A NOVEL METHOD FOR HANDWRITTEN DIGITRECOGNITION SYSTEM

Team id: PNT2022TMID42530
Team Leader: Shamsundar R
Team Member 1: Karthikeyan.K
Team Member 2: Pulla Rakesh
Team Member 3: Ezhil raj

7. CODING &SOLUTIONING (Explain the features addedin the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11.CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub link

1. INTRODUCTION

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, 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 learning tools have been developed like scikit-learn, scipy-image etc. and pybrains, Keras, Theano, 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, Convolutional Neural Networks with Back Propagation for Image Processing, Deep Mind by Google for creating Art by learning from existing artist styles. 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 oddnumbered years. Both of these conferences are endorsed by the IEEE. Active areas of research include: Online Recognition, OfflineRecognition, Signature Verification, PostalAddress Interpretation,Bank-Check Processing, Writer Recognition. Classification of images and patterns has been one of the major implementation of Machine Learning and Artificial Intelligence. People are continuously trying to make computers intelligent so that they can do almost all the work done by humans Handwriting recognition system is the most basic and an important step towards this huge and interesting area of Computer Vision. 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.

1.1 Project Overview

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 multilayer CNN using Keras with Theano 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 neworks can all most mimic the human brain and are a key ingredient in image processing field. The advance of handwriting processing results from a combination of various elements, for example: improvements in the recognition rates, the use of complex systems to integrate various kinds of information, and new technologies such as high quality high speed scanners and cheaper and more powerful CPUs. Some handwriting recognition system allows us to input our handwriting into the system 1.2 PURPOSE Handwritten digits can be done either by controlling a mouse or using a thirdparty drawing tablet. The input can be converted into typed text or can be left as an "ink object" in our own handwriting. We can also enter the text we would like the system to recognize into any Microsoft Office program file by typing. We can do this by typing 1s and 0s. This works as a Boolean variable. Handwriting recognition [4] is not a new technology, but it has not gained public attention until recently. The ultimate goal of designing a handwriting recognition system with an accuracy rate of 100% is quite illusionary, because even human beings are not able to recognize every handwritten text without any doubt. For example, most people can not even read their own notes.

2. LITERATURE SURVEY

• . **Title** : A Novel Method For Handwritten Digit Recognition System
Base paper Link : htt HYPERLINK "http://troindia.in/journal%20%20/ijcesr/vol6iss6part2/32-%2036.pdf"p HYPERLINK "http://troindia.in/journal%20%20/ijcesr/vol6iss6part2/32-%2036.pdf"://troindia.in/journal H Y P E R L I N K " h t t p : / / t r o i n d i a . i n / j o u r n a l % 2 0 % 2 0 / i j c e s r / v o l 6 i s s 6 p a r t 2 / 3 2 - % 2 0 3 6 . p d f " HYPERLINK "http://troindia.in/journal%20%20/ijcesr/vol6iss6part2/32-%2036.pdf"/ijcesr/vol6iss6part2/32- HYPERLINK "http://troindia.in/journal%20%20/ijcesr/vol6iss6part2/32-%2036.pdf" HYPERLINK "http://troindia.in/journal%20%20/ijcesr/vol6iss6part2/32-%2036.pdf"36.pdf

Authors : Rohini.M , Dr.Surendran
Abstract : 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 multilayer CNN using Keras with Theano and Tensorflow. MNIST is a dataset which is widely used for handwrittendigit recognition. The dataset consist of 60,000 training images and 10,000 test images.The artificial neural neworks 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. •
.
Title : Handwritten Digit Recognition Using Machine Learning Algorithms
Base paper Link : https://www.researchgat e.net /publication/326408524_ Handwritten_Digit

Authors : S. M. Shamim, Md Badrul Alam Angona Miah, Angona Sarker, Masud Rana, Abdulla AlJobair
Abstract : Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off -line handwritten digit recognition based on different machine

learning technique. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Several machines learning algorithm namely, Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48 and Random Tree has been used for the recognition of digits using WEKA. • .

Title : A Novel Handwritten Digit Classification System Based on Covolutional Neural Network Approach
Base paper Link : https://www.google.com /url?sa=t HYPERLINK "http://www.google.com/url?sa=t&source=web&r"& HYPERLINK "http://www.google.com/url?sa=t&source=web&r"source=web HYPERLINK "http://www.google.com/url?sa=t&source=web&r"& HYPERLINK "http://www.google.com/url?sa=t&source=web&r"r ct=j& HYPERLINK "http://ijcsit.com/docs/Volume%25202/vol2issue4/ijcsit201102046"url=http://ijcsit.com /docs/Volume%25202/vo l2issue4/ijcsit201102046 3.pdf&ved=2ahUKEwjvuY CuvIr6AhWPTGwGHa9CQQFnoECBEQAQ&usg=AOvVaw0VhCTv7Yxb_dsN MzJnfRgt
Authors : Ali Abdullah Yahya
Abstract : 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. To address the shortcomings of these algorithms, our paper presents the following contributions: Firstly, after taking the domain knowledge into consideration, the size of the effective receptive field (ERF) is calculated. 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 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. Fourthly, to simulate the realworld natural influences that can affect image quality, we propose to add an additive white Gaussian noise with $\sigma = 0.5$ to the MNIST dataset. As a result, our CNN algorithm achieves state-oftheart results in handwritten digit recognition, with a recognition accuracy of 99.98%, and 99.40% with 50% noise. • .

Title : MultiLanguage Handwritten Digits Recognition based on Novel Structural Features
Base paper Link : https://www.google.com /url?sa=t HYPERLINK "http://www.google.com/url?sa=t&source=web&r"& HYPERLINK "http://www.google.com/url?sa=t&source=web&r"source=web HYPERLINK "http://www.google.com/url?sa=t&source=web&r"& HYPERLINK "http://www.google.com/url?sa=t&source=web&r"r ct=j&url=https://library.i maging.org/admin/apis/p ublic/api/ist/website/do wnloadArticle/jist/63/2/a

rt00004&ved=2ahUKEwj 02pvSvYr6AhVVFbcAHWi rCxs4FBAWegQICxAB&us g=AOvVaw3QynwJXDON 6433Ky_5LnTI

Authors : Ali Abdullah Yahya
Abstract : Automated handwritten script recognition is an important task for several applications. In this article, a multi-language handwritten numeral recognition system is proposed using novel structural features. A total of 65 local structural features are extracted and several classifiers are used for testing numeral recognition. Random Forest was found to achieve the best results with an average recognition of 96.73%. The proposed method is tested on six different popular languages, including Arabic Western, Arabic Eastern, Persian, Urdu, Devanagari, and Bangla. In recent studies, single language digits or multiple languages with digits that resemble each other are targeted. In this study, the digits in the languages chosen do not resemble each other. Yet using the novel feature extraction method a high recognition accuracy rate is achieved. Experiments are performed on well-known available datasets of each language. A dataset for Urdu language is also developed in this study and introduced as PMU-UD. Results indicate that the proposed method gives high recognition accuracy as compared to other methods. Low error rates and low confusion rates were also observed using the novel method proposed in this study. c 2019 Society for Imaging Science and Technology

2.1 Existing Problem

Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years. In our study there are many fields concern with numbers, for example, checks in banks or recognizing numbers in car plates, the subject of digit recognition appears. A system for recognizing isolated digits may be as an approach for dealing with such application. In other words, to let the computer understand the Arabic numbers that is written manually by users and views them according to the computer process.

Scientists and engineers with interests in image processing and pattern recognition have developed various approaches to deal with handwriting number recognition problems such as, minimum distance, decision tree and statistics.

2.2 References

Year: 2021

Authors: Ayush Kumar Agrawal and Vineet Kumar Awasthi An artificial neural network has one hidden layer between the input and output layers, whereas a deep neural network has numerous hidden layers with input and output layers. Deep neural networks use several hidden layers to increase model performance and achieve higher accuracy compared to accuracy of machine learning models. Most researchers do their research in the area of pattern recognition. In the field of pattern recognition, there are many patterns that can be used, including handwritten numbers, characters, pictures, faces, sounds, and speech. This study focuses on the classification and recognition of handwritten digits.1000 were utilized as test samples and 1000 were training samples.10000 picture samples make up the USPS dataset, of which 7291 serve as training samples and 2007 serve as testing samples. We've used the proposed deep neural network technique in this paper to classify and identify data from the ARDIS and USPS datasets. The suggested model consists of six layers with softmax and relu activation functions. After model implementation, accuracy for ARDIS samples reached 98.70% testing and 99.76% training, which is greater than accuracy from prior research. Additionally, using the USPS samples dataset, 98.22% training accuracy and 93.01% testing accuracy were attained. When compared to earlier methodologies, the data show that deep neural networks perform incredibly well.

2.3 Problem Statement Definition

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive yourproduct or service.

Reference: https://miro.com/templates/customer-problem-statement/

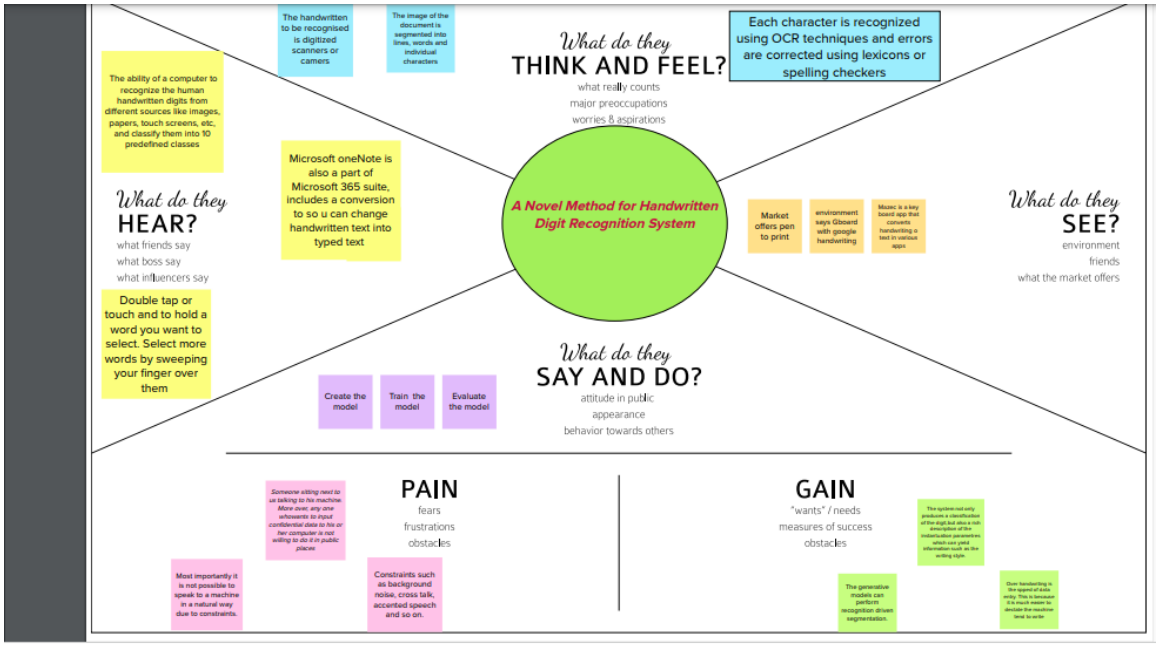
A Novel Method for Handwritten Digit Recognition System:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel

3.IDEATION AND PROPOSED SOLUTION

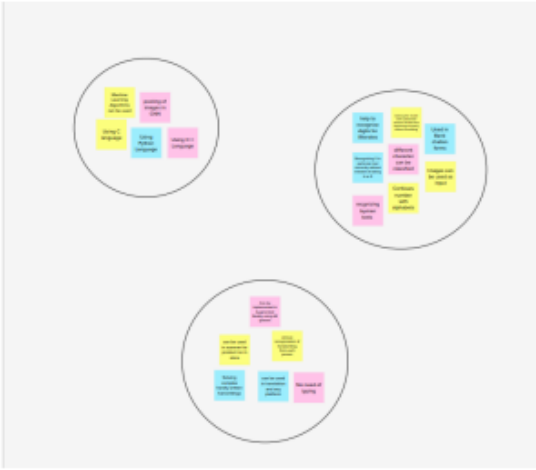
3.1 Empathy Map canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation and Brainstorming

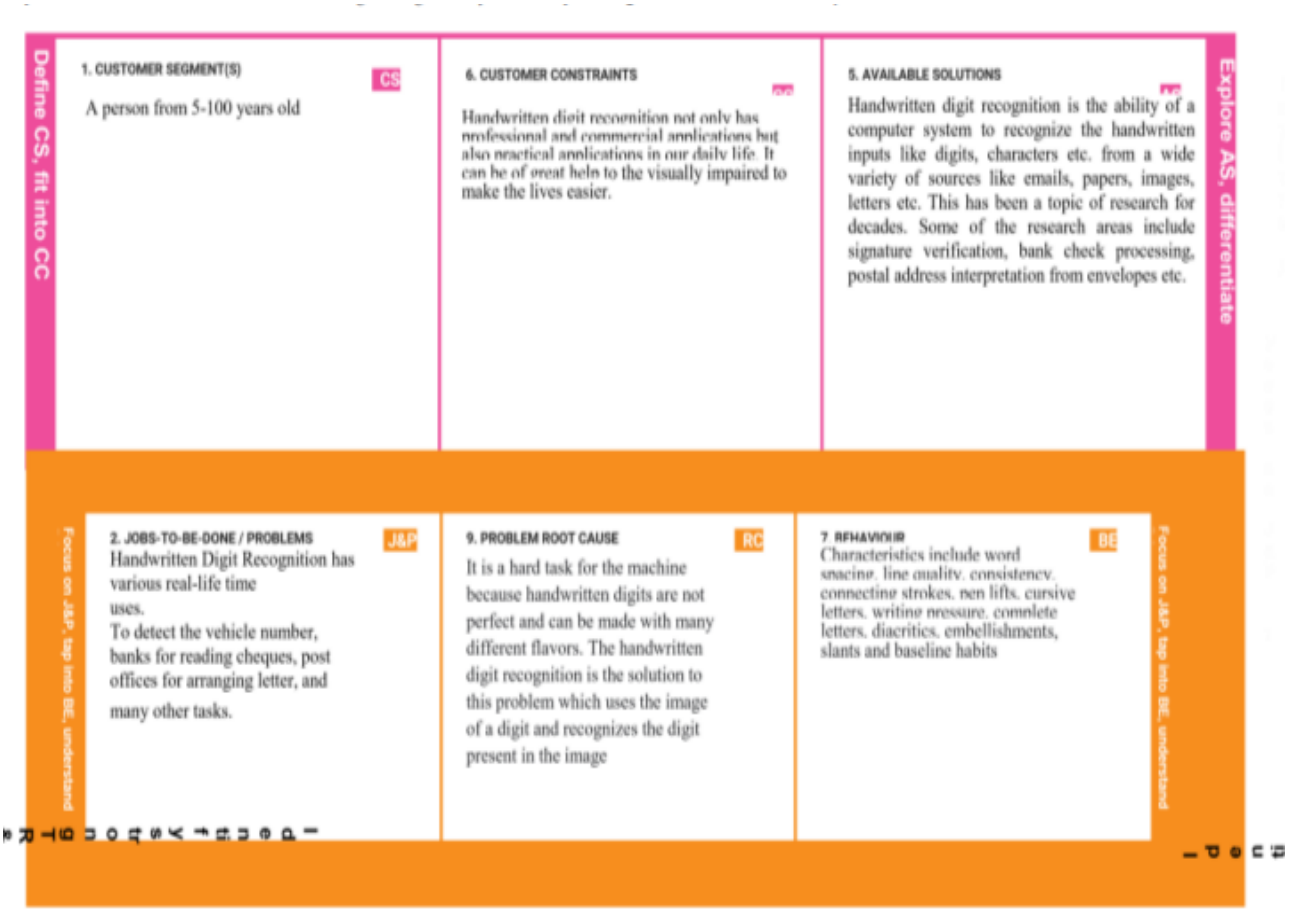
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.



3.3 Proposed solution

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Statement**: The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. **Description**: It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes
2.	Idea / Solution description	1. It is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, and touch defenses. 2. It allows user to translate all those signature and notes into electronic words in a text document format and this data only requires far less physical copies.
3.	Novelty / Uniqueness	Accurately recognize the digits rather than recognizing all the characters like OCR.
4.	Social Impact / Customer Satisfaction	1. Artificial Intelligence developed the app called handwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation
5.	Business Model (Revenue Model)	1. This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective traffic management.

3.4 Proposed solution Fit



4.REQUIREMENT ANALYSIS

4.1 Functional Requirements

i) Image Data:

Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorise them into ten established classifications (0-9). In the realm of deep learning, this has been the subject of countless studies.

ii) Website:

Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.

iii) Digit Classifier Model:

To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training and validation data first.

iv) Cloud:

The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual

platform that enables unlimited storage and access to your data over the internet.

v)Modified National Institute of Standards and Technology dataset:

The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

4.2 Non Functional Requirements:

i**) Usability:**

One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.

i**i) Security:**

The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. The generative models are capable of segmentation driven by recognition. The procedure uses a relatively.

iii) Reliability:

The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances. Numerous techniques

and algorithms, such as Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers.

iv) Accuracy:

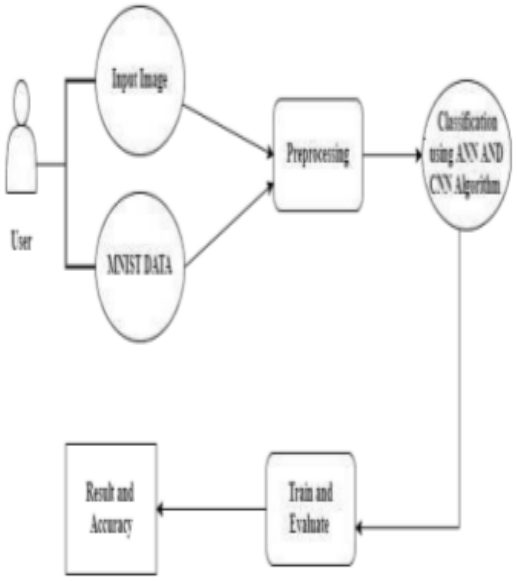
With typed text in high -quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.

5. PROJECT DESIGN

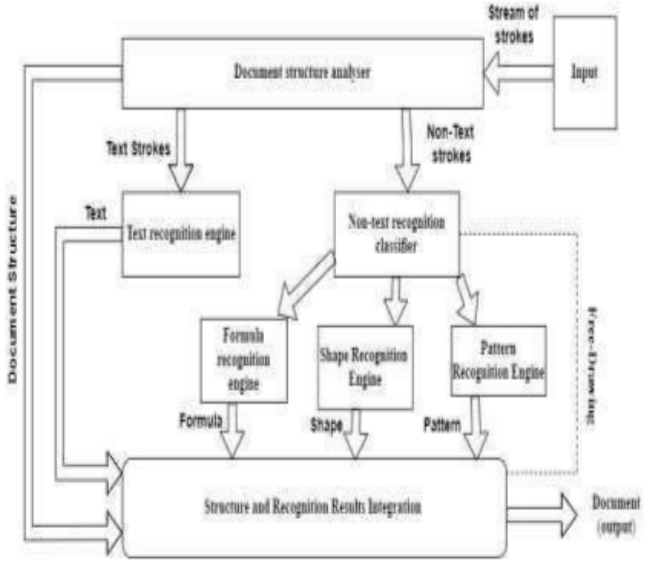
5.1 Data Flow Diagrams

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.

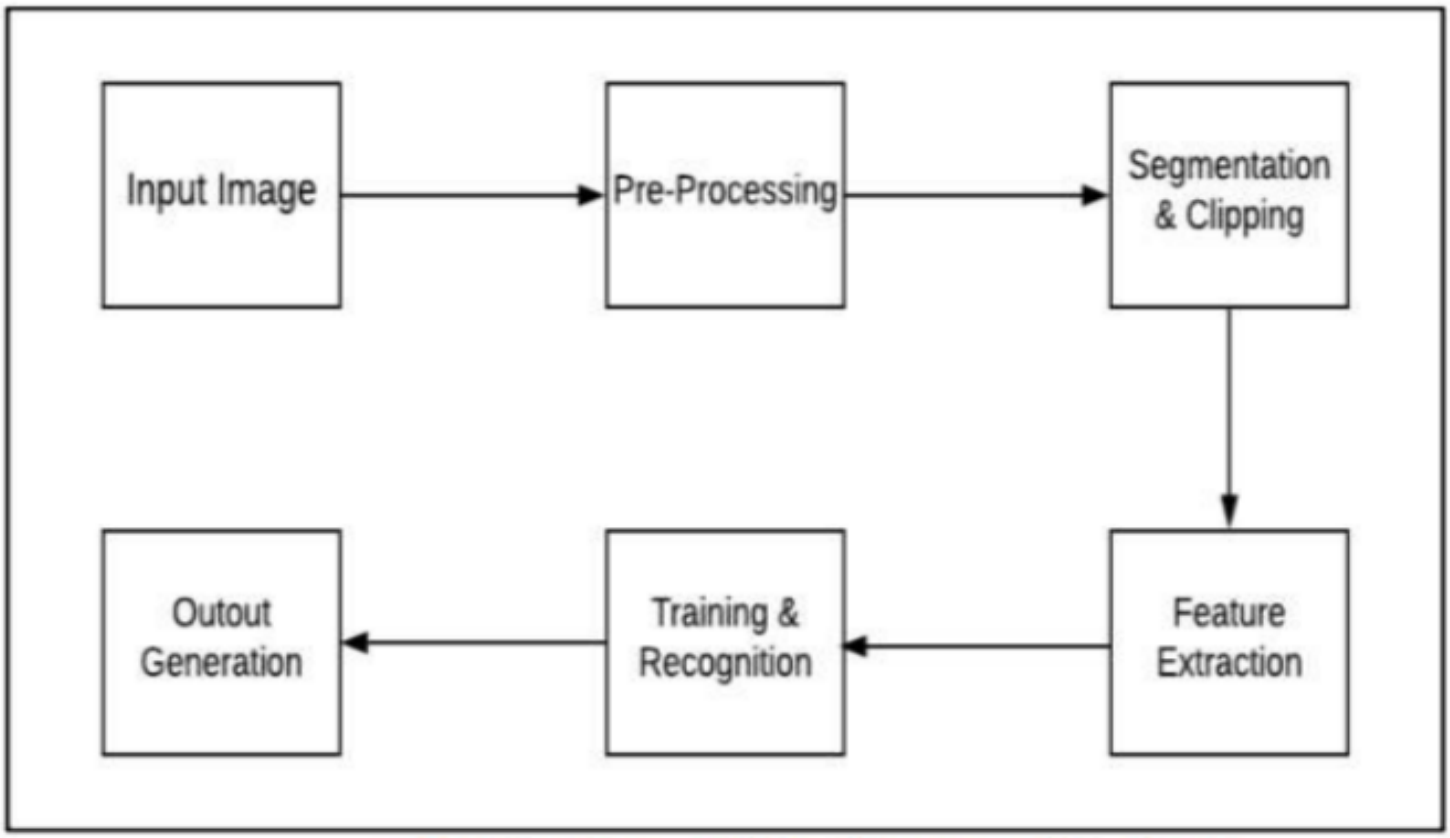
Example: (Simplified) FLOW



Example: DFD Level 0 (Industry Standard)



5.2 Solution and Technical Architecture



Architecture of the Proposed System

5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNIST data set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	Low	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

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

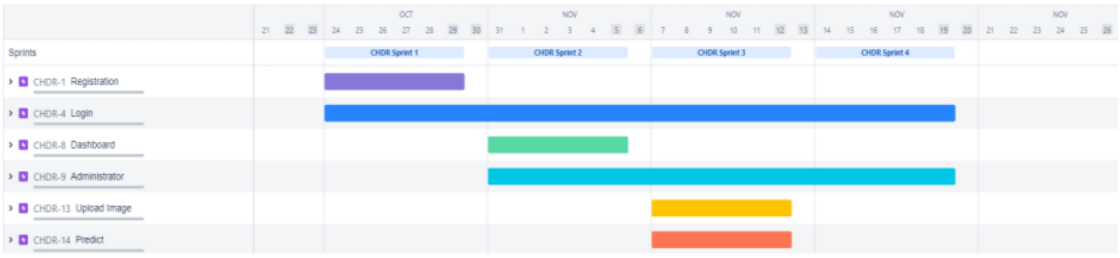
6.1 Sprint planning and Estimation

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

6.3 Reports From JIRA

Figure 1:



7. CODING AND SOLUTIONING

Feature 1:

1.Registration:

User need to register their credentials for the authentication aka login and may use the application.

2.Login:

User need to put username and password into the forms field and they will get authenticated if their credentials were right and redirected to the home page with appropriate link to get into the extraction page

3.Logout:

User may logout the applications if they need to exit the application

Feature 2:

1.Recognition:

i) User can give input as images.
ii) By this digits can be easily recognized and it can be in JPEG or PNG format.
iii) Data accuracy is fully protected.

8.TESTING

8.1 Test Cases

```
#import the library for loading the model
 from keras.models import load_model
 import matpotlib.pyplot as plt


 #Load the Model model
#load_model("MNIST.h5")
 from PIL import Image
import numpy as np
 import matplotlib.pyplot as plt


 #Preview the Sample data
 img = Image.open(r"D:\Images\image1.jpg")
 plt.imshow(img)


 #Test the model with User's Input
for index in range(1,5):


    img = Image.open(r"D:\Images\image" +str(index)+'.jpg').convert("L")
    img = img.resize((28,28))
    img2arr = np.array(img)
    img2arr = img2arr.reshape(1,28,28,1)
    y_pred = model.predict(img2arr)
    print(ypred)
    print(f"The image{index} predicted as value {np.argmax(y_pred)}")
```

8.2 USER ACCEPTANCE:

Purpose of Document:

The purpose of this document is to briefly explain the test coverage and open issues of the [A Novel Method for Handwritten Digit Recognition System] project at the time of the release to User Acceptance Testing (UAT).

Analyzing the Data

```
print(X_train.shape)
print(X_test.shape)
```

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

```
X_train[0]
```

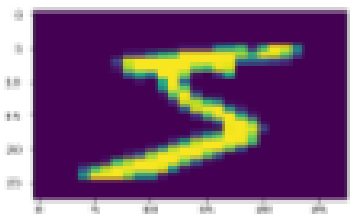
```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   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,   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,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
         90,   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, 139, 253,
        190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
        253,  70,   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,  35,
        241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
```

```
        0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  45, 186, 253, 253, 150,  27,   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,  16,  93, 252, 253, 187,   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, 249, 253, 249,  64,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
        148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
        253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
        253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
        195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
         11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   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,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0]], dtype=uint8)
```

```
y_train[0]
```

```
5
```

```
plt.imshow(X_train[0])
```



9. RESULTS:

Loading the 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 11490434/11490434 [==============================] - 0s 0us/step

10. ADVANTAGES & DISADVANTAGES:

10.1 ADVANTAGES:

- They can scan the handwritten text
- it extracts text & can embed the text
- It reduces the hardness in recognizing variety of handwritten digits
- Can be used by clerks, financiers, accountants

10.2 DISADVANTAGES

- To Scan the fle they need camera
- They need internet connection
- Need to have basic computer knowledge

11.CONCLUSION

Using Neural Network system, back-propagation learning, to recognize handwritten digits was very successful. An image, which contained 100 samples of each number, was trained and tested.

The accuracy rate of recognizing the number was 99%. This accuracy rate is very high. From the training and testing results, it was concluded that the system had more trouble identifying numeral.

This maybe caused by the fact that the digit is running together or maybe it is not fully connected. The system was not stable. It gave different training and testing results every day for each numeral.

12.FUTURE SCOPE

It will need to take a close look at the system and should look for improvements for the future. From the net-file, the system was able to produce an image-file. The image-file produced showed the recognized number.

This part will also need more improvements. Apart from the above problems and parts that need improvements, the overall recognition system was successful.

13. APPENDIX

13.1 Source Code

python code1:

```
from flask import Flask, render_template, request
from scipy.misc import imsave, imread, imresize
import numpy as np
import keras.models
import re
 import base64
 import sys
import os


 sys.path.append(os.path.abspath("./model"))
 from load import *
 app = Flask(__name__)
global model, graph
model, graph = init()


@app.route('/')
def index():
 return render_template("index.html")

@app.route('/predict/', methods=['GET','POST'])
def predict(): # get data from drawing canvas and save as image
parseImage(request.get_data())


# read parsed image back in 8-bit, black and white mode (L)
 x = imread('output.png', mode='L')
 x = np.invert(x)
 x = imresize(x,(28,28))


# reshape image data for use in neural network
x = x.reshape(1,28,28,1)
```

```
with graph.as_default():
 out = model.predict(x)
 print(out) print(np.argmax(out, axis=1))
 response = np.array_str(np.argmax(out, axis=1))
 return response

def parseImage(imgData):
 # parse canvas bytes and save as output.png
imgstr = re.search(b'base64,(.*)', imgData).group(1)
 with open('output.png','wb') as output:
 output.write(base64.decodebytes(imgstr))


if __name__ == '__main__':
app.debug = True
port = int(os.environ.get("PORT", 5000))
 app.run(host='0.0.0.0', port=port)
```

python code 2:

```
import numpy as np
import keras.models
from scipy.misc import imread, imresize,imshow
import tensorflow as tf

from keras.models import Sequential
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D

def init():
 num_classes = 10
 img_rows, img_cols = 28, 28
 input_shape = (img_rows, img_cols, 1)
 model = Sequential()
 model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=input_shape))
 model.add(Conv2D(64, (3, 3), activation='relu'))
 model.add(MaxPooling2D(pool_size=(2, 2)))
 model.add(Dropout(0.25))
 model.add(Flatten())
 model.add(Dense(128, activation='relu'))
```

```
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))



#load woeights into new model
model.load_weights("weights.h5")
print("Loaded Model from disk")


#compile and evaluate loaded model
model.compile(loss=keras.losses.categorical_crossentropy,
optimizer=keras.optimizers.Adadelta(),
metrics=['accuracy'])
#loss,accuracy = model.evaluate(X_test,y_test)
#print('loss:', loss)
#print('accuracy:', accuracy)
```

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>Handwritten Recognition System</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<header class="header">
<nav class="navbar">
<ul>
<li>
<a href="#">Home</a>
</li>
<li>
<a href="recognize.html">Recognize</a>
</li>
</ul>
```

```
 </nav>
 </header>
 <div class="bg-pic"></div>
 <main class="main">
 <h1 class="main-heading">Handwritten Recognition System</h1>
 <p class="content">
 <em>
 Handwritten Text Recognition is a technology that is much needed in this world as of today. This digit
 Recognition system is used to recognize the digits from different sources like emails, bank cheque,
papers, images, etc. Before proper implementation of this technology we have relied on writing texts
 with our own hands which can result in errors. It's difficult to store and access physical data with
 efficiency. The project presents recognizing the handwritten digits (0 to 9) from the famous MNIST
 dataset. Here we will be using artificial neural networks convalution neural network.
 </em>
 </p>
 </main>
</body>
</html>
```

style.css:

```
 #clear_button{
 margin-left: 15px;
 font-weight: bold;
 color: blue;
}
#confidence{
 font-family: 'Josefin Sans', sans-serif;
 margin-top: 7.5%;
}
#content{
 margin: 0 auto;
 padding: 2% 15%;
 padding-bottom: 0;
}
```

```
.welcome{
 text-align: center;
 position: relative;
 color: honeydew;
 background-color: greenyellow;
 padding-top: 1%;
 padding-bottom: 1%;
 font-weight: bold;
 font-family: 'Prompt', sans-serif;
}
#team_id{
 text-align: right;
 font-size: 25px;
 padding-right: 3%
 #predict_button{
 margin-right: 15px;
 color: blue;
 font-weight: bold;
}
#prediction_heading{
 font-family: 'Josefin Sans', sans-serif;
 margin-top: 7.5%;
}
#result{
 font-size: 5rem;
}
#title{
 padding: 1.5% 15%;
 margin: 0 auto;
 text-align: center;
}
.btn {
 font-size: 15px;
 padding: 10px;
 -webkit-appearance: none;
 background: #eee;
 border: 1px solid #888;
 margin-top: 20px;
 margin-bottom: 20px;
}
.buttons_div{
```

margin-bottom: 30px;
margin-right: 80px;

13.1 Github :

https://github.com/IBM-EPBL/IBM-Project-9369-1658998253