PURBANCHAL UNIVERSITY

Acme Engineering College

Sitapaila, Kathmandu, Nepal

****Project Report

On

**HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTION NEURAL NETWORKS**

**Submitted by:**

Aashish Sharma

Bishesh Rana

Hemraj Parajuli

Sujata Silwal

**Submitted to:**

Supervisor Er. Roshani Ghimire

DEPARTMENT OF COMPUTER & ELECTRONICS& COMMUNICATION

ENGINEERING

February, 2021

**ACKNOWLEDGEMENT**

We would like to express our gratitude towards all the people all the people from different sectors, Professional, and non-professionals who have motivated and helped in our project Firstly, we would like to thank the department of computer and electronics and communication engineering for allowing us to explore our hidden potentials and test our technical skill by assigning us the project. We would like to express our special thanks to Er.Sarad Poudel, head of department (HOD), for his perpetual encouragement, generous help and inspiring guidance. Similarly, we are thankful to our project supervisor Er.Roshani Ghimire for her support and guidance in building this project. Finally, we feel indebted to our teachers, friends and colleagues who helped us in working on this project and stood by us whenever we needed some guidance and help.

**ABSTRACT**

This project involves the design and construction of a neural network with deep learning and optimizes that network to recognize the handwriting of any person.

“**Handwritten Recognition system**” is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch screen, and other devices. The image of the written text may be sensed “**offline**” from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed “**online**” for example by a pen-basedcomputer screen, surface, a generally easier task as there are more clues available.

The main aim of this project is to design a neural network that makes the machine able to understand and recognize numeric characters.

On top of all this, we are going to create a mobile application that allows remote use of this system which acts as a user interface between system and user.

Contents

[1.1 Background and overview 1](#_Toc64236603)

[1.2 Problem statement 3](#_Toc64236604)

[1.3 Project Objectives 4](#_Toc64236605)

[1.3.1 General Objectives 4](#_Toc64236606)

[1.3.2 Specific Objectives 4](#_Toc64236607)

[1.4 Project scope and limitation 4](#_Toc64236608)

[1.5 Project Feature 5](#_Toc64236609)

[2. LITERATURE REVIEW 6](#_Toc64236610)

[3. FEASIBILITY STUDY 9](#_Toc64236611)

[3.1 Technical feasibility 9](#_Toc64236612)

[3.2 Economic feasibility 10](#_Toc64236613)

[3.3 Operational feasibility 10](#_Toc64236614)

[4.METHODOLOGY 12](#_Toc64236615)

[4.1 System development model 12](#_Toc64236616)

[4.2 Type of SDLC 14](#_Toc64236617)

[4.3 Block Diagram and Algorithm 16](#_Toc64236618)

[4.5 Diagram 21](#_Toc64236619)

[4.5.1 Context Diagram 21](#_Toc64236620)

[4.5.2 Level 0 Data Flow Diagram 22](#_Toc64236621)

[4.5.3 Level 1 Data Flow Diagram 23](#_Toc64236622)

[4.5.4 ER Diagram 24](#_Toc64236623)

[4.5.5 Use Case Diagram 25](#_Toc64236624)

[4.5.6 Sequence diagram 27](#_Toc64236625)

[5. PROJECT SPECIFICATION 29](#_Toc64236626)

[5.1 Functional requirements 29](#_Toc64236627)

[5.2 Non-Functional requirements 30](#_Toc64236628)

[5.3 System requirements 30](#_Toc64236629)

[5.3.1 CNN (convolutional neural network) 30](#_Toc64236630)

[5.3.2 TensorFlow 31](#_Toc64236631)

[5.3.3 NumPy 32](#_Toc64236632)

[5.3.4 Keras 33](#_Toc64236633)

[5.3.5 MINIST Dataset 33](#_Toc64236634)

[6. TESTING 35](#_Toc64236635)

[6.1. Software Testing Overview 35](#_Toc64236636)

[7. DISCUSSION AND CONCLUSION 41](#_Toc64236637)

[8. ANNEX A - SYSTEM SNAPSHOTS 41](#_Toc64236638)

[9. REFERENCE 45](#_Toc64236639)

**LIST OF FIGURES**

Figure Page

1. Step of SDLC…………………………………………………..

2. Incremental Model …………………………………………….

3. Block Diagram of system………………………………………

4. Architecture Of Neural Network………………………………

5. Context Diagram……………………………………………….

6. Level 1 DFD……………………………………………………

7. Level 2 DFD……………………………………………………

8. ER Diagram…………………………………………………….

9. Use Case Diagram………………………………………………

10. Annex………………………………………………………….

**ABBREVIATIONS**

CNN :- Convolutional Neural Network

PC :- Personal Computer

MNIST :- Modified National Institute of Standard and Technology database

ANN :- Artificial Neural Network

DFD :- Data Flow Diagram

SVM :- Support Vector Machine

KNN :- K Nearest Neighbor

CRCN :- Coherence Recurrent Convolutional Network

NORB :- NYC Object Recognition Benchmark

CIFAR :- Canadian Institute For Advance Research

# 1. INTRODUCTION

## 1.1 Background and overview

Recognition is identifying and distinguishing things or an individual from the experience and learning. Similarly, Digit Recognition is nothing but recognizing or identifying the digits in any document. Digit recognition framework is simply the working of a machine to prepare itself or interpret the digits. Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheque, papers, pictures, and so forth and in various situations for web based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheque, digits entered in any forms etc.

Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits. Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving, learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents from any document. Deep Learning models can accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large data sets in order to recognize digits from distinctive sources. Handwriting recognition of characters has been around since the 1980s. The task of handwritten digit recognition, using a classifier, has extraordinary significance and use such as – online digit recognition on PC tablets, recognize zip codes on mail, processing bank check amounts, numeric sections in structures filled up by hand (For example: tax forms) and so on. There are diverse challenges faced while attempting to solve this problem. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins. The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MNIST data set of images of handwritten digits [1].

With time the numbers of fields are increasing in which deep learning can be applied. In deep learning, Convolutional Neural Networking (CNN) is being used for visual imagery analyzing. Object detection, face recognition, robotics, video analysis, segmentation, pattern recognition, natural language processing, spam detection, topic categorization, regression analysis, speech recognition, image classification is some of the examples that can be done using Convolutional Neural Networking. The accuracy in these fields including handwritten digits recognition using Deep Convolutional Neural Networks (CNNs) have reached human level perfection. Mammalian visual systems’ biological model is the one by which the architecture of the CNN is inspired. Cells in the cat’s visual cortex are sensitized to a tiny area of the visual field identified which is recognized as the receptive field. It was found by D. H. Hubel et al. in 1062. The non-cognition, the pattern recognition model inspired by the work of D. H. Hubel et al. was the first computer vision. It was introduced by Fukushima in 1980. In 1998, the framework of CNNs is designed by LeCun et al which had seven layers of convolutional neural networks. It was adept in handwritten digits classification direct from pixel values of images. Gradient descent and back propagation algorithm are used for training the model. In handwritten recognition digits, characters are given as input. The model can be recognized by the system. A simple artificial neural network (ANN) has an input layer, an output layer and some hidden layers between the input and output layer. CNN has a very similar architecture as ANN. There are several neurons in each layer in ANN. The weighted sum of all the neurons of a layer becomes the input of a neuron of the next layer adding a biased value. In CNN the layer has three dimensions. Here all the neurons are not fully connected. Instead, every neuron in the layer is connected to the local receptive field. A cost function generates in order to train the network. It compares the output of the network with the desired output. The signal propagates back to the system, again and again, to update the shared weights and biases in all the receptive fields to minimize the value of cost function which increases the network’s performance. The goal of this article is to observe the influence of hidden layers of a CNN for handwritten digits. We have applied a different type of Convolutional Neural Network algorithm on Modified National Institute of Standards and Technology (MNIST) dataset using TensorFlow, a Neural Network library written in python. The main purpose of this paper is to analyze the variation of outcome results for using a different combination of hidden layers of Convolutional Neural Network. Stochastic gradient and backpropagation algorithm are used for training the network and the forward algorithm is used for testing [2].

## 1.2 Problem statement

The handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Also, people write the same digit in many ways: the digit ‘1’ is written as ‘1’, ‘1’, ‘1’ or ‘1’. Similarly, 7 may be written as 7, 7, or 7. This problem is faced more when many people write a single digit with a variety of different handwriting. Lastly, the uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits. Now we introduce the concepts and algorithms of deep learning and machine learning.

## 1.3 Project Objectives

### 1.3.1 General Objectives

* To recognize handwritten digit in real work for autonomous machine processing.
* To be familiarize with neural network.
* To build the convolutional neural network which is capable of correctly classify handwritten digit.
* Promote the use of emerging technology based on Machine learning, Artificial intelligence and Deep learning.

### 1.3.2 Specific Objectives

* To recognize the handwritten digit.

## 1.4 Project scope and limitation

The image recognition is the hottest field of computer science. Nowadays it has broad scope in automated system. So, we are trying to make a system that recognize the handwritten digit by using the convolution neural network. Some of our project scope and limitation are mention below.

* To recognize the digit from the bank check.
* To recognize the digit from the recharge card.
* To identify number plates of vehicle.
* To recognize the digit from any form.
* Recognize zip codes on mail for postal mail sorting.
* Online handwriting recognition on computer tablets.
* Numeric entries in forms filled up by hand (for example: tax forms).

## 1.5 Project Feature

* System will easily recognize the handwritten digit.
* System will easy to implement because of the low-cost features and development.
* Easy to use mobile application, which works both online and offline.
* Easy to use it.
* We use machine learning and deep learning concepts

# 2. LITERATURE REVIEW

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. Although, this difference does not make any issues to people, yet, anyway it is increasingly hard to instruct computers to interpret general handwriting. For the image recognition issue, for example, handwritten classification, it is essential to make out how information is depicted onto images.

Handwritten Recognition from the MNIST data set is well known among scientists as by utilizing different classifiers for various parameters, the error rate has been decreased, for example, from linear classifier (1-layer NN) with 12% to 0.23% by a board of 35 convolution neural systems. The scope of this is to implement a Handwritten Digit Recognition framework and think about the diverse classifiers and different techniques by concentrating on how to accomplish close to human performance. For an undertaking of composing diverse digits (0-9) for various people the general issue confronted would be of digit order issue and the closeness between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 and so forth [1].

CNN is playing an important role in many sectors like image processing. It has a powerful impact on many fields. Even, in nanotechnologies like manufacturing semiconductors, CNN is used for fault detection and classification. Handwritten digit recognition has become an issue of interest among researchers. There are many papers and articles are being published these days about this topic. In research, it is shown that Deep Learning algorithm like multilayer CNN using Keras with Theano and TensorFlow gives the highest accuracy in comparison with the most widely used machine learning algorithms like SVM, KNN. Because of its highest accuracy, Convolutional Neural Network (CNN) is being used on a large scale in image classification, video analysis, etc. Many researchers are trying to make sentiment recognition in a sentence.

CNN is being used in natural language processing and sentiment recognition by varying different parameters. It is challenging to get a good performance as more parameters are needed for the large-scale neural network. Many researchers are trying to increase the accuracy with less error in CNN. In another research, they have shown that deep nets perform better when they are trained by simple back propagation. Their architecture results in the lowest error rate on MNIST compare to NORB and CIFAR10. Researchers are working on this issue to reduce the error rate as much as possible in handwriting recognition. In one research, an error rate of 1.19% is achieved using 3-NN trained and tested on MNIST. Deep CNN can be adjustable with the input image noise. Coherence recurrent convolutional network (CRCN) is a multi-modal neural architecture. It is being used in recovering sentences in an image. Some researchers are trying to come up with new techniques to avoid drawbacks of traditional convolutional layers. Ncfm (No combination of feature maps) is a technique which can be applied for better performance using MNIST data sets. Its accuracy is 99.81% and it can be applied for large scale data. New applications of CNN are developing day by day with many kinds of research. Researchers are trying hard to minimize error rates. Using MNIST data sets and CIFAR, error rates are being observed. To clean blur images CNN is being used. For this purpose, a new model was proposed using MNIST data set. This approach reaches an accuracy of 98% and loss range 0.1% to 8.5%. In Germany, a traffic sign recognition model of CNN is suggested. It proposed a faster performance with 99.65% accuracy. Loss function was designed, which is applicable for light-weighted 1D and 2D CNN. In this case, the accuracy was 93% and 91% respectively [2].

# 

# 

# 3. FEASIBILITY STUDY

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment-in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such study precedes technical development and project implementation.

**Areas of project feasibility**

## 3.1 Technical feasibility

This assessment focuses on the technical resources available to the organization, It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves evaluation of the hardware, software and other requirements of the proposed system,

This project is technically feasible. The primary technical requirements include having the android mobile to use and test the application. If someone wants to further develop or upgrade the mobile application, Android Studio (3.0+) is required, where Python language for Android is used as a base level programming language. The TensorFlow is developed by google, which is the open-source platform and uses NumPy, Image classifier and koras for developing the Convolutional Neural Network (CNN) model. The system also can be developed if the new technology is acquired. Thus, all the ends of the technical feasibility were met.

## 3.2 Economic feasibility

This assessment typically involves a cost/benefits analysis of the project, helping organization determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility helping decision makers determine the positive economic benefits to the organization that the proposed project will provide.

This project is economically feasible. Since there is no need for any hardware equipment during the time of project development and implementations, the cost of the project hardware is zero. Similarly, the software loaded for this project is open source. Moreover, the technical requirements were already available free of cost and the software is open source so there was no father expenditure for buying software packages.

## 3.3 Operational feasibility

This assessment involves undertaking a study to analyze and determine whether-and how well-the organizations needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

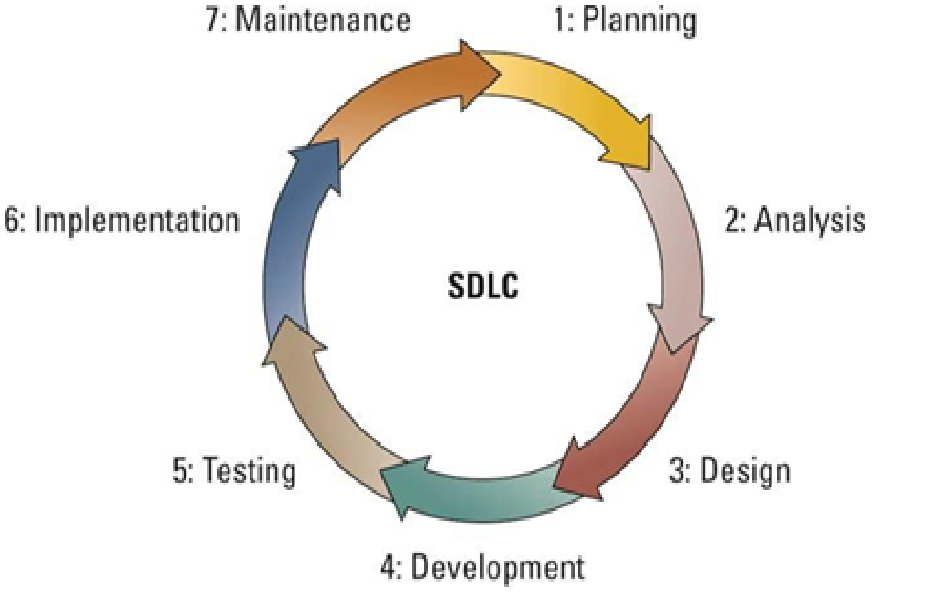
This project is operationally feasible. The proposed system is highly user friendly and is much easier to interact with the system. This system will be well documented for potential stakeholders. Therefore, the potential stakeholders will readily accept the system as they can easily get the result.

# 

# 4.METHODOLOGY

## 4.1 System development model

Different methodology is chosen according to the project. SDLC is the series of steps for the development of an application or software. The software development life cycle provides the series of sequence in which each step uses the result of the previous state. It aims to be the standard defines all the tasks required for developing and maintaining software. A typical software development life cycle consists of the following stages.



**Figure:Steps of SDLC**

**1.Planning**

Planning focuses on the scope of the project the outputs of the planning phase include project plans schedules, cost estimations, and procurement requirements.

**2. Requirements analysis**

The IT team gathers requirements form business stakeholder and subject matter experts. The output of this phase is usually a document that lists this requirement, agile methods, by contrast may produce a backlog of tasks to be performed.

**3. Designing the architecture**

One requirement is understood, the design process takes place. Output include design document that list the patterns and components selected for the project, code produced by the spikes, used as a starting point for development.

**4. Building or development stage**

On receiving system design document, the works is divided in modules/ units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle. The output of this phase is testable, functional software.

**5. Testing Stage**

Testing phase of the SDLC is arguably one of the most important. It is impossible to deliver quality software without testing. Methods for testing can include code quality, unit testing (function testing), integration testing, performance testing, security testing. The output of the testing phase is a functional software ready for deployment to a production environment.

**6. Implementation**

After successful testing the produce is delivered/ deployed to the customer for their use. As soon as the product of given to the customer, they will first do the beta testing. If any change is required of if any bugs are caught, then they will report it to the engineering team. Once those changes ant made, or the bugs are fixed then the final deployment will happen.

**7. Maintenance**

This phase is the “end of the beginning “. Though the SDLC doesn’t end here. Software must be monitored constantly to ensure proper operation. Once when the customers start using the developed system then the actual problem come up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

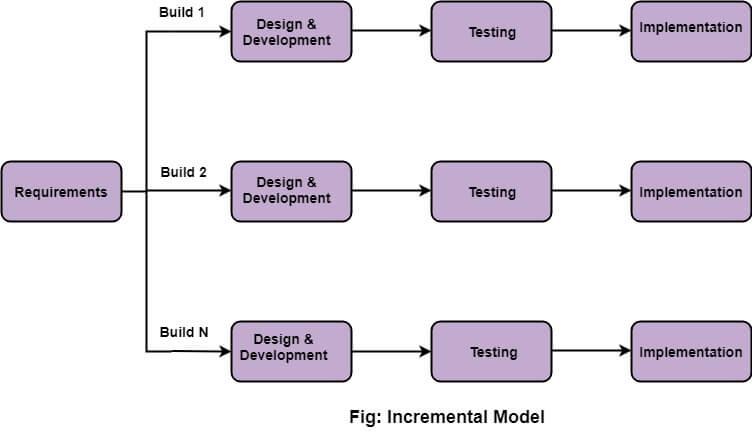
## 4.2 Type of SDLC

1. Waterfall Model
2. V-shaped Model
3. Prototyping Model
4. Spiral Model
5. Iterative and incremental Model
6. Agile Model

**Proposed Methodology: Incremental Model**

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.

Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.



**Source:https://www.javatpoint.com/software-engineering-incremental-model**

**Advantages of Incremental model:**

* Generates working software quickly and early during the software life cycle.
* This model is more flexible – less costly to change scope and requirements.
* It is easier to test and debug during a smaller iteration.
* In this model customer can respond to each built.
* Lowers initial delivery cost.
* Easier to manage risk because risky pieces are identified and handled during it’d iteration.

**Disadvantages of Incremental model:**

* Needs good planning and design.
* Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
* Total cost is higher than waterfall.

**When to use the Incremental model:**

* This model can be used when the requirements of the complete system are clearly defined and understood.
* Major requirements must be defined; however, some details can evolve with time.
* There is a need to get a product to the market early.
* A new technology is being used
* Resources with needed skill set are not available
* There are some high-risk features and goals.

## 4.3 Block Diagram and Algorithm

The processed methodology uses some techniques to remove the background noise and features extraction to detect and classify the handwritten digits

The proposed method comprises of 4 phases:

1. Pre-processing

2. Segmentation

3. Feature extraction

4. Classification and recognition

Take input from user

Pre-Processing user Input

Segmentation

Feature Extraction

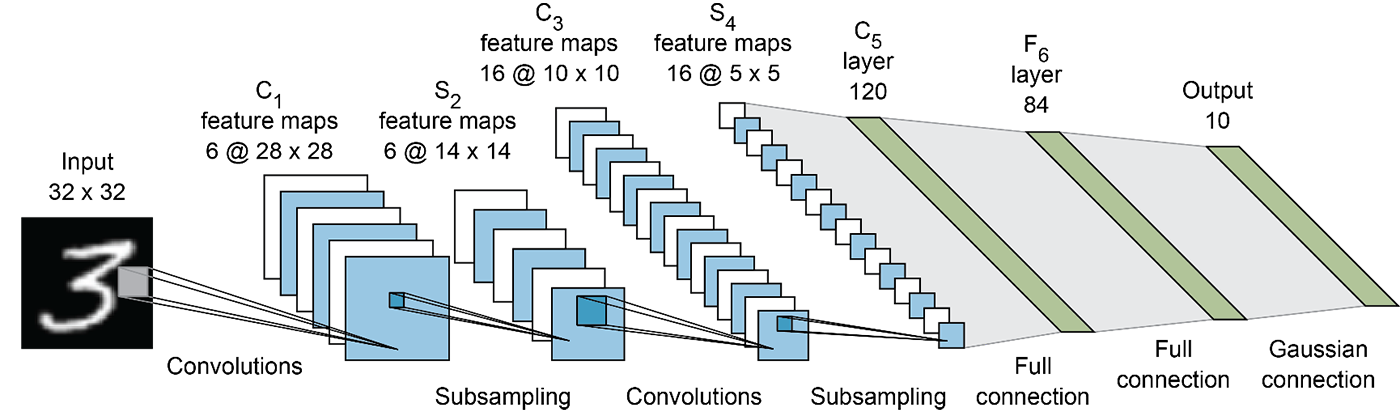
Classification and Recognition

**Fig: Block diagram of the system**

First, user uploads image of any digit which he wants to recognize. The image will be processed by the system. 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. For the most part, noise filtering, smoothing and standardization are to be done in this stage. It additionally characterizes a smaller portrayal of the example. Binarization changes over a gray scale image into a binary image. Pre-processed digit images are segmented into a sub image of individual digits. Each individual digit is re-sized into pixels. An edge detection technique is being used for segmentation of data set images. After the completion of pre-processing and segmentation stage, the pre-processed image is represented in the form of the matrix which contains pixels of the images that are of very large size. It will be valuable to represent the digit in the images which contains the necessary information. Redundancy from the data is removed in feature extraction. In the classification and recognition step the extracted feature vectors are taken as an individual input to three classifiers. They are K-Nearest Neighbor, Random forest classifier and support vector machine. The KNN algorithm considers new data point as its input and performs classification by calculating distance between new and labeled data points. The RFC classifier can deal with missing quantities. SVM provides a regularization parameter which avoids the over fitting problems. On running the system code the output is generated that shows which is the digit uploaded by the user and also displays the accuracy rate predicted by the model.

4.4 Working principle

The application runs by getting the user input in this case it is a numeric character from 0-9.



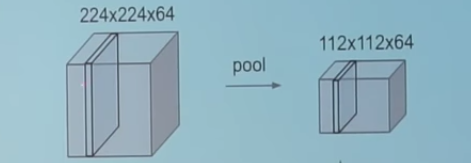
**Fig: Architecture of neural network**

**Convolutional**

The convolution layer is the main building block of a convolutional neural network. As we go deeper to other convolution layers, the filters are doing dot products to the input of the previous convolution layers. So, they are taking the smaller colored pieces or edges and making larger pieces out of them.

**Pooling**

A pooling layer is another building block of a CNN.



Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network. Pooling layer operates on each feature map independently. The most common approach used in pooling is max pooling.

**Fully connected**

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

**Receptive field**

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from *every* element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically, the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its receptive field. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer

**Weights**

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning, in a neural network, progresses by making iterative adjustments to these biases and weights.

The vector of weights and the bias are called filters and represent particular feature of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons can share the same filter. This reduces memory footprint because a single bias and a single vector of weights are used across all receptive fields sharing that filter, as opposed to each receptive field having its own bias and vector weighting

## 

## 4.5 Diagram

### 4.5.1 Context Diagram

A context diagram, sometimes called a level 0 data-flow diagram, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of information between the system and external entities. The entire software system is shown as a single process.

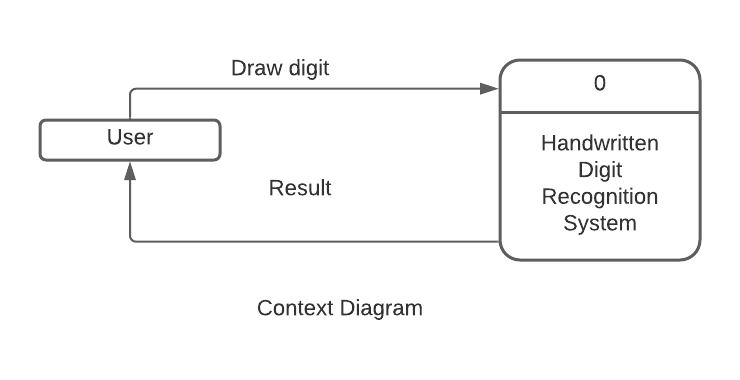


Figure: Context diagram

As shown in above figure, the context diagram has two element, user and system. User gives the input image to the system. System evaluate the image and gives result to user.

### 4.5.2 Level 0 Data Flow Diagram

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describe the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Level 0 DFD gives the overall design and data flow of the proposed system.

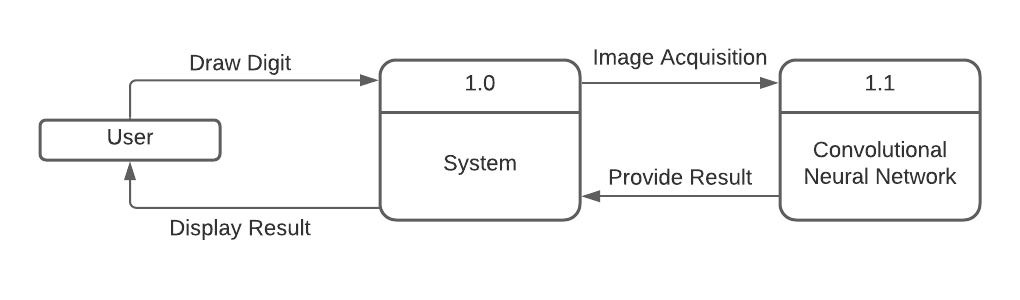
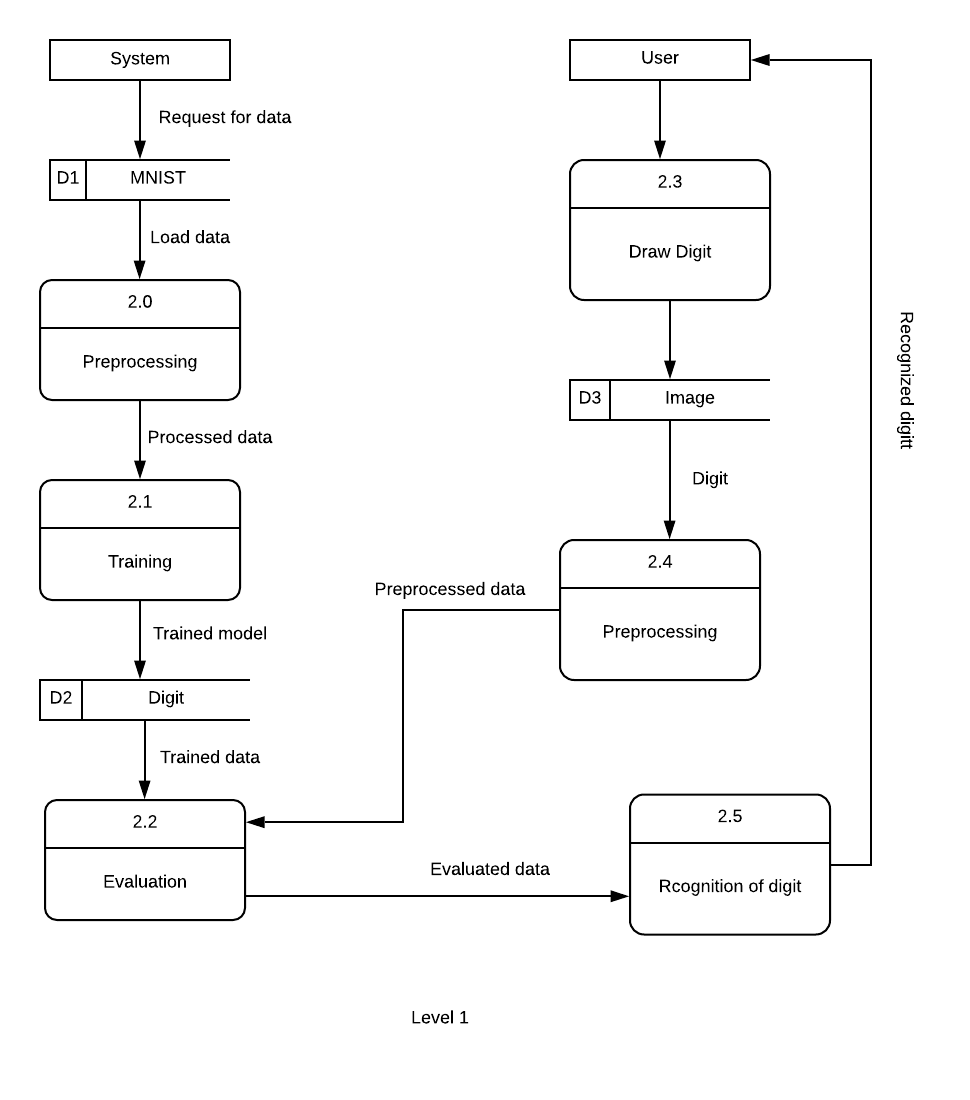


Figure: level 0 data flow diagram

The figure shows the DFD level 0 of the proposed system. Here user draw the digit through the system application and that image is feed to the convolution neural network. The output generated by the CNN is displayed to the system application.

### 

### 4.5.3 Level 1 Data Flow Diagram



Encoded

Character

Figure: level 1 Data flow diagram

Above shown figure gives the detail of the proposed system. First, image acquisition is performed. Then image is reshaped into 28\*28 pixel. Further image noise is removed by enhancing the image contrast and getting rid of small objects. The feature is extracted by combining all the distinguishing descriptors into a tanning feature matrix. Further the extracted feature vectors are taken as an individual input to three classifiers. They are K-Nearest Neighbor, Random forest classifier and support vector machine. The KNN algorithm considers new data point as its input and performs classification by calculating distance between new and labeled data points. The RFC classifier can deal with missing quantities. SVM provides a regularization parameter which avoids the over fitting problems. On running the system code the output is generated that shows which is the digit uploaded by the user and also displays the accuracy rate predicted by the model.

### 4.5.4 ER Diagram

An entity relationship diagram shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagram are used to sketch out the design of a database.

System application

User

CNN

has

has

Figure: Entity relationship diagram

### 4.5.5 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

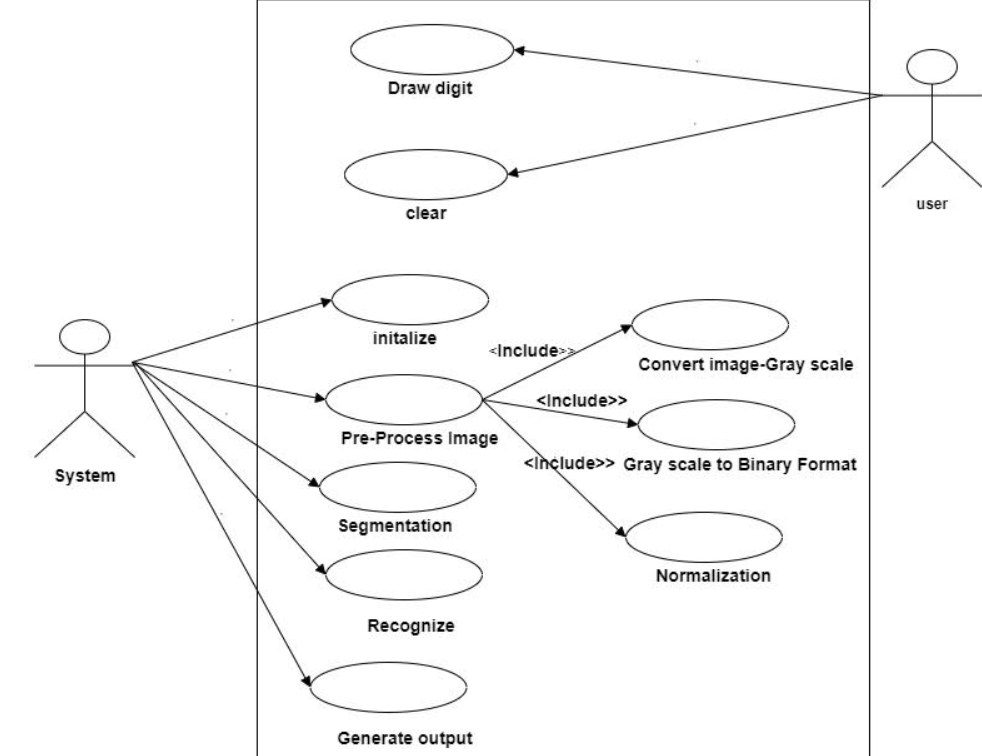
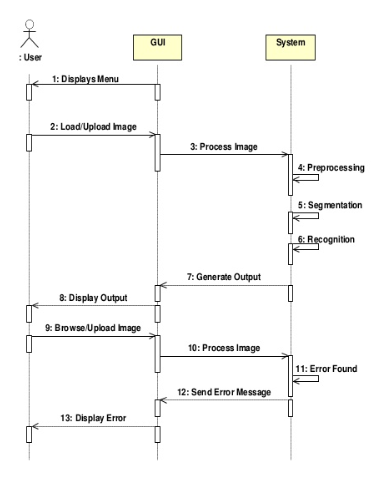


Figure: Use case diagram

In our system, there will be two entities, user and system. System has mobile application or system interface where we can draw digit or click photo of handwritten digit. Our system has several use case and they are draw digit, image acquisition, preprocessing, segmentation, extraction, classification, recognition.

### 

### 4.5.6 Sequence diagram



**Fig: Sequence diagram**

The figure illustrates the sequence diagram of the proposed model. The figure describes the sequence of steps to be taken while performing execution. The CNN model works in the following sequence. User uploads a image of any digit which he wants to recognize. The image will be processed by the system. On running the system code, the output is generated that shows which is the digit uploaded by the user and displays the accuracy rate predicted by the model. On uploading image with different resolutions other than the one mentioned in the code, the output generated shows error and displays an error message to the user.

# 5. PROJECT SPECIFICATION

A software requirements specification is a description of a software system to be developed. The SRS lays out function and non-functional requirements, and it may include a set of use cases that describes user interaction that the software must provide to the user for perfect interaction.

The SRS documents lists enough and necessary requirements for the project developments. To derive the requirements, the developer needs to have clear and thorough understanding of the product underdevelopment. This is achieved through detailed and continuous communication with the project team and customer throughout their software development process.

## 5.1 Functional requirements

Functional requirements define a function of a system or its components. A function is described as a set of inputs, the behavior and output. Its describes what the system should do. Function requirements specified that a system or components must be able to allow the user to perform function. The main purpose of the system is to help user in detection of numeric characters drawn by the user and display with simple user interface. The system will have two entities: User and convolutional neural networks. User can use the application to draw numeric character and obtain the digitized numeric character. Another entity convolutional neural network is the core part of the system which handles preprocessing, segmentation, extraction, classification, recognition, and finally display the most accurate result to the users.

## 5.2 Non-Functional requirements

Non- functional requirements are requirements that specifies criteria that can be used to judge the operation of the system, rather than specific behavior. In contrasts to functional requirements, it defines specific behavior of the function. These defines system properties and constraints such as:

1. Performance: Handwritten numeric characters will be recognized with an accuracy of about 95% and more.
2. Functionality: This software will deliver on the functional requirements mention in the documents.
3. Availability: This system will retrieve the handwritten regions only if it contains written characters in it.
4. Flexibility: It provides the region to draw the character.

## 5.3 System requirements

### 5.3.1 CNN (convolutional neural network)

CNNs are regularized version of multilayer perceptron. Multilayer perceptron usually refers to fully connected networks, that is , each neural in one layer is connected to all the neuron in the next layer. The fully connectedness of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. However, CNNs take a different approach towards regularization: they take advantages of the hierarchical pattern in data are assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extremity.

They are also known as shifts invariant or space invariant artificial neural networks (SIANN), based on their shared weights architecture and translation invariant characteristics. CNN are inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little preprocessing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand engineered. This independents from prior knowledge and human effort in feature design is a major advantage. They have application in image and video recognition, recommender systems, image classification, medical image analysis and natural language processing.

CNN is a deep learning algorithm which can take in an input image, assign important (learnable weight and biases) to various aspects / objects in the image and be able to differentiate one from the other. The preprocessing required in a convnet is much lower as compared to other classification algorithms. While in primitive method filter are hand engineered, with enough training, convnets can learn these filter/characteristics.

The architecture of the convnets is analogues to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the visual cortex. Individual neurons respond to stimuli only in a restricted region of the visual field. A collection of such fields overlaps to cover the entire visual area.

### 5.3.2 TensorFlow

TensorFlow is a google brains second generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extension for general purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, Mac OS, Windows and mobile computing platforms including android and iOS.

Is flexible architecture allowing for the easy deployment of computation across a variety of platforms (CPUs, GPUs and TPUs) and from desktop to clusters of servers to mobile and edge devices. TensorFlow is an end to end open source platform for machine learning. It has a comprehensive, flexible eco system of tools, libraries and community resources that lets researchers push the state of the art in ML and developers easily build and deploy ML powered applications.

TensorFlow computations are expressed as stateful dataflow graph. The name TensorFlow derived from the operation that such neural networks perform on multidimensional data arrays, which are refer to as tensors. During the google I/O conference in June 2016, jeff Dean stated that 1500 repositories in GitHub mentioned TensorFlow of which only 5 where form google.

In Jan 2018, google announced TensorFlow 2.0. In March 2018, google announced TensorFlow.js version 1.0 for machine learning for JavaScript and TensorFlow graphics for deep learning in computer graphics.

### 5.3.3 NumPy

Python is a high-level, dynamically typed metaparadigm programming language. Python code is often said to be almost like pseudo code, since is allows you to express very powerful ideas in very few lines of code while being very readable. NumPy is the core library for scientific computing in python. It provides a high- performance multidimensional array object, and tools for working with these arrays.[11]

NumPy is a python library which provides function for working with multi-dimensional data such as matrices of pixels that represent images. The NumPy library was used to test the idea of adding distortions to the image input in order to increase the performance of the CNN and classifier. NumPy is also used to divide the pixel the pixel values of the input images while transforming them to a size of 128x128.

NumPy is a general-purpose array- processing packing. It provides a high-performance multi-dimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with python. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

### 5.3.4 Keras

Keras is a high-level neural network API, written in python and capable of running on top of tensor flow, CNTK, or theano.it was developed with a focus on enabling fast experimentation. It allows for easy and fast prototyping, supports both convolutional network and recurrent networks, as well as combinations of the two and runs seamlessly on CPU and GPU. Designed to enables fast experimentation with deep neural network, it focuses on being user- friendly, modular, and extensible. Keras contains numerous implementations of commonly used neural-network building block such as layers, objectives activation function, optimizers, and a host of tools to make working with image and text data easier. Keras allows users to productize deep models on smartphones (iOS and android), on the web, or on the java Virtual machine. it also allows use of distributed training of deep-learning models on clusters of Graphics processing Unit (GPU)and tensor processing units (TPU).

Tf.Keras is TensorFlow’s implementation of the keras API specification. This is a high-level API to build and train models that includes first-class support for tensor flow-specific functionality, such as eager execution, tf data pipelines, and estimators. tf.keras makes Tensor flow easier to use without , sacrificing flexibility and performance.[12]

### 5.3.5 MINIST Dataset

All digit images have been size-normalized and centered in a fixed size image of 28 x 28 pixels. In the original dataset each pixel of the image is represented by a value between 0 and 255, where 0 is black, 255 is white and anything in between is a different shade of grey.

Here are some examples of MNIST digits:

0 1 2 3 4 5

For convenience we pickled the dataset to make it easier to use in python. It is available for download [here](http://deeplearning.net/data/mnist/mnist.pkl.gz). The pickled file represents a tuple of 3 lists: the training set, the validation set and the testing set. Each of the three lists is a pair formed from a list of images and a list of class labels for each of the images. An image is represented as NumPy 1-dimensional array of 784 (28 x 28) float values between 0 and 1 (0 stands for black, 1 for white). The labels are numbers between 0 and 9 indicating which digit the image represents. The code block below shows how to load the dataset.

# 

# 6. TESTING

## 6.1. Software Testing Overview

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

**Software Validation**

Validation is process of examining whether the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

* Validation ensures the product under development is as per the user requirements.
* Validation answers the question – "Are we developing the product which attempts all that user needs from this software?".
* Validation emphasizes on user requirements.

**Software Verification**

Verification is the process of confirming if the software is meeting the business requirements and is developed adhering to the proper specifications and methodologies.

* Verification ensures the product being developed is according to design specifications.
* Verification answers the question– "Are we developing this product by firmly following all design specifications?"
* Verifications concentrates on the design and system specifications.

Target of the test are -

* **Errors** - These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.
* **Fault** - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.
* **Failure**- failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

**Manual Vs Automated Testing**

Testing can either be done manually or using an automated testing tool:

* **Manual** - This testing is performed without taking help of automated testing tools. The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager.

Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.

* **Automated** This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools.

A test needs to check if a webpage can be opened in Internet Explorer. This can be easily done with manual testing. But to check if the webserver can take the load of 1 million users, it is quite impossible to test manually.

There are software and hardware tools which helps tester in conducting load testing, stress testing, regression testing.

**Testing Approaches**

Tests can be conducted based on two approaches –

* Functionality testing
* Implementation testing

When functionality is being tested without taking the actual implementation in concern it is known as black-box testing. The other side is known as white-box testing where not only functionality is tested but the way it is implemented is also analyzed.

Exhaustive tests are the best-desired method for a perfect testing. Every single possible value in the range of the input and output values is tested. It is not possible to test each and every value in real world scenario if the range of values is large.

**Black-box testing**

It is carried out to test functionality of the program. It is also called ‘Behavioral’ testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested ‘ok’, and problematic otherwise.

In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

Black-box testing techniques:

* **Equivalence class** - The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
* **Boundary values** - The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
* **Cause-effect graphing** - In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
* **Pair-wise Testing** - The behavior of software depends on multiple parameters. In pairwise testing, the multiple parameters are tested pair-wise for their different values.
* **State-based testing** - The system changes state on provision of input. These systems are tested based on their states and input.

**White-box testing**

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as ‘Structural’ testing.

In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

The below are some White-box testing techniques:

* **Control-flow testing** - The purpose of the control-flow testing to set up test cases which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.
* **Data-flow testing** - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified.

Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels -

**Unit Testing Testing Levels**

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

**Integration Testing**

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updating etc.

**System Testing**

The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

* **Functionality testing** - Tests all functionalities of the software against the requirement.
* **Performance testing** - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.
* **Security & Portability** - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

**Acceptance Testing**

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

* **Alpha testing** - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.
* **Beta testing** - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

**Regression Testing**

Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code. This is known as regression testing.

**Testing Documentation**

Testing documents are prepared at different stages -

**Before Testing**

Testing starts with test cases generation. Following documents are needed for reference –

* **SRS document** - Functional Requirements document
* **Test Policy document** - This describes how far testing should take place before releasing the product.
* **Test Strategy document** - This mentions detail aspects of test team, responsibility matrix and rights/responsibility of test manager and test engineer.
* **Traceability Matrix document** - This is SDLC document, which is related to requirement gathering process. As new requirements come, they are added to this matrix. These matrices help testers know the source of requirement. They can be traced forward and backward.

**While Being Tested**

The following documents may be required while testing is started and is being done:

* **Test Case document** - This document contains list of tests required to be conducted. It includes Unit test plan, Integration test plan, System test plan and Acceptance test plan.
* **Test description** - This document is a detailed description of all test cases and procedures to execute them.
* **Test case report** - This document contains test case report as a result of the test.
* **Test logs** - This document contains test logs for every test case report.

**After Testing**

The following documents may be generated after testing:

* Test summary - This test summary is collective analysis of all test reports and logs. It summarizes and concludes if the software is ready to be launched. The software is released under version control system if it is ready to launch.

**Testing vs. Quality Control, Quality Assurance and Audit**

We need to understand that software testing is different from software quality assurance, software quality control and software auditing.

* **Software quality assurance** - These are software development process monitoring means, by which it is assured that all the measures are taken as per the standards of organization. This monitoring is done to make sure that proper software development methods were followed.
* **Software quality control** - This is a system to maintain the quality of software product. It may include functional and non-functional aspects of software product, which enhance the goodwill of the organization. This system makes sure that the customer is receiving quality product for their requirement and the product certified as ‘fit for use’.
* **Software audit** - This is a review of procedure used by the organization to develop the software. A team of auditors, independent of development team examines the software process, procedure, requirements and other aspects of SDLC. The purpose of software audit is to check that software and its development process, both conform standards, rules and regulations.

# 

# 

# 7. DISCUSSION AND CONCLUSION

Handwritten digit recognition is the first step to the vast field of Artificial Intelligence and Computer Vision. With the advance of technology, every day there are new algorithms coming up which can make computers do wonders.

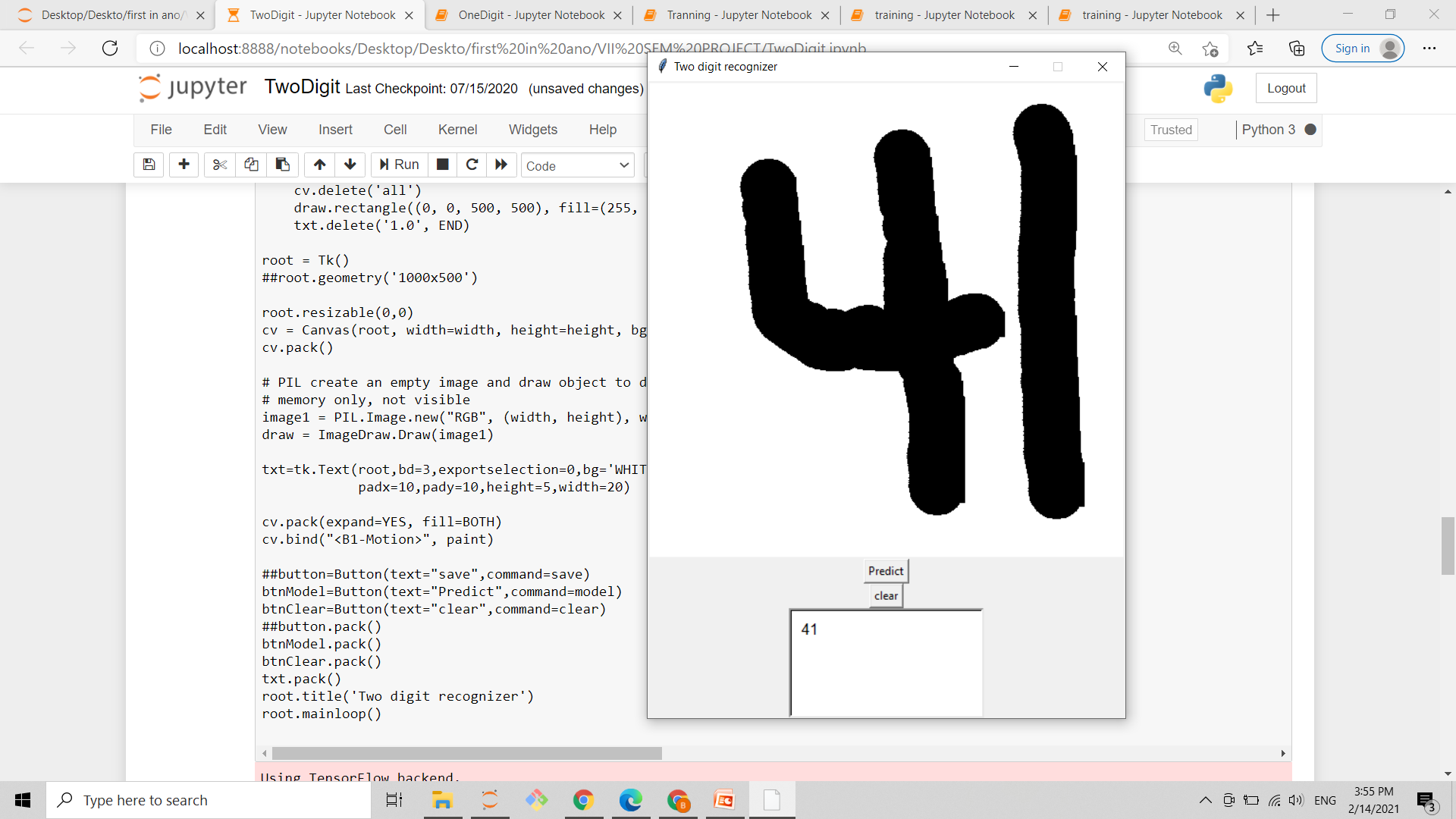
Machines can recognize images and understand handwriting of different people which humans themselves have failed to understand or comprehend from the image. As seen from the results of the experiment, CNN proves to be far better than other classifiers. The results can be made more accurate with more convolution layers and a greater number of hidden neurons. The target of the proposed work was to open the way to digitization. Though the goal is to just recognize the digits, but it can be extended to letters and then a person’s handwriting. This can completely abolish the need of typing. Digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques of deep learning. The broader aim in mind was to develop a M.L. model that could recognize people's handwriting. However, as we began developing the model, we realized that the topic in hand was too tough and would require tremendous data to learn. Example to accurately classify a cursive handwriting will be very tough. Thus, we settled on classifying a given handwritten digit image as the required digit using CNN algorithms, consequently testing its accuracy and we get 99.45 accuracy with loss of 0.0188. In future we are planning to further explore the topic to recognize people’s handwriting. Limitations of our project are:

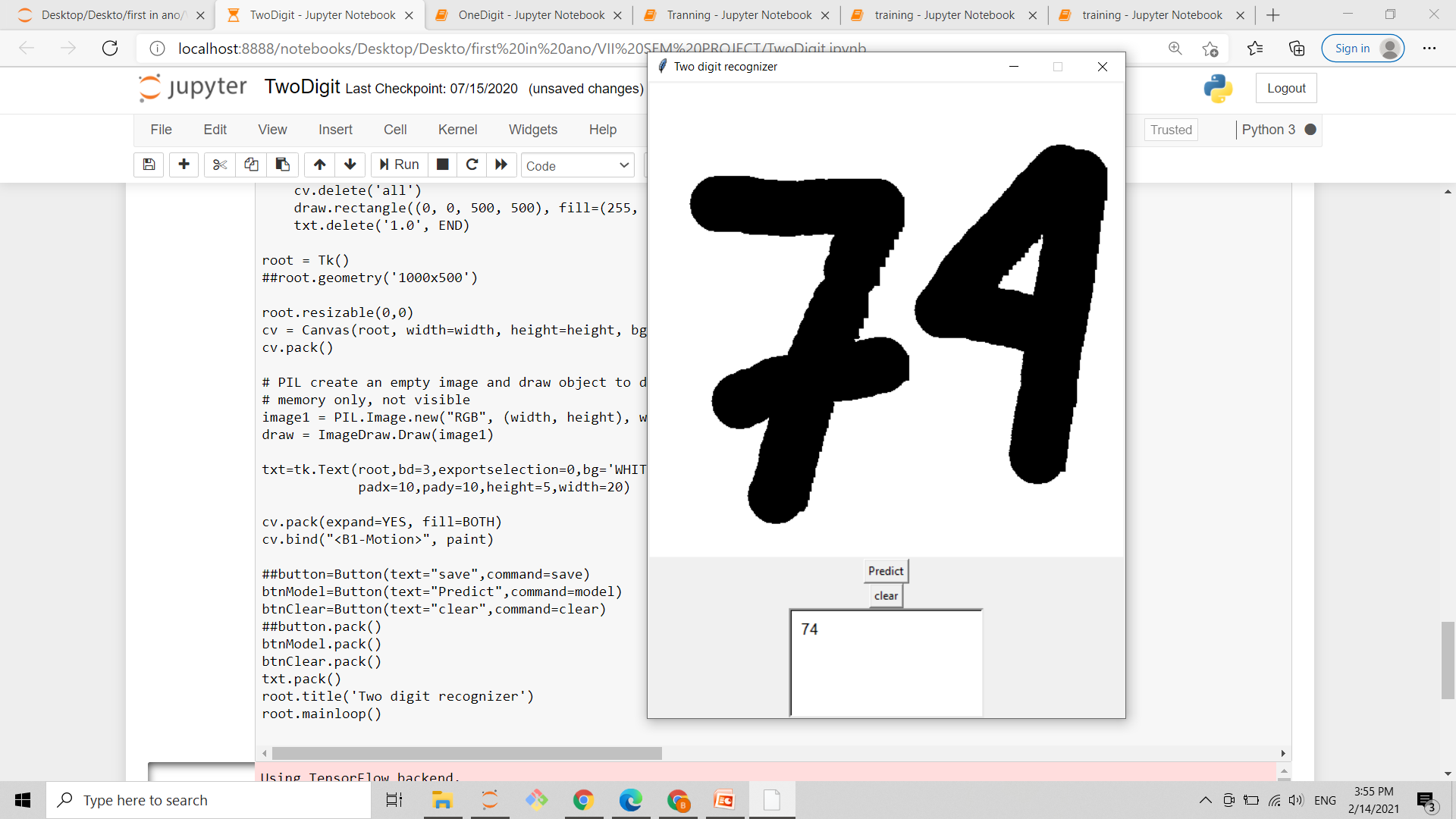
1. This application can be used for numeric character only.

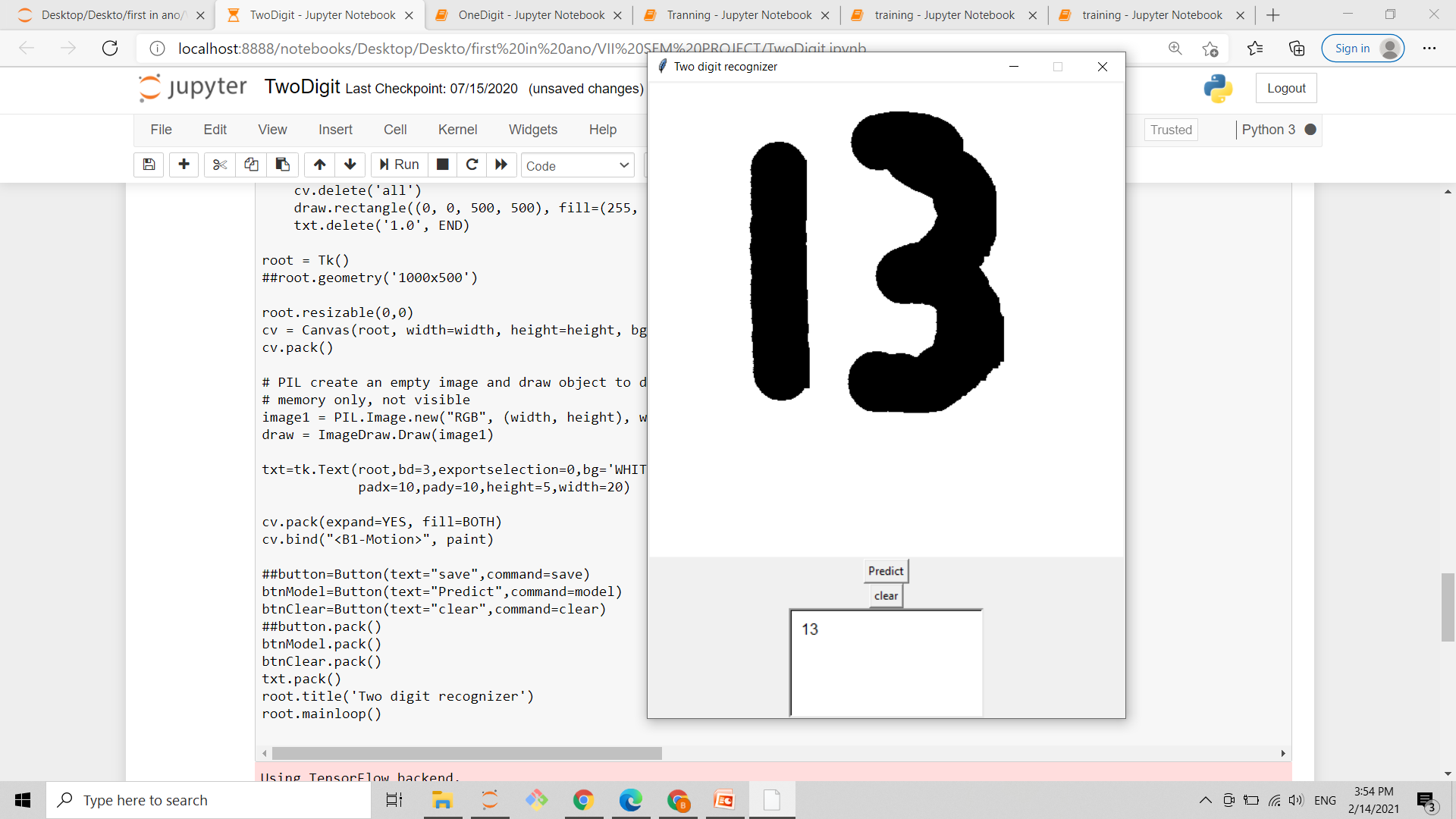
2. The application correctly predict for one and two digit. If there is more than two digits then it print predicted digit in list.

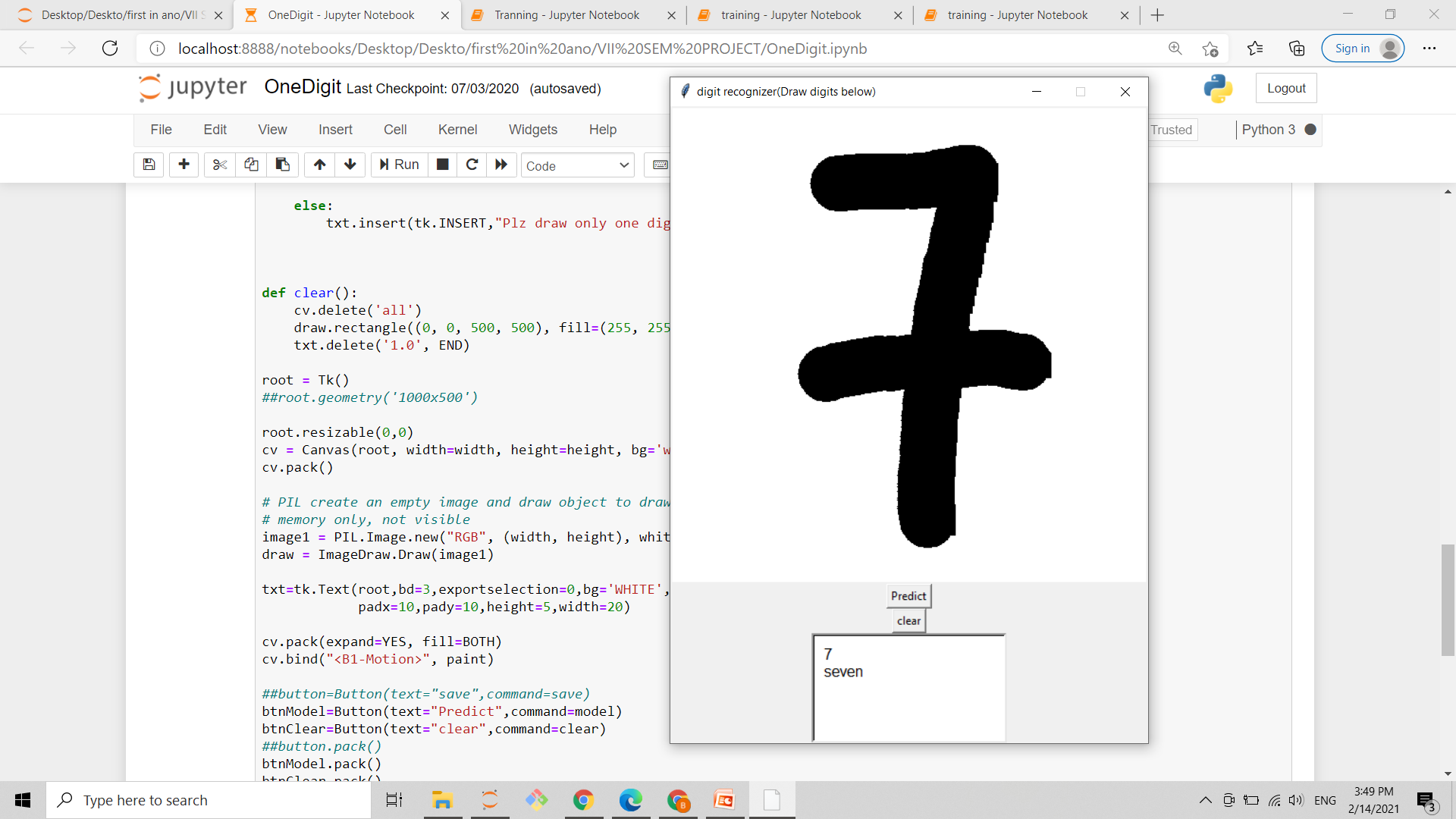
3. The digit should be written by continuous line.

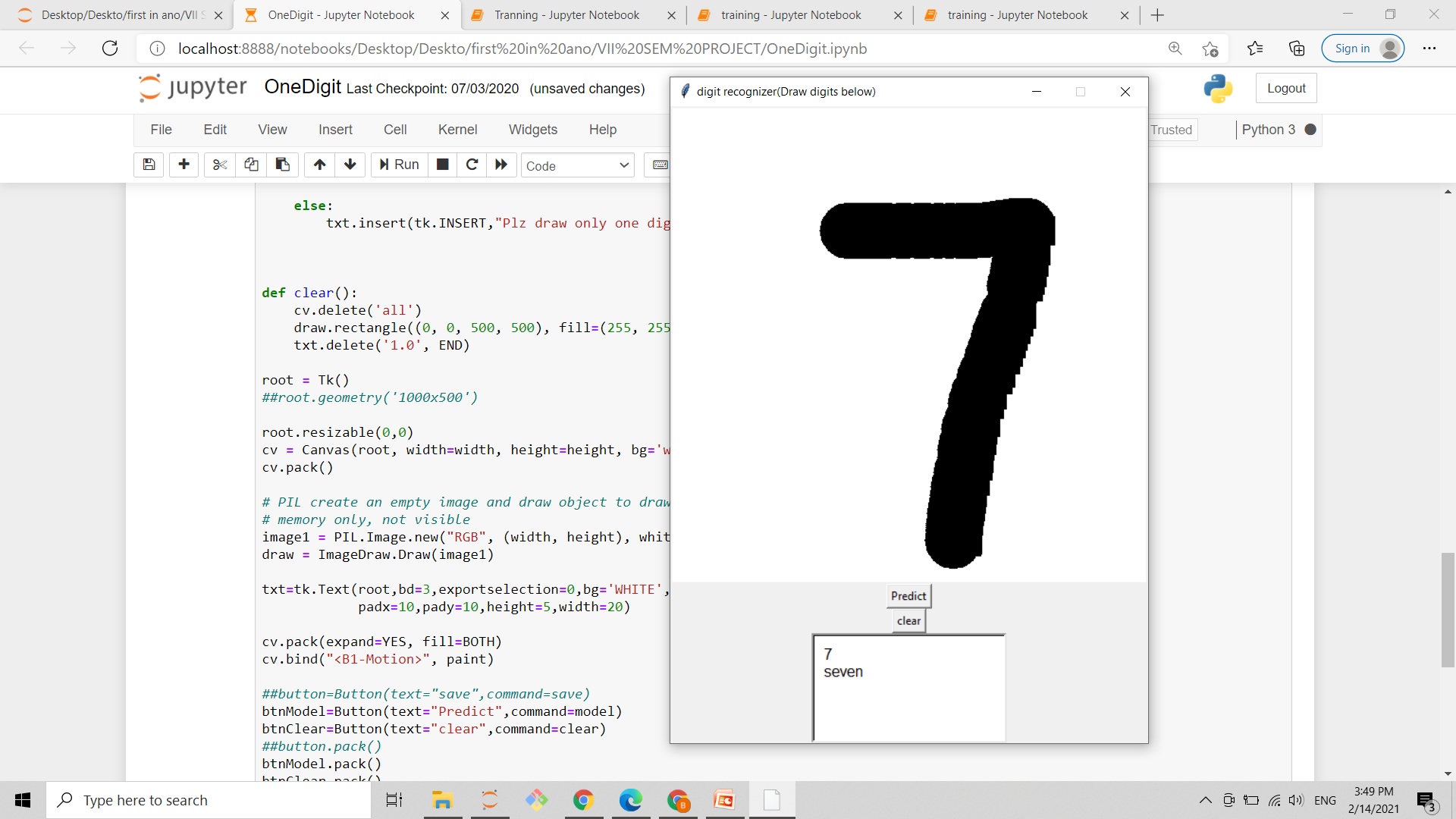
# 8. ANNEX A - SYSTEM SNAPSHOTS











# 9. REFERENCE

1.<https://ijisrt.com/wp-content/uploads/2019/06/IJISRT19JU358.pdf>(accessed 5 Dec 2019).

2. <https://arxiv.org/ftp/arxiv/papers/1909/1909.08490.pdf> (accessed 5 Dec 2019).

3. <https://www.smartdraw.com/use-case-diagram/>( accessed10 Dec 2019).

4. <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05>(accessed 20 Dec 2019).

5.https://www.javatpoint.com/software-engineering-incremental-model(accessed 14 Jan 2020).