# A Novel Method For A Handwritten Digit Recognition System

**PROJECT REPORT**

**BASED ON A NOVEL METHOD FOR A HANDWRITTEN DIGIT RECOGNITION SYSTEM**

**Team ID : PNT2022TMID14753**

**Kumaraguru College of Technology**

**Coimbatore-641049.**

**Team Members**

**Akash S(Team Lead) –** 7179KCTKCTKCTKCTKCTKCTKCT19BCS201 **(19BCS201)**

**Raghunandan UR –** 7179KCTKCTKCTKCTKCTKCTKCT19BCS211 **(19BCS211)**

**Karthick M –** 7179KCTKCTKCTKCTKCTKCTKCT19BCS207 **(19BCS207)**

**Sibi S –** 7179KCTKCTKCTKCTKCTKCTKCT19BCS070 **(19BCS070)**

## 1. INTRODUCTION

### 1.1 Project Overview

Traditional handwriting recognition algorithms have depended heavily on existing information andcustomized features. It is difficult to train an optical character recognition (OCR) system using these requirements. Deep learning approaches are the main focus of handwriting recognition research, which has recently produced ground-breaking results. However, the exponential increase in the volume of handwritten information and the accessibility of enormous computing capacity necessitates an enhancement in recognition rate and warrant additional study. Convolutional neural networks (CNNs) are the most successful method for resolving handwriting recognition issues because they are very good at understanding the layout of handwritten characters and words in ways that facilitate the automatic extraction of distinctive features. We explore different design alternatives for CNN-based handwritten digit identification in the proposed study, including layer count, receptive field, stride size, padding, kernel size, and dilution.Too attain accuracy even greater than ensemble architectures, as well as decreased operational complexity and expense, a CNN architecture is presented. Additionally, we provide a suitable arrangement of learning parameters for creating a CNN that enables us to set a new absolute record for categorizing MNIST handwritten digits.

### 1.2 Purpose

One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include form data entry, bank check processing, postal mail sorting, and others. The capacity to create an effective algorithm that recognizes handwritten digits given by users via a tablet, scanne ,and other digital devices is at the issue's core.

## 2. LITERATURE SURVEY

### 2.1 Existing problem

| Paper | Authors | Publish Date | Problem | Conclusion | Accuracy |
|---|---|---|---|---|---|
| A novel method for Handwritten Digit Recognition with Neural Networks | Malothu Nagu, N.Vijay Shankar, K.Annapurna | 2011 | Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition. This is a system widely used in the world to recognize zip codes or postal codes for mail sorting. Different techniques can be used to | Two techniques researched in this paper are Pattern Recognition and Artificial Neural Networks (ANN). Both techniques are defined and different methods for each technique are also discussed. Bayesian Decision Theory, Nearest Neighbor rule, and Linear | Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high |

| | | | | | |
|---|---|---|---|---|---|
| | | | recognize handwritten characters | | |
| A novel method for Handwritten Digit Recognition using Deep Learning | Rohini. M and Dr.D.Surendran | 2019 | Convolution Neural Networks which is one of the important architectures of deep learning. Understanding CNN and applying it to the handwritten recognition system is the major target of the proposed system. There is a reason behind using CNN for handwritten digit recognition. Let us consider a multi-layer feed-forward neural network to be applied to the MNIST dataset, which contains images of 28×28 pixels (roughly 784 pixels) | A convolution neural network considers the mapping of image pixels with the neighborhood space rather than having a fully connected layer of neurons. Convolution Neural Networks have been proven to be a very important and powerful tool in signal and image processing. Even in the fields of computer vision such as handwriting recognition, natural object classification, and segmentation, CNN has been a much better tool compared to all other previously implemented tools. Paper 3: A novel Handwritten Digit Classification System Based o | During training and testing, the accuracy rate reached 94%. |
| A novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach | Ali Abdullah Yahya, Jieqing Tan and Min Hu | 2020 | An enormous number of CNN classification algorithms have been proposed in the literature. Nevertheless, in these algorithms, appropriate filter size selection, data preparation, limitations in datasets, and noise have not been taken into consideration. As a consequence, most of the algorithms have failed to make a noticeable improvement in classification accuracy. | Calculating the size of the ERF helps us to select a typical filter size which leads to enhancing the classification accuracy of our CNN. Secondly, unnecessary data leads to misleading results and this, in turn, negatively affects classification accuracy. To guarantee the dataset is free from any | Gaussian noise with σ = 0.5 to the MNIST dataset. As a result, our CNN algorithm achieves state-of-the-art results in handwritten digit recognition, with a recognition |

| | | | | | |
|---|---|---|---|---|---|
| | | | To address the shortcomings of these algorithms, our paper presents the following contributions: Firstly, after considering the domain knowledge, the size of the effective receptive field (ERF) is calculated. | redundant or irrelevant variables to the target variable, data preparation is applied before implementing the data classification mission. Thirdly, to decrease the errors of training and validation, and avoid the limitation of datasets, data augmentation has been proposed. | accuracy of 99.98%, and 99.40% with 50% noise. |
| A Novel Approach for Handwritten Character Recognition Using K-NN Classifier | Abhay Mishra, Krishan Kumar, Parveen Kumar, and Prakhar Mittal | 2020 | In this digital era, it is crucial to identify the authenticity of the words where the writer's identification becomes a big challenge. This paper highlights an efficient approach to recognizing the character from the handwritten document using a k-nearest neighbor algorithm. | Then, a supervised-learning algorithm is employed to identify the character. From the experimental results, it is observed with our proposed model, we achieved about 92% accuracy for the digits and about 94.15% accuracy for the English alphabet. To see the merits of the proposed model, the comparison is made against the state-of-the-art models | 92% accuracy for the digits and about 94.15% accuracy for the English alphabet. |

## 2.2 References


- J. M. Alghazo, G. Latif, L. Alzubaidi, and A. Elhassan, "Multi-language handwritten digits recognition based on novel structural features," Journal of Imaging Science and Technology, vol. 63, no. 2, 2019, DOI: 10.2352/J.IMAGING.TECHNOL.2019.63.2.020502.

- S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten DigitRecognition Using Convolutional Neural Networks (CNN)," Sensors 2020, Vol. 20, Page 3344, vol. 20, no. 12, p. 3344, Jun. 2020, DOI: 10.3390/S20123344.

- S. Ali, J. Li, Y. Pei, M. S. Aslam, Z. Shaukat, and M. Azeem, "An Effective and Improved CNN-ELM Classifier for Handwritten Digits Recognition and Classification," Symmetry 2020, Vol. 12, Page 1742, vol. 12, no. 10, p. 1742, Oct. 2020, DOI: 10.3390/SYM12101742.

- K. Urazoe, N. Kuroki, T. Hirose, and M. Numa, "Combination of Convolutional Neural Network Architecture and its Learning Method for Rotation-Invariant Handwritten Digit Recognition," IEEJ Transactions on Electrical and Electronic Engineering, vol. 16, no. 1, pp. 161–163, Jan. 2021, DOI: 10.1002/TEE.23278.

- D. Beohar and A. Rasool, "Handwritten digit recognition of MNIST dataset using deep learning state-of-the-art artificial neural network (ANN) and Convolutional Neural Network (CNN)," 2021 International Conference on Emerging Smart Computing and Informatics, ESCI 2021, pp. 542–548, Mar. 2021, DOI: 10.1109/ESCI50559.2021.9396870.

- S. Albahli, M. Nawaz, A. Javed, and A. Irtaza, "An improved faster-RCNN model for handwritten character recognition," Arab J Sci Eng, vol. 46, no. 9, pp. 8509–8523, Sep. 2021, DOI: 10.1007/S13369-021-05471-4.

- A. A. Yahya, J. Tan, and M. Hu, "A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach," Sensors 2021, Vol. 21, Page 6273, vol. 21, no. 18, p. 6273, Sep. 2021, DOI: 10.3390/S21186273.

- K. Senthil Kumar, S. Kumar, and A. Tiwari, "Realtime Handwritten Digit Recognition Using Keras Sequential Model and Pygame," Lecture Notes in Networks and Systems, vol. 177 LNNS, pp. 251–260, 2021, DOI: 10.1007/978-981-33-4501-0_24/COVER.

- G. J. S. Padmashali and D. Kumari, "Handwritten Digit Recognition Using Deep Learning," International Journal of Research in Engineering, Science and Management, vol. 4, no. 7, pp. 182–185, Jul. 2021, Accessed: Sep. 15, 2022. [Online]. Available: http://journals.resaim.com/ijresm/article/view/1026

- V. Agrawal and J. Jagtap, "Convolutional Vision Transformer for Handwritten Digit Recognition," 2022, DOI: 10.21203/rs.3.rs-1984839/v1.

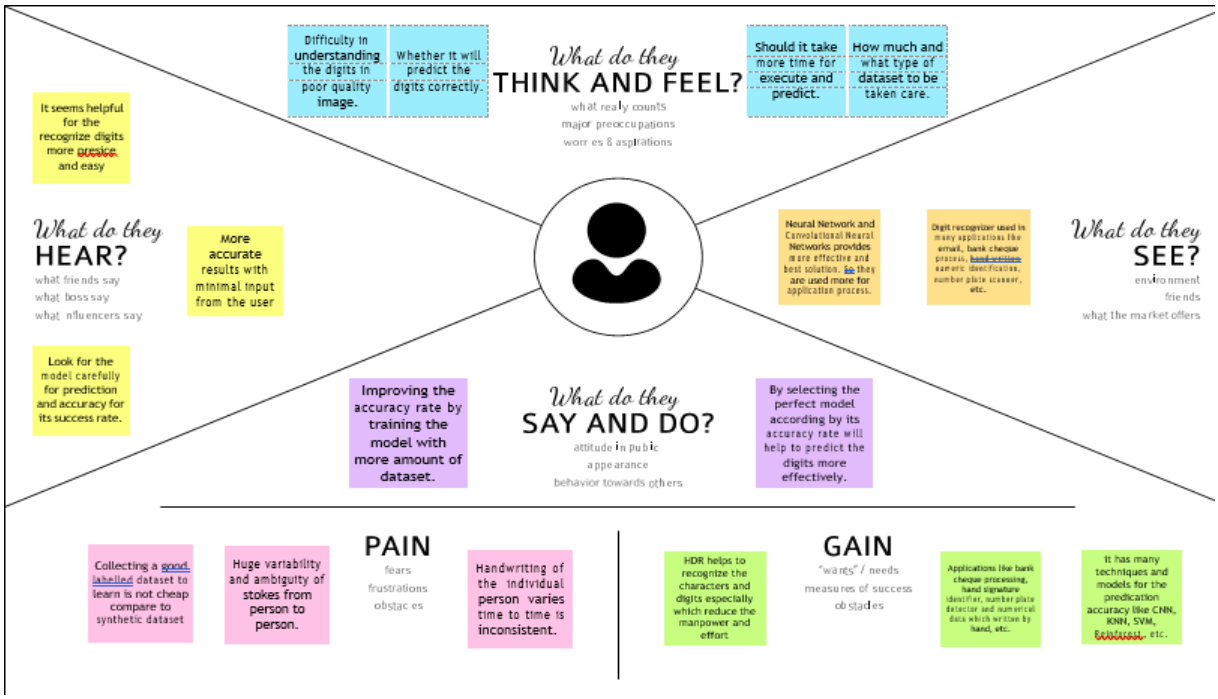

### 2.3 Problem Statement Definition

Character handwriting recognition has been around since the 1980s. Handwritten digit recognition using a classifier offers a wide range of applications, including digital digit recognition on PC devices, recognizing zip codes on mail, handling bank cheque amounts, numeric portions in structures filled out by hand (for example, tax forms), and so on. There areseveral difficulties encountered while attempting to address this problem. The digits are not necessarily the same height, width, orientation, or location concerning the margins. The primary goal was to implement a pattern characterization approach for perceiving handwritten digits using the MNIST data collection of photographs of handwritten digits (0 - 9). Machine Learning provides a variety of approaches for reducing human effort in detecting manually typed numbers. Deep Learning is a technology that educates computers to do what people do naturally: learning via examples. Human efforts in seeing, learning, and recognizing many other areas can be reduced by using deep learning approaches. The machine learns to do classification tasks from images or the text of any document using deep learning. Modelsusing deep learning can achieve state-of-the-art accuracy, outperforming humans.

| Problem Statement (PS) | I am (Customer) | I'm trying to | But | Because | Which makes me feel |
|---|---|---|---|---|---|
| PS-1 | Business Analyst working in a Financial Company | To calculate annual turnover for different departments | Facing difficulties to carry out analytics on handwritten data from customers andworkers | The handwritten digits could be of different sizes and styles according to the ones | Reduces Productivity |
| PS-2 | Student | Sort the collected letters based on the area in which they will be delivered | Facing Difficulties in understanding and organizing the letters | The Handwritten Postal codes are almost unrecognizable and misunderstood postal codes would mean incorrect delivery | Incompetent and Angry |

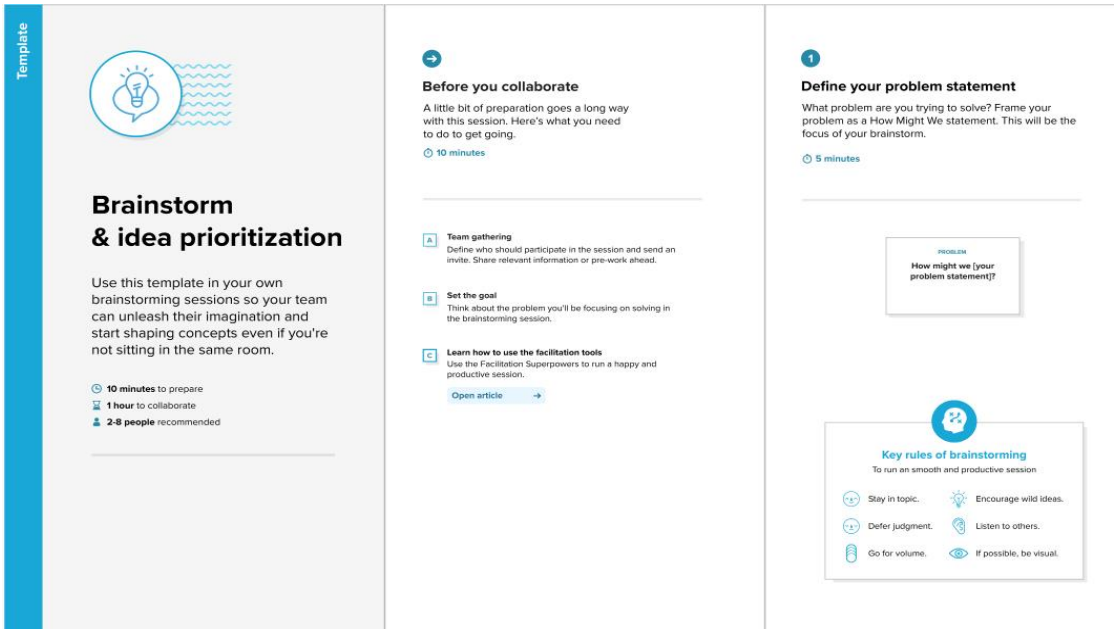
## 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas

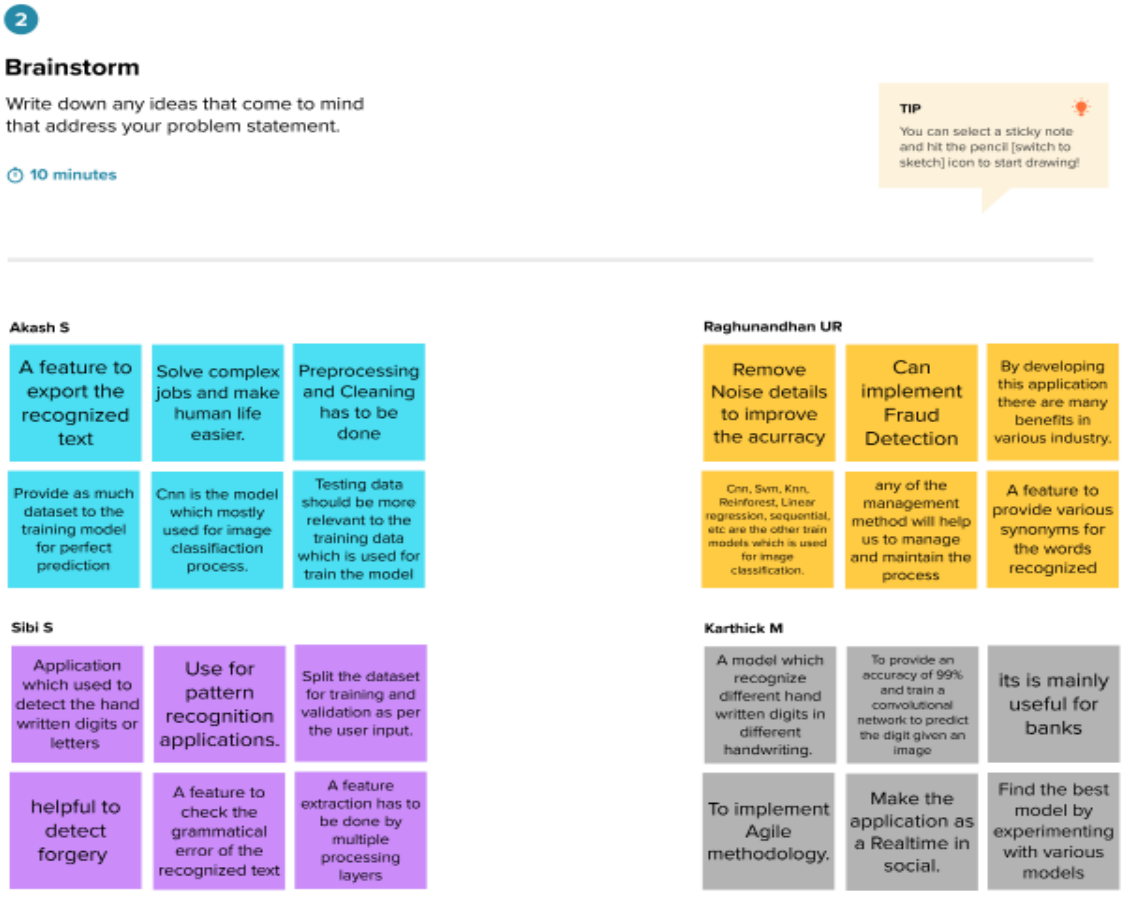


## 3.2 Ideation & Brainstorming

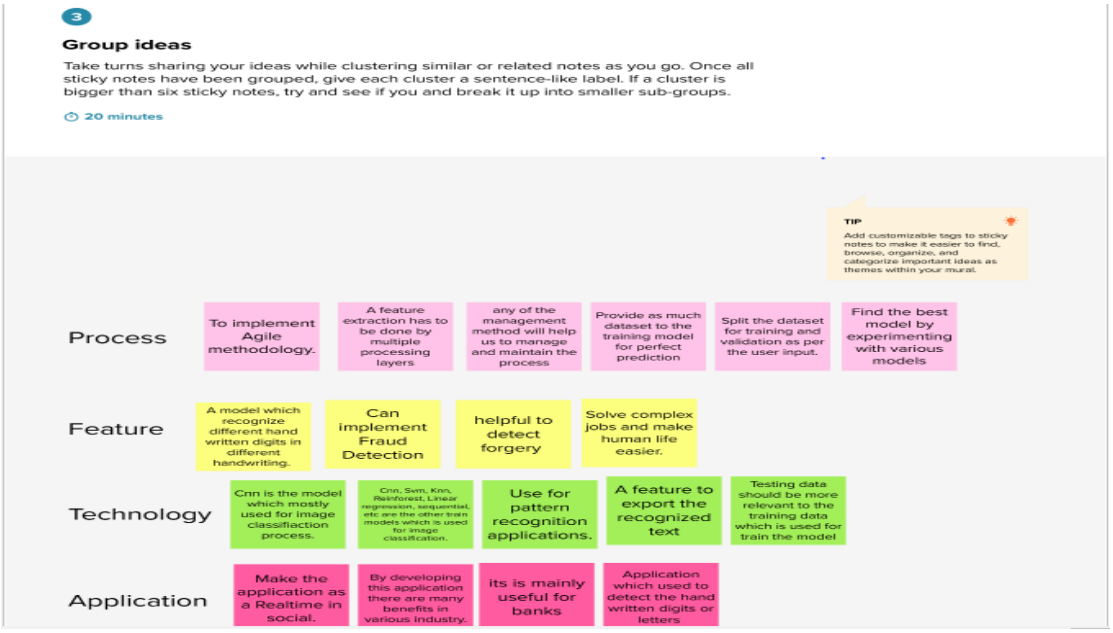
### Step 1



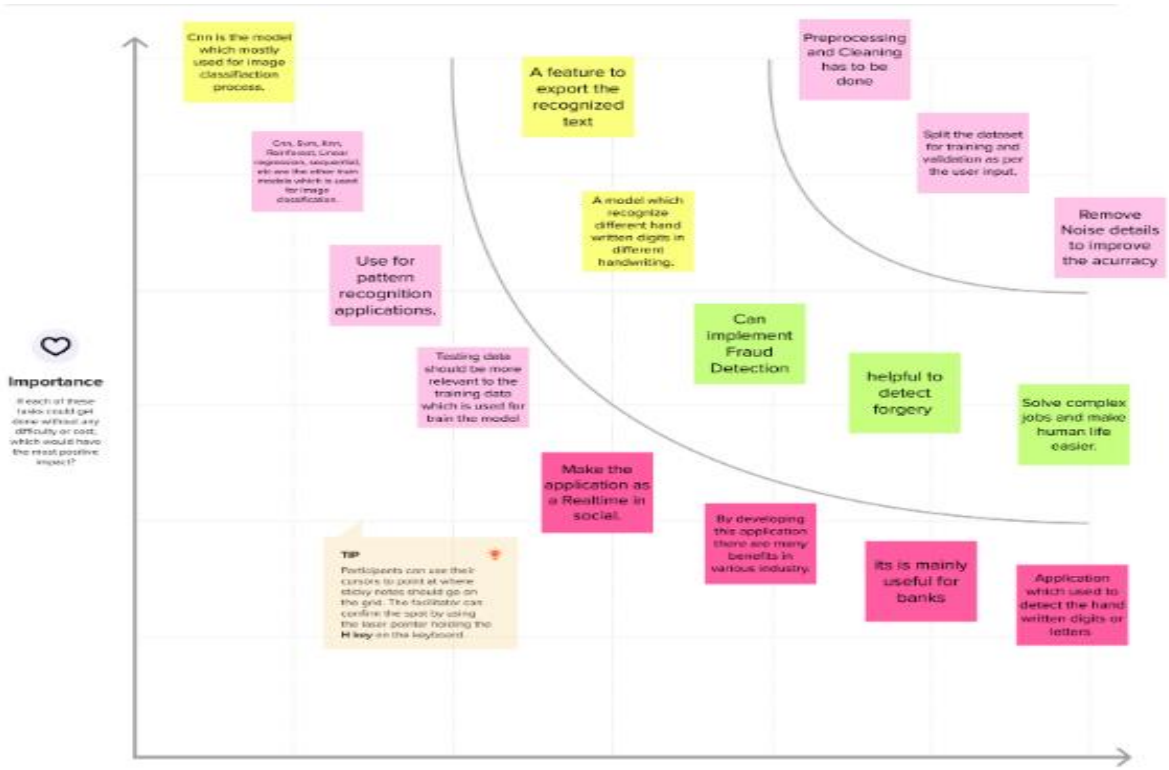

### Step 2



## Step 3


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.
Process
To implement Agile methodology.
A feature extraction has to be done by multiple processing layers
any of the management method will help us to manage and maintain the process
Provide as much dataset to the training model for perfect prediction
Split the dataset for training and validation as per the user input.
Find the best model by experimenting with various models
Feature
A model which recognize different hand written digits in different handwriting.
Can implement Fraud Detection
helpful to detect forgery
Solve complex jobs and make human life easier.
Technology
Cnn is the model which mostly used for image classifiaction process.
Cnn, Svm, Knn, Reinforest, Linear regression, sequential, etc are the other train models which is used for image classifiaction.
Use for pattern recognition applications.
A feature to export the recognized text
Testing data should be more relevant to the training data which is used for train the model
Application
Make the application as a Realtime in social.
By developing this application there are many benefits in various industry.
its is mainly useful for banks
Application which used to detect the hand written digits or letters

## Step 4


Importance
Cnn is the model which mostly used for image classifiaction process.
A feature to export the recognized text
Preprocessing and Cleaning has to be done
Split the dataset for training and validation as per the user input.
Remove Noise details to improve the acurracy
A model which recognize different hand written digits in different handwriting.
Use for pattern recognition applications.
Can implement Fraud Detection
helpful to detect forgery
Solve complex jobs and make human life easier.
Testing data should be more relevant to the training data which is used for train the model
Make the application as a Realtime in social.
By developing this application there are many benefits in various industry.
its is mainly useful for banks
Application which used to detect the hand written digits or letters

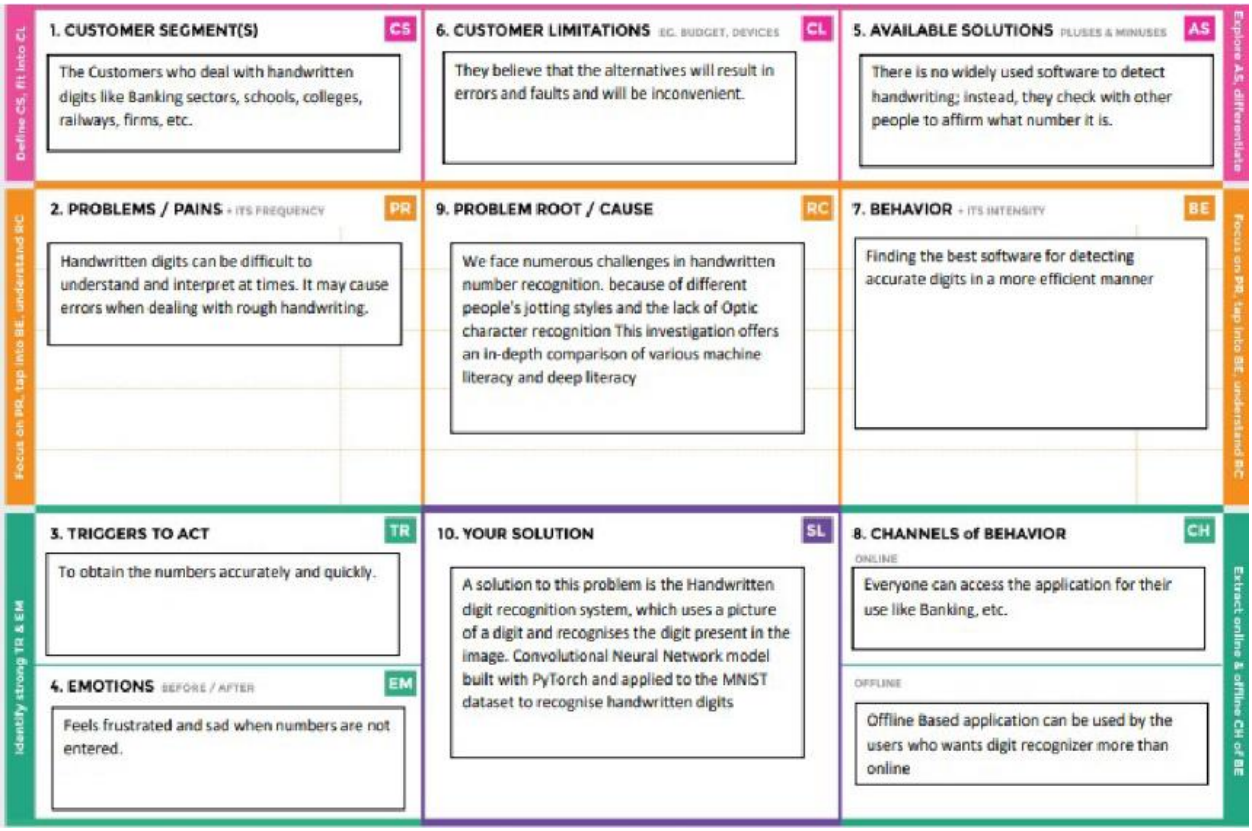
### 3.3 Proposed Solution

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | In the modern world, digit recognition is crucial. It is capable of solving increasingly difficult problems andmaking humans' jobs easier. Handwritten digit recognition is one example. This is a worldwide system for recognizing zip codes or postal codes for mail sorting. Handwritten digit recognition can be accomplished using a variety of approaches. The machine has a difficult duty because handwrittendigits are not flawless and can be generated with a variety of flavors. The solution to this issue is handwritten digit recognition, which uses an image ofa digit and identifies the digit represented in theimage. |
| 2. | Idea / Solution description | Handwritten digit recognition is performed using the MNIST dataset which contains 60,000 training images of handwritten digits from zero to nine and 10,000 images for testing. So, the MNIST dataset has10 different classes. In this project, we are going to implement a handwritten digit recognition application trained using the Convolutional Neural Networks model. In the end, a GUI is built where theuser gives the handwritten digit as input where it is recognized and the result is displayed immediately. |
| 3. | Novelty / Uniqueness | This project introduces an operative strategy for dealing with novelty in the handwritten visual recognition domain. A perfect transcription agent would be able to distinguish known and unknown characters in a picture, as well as determine anyaesthetic variations that may occur inside or between texts. The existence of novelty has shown to be a major stumbling block for even the most |

| | | |
|---|---|---|
| | | robust machine learning-based algorithms for these activities. Novelty in handwritten papers might include, among other things, a change in the writer, character properties, writing attributes, or overall document appearance. Instead of examining eachelement separately, we believe that an integrated agent capable of processing known characters and novelties concurrently is a superior technique. The handwritten digit recognition problem can be seen asa subtask of the optical character recognition (OCR) problem. |
| 4. | Social Impact / Customer Satisfaction | There are many benefits associated with the handwriting recognition system. In addition to reading postal addresses and bank check amounts, it is also useful for reading forms. Furthermore, it's usedin fraud detection because it makes it easy to compare two texts and determine which one is a copy. As a result, this system fulfills customers' expectations, as it is a novel method for recognizing handwritten digits, ensuring high accuracy for the model and meeting all customer expectations. Users will save a lot of time and effort if the system provides various synonyms for the words recognized. Becausethe users in rural areas will be using their regional language, this proposed system should be able to detect those digits as well. As the system is being usedin socially crowded places such as banks to check amounts, it should be fast and reliable. As it is designed to solve real-world problems, it should be highly reliable and trustworthy in every way, and users throughout the world should be able to use it effectively. |
| 5. | Business Model (Revenue Model) | A revenue model means understanding how a startup can make money. Our major revenue sourcesconsist of ***sales, government funds, and public donations.*** The introduction of novel ideas increases revenue streams, such as introducing gesture or touch features, voice readout of recognized digits, Etc. |
| 6. | Scalability of the Solution | One of the approaches to make the handwritten digitrecognition system scalable is to make use of cloud-native methods. For example, one of the cloud solutions for making AI scalable is IBM Cloud. IBM Cloud Build helps run and manage AI models andoptimize decisions at scale across any cloud. The advantage of using the cloud to make solutions scalable is that we can deploy our AI application on |

| | | |
|---|---|---|
| | | the specific cloud environment that best supports our business needs. We can take advantage of built-in security capabilities and AI model monitoring. We can Automate AI lifecycles with ModelOps pipelines, deploy and run models through one-click integration and also prepare and build models visually and programmatically. Looking at these advantages, we can drive better business outcomes by optimizing our decisions and also making our solution scalable using<br>the cloud. |

### 3.4 Problem Solution fit



## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | MNIST Dataset | The Modified National Institute of Standards andTechnology dataset (MNIST) database of handwritten digits has a training set of 60,000 examples and a test set of 10,000 examples. |
| FR-2 | Data preprocessing | Improves the image by doing some operations tothe input image to prepare it for segmentation. |
| FR-3 | GUI | Enables the user to insert a handwritten image andreceive the digits in digital form. designed to facilitate virtualization. |
| FR-4 | Image Data | The ability of a computer to recognize human handwritten digits from various sources, such as images, documents, touch screens, etc., and classify them into ten recognized classes is knownas handwritten digit recognition (0-9).<br><br>This has received a great deal of research in thefield of deep learning. |
| FR-5 | Digit Classifier Model | Utilize the MNIST collection of handwritten digits to train a convolutional network to predict a digit from an image. Assemble the data for training and validation first. |
| FR - 6 | Evaluation | Ensure that the digit is correctly recognized by themodel and produces accurate output. |

| | | |
|---|---|---|
| FR - 7 | Website | The code, graphics, and other components of a website are made available online by web hosting. Every website is hosted by a server. Theamount of server space provided to a website depends on the hosting type. The four primary types of hosting are shared dedicated, VPS, andreseller. |

### 4.2 Non-Functional requirements

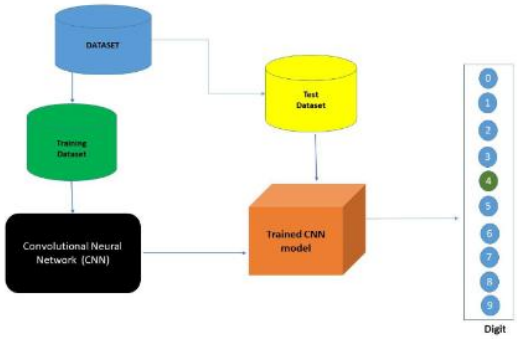
Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | To accurately recognize and comprehend handwritten digits mechanically. |
| NFR-2 | **Security** | In addition to classifying the digit, the algorithm also generates a full description of the instantiation parameters, which could disclose detailslike the writing style.<br>The generative models are capable of segmentation driven by recognition. |
| NFR-3 | **Reliability** | The neural network makes use of the samples to automatically determine rulesfor reading handwritten digits. By increasing the number of training instances, the network may also learn more about handwriting and hence improve its accuracy.<br>To recognize handwritten numbers, a variety of methods and algorithms canbe employed, including Deep Learning/CNN, SVM, Gaussian NaiveBayes, KNN, Decision Trees, RandomForests, etc. |
| NFR-4 | **Performance** | High, as deep learning models are created usingartificial neural networks that are trained on thetraining set of images. employing the CNN algorithm for quick prediction. |
| NFR-5 | **Availability** | Anyone can quickly access the system through a web application, making it very accessible fordesktop and mobile browsers. |
| NFR-6 | **Scalability** | Works with various other datasets with differentlanguages and writing styles. |

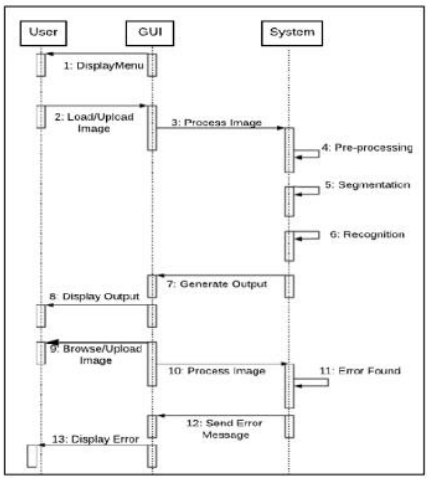
## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams

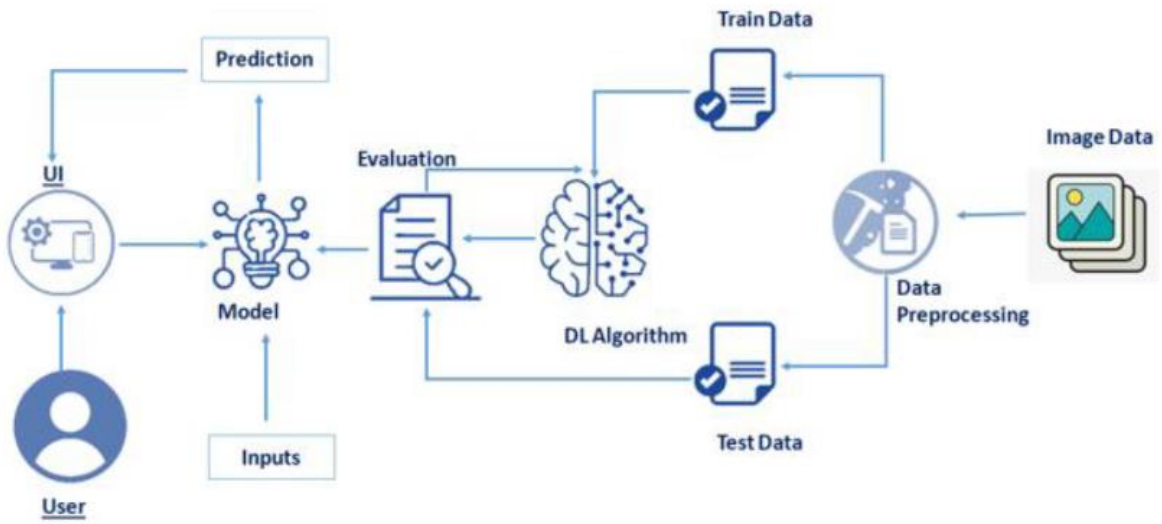
Example: (Simplified)




Example: DFD Level 0 (Industry Standard)




### 5.2 Solution & Technical Architecture



### 5.3 User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer (Mobile user/ web user) | Home | USN - 1 | As a user, I can view the guidelines given on how to use the website. | I can see the limitations of this programme and the awareness of how to use it. | Low | Sprint -1 |
| | | USN - 2 | As a user, I can view the video instructions provided to use the website. | I can learn how to use this application through a hands-on approach. | Low | Sprint -1 |
| | | USN - 3 | As a user, I can interact with the GUI to navigate through the website. | I can use the website in a user-friendly manner. | Low | Sprint -1 |
| | Recognize | USN - 4 | As a user, I can upload images using various upload options. | I can upload the images of the handwriting to be recognised from various sources | High | Sprint - 2 |
| | | USN - 5 | As a user, I can draw the character of the handwriting to be recognised in the drawing space available. | I can use the GUI to draw on the screen. | High | Sprint - 2 |
| | | USN - 6 | As a user, I can use the web application anywhere virtually. | The application is portable, so I can use it anywhere. | High | Sprint - 1 |
| Customer Care Executive | | USN - 7 | As a user, I can use the web application anywhere virtually. | The application is portable, so I can use it anywhere. | Medium | Sprint - 2 |
| Administrator | | USN - 8 | As a user, I can use the web application anywhere virtually. | The application is portable, so I can use it anywhere. | High | Sprint - 2 |

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

| Sprint | Functional Requirement (Epic) | User Story Number | User Story /Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Data Collection & Pre-processing | USN-3 | As a user, I can upload any kind of image with the pre-processing stepinvolved in it. | 10 | High | Raghunandan U R<br>Akash S |
| Sprint-2 | Upload input Image | USN-4 | As a user, I can input images of digital documents, handwritten documents, or images into the application. | 5 | Medium m | Sibi S Karthick M |

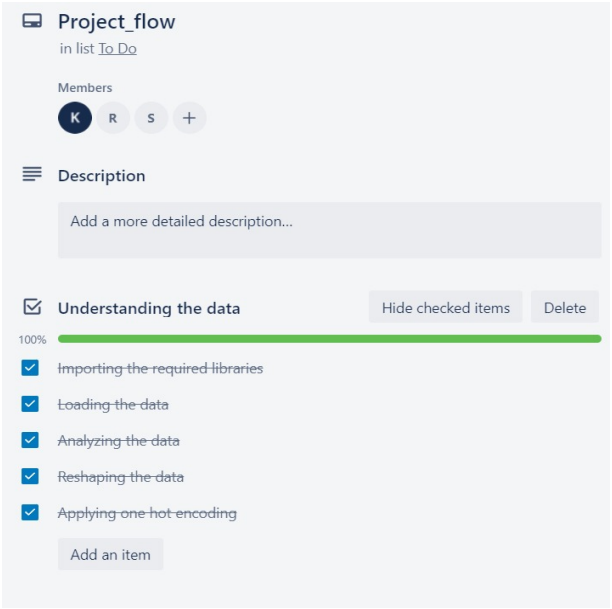
| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-2 | Building the ML model | USN-5 | As a user, I will get an application withan ML model which provides high accuracy of recognized handwritten digits. | 20 | High | Raghunandan U R<br>Akash S |
| Sprint-2 | | USN-6 | As a user, I can pass the handwritten digit image for recognizing the digit. | 5 | Medium m | Sibi S<br>Karthick M |
| Sprint-3 | Building the UI Application | USN-7 | As a user, I will upload the handwritten digit image to the application by clicking an upload button. | 3 | Medium m | Raghunandan U R<br>Akash S |
| Sprint-3 | | USN-8 | As a user, I can know the details ofthe fundamental usage of the application. | 3 | Low | Raghunandan U R<br>Akash S |
| Sprint-3 | | USN-9 | As a user, I can see the predicted/recognized digits in the application. | 4 | High | Sibi S |
| Sprint-4 | Train and deploy the model in IBM Cloud | USN-10 | As a user, I can access the web application and make use of the product from anywhere. | 10 | High | Akash S |
| Sprint-4 | Recognize the digit | USN-11 | As a user, I can get the recognrecognized as output from the images of digital documents or images. | 10 | Medium m | Akash |

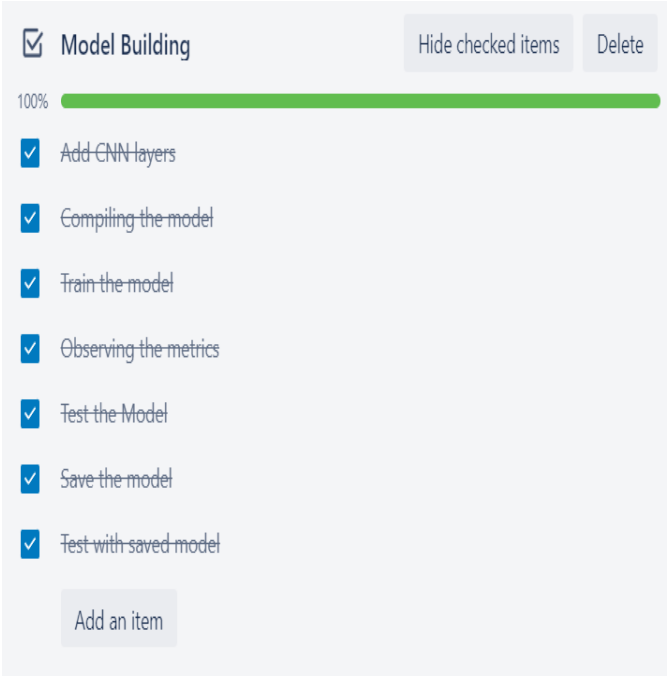
## 6.2 Sprint Delivery Schedule

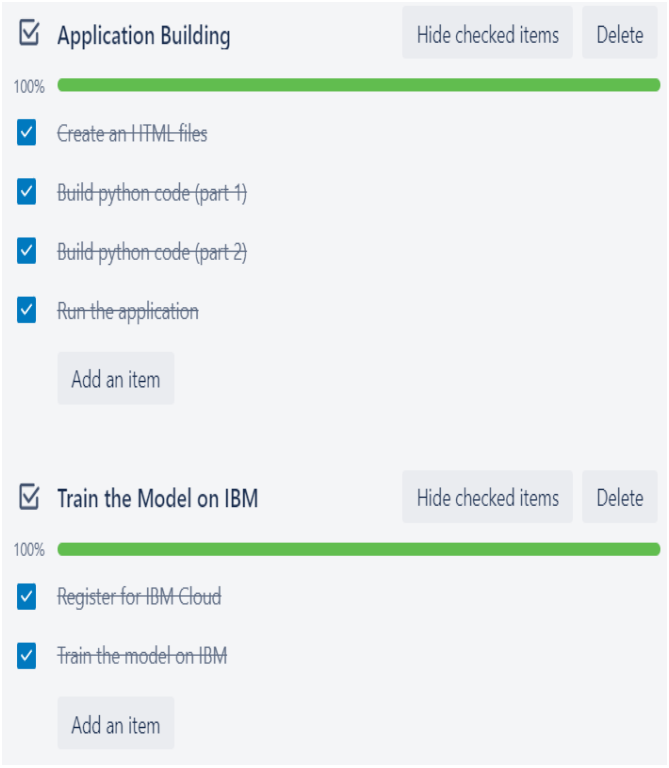
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint ReleaseDate (Actual) |
|---|---|---|---|---|---|---|
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 30 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 10 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

## 6.3 Reports from Tello

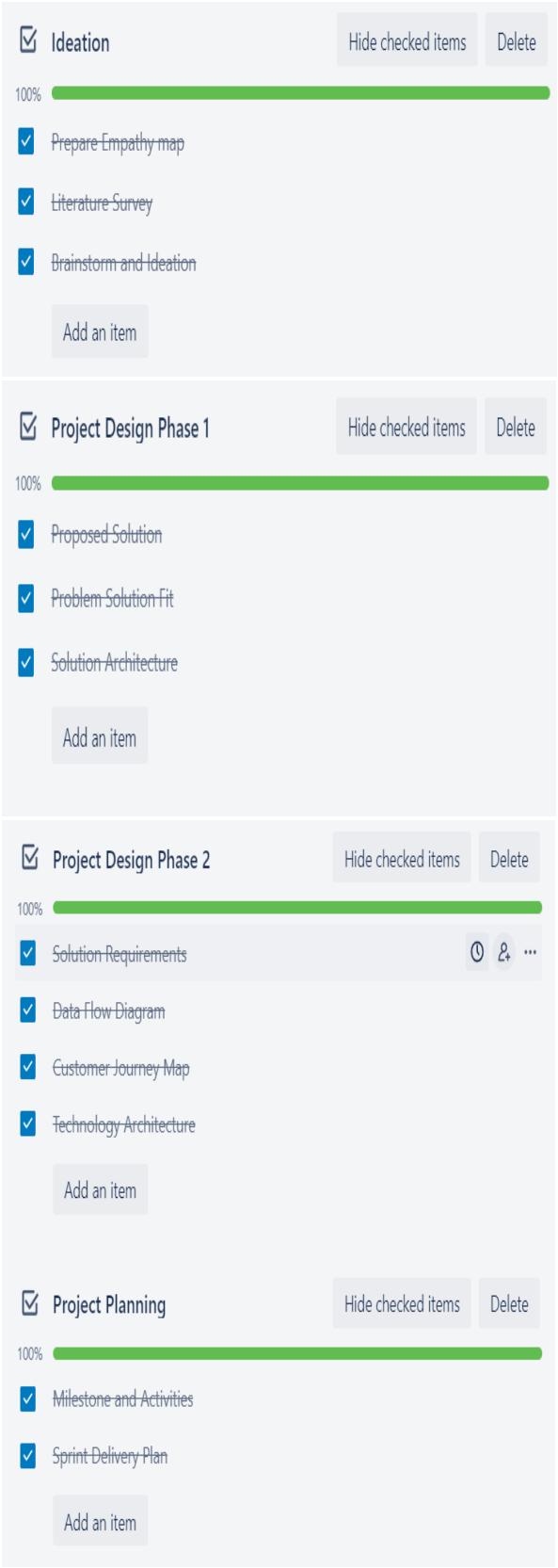
1. **Project Flow**




Model Building
Hide checked items
Delete
100%
Add CNN layers
Compiling the model
Train the model
Observing the metrics
Test the Model
Save the model
Test with saved model
Add an item


Application Building
Hide checked items
Delete
100%
Create an HTML files
Build python code (part 1)
Build python code (part 2)
Run the application
Add an item
Train the Model on IBM
Hide checked items
Delete
100%
Register for IBM Cloud
Train the model on IBM
Add an item

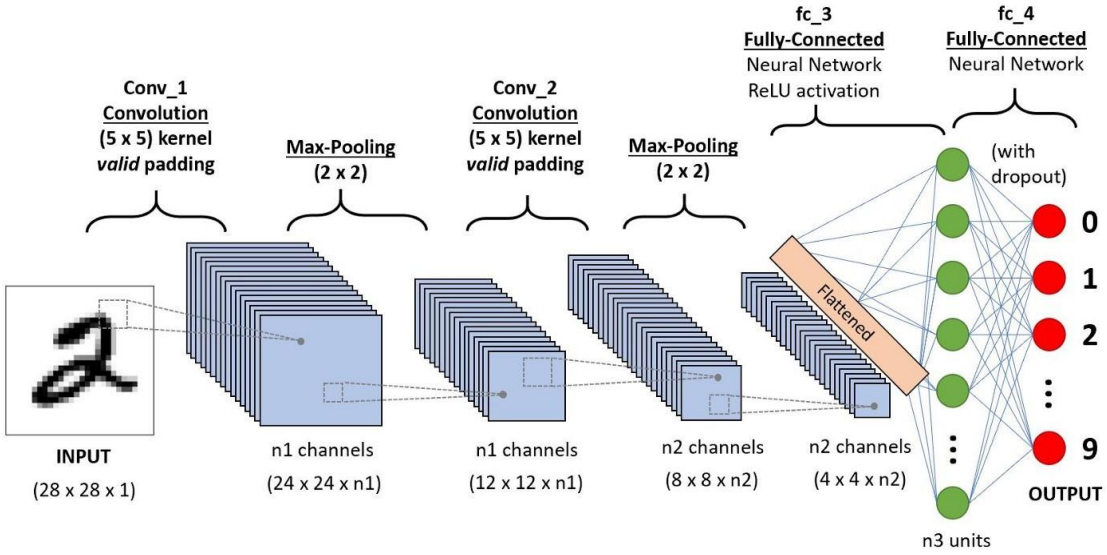
## 2. Documentation flow:



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

### 7.1 Feature 1

- Using CNN Model in our Project: CNN is b model known to be Convolutional Neural Network and in recent times it has gained a lot of popularity because of its usefulness. CNN uses multilayer perceptrons to do computational work
- CNN uses relatively little pre-processing compared to other image classification algorithms. This means the network learns through filters that in traditional algorithms were hand-engineered. So, for image processing tasks CNNs are the best-suited option.




### 7.2 Feature 2

- Using Flask application in our Project: Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has nodatabase abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies, and several common framework-related tools.

## 8. TESTING

### 8.1 Test Cases

| Expected Result |
| --- |
| Predict page should open |
| The home page should open |
| Predict page should open properly |
| Choose File should be uploaded |
| A preview of the image should be viewed on the predicted page |
| It should move from the predicted page to the result page. |
| it should validate the file uploaded |
| check whether the file is reshaped and stored as a variable in main.py |
| check whether the file loads the dataset and predicts the digit |
| check whether the result page is displayed. |
| check whether the result page displays the correct answer |
| check whether the result page contains the upload again button |

| Test Scenario |
| --- |
| Verify whether the user can use the recognize button |
| Verify whether the user can use the home button |
| Verify whether the predicted page is opening |
| Verify if we can upload the file to be recognized |
| Verify whether they can view the preview of the thethemeich is being uploaded |
| Verify whether the user is to bo use the recognize button |
| Verify whether it validates the file |
| Verify whether it accepts the file uploaded |
| Verify whether it loads the dataset and predicts the digit |
| Verify whether the result is displayed |
| Verify whether the result displayed is correct |

## 8.2 User Acceptance Testing

### 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 | 0 | 4 | 2 | 3 | 9 |
| Duplicate | 0 | 0 | 3 | 0 | 3 |
| External | 0 | 0 | 0 | 1 | 1 |
| Fixed | 0 | 4 | 5 | 4 | 13 |
| Not Reproduced | 0 | 0 | 0 | 0 | 0 |
| Skipped | 0 | 0 | 0 | 1 | 1 |
| Won't Fix | 0 | 0 | 0 | 1 | 1 |
| Totals | 0 | 8 | 10 | 10 | 26 |

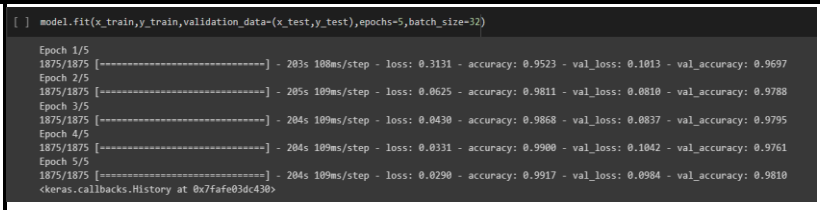
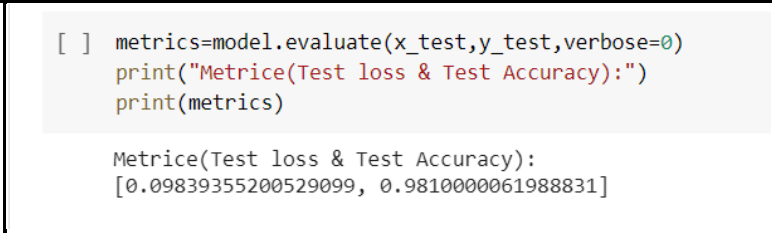
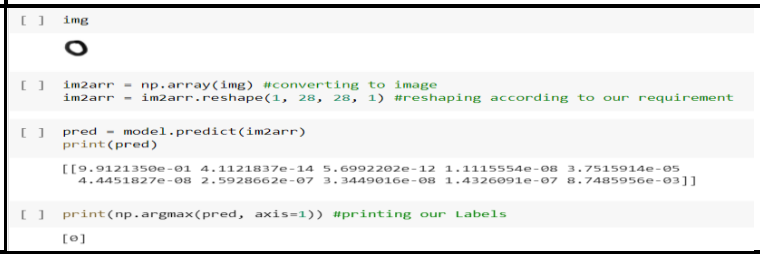
### Test case analysis

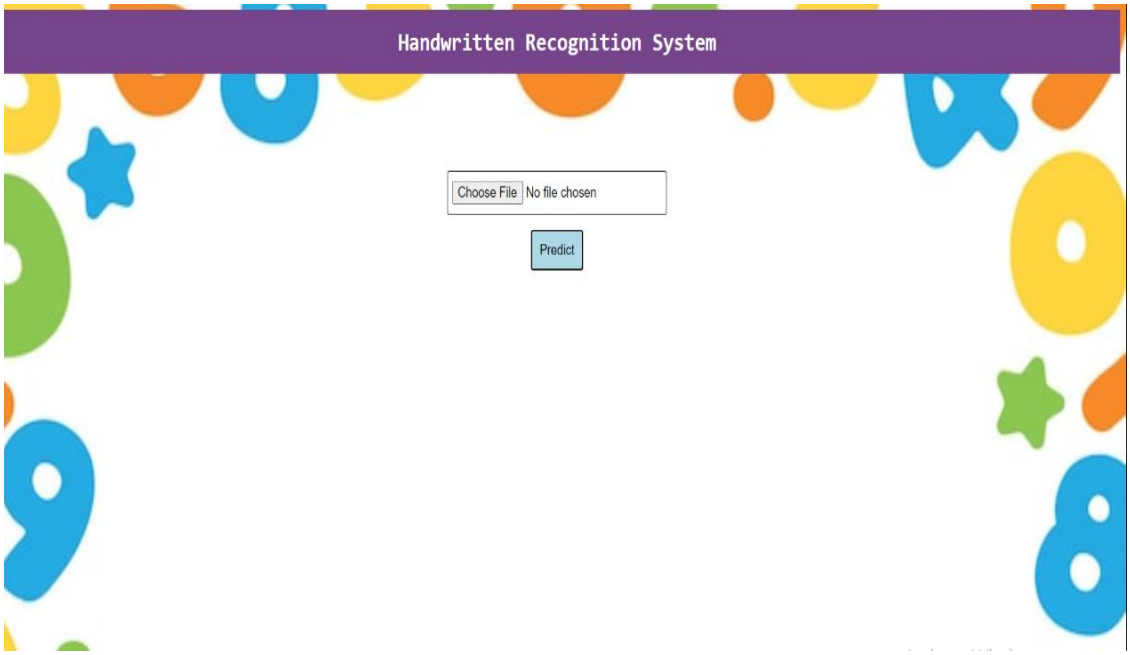
This report shows the number of test cases that have passed, failed, and untested

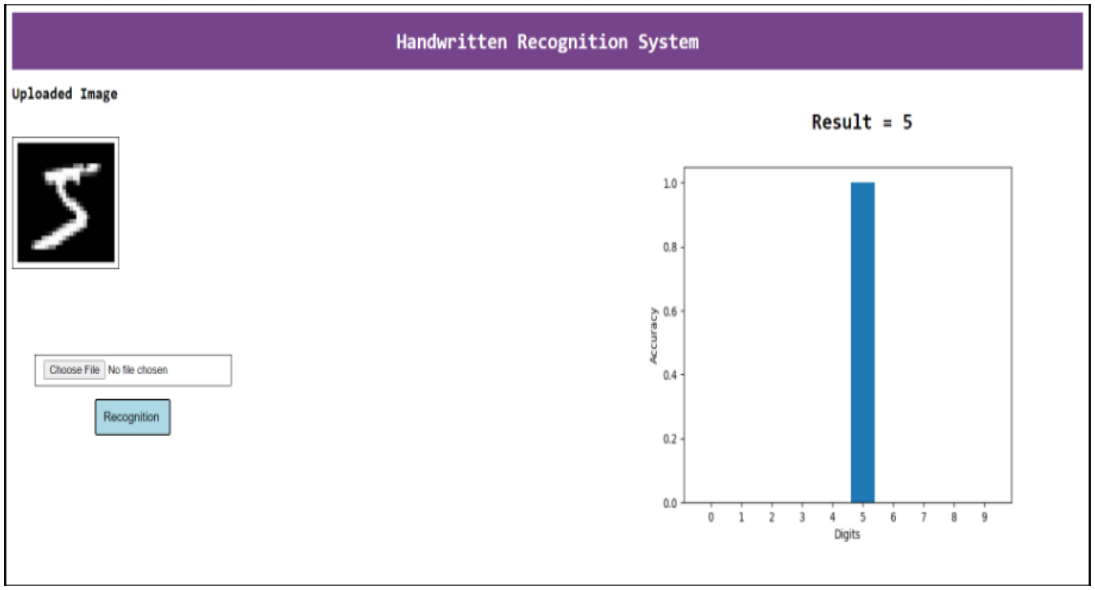
| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Print Engine | 7 | 0 | 0 | 7 |
| Client Application | 10 | 0 | 0 | 10 |
| Security | 1 | 0 | 0 | 1 |
| Outsource Shipping | 2 | 0 | 0 | 2 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

## 9. RESULTS

### 9.1 Performance Metrics

| S.No. | Parameter | Values | Screenshot |
|---|---|---|---|
| 1 | Accuracy | Training Accuracy 96.7% Validation Accuracy 97.12% |  |
| 2 | Confusion Matrix | Test loss 0.098 |  |
| 3 | Classification Report | 0 |  |






## 10. ADVANTAGES & DISADVANTAGES

Handwriting Recognition has many advantages that made it grow rapidly in the technology world now. There are many different kinds of technologies that enable others to take advantage of handwriting recognition. The way this work was when people write letters a different way and they let the computer know what the intended letter was and change it into a text document Certain cell phones have handwriting recognition system in them. The advantage of this is that it allowspeople to write on their cell phones using a stylus and then the phone software translates the written words to the phone in text.

The disadvantage of handwriting recognition technologies is that not everyone's handwriting is thesame, everyone writes differently. This starts the problem in handwriting recognition technology when it needs to translate a person's handwriting into the type and because of this problem many companies failed to perform well because many couldn't effectively use the program well enough.

## 11. CONCLUSION

Handwritten Digit Recognition using Deep learning methods has been implemented. CNN hasbeen trained and tested on the same data in RotoWire the comparison between the classifiers. Utilizing these deep learning techniques, a high amount of accuracy can be obtained. Compared to otherresearch methods, this method focuses on which classifier works better by improving the accuracy of classification models by more than 99%. Using Keras as backend and The Tensorflow as the software, a CNN model is able to up to 99%

## 12. FUTURE SCOPE

The proposed system takes 28x28 pixel-sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build a Handwritten Character Recognition System which recognizes human handwritten characters and predicts the output.

## 13. APPENDIX

### Source Code

**index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta HTTP-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home</title>
  <style>
    body{
      background-image: url('templates\assests\color.jpg')
      background-repeat: no-repeat;
      background-size: cover;
      box-sizing: border-box;
      font-family: monospace;
    }
    a {
      text-decoration: none;
      color: white;
    }
    li {
      float: left;
      margin: 10px;
      color: white;
    }
    #nav {
      background-color: #76448A;
      width: 100%;
      height: auto;
      display: flex;
      justify-content: center;
    }
    #head {
      display: flex;
      justify-content: center;
```

```
  align-items: center;
}
#nav1 {
  background-color: rgb(255, 255, 255, 0.5);
  margin-top: 100px;
}
p {
  font-size: 25px;
  text-indent: 60px;
  line-height: 2.0;
}
h1 {
  color: white;
}
ul {
  position: absolute;
  right: 5px;
}
input[type=submit]{
  background-color: lightblue;
  border-radius: 3px;
  font-size: 15px;
  padding: 10px;
}
input[type=submit]:hover{
  background-color: #808B96;
  border-radius: 3px;
  font-size: 15px;
  color: #fff;
  padding: 10px;
}
input[type=file]{
  border: 1px solid ;
  border-radius: 3px;
  display: inline-block;
  font-size: 15px;
  padding: 10px 5px 10px 5px;
```

```
      cursor: pointer;
    }
  </style>
</head>
<body">
  <div id="nav">
    <h1>Handwritten Recognition System</h1>
  </div>
  <div id="nav1">
    <form action="/predict" method="POST" enctype="multipart/form-data" style="text-align:center;">
      <input type="file" name="file"  class="custom-file"/>
      <br>
      <br>
      <input type="submit" value="Predict" name="submit">
    </form>
  </div>
</body>
</html>
```

**main.html**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    body{
      background-image: url("/assests/color.jpg");
      background-repeat: no-repeat;
      background-size: cover;
      box-sizing: border-box;
      font-family: monospace;
    }
    #orgimg {
```

```
  position: absolute;
  height: 150px;
  width: 150px;
  top: 150px;
  border: 1px solid black;
}
#graph {
  position: absolute;
  height: 500px;
  width: 600px;
  top: 125px;
  right: 50px;
}
a {
  text-decoration: none;
  color: white;
}
li {
  float: left;
  margin: 10px;
  color: white;
}
#nav {
  background-color: #76448A;
  width: 100%;
  height: auto;
  display: flex;
  justify-content: center;
}
#head {
  display: flex;
  justify-content: center;
  align-items: center;
}
#nav1 {
  background-color: rgb(255, 255, 255, 0.5);
  position: absolute;
```

```
    top: 400px;
    left: 40px;
   }
   p {
    font-size: 25px;
    text-indent: 60px;
    line-height: 2.0;
   }
   #nav h1 {
    color: white;
   }
   ul {
    position: absolute;
    right: 5px;
   }
   input[type=submit]{
    background-color: lightblue;
    border-radius: 3px;
    font-size: 15px;
    padding: 10px;
   }
   input[type=submit]:hover{
    background-color: #808B96;
    font-size: 15px;
    border-radius: 3px;
    color: #fff;
    padding: 10px;
   }
   input[type=file]{
    border: 1px solid ;
    display: inline-block;
    padding: 6px 12px;
    cursor: pointer;
   }
  </style>
</head>
<body>
```

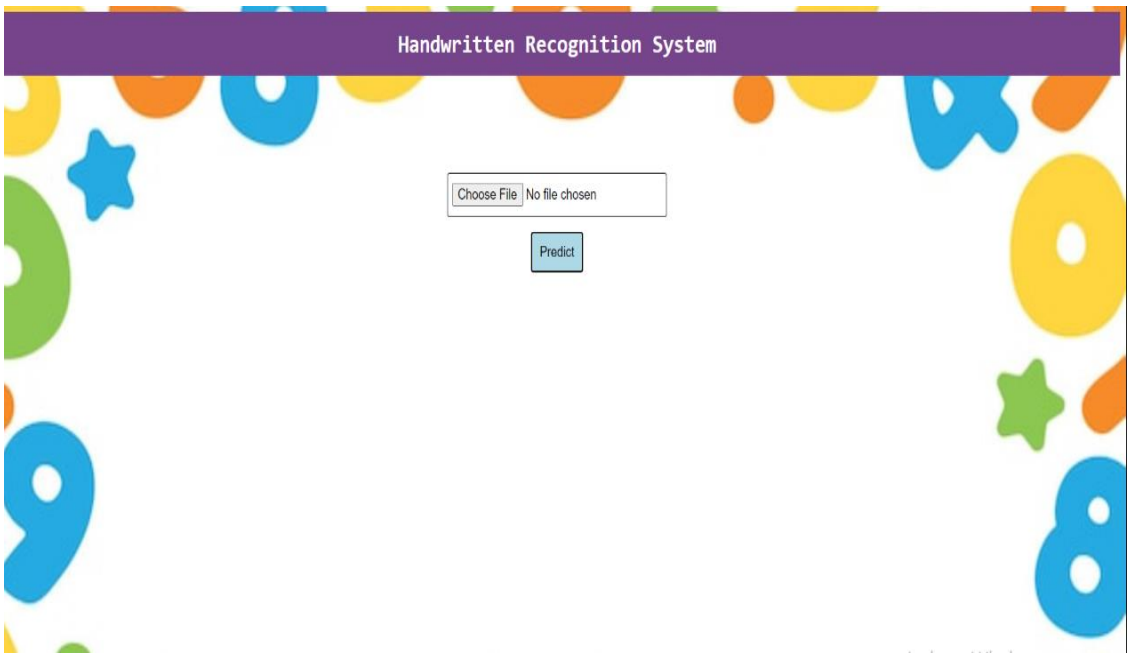
```
  <div id="nav">
    <h1>Handwritten Recognition System</h1>
  </div>
  <h2>Uploaded Image</h2>
  <img class="img1" src="./static/result.png" id="orgimg" />
  <div id="nav1">
    <form action="/predict" method="POST" enctype="multipart/form-data" style="text-align:center;">
      <input type="file" name="file" />
      <br>
      <br>
      <input type="submit" value="Recognition "/>
    </form>
  </div>
  <img class="img1" src="./static/graph.png" id="graph" />
  <h1 style="position:absolute;top: 100px;right: 250px;">Result = {{showcase}}</h1>
</body>
</html>
```

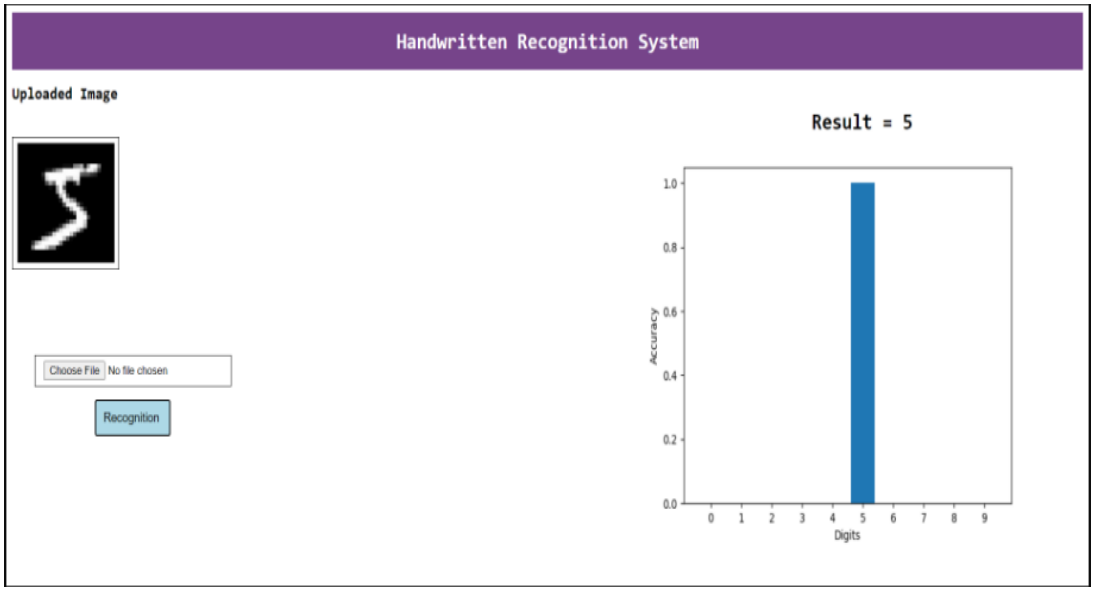
**App.py**

```
import os
import numpy as np
import pandas as pd
from PIL import Image
import matplotlib.pyplot as plt
import tensorflow as tf
from keras.models import load_model
from flask import Flask, render_template, request
from flask import Flask, render_template
model = load_model("models\mnist.h5")
app = Flask(__name__)
def predictRes():
    global model
    img = Image.open("./static/result.png").convert("L")
    img = img.resize((28, 28))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1, 28, 28, 1)
    y_pred = model.predict(im2arr)
    re = list(y_pred[0]).index(max(y_pred[0]))
```

```
    plt.bar(list(range(10)), y_pred[0], align="center")
    plt.xticks(list(range(10)), list(range(10)))
    plt.xlabel("Digits")
    plt.ylabel("Accuracy")
    plt.savefig('./static/graph.png')
    plt.clf()
    plt.close()
    return re
@app.route('/')
def home():
    return render_template('./index.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['file']
        f.save("./static/result.png")
        res = predictRes()
        return render_template('./main.html', showcase=str(res))
if __name__ == '__main__':
    app.run()
```

## Output:






**A model trained on IBM:**

```
#Importing the required libraries

pip install Keras
import NumPy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer
from tensorflow.keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt  #used for data visualization
from tensorflow.keras.models import load_model
from PIL import Image

#Loading the Data
(x_train, y_train), (x_test, y_test)=mnist.load_data ()
```

```
(x_train, y_train), (x_test, y_test)=mnist.load_data ()

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
```

```
#Analyzing the Data
print (x_train.shape)  #shape is used for give the dimens ion values #60000-rows 28x28-pixels
```

print (x_test.shape)
x_train[0]

```
[ ] #Analyzing the Data
```

```
[ ] print (x_train.shape)  #shape is used for give the dimens ion values #60000-rows 28x28-pixels
    print (x_test.shape)
```

```
(60000, 28, 28)
(10000, 28, 28)
```

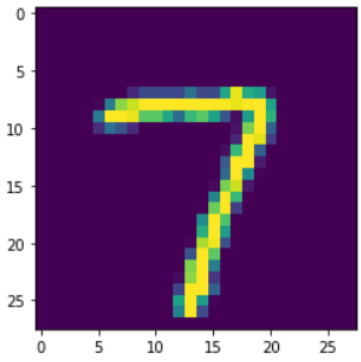
```
x_train[0]
```

```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
```

plt.imshow(x_train[5100])
np.argmax(y_train[5100])

```
plt.imshow(x_train[5100])
```

```
<matplotlib.image.AxesImage at 0x7fafe06716a0>
```




```
[ ] np.argmax(y_train[5100])
```

```
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')

#Applying one hot encoding
classes = 10
y_train = np_utils.to_categorical (y_train, classes)
y_test = np_utils.to_categorical (y_test, classes)


#Adding  CNN Buliding
model=Sequential()
model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(Conv2D(64,(3,3), activation = 'relu'))
model.add(Flatten())

#Compiling The Model
model.compile(loss='categorical_crossentropy',optimizer="Adam",metrics=['accuracy'])

#Training the Model
model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=5,batch_size=32)
```

```
model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=5,batch_size=32)

Epoch 1/5
1875/1875 [==============================] - 203s 108ms/step - loss: 0.3131 - accuracy: 0.9523 - val_loss: 0.1013 - val_accuracy: 0.9697
Epoch 2/5
1875/1875 [==============================] - 205s 109ms/step - loss: 0.0625 - accuracy: 0.9811 - val_loss: 0.0810 - val_accuracy: 0.9788
Epoch 3/5
1875/1875 [==============================] - 204s 109ms/step - loss: 0.0430 - accuracy: 0.9868 - val_loss: 0.0837 - val_accuracy: 0.9795
Epoch 4/5
1875/1875 [==============================] - 204s 109ms/step - loss: 0.0331 - accuracy: 0.9900 - val_loss: 0.1042 - val_accuracy: 0.9761
Epoch 5/5
1875/1875 [==============================] - 204s 109ms/step - loss: 0.0290 - accuracy: 0.9917 - val_loss: 0.0984 - val_accuracy: 0.9810
<keras.callbacks.History at 0x7fafe03dc430>
```

```
#Observing The Metrics
metrics=model.evaluate(x_test,y_test,verbose=0)
print("Metrice(Test loss & Test Accuracy):")
print(metrics)
```

```
metrics=model.evaluate(x_test,y_test,verbose=0)
print("Metrice(Test loss & Test Accuracy):")
print(metrics)

Metrice(Test loss & Test Accuracy):
[0.09839355200529099, 0.9810000061988831]
```

```
#Test the Model
prediction=model.predict(x_test[:4])
print(prediction)
```

```
prediction=model.predict(x_test[:4])
print(prediction)
```

```
[[2.66373620e-13 5.15261426e-21 1.50421166e-13 1.66002376e-08
  2.93580552e-20 7.28840418e-18 6.59416025e-23 1.00000000e+00
  1.48631650e-12 5.62014844e-12]
 [1.01646545e-10 1.14599290e-17 1.00000000e+00 3.88194745e-18
  1.34460339e-18 7.51504113e-23 2.78395113e-10 8.21130562e-20
  2.91733380e-13 7.73047336e-22]
 [1.32042760e-12 1.00000000e+00 1.19916399e-09 1.56342046e-16
  7.32658212e-10 3.01171761e-11 2.41586612e-10 6.36250774e-10
  3.01473499e-11 2.16507127e-15]
 [1.00000000e+00 8.08110351e-20 1.30329358e-11 2.95739436e-15
  1.90499827e-16 1.85495303e-14 5.43617018e-12 9.05207373e-14
  3.09776564e-13 2.79464452e-10]]
```

```
print(np.argmax(prediction,axis=1))
print(y_test[:4])
```

```
print(np.argmax(prediction,axis=1))
print(y_test[:4])
```

```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

```
#Saving the model
model.save("Model/digitrec.h5")
cd models
!tar -zcvf hdr_deployment.tgz digitrec.h5
ls -1
```

```
[ ]  #Saving the model
```

```
[ ]  model.save("Model/digitrec.h5")
```

```
     cd models
```

```
     [Errno 2] No such file or directory: 'models'
     /home/wsuser/work
```

```
[ ]  !tar -zcvf hdr_deployment.tgz digitrec.h5
```

```
     tar: digitrec.h5: Cannot stat: No such file or directory
     tar: Exiting with failure status due to previous errors
```

```
[ ]  ls -1
```

```
     hdr_deployment.tgz
     Model/
```

```
!pip install watson-machine-learning-client --upgrade
from ibm_watson_machine_learning import APIClient
```

```
[ ]  !pip install watson-machine-learning-client --upgrade

     Collecting watson-machine-learning-client
       Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
          |████████████████████████████████| 538 kB 14.7 MB/s eta 0:00:01
     Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
     Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
     Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
     Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
     Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
     Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
     Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
     Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
     Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
     Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learn
     Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learni
     Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning
     Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21
     Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,
     Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-ma
     Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-
     Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-le
     Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client
     Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
     Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
     Installing collected packages: watson-machine-learning-client
     Successfully installed watson-machine-learning-client-1.0.391
```

```
from ibm_watson_machine_learning import APIClient
client.spaces.get_details()
def guid_from_space_name(client,deploy):
  space = client.spaces.get_details()
  return (next(item for item in space['resources'] if item['entity']['name']==deploy)['metadata']['id'])
 space_uid = guid_from_space_name(client,'Classification')
print("Space UID = " + space_uid)
client.set.default_space(space_uid)
```

```
client.software_specifications.list(limit=100)
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
```

```
software_space_uid = client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
model_details = client.repository.store_model(model='hdr_deployment.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"Digit Recognition System",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
      })   model_details
```

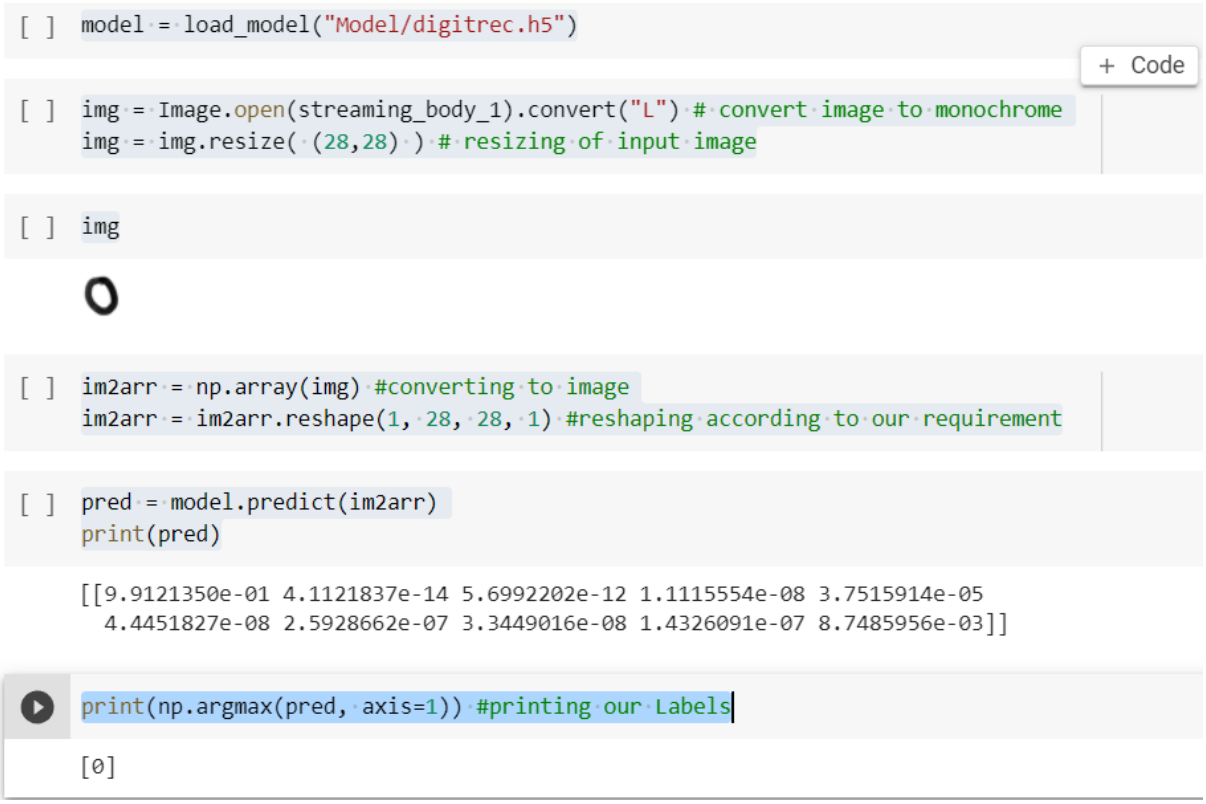
```
model_details
```

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_2.7'},
 'metadata': {'created_at': '2022-11-13T12:57:18.607Z',
  'id': 'ee04c4b7-ea90-4d1b-aa13-3259091a19c9',
  'modified_at': '2022-11-13T12:57:22.476Z',
  'name': 'Digit Recognition System',
  'owner': 'IBMid-663002IV3Z',
  'resource_key': 'f2e40b5f-7218-465e-a4c8-ed7a137f9781',
  'space_id': 'cca72fe8-1ea2-4559-a71c-c401ad862870'},
 'system': {'warnings': []}}
```

```
model_id = client.repository.get_model_id(model_details)
model_id
client.repository.download(model_id,'DigitRecog_IBM_model.tar.gz')
ls
```

```
#Test with Saved Model
from tensorflow.keras.models import load_model
from keras.preprocessing import image
from PIL import Image
import numpy as np

model = load_model("Model/digitrec.h5")
img = Image.open(streaming_body_1).convert("L") # convert image to monochrome
img = img.resize( (28,28) ) # resizing of input image
img
im2arr = np.array(img) #converting to image
im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our requirement
pred = model.predict(im2arr)
print(pred)
print(np.argmax(pred, axis=1)) #printing our Labels
```




**GitHub & Project Demo Link**

https://github.com/IBM-EPBL/IBM-Project-10932-1659245794

https://github.com/IBM-EPBL/IBM-Project-10932-1659245794/tree/master/Final%20Deliverables/Team%20Lead%20-%20Akash%20S