VISUAL DIGIT CLASSIFIER

VISUAL DIGIT CLASSIFIER

ABSTRACT

Image recognition is widely used in the field of computer vision today. As a kind of image recognition, digit recognition is widely used. Today, the online recognition technology in digit recognition is relatively mature while the offline recognition technology is not. This project mainly introduces an offline recognition system for handwritten digits based on Feed forward Neural Network. The system uses the MINST dataset as a training sample and pre-processes the picture and implemented it using React and Django. It uses a normal Linear layer which takes the input and do the feature extraction and pull the result into a one-dimensional vector. And finally find the highest probability point to determine the result to achieve handwritten digit recognition with the SoftMax function. This application uses React JS as its frontend and Django as the mainframe server. The application of this system can greatly reduce labor costs and improve work efficiency, which is of great significance in many fields.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER1
INTRODUCTION

1. INTRODUCTION

The issue of manually written numerals acknowledgement has been broadly concentrated lately and huge number of pre-processing strategies and arrangement calculations have been created. Notwithstanding, transcribed numerals acknowledgment is as yet a test for us. It can be applied to various fields of application for detection of 10 digits from 0 to 9 using image processing. For example, a mobile robot recognizes the exit number in the subway station to guide people with disabilities, and the robot in elevator could help to press the number button of the destination floor. Also, number detection has already been used for license plate (LP) localization. There have been several efforts like image binarization and labeling methods for number detection. However, this previous method is vulnerable to illumination variation. Our proposed method is using several local features, which are robust to illumination, so that it is possible to detect even when the illumination changes. The proposed number detection can detect only numbers in black/plain backgrounds. We were able to achieve an accuracy rate of 78.4%.

1.1 OBJECTIVE:

The rapid growth of new documents and multimedia news has created new challenges in pattern recognition and machine learning. Handwriting character recognition has become a standard research area due to advances in technologies such as the handwriting capture devices and powerful mobile computers. The objective of this project is to implement a classification algorithm to recognize handwritten digits (0- 9). The main purpose of this project is to find out the recognition performance of the methods. In order to analyze the performance of the methods, data is needed to be used for training using deep learning methods. Then digit data is tested on the desired deep learning technique. The major goal of the proposed system is understanding the Neural Network operation, and applying it to the handwritten recognition using React JS and implementing it using Django.

1.2 PROJECT DESCRIPTION

This project aims to recognize the handwritten digits by using tools from Machine Learning to train the classifiers, so it produces a high recognition performance. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins. Our goal was to implement a pattern classification method to recognize the handwritten digits provided in the MINIST data set of images of hand written digits (0-9). This project is designed using React Js and keeping Django as mainframe server.

The dataset used for our application is composed of 300 training images and 300 testing images, and is a subset of the MNIST dataset (originally composed of 60,000 training images and 10,000 testing images). Each image is a 28 x 28 grayscale (0-255) labeled representation of an individual digit. It is an excellent database for machine learning and pattern recognition methods while needing minimal efforts in preprocessing and formatting.

The most distinct problem when identifying handwritten forms in the data capture procedure is poor quality or illegible handwriting. One clear example is where Educational Summit in 2012 found that 25- 35% of pupils at a secondary school have not obtained competency in handwriting skills. That means the forms filled out by hand could produce an on-going challenge to the data gathering procedure.

1.3 SCOPE

The scope of this paper is handwritten digit recognition regarding the application of machine learning algorithms based on image pre-processing and feature extraction.

Additionally, the purposes are not only to improve the current recognition performance, but also to seek the highest reliability in the applications of handwritten digits.

This thesis has the following limitations:

- A handwritten digit dataset is vague in essence because there may not always be perfectly straight lines, and different people's writings are more or less sloped;
- The curves are not necessarily smooth like the printed characters;

1.4 LITERATURE SURVEY

[1] K. Gaurav, Bhatia P. K. [5]

This paper deals with the various pre-processing techniques involved in the character recognition with different kind of images ranges from a simple handwritten form based documents and documents containing colored and complex background and varied intensities. In this, different preprocessing techniques like skew detection and correction, image enhancement techniques of contrast stretching, binarization, noise removal techniques, normalization and segmentation, morphological processing techniques are discussed.

It was concluded that using a single technique for preprocessing, we can't completely process the image. However, even after applying all the said techniques might not possible to achieve the full accuracy in a preprocessing system.

[2] Sushree Sangita Patnaik and Anup Kumar Panda May 2011

This paper proposes the implementation of particle swarm optimization (PSO) and bacterial foraging optimization (BFO) algorithms which are intended for optimal harmonic compensation by minimizing the undesirable losses occurring inside the APF itself. The efficiency and effectiveness of the implementation of two approaches are compared for two different conditions of supply. The total harmonic distortion (THD) in the source current which is a measure of APF performance is reduced drastically to nearly 1% by employing BFO. The results demonstrate that BFO outperforms the conventional and PSO based approaches by ensuring excellent functionality of APF and quick prevail over harmonics in the source current even under unbalanced supply.

[3] Salvador España-Boquera ,

In this paper hybrid Hidden Markov Model (HMM) model is proposed for recognizing unconstrained offline handwritten texts. In this, the structural part of the optical model has been modelled with Markov chains, and a Multilayer Perceptron is used to estimate the emission probabilities. In this paper, different techniques are applied to remove slope and slant from handwritten text and to normalize the size of text images with supervised learning methods. The key features of this recognition system were to develop a system having high accuracy in preprocessing and recognition, which are both based on ANNs.

In a modified quadratic classifier-based scheme to recognize the offline handwritten numerals of six popular Indian scripts is proposed. Multilayer perceptron has been used for recognizing Handwritten English characters [8]. The features are extracted from Boundary tracing and their Fourier Descriptors. The character is identified by analyzing its shape and comparing its features that distinguish each character. Also, an analysis has been carried out to determine the number of hidden layer nodes to achieve high performance of the back propagation network. A recognition accuracy of 94% has been reported for Handwritten English characters with less training time.

CHAPTER 2

SYSTEM ANALYSIS

2.1 SYSTEM ANALYSIS

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action and the software should run with multiple threads.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal and it should be free of relating with the multiple threads. .

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained,

from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Existing System

These days, an ever-increasing number of individuals use pictures to transmit data. It is additionally main stream to separate critical data from pictures. Image Recognition is an imperative research area for its generally used applications. In general, the field of pattern recognition, one of the difficult undertakings is the precise computerized recognition of human handwriting.

Without a doubt, this is a very difficult issue because there is an extensive diversity in handwriting from an individual to another individual.

Proposed System

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. We proposed new fresh approach to existing work for improvements of learning algorithm and novel feature extraction method for Handwritten Digit Recognition. We focus to implementing method and architecture uses GPUs for parallel processing to speed up learning process Our study forces us to improve following parameters to be considered to improve the system.

1) Decrease training time without affecting accuracy

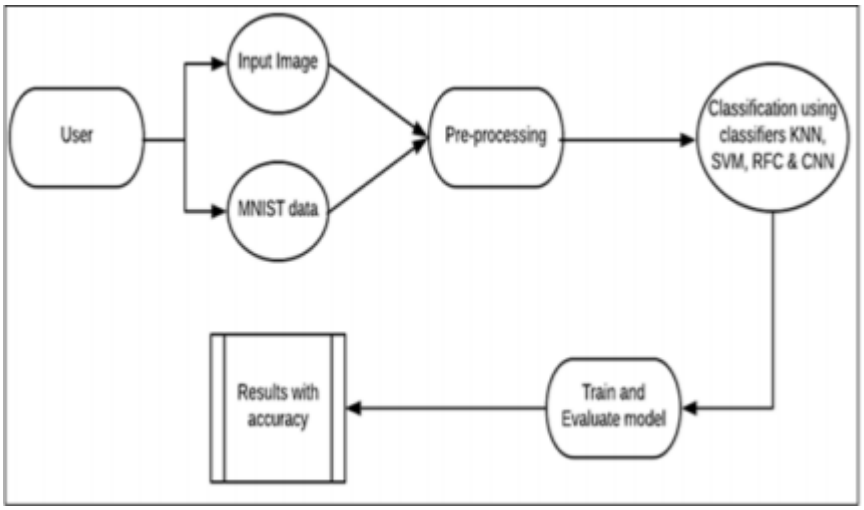
2) Implementing method and architecture uses GPU for parallel processing to speed up learning process.

3) Implementing it as a web-based application using React JS and Django which makes it easier for the user to access it.

2.2 SYSTEM ARCHITECTURE

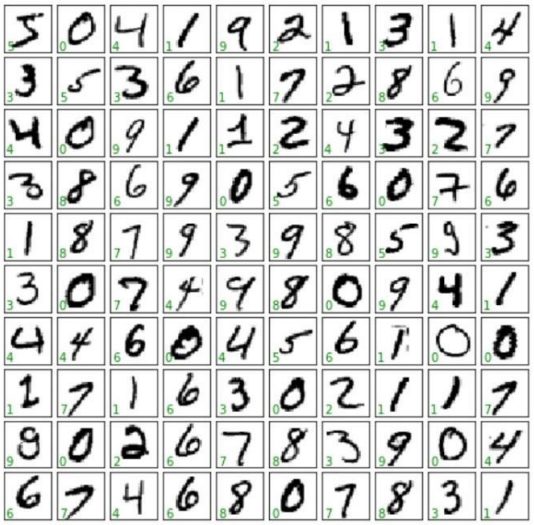
Introduction

The reason behind this document is to look into the design possibilities of the proposed system, such as architecture design, block diagram, sequence diagram, data flow diagram and user interface design of the system in order to define the steps such as pre-processing, classification, training and recognition of digits.



A. Pre-Processing:

The role of the pre-processing step is it performs various tasks on the input image. It basically upgrades the image by making it reasonable for segmentation. The fundamental motivation behind pre-processing is to take off a fascinating example from the background. For the most part, noise filtering, smoothing and normalization are to be done in this stage. The pre-processing additionally characterizes a smaller portrayal of the example. Binarization changes over a gray scale image into a binary image. The initial approach to the training set images that are to be processed in order to reduce the data, by thresholding them into a binary image. The below figure shows a sample of images taken from the MNIST database. For the slant detection and correction of the handwritten digits, the first step is to find the center of mass in the image to determine how much is needed to offset the image. Then, next step is to define the maximum angle of the raw image 45° and detect the covariance matrix of the 55 image pixel strengths.



B. **Classification and Recognition:**

In the classification and recognition step the extracted feature vectors are taken as an individual input to each of the following classifiers. In order to showcase the working system model extracted features are combined and defined using a pretrained mode resnet18.

C. Digit Recognition

The digit is extracted from the image after the pre-processing. The hog classifier is used to extract the features of the image [7]. Digit Classification Using HOG Features

- Acquire a labelled data set with images of the desired object.
- Partition the data set into a training set and a test set.
- Train the resnet18 using features extracted from the training set.
- Test the classifier using features extracted from the test set.

Implementation

A. SOFTWARE PLATFORM

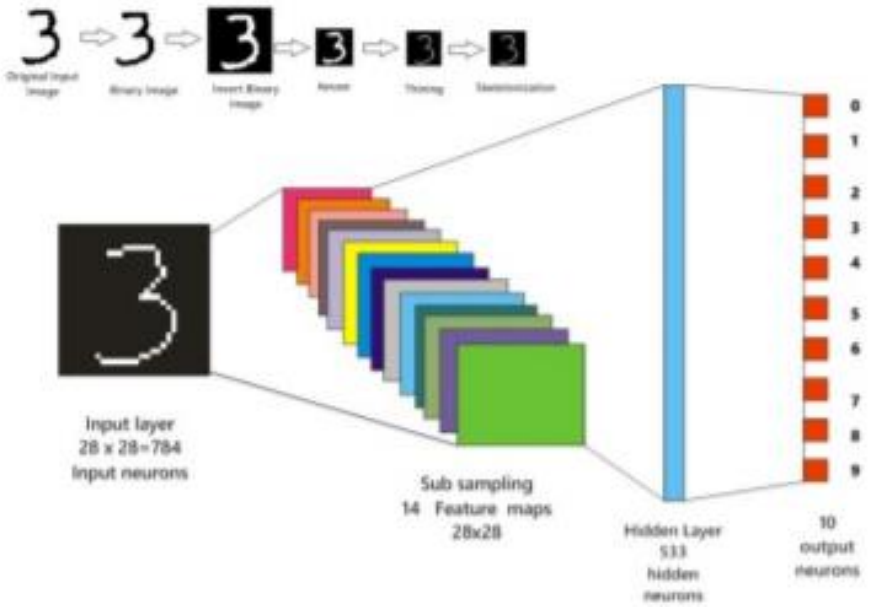
Python 3.7: Python is broadly utilized universally and is a high-level programming language. It was primarily introduced for prominence on code, and its language structure enables software engineers to express ideas in fewer lines of code. Python is a

programming language that gives you a chance to work rapidly and coordinate frameworks more effectively.

React JS: The canvas draw, i.e., the area where the user draws the image is designed using react. React is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Pytorch: Pytorch is where the neural network implementation is done. A pretrained model called resnet18 is used to train the data. Training is done using transfer learning.

2.2.1 Architecture:



2.3 UML DIAGRAMS

2.3.1 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. User first draw/ upload an image which is sent to the system. The system will take the image and do the appropriate pre processing and give the image to

The purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

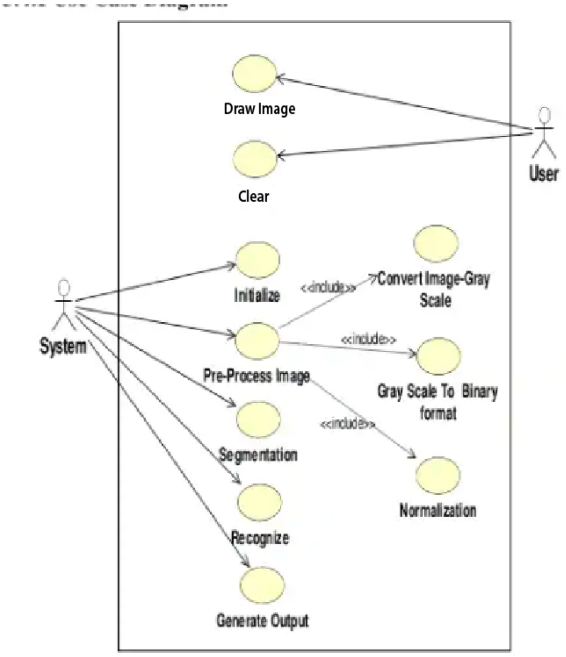
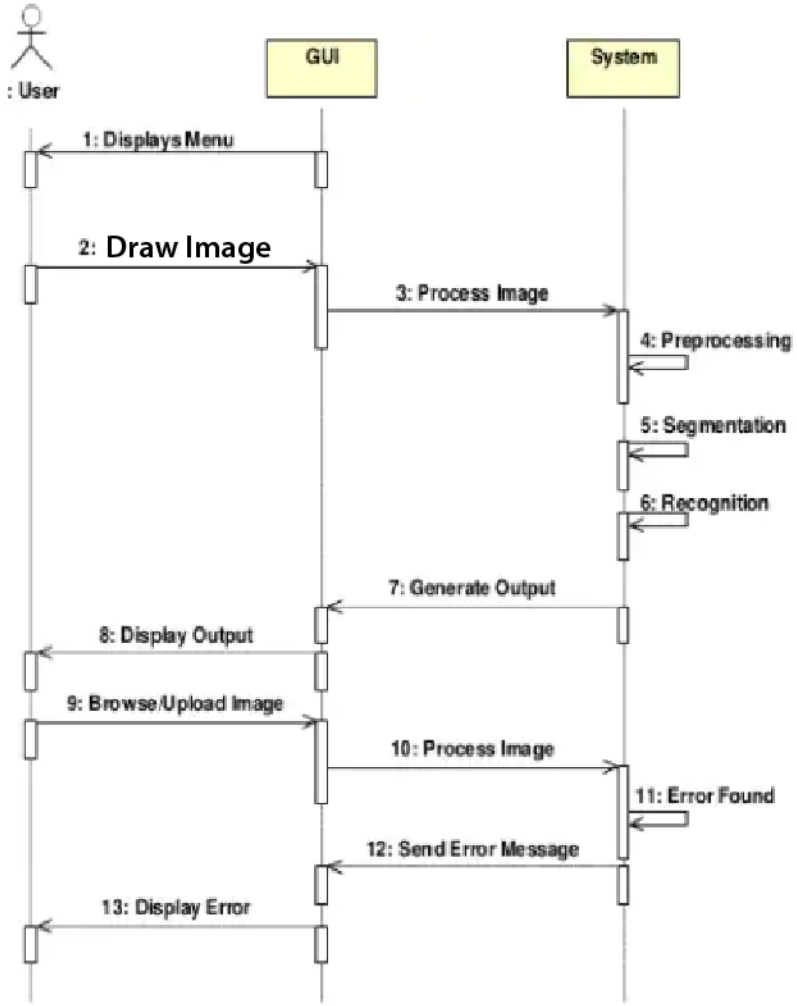


Fig 2.3.1.1.Use Case Diagram

2.3.2. SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario



2.3.3.1 Sequence Diagram

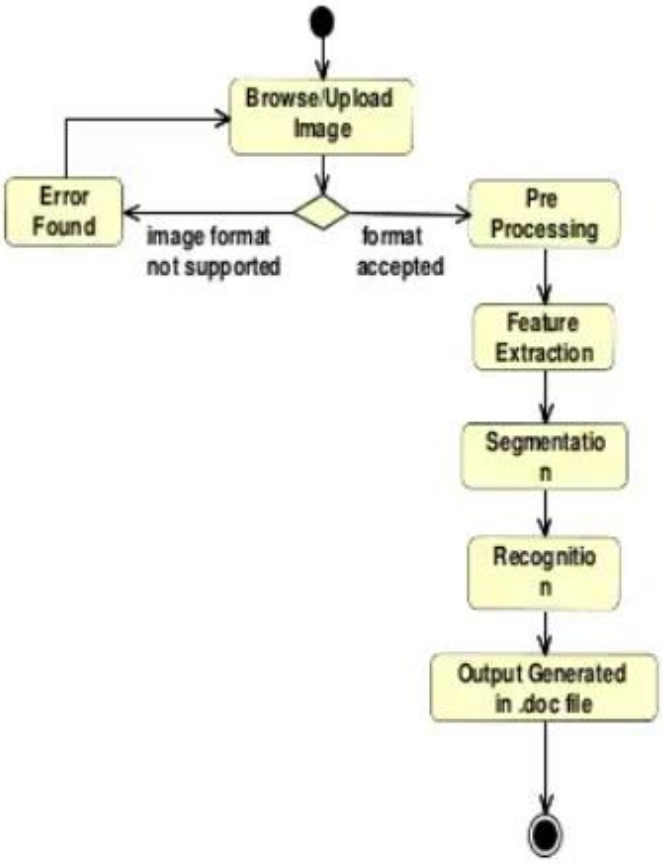
2.3.3. ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions[1] with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organisational processes (i.e. workflows).[2][3] Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows.

The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
- an encircled black circle represents the end (final state).



CHAPTER 3

MODULES

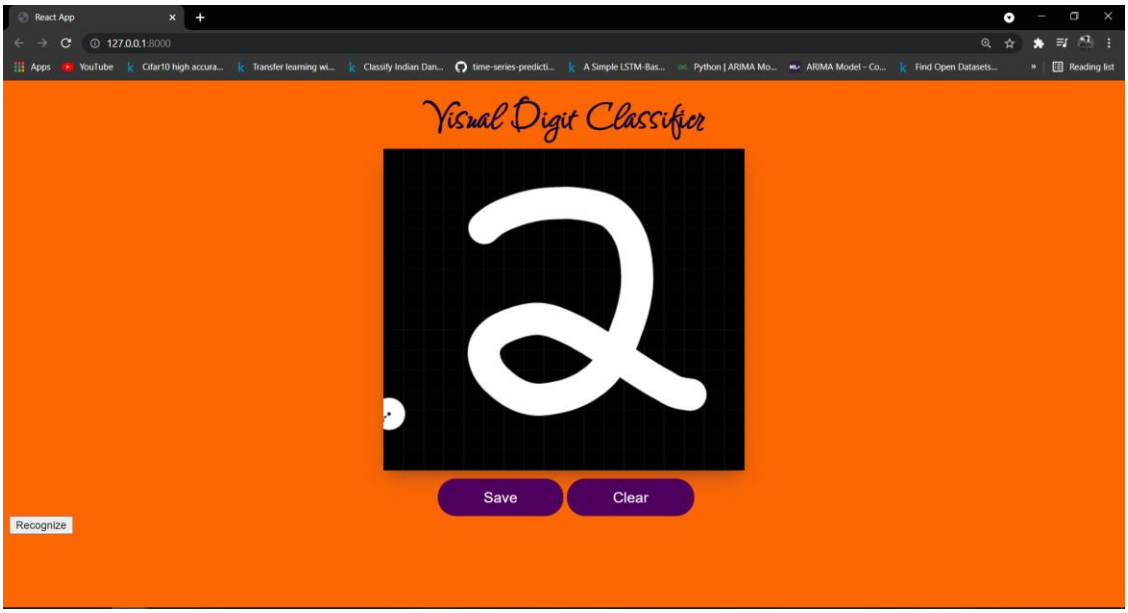
3.MODULES

- CanvaDraw – Paint tool
- Recognize

3.1MODULES DESCRIPTION:

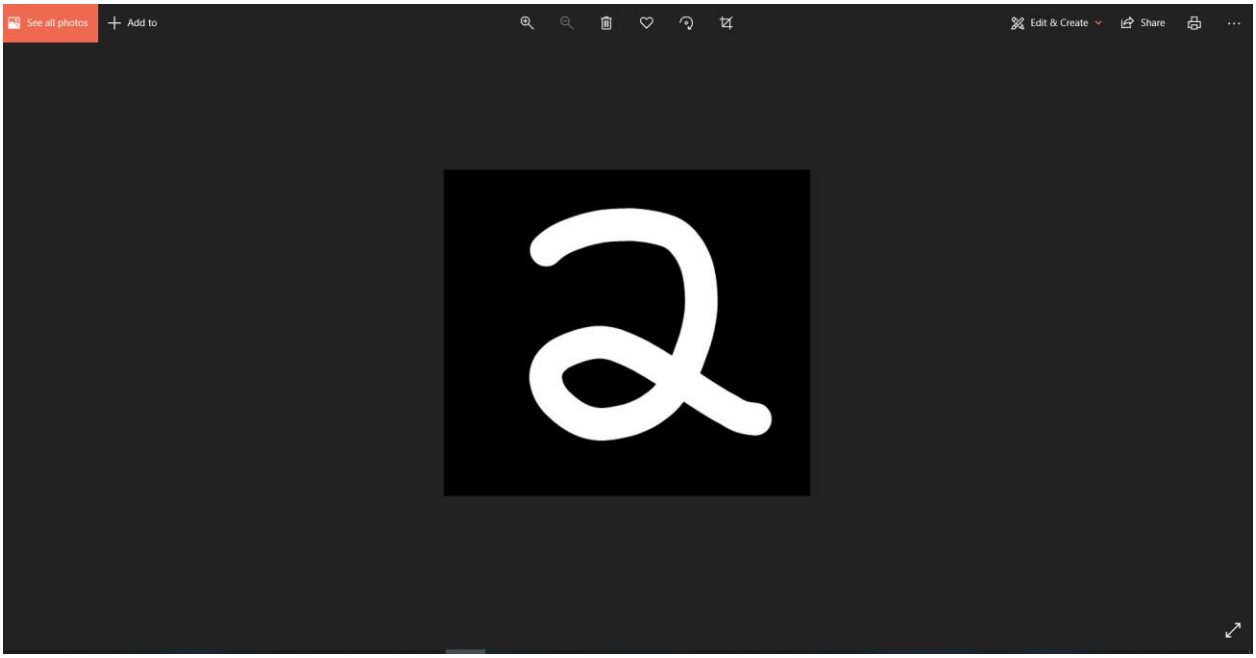
3.1.1 CanvaDraw:

Admin logins with username and password. Admin authenticates all the information .User can also login and can view the information. User can't update the information. Admin can delete the users and can add the users. So that the users can view only the information after completing their login with their username and password. Everything is stored in db and it can be viewed only by the admin.



3.1.2 Save Image

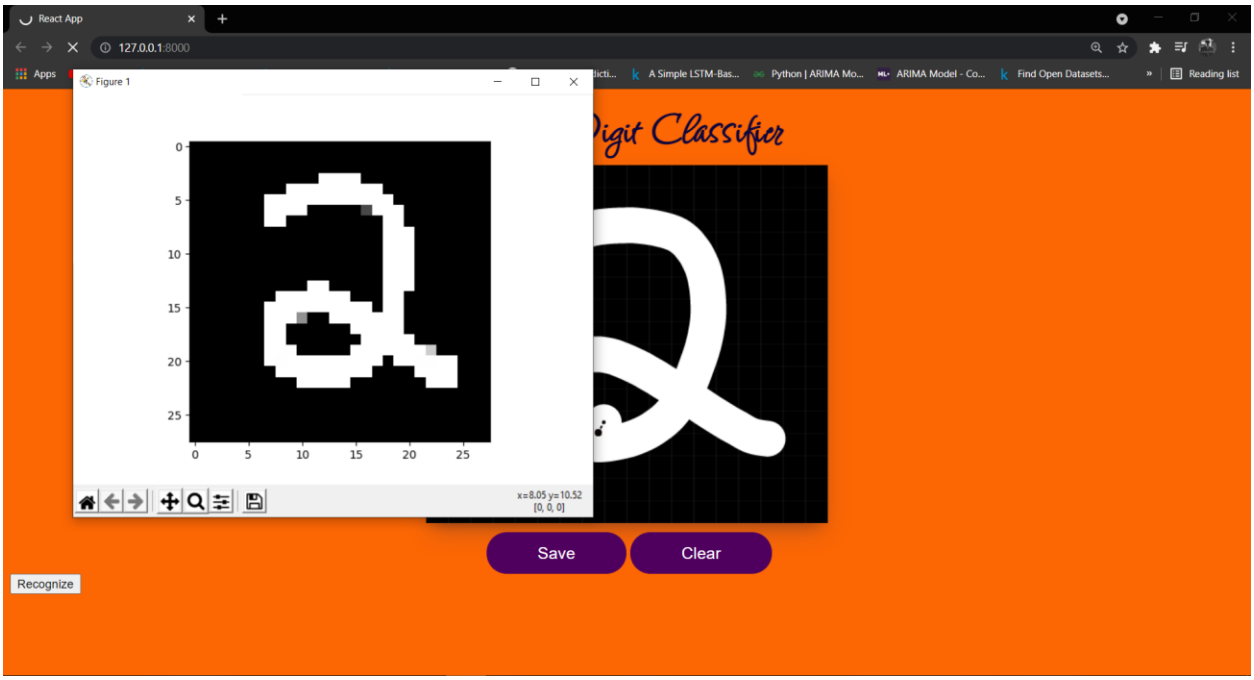
This module helps to register the details about the User Account Information. This is done by entering some personal details, such as Name,address,Occupation



To create an account customer have to enter their details such as Name, Age, Gender, Address, Occupation, Date of birth and Amount how much they want to credit in the account. The request are sent to the admin of the bank to create an account.

3.2: Recognize :

This module will recognize the text that the user draws in the canvas window. It uses python Text-To-Speech conversion to convert the words that are given in text to speech. This is how the recognition process takes place.


React App
127.0.0.1:8000
Figure 1
Digit Classifier
Save
Clear
Recognize

CHAPTER 4

CONCLUSION AND FUTURE ENHANCEMENTS

4. CONCLUSION

In this paper, the Handwritten Digit Recognition using Deep learning methods has been implemented. The most widely used Machine learning algorithms, Feed forward neural networks is implemented and have been trained and tested on the same data in order acquire the comparison between the classifiers. Utilising these deep learning techniques, a high amount of accuracy can be obtained. Using Pytorch as backend, a model is able to give accuracy of about 78.74%. It is then implemented by designing a paint tool using React JS and integrating both using the mainframe server Django.

4.1 FUTURE ENHANCEMENT:

In this paper, although the method of addressing the research question was found by training on the MNIST database, there are still some problems that need to be explored and solved in the future. For example, the accuracy of the feed forward models based on the combination of preprocessing is smaller than the initial experiment. So, the causes of these problems mentioned above should be analyzed and found to be resolved in the future. There are also some natural expansions to this research that would assist extend and reinforcing the results. The benchmark database of MNIST was developed for this work, and it is an excellent database for machine learning and pattern recognition methods while making minimal efforts in preprocessing and formatting. However, not all handwritten digit sets are normalized in size, or centered and stored sequentially as 28x28 pixel images in grayscale in the actual cases. Hence, it would be necessary to add similar experiments with distinct databases regarding the features array dimension and various language scripts such as Chinese, Arabic, French, et

APPENDICES

SAMPLE CODING

Canvas.js

```
import React,{createRef} from 'react';
import CanvasDraw from "react-canvas-draw";

import Button from '../components/UI/Button';

import styles from './canvas.module.css';


const Canvas= () =>{


    const storeCanvas=createRef();


    const saveHandler= () =>{

        const imgCanvas=storeCanvas.current.canvas.drawing.toDataURL('image/jpeg'
);

        const convertImg = document.createElement('a');

        const imgName = 'digit.jpeg';

        convertImg.download =imgName;

        convertImg.href =imgCanvas;

        convertImg.click()

    }






    const clearHandler= () =>{

        storeCanvas.current.clear()

    }
```

```
    return(
        <div>

         <CanvasDraw
            ref={storeCanvas}
            brushColor="white"
            backgroundColor="#000000"
            canvasWidth="450px"
            className={styles.canvas}
         />

         <Button
         type="submit"
         onClick={saveHandler}>
             Save
        </Button>

            <Button
            type="submit"
            onClick={clearHandler}>
                Clear
            </Button>

        </div>
    )
}
export default Canvas;
```

App.js

```
import './App.css';
import Canvas from './components/canvas';

function App() {
  return (
    <div className="App">
      <h1>Visual Digit Classifier</h1>
      <Canvas />
    </div>
  );
}
export default App;
```

view .py

```
from django.shortcuts import render
from django.views.generic import TemplateView
from django.http import HttpResponse
from .inference.model import recognize
import pyttsx3

# Create your views here.
file_path = "F:/vdc/Hello.jpeg"
def index(request):
    return render(request, TemplateView.as_view(template_name = 'index.html'))

def numera(audio):
    engine = pyttsx3.init('sapi5')
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[1].id)
    engine.setProperty('rate',160)
    engine.say(audio)
    engine.runAndWait()

def recog(request):

    result = recognize(file_path)
    print(result)
    a=str(list(result.keys())[1])
    numera("The number you have written is:"+a)
    mydict = {"recognized":"Possibilites: "+str(result)}
    return render(request, 'index.html',mydict)
```

urls.py

```
from django.contrib import admin
from django.urls import path
from digit_classifier import views
from django.views.generic import TemplateView

urlpatterns = [
    path('', TemplateView.as_view(template_name = 'index.html')),
    path('recog', views.recog),
    path('admin/', admin.site.urls),
]
```