CHAPTER-1
INTRODUCTION

1.1 Motivation

Fake Indian Currency Note (FICN) is a term used by officials and media to refer to counterfeit currency notes circulated in the Indian economy. In 2012, while responding to a question in parliament, the Finance Minister, P. Chidambaram, admitted that there is no confirmed estimate of fake currency in India. However, several central and state agencies are working together, and the Ministry of Home Affairs has constituted the Fake Indian Currency Notes Co-ordination Center (FCORD) to curb this menace. Identifying a currency note as fake or real from its image is essentially a binary image classification task. Here we test the feasibility of CNN models for fake currency identification, which can be trained without manual feature extraction on raw images of currency notes with a simple, efficient and very accurate approach.

1.2 Commencement

In the last eight years more than 3.53 lakh cases of counterfeit currency detection in India's banking channels is heighten according to latest government reports. The practice of counterfeiting became more refined with the arrival of paper currency. The Indian Government has taken a astonishing stride of demonetizing 500 and 1000 Rs. notes. Prime Minister Shree. Narendra Modi stated that one of the cognition for this policy was to counter the climbing menace of counterfeit Indian Currency notes.

However, the Indian banks acknowledged an all-time peak amount of fake currency and also noticed an over 480% increment in doubtful transactions after demonetization, a first ever report on questioning credits ended in the wake of 2016 notes ban has has discovered . The Reserve Bank of India(RBI) is the only one which has the singular authority to issue bank notes in India. The RBI being the highest monetary authority in the country, prints the currency notes of all denominations from Rs.2 to 2000. Several security features have been published by the RBI so that the counterfeit notes can be detected by the general public.

1.3 Problem Definition

In the existing system firstly we have the step called image acquisition means we

have to take input as the image only through the scanner and in this there is no use of any digital camera to capture the image in the real time system. In this existing architecture, only the front part of the note is take into consideration and not the rear part. After that we have next step called as pre-processing method. In this there are basically 3 to 4 sub stages involved like pre-processing, grayscale conversion, edge detection and segmentation.The existing system doesn't give efficient results and it's accuracy rate is very low. Image processing and pattern recognition area is the base area for feature extraction.

There are some techniques which are specifically from the application area of Image processing and pattern recognition and using those techniques the researchers achieved the accuracy rate ranges from 77% to 100%.

Few of the area are those which are the combination of machine learning and Image processing and therein the accuracy rate ranges from 96% to 100%.

The variation in accuracy rate is due to various factors such as choice of feature, type of note, number of features considered, country specific currency security feature, cost efficiency of the program.

1.4 Solution

Automatic currency note recognition technology is specific to a country and can be generalized with standard banknotes of each country. If there is a system which can identify a currency note as fake through a camera image is one promising direction towards solving this problem. Convolutional neural network models have seen tremendous success in image classification tasks. And identifying a currency note as fake or real from its image is essentially a binary image classification task. Here we test the feasibility of CNN models for fake currency identification, which can be trained without manual feature extraction on raw images of currency notes with a simple, efficient and very accurate approach.

However, distinguishing a counterfeit note just by visual per lustration is not an easy task. Moreover, an average person is unaware of all the security features. Developing applications which can detect a currency note to be counterfeit by a camera image can help solve this problem. Deep learning models have witnessed a

tremendous success in image classification tasks . Our model proposes a binary image classification task with two classes-fake or real. The Deep CNN model we have built helps us detect the counterfeit note without actually manually extracting the features of images. By training the model on the generated data set, the model learns on it and helps us detect a counterfeit note.

1.5 Objectives

The main objective of the project is to determine whether the currency is counterfeit one or the original one. The term Counterfeit means money is imitation of the currency produced without the legal sanction of the state or government producing or using this fake money is a form of fraud or forgery. So, by using CNN we can detect the fake currency with high accuracy and we can prevent the fraud and forgery of the currencies. Identify the fake Indian currency notes automatically using CNN Algorithm. Although there were many methods in existence, this method was designed to overcome the drawbacks of the previous methods. This method gives a faster and more accurate output when compared to the other techniques.

CHAPTER 2
LITERATURE SURVEY

2.1 Introduction

Fake currency detection is a serious issue worldwide, affecting the economy of almost every country including India. Currency duplication also known as counterfeit currency is a vulnerable threat on economy. It is now a common phenomenon due to advanced printing and scanning technology. The possible solutions are to use either chemical properties of the currency or to use its physical appearance. The approach presented in this paper is based upon physical appearance of the Indian currency. Image processing algorithms have been adopted to extract the features such as security thread, intaglio printing (RBI logo) and identification mark, which have been adopted as security features of Indian currency. Hence, we propose a more user friendly and portable solution to this problem in form of an mobile app coupled with cloud storage.

2.1.1 Detection Of Fake Currency Using Image Processing

The advancement of color printing technology has increased the rate of fake currency note printing and duplicating the notes on a very large scale. Few years back, the printing could be done in a print house, but now anyone can print a currency note with maximum accuracy using a simple laser printer. As a result the issue of fake notes instead of the genuine ones has been increased very largely. India has been unfortunately cursed with the problems like corruption and black money .And counterfeit of currency notes is also a big problem to it. This leads to design of a system that detects the fake currency note in a less time and in a more efficient manner. The proposed system gives an approach to verify the Indian currency notes. Verification of currency note is done by the concepts of image processing.

2.1.2 Real Time Fake Currency Note Detection using Deep Learning

Great technological advancement in printing and scanning industry made counterfeiting problem to grow more vigorously. As a result, counterfeit currency affects the economy and reduces the value of original money. Thus it is most needed to detect the fake currency. Most of the former methods are based on hardware and image processing techniques. Finding counterfeit currencies with these methods is less

efficient and time consuming. To overcome the above problem, we have proposed the detection of counterfeit currency using a deep convolution neural network. Our work identifies the fake currency by examining the currency images. The transfer learned convolutional neural network is trained with two thousand, five hundred, two hundred and fifty Indian currency note data sets to learn the feature map of the currencies.

2.1.3 Fake Banknote Recognition Using Deep Learning

Recently, some state-of-the-art works have used deep learning-based architectures, specifically convolutional neural networks (CNNs), for banknote recognition and counterfeit detection with promising results. However, it is not clear which design strategy is more appropriate (custom or by transfer learning) in terms of classifier performance and inference times for massive data applications. This paper presents a comparison of the two design strategies in various types of architecture. For the transfer learning (TL) strategy, the most appropriate freezing points in CNN architectures (sequential, residual and Inception) are identified. In addition, a custom model based on an AlexNet-type sequential CNN is proposed. Both the TL and the custom models were trained and compared using a Colombian banknote dataset.

2.2 Existing System

- In the Existing System, it takes input as the image only through the scanner and it only scans the front part of the currency note.
- Next step is pre - processing method in this there are basically 3-4 sub stages involved.
- The sub stages are Grayscale Conversion, Edge Detection, Image Segmentation, Feature Extractions.
- In the existing System it only processes the images by using the image processing techniques .
- The recognition of patterns doesn’t take place by the image processing.
- Due to the different sizes and patterns of notes the image processing techniques cannot detect the fake notes efficiently and gives low accuracy rate.

2.2.1 Disadvantages of Existing System

- The existing system cannot scans the Rear part of the Currency note.
- Fake Currency note is not detected efficiently.
- The existing system gives less than 90% accuracy rate.

- Image processing and pattern recognition area is the base area for feature extraction there are some specific application areas in it which leads this technique to achieve the low accuracy.

Fig.2.2.1.1: Blue Ray Scanner

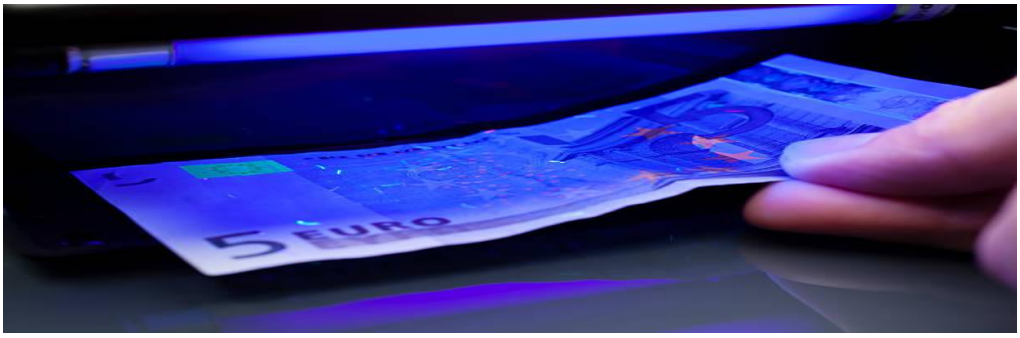


Fig. 2.2.1.2: Scanning Currency Note

2.3 Proposed System

- In the proposed system, we take the scanned images as a input from the dataset which contains the scanned images of both fake and original images.
- We can build algorithm and it will analyse automatically using Machine Learning methods.
- We can use this is in AI robots and in this we can also Image processing, CNN , SVM algorithms.
- By using CNN algorithm the currency notes are trained .
- By using SVM algorithm we can classify the notes as fake or real.
- The Scanned image is compared with the image from the dataset then a matrix is formed for the scanned image.
- The dataset contains the original images of currency notes each original note contains the matrix.
- If the scanned image matrix and the dataset image matrix is same then it is a

original currency note otherwise it is a fake currency note.

2.3.1 Advantages of Proposed System

- By using CNN algorithm we can efficiently detect the fake currency notes.
- Few of the areas are those which are the combination of Machine learning and image processing and therein the accuracy rate is high.
- Without the manual involvement we can detect the fake currency notes automatically by using CNN.
- CNN will efficiently train the model to detect the fake or real currency notes.
- We get the accuracy rate upto 96% to 100%.

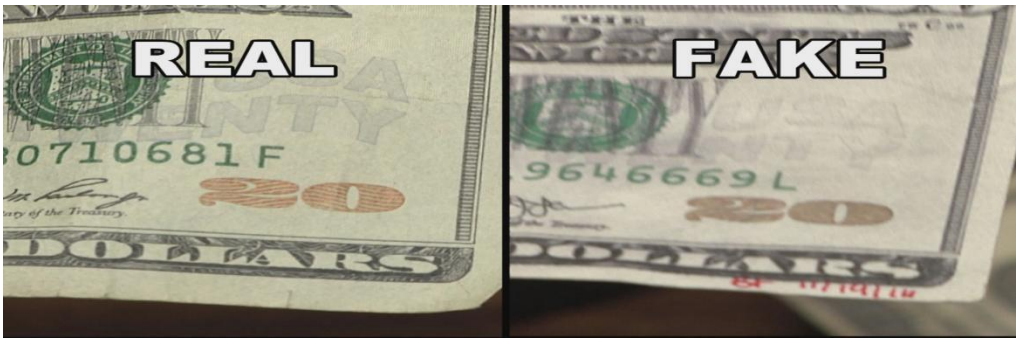


Fig.2.3.1.1: Real and Fake Currency

CHAPTER-3

ANALYSIS

3.1 Introduction

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches .The most critical section of the project plan is a listing of high level product requirements, also referred to as goals .All of the software product requirements to be developed during the requirements definition stage flow from one or more of the these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high-level estimates of effort for the out stages.

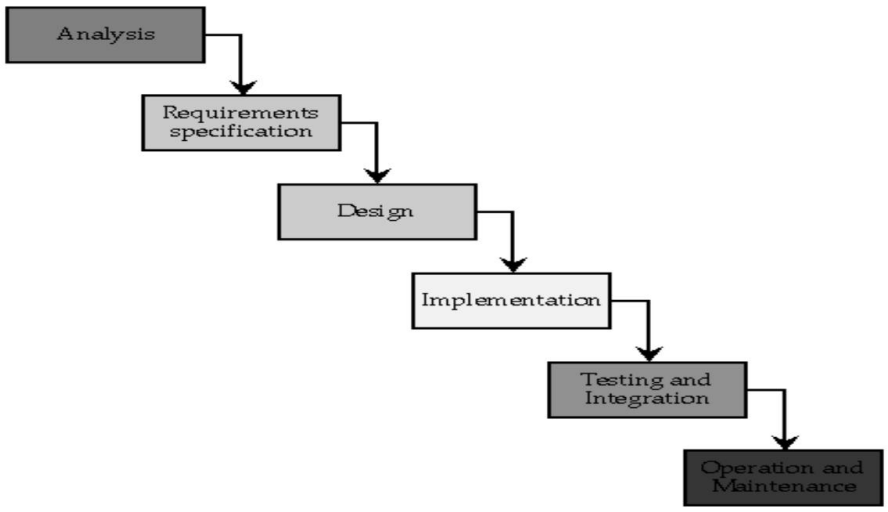


Fig.3.1:Proposed System

3.2 Requirement Analysis

This process is also known as feasibility study. In this phase, the development team studied the site requirement. They investigate the need for possible dynamic representation of the site and increase security features. By the end of feasibility study,

the team furnishes a document that holds the different specific recommendations for the candidate system. It also includes personnel assignments, costs, project schedules, target dates etc. the requirement gathering process is intensified and focused specially on software. The essential purpose of this phase is to find the need and to define the problem that needs to be solved. During this phase following facts were gathered.

- Determined the user need
- Identified the facts
- Establish the goals and objective for the proposed system
- Feasibility for the new system

3.3 System Analysis and Design

In this phase the software's overall structure and its nuances are defined. In terms of client server technology, the no of tiers needed for the package architecture, database design, data structure design etc, are defined in this phase. Analysis and Design are very crucial in entire development cycle. Any glitch in this phase could be expensive to solve in the later stage of software development. Hence following is the essential approach taken during website designing:

- DFD
- Database Designing
- Form Designing
- Pseudo code for methods

3.4 Testing

Once the code is generated, the website testing begins. Different testing methodologies are done to unravel the bugs that were committed during the previous phases. Different testing methodologies are used:

- Acceptance testing
- White Box Testing
- Black Box Testing

3.5 Requirement Specification

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system yow more complex it became evident that the goal of

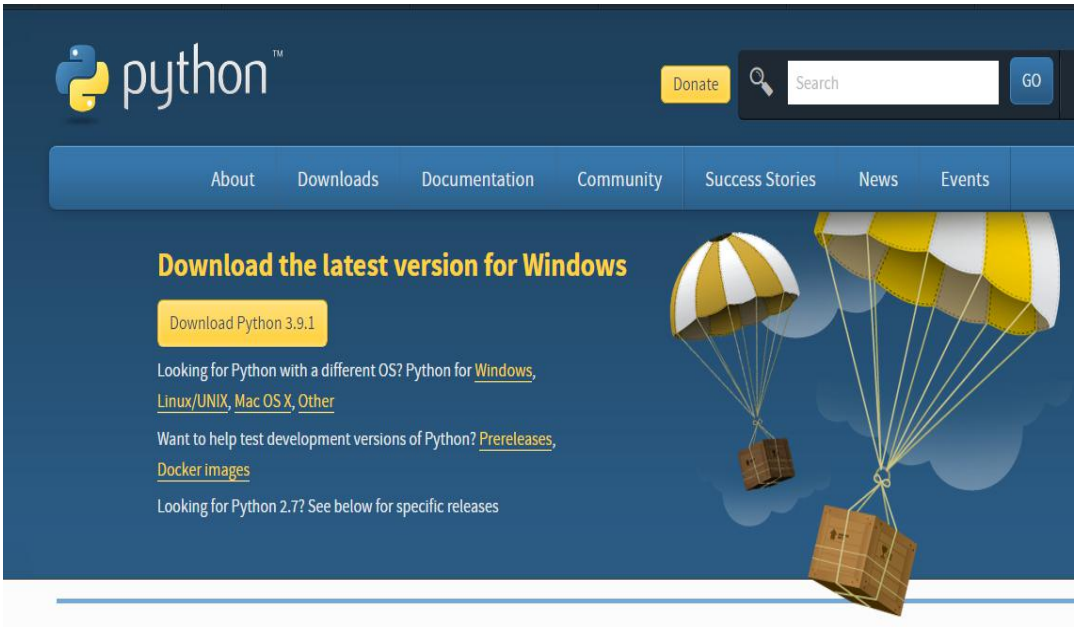
the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software is initiated by the client needs. The SRS is the means of translating the ideas of the minds of the clients (the input) into a formal document (the output of the requirement phase). The SRS phase consists of two basic activities: The process is order and more nebulous of two, deals with understand the problem, the goal and constraints. Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The requirements phase terminates with the production of validate SRS document. Producing the SRS document is the basic goal of this phase. The Software Requirements Specification (SRS) begins the translation process that converts the software requirements into the language the developers will use. The SRS draws on the use-cases from the User Requirement Document (URD) and analyses the situations, and omissions before development progresses significantly under mistaken assumptions.

3.6 SOFTWARE INSTALLATION FOR THIS PROJECT

3.6.1 Python Installation

Step-1:

Go to https://www.python.org/downloads. The most recent version of Python will always appear on the "Download" button near the top of the page.



Step-2:

Once the download is complete, run the exe for install Python. Now click on Install Now.

Step-3:

You can see Python installing at this point.

Step-4:

When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

3.6.2 Pycharm Installation

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

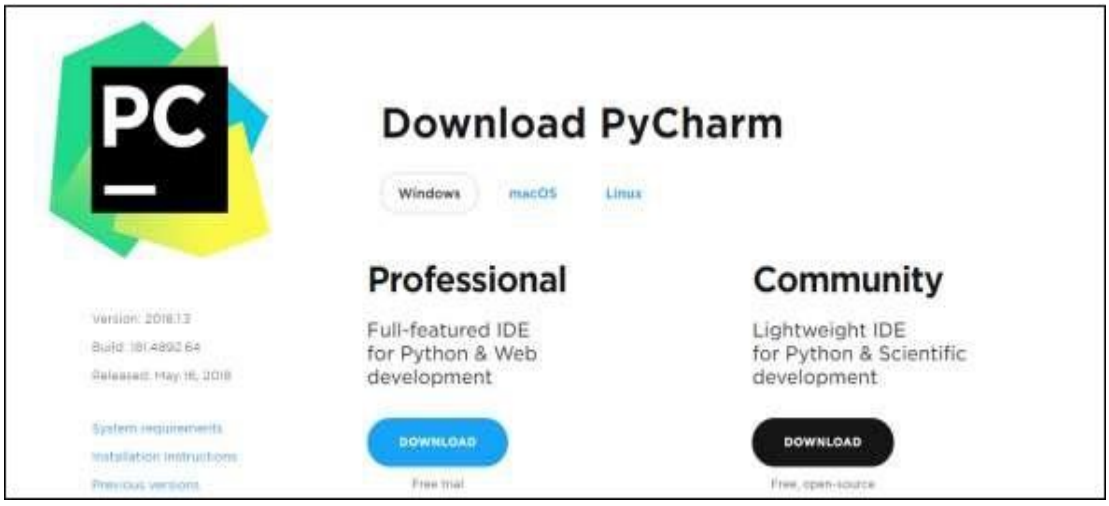
PyCharm offers some of the best features to its users and developers in the following aspects:

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask.

Step-1:

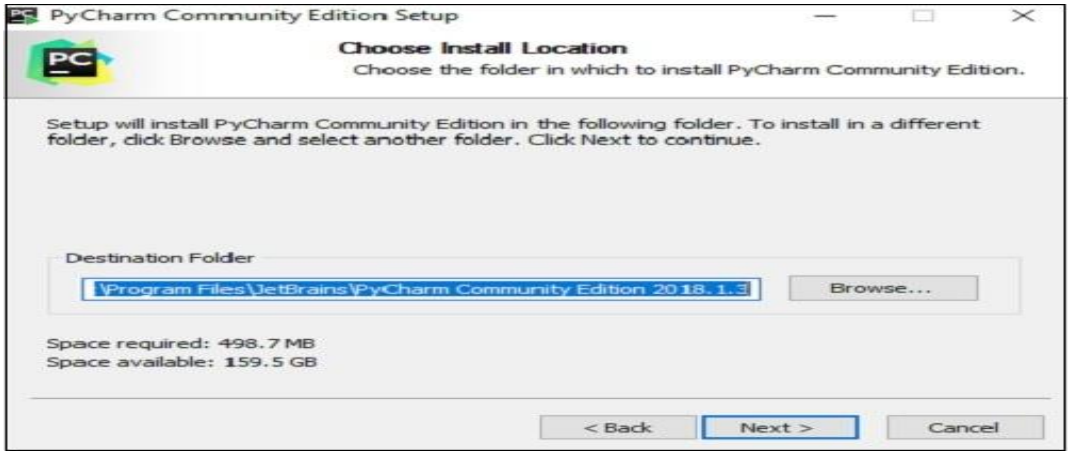
- Download the required package or executable from the official website of PyCharm https://www.jetbrains.com/pycharm/download/#section=windowsHere you will observe two versions of package for Windows as shown in the screenshot given below −

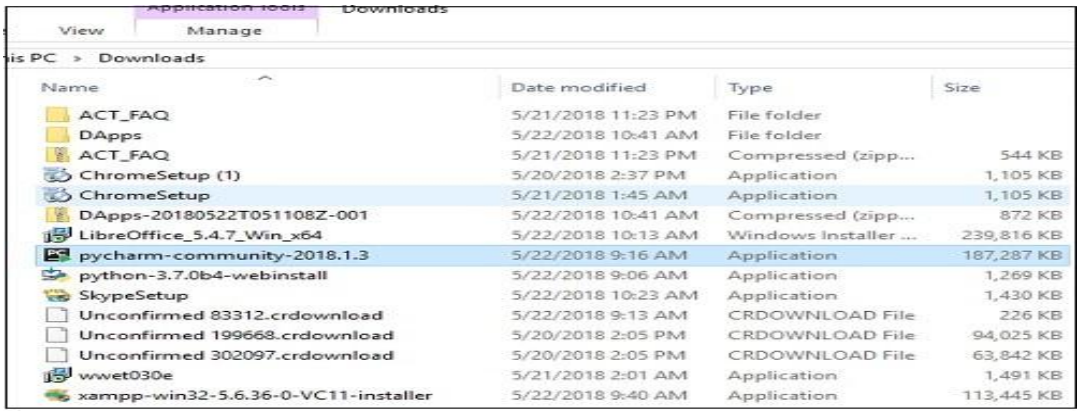
- Note that the professional package involves all the advanced features and comes with free trial for few days and the user has to buy a licensed key for activation beyond the trial period. Community package is for free and can be downloaded and installed as and when required. It includes all the basic features needed for installation. Note that we will continue with community package.



Step-2:

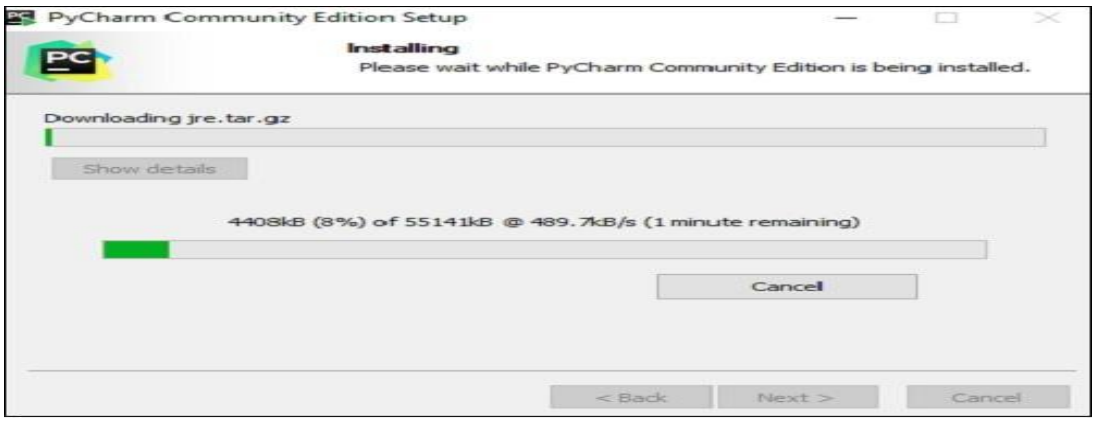
Download the community package (executable file) onto our system and mention a destination folder .





Step-3:

Now, begin the installation procedure similar to any other software package.

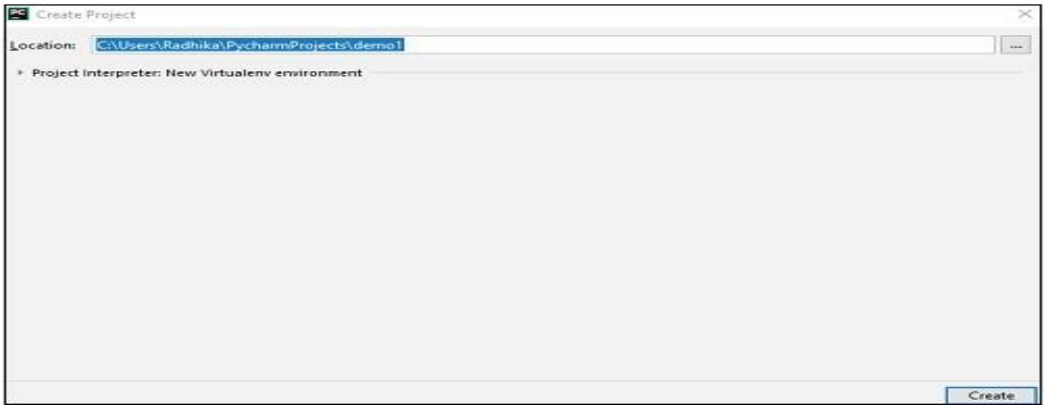


Step-4:

Once the installation is successful, PyCharm asks you to import settings of the existing package if any.

Step-5:

This helps in creating a new project of Python where you can work from the scratch. Note that unlike other IDEs, PyCharm only focusses on working with projects of Python scripting language



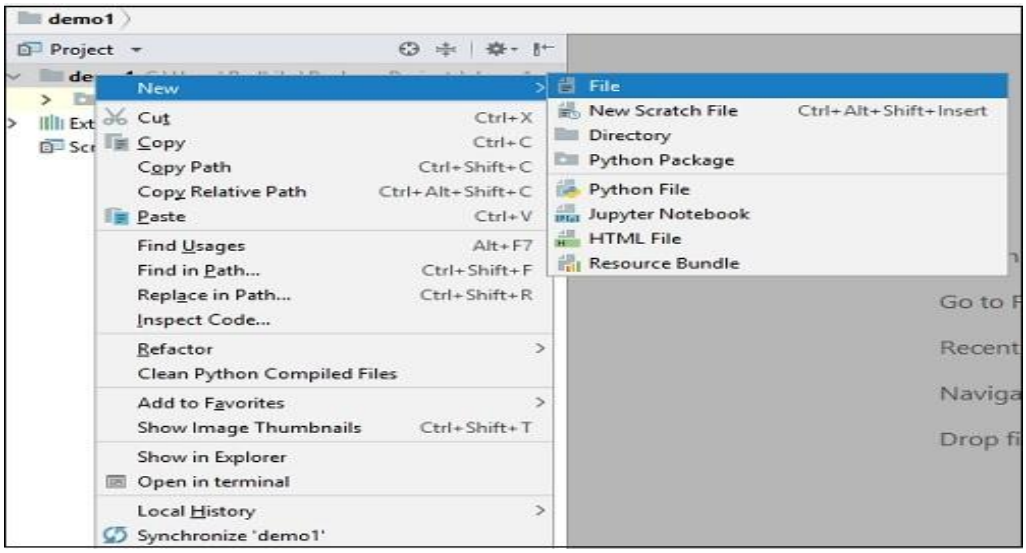
Step-6:

When you launch PyCharm for the first time, you can see a welcome screen with entry points to IDE such as −

- Creating or opening the project
- Checking out the project from version control
- Viewing the documentation
- Configuring the IDE

Step-7:

we created a project named **demo1** and we will be referring to the same project throughout this tutorial. Now we will start creating new files in the same project to understand the basics of PyCharm Editor.



Step-8:

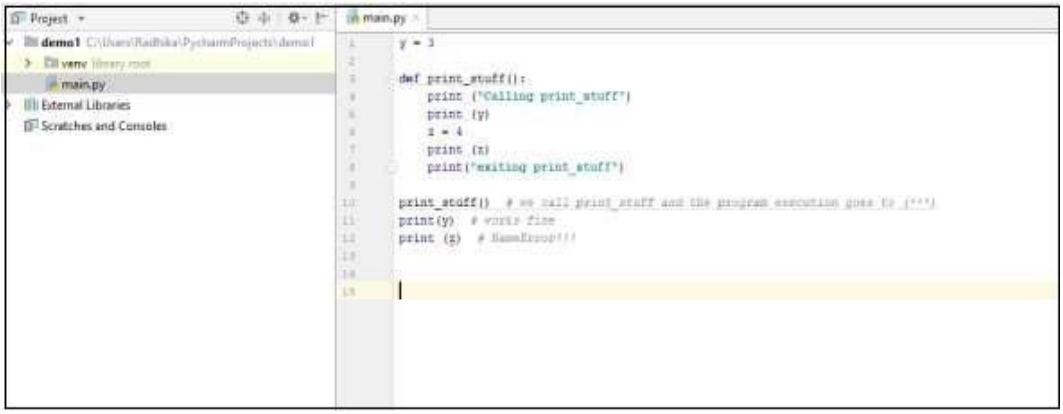
The above snapshot describes the project overview of demo1 and the options to create a new file. Let us create a new file called **main.py**.

The code included in main.py is as follows −

```
y = 3

def print_stuff():
  print ("Calling print_stuff")
  print (y)
  z = 4
  print (z)
  print("exiting print_stuff")

print_stuff() # we call print_stuff and the program execution goes to (***)
print(y) # works fine
print (z) # NameError!!!
```

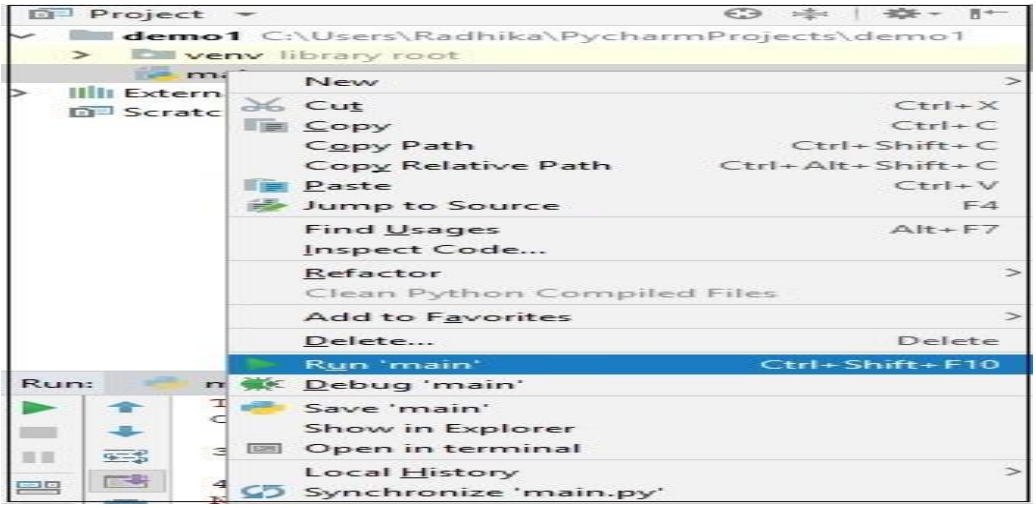
Step-9:

The code created in the file **main.py** using PyCharm Editor is displayed as shown below.



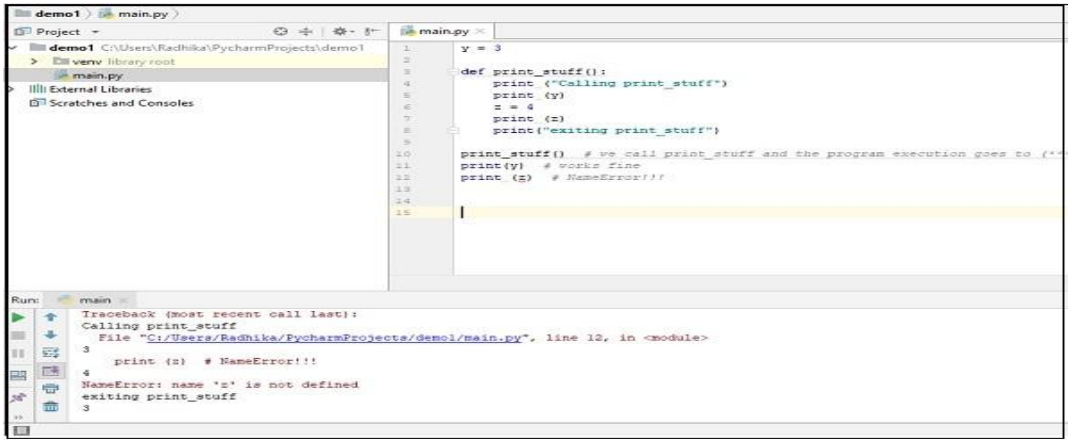
Step-10:

This code can be run within IDE environment. The basic demonstration of running a program.



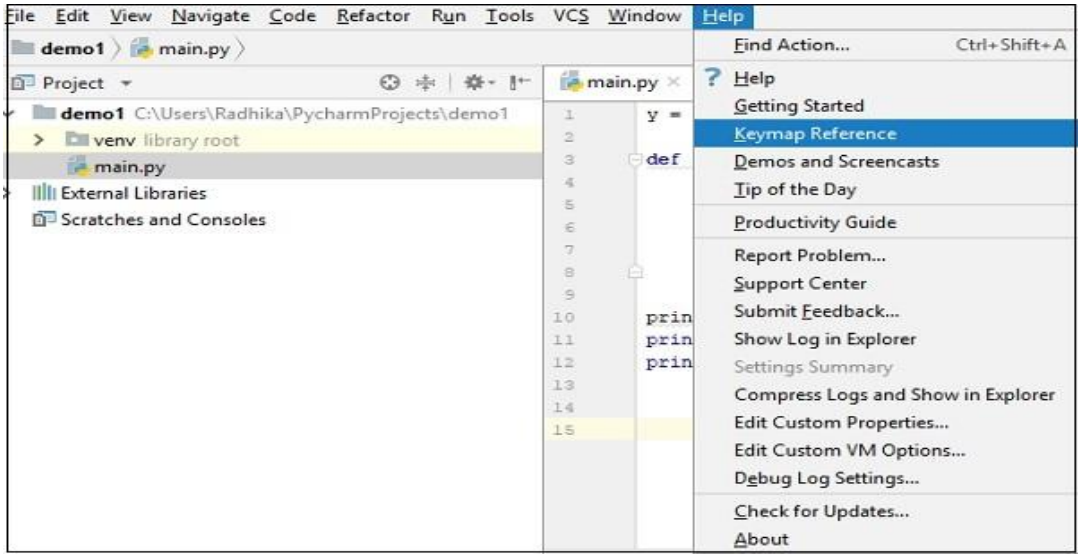
Step-11:

Note that we have included some errors within the specified code such that console can execute the code and display output as the way it is intended to.



Step-12:

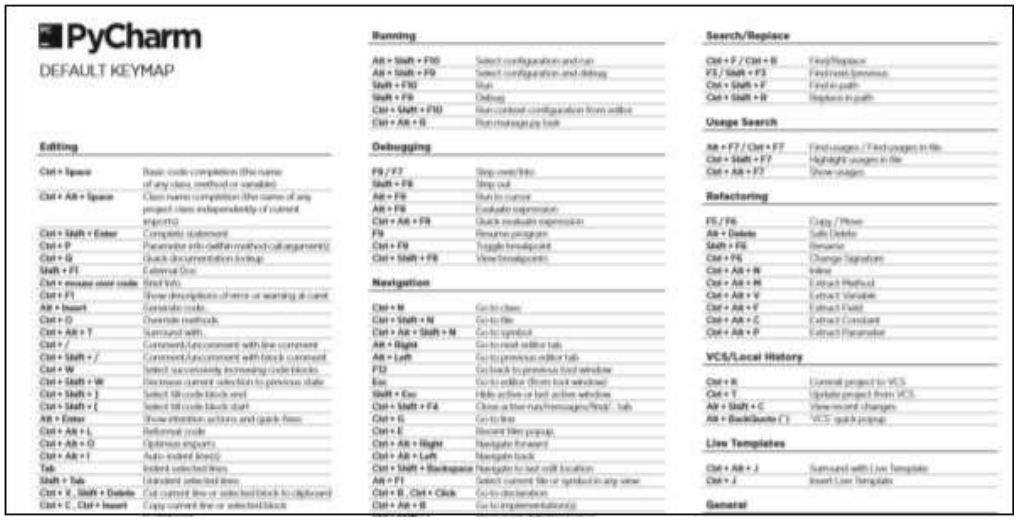
You can find the list of Keymaps available in the file menu **Help -> Keymap Reference** as shown in the screenshot given below −



Step-13:

- You can find the list of Keymaps and the available shortcuts in PDF format as shown below.

- The default Keymap for Windows and Linux operating systems is default, while in Mac OS the default Keymap is OSX 10.5.


PyCharm

DEFAULT KEYMAP

Editing

Ctrl + Space	Basic code completion (the name of any class, method or variable)
Ctrl + Alt + Space	Class name completion (the name of any project class independently of current imports)
Ctrl + Shift + Enter	Complete statement
Ctrl + P	Parameter info (within method call arguments)
Ctrl + Q	Quick documentation lookup
Shift + F1	External Doc
Ctrl + mouse over code	Brief Info
Ctrl + F1	Show descriptions of error or warning at caret
Alt + Insert	Generate code...
Ctrl + O	Override methods...
Ctrl + Alt + T	Surround with...
Ctrl + /	Comment/Uncomment with line comment
Ctrl + Shift + /	Comment/Uncomment with block comment
Ctrl + W	Select successively increasing code blocks
Ctrl + Shift + W	Decrease current selection to previous state
Ctrl + Shift +]	Select till code block end
Ctrl + Shift + [	Select till code block start
Alt + Enter	Show intention actions and quick fixes
Ctrl + Alt + L	Reformat code
Ctrl + Alt + O	Optimize imports
Ctrl + Alt + I	Auto-indent line(s)
Tab	Indent selected lines
Shift + Tab	Unindent selected lines
Ctrl + X, Shift + Delete	Cut current line or selected block to clipboard
Ctrl + C, Ctrl + Insert	Copy current line or selected block

Running

Alt + Shift + F10	Select configuration and run
Alt + Shift + F9	Select configuration and debug
Shift + F10	Run
Shift + F9	Debug
Ctrl + Shift + F10	Run context configuration from editor
Ctrl + Alt + R	Run manage.py task

Debugging

F8 / F7	Step over/into
Shift + F8	Step out
Alt + F9	Run to cursor
Alt + F8	Evaluate expression
Ctrl + Alt + F8	Quick evaluate expression
F9	Resume program
Ctrl + F8	Toggle breakpoint
Ctrl + Shift + F8	View breakpoints

Navigation

Ctrl + N	Go to class
Ctrl + Shift + N	Go to file
Ctrl + Alt + Shift + N	Go to symbol
Alt + Right	Go to next editor tab
Alt + Left	Go to previous editor tab
F12	Go back to previous tool window
Esc	Go to editor (from tool window)
Shift + Esc	Hide active or last active window
Ctrl + Shift + F4	Close active run/messages/find/... tab
Ctrl + G	Go to line
Ctrl + E	Recent files popup
Ctrl + Alt + Right	Navigate forward
Ctrl + Alt + Left	Navigate back
Ctrl + Shift + Backspace	Navigate to last edit location
Alt + F1	Select current file or symbol in any view
Ctrl + B, Ctrl + Click	Go to declaration
Ctrl + Alt + B	Go to implementation(s)

Search/Replace

Ctrl + F / Ctrl + R	Find/Replace
F3 / Shift + F3	Find next/previous
Ctrl + Shift + F	Find in path
Ctrl + Shift + R	Replace in path

Usage Search

Alt + F7 / Ctrl + F7	Find usages / Find usages in file
Ctrl + Shift + F7	Highlight usages in file
Ctrl + Alt + F7	Show usages

Refactoring

F5 / F6	Copy / Move
Alt + Delete	Safe Delete
Shift + F6	Rename
Ctrl + F6	Change Signature
Ctrl + Alt + N	Inline
Ctrl + Alt + M	Extract Method
Ctrl + Alt + V	Extract Variable
Ctrl + Alt + F	Extract Field
Ctrl + Alt + C	Extract Constant
Ctrl + Alt + P	Extract Parameter

VCS/Local History

Ctrl + K	Commit project to VCS
Ctrl + T	Update project from VCS
Alt + Shift + C	View recent changes
Alt + BackQuote (`)	VCS quick popup

Live Templates

Ctrl + Alt + J	Surround with Live Template
Ctrl + J	Insert Live Template

General

Step-14:

- You can also view the list of Keymaps available using the **Settings** option in Windows and Linux Operating system (Preferences in Mac OS) as shown in the screenshot given below −

- The default Keymap includes various sections for Editor Actions, Main Menu, Tool Windows, External tools, Version Control System, Macros, Quick Lists, Plug-ins and Other options as well.

CHAPTER 4
REQUIREMENTS

4.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python is easy programming language. Python contains many number of libraries. It can also used to build the websites, applications etc. Python is easy to Read, Learn,and Write.Python is Productive Language, Interpreted Language, Dynamically Typed , Free and open source, portability, vast libraries support. Python has a simple syntax to the english language. Allows the developers to write the programs with fewer line. It runs on an interpreter system, meaning the code can executed as soon as it is written and prototyping is very quick.Python has become one of the most popular programming languages in the world in recent years. It's used in everything from machine learning to building websites and software testing. It can be used by developers and non-developers alike.

Python is an object-oriented programming language like Java. Python is called an interpreted language. Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages. The standard implementation of python is called “cpython”. It is the default and widely used implementation of Python. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.Python is a popular general-purpose programming language that can be used for a wide variety of applications.

4.2 Pycharm

PyCharm is a hybrid platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE. PyCharm is available in three editions: Community (free and open-sourced): for smart and intelligent Python development, including code assistance, refactorings, visual debugging, and version control integration.PyCharm is available in three editions: Professional, Community, and Edu. The Community and Edu editions are open-source projects and they are free, but they have fewer features. PyCharm Edu provides courses and helps you learn programming with Python.

PyCharm is an Integrated Development Environment (IDE) used for computer programming particularly designed for Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. PyCharm is developed by the Czech company JetBrains. We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

4.2.1 Features of Pycharm

Object-oriented development of a package: PyCharm allows you to create a full-fledged package with classes, subclasses, tests, GUIs, configurations, and the like.

Debugger: PyCharm contains a debugger that allows you to interrogate the behaviour of your code step-by-step while in a notebook you'd have to use the print function to tell you what's going wrong.

Intelligent Code Editor: It helps us write high-quality code. It consists of color schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.It helps identify errors easily.

Assistance for Many Other Web Technologies: It helps developers create web applications in Python.It supports popular web technologies such as HTML, CSS, and JavaScript.Developers have the choice of live editing with this IDE. At the same time, they can preview the created/updated web page.The developers can follow the changes directly on a web browser.

4.3 Hardware Requirements

Hardware requirements which we need in this project are:

1. I3 Processor : P-IV
2. Memory : 2GB RAM(Min)
3. Hard Drive : 40GB
4. Monitor : SVGA 21

4.4 Software Requirements

Software requirements which we need in this project are:

1. Operating System : Windows 7 Ultimate or higher
2. Tools : Pycharm
3. Languages Used : Python
4. Library : Keras, Numpy, Scipy, Scikit, Tensorflow,Tkinter.
5. Front end : Django
6. Back end : Python

CHAPTER 5

DESIGN

5.1 Introduction

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

5.2 Why Use UML in Projects?

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modeling Language (UML) was designed to respond to these needs. Simply, Systems design refers to the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements which can be done easily through UML diagrams. In this project two basic UML diagrams have been explained

1. Use Case Diagrams
2. Sequence Diagrams

5.3 Use case Diagram

In software engineering, a use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use case analysis. Its purpose is to

present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behaviour of the system from the external point of view. The actors are outside the boundary of the system, whereas the use cases are inside the boundary of the system.In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent: Scenarios in which your system or application interacts with people, organizations, or external systems. Goals that your system or application helps those entities (known as actors) achieve.

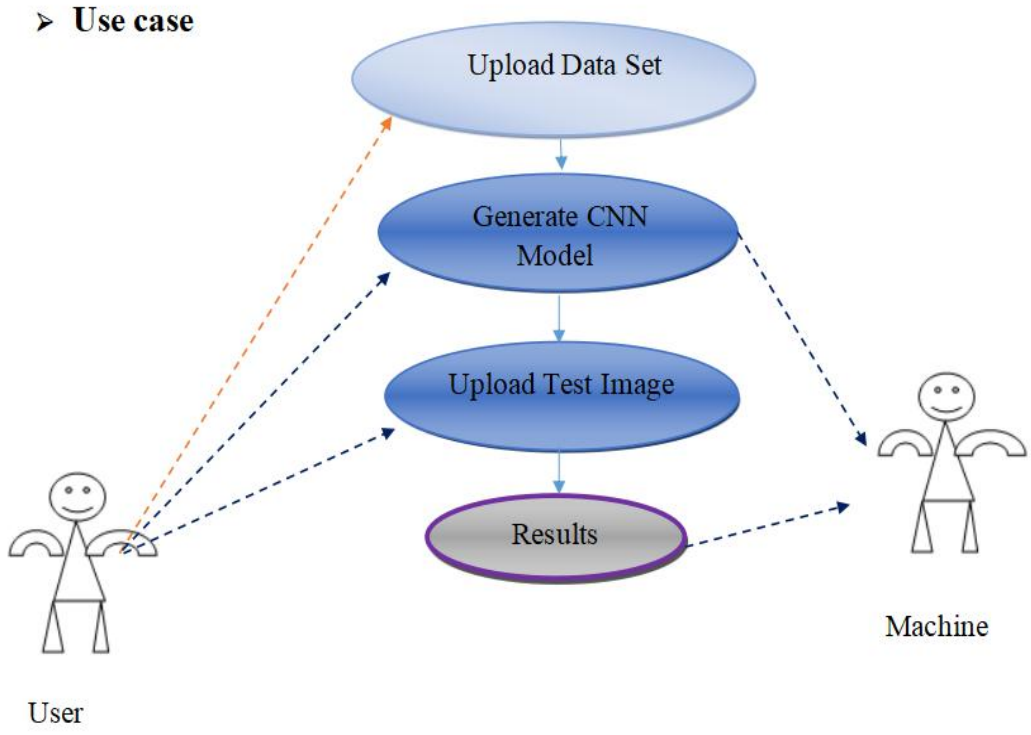


Fig.5.3:Use Case Diagram

5.4 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or

to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.A sequence diagram is the most commonly used interaction used interaction diagram. Interaction diagram – An interaction diagram is used to show the interactive behaviour of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system. Sequence diagram – A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

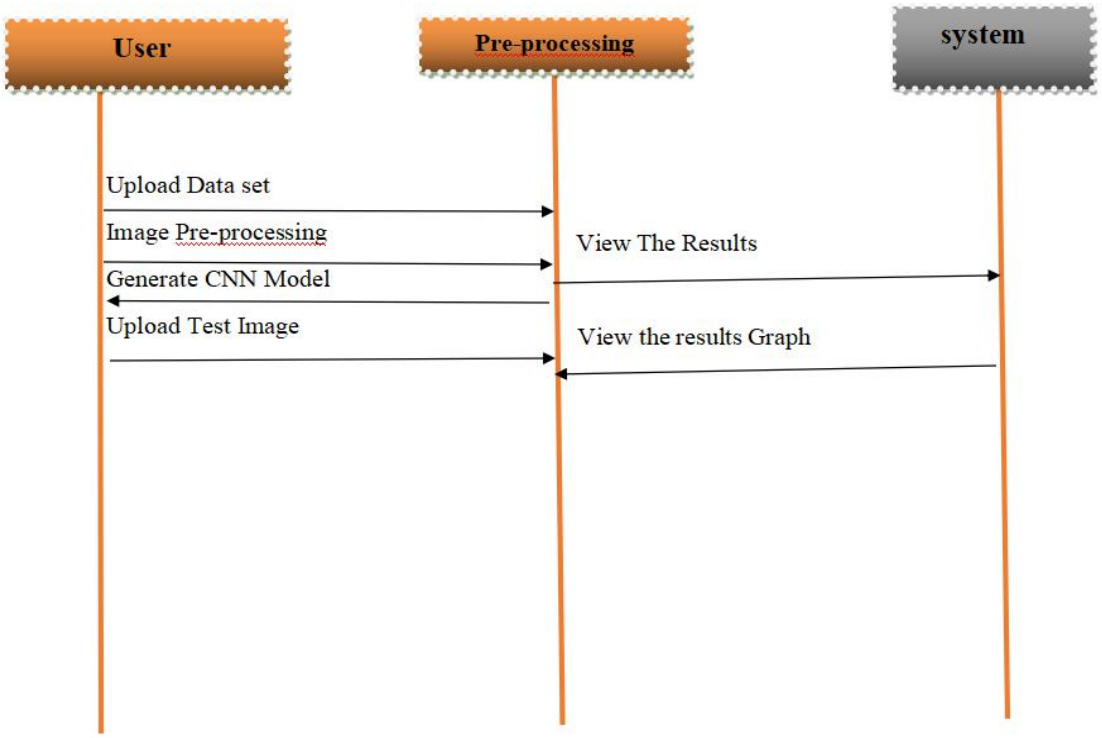


Fig.5.4:Sequence Diagram

CHAPTER-6

IMPLEMENTATION AND RESULTS

6.1 Introduction

In recent years a lot of fake currency note is being printed which have caused great loss and damage towards society. So, it has become a necessity to develop a tool to detect fake currency. This project proposes an approach that will detect fake currency note being circulated in our country by using their image. Our project will provide required mobility and compatibility to most peoples as well as credible accuracy for the fake currency detection.

6.2 Technology Used

6.2.1 Machine Learning:

Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artifical intelligence. Machine learning algorithms build a mathematical modell based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analysis.

Machine learning uses data to detect various patterns in a given data set.

1.It can learn from past data and improve automatically.

2.It is a data-driven technology.

3.It is much similar to data mining as it also deals with the huge amount of the data.

6.2.2 Deep Learning :

Deep Learning is a type of machine learning and artifical intelligence(AI)that imitates the way humans gain certain types of knowledge. Deep Learning is an important element of data science, which includes statistics and predictive modelling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing, and interpreting large amounts of data; deep learning makes this process faster and easier. Deep learning algorithms run data through several “layers” of neural network algorithms, each of which passes a simplified representation of the data to the next layer.The ability to process large numbers of features makes deep learning very powerful when dealing with unstructured data. However, deep learning algorithms can be overkill for less complex problems because they require access to a vast amount of data to be effective. For instance, Image Net, the common benchmark for training deep learning models for comprehensive image recognition, has access to over 14 million images.

Deep learning algorithms learn progressively more about the image as it goes through each neural network layer. Early layers learn how to detect low-level features like edges, and subsequent layers combine features from earlier layers into a more holistic representation. For example, a middle layer might identify edges to detect parts of an object in the photo such as a leg or a branch, while a deep layer will detect the full object such as a dog or a tree.If the data is too simple or incomplete, it is very easy for a deep learning model to become over fitted and fail to generalize well to new data. As a result, deep learning models are not as effective as other techniques (such as boosted decision trees or linear models) for most practical business problems such as understanding customer churn, detecting fraudulent transactions and other cases with smaller data sets and fewer features. In certain cases like multi class classification, deep learning can work for smaller, structured data sets.

6.3 Algorithms Used

6.3.1 CNN (CONVOLUTIONAL NEURAL NETWORK):

Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with

enough training, ConvNets have the ability to learn these filters/characteristics.Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

- Convolutional layer
- Pooling layer
- Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object. ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

6.3.2 SVM(Support Vector Machine) Algorithm:

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

6.4 Libraries Used

6.4.1 Keras :

Keras is a deep learning API written in Python, running on top of the machine learning platform Tensor Flow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras is:

- **Simple** -- but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.
- **Flexible** -- Keras adopts the principle of progressive disclosure of complexity: simple work flows should be quick and easy, while arbitrarily advanced work flows should be possible via a clear path that builds upon what you've already learned.
- **Powerful** -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.
- These are the some of the keys points for the keras.

6.4.2 SciPy:

SciPy, a scientific library for Python is an open source, BSD-licensed library for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The main reason for building the SciPy library is that, it should work with NumPy arrays. It provides many user-friendly and efficient numerical practices such as routines for numerical integration and optimization.

6.4.3 Scikit learn :

Scikit - learn or Sklearn is the most useful and robust library for machine learning in

Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

6.4.4 Tensor Board :

Tensor Board is a tool for providing the measurements and visualizations needed during the machine learning work flow. It enables tracking experiment metrics like loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more.

6.4.5 Tensor Flow :

Tensor Flow provides a collection of work flows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser, or on-device no matter what language you use. Load & pre process data. Build, train & reuse models, deploy.

6.4.6 NumPy :

NumPy is one of the most commonly used packages for scientific computing in Python. It provides a multidimensional array object, as well as variations such as masks and matrices, which can be used for various math operations.NumPy contains **a** multi-dimensional array and matrix data structures. It can be utilised to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines. Therefore, the library contains a large number of mathematical, algebraic, and transformation functions.

6.4.7 Matplotlib :

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

6.4.8 Pillow :

The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation. Pillow module allows you to pull some statistics data out of image using histogram method, which later can be used for statistical analysis and automatic contrast enhancement.

6.5 Django:

6.5.1 Introduction

Django is a Web Application Framework which is used to develop web applications. Our Django Tutorial includes all topics of Django such as introduction, features, installation, environment setup, admin interface, cookie, form validation, Model, Template Engine, Migration, MVT etc. All the topics are explained in detail so that reader can get enough knowledge of Django. Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line: **The web framework for perfectionists with deadlines.**

By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains its development and release cycle.

Django was released on 21, July 2005. Its current stable version is 2.0.3 which was released on 6 March, 2018.

Django is widely accepted and used by various well-known sites such as:

- Instagram
- Mozilla
- Disqus
- Pinterest
- Bitbucket
- The Washington Times

6.4.2 Features of Django

- Rapid Development

- Secure
- Scalable
- Fully loaded
- Versatile
- Open Source
- Vast and Supported Community

6.6 Input and Output Screen Design (Screens)

6.6.1 Back End

Step-1:

Importing the required libraries to the project they are Keras, SciPy, Scikit learn, Tensor board, Tensor Flow, NumPy, Matplotlib, Pillow.

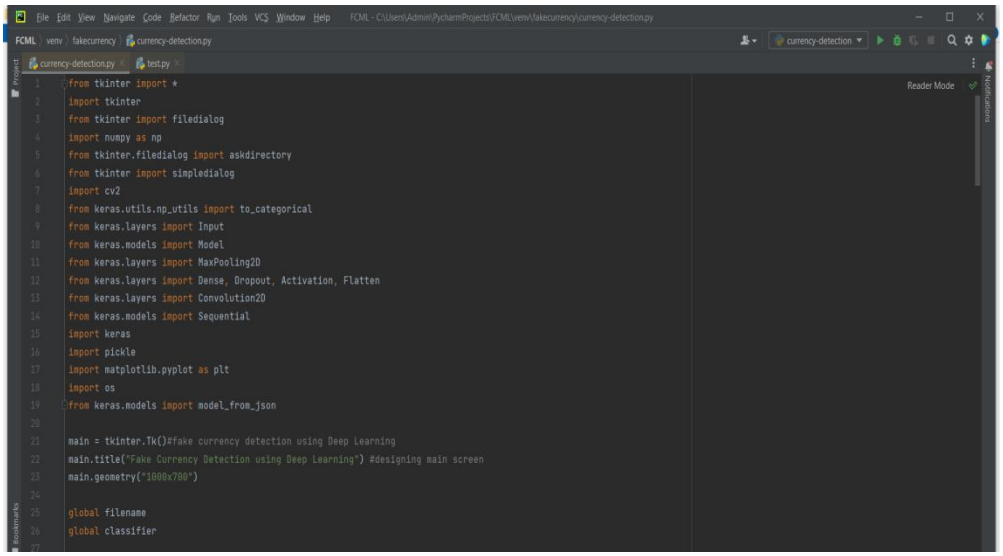


Fig.6.6.1.1:Importing Libraries

Step-2 :

Uploading the data set when the data set is uploaded, it shows the data set is loaded.

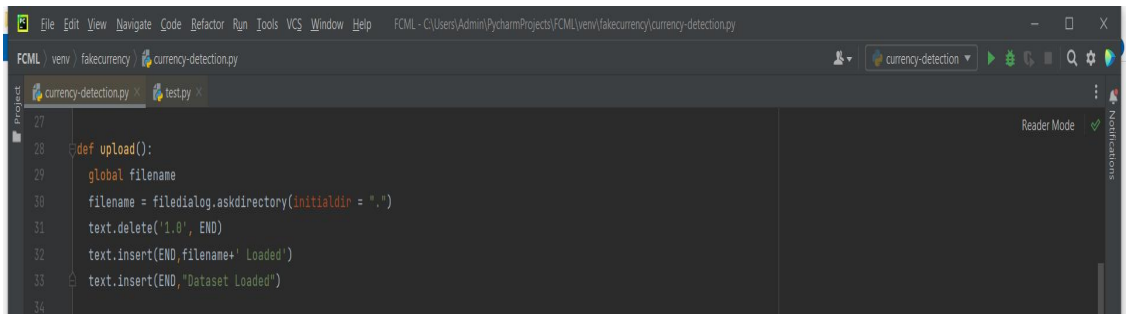


Fig.6.6.1.2: Uploading Data Set

Step-3 :

Code to Generate CNN Model which gives the accuracy rate of our trained data set.

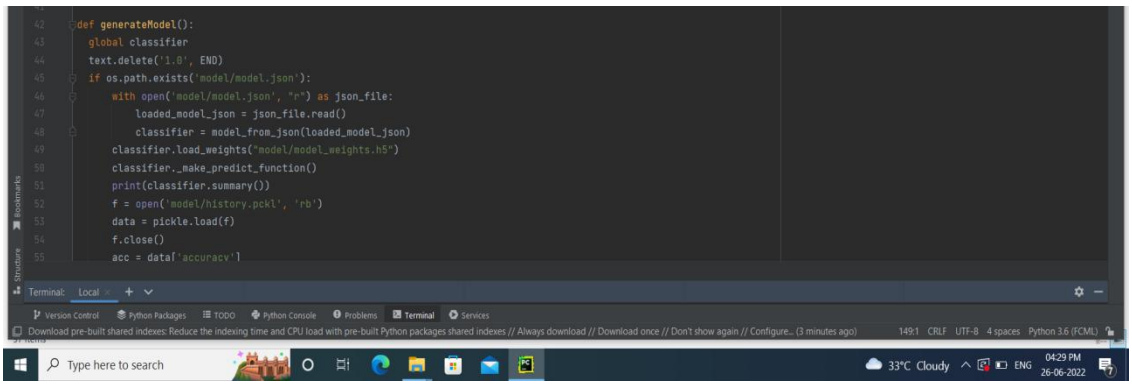


Fig.6.6.1.3: Generated CNN Model Code(a)

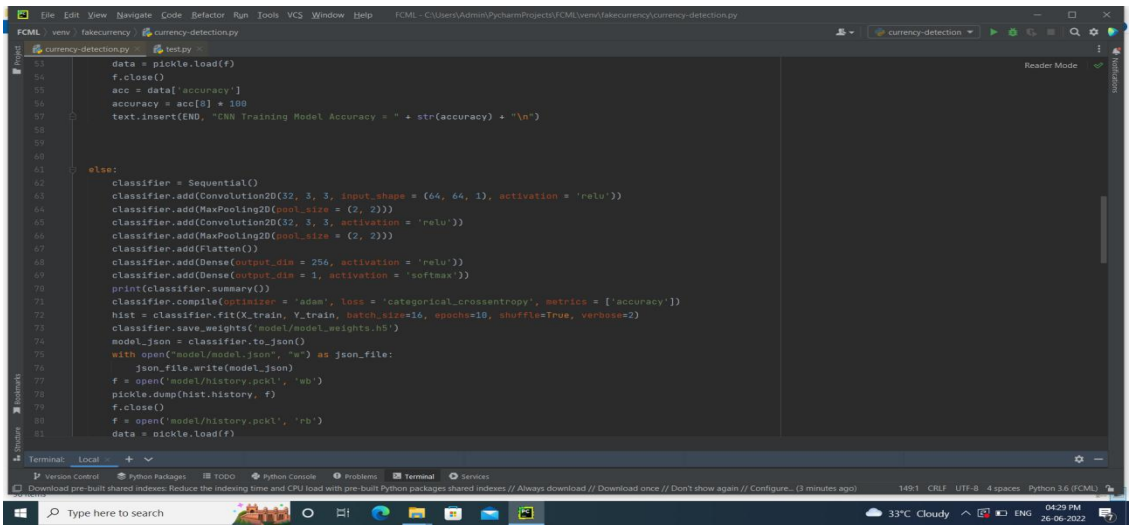


Fig.6.6.1.3: Generated CNN Model Code(b)

Fig.6.6.1.3:Generating CNN model Code(c)

Step-4:

The below figure is code for prediction, It predicts whether the currency note is real or fake. If predict is 0 the currency note is Fake otherwise Real.

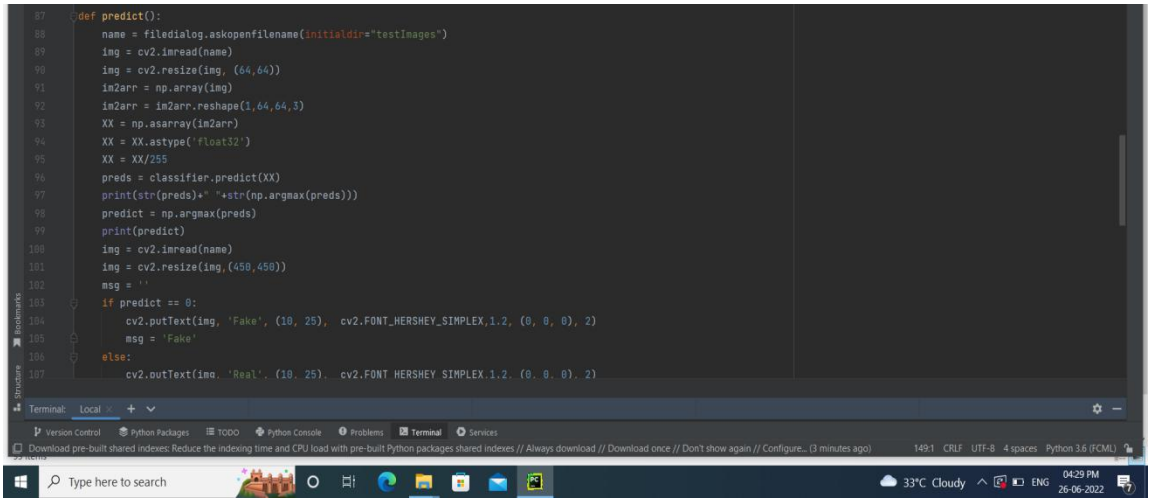


Fig.6.6.1.4: Prediction Code(a)

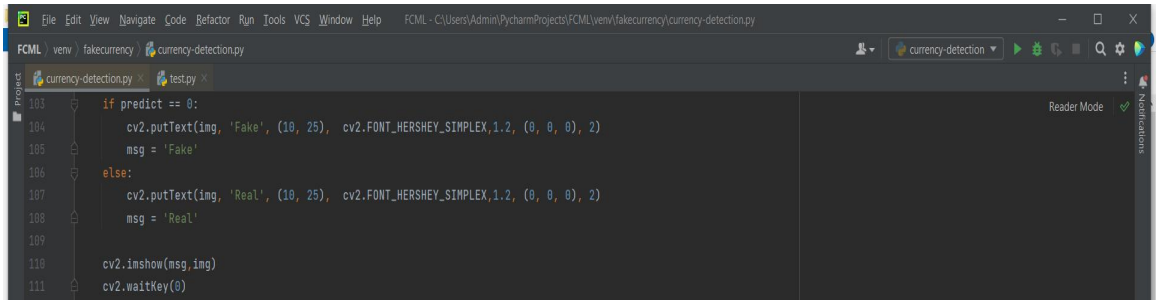
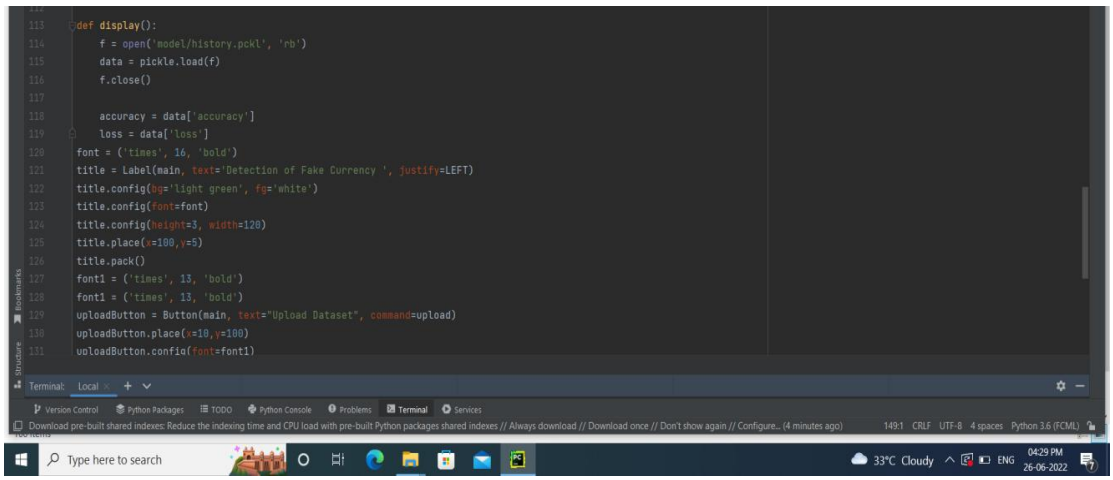


Fig.6.6.1.4: Prediction Code(b)

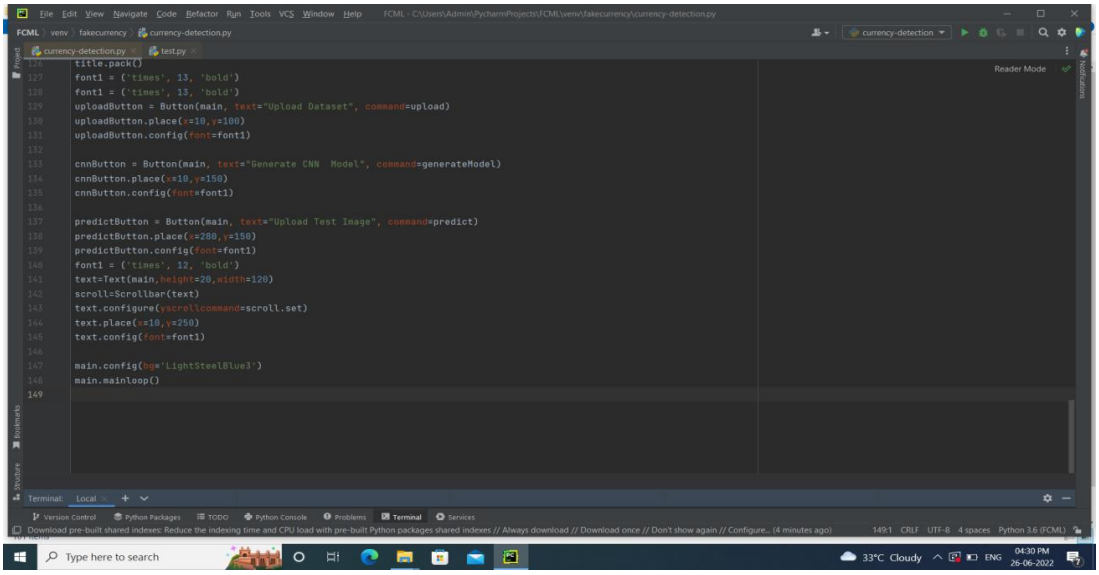
Step-5:

The below figures are the code to display the buttons on the screen the following are the buttons, they are Upload data set, Generate CNN model, Upload Test Image.



```
def display():
    f = open('model/history.pckl', 'rb')
    data = pickle.load(f)
    f.close()

    accuracy = data['accuracy']
    loss = data['loss']
font = ('times', 16, 'bold')
title = Label(main, text='Detection of Fake Currency ', justify=LEFT)
title.config(bg='light green', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=100,y=5)
title.pack()
font1 = ('times', 13, 'bold')
font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Upload Dataset", command=upload)
uploadButton.place(x=10,y=100)
uploadButton.config(font=font1)
```

Fig.6.6.1.5: Display Code(a)

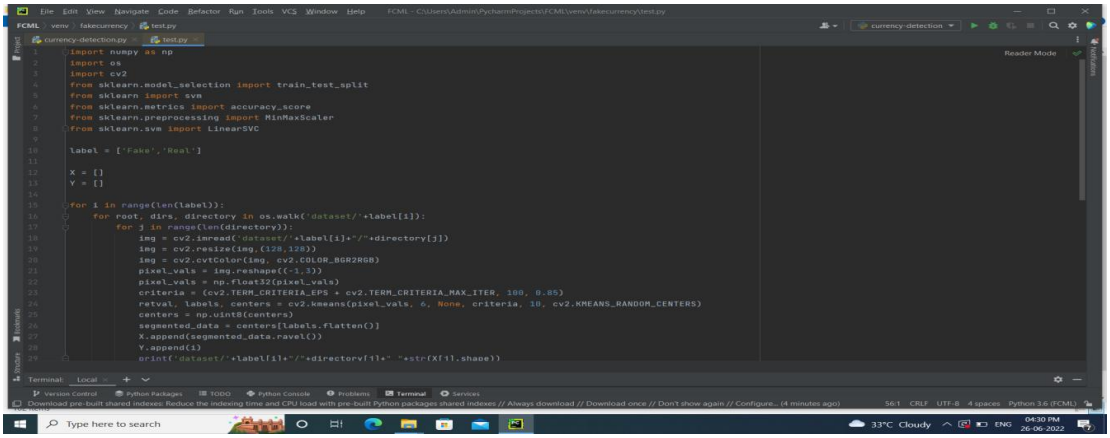


```
title.pack()
font1 = ('times', 13, 'bold')
font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Upload Dataset", command=upload)
uploadButton.place(x=10,y=100)
uploadButton.config(font=font1)

cnnButton = Button(main, text="Generate CNN  Model", command=generateModel)
cnnButton.place(x=10,y=150)
cnnButton.config(font=font1)

predictButton = Button(main, text="Upload Test Image", command=predict)
predictButton.place(x=280,y=150)
predictButton.config(font=font1)
font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=120)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=250)
text.config(font=font1)

main.config(bg='LightSteelBlue3')
main.mainloop()
```

Fig .6.6.1.5: Display Code(b)

Step -6 :

Code to check whether the Note is FAKE or REAL.



```
import numpy as np
import os
import cv2
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import LinearSVC

label = ['Fake','Real']

X = []
Y = []

for i in range(len(label)):
    for root, dirs, directory in os.walk('dataset/'+label[i]):
        for j in range(len(directory)):
            img = cv2.imread('dataset/'+label[i]+'/'+directory[j])
            img = cv2.resize(img,(128,128))
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            pixel_vals = img.reshape((-1,3))
            pixel_vals = np.float32(pixel_vals)
            criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.85)
            retval, labels, centers = cv2.kmeans(pixel_vals, 6, None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS)
            centers = np.uint8(centers)
            segmented_data = centers[labels.flatten()]
            X.append(segmented_data.ravel())
            Y.append(i)
            print('dataset/'+label[i]+'/'+directory[j]+" "+str(X[j].shape))
```

Fig.6.6.1.6: Detection of Fake or Real Note Code(a)

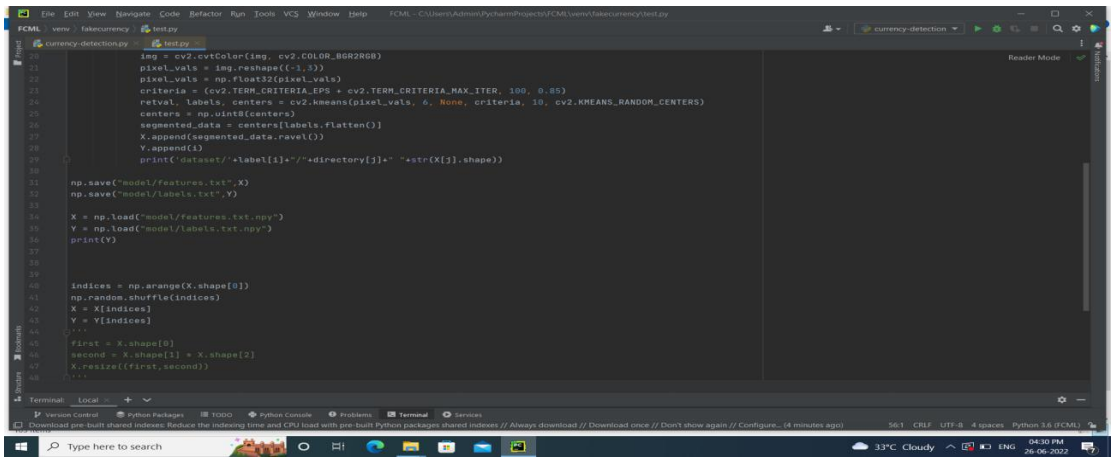


Fig.6.6.1.6: Detection of Fake or Real Note Code(b)

6.6.2 Front End

The below is displayed when we execute the program, the detection of fake currency screen is displayed it contains three buttons they are Upload Data set, Generate CNN Model, Upload Image.

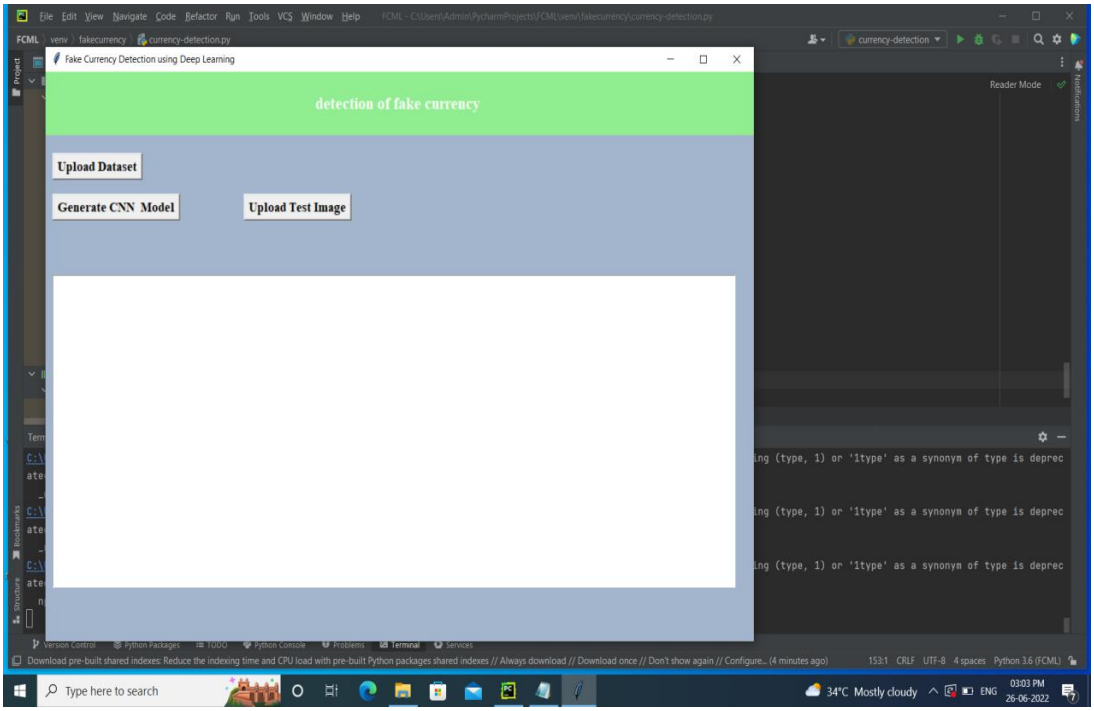


Fig.6.6.2.1: Front End Page

Step-1 : Uploading Data set

When we click on the Upload data set button from the screen, it redirect to the path where we contain the our data set, in that select the folder which contains the data set after selecting folder click on the Select Folder.

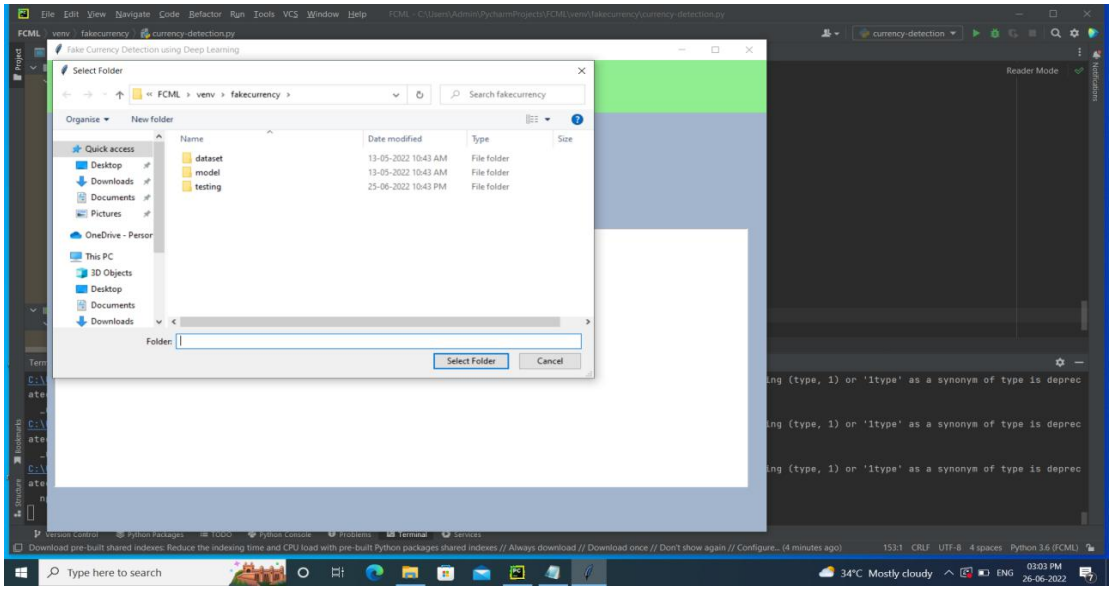


Fig.6.6.2.2:Data set Folder

After selecting the folder, the page shows that the Data set is loaded and also shows the path of that selected data set

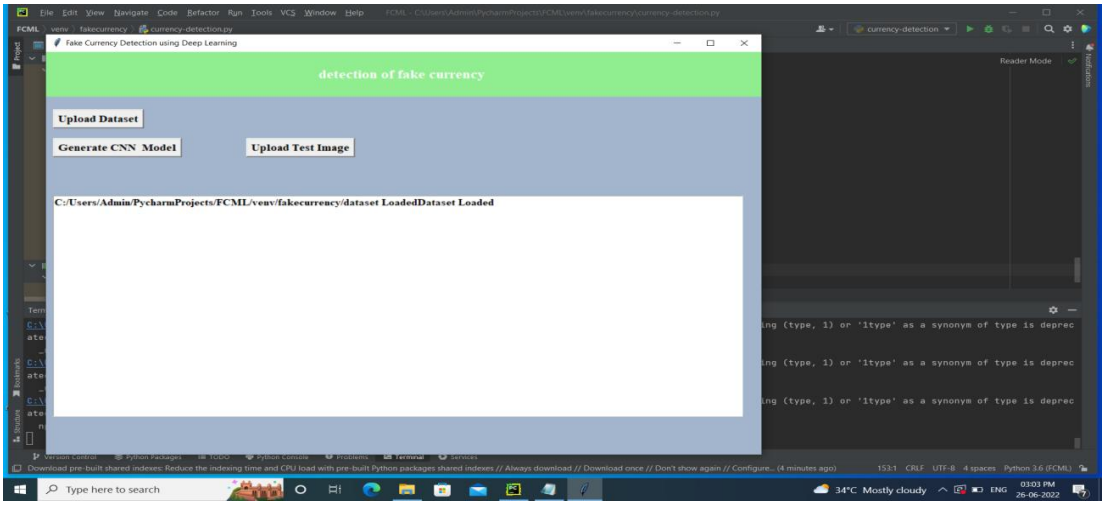


Fig.6.6.2.3:Data Set Loaded

Step-2 : Generate CNN Model

Next button is Generate CNN Model, by clicking that button we get the Accuracy of our CNN Training Model.

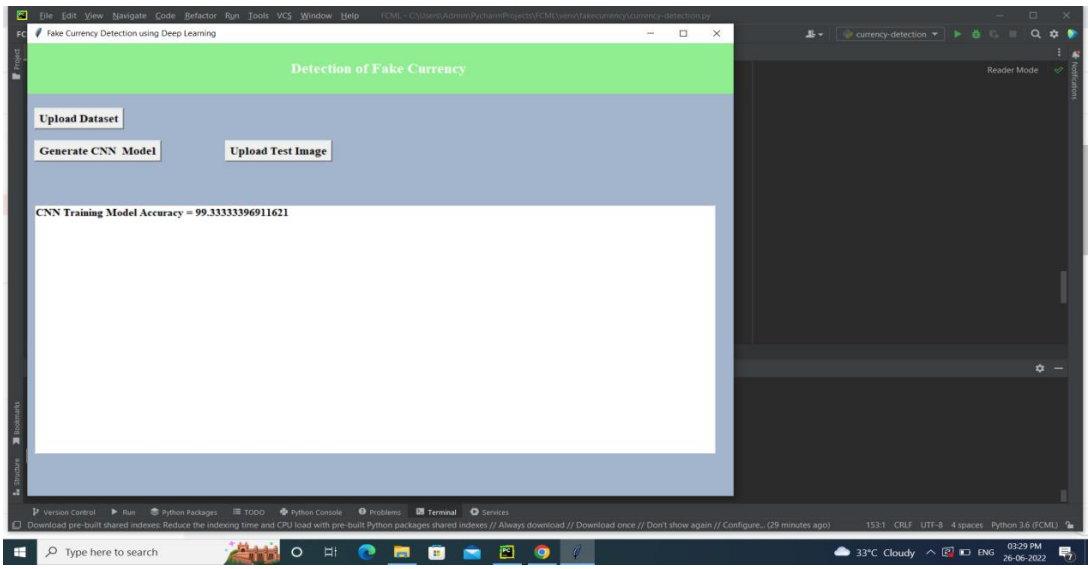


Fig.6.6.2.4: Generate CNN Model

Step -3:

Next button is Upload test Image, when we click on that button it redirect to the folder where the test images are present, in that select any one of the image and click on open.

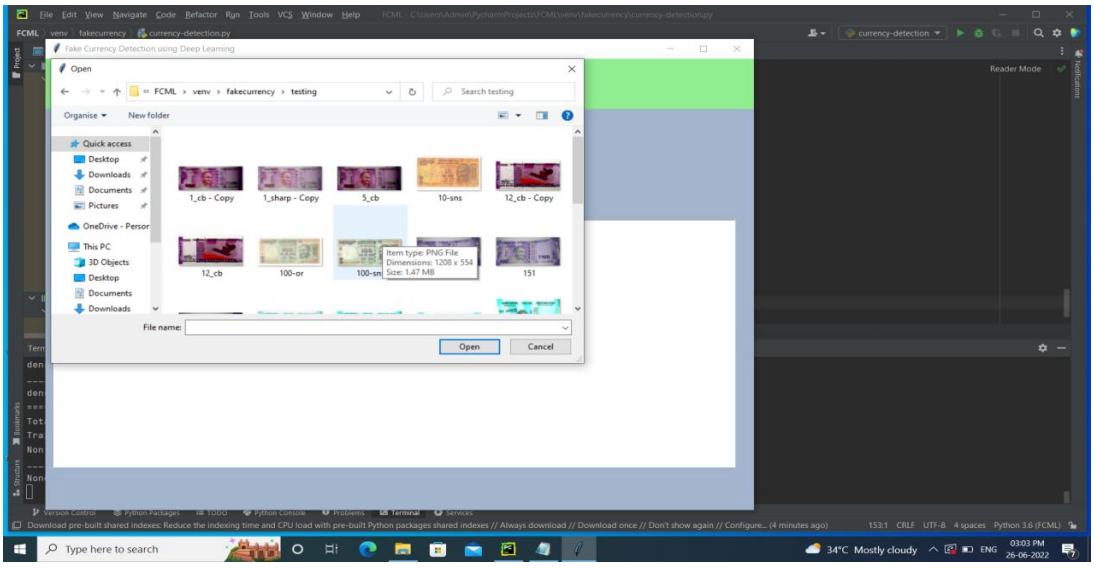


Fig.6.6.2.5: Test Images

Select the image and click on open, it shows if that is Fake Note or Real Note.

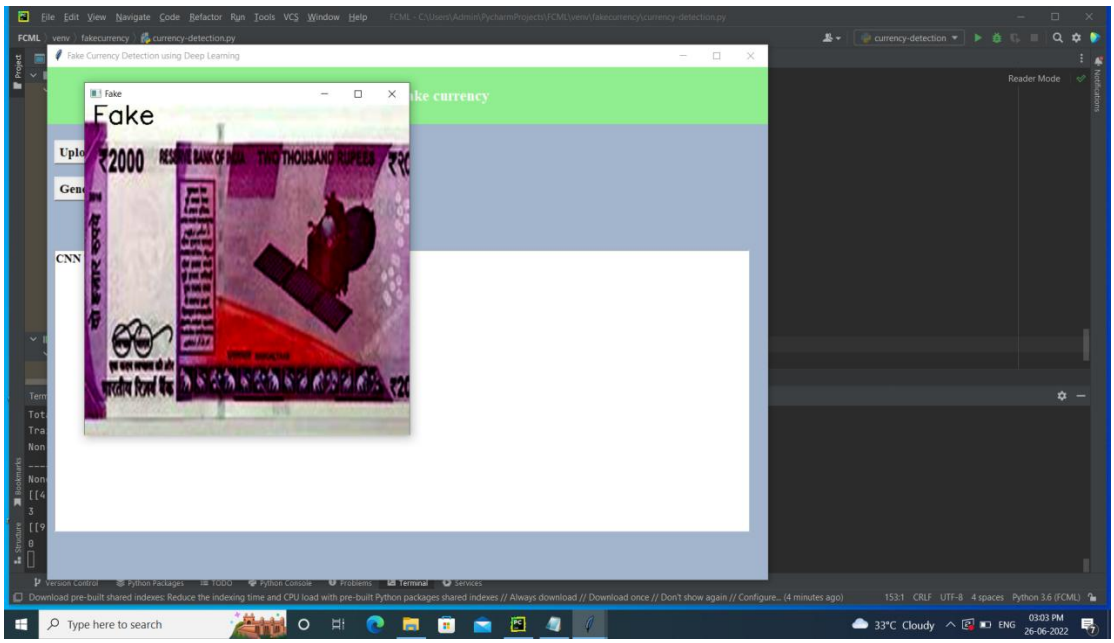


Fig.6.6.2.6: Fake Note

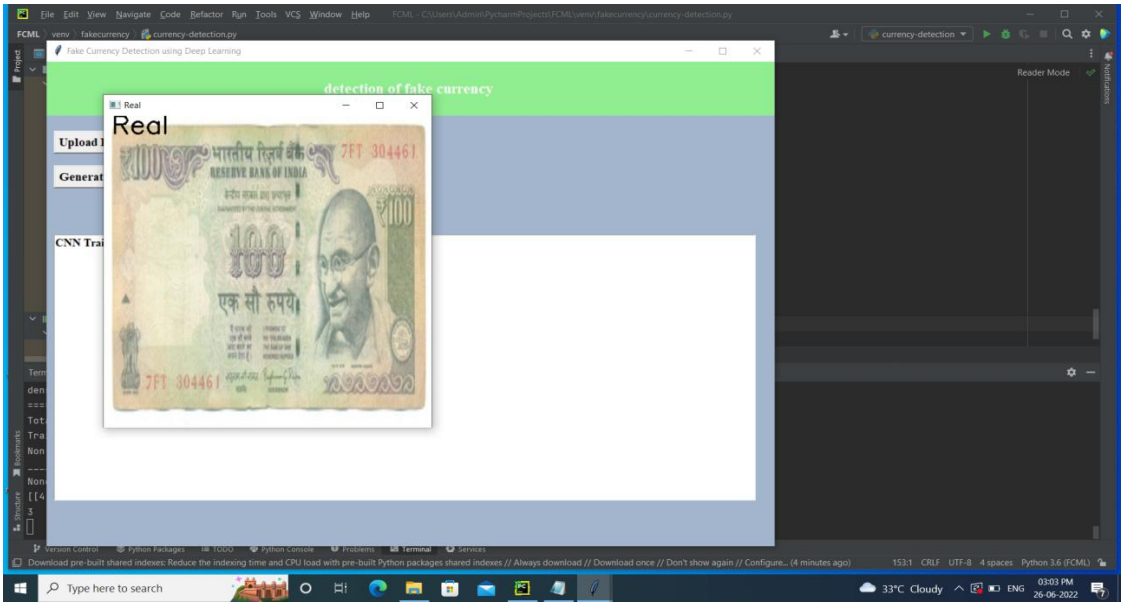


Fig.6.6.2.7: Real Note

CHAPTER-7

TESTING

7.1 Introduction

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

7.2 Methodology used for Testing

The completion of a system will be achieved only after it has been thoroughly tested. Though this gives a feel the project is completed, there cannot be any project without going through this stage. Hence in this stage it is decided whether the project can undergo the real time environment execution without any break downs, therefore a package can be rejected even at this stage.

- Testing methods Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

- Black box testing - Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

- White box testing - White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms. White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the

software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

- Grey Box Testing - Grey box testing involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output do not qualify as "grey box," because the input and output are clearly outside of the "black-box" that we are calling the system under test. This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test. Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

- Acceptance testing - Acceptance testing can mean one of two things: o A smoke test is used as an acceptance test prior to introducing a build to the main testing process. o Acceptance testing performed by the customer is known as user acceptance testing (UAT).

- Regression Testing - Regression testing is any type of software testing that seeks to uncover software regressions. Such regression occurs whenever software functionality that was previously working correctly stops working as intended. Typically, regressions occur as an unintended consequence of program changes. Common methods of regression testing include re-running previously run tests and checking whether previously fixed faults have reemerged.

- Non Functional Software Testing – Performance testing checks to see if the software can handle large quantities of data or users. This is generally referred to as software scalability. This activity of Non-Functional Software Testing is often times referred to as Load Testing. Stability testing checks to see if the software can continuously function well in or above an acceptable period. This activity of NonFunctional Software Testing is often times referred to as indurations test. Usability testing is needed to check if the user interface is easy to use and understand. Security testing is essential for software which

processes confidential data and to prevent system intrusion by hackers.

7.3 Software Testing Strategies

A software testing strategy provides a road map for the software developer. Testing is a set of activities that can be planned in advance and conducted systematically. For this reason, a template for software testing a set of steps into which we can place specific test case design methods should be defined for software engineering process. Any software testing strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer-based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.

7.4 Unit Testing

Unit testing focuses verification efforts in smallest unit of software design (module).

- Unit test considerations
- Unit test procedures

7.5 Integration Testing

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. There are two types of integration testing:

- Top-Down Integration: Top down integration is an incremental approach to construction of program structures. Modules are integrated by moving down wards throw the control hierarchy beginning with the main control module.

- Bottom-Up Integration: Bottom up integration as its name implies, begins construction and testing with automatic modules.
- Regression Testing: In this contest of an integration test strategy, regression testing is the re execution of some subset of test that have already been conducted to ensure that changes have not propagate unintended side effects.

7.6 Summary

This software has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly, and has required options, which can be utilized by the user to perform the desired operations. Application software meets the information requirements specified to a great extent. The system has been designed keeping in view the present and future requirements in mind and made very flexible. The goals that are achieved by the software are instant access, improved productivity, Optimum utilization of resources, and efficient management of recur.

CONCLUSION

Machine learning has gained tremendous success in image classification tasks. Our architecture which is based on Deep CNN works as feature extractor eliminating the need to apply image processing technique and manually checking the presence of security features in the note. The generated dataset has successfully helped conduct experiments and tried to mimic the real-world scenario. The application built will be useful to any common person to detect a counterfeit note. Future scope includes trying out new Deep CNN architectures to increase the accuracy of the model. Increasing the data-set, so that the model gets trained better and produce better results.

REFERENCES

Journal Article

[1]. Tusher Agasti, Gajanan Burand, Prathik Wade and P Chitra "Fake Currency Detection Using Image Processing.pdf" , School Of Electronics Engineering, VIT University, Vellore632014, Tamil Nadu,India.

Journal Article

[2]. Cesar G.Pachon, Dora M. Ballesteros and Diego Renza "Fake Banknote Recognition Using Deep Learning.pdf" , Faculty of Engineering , Universidad Militar Nueva Granada , Bogota 110111, Colombia.