A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

A PROJECT REPORT

Submitted by

LOKESH P	**211419104151**
MOHAMMED ABDULLAH	**211419104167**
MADHAN MONOHARAN M	**211419104154**
KEERTHIVASHAN A K R	**211419104137**
TEAM ID:	**PNT2022TMID00805**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALARENGINEERINGCOLLEGE

(An Autonomous Institution, Affiliated to AnnaUniversity, Chennai)

DECEMBER2022

Project Report

1. **INTRODUCTION**
    a. Project Overview
    b. Purpose
2. **LITERATURE SURVEY**
    a. Existing problem
    b. References
    c. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    a. Empathy Map Canvas
    b. Ideation & Brainstorming
    c. Proposed Solution
    d. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    a. Functional requirement
    b. Non-Functional requirements
5. **PROJECT DESIGN**
    a. Data Flow Diagrams
    b. Solution & Technical Architecture

c. User Stories

6. **PROJECT PLANNING & SCHEDULING**

    a. Sprint Planning & Estimation

    b. Sprint Delivery Schedule

    c. Reports from JIRA

7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**

    a. Coding

8. **TESTING**

    a. Test Cases

    b. User Acceptance Testing

9. **RESULTS**

    a. Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, We compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolution neural networks and with Deep Learning algorithm like multi layer CNN using K eras with Thea no and Tensorflow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images. The artificial neural networks can all most mimic the human brain and are a key ingredient in image processing field. For example Convolution Neural networks with back propagation for image processing. The applications where these handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.

1.2 Purpose

HANDWRITTEN digit recognition is the ability of a computer system to recognize the handwritten inputs like digit s, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc. Here comes the use of Deep Learning. In the past decade, deep learning has become the hot tool for Image Processing, object detection, handwritten digit and character recognition etc. A lot of machine l earning tools have been developed like sci kit-learn, sci-py-image etc. and py brains, Kera s, Thea no, Tensorflow by Google, TFLearn etc. for Deep Learning. These tools make the applications robust and therefore more accurate. The Artificial Neural Networks can almost mimic the human brain and are a key ingredient in image processing field. For example, Convolution al Neural Networks with Back Propagation for Image Processing, Deep Mind by Google for cr eating Art by learning from existing artist styles etc.. Handwriting Recognition has an active community of academics studying it. The biggest conferences for handwriting recognition are the International Conference on Frontiers in Handwriting Recognition (ICFHR), held in even-numbered years, and the International Conference on Document Analysis and Recognition (ICDAR), held in odd-numbered years. Both of these conferences are endorsed by the IEEE. Active areas of research include: Online Recognition, Offline Recognition, Signature Verification, Postal-Address Interpretation, Bank-Check Processing, Writer Recognition.

Convolution Neural Networks has been proved to be a very important and powerful tool in signal and image e processing. Even in the fields of computer vision such has handwriting recognition, natural object classification an d segmentation, CNN has been a much better tool compared to all other previously implemented tools . The broader aim in mind was to develop a M.L. model that could recognize people's handwriting. However, as we began develop ing the model we realized that the topic in hand was too tough and would require tremendous data to learn. Example to accurately classify a cursive handwriting will be very tough. Thus we settled on classifying a given handwritten di git image as the required digit using three different algorithms and consequently testing its accuracy.

2. LITERATURE SURVEY

2.1 Existing problem

Hermans et al. [1] have addressed the MNIST handwritten digit classification problem. In this context, 10 iteration s are used for each image in the MNIST dataset; in other words, each input digit is repeated for 10 masking periods. In their experiments, the authors focused on both an MNIST handwritten digit classification dataset, and a TIMIT phoneme classification dataset. In both MNIST and TIMT datasets, the authors found that optimizing the input encoding can make great improvements over random masks.
Mohapatra et al. [2] proposed a new method for classifying MNIST handwritten digit images. In their new method, the authors used the discrete cosine space-frequency transform to extract image features and artificial neural network classifiers to solve the classification problem. In order to reduce the computational cost, the authors proposed to normalize all the images of the MNIST handwritten digit dataset and exclude undesirable boundary pixels.

Kussul and Baidyk [3] proposed a new neural classifier limited receptive area (LIRA) for MNIST handwritten digit images classification. In the LIRA classifier, the sensor layer is followed with the associative layer, and the train able connections are used to connect the associative layer with the output layer. Experiments with MNIST handwritten digit images show that the LIRA classifier has achieved a classification accuracy of 99.41%. In order to classify MN IST handwritten digit images,

Ahlawata and Choudharyb [4] proposed to build a hybrid classification model by integrating convolutional neural networks and support vector machines (SVM). In this context, the authors used convolutional neural networks to extract the features of the image, while SVM was used as a binary classifier. Based on experimental results the authors have achieved a classification accuracy of 99.28%.

Chazal et al. [5] proposed to use identical network topologies to compare between two weight optimization methods using MNIST handwritten digit classification database. In the first weight optimization methods, the authors use the extreme learning machine algorithm. While back propagation algorithm is used in the second weight optimization methods. Based on their experimental results, the authors found that the weight optimization method that uses the extreme learning machine is much faster than the one that uses the back propagation algorithm.

In [6], Ma and Zhang adopted deep analysis with multi-feature extraction to build a handwritten digit classification method. In order to exclude negative information and maintain relevant features, the images of various sizes were normalized, and projection features were extracted from pre-processed images. Distribution features and projection features are also used to classify MNIST handwritten digit datasets.

In [7], Lopes et al. addressed the MNIST handwritten digit classification problem and used the classifier optimum-path forest to classify this dataset. The authors used the signature of the characters as a feature extractor and the optimum-path forest algorithm as a handwritten digits classifier. According to the experimental results, the authors achieved an average accuracy of 99.53%.

Aly and Allotairi [8] proposed a new unsupervised deep convolutional self-organizing maps network where an MN IST handwritten digit database is used to evaluate the proposed network. The authors used 2D self-organizing maps to extract the features of the image. In their proposed network, the authors partitioned the input image (28 × 28) into s mall patches and represented each small patch by N-coordinates of the winner neuron in the deep convolutional self organizing maps network. The authors have achieved remarkable results on the MNIST handwritten digits dataset. Nevertheless, they failed to achieve good results with the data augmentation technique.

Man et al. [9] proposed to apply a single-hidden layer feed forward network to classify MNIST and USPS handwritten digit images. In this context, the authors adopted the batch learning type of least-squares to improve the input and output weights. In order to minimize the sensitivity of the single-hidden layer feed forward network, the authors proposed to adjust the regularization parameters.

Supervised learning in spiking neural networks was adopted in [10] for handling the MNIST handwritten digit recognition problem. In this network, the authors have encoded all data and processed this data in the spike domain at sparse biological spike rates. In this sense, the authors claimed that utilizing the precise spike time for classification resulted in better classification accuracy. Based on the experimental results, the authors achieved a classification accuracy of 98.17%.

To correctly represent the data in a feature space, Cardoso and Wichert proposed to use the output of a biologically inspired model for visual recognition as a feature space. In this aspect, the output of a biologically inspired model is used to train the linear classifier, and MNIST and USPS handwritten digits datasets are used to evaluate the proposed model. In the training process, the authors proposed to reiterate the unsupervised learning 10 times and retain the best by cross-validation on the training set to select the stimuli for simple cell layers.

Niu and Suen proposed to combine two different artificial neural network classifiers, convolutional neural network and support vector machine. The authors used convolutional neural network as feature extractor to extract features from the raw images and used support vector machine as a classifier to classify the MNIST handwritten digit data base. The authors achieved a recognition rate of 94.40%. However, the complexity of this hybrid model makes it impractical for some neural network applications such as self-driving cars. Based on and support vector machine,

Lauer et al. proposed a new handwritten digits recognition method. In this method, the LeNet5 convolutional neural network is used as a feature extractor, and support vector machine is used as a classifier. In this aspect, the authors used affine transformations and elastic distortions to improve classification accuracy. However, the combination of these two networks made this method quite complex, which contributed to slowing the network speed. In the literature, there is a large amount of CNN classification algorithms. However, most of these algorithms have not taken the following points into consideration: (1) appropriate filter size selection, (2) data preparation, (3) limitations in datasets, and (4) noise. For these reasons, they have failed to achieve remarkable enhancements in the image classification accuracy

2.2 References

[1] Machine learning models for mathematical symbol recognition: A stem to stern literature analysis - Vinay Kukreja & Sakshi

[2] A Novel Method for the Recognition of Isolated Handwritten Arabic Characters - Ahmed Sahlol, Cheng Suen

[3] A novel method for offline handwriting-based writer identification - Zhenyu He; Bin Fang; Jianwei Du; Yuan Yan Tang; Xinge You

[4] A Novel Method for Recognition of Persian Alphabet by Using Fuzzy Neural Network - Mohammad Mehdi Motahari Kia; Jafar A. Alzubi; Mehdi Gheisari; Xiaobo Zhang; Mohamadtaghi Rahimi; Yongrui Qin

[5] K. Gaurav and Bhatia P. K., “Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition”, 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2014.

[6] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., “Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2014

[7] Reena Bajaj, Lipika Dey, and S. Chaudhury, “Devnagari numeral recognition combining decision of multiple connectionist classifiers”, Sadhana, Vol.27, part. 1, pp.-59-72, 2011.

[8] U. Pal, T. Wakabayashi and F. Kimura, “Handwritten numeral recognition of six popular scripts,” Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2010.

[9] Ishani Patel,Virag Jagtap ,Ompriya Kale ,“A Survey on Feature Extraction Methods for Handwritten Digits Re cognition”, IJCA 0975 – 8887), Volume 107 – No 12,Dec (2015).

[10] Bharatratna P. Gaikwad, Ramesh R. Manza, Ganesh R. Manza, "Automatic Video Scene Segmentation to Separate Script and Recognition", Advances in Intelligent Systems and Computing Volume 328, pp 225-235, (2015).

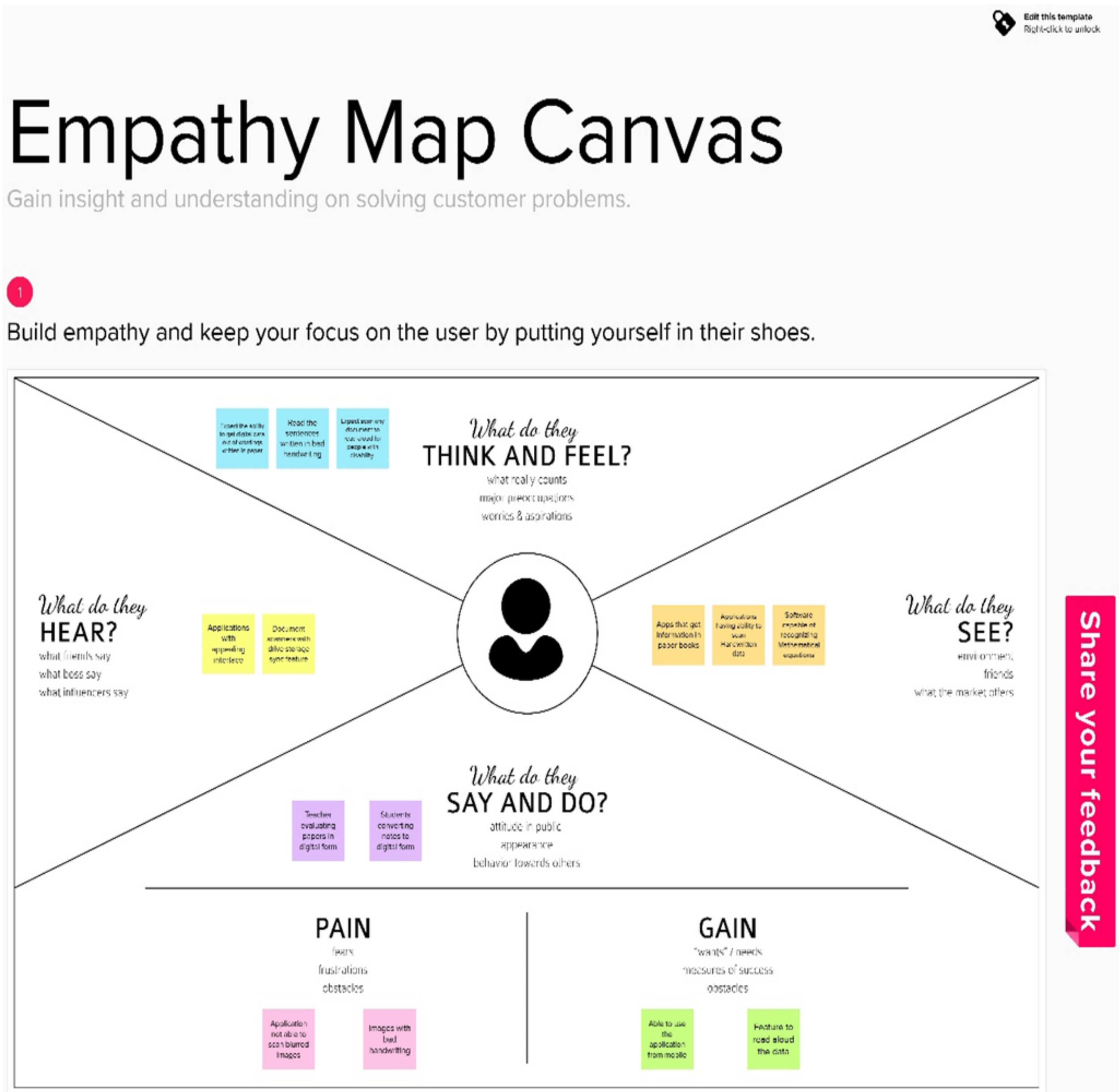
2.3 Problem Statement Definition

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different s apes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using PyTorch library over the MNIST dataset to recognize handwritten digits. Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defenses, etc, and classify. them into 10 predefined classes (0-9). This has been a Content of bottomless- exploration in the field of deep literacy. Number recognition has numerous operations like n umber plate recognition, postal correspondence sorting, bank check processing, etc .

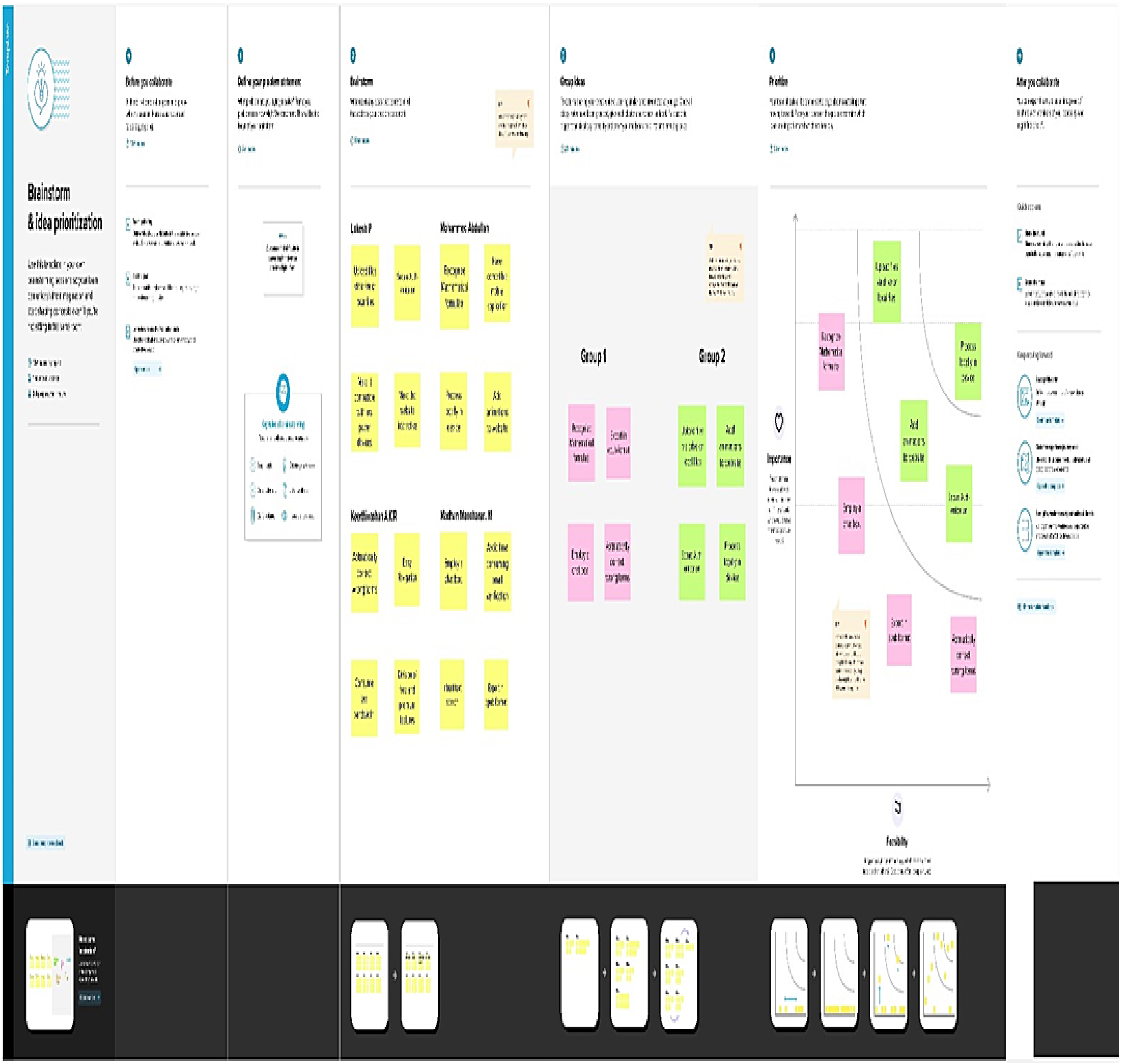
In Handwritten number recognition, we face numerous challenges . because of different styles of jotting of different peoples as it is not an Optic character recognition. This exploration provides a comprehensive comparison between different machine literacy and deep literacy algorithms for the purpose of handwritten number recognition. For this, we've used Support. Vector Machine, Multilayer Perceptron, and Convolutional. Neural Network. The comparison between these algorithm s is carried out on the base of their delicacy, crimes, and .testing- training time corroborated by plots and maps that have been constructed using matplotlib for visualization.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

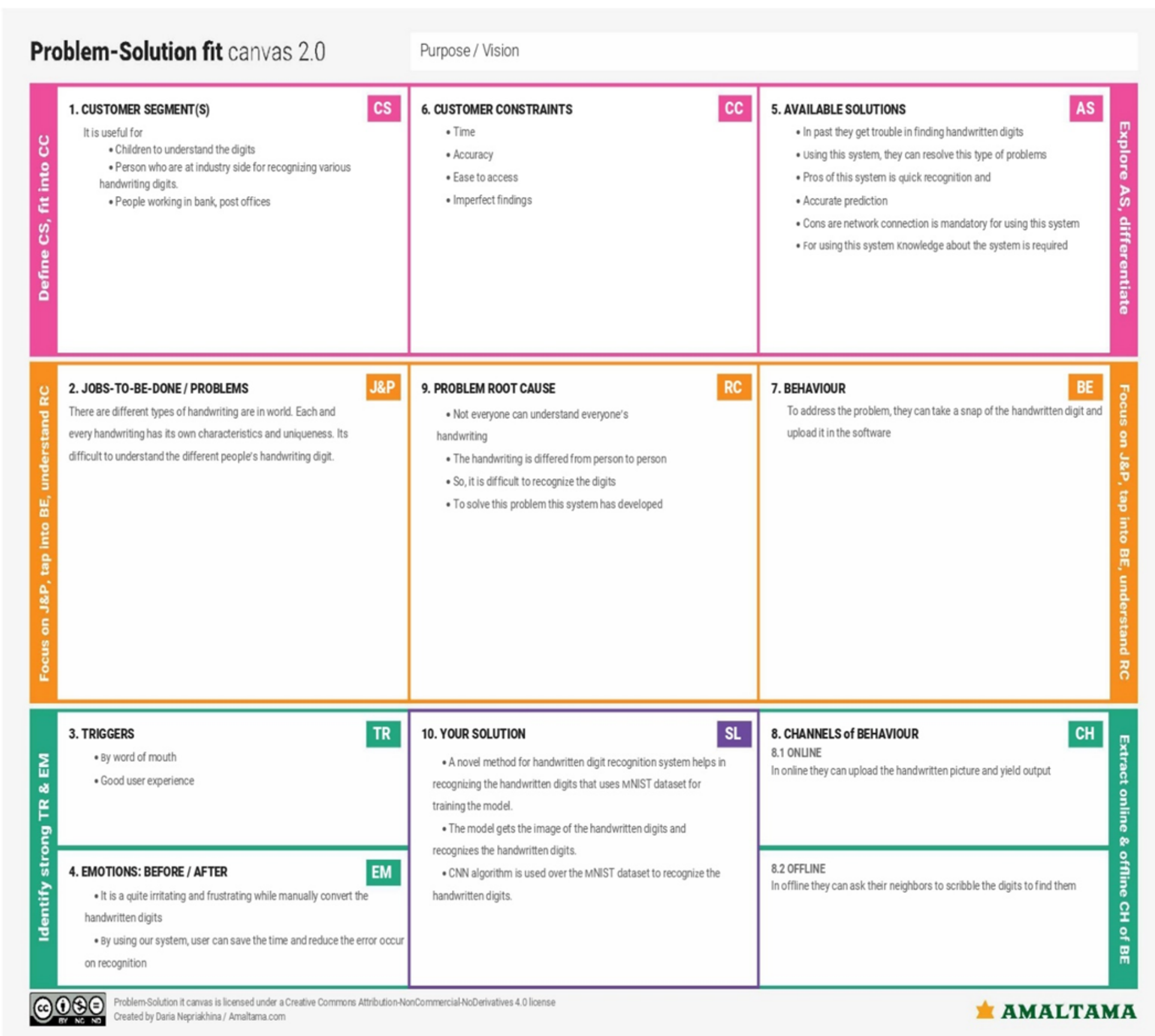


3.3 Proposed Solution

S. No.	Parameter	Description
1.	**Problem Statement**	**The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image.**
2.	**Idea / Solution description**	**We came up with a solution that recognizes the handwritten digits by means of a deep learning model.**
3.	**Novelty / Uniqueness**	**The system recognizes input given by user in a precise and efficient manner.**
4.	**Social Impact / Customer Satisfaction**	**The aim of a handwriting digit recognition system is to convert handwritten digits into machine readable formats. The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits.**

5.	**Business Model (Revenue Model)**	**Pay per use – each time a person needs the service he can avail it by paying for the use.**
6.	**Scalability of the Solution**	**It can be implemented using any we framework and can be made available to everyone in need.**

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	It is necessary to register
FR-2	Uploading the image	Please upload a handwritten digit image in the format provided
FR-3	Using a web browser	Digit recognition requires a desktop or mobile browse

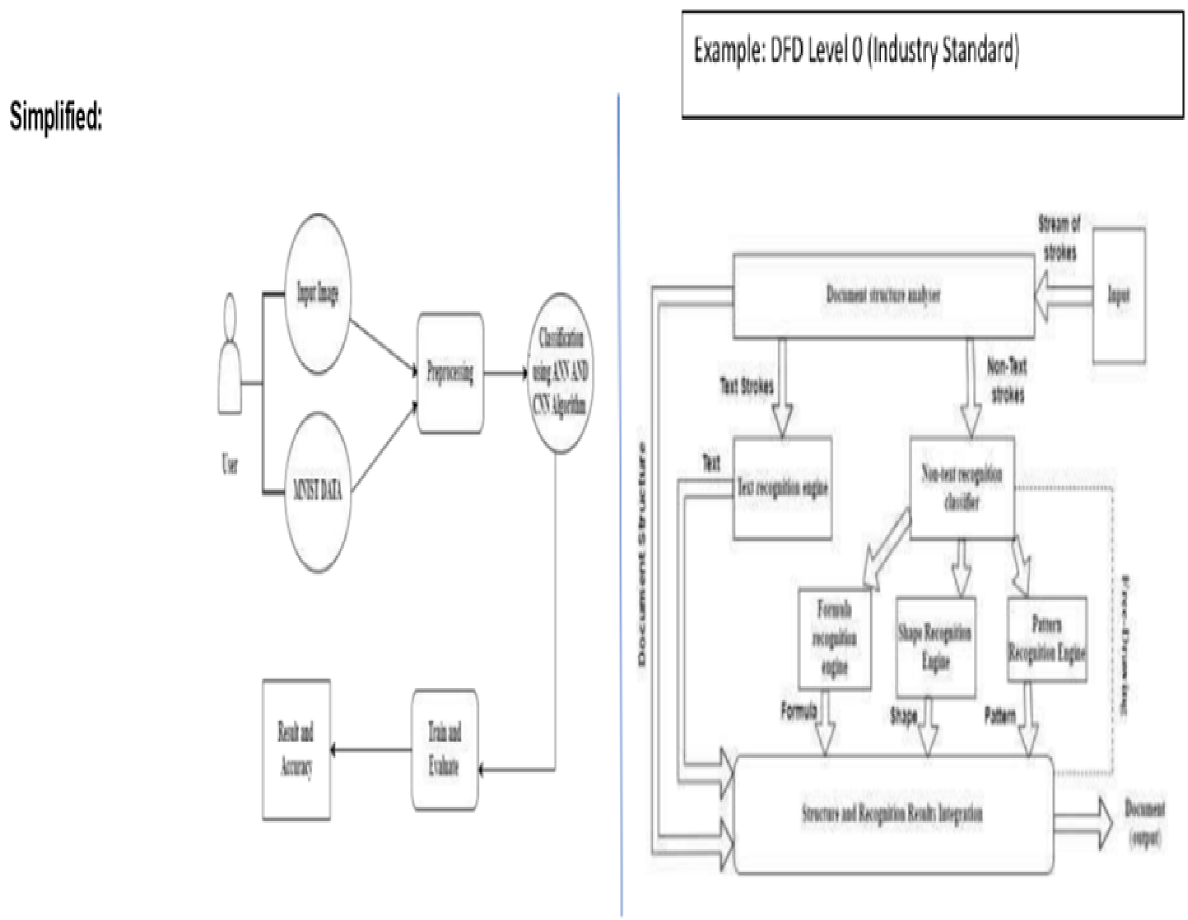
4.2 Non - Functional requirement

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Handwritten recognition digits are easy to get and simple to understand.
NFR-2	Security	Our application does not take any security measures.
NFR-3	Reliability	Be able to endure the long periods of time without errors.
NFR-4	Performance	The performance of a lightweight application.

5. 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

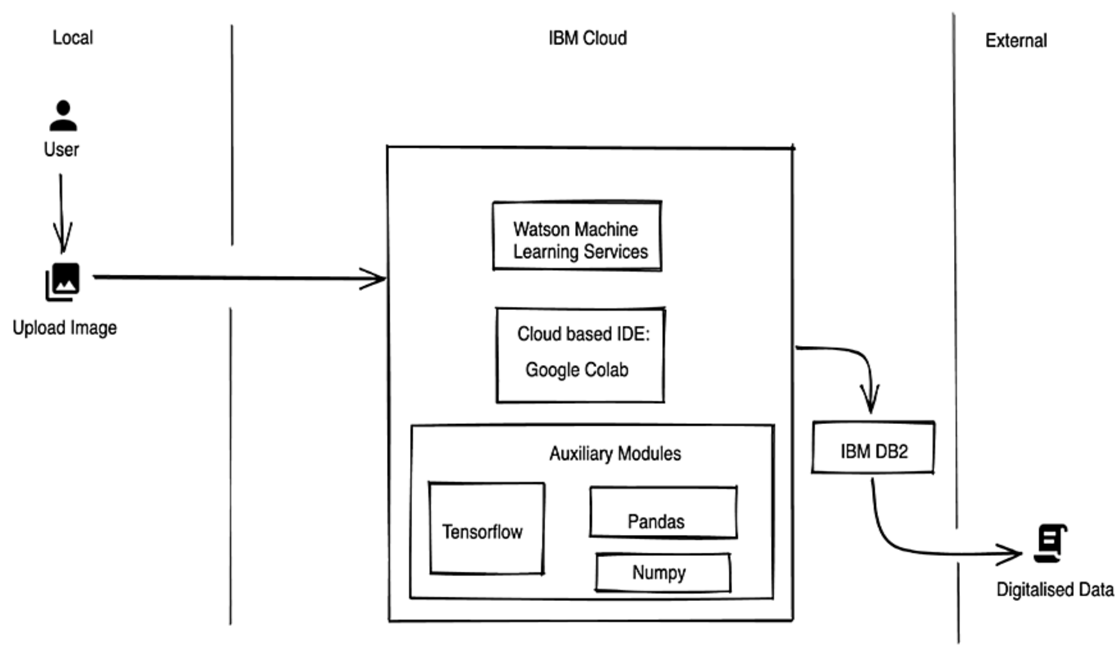
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- When we used models pretrained on unrelated Image Net dataset for the construction of the ensemble architectures

- It significantly enhanced the performance on detecting PD compared to untrained models.

- Our finding suggests a promising direction, where unrelated training data can be considered when insufficient or no training data is available for a particular application.

Example - Solution Architecture Diagram :



5.3 User Stories

To list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User (subject that submits the photograph)	Uploads image	USN-1	Users can upload the pictures to the website to achieve a desired result	We can upload an image	High	Sprint-1
Examination	Image processing	USN-1	Users may examine the projections and outcomes with accuracy.	We can get results instantly.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and 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, the dataset can be collected from a variety of sources with different handwritings.**	**10**	**Low**	**Lokesh.P, Madhan Manoharan**
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**	**Keerthivashan, Madhan Manoharan**

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**	**Mohammed Abdullah, Lokesh.P**
Sprint-2	**Add CNN layers**	**USN-4**	**Creating the model and adding the input, hidden, and output layers to it.**	**5**	**High**	**Mohammed Abdullah, Keerthivashan**
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**	**Madhan Manoharan, Keerthivashan**
Sprint-2	**Train & test the model**	**USN-6**	**As a user, let us train our model with our image dataset.**	**6**	**Medium**	**Mohammed Abdullah, Lokesh.P**
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**	**Madhan Manoharan, Keerthivashan**

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**	**Lokesh.P,** **Madhan Manoharn**
Sprint-3		**USN-9**	**As a user, I can know the details of the fundamental usage of the application**	**5**	**Low**	**Mohammed Abdullah, Madhan Manoharan,**
Sprint-3		**USN-10**	**As a user, I can see the predicted / recognized digits in the application.**	**5**	**Medium**	**Madhan Manoharan, Keerthivashan**
Sprint-4	**Train the model on IBM**	**USN-11**	**As a user, I train the model on IBM and integrate flask/Django with scoring**	**10**	**High**	**Madhan Manoharan, Keerthivashan, Mohammed Abdullah, Lokesh.P**

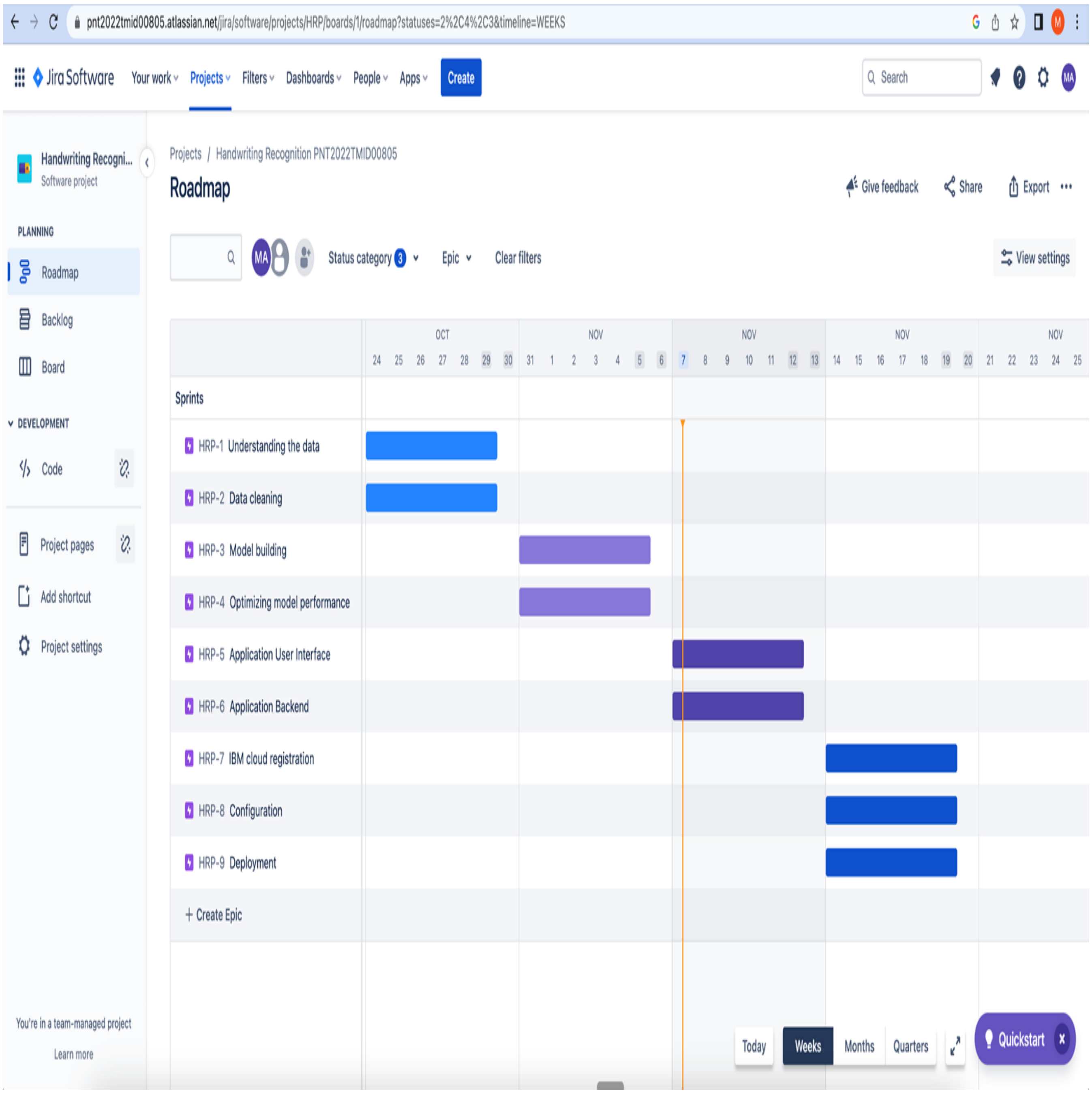
			end point			
Sprint-4	**Cloud Deployment**	**USN-12**	**As a user, I can access the web application and make the use of the product from anywhere**	**10**	**High**	**Mohammed Abdullah, Madhan Manoharan**

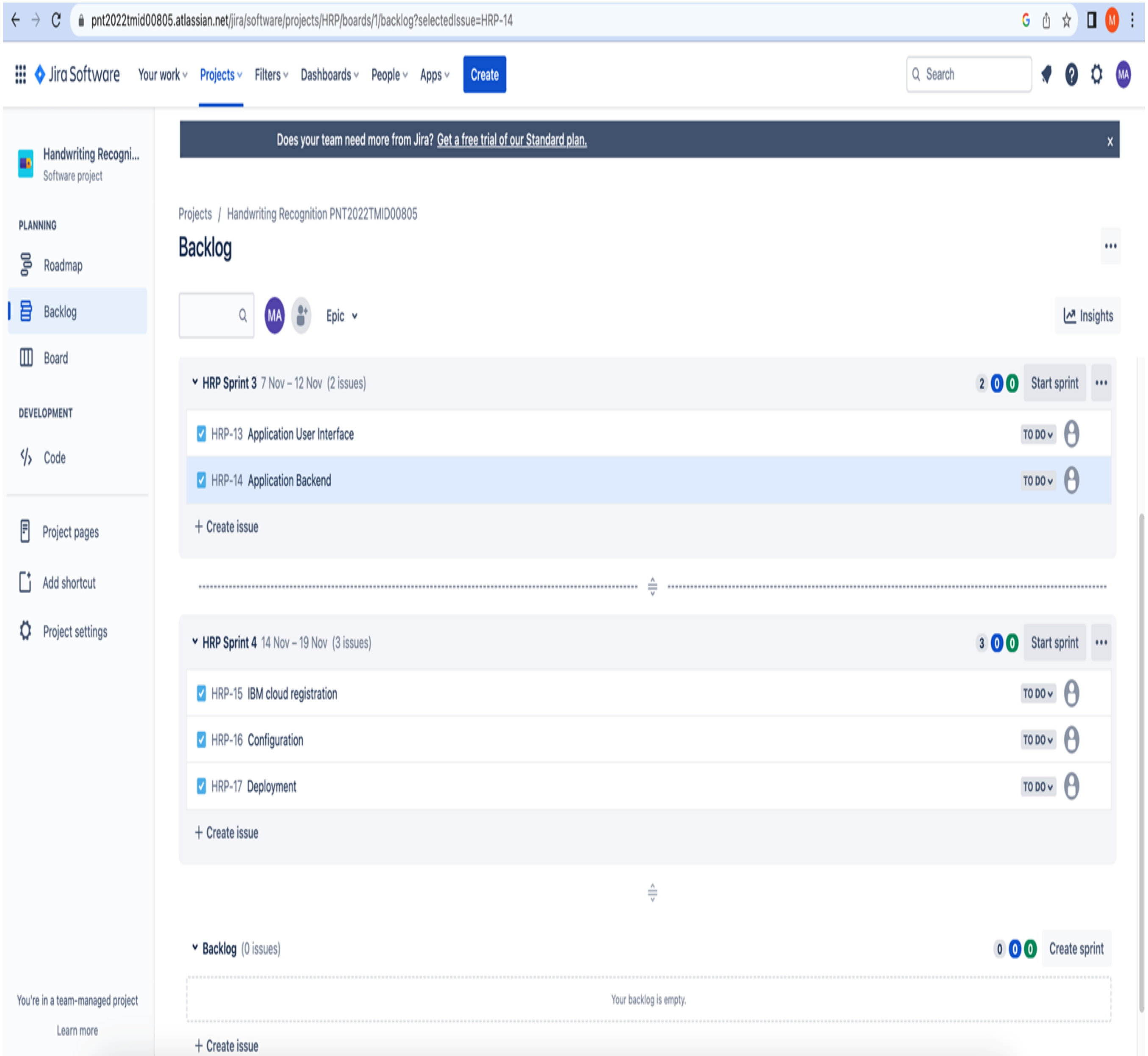
6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	**Understanding thedata**	**USN-1**	**Import MNISTdata for handwritten digitrecognition prediction, and analyse thedata**	**8**	**Medium**	**Keerthivashan, Madhan Manoharan**
Sprint-1	**Data cleaning**	**USN-2**	**Reshape the data, and apply one hot encodingto data**	**12**	**High**	**Mohammed Abdullah, Lokesh.P**
Sprint-2	**Model building**	**USN-3**	**Add CNN layers to train the model by**	**8**	**High**	**Keerthivashan, Madhan Manoharan**

			compiling it.			
Sprint-2	Optimizing model performance	USN-4	Test the model after saving, and observethe metrics to retrain if required efficiency is not met.	12	High	Mohammed Abdullah, Lokesh.P
Sprint-3	Application UserInterface	USN-5	Create an Html files for home page and outputresult, with CSS styling.	8	Medium	Lokesh.P, Madhan Manoharan
Sprint-3	Application Backend	USN-6	Create backendusing flask	12	High	Keerthivashan, Mohammed Abdullah
Sprint-4	IBMcloud registration	USN-7	Register forIBM Cloud	4	Low	Lokesh.P, Madhan Manoharan
Sprint-4	Configuration	USN-8	Train the model in IBM cloud	8	Medium	Mohammed Abdullah, Keerthivashan
Sprint-4	Deployment	USN-9	Deploy in the trained model in cloud	8	Medium	Madhan Manoharan, Lokesh.P

6.3 Reports from Jira




pnt2022tmid00805.atlassian.net/jira/software/projects/HRP/boards/1/backlog?selectedIssue=HRP-14
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
MA
Does your team need more from Jira? Get a free trial of our Standard plan.
Handwriting Recogni...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Handwriting Recognition PNT2022TMID00805
Backlog
Epic
Insights
HRP Sprint 3 7 Nov – 12 Nov (2 issues)
2 0 0
Start sprint
HRP-13 Application User Interface
TO DO
HRP-14 Application Backend
TO DO
+ Create issue
HRP Sprint 4 14 Nov – 19 Nov (3 issues)
3 0 0
Start sprint
HRP-15 IBM cloud registration
TO DO
HRP-16 Configuration
TO DO
HRP-17 Deployment
TO DO
+ Create issue
Backlog (0 issues)
0 0 0
Create sprint
Your backlog is empty.
+ Create issue

7 . CODING & SOLUTIONING

7.1 INFORMATION CENTER

Base.html

```
<!DOCTY
PE html>
        <html lang="en">

        <head>
          <meta charset="UTF-8">
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
          <title>Handwrtten Digit Recognition</title>

          <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"


            integrity="sha384-
        Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
        crossorigin="anonymous">
          <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.3/css/bulma.min.css">

          <link rel="stylesheet" href="{{ url_for('static', filename='index.css') }}" />

          <script type="text/javascript" src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

          <script type="text/javascript" src="{{ url_for('static', filename='index.js') }}"></script>


        </head>

        <body>

          {% block content %}{% endblock %}

        </body>
```

```
</html>
```

Index.html -

```
{% extends
"base.html"
%}
{% block content %}

<div class="p-5 text-center">
  <h1 class="title">A Novel Method for Handwritten digit Recognition System</h1>

  <div class="box text-center">
    <div class="p-5 box text-center">
      <div class="p-5 box text-center">
        <div class="p-5 box text-center">
          <div class="box text-center">
            <form class="p-3 text-center" action="/" method="POST" enctype="multipart/form-data">

              <b>
                <input class="form-control" type="file" name="imagefile"> <br><br>
                <input class="btn btn-primary mt-3" type="submit" value="Predict Image">
              </b>
            </form>
            <p>Upload the Image to get digital output</p>
          </div>
          <div class="box text-center">
            <h4>The results will be displayed here</h4>
            <b>
              <p>{{pred}}</p>
            </b>
          </div>
          <div class="box text-center">
            <p>CREDITS:</p>
```

```
<h4>
  <b>Mohammed Abdullah - Lokesh.P - Keerthivashan.A.K.R - Madhanmanoharan.M</b>

</h4>
```

```
<p>TEAM ID:</p>
<h4>
  PNT2022TMID00805
</h4>
 </div>
 </div>
 </div>
 </div>
 </div>
</div>

{% endblock %}
```

Index.css

```
.main-
container {

  display: flex;
  flex-direction: column;
  justify-content: center;
  align-content: center;
  align-items: center;
}

.box {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  align-self: center;
}
```

```
.box p {
    margin-top: 2rem;
}

.title {
    text-align: center;
}
```

App.py

```
from flask import
Flask,
render_template,
request

import make_prediction

app = Flask(__name__)


@app.route('/', methods=['GET'])
def main():
    # return "<h1>Test</h1>"
    return render_template('index.html')


@app.route('/', methods=['POST'])
def predict():
    imageFile = request.files['imagefile']
    imgPath = "./images/"+imageFile.filename
    imageFile.save(imgPath)
    # arr = np.array([[data1, data2, data3, data4]])
    pred = make_prediction.predict()
    pred = "The digitalised output is"+pred
    return render_template('index.html', prediction=pred)
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

make_prediction.py

```
from tensorflow.keras.utils
import load_img

from tensorflow.keras.utils import img_to_array
from keras.models import load_model



def argmax(a):
    return list(a).index(max(a))



def load_image(filename):
    test_img = load_img(filename, grayscale=True, target_size=(28, 28))

    img = img_to_array(test_img)
    img = img.reshape(1, 28, 28, 1)
    img = img.astype('float32')
    img = img / 255.0
    return img, test_img



def predict():
    img, test_img = load_image('sample_image-768x763.png')
    model = load_model('model/model.h5')

    predict_value = model.predict(img)
    digit = argmax(predict_value)
    # print(digit)
    return predict_value
```

Requirements.txt

```
flask==0.1.0
numpy==1.19.3
tensorflow==2.8.0
```

8.Testing

8.1 Test Cases

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issuesof the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This reportshows the numberof 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	9	3	0	6	18
Duplicate	0	1	5	0	6
External	6	4	2	0	12
Fixed	15	8	5	6	34
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	30	16	14	13	73

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	15	0	0	15
Security	4	0	0	4
Outsource Shipping	3	0	0	3

Exception Reporting	5	0	0	5
Final ReportOutput	4	0	0	4
Version Control	1	0	0	1

8.2 User Acceptance Testing

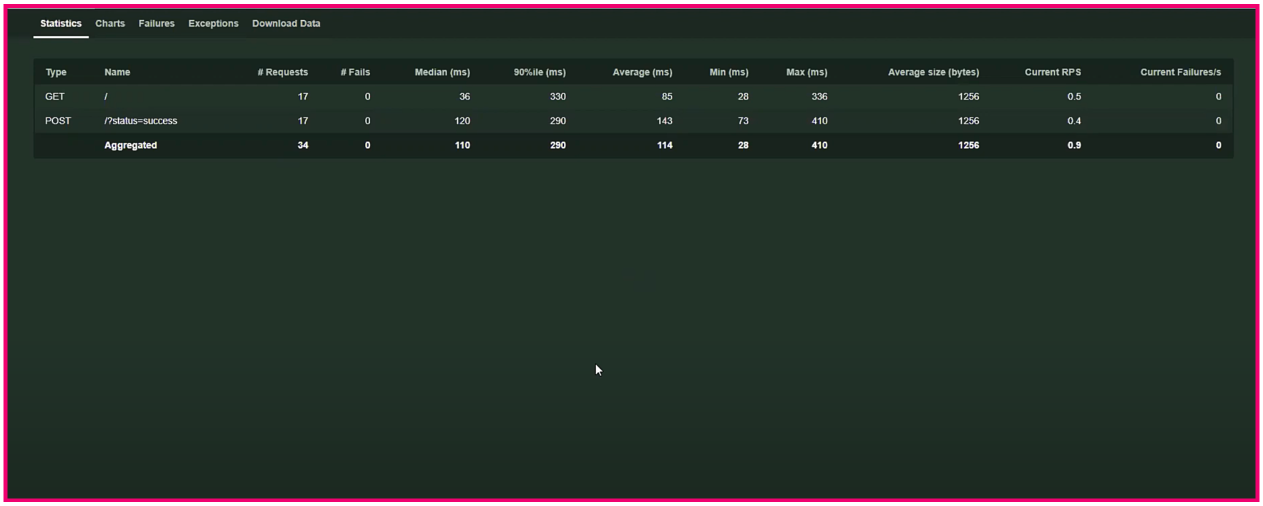
Date	3-Nov-22
Team ID	PNT2022TMID00805
Project Name	Project - A Novel Method for Han
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
UPLDIMG_TC_OO1	Functional	Home Page	verify whether the user is uploading image or not	A image with any letter	1. click on the upload image 2. upload the image from local storage	An image is in file	if the the uploaded is not image it should display "the format is supported"	Working as expected	Pass	Steps are clear to understand	Y		
ConnectDB_TC_OO2	Backend connection	Home Page	Verify whether the database is connected or not	Database	1. establish connection the between the Database and Web app 2. confirm the connection is done		connection is established	Working as expected	Pass	Steps are clear to follow	Y		
OutPut_TC_OO3	Functional	Home page	Check whether the AI model is recognising the letter	AI model	1. upload the image 2. whether the AI model is running the image	number 5 in image	AI model should recognising the image	Working as expected	Pass	legimate steps are given	N		
correctletter_TC_O04	Functional	Output Page	AI model is identifying the right digit	working web app	1. upload the image 2. whether the AI model is running the image	number 6 in image	right letter is deteting from letter	workin as wxcepted	pass	output is fenerated	Y		

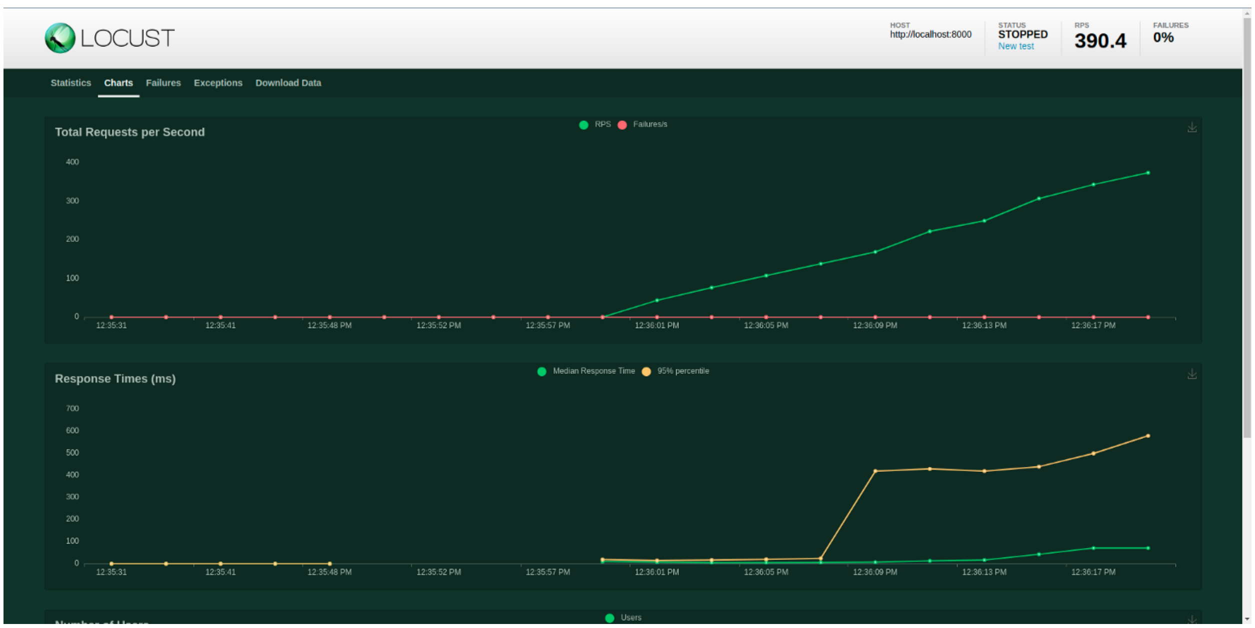
9.Results

9.1 Performance Metrics

For the performance testing load testing was performed. Locust was used to test the application.



Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	17	0	36	330	85	28	336	1256	0.5	0
POST	/?status=success	17	0	120	290	143	73	410	1256	0.4	0
	Aggregated	**34**	**0**	**110**	**290**	**114**	**28**	**410**	**1256**	**0.9**	**0**



10. Advantages & Disadvantages

10.1 Advantages

1) the system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as

the writing style;
2) the generative models can perform recognition driven segmentation;
3) the method involves a relatively small number of parameters and hence training is relatively easy and fast
4) unlike many other recognition schemes, it does not rely on some form of pre-normalization of input images, but can handle arbitrary scalings, translations and a limited degree of image rotation.

10.2 Disadvantages

The disadvantage is that it is not done in real time as a person writes and therefore not appropriate for immediate text input.

11.Conclusion

There are numerous applications for handwritten digit recognition, including medical, banking, student management, and taxation.The handwritten image is classified using classifiers like KNN, SVM, and CNN. as per the review, CNN is providing better performance than others. Stages of HDR using CNN classifier is discussed in this paper. MNIST dataset consist of handwritten numbers from 0-9 and it is a standard dataset used to find performance of classifiers. HDR consists of three different stages. First is preprocessing where dataset is converted into binary form and image processing has been applied on it. Second stage is segmentation where the image is converted into multiple segments. Third stage is feature extraction where features of image are identified. Last stage is classification where
classifiers like KNN, SVM, CNN are used. Results of HDR is improved a lot by using CNN classifier but it can be improved further in terms of complexity, duration of execution and accuracy of results by making combination of classifiers or using some additional algorithm with it.

12.Future enhancement

- Facial imotion Recognition is used in car board system depending on information of the mentality of the driver can be provided to the system to initiate his/her and the customer safety.
- Development of a facial emotion recognition system implementing the computer visions

and enhancing the advanced feature extraction and classification.This system can be used in digital in security systems which can identify a person in any form of expression he presents himself.

- The future development of the applications based on algorithms of deep and machine learning is practically boundless. In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve
- the solutions to many problems. In future, the application of these algorithms lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people, we can use these algorithms in hospitals application for detailed medical diagnosis, treatment and monitoring the patients, we can use it in surveillances system to keep tracks of the suspicious activity under the system, in fingerprint and retinal scanners, database filtering applications, Equipment checking for national forces and many more problems of both major and minor category. The advancement in this field can help us create an environment of safety, awareness and comfort by using these algorithms in day-to-day application and high-level application

13.Appendix

Github

https://github.com/IBM-EPBL/IBM-Project-3542-1658575041

Project Demo Link


Recognition of digits in:
1. Bills
2. Vehicle license plates
3. Postal systems
4. Historical writings
5. Banks