HANDWRITTEN DIGIT RECOGNITION

Major project report submitted in partial fulfillment of the

requirement for the degree of Bachelor of Technology

in

Computer Science & Engineering

By

Vaibhav Jariyal (191373)

UNDER THE SUPERVISION OF

Dr Emjee Puthooran

Dr Pankaj Dhiman

Department of Computer Science and Engineering & Information Technology ,Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

ertify that the
which I submitte
Technology at
fulfilment of the
mputer Science
n work perform
er the guidan
onics and Comm
epartment of Co
tioned pro

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …….

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ __**Department:** _____ **Enrolment No** _____

Contact No. _____**E-mail.** _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Acknowledgement

All compliments and praise are due to God who empowered me with strength and sense of devotion to successfully accomplish this project work successfully.

I am really grateful and wish my profound indebtedness to Deep Knowledge & keen interest of my supervisor in the field of “Artificial Intelligence” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Emjee Puthooran, Department of ECE and Dr. Pankaj Dhiman for his kind help to finish my project. I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking. Finally, I must acknowledge with due respect the constant support and patients of my parents.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Sr. No.	Abbreviation	Full Form
1.	SVM	Support Vector Machine
2.	CNN	Convolutional Neural Network
3.	AI	Artificial Intelligence
4.	MNIST	Modified National Institute Of Standards and Technology

LIST OF FIGURES

ABSTRACT

Due to its widespread use, handwriting recognition has drawn a lot of interest in the domains of machine learning and pattern identification. The application domain for optical character recognition (OCR) and handwritten character recognition (HCR) is specific.

For character recognition in a system for handwriting recognition, several approaches have been suggested. Despite this, a substantial number of studies and papers describe the procedures for transforming the text of a paper document into a machine-readable format.

Character recognition technology may be crucial in the near future in order to process and digitize existing paper documents in order to establish a paperless environment.
In this project we will try to identify the digit written irrespective of the orientation,style of handwriting of the person.

It happens because of the MNIST data.It contains 60000 images(training) and 10000 images(testing).

Chapter-1: INTRODUCTION

1.1 Introduction

Artificial intelligence and computer technology both heavily rely on machine learning and deep learning. The developers are delving into machine learning and deep learning methods to make machines more intelligent.

Humans acquire new skills by repeatedly rehearsing and performing previously learned tasks.

So that it learns how to carry out the chores by heart. When this happens, his brain's neurons automatically fire, enabling them to carry out the learnt task swiftly.

For different kinds of issues, it employs various neural network topologies. The capability of computers in detecting human handwritten digits is known as handwritten digit recognition.

As handwritten numerals are imperfect and can be generated with a variety of tastes, it is a difficult task for the machine.

The majority of the time in daily life, handwritten digit communication is used as a type of dialogue and a way to record information that will be communicated with certain people.

Because different communities may employ different handwriting styles while still controlling to draw the same patterns of characters in their recognised script, one of the difficulties in recognising handwritten characters entirely resides in the variance and distortion of the handwritten character set.

Because there might not always be precise, straight lines, handwritten digit datasets can be ambiguous. The basic objective of feature extraction in digit recognition is to eliminate redundant information from the input and create a word image that is more effectively represented by a set of numerical properties.

Convolutional Neural Network(CNN):

Neural networks are a subset of machine learning and are at the base of deep learning algorithms, as was stated in the Neural Networks Learn Hub page.

They are made up of node levels, each of which includes an input sequence, one or more hidden layers, and an output layer.

Each node has a threshold and weight which are associated to one another.

Any node whose output exceeds the defined threshold value is activated and begins providing data to the network's topmost layer. Eventually, no data is transmitted to the network's next phase.

CNN has three layers-
Convolutional Layer
Pooling Layer
Fully Connected(FC) Layer

The core aspect of a CNN is the convolutional layer, which is also where the most of processing takes place. It needs raw data, a filter, and a feature space, among other things.

Suppose that now the input will be a colour image that is composed of a 3D pixel matrix. As a result, the input will now have three dimensions—height, width, and depth—that are analogous to RGB in an image.

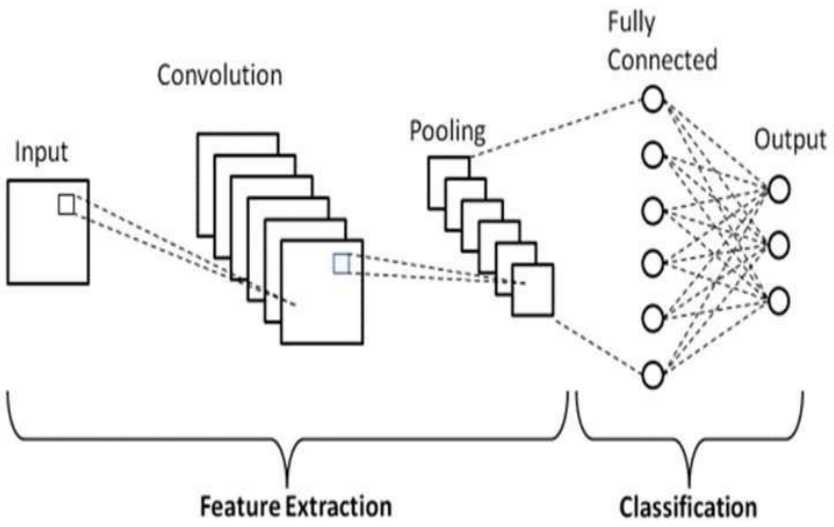
Additionally, we have a specific feature, also referred to as a kernel or filter, which will move through the image's receptive fields and decide if the feature is there. Convolution describes this process.

The pooling operation sweeps a filter across the entire input similarly to the convolutional layer, with the distinction that this filter lacking weights. Instead, the kernel generates the output array by applying an aggregation function to the values in the receptive field. The two main forms of pooling are

Max Pooling-Selects the pixel with max value to send to the output array.

Average Pooling-It calculates the avg value in the field to send to the output array.

Fully Connected Layer-
Based on the parameters that were obtained from the previous layers and their respective filters, this layer performs the classification operation. FC layers often utilise an activation function called softmax to categorize data suitably, producing a probability that ranges from zero to 1. Convolutional and pooling layers generally use ReLu functions.



2

Figure 1:Basic Architecture of CNN model

CNNs are neural networks which are fully connected & feed forward. CNNs are very good at cutting down on the number of variables without sacrificing model quality.

Images have a high degree of dimension since each pixel is treated as a feature, which is ideal for CNNs' abilities as described above. Additionally, CNNs were built by considering images while also setting standards in text processing. CNNs are trained to recognize object edges in any image.

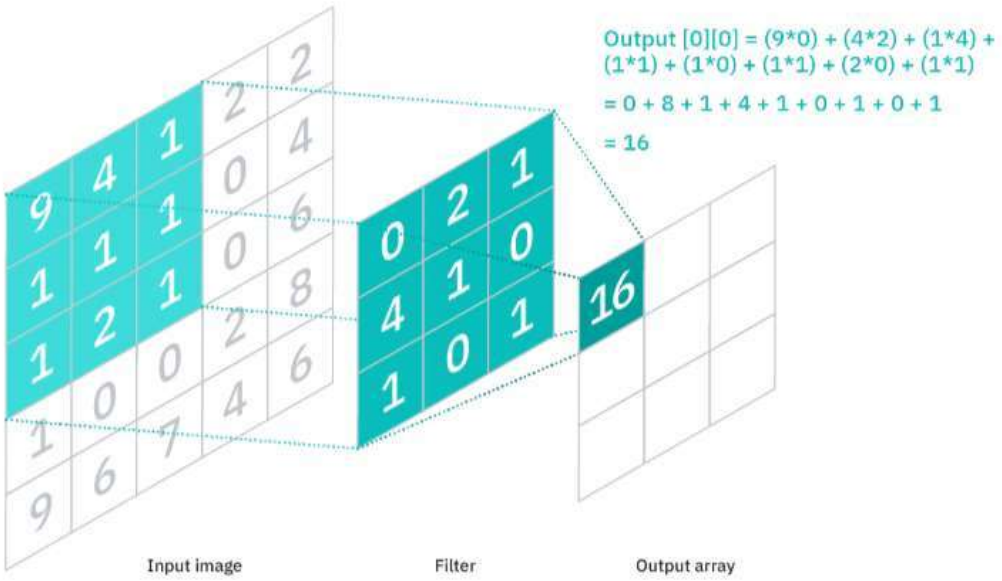


Figure 2: Convolutional Layer

1.2 Problem Statement

Handwriting number identification is a difficult topic that has been the subject of extensive research, particularly in recent years.

Numerous areas of our study deal with numbers, such as reading numbers on bank checks or license plates, making digit recognition an important topic. A method for handling such an application could be a system for identifying solitary digits.

In other words, to enable the computer to comprehend Arabic numbers entered manually by users and display them in accordance with computer-generated views.

Many methods, including minimal distance, decision trees, and statistics, have been developed by scientists and engineers with an interest in pattern recognition and image processing to address the issue of handwriting number recognition.

The main idea behind the project is to implement the machine learning/deep learning model to train the data,extract the features and classify the digit.

1.3 Objectives

The objective was to develop an appropriate method which might generate a handwritten character just by taking a picture of that letter.

If one inquires about image processing, this cannot be solved because there may be a great deal of noise in the captured image that is difficult for an individual to regulate.

The major issue is that when a human types a handwritten character, in this case a digit, he has no clue whether he needs to draw it using the existing pixels or just by copying an existing standard image. While a machine can do it, a human cannot. Therefore, one cannot detect that by matching just the pixels.

To create a model which is less computation intensive.

1.4 Methodology

5.1.1 - Data Collection

Using pre-collected data, such as datasets from Kaggle, UCI, etc., still fits into this phase. The quantity and quality of your data determine how accurate our model is. The outcome of this step is typically a representation of data (Guo simplifies by specifying a table).

5.1.2 - Data Preparation

• Gather data and get it ready for training

• Visualize data to identify meaningful associations between variables, class imbalances (bias alert!), or do other exploratory analysis. Randomize data to eliminate the impact of the specific sequence in which we acquired and/or otherwise processed our data.

5.1.3 - Select a Model Different algorithms are appropriate for various jobs; select the best algorithm.

5.1.4 - Train the Model In in order to answer a question or make a prediction accurately, training involves learning values for m (or W) and b (where x is the input and y is the output). Training occurs during each iteration of the procedure.

5.1.5 - Evaluate the Model

• Tests the model against never-before-seen data

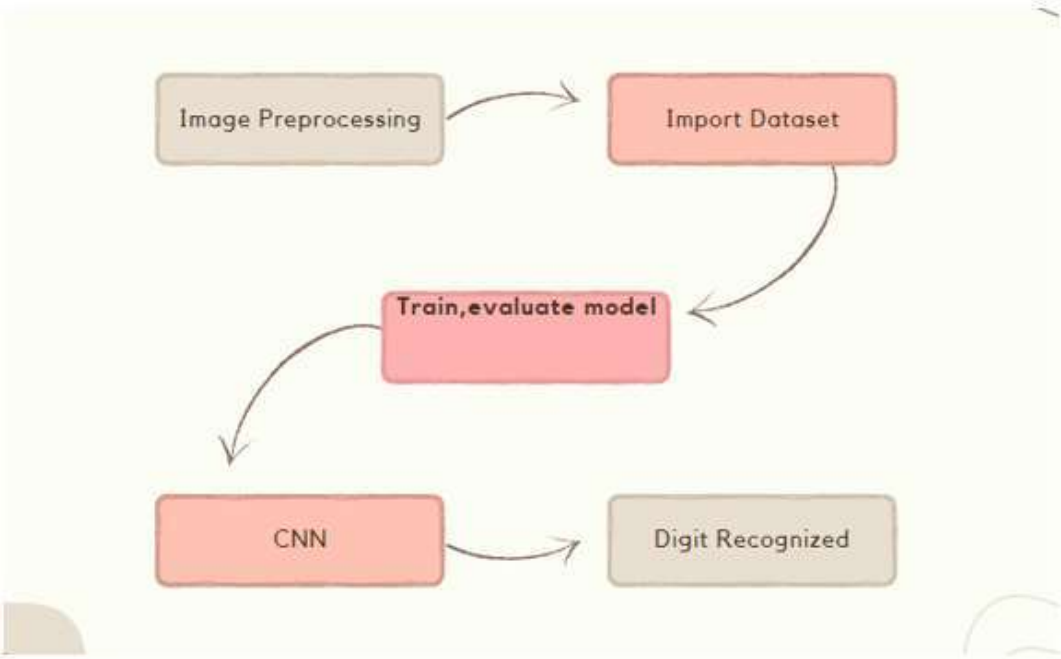
• Uses some metric or set of metrics to "evaluate" the model's performance objectively

• Even though the model is being tuned, this unobserved data is intended to be a small representation of model performance in the real world (as opposed to test data, which does not).

•Good train/evaluation split Depending on the domain, data accessibility, dataset specifics, etc., 80/20, 70/30, or something similar.

5.1.6 - Parameter Tuning
• This stage deals with the "artform" of hyperparameter tweaking, which is not the same as science.

5.1.7 - Make Predictions
• Additional (test set) data that have been withheld from the model up to this phase (and for which class labels are known) are utilised to test the model; this provides a more accurate representation of how the model will behave in the actual world.



1.5 Organization

The project report has been organized into five different chapters which are as below:

Chapter 1: Introduction of the project,the problem statement on which the project is based.The algorithms used are explained a bit .

Chapter 2:In this chapter, the background knowledge behind and the project work have been made more transparent by the use of related information in standard Journals,Books,Internet websites etc.

Detail about how handwritten digit recognition has been solved by convolutional neural networks.

Chapter 3:In this chapter our main aim was to explain the step by step method to build for the research done in this report.

Our first step is used to preprocessing and training the MNIST dataset.

It mainly outlines the neural network used and the features used for building the model.

Lastly,it includes ways for checking accuracy to predict the accuracy.

Chapter 4:To give us a clear picture of the precision of the model created, the report includes the experiments carried out and the associated findings collected at different stages of the project in this chapter. When an experiment is finished, the model builds on every step to produce the network that best matches the data the best.

In this chapter, we evaluate the accuracy of the various audio input types before evaluating the network outputs that have been incorrectly labelled and deciding if the input was correctly classified.

Chapter 5:This chapter has a summary and conclusion of the report.It also outlines areas for further research/future work that can be added to it.

Any new innovation,ideas are also shared.

Chapter 2: LITERATURE SURVEY

Although the terms "machine learning," "deep learning," and "artificial intelligence" (AI) are sometimes used alternately and together once, machine learning and deep learning are only various branches of the much larger science of AI. Machine learning is a method for automatically analyzing data and creating a model from it.

To be more specific and explicit, machine learning refers to the creation of an algorithm that allows a computer to comprehend human language and deliver results based on input from the data.
As opposed to a professional programmer defining particular criteria and instructions to construct the model, the system may learn from information that is given to it and can adapt to the data changes. We may say that within the past few years, machine learning has gained a lot of popularity.

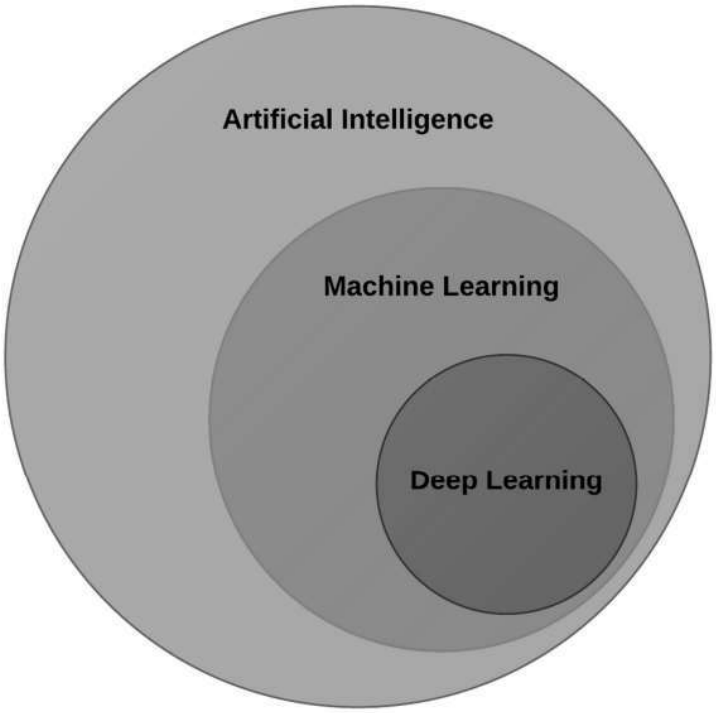


Figure 3: Artificial intelligence vs Machine Learning vs Deep Learning

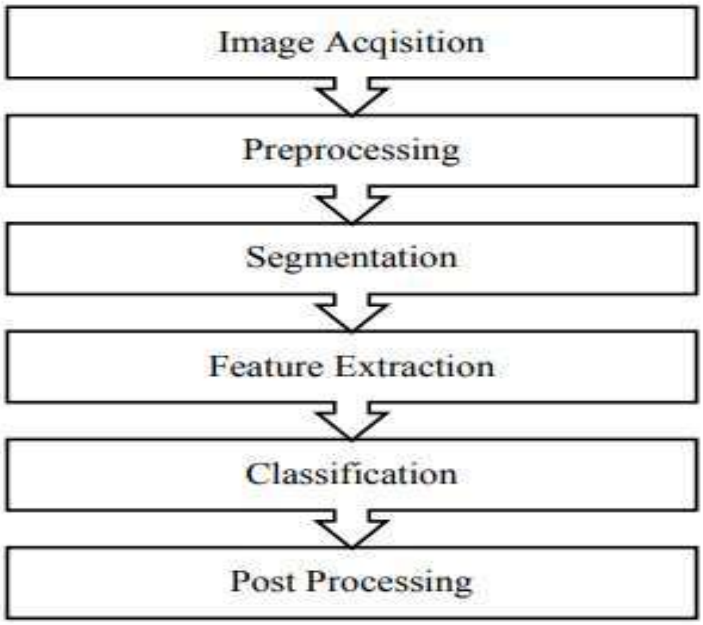
Various kinds of machine learning algorithms are better suited for various applications than others, due to the type of application and accessible datasets.

There are typically four main learning algorithms in machine learning. supervised learning, unsupervised, semi-supervised learning, and reinforcement learning are the main learning algorithms.

A fully labelled dataset is used in supervised learning to build a mathematical model that connects the dataset's inputs. With a fully developed model, supervised learning can be used to foretell the outcomes for fresh input data points.

We have used MNIST dataset by supervised learning to get useful features from the unlabelled datasets.

Handwriting recognition method is a system that enables a computer to identify characters and other symbols which are handwritten in human handwriting.
Online and offline handwriting recognition are the two categories used to describe handwriting recognition.



Block Diagram Of Image Recognition

As part of their study for the MNIST, Mohd Razif Shamsuddin and his coworkers investigated machine learning models.

Through data preprocessing, they were able to generate binary (black and white) and grayscale versions of the MNIST dataset.

They also used these 2 distinct datasets to conduct out the model training.

In the grayscale dataset;the accuracy rate for the CNN model was 99.4%, random forest model was 94%, and extremely random trees model was 94%. The accuracy rates for the CNN model, Random Forest model, and Extremely Randomized Trees model in the binary data set were 90.1%, 91%, and 92%, respectively.

These tests demonstrated the impact and necessity of choosing the appropriate data preparation step and methods for machine learning on success rate.

Convolutional neural networks and support vector algorithms were combined by Fred M. Agarap to construct a machine learning architecture for image categorization issues.

In addition to the model where he introduced a support vector machine using the ReLu function as the activation function in the CNN structure, he also developed another system where the softmax function was used as the activation function of the output nodes.

The "CNNSoftmax" and "CNN-SVM" models were trained using the MNIST and Fashion-MNIST datasets, and the findings were compared. With the "CNN-Softmax" structure, it achieved a success rate of 99.23% in the MNIST dataset and a success rate of 91.86% in the Fashion-MNIST dataset; with the "CNNSVM" structure, it achieved a success rate of 99.04% in the MNIST dataset and a success rate of 90.72% in the Fashion-MNIST dataset.

Rabia Karakaya tested support vector machine, decision tree, random forest, artificial neural network, K-nearest neighbour method, and K-mean algorithm using the MNIST dataset in her work on handwriting classification with machine learning algorithms. She used the Scikit Learn library and tools to perform her assignment.

She used the polynomial kernel function in the support vector machine model to get 90% accuracy rate in the test results of the models trained using the MNIST dataset using all 60000 data.

Furthermore, it achieved 97% accuracy in the random forest model, 97% accuracy in the artificial neural network model, and 87% accuracy inside the decision tree model. It obtained 96% accuracy in the K-Nearest Neighbor algorithm model in 865.932 seconds and 98% accuracy in the K-Mean method model.

Sandhya Arora applied the intersection, shadow, chain code, and straight line fitting feature extraction procedures, which total four.

For character images, shadow features are estimated globally, whereas intersection features, chain code histogram features, and line fitting features are computed by dividing the character image.

Devanagari characters had an overall identification percentage of 92.80% in experiments using a dataset of 4900 samples.

T. Som has discussed about fuzzy membership function-based HCR technique. 20 x 10 pixels are the average for character picture sizes. Ten images of each character are merged to get the average image.

Character projection in both the vertical and horizontal directions is used to determine the bonding box round the character. After already being reduced to a bounding box, the image is then scaled to 10 X 10 pixels.

The process is then finished, so each thinned image is put into a 100 by 100 canvas one by one. Characters are classified based on the test picture's similarity score to the fusion image.

Chapter 3: SYSTEM DEVELOPMENT

System Requirement:
Pc with minimum requirements. The tools and hardware/software requirements are also needed.

Hardware Requirements:

RAM: Min 4 GB.
Processor: Intel(R) i3 or greater with 2.00 Ghz.
Internet: Yes.(Broadband/WiFi)
Webcam: Yes

Tools and Technologies:

Pandas: It is a Python library that offers a quick, adaptable, and expressive data structure to make manipulating "relational" or "labelled" data simple and intuitive. It seeks to serve as a basic, high-level building block for Python's use in actual, real-world data analysis.

Numpy: It is a general-purpose package for array processing. high-performance multidimensional array objects and tools for working with those arrays are made available. Python's fundamental scientific computing package The programme is open source.

Matplotlib: It is a comprehensive package for Python that enables the creation of static, animated, and interactive visualizations. Matplotlib makes either difficult and basic tasks possible.

OpenCV:The Opencv (Open Source Computer-Vision Library) is a very up-to-date Python DIL library for image processing. This module makes it simple to bring one image file and its pixel values to the surface.

A common framework and module for computer vision technologies are provided by this library. The fact that this tool is completely free and is flexible enough to evolve according to programmer input is its most notable factor.

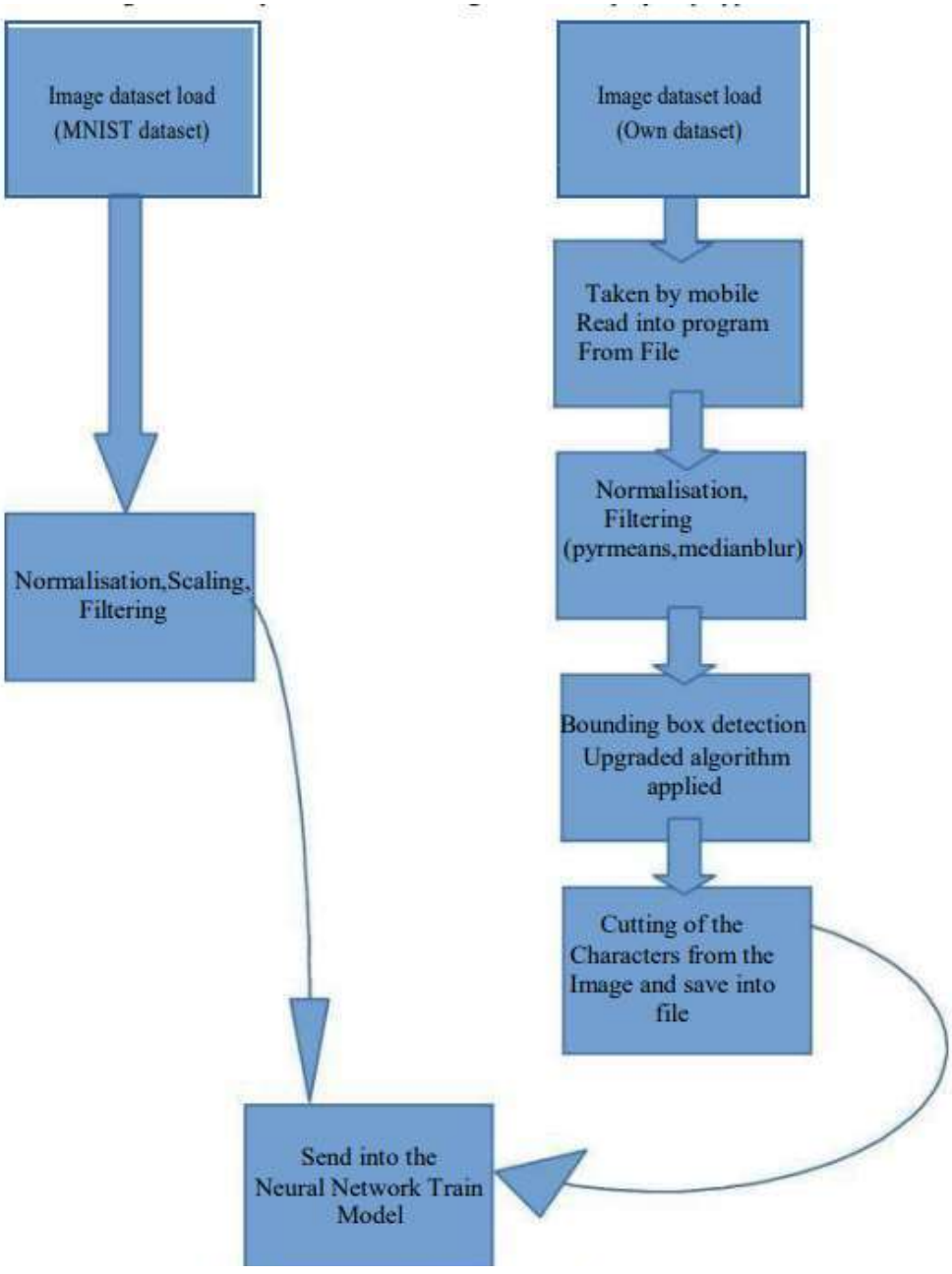


Image dataset load
(MNIST dataset)
Image dataset load
(Own dataset)
Taken by mobile
Read into program
From File
Normalisation,
Filtering
(pyrmeans,medianblur)
Bounding box detection
Upgraded algorithm
applied
Cutting of the
Characters from the
Image and save into
file
Normalisation,Scaling,
Filtering
Send into the
Neural Network Train
Model

IMAGE PROCESSING

Image processing is a process for applying various procedures to an image in order to improve it or extract certain useful information from it. It is a form of signal processing where the input is an image and the output could be another image, or characteristics or during those to the input image.

It is a form of signal processing where the input is an image and the output could be another image, or characteristics or during those to the input image.

Image processing is one of technologies that is presently expanding quickly. It is a key topic of study in both the engineering and computer science fields.

Basically, image processing involves the following three steps:

1) Uploading the picture through image-acquisition software

2) Examining and modifying the picture

3) A report or modified image that was the outcome of the output from the image analysis.

The influence of image processing will be discussed in two parts since this project is currently in two stages.

1) Utilising training data for image processing.

2) Image processing for data testing.

Image Processing in training data

The neural network model was trained using 2 different datasets in the training dataset.

I Dataset from the MNIST (Modified National Institute of Standards and Technology database)

ii) Self-created dataset.

This two dataset nearly loads the training model with 60500 elements and correctly repeatedly trains the model nearly five times.

A big database of handwritten numbers can be found in the MNIST database (Modified National Institute of Standards and Technology database). The database is frequently used for machine learning training and testing.

It was created by "re-mixing" samples from the initial datasets from NIST. Since the testing dataset was obtained from American high school students, and the training dataset was collected from American Census Bureau employees, the designers of NIST concluded that it was improper for machine learning research.

In addition, the MNIST black and white images were anti-aliased, which added grayscale levels, and normalized to fit within a 28x28 pixel bounding box.

This dataset, which was separately created using handwritten characters, almost adds 500 entries during training. This dataset is more effective than MNIST for this project and is used at the end of the training model because of model may face common concerns after testing. This dataset contains unique qualities.

1. The image's value is not fixed.

2. A mobile picture of a piece of paper is obtained

3. Saving it to the particular folder after the process of cutting and resizing.

Before transmitting the dataset to the neural model, a lot of image analysis effort was required. This picture or data has a number of unlabelled elements that need to be under control.

1. Multiple words are chosen randomly; you should cut them.

2. The real-time mobile image may have various tones.

3. A noisy image will appear; we must eliminate the noise.

4. Based on the quality loss, the image needs to be resized.

Mobile picture of characters written on a paper

Characters from a paper are represented in a mobile image.
This phase involves writing some characters on black and white paper since they are easier to read, unless you can write on a page with margins as well.

It will be easier to write with a sketch pen, marker, or gel pen because they produce continuous characters.

As ballpoint pens and good pencils can create gaps in a character's architecture, they cannot produce good writing, which is necessary if the character is to be scanned and detected by machines for optical character recognition (OCR).

Take a picture after you've wrote something on paper.
To finish this task, simply take a nice image of the characters; no scanning is required.

Normalisation

Real-world data endures in a variety of formats, and if a plot of the black-white variance is made, it will also produce results with the median filled. For this reason, normalisation is necessary in order to have an accurate cleaning procedure.

Here, the image has been normalized using MINMAX Normalisation method. Here, the picture's colour range has been eliminated, and a range—such as 0 and 1—has been provided for each and every pixel.

$$Vi` = (vi-minA)*(newmaxA-newminA)/(maxA-minA)$$

Eq 3 MinMax Normalization

where vi' is the new, normalised pixel number (with a range of min to max), vi is the old pixel number, and the newmin to newmax are the new pixel's valid values.
Several respond to different methods exist, however minmax was selected because because

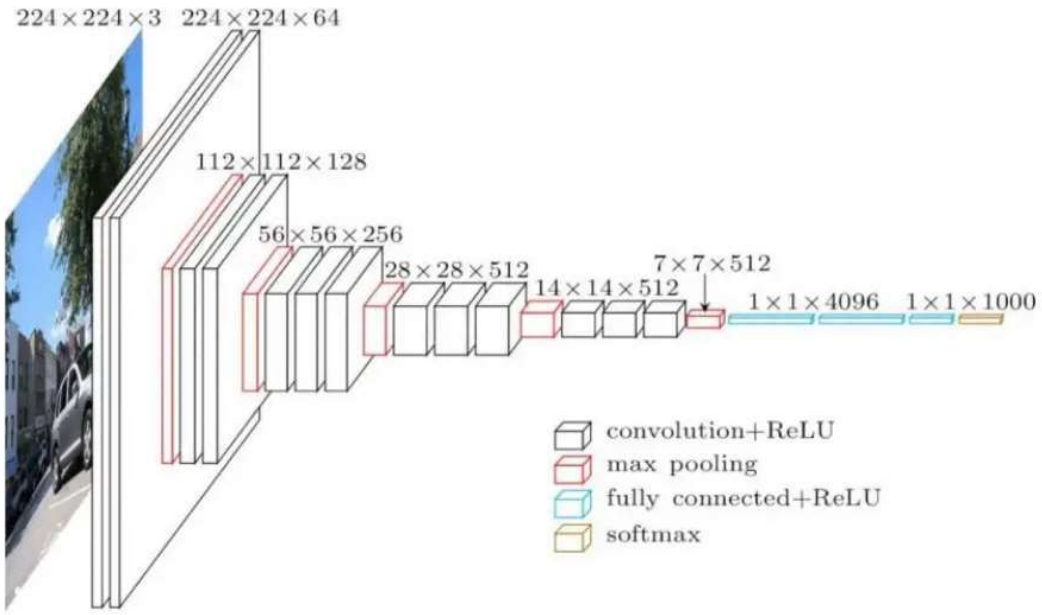
has been shown to work well for cleaning handwritten characters. It more effectively turns noisy (grey) pixels into white and turns black pixels to black.

Filtering

An image can be improved or softened using filters. Cells with exceptionally high or relatively low frequency are removed throughout this process.
By only preserving the lower frequencies, low pass filters eliminate the high frequencies.

The low pass filter works by removing noise from the image and smoothing it down. By only maintaining the higher frequencies, high pass filters eliminate the lower frequencies.

High pass filters are used to sharpen images with a solid black background or no background at all. We have used it first as the lower filter for noise removal.



Contour detection

After the image has undergone noise cancellation, normalisation, and filtering, the contour detection approach is fully applied.

There are some serious issues with this approach that only apply to handwritten characters. This study discusses various contour detection techniques.

To address the actual contour detection problems, one must deal with several components one at a time.Contours are a few curves or lines that join spots along the boundary that are the same hue or intensity.

This tool, which takes the subjected forms from an image, is incredibly helpful for object recognition and detection.

Binary images or canny images must be taken where the subject forms and background are drastically different intense for greater contour accuracy.

Hierarchy

In CV2, there are three arguments. The source image is the first parameter of findContours(), following by the contour retrieval mode and the contour approximation method.

The contour approximation method and contour retrieval mode are covered by the contour hierarchy. Various types of retrieval and approximation techniques exist.

findontours() finds all contours in an image, although in handwritten character sets some contours may sometimes or regularly remain within or be connected to another contour.

Parent, child, and sibling are the names of the connected inner and outer contours, respectively.

The hierarchy of contours describes this type of interaction.

Many opencv methods or functions can uniquely define the hierarchy of contours.

cv2. RETR LIST (): Children and parents are on the same hierarchy level.

cv2.RETR_EXTERNAL():No other family members are searched, only the parent.

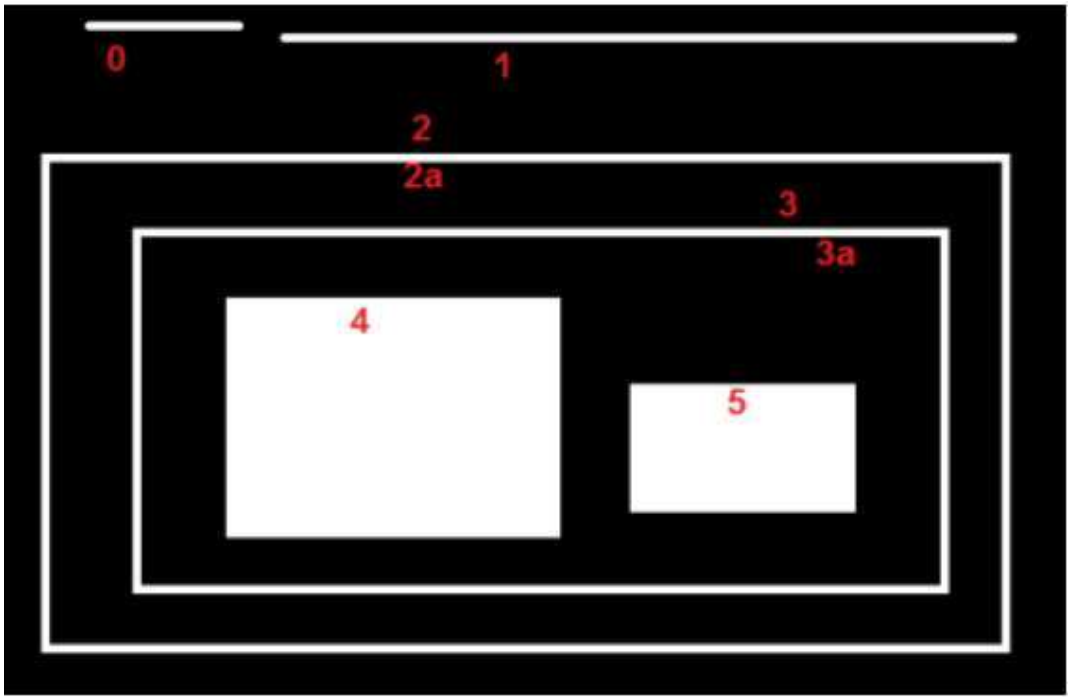
cv2. RETR CCOMP ():Accepts only members of the second hierarchy level; no parents.

cv2.RETR_TREE():Removes the entire family, including the grandfather, son and grandchildren.

The contour now depends entirely on the pixel variance, i.e. as soon as a deviation is detected, the contour is applied to the pixel.

Therefore, the extraction of handwritten digits is ineffective because it analyzes the contours of all siblings in an image before applying to the whole family, although only the top member of the family is needed

This cannot be taken into account at all, because while there may be a child in one area, there may be a parent in another.



Enhancement of the bounding box

The problems with the bounding box notion are to be improved.

An object in a given image or video can be detected using a frame called a bounding box, which can be a rectangle, an oval, or a circle.

In handwritten character recognition, rectangular boxes are mainly used to describe the target of the position box. Bounding Box returns the output in a four-valued list

Matrix insertion

Therefore, the inner bounding boxes must be deleted, leaving only the outer bounding boxes that define the boundaries of the characters. A matrix or 2-D list can contain all values of the bounding boxes.

Delete duplicates or bounded box

It may be necessary to delete several bounded boxes, since the pixels within a character may change even if they are bounded by another box. Only the outer frame is taken into account.

The upper left point [x, y], the width and the height of each bounding box have been recorded in our matrix for the image.

Take each pair of lists from the matrix and determine whether or not the bounding box contains any bounding boxes.

Verify whether the x and y axes of one bounding box point within the other.
If inside, verify that the bounding box is completely inside by checking the width+x and height+y.

If so, remove it or replace another in the main matrix.
Take only distinct values, after all. The matrix is then sorted as those to fit it into the various positions, contours must be obtained.

For instance, the contour that was created for the number 8 is located in a matrix, thus we must fit that shape solely onto the number 8 and similarly for the remaining letters.

Inside contour will be present if one bounding box starts inside the another one but its height or width exceeds the smaller one, which is typically outward.

Only within contours that are completely bounded will be removed or ignored.

Putting contours

After finding the appropriate contours in the image, the ideal contour bounding box recognition will find as many bounding boxes as there are characters in image.

On obtaining the bounding boxes, it is necessary to sort them by eliminating duplicates before positioning them on the image setting contours.

```
x,y,w,h =matrix[i] imag = cv2.rectangle(image,(x,y),(x+w,y+h),(0,0,255),2)
```

The images on which the boxes will be drawn is the rectangle function's very first attribute. The upper left corners in the contour in the 2-D image constitute the second attribute (x, y).

The right side bound and the lower bound of the image are indicated by the third property. The colour of the contour box (r,g,b) will be the fourth feature, and the thickness of the bounding box will be the fifth.

Neural Network

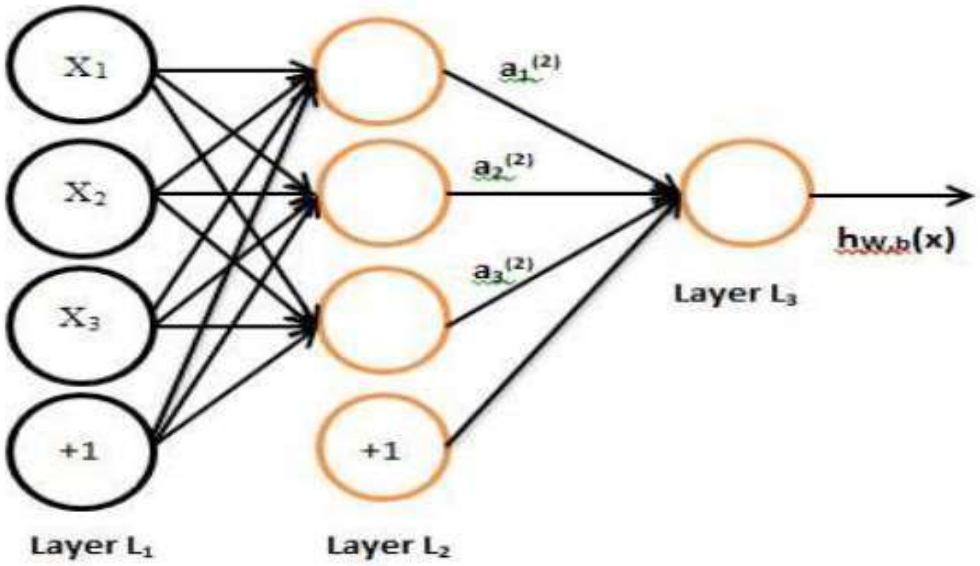
The neural network is a machine learning concept. It forms a model that simulates a brain unit, and then, like a toddler, it teaches that model with countless datasets. For my project, those datasets are the digits. A neural network has two components.

1) Training
2) Testing

Every neural network can be divided in three different parts

1) The Input layer
2) The Hidden layer
3) The Output layer

Many of our basic "neurons" are connected to create a neural network, where output of one is input of second.



2) Hidden Layer:There are layers of nodes between both the output and input levels. One or more of these layers might exist. There will be more actions, more weights, and less change because of the gap in this layer as more hidden layers are created.

The most major element of the neural network is this layer. The weights are changed by return input in many ways.

3) Output Layer:A layer of nodes that creates the output variables, after which it can repeat based on the targets.

Testing Part

The testing part incorporates some of the latest machine learning part real-part updates. The techniques for image processing, involving webcam permission, have been discussed in the past.

This is the reasons it hasn't been covered here. Hence, this portion includes a ransom upgrade in the method, that is really minor in state but not at all simple to build.

After performing the entire train process and adding the webcam image into the predictive model neural network, let's look more closely to the model.

Set the input pixel values into the numpy array value so that the argmax function can determine the output figure after considering a range. The neural network's feed forward step will estimate the range.

No neural network can achieve 100% accuracy, but if it problems to recognise the majority of the test figures, it probably needs more practise.

The model has been trained for more than 23 times to improve it, and naturally not with the same dataset every time, the outcomes are discussed quickly. The neural model was trained using a number of datasets, including from the web, images and handwritten papers.

1) And also from the model's unique creations and the MNIST dataset.

2) Setup the numpy system to accept kernel as well as data as input.

3) Define the input layer, the hidden layers, the output layers, and activation functions used in each layer as you build the neural network model.

4) While training the dataset, compare the objective and the output predicted value each time, and set the right number of epochs to match to the errors that come up.

5) Following the setup of the weight value by the neural network on its set and the setting of the dataset to the unbiased, the model is checked using checksum value removal and biased value removal.

6) Using Google Colab Display and Python Javascript, connect the webcam and take a picture.

7) Validate the probability parameter for the produced figure post scaling,removing noise, and normalisation.

8) Received the greatest prediction value?
match the figure to the neurally detected information.

Chapter-4: PERFORMANCE ANALYSIS

4.1 Requirements

4.1.1 Language Used

Python is the language used in this project. Deep learning and machine learning are the fields in which this phrase is employed. Machine learning (ML), which is a part of artificial intelligence, includes deep learning (AI). Making machines and computers think and reason similarly to human brains is the aim. In essence, it imitates how a person might learn new information.

At the Dutch National Institute of Computer Science and Maths, Guido van Rossum created Python about in 1990s.

Python is a programming language used in building software and websites, to automate task, and obtaining results from data.Due to its adaptability and easy use for beginners , it is among the most well-known and frequently used programming languages.

Python is commonly used for developing websites and applications, automating repetitive tasks, and presenting data. Many non-programmers, including accountants and scientists, have adopted Python because it is very simple to learn and can be used for a wide range of daily tasks.

A financial institution can now manipulate and analyze data as well as handle other data-related tasks. Python has become an essential component of data science, with data analysts and other professionals using language to perform complex statistical calculations, generate data visualizations, and create machine learning algorithms.

Python provides packages and modules that encourage the modularity and reuse of code in programmes. On all major systems, the Python interpreter and comprehensive standard library are freely distributable in source or binary form.

4.1.2 Libraries Used

Pandas:It is a Python package that provides a fast, flexible and expressive data structure for easy and intuitive manipulation of "relational" or "labeled" data. It aims to be a fundamental, high-level building block for practical, real-world data analysis in Python.

Numpy:It is a general purpose array processing package. Provides high-performance multidimensional array objects and tools for manipulating those arrays. Basic package for Python scientific computing. It is open source software.

Matplotlib:For 2D displays of arrays, Matplotlib is a fantastic Python visualization library. A multi-platform data visualization package called Matplotlib was created to deal with the larger SciPy stack and is based on NumPy arrays. In the year 2002, John Hunter first presented it.

Dataset used here is MNIST which is a standard data for image visualization and contains 60000 training images and 10000 testing images so that the person's writing style,orientation and style doesn't impact the result.

One of visualization's biggest advantages is that it gives us visual access to vast volumes of data in forms that are simple to understand. There are numerous plots in Matplotlib, including line, bar, scatter, histogram, etc.

TensorFlow:A free and open-source software library for artificial intelligence and machine learning is called TensorFlow. Even though it can be applied to many different tasks, deep neural network training and inference are given special attention.

4.1.3 System Requirements

A computer which can perform the required tasks of Machine learning and Deep learning. Jupyter Notebook/Google Colaboratory are needed.

4.1.4 Hardware Requirements

RAM: At least 4 GB.
Processor: Intel(R) core (TM) i3 or more.
2.00 Ghz.
Internet connectivity: Yes.(Broadband or wi fi)
Webcam connectivity: Yes

4.2 Analyzing the Results

We have uploaded the dataset.

```
print(df.columns)
```

```
Index(['label', '1x1', '1x2', '1x3', '1x4', '1x5', '1x6', '1x7', '1x8', '1x9',
       ...
       '28x19', '28x20', '28x21', '28x22', '28x23', '28x24', '28x25', '28x26',
       '28x27', '28x28'],
      dtype='object', length=785)
```

```
df.head(n=10)
```

	label	1x1	1x2	1x3	1x4	1x5	1x6	1x7	1x8	1x9	...	28x19	28x20	28x21	28x22	28x23	28x24	28x25	28x26	28x27	28x28
0	5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	9	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
7	3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
9	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

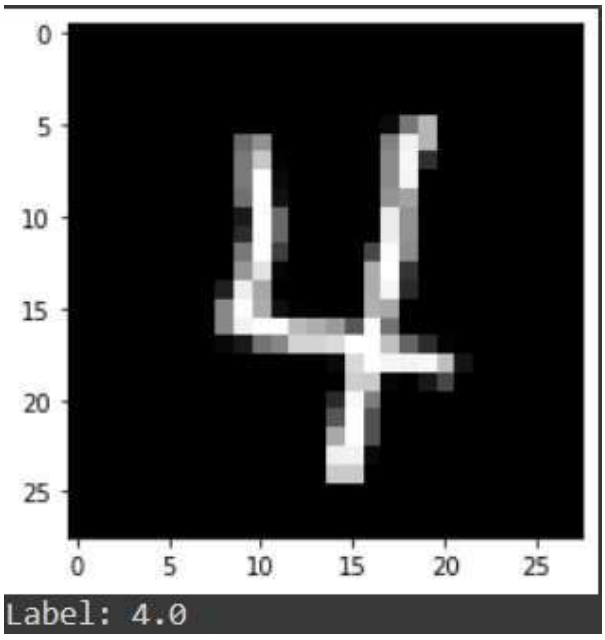
10 rows × 785 columns

```
print(X_train.shape,Y_train.shape)
print(Y_train)
print(X_train)
print(X_test.shape,Y_test.shape)
print(Y_test)
print(X_test)
df = pd.DataFrame(X_test)
df.to_csv('mnist_test.csv')
```

```
print(X_test)
print(Y_test)
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
[4 7 8 8 5 6 6 8 2 2 7 8 0 4 3 4 5 2 4 2 4 8 1 0 1 0 0 1 8 2 9 3 6 6 2 7 9
 8 3 0 1 1 9 2 1 3 5 4 4 5]
```

```
for test in X_test:
    drawImg(test)
    print("Label:", knn(X_train, Y_train, test))
```



```
accuracy=calculate_accuracy(X_test,Y_test,X_train,Y_train,5)

print(accuracy)
```

```
0.98
```

The accuracy of model comes out to be 98%.

Chapter-5: CONCLUSIONS

5.1 Conclusions

The identification of the digits is the focus of our project, Handwritten Digit Recognition.

Building an automatic handwritten digit recognition method is the major goal of this project, which will be used to recognise handwritten digit strings.

This is an OCR project (Optical Character Recognition). This non-fundable project was created with much enthusiasm and covers several external concepts related to statistical modelling and optimization techniques.

5.2 Future Scope

The task of handwritten digit recognition using a classifier is very important and has many applications, including online handwriting recognition on computer tablets, sorting postal mail by zip code, processing bank check amounts, and numeric entries in forms filled out by hand (such as tax forms).

A large field of artificial intelligence and machine learning can be used to improve this project. The world can imagine a software tool that can recognize text in an image and present it to others, like such a business name detector. Or, this undertaking could be expanded to a more thorough knowledge of all character sets in existence.

The entire English alphabet was not chosen for this project since there will be an increasing number of training and testing values that the neural model will be unable to distinguish. Imagine a roadside AI-modeled automobile sensor that only requires users and provide the destination.

These breakthroughs are all application of texture analysis, which is where cutting-edge image processing method, neural network training models, and cutting-edge AI concepts will emerge. Further analysis can be done for these applications. The fact that this project was totally accomplished using free and readily accessible resources and packages may also be a project restriction.

Because not all ML libraries and cutting-edge software are provided for free, the money is crucial.

Aside from those, the majority of graphical platforms, such as Watson Studio or AWS, are used by developers to do some tasks.

These are primarily paid platforms where numerous machine learning initiatives are being undertaken.

5.3 Applications Contributions

Future applications of these algorithms will range from everyday users to high-level authorities. For instance, we can employ these algorithms in hospitals to provide detailed clinical diagnosis, treatments, and patient monitoring. We can also use them in surveillance systems. These applications will range from common users to high-level authorities because of the distinction of the algorithms described above.

REFERENCES

[1] Kai Ding, Zhibin Liu, Lianwen Jin, Xinghua Zhu, “A Comparative study of GABOR feature and gradient feature for handwritten chinese character recognition”, International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182-1186, Beijing, China, 2-4 Nov. 2007.

[2] Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition", International Journal of Engineering Research and Applications, Vol. 2, Issue 1, pp. 659-662, Jan-Feb 2012.

[3] Bhatia Neetu, “Optical Character Recognition Techniques”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.

[4] Liana M. Lorigo and Venu Govindaraju, “Offline Arabic Handwriting Recognition: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28 Issue 5, May 2006.

[5] K. Gaurav and Bhatia P. K., “Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition”, 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

[6] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., “Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

[7] U. Pal, T. Wakabayashi and F. Kimura, “Handwritten numeral recognition of six popular scripts,” Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2007

APPENDICES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from google.colab import files
uploaded = files.upload()

df=pd.read_csv('mnist_train.csv')
print(df.shape)

print(df.columns)

df.head(n=10)

data=df.values
print(data.shape)

X=data[:,1:]
Y=data[:,0]
print(X.shape,Y.shape)

split=int(0.8*X.shape[0])
print(split)

X_train=X[:split,:]
Y_train=Y[:split]
X_test=X[split:,:]
Y_test=Y[split:]

print(X_train.shape,Y_train.shape)
print(Y_train)
print(X_train)
print(X_test.shape,Y_test.shape)
print(Y_test)
print(X_test)
df = pd.DataFrame(X_test)
df.to_csv('mnist_test.csv')
```

"""Visualizing training data"""

def drawImg(sample):

```
  img=sample.reshape((28,28))
    plt.imshow(img,cmap='gray')
    plt.show()

   drawImg(X_train[6])
print(Y_train[6])

def euclidean_distance(x1,x2):
    return np.sqrt(np.sum((x1-x2)**2))

def knn(X,Y,query_point,k=5 ):
    vals=[]
    m=X_train.shape[0]

    for i in range(m):
        d=euclidean_distance(query_point,X[i])
        vals.append((d,Y[i]))

    vals=sorted(vals)
    #print(vals)
    #Nearest/First k points
    vals=vals[:k]

    vals=np.array(vals)
    #print(vals)
    new_vals=np.unique(vals[:,1],return_counts=True)
    index=new_vals[1].argmax()
    pred=new_vals[0][index]

    return pred

X_test=X_test[:50]
Y_test=Y_test[:50]

print(X_test)
print(Y_test)

for test in X_test:
    drawImg(test)
    print("Label:", knn(X_train, Y_train, test))

def calculate_accuracy(X_test, Y_test, X_train, Y_train, k=5):

    predictions = []
```

```
  for test_point in X_test:
      pred_label = knn(X_train, Y_train, test_point)
      predictions.append(pred_label)

    predictions = np.array(predictions)

    accuracy = (predictions == Y_test).sum() / Y_test.shape[0]
    return accuracy

accuracy=calculate_accuracy(X_test,Y_test,X_train,Y_train,5)

print(accuracy)

import os
import PIL
import cv2
import glob
import numpy as np
from tkinter import *
from PIL import Image,ImageDraw,ImageGrab

from keras.models import load_model
model = load_model(r'C:\Users\Dell\Downloads\model.h5')
print("Run APP")

root=Tk()
root.resizable(0,0)
root.title("Handwritten Digit Recognition GUI App")

lastx,lasty=None,None
image_number=0

cv=Canvas(root,width=640,height=480,bg='white')
cv.grid(row=0,column=0,pady=2,sticky=W,columnspan=2)

cv.bind('<Button-1>',activate_event)

btn_save=Button(text="Recognize Digit",command=Recognize_Digit)
btn_save_grid
```