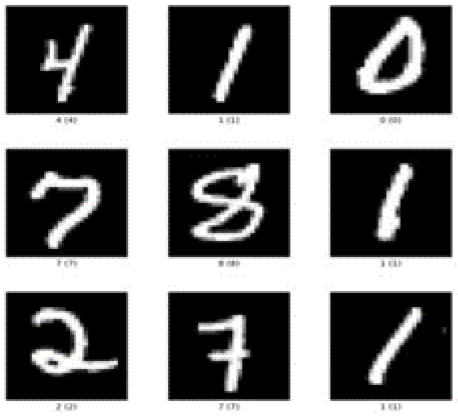


5.2 Solution & Technical Architecture

MNIST Dataset:

One of the interesting research projects is the recognition of handwriting. It is the ability of a computer to automatically recognise and comprehend handwritten numbers or letters. Every aspect of life is being digitalized to lessen the need for human labour as a result of advancements in science and technology. Thus, handwritten digit recognition is required in many realtime applications. The MNIST data collection, which contains 70000 handwritten digits, is frequently utilised for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. This image is examined by the model, which then sends the results back to the user interface.

There are 60,000 training and 10,000 testing labelled handwritten digit images in the MNIST Handwritten Digit Recognition Dataset. There are 28 pixels in height and 28 pixels in width in each image, for a total of 784 (2828) pixels. A single pixel value corresponds to each pixel. It tells whether a pixel is bright or dark (larger numbers indicates darker pixel). An integer from 0 to 255 makes up this pixel value.



PROCEDURE:

- Set up the most recent TensorFlow library. •Configure the model's dataset.

1. Construct a single layer perceptron model to categorise the handwritten digits.
2. Plot the accuracy change over time.
3. Assess the model based on the test data.
4. Examine the summary of the model.

5. To construct a multi-layer perceptron, add a hidden layer to the model.

6. Incorporate Dropout to avoid overfitting and evaluate its impact on accuracy.

- Adding more Hidden Layer neurons and evaluating the impact on accuracy.

- Test the impact of various optimizers on accuracy.

- Boost the number of hidden layers and assess the impact on accuracy.

7. Test the impact of changing the batch size and epochs on accuracy.

This project will be approached utilising a three-layered neural network.

The input layer:

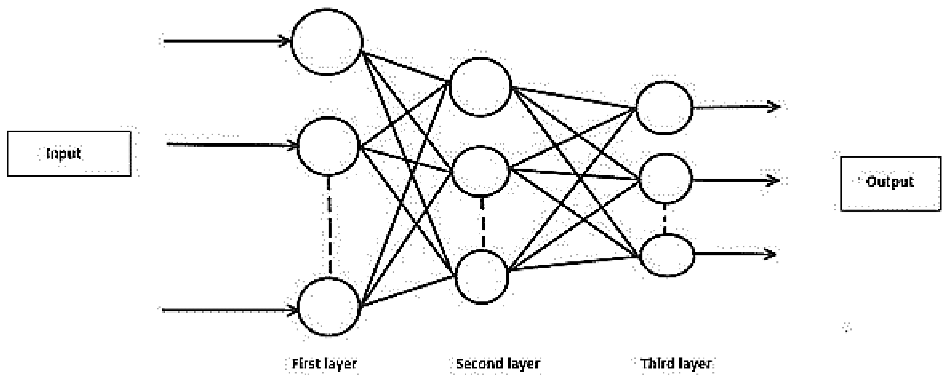
1. It transfers the characteristics from our example layers to the following layer so that the subsequent layer's activations can be calculated.

The hidden layer:

2. These ties for the network are built up of hidden units known as activations. Depending on our needs, there can be a variety of concealed layers.

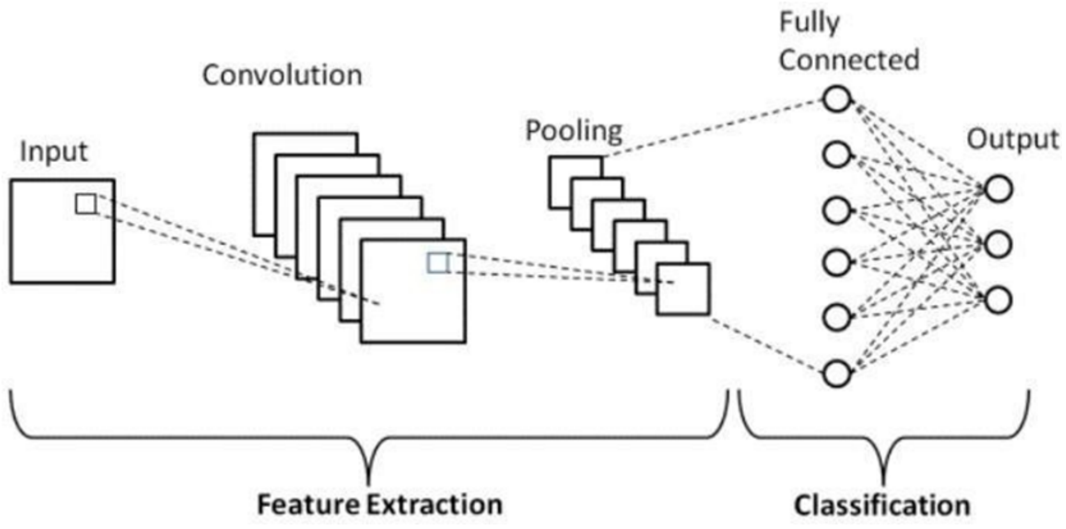
The output layer:

3. The nodes in this layer are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions.



Forward Propagation Process:

It's a simple process through which the CNN module will extract the features and categorise the image using them. The architecture displays the network's input layer, hidden layers, and output layer. The feature extraction phase of the network involves multiple layers and uses convolution and subsampling.



Working:

Neural networks take in data and process it through a number of secret layers. Each hidden layer is composed of a group of neurons, each of which is completely linked to every neuron in the layer above. A single layer of neurons has totally independent functioning. "Output layer" refers to the final layer that is entirely connected.

Convolution Layer:

The fundamental component of a CNN is the convolutional layer. The parameters of the layer are a set of learnable filters that cover the entire depth of the input volume but have a narrow receptive field. Each filter is convolved across the width and height of the input volume during the forward pass, computing the dot product between each filter entry and the input to create a two-dimensional activation map of the filter. As a result, the network picks up filters that turn on when they spot a certain kind of feature at a particular location in the input.

Feature Extraction:

The weights of each neuron in a feature are the same. In this manner, the same feature is recognised by all neurons at various locations in the input image. Limit the number of unrestricted parameters.

Subsampling Layer: Reducing the overall size of a signal is referred to as subsampling, sometimes known as down sampling. Each feature map's spatial resolution is decreased by the subsampling layers. Shift or distortion invariance is attained, and the impact of sounds is lessened.

Pooling layer:

In a Convent architecture, it is typical to sporadically introduce a Pooling layer between succeeding Conv layers. In order to decrease the number of parameters and computation in the network and, as a result, control overfitting, it gradually shrinks the spatial size of the representation. Every depth slice of the input is independently processed by the Pooling Layer, which then applies the MAX operation to resize each slice spatially.

TensorFlow:

An open-source machine learning library for both research and production is called TensorFlow. TensorFlow provides developers of all skill levels with APIs for desktop, mobile, web, and cloud applications. To get started, refer to the sections below. We can achieve text output and sound output by scanning the number digit and converting it to png format using the python3 command in terminal.

RESULT:

We are not claiming that our results are infallible, as with any endeavour in the fields of machine learning and image processing. There is always opportunity for methodological development in the field of machine learning; there will always be a fresh new idea that solves the same problem more effectively. Three models were used to test the application: Convolution Neural Network, Multi-Layer Perceptron (MLP), and (CNN). The classifier accuracy varies with each model, allowing us to determine which is more accurate.

3.3 USER 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

a.Sprint Planning & Estimation

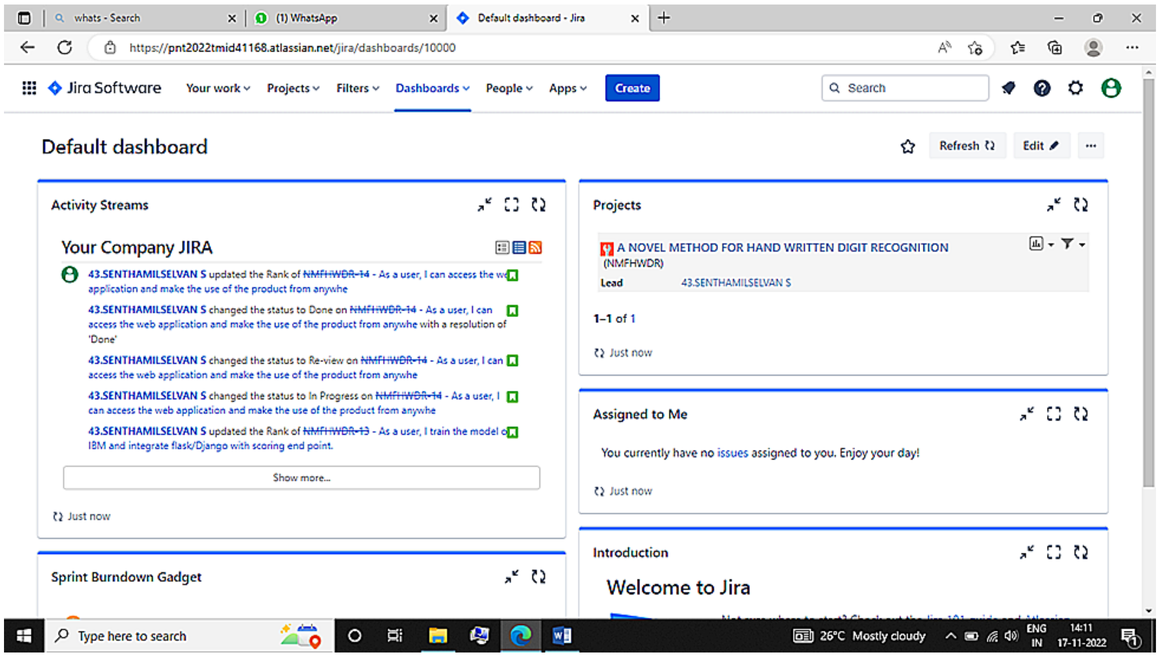
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Senthamilselvan S
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split datainto 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-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 uploadthe 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 themodel on IBM and integrate flask/Django withscoring 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

b.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

Reports From JIRA



Default dashboard

Activity Streams

Your Company JIRA

43.SENTHAMILSELVAN S updated the Rank of ~~NMFHWDR-14~~ - As a user, I can access the we application and make the use of the product from anywhe

43.SENTHAMILSELVAN S changed the status to Done on ~~NMFHWDR-14~~ - As a user, I can access the web application and make the use of the product from anywhe with a resolution of 'Done'

43.SENTHAMILSELVAN S changed the status to Re-view on ~~NMFHWDR-14~~ - As a user, I can access the web application and make the use of the product from anywhe

43.SENTHAMILSELVAN S changed the status to In Progress on ~~NMFHWDR-14~~ - As a user, I can access the web application and make the use of the product from anywhe

43.SENTHAMILSELVAN S updated the Rank of ~~NMFHWDR-13~~ - As a user, I train the model o IBM and integrate flask/Django with scoring end point.

Show more...

Just now

Sprint Burndown Gadget

Projects

A NOVEL METHOD FOR HAND WRITTEN DIGIT RECOGNITION (NMFHWDR)

Lead 43.SENTHAMILSELVAN S

1–1 of 1

Just now

Assigned to Me

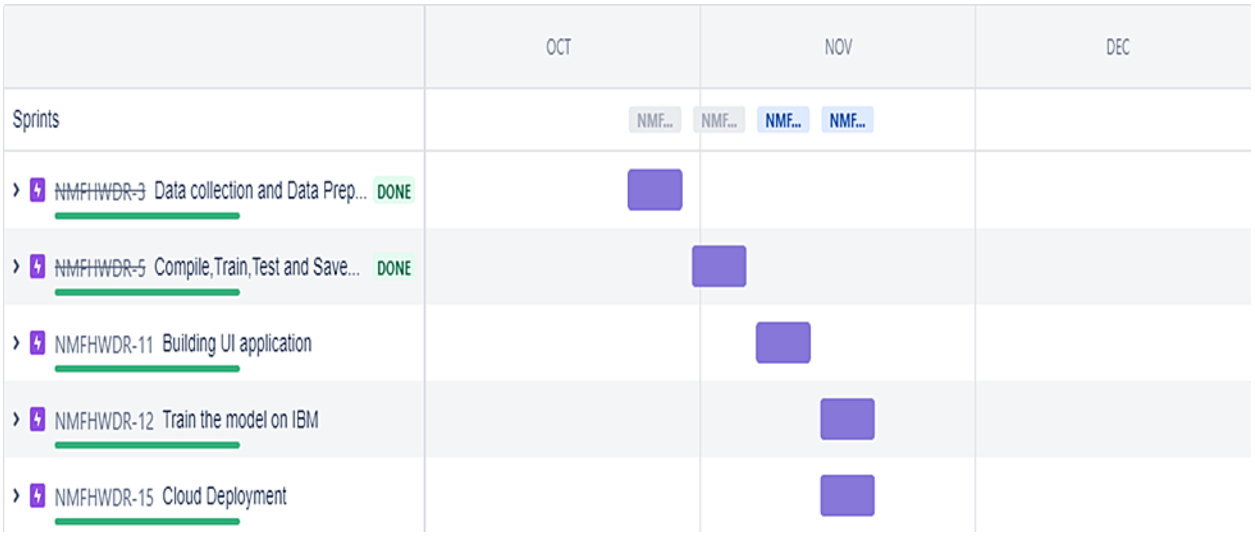
You currently have no issues assigned to you. Enjoy your day!

Just now

Introduction

Welcome to Jira

Roadmap of JIRA



~~NMFHWDR-3~~ Data collection and Data Prep... DONE

~~NMFHWDR-5~~ Compile,Train,Test and Save... DONE

NMFHWDR-11 Building UI application

NMFHWDR-12 Train the model on IBM

NMFHWDR-15 Cloud Deployment

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
- Report & Analytics
- Integrated mobile app
- Project backlogs
- Project & issue customization
- Granular user permissions
- Workflow customization

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
- Report & Analytics
- Integrated mobile app
- Project backlogs
- Project & issue customization
- Granular user permissions
- Workflow customization

- Release planning
- Sprint planning
- CI/CD integrations
- Issue management
- Project Backlog
- Jira Service Management integration
- Feature flagging
- Developer tool 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.

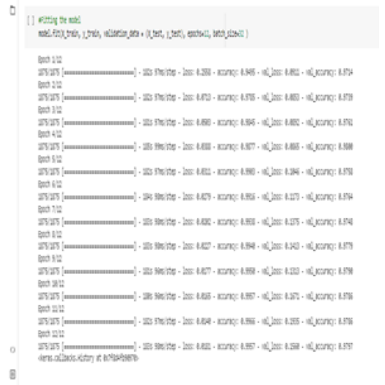
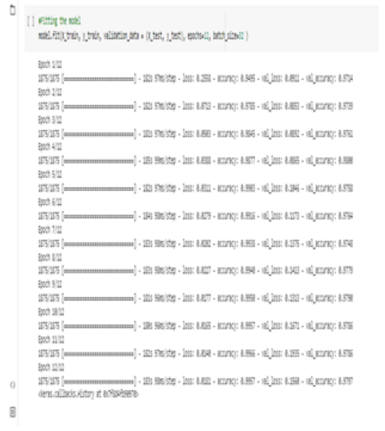
```
y_pred = model.predict_classes(im2arr)

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**
1.	Model Summary	Fitting the Model- ● loss: 0.0181 ● val_loss: 0.1560	
2.	Accuracy	Training Accuracy – 0.9957 Validation Accuracy -0.9797	

```
[ ] #Fitting the model
    model.fit(X_train, y_train, validation_data = (X_test, y_test), epochs=12, batch_size=32 )

    Epoch 1/12
    1875/1875 [==============================] - [illegible]
    Epoch 2/12
    1875/1875 [==============================] - [illegible]
    Epoch 3/12
    1875/1875 [==============================] - [illegible]
    Epoch 4/12
    1875/1875 [==============================] - [illegible]
    Epoch 5/12
    1875/1875 [==============================] - [illegible]
    Epoch 6/12
    1875/1875 [==============================] - [illegible]
    Epoch 7/12
    1875/1875 [==============================] - [illegible]
    Epoch 8/12
    1875/1875 [==============================] - [illegible]
    Epoch 9/12
    1875/1875 [==============================] - [illegible]
    Epoch 10/12
    1875/1875 [==============================] - [illegible]
    Epoch 11/12
    1875/1875 [==============================] - [illegible]
    Epoch 12/12
    1875/1875 [==============================] - [illegible]
    <keras.callbacks.History at [illegible]>
```

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.

1.Purpose of Document

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

2.Defect Analysis

This report showsthe 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

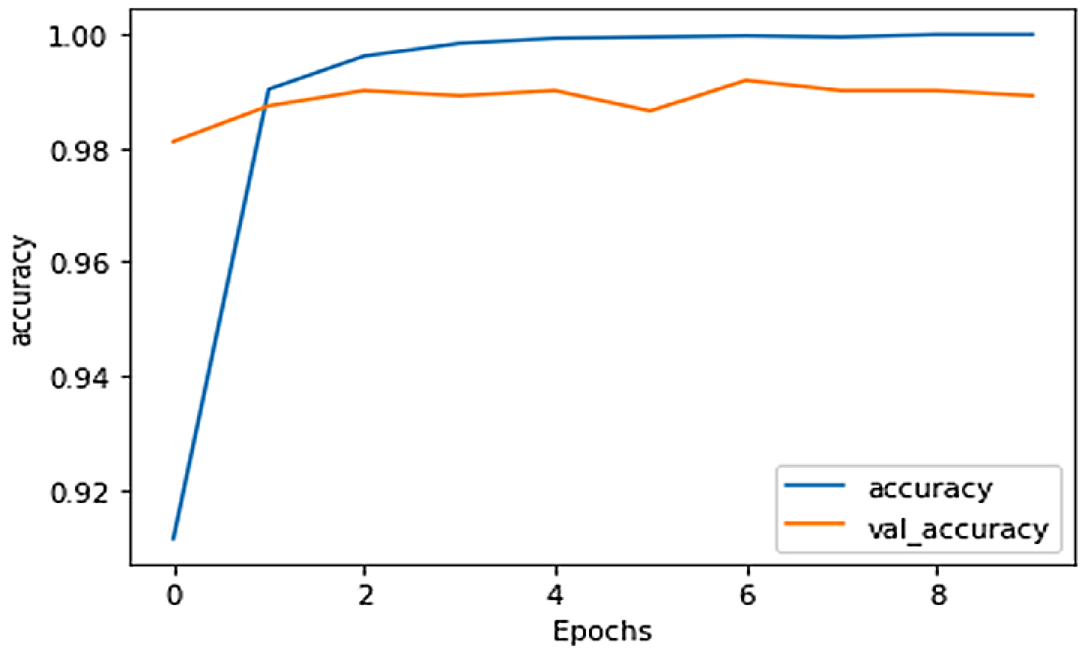
3.Test Case Analysis

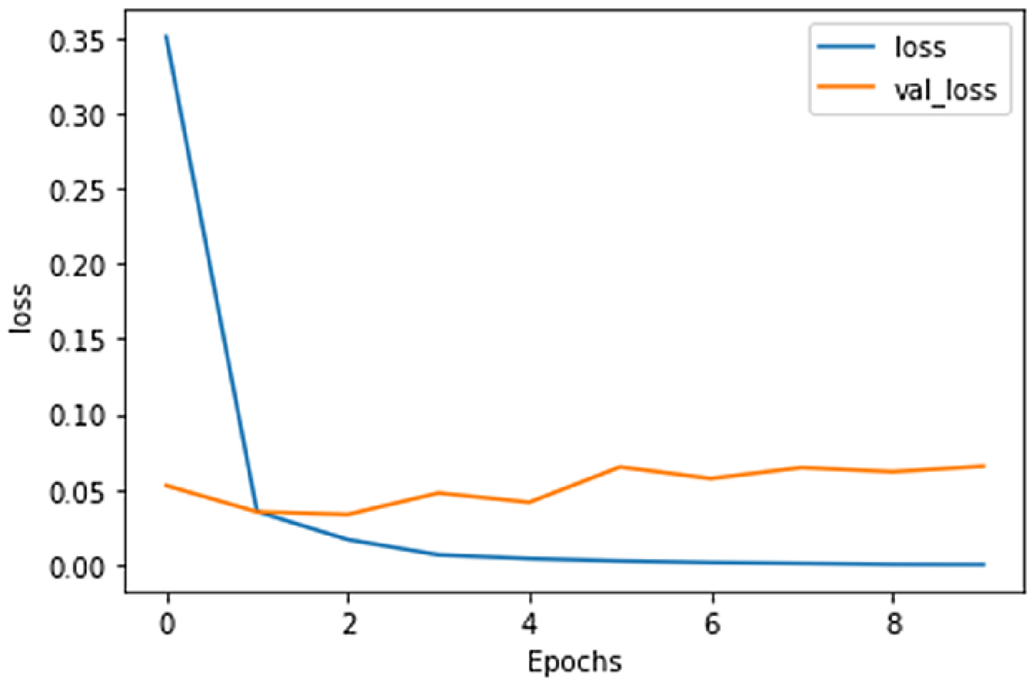
This report shows the number of test cases that have passed, failed,and untested

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 ReportOutput	3	0	0	3
Version Control	6	0	0	5

RESULTS

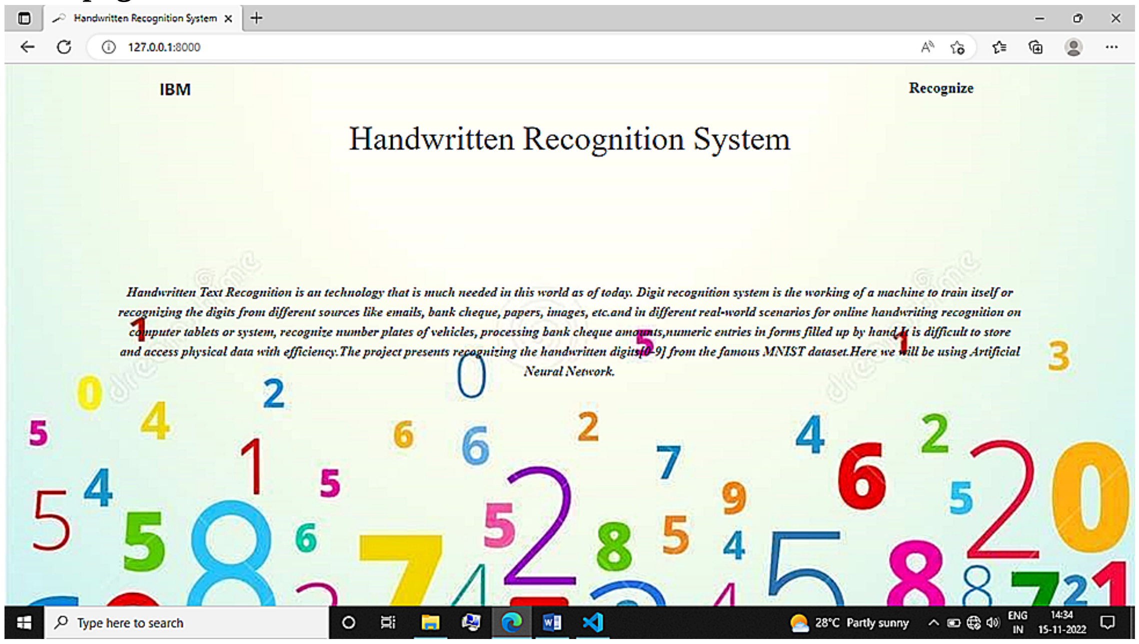
9.1 Performance metrics





OUTPUT & SCREENSHOTS

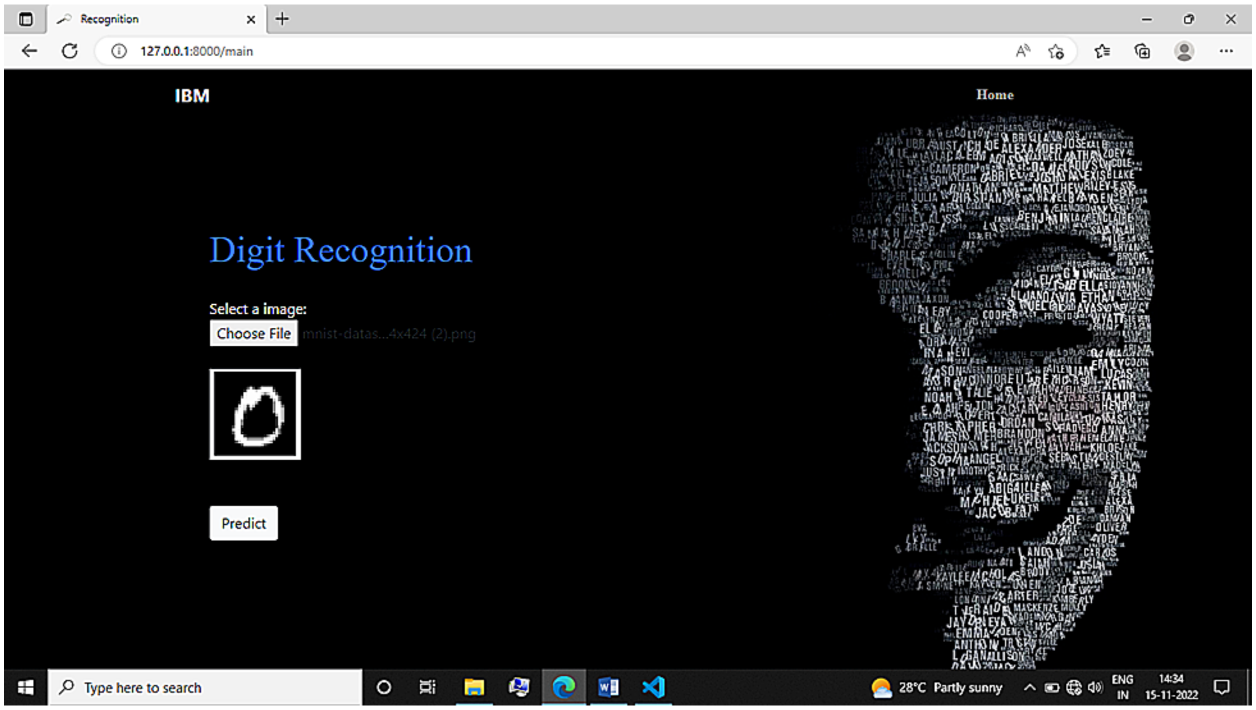
Index page:



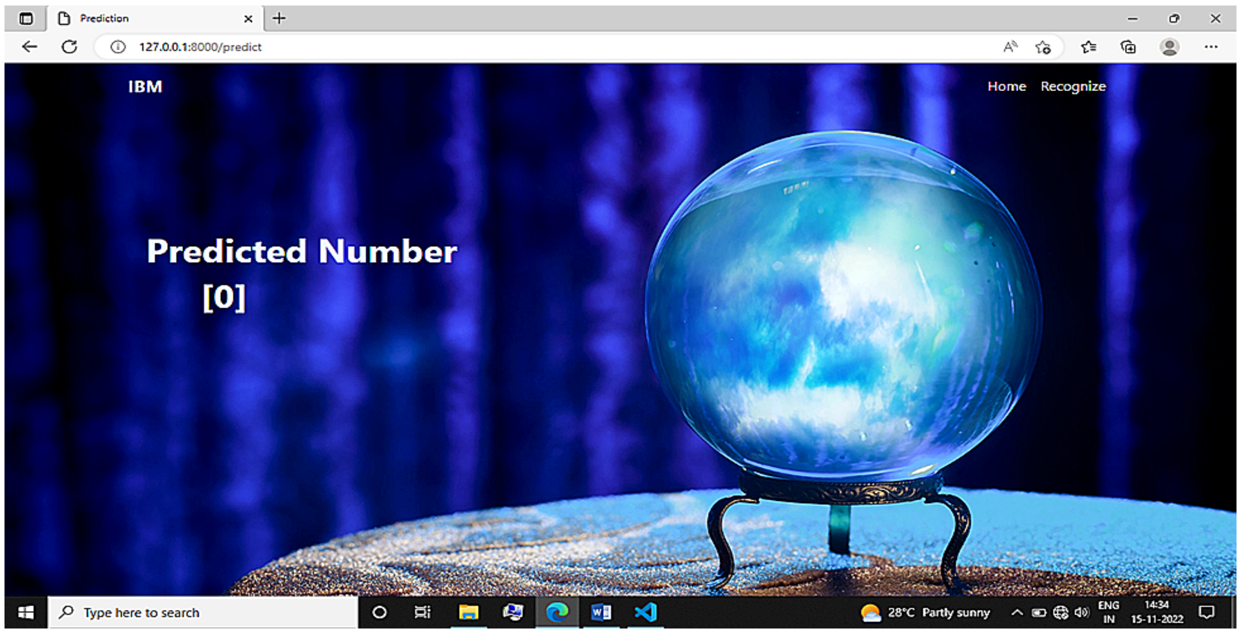
Handwritten Recognition System

Handwritten Text Recognition is an technology that is much needed in this world as of today. 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 vehicles, processing bank cheque amounts,numeric entries in forms filled up by hand.It is difficult to store and access physical data with efficiency.The project presents recognizing the handwritten digits[0-9] from the famous MNIST dataset.Here we will be using Artificial Neural Network.

Main page:



Prediction Page:



ADVANTAGES & DISADVANTAGES

Advantages:

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

Disadvantages:

1. **It cannot predict a sequence of numbers in a single process.**
2. **It can predict only a single digit.**
3. **It is a sample project .**
4. **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: https://youtu.be/2aIG9q3exRA