Karnatak  University

Dharwad

A picture containing text, device

Description automatically generated

**Global Educare Foundation’s**

**Global College Of Computer Application**

**A Project Report on**

**“Handwritten Digit Recognition”**

**Submitted in partial fulfillment for the award of the degree**

Bachelor of Computer Application

Of Karnatak University, Dharwad.

**Under the Guidance of**

Prof. **Ashwinikumar Koti**

**Submitted By**

Vishwajeet Kulkarni – 20U10586

Akash Kalkeri – 20U10507

Karnatak  University

Dharwad



**Global Educare Foundation’s**

**Global College Of Computer Application**

**CERTIFICATE**

This is to certify that Mr \_ Vishwajeet Kulkarni\_\_\_\_\_\_\_\_\_ has satisfactorily completed Project work entitled “\_Handwritten Digit Recognition\_\_\_\_\_\_” for the partial fulfillment of Bachelor of Computer Application by Karnatak University, Dharwad for the academic year 2021-2022.

Candidate Name: - Vishwajeet Kulkarni\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Register Number: - \_20U10586\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**HOD Project Guide /Principal**

**Prof. Shweta Konanavar Prof.Ashwinikumar Koti**

**Examiner 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Examiner 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Acknowledgement

Apart from the efforts of us, the success of any project depends on the encouragement and Guidelines. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We express our gratitude and regards to our respected and beloved Dean,Dr. Mahesh Deshpande & also our beloved principal and guide Prof. Ashwinikumar Koti for every support extended.

The acknowledgement would be incomplete if we don’t mention our gratitude to staff members of BCA department. We are highly grateful to our external guide who helped us throughout the project.

Finally last but not least we want to forward warm regards to our beloved parents and friends because without their help and support nothing would be possible.

**STUDENT NAME**

**Vishwajeet Kulkarni**

**Akash Kalkeri**

|  |  |
| --- | --- |
| CHAPTER 1. INTRODUCTION   * 1. EXISTING SYSTEM   2. PROBLEM DEFINITION   3. PROPOSED SYSTEM | Page.no  5-10 |
| CHAPTER 2. DESIGN/STRUCTURE  2.1 ACTIVITY DIAGRAM  2.2 DATA FLOW DIAGRAM | 11-13 |
| CHAPTER 3. SYSTEM DESIGN | 14-15 |
| CHAPTER 4. REQUIREMENTS  4.1 HARDWARE REQUIREMENTS  4.2 SOFTWARE REQUIREMENTS | 16-17 |
| CHAPTER 5. CODING  5.1 INTRODUCTION TO PROGRAMMING LANGUAGE  5.2 MODULES  5.3 SAMPLE CODE | 18-27 |
| CHAPTER 6. SCREENSHOTS | 28-32 |
| CHAPTER 7. CONCLUSION  7.1 OUTCOME  7.2 FUTURE SCOPE/ENHANCEMENT | 33-34 |
| CHAPTER 8. REFERENCES AND BIBLIOGRAPHY | 35-36 |

**CONTENTS**

**CHAPTER - 1**

**INTRODUCTION**

Artificial intelligence (AI) is the ability of machines to replicate or enhance human intellect, such as reasoning and learning from experience. Artificial intelligence has been used in computer programs for years, but it is now applied to many other products and services. For example, some digital cameras can determine what objects are present in an image using artificial intelligence software. In addition, experts predict many more innovative uses for artificial intelligence in the future, including smart electric grids.

To understand the idea behind AI, you should think about what distinguishes human intelligence from that of other creatures – our ability to learn from experiences and apply these lessons to new situations. We can do this because of our advanced brainpower; we have more neurons than any animal species.

Today’s computers don’t match the human biological neural network – not even close. But they have one significant advantage over us: their ability to analyze vast amounts of data and experiences much faster than humans could ever hope.

AI lets you focus on the most critical tasks and make better decisions based on acquired data related to a use case. It can be used for complex tasks, such as predicting maintenance requirements, detecting credit card fraud, and finding the best route for a delivery truck. In other words, AI can automate many business processes leaving you to concentrate on your core business.

Research in the field is concerned with producing machines to automate tasks requiring intelligent behavior. Examples include control, planning and scheduling, the ability to answer diagnostic and consumer questions, handwriting, natural language processing and perception, speech recognition, and the ability to move and manipulate objects.

**History of AI and how it progressed**

With so much attention on modern artificial intelligence, it is easy to forget that the field is not brand new. AI has had a number of different periods, distinguished by whether the focus was on proving logical theorems or trying to mimic human thought via neurology.

Artificial intelligence dates back to the late 1940s when computer pioneers like Alan Turing and John von Neumann first started examining how machines could “think.” However, a significant milestone

in AI occurred in 1956 when researchers proved that a machine could solve any problem if it were allowed to use an unlimited amount of memory.

The early years of the 21st century were a period of significant progress in artificial intelligence. The first major advance was the development of the self-learning neural network. By 2001, its performance had already surpassed human beings in many specific areas, such as object classification and machine translation. Over the next few years, researchers improved its performance across a range of tasks, thanks to improvements in the underlying technologies.

The second significant advancement in this period was the development of generative model-based reinforcement learning algorithms. Generative models can generate novel examples from a given class, which helps learn complex behaviors from very little data. For example, they can be used to learn how to control a car from only 20 minutes of driving experience.

In addition to these two advances, there have been several other significant developments in AI over the past decade. There has been an increasing emphasis on using deep neural networks for computer vision tasks, such as object recognition and scene understanding. There has also been an increased focus on using machine learning tools for natural language processing tasks such as information extraction and question answering. Finally, there has been a growing interest in using these same tools for speech recognition tasks like automatic speech recognition (ASR) and speaker identification (SID).

**Different fields of AI to clear common misconceptions**

It’s essential to know that AI is not a single field but a combination of various fields. Artificial Intelligence (AI) is the general term for being able to make computers do things that require intelligence if done by humans. AI can be broken down into two major fields, Machine Learning (ML) and Neural Networks (NN). Both are subfields under Artificial Intelligence, and each one has its methods and algorithms to help solve problems.

Diagram, venn diagram

Description automatically generated

Venn Diagram describing the Structure of AI.

Machine Learning

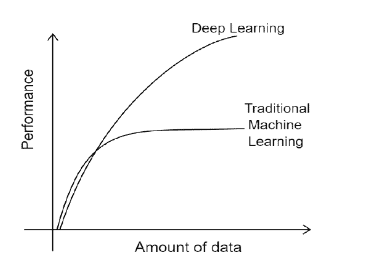
Machine Learning (ML) makes computers learn from data and experience to improve their performance on some tasks or decision-making processes. ML uses statistics and probability theory for this purpose. Machine learning uses algorithms to parse data, learn from it, and make determinations without explicit programming. Machine learning algorithms are often categorized as supervised or unsupervised. Supervised algorithms can apply what has been learned in the past to new data sets; unsupervised algorithms can draw inferences from datasets. Machine learning algorithms are designed to strive to establish linear and non-linear relationships in a given set of data. This feat is achieved by statistical methods used to train the algorithm to classify or predict from a datase

Deep Learning

Deep learning is a subset of machine learning that uses multi-layered artificial neural networks to deliver state-of-the-art accuracy in object detection, speech recognition and language translation. Deep learning is a crucial technology behind driverless cars and enables the machine analysis of large amounts of complex data — for example, recognizing the faces of people who appear in an image or video.

Neural Networks

Neural networks are inspired by biological neurons in the human brain and are composed of layers of connected nodes called “neurons” that contain mathematical functions to process incoming data and predict an output value. Artificial neural network learns by example, similarly to how humans learn from our parents, teachers, and peers. They consist of at least three layers: an input layer, hidden layers, and an output layer. Each layer contains nodes (also known as neurons) which have weighted inputs that compute the output.



The performance of traditional machine learning models plateau and throwing any more data doesn’t help improve the performance. Deep learning models continue to improve in performance with more data.

OBJECTIVES OF THE PROJECT :-

The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

# • 1.1 Existing System:-

Various algorithms used for implementing handwritten digit recognition systems consist of Proximal Support Vector Machine (PSVM), Multilayer Perceptron, Support Vector Machine (SVM), Random Forest, Bayes Net, Naive Bayes, J48, Random Tree. According to previous research work, accuracies provided by these algorithms are of the order of: ● Proximal SVM - 98% ● Multilayer Perceptron - 90% ● SVM - 87% ● Random Forest - 85% ● Bayes Net - 84% ● Naive Bayes - 81% ● J48 - 79% ● Random Tree - 75% .

Even though these algorithms may prove to be useful in some of the applications based on this technology, many other applications such as banking industry applications require better results which can be achieved using other algorithms as compared to the algorithms that are mentioned above.

# • 1.2 Problem definition:-

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc. The main problem lies within the ability on developing an efficient algorithm that can recognize hand written digits, which is submitted by users by the way of a scanner, tablet, and other digital devices. This project presents an approach to off-line handwritten digit recognition based on different machine learning techniques. The main objective of this project is to ensure the effectiveness and reliability of the approached recognition of handwritten digits.

# • 1.3 Proposed System:-

In this project, we are going to implement a handwritten digit recognition app using the MNIST dataset. We will be using a special type of deep neural network that is **Convolutional Neural Network (CNN)**. In the end, we are going to build a GUI in which you can draw the digit and recognize it straight away, and also accuracy of digit recognized is displayed in percentage. The digit can be handwritten using cursor/pointer on the GUI and the digit entered can be predicted.

# CHAPTER-2

# DESIGN/STRUCTURE

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, feature extraction, segmentation, classification and recognition of digits.

**2.1 ACTIVITY DIAGRAM**

**Diagram

Description automatically generated**

In this project, we have successfully built a Python deep learning project on handwritten digit recognition app. We have built and trained the Convolutional neural network which is very effective for image classification purposes. Later on, we built the GUI where we draw a digit on the canvas then we classify the digit and show the results.

Here the user inputs handwritten digits on canvas which will be processed in comparison with MNIST data and the result with accuracy is displayed.

**2.2 DATA FLOW DIAGRAM**

**Diagram

Description automatically generated**

**CHAPTER-3**

SYSTEM DESIGN

The project is designed completely using python 3.10 using various python libraries, GUI (Graphical User Interface), CNN (Convolutional Neural Network) and MNIST dataset.

To build the handwritten Digit Recognition the following design and steps have been implemented.

**Import the Libraries and Dataset**

First, we are going to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So, we can easily import the dataset and start working with it. The mnist.load\_data() method returns us the training data, its labels and also the testing data and its labels.

**Pre-process the data.**

The image data cannot be fed directly into the model so we need toperform some operations and process the data to make it ready for our neural network. The dimension of the training data is (60000,28,28). The CNN model will require one more dimension, so we reshape the matrix to shape (60000,28,28,1).

**Create the Model.**

Now we will create our CNN model in Python data science project. A CNN model generally consists of convolutional and pooling layers. It works better for data that are represented as grid structures, this is the reason why CNN works well for image classification problems. The dropout layer is used to deactivate some of the neurons and while training, it reduces offer fitting of the model. We will then compile the model.

**Training the Model.**

The model.fit() function of Keras will start the training of the model. It takes the training data, validation data, epochs, and batch size.

It takes some time to train the model. After training, we save the weights and model definition in the ‘mnist.h5’ file.

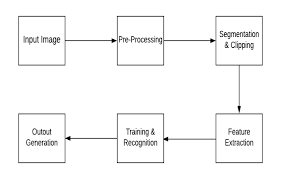
**Evaluating the Model**

We have 10,000 images in our dataset which will be used to evaluate how good our model works. The testing data was not involved in the training of the data therefore, it is new data for our model. The MNIST dataset is well balanced so we can get around 99% accuracy.

**Creating the Model**

Now for the GUI, we have created a new file in which we build an interactive window to draw digits on canvas and with a button, we can recognize the digit. The Tkinter library comes in the Python standard library. We have created a function predict\_digit() that takes the image as input and then uses the trained model to predict the digit.

Then we create the App class which is responsible for building the GUI for our app. We create a canvas where we can draw by capturing the mouse event and with a button, we trigger the predict\_digit() function and display the results.

****

**Basic Design Block Diagram**

**CHAPTER-4**

REQUIREMENTS

The Python project requires you to have basic knowledge of Python programming, deep learning with Keras library and the Tkinter library for building GUI. The software is not complex hence can be run even on lower specs laptops or PC.

**4.1 HARDWARE REQUIREMENTS.**

-A Laptop or Desktop.

-Memory - ROM 256 GB or more.

RAM 2 GB or more.

**4.2 SOFTWARE REQUIREMENTS.**

- Windows 7 or higher.

- Python 2.9 or higher (currently using 3.10)

- Terminal.

**CHAPTER-5**

**CODING.**

5.1 INTRODUCTION OF PROGRAMMING LANGUAGE

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limitedto only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL)natural language processing, data science etc. Python has a lot of libraries and tools for every need of this project. For Handwritten Digit Recognition the python libraries and tools such as Numpy, Keras, TensorFlow, Tkinter and Pillow are used.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

5.2 MODULES.

**PYTHON and LIBRARIES**

The python idle editor is used to write and edit code for the entire software. To run the program code, it has to be run through the terminal. To install the libraries the following command is used in the terminal- pip install numpy, tensorflow, keras, pillow.

1. Keras: Keras is a powerful and easy-to-use free open-source Python library for developing and evaluating deep learning models. 18 It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

2. TensorFlow: TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow tutorial is designed for both beginners and professionals. Our tutorial provides all the basic and advanced concept of machine learning and deep learning concept such as deep neural network, image processing and sentiment analysis. TensorFlow is one of the famous deep learning frameworks, developed by Google Team. It is a free and open-source software library and designed in Python programming language, this tutorial is designed in such a way that we can easily implements deep learning project on TensorFlow in an easy and efficient way. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems. It can run on single CPU systems, GPUs as well as mobile devices and largescale distributed systems of hundreds of machines.

3. Numpy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. Numpy which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. It is an open source project and you can use it freely. NumPy stands for Numerical Python. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important. 19

4. Pillow: Pillow is a free and open source library for the Python programming language that allows you to easily create &s manipulate digital images. Pillow is built on top of PIL (Python Image Library). PIL is one of the important modules for image processing in Python. However, the PIL module is not supported since 2011 and doesn’t support python 3. Pillow module gives more functionalities, runs on all major operating system and support for python 3. It supports wide variety of images such as “jpeg”, “png”, “bmp”, “gif”, “ppm”, “tiff”. You can do almost anything on digital images using pillow module. Apart from basic image processing functionality, including point operations, filtering images using built-in convolution kernels, and color space conversions.

5. Tkinkter: Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. We need to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So we can easily import the dataset through Keras. The mnist.load\_data() method returns the training data, its labels along with the testing data and its labels.

**MNIST Data Set:**

Modified National Institute of Standards and Technology (MNIST) is a large set of computer vision dataset which is extensively used for training and testing different systems. It was created from the two special datasets of National Institute of Standards and Technology (NIST) which holds binary images of handwritten digits. The training set contains handwritten digits from 250 people, among them 50% training dataset was employees from the Census Bureau and the rest of it was from high school students. However, it is often attributed as the first datasets among other datasets to prove the effectiveness of the neural networks.

The database contains 60,000 images used for training as well as few of them can be used for cross validation purposes and 10,000 images used for testing. All the digits are grayscale and positioned in a fixed size where the intensity lies at the centre of the image with 28×28 pixels. Since all the images are 28×28 pixels, it forms an array which can be flattened into 28\*28=784 dimensional vector. Each component of the vector is a binary value which describes the intensity of the pixel.

**CNN:**

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

The convolutional layers are the key component of a CNN, where filters are applied to the input image to extract features such as edges, textures, and shapes. The output of the convolutional layers is then passed through pooling layers, which are used to down-sample the feature maps, reducing the spatial dimensions while retaining the most important information. The output of the pooling

layers is then passed through one or more fully connected layers, which are used to make a prediction or classify the image.

CNNs are trained using a large dataset of labeled images, where the network learns to recognize patterns and features that are associated with specific objects or classes. Once trained, a CNN can be used to classify new images, or extract features for use in other applications such as object detection or image segmentation.

CNNs have achieved state-of-the-art performance on a wide range of image recognition tasks, including object classification, object detection, and image segmentation. They are widely used in computer vision, image processing, and other related fields, and have been applied to a wide range of applications, including self-driving cars, medical imaging, and security systems.

* A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals.
* CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces.
* This characteristic that makes convolutional neural network so robust for computer vision.
* CNN can run directly on a underdone image and do not need any pre-processing.
* A convolutional neural network is a feed forward neural network, seldom with up to 20.
* The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer.
* CNN contains many convolutional layers assembled on top of each other, each one competent of recognizing more sophisticated shapes.
* With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces.
* The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a alike fashion and even use the knowledge for a multitude of duty such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

The construction of a convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order.

It is the sequential design that give permission to CNN to learn hierarchical attributes.

In CNN, some of them followed by grouping layers and hidden layers are typically convolutional layers followed by activation layers.

**Train the Model.**

The model.fit() function of Keras will start the training of the model. It takes the training data, validation data, epochs, and batch size.

It takes some time to train the model. After training, we save the weights and model definition in the ‘mnist.h5’ file.

5.3 SAMPLE CODE

To Train the model

import keras

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D, MaxPooling2D

from keras import backend as K

# the data, split between train and test sets

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

print(x\_train.shape, y\_train.shape)

x\_train = x\_train.reshape(x\_train.shape[0], 28, 28, 1)

x\_test = x\_test.reshape(x\_test.shape[0], 28, 28, 1)

input\_shape = (28, 28, 1)

# convert class vectors to binary class matrices

y\_train = keras.utils.to\_categorical(y\_train, 10)

y\_test = keras.utils.to\_categorical(y\_test, 10)

x\_train = x\_train.astype('float32')

x\_test = x\_test.astype('float32')

x\_train /= 255

x\_test /= 255

print('x\_train shape:', x\_train.shape)

print(x\_train.shape[0], 'train samples')

print(x\_test.shape[0], 'test samples')

batch\_size = 128

num\_classes = 10

epochs = 10

model = Sequential()

model.add(Conv2D(32, kernel\_size=(5, 5),activation='relu',input\_shape=input\_shape))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Flatten())

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.3))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical\_crossentropy,optimizer=keras.optimizers.Adadelta(),metrics=['accuracy'])

hist = model.fit(x\_train, y\_train,batch\_size=batch\_size,epochs=epochs,verbose=1,validation\_data=(x\_test, y\_test))

print("The model has successfully trained")

score = model.evaluate(x\_test, y\_test, verbose=0)

print('Test loss:', score[0])

print('Test accuracy:', score[1])

model.save('mnist.h5')

print("Saving the model as mnist.h5")

Code for the final GUI predictor

from keras.models import load\_model

from tkinter import \*

import tkinter as tk

import win32gui

from PIL import ImageGrab, ImageOps

import numpy as np

model = load\_model('mnist.h5')

def predict\_digit(img):

#resize image to 28x28 pixels

img = img.resize((28,28))

#convert rgb to grayscale

img = img.convert('L')

img = ImageOps.invert(img)

img = np.array(img)

#reshaping to support our model input and normalizing

img = img.reshape(1,28,28,1)

img = img/255.0

#predicting the class

res = model.predict([img])[0]

return np.argmax(res), max(res)

class App(tk.Tk):

def \_\_init\_\_(self):

tk.Tk.\_\_init\_\_(self)

self.x = self.y = 0

# Creating elements

self.canvas = tk.Canvas(self, width=300, height=300, bg = "white", cursor="cross")

self.label = tk.Label(self, text="Draw..", font=("Helvetica", 48))

self.classify\_btn = tk.Button(self, text = "Recognise", command = self.classify\_handwriting)

self.button\_clear = tk.Button(self, text = "Clear", command = self.clear\_all)

# Grid structure

self.canvas.grid(row=0, column=0, pady=2, sticky=W, )

self.label.grid(row=0, column=1,pady=2, padx=2)

self.classify\_btn.grid(row=1, column=1, pady=2, padx=2)

self.button\_clear.grid(row=1, column=0, pady=2)

#self.canvas.bind("<Motion>", self.start\_pos)

self.canvas.bind("<B1-Motion>", self.draw\_lines)

def clear\_all(self):

self.canvas.delete("all")

def classify\_handwriting(self):

HWND = self.canvas.winfo\_id() # get the handle of the canvas

rect = win32gui.GetWindowRect(HWND) # get the coordinate of the canvas

a,b,c,d = rect

rect=(a+4,b+4,c+100,d+100)

im = ImageGrab.grab(rect)

digit, acc = predict\_digit(im)

self.label.configure(text= str(digit)+', '+ str(int(acc\*100))+'%')

def draw\_lines(self, event):

self.x = event.x

self.y = event.y

r=8

self.canvas.create\_oval(self.x-r, self.y-r, self.x + r, self.y + r, fill='black')

app = App()

mainloop()

**CHAPTER-6**

SCREENSHOTS

6.1 TERMINAL COMMANDS

Command to Train the model.

Text

Description automatically generated

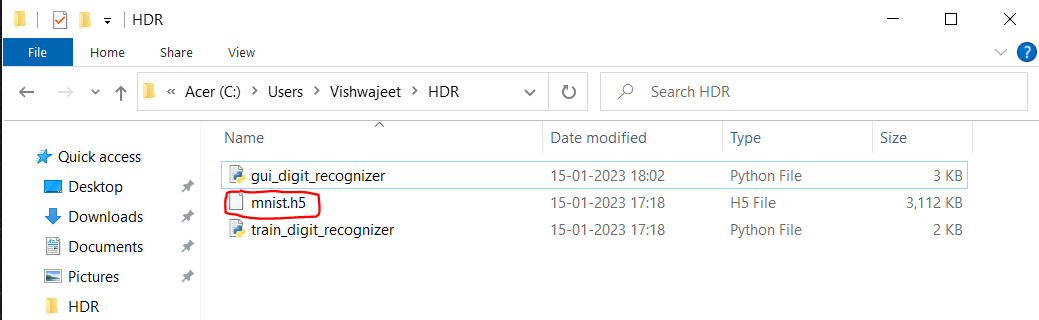
Command to run the GUI.

Text

Description automatically generated

6.2 Output

The Training Model Outputs ‘mnist.h5’ file.



Running the gui\_digit\_recognizer python produces the GUI canvas to draw digits on.

Graphical user interface, application, Word

Description automatically generated

Digit Recognition Output.



Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**CHAPTER-7**

CONCLUSION

Our project HANDWRITTEN DIGIT RECOGNITION deals with identifying the digits. The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings. In this project, CNN (Convolutional Neural Networks) architecture is used to achieve high performance on the digit string recognition problem.

The GUI predicts the digit with accuracy percentage but the software doesn’t have the option to show that the digit as unrecognizable instead it shows the nearest resembling digit. The option to build that was considered but due the complex nature of the code it was considered beyond scope which was recognize the handwritten digits.

7.1 OUTCOME

The software recognizes all digits that is 0-9 successfully. But at times it can recognize a given digit incorrectly. This occurs very rarely and can be tried again by clearing the canvas and entering the digit by clicking the recognize option. The thing to be noted here is the software tries to learn from all the inputs given to it through the CNN and Deep learning so there may occur very few incorrect recognition. Sometimes the incorrect recognition may occur due to handwriting itself as it’s a bit difficult to input digits through the touchpad/mouse as compared to writing physically on a paper.

The software recognizes digits in various handwritings. It also shows the accuracy with which it is making the prediction. The software since is based on AI specifically Deep Learning it constantly learns from the inputs given. Therefore the software becomes more accurate with use.

7.2 FUTURE SCOPE/ENHANCEMENT

The current project is built using python and the concept of deep learning and CNN. The only mode of input is mouse via GUI. And the right now only digits are recognized

The project has scope to enhance on all fronts that is it can be developed to use various other neural networks and tools to not only use the GUI but through various mediums to recognize the digits.

As we are talking about enhancement the software can be developed to recognize not only handwritten digits but texts and stings and symbols.

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 Handwritten Character Recognition System which recognizes human handwritten characters and predicts the output.

The recognition system has huge scope in tasks such as Postal Mail Sorting, Bank Cheque Processing, Form Data Entry, etc.

**CHAPTER-8**

REFERRNCES AND BIBLIOGRAPHY

• Websites referred

[www.stackoverflow.com](http://www.stackoverflow.com)

www.pythonprogramming.net

[www.codecademy.com](http://www.codecademy.com)

[www.tutorialspoint.com](http://www.tutorialspoint.com)

[www.geeksforgeeks.com](http://www.geeksforgeeks.com)

[www.wikipedia.org](http://www.wikipedia.org)

[www.youtube.com](http://www.youtube.com)

* Other references

ACMEGRADE training.