

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgavi-590 018, Karnataka, India



A PROJECT REPORT

On

“FAKE CURRENCY DETECTION USING DEEP LEARNING”

Submitted in Partial Fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

G V KEERTHANA

USN: 1SJ18CS033

GEETHA A

USN: 1SJ18CS036

GUTTA SAI SANDHYA

USN: 1SJ18CS037

KUSUMA K N

USN: 1SJ18CS050

Carried out
B G S R&D Centre,
Dept of CSE,
SJCIT

Under the guidance of
Prof. CHANDANA K R BE, M Tech
Assistant Professor
Dept. Of CSE, SJCIT



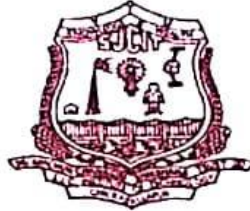
S J C INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHIKKABALLAPUR-562101

2021-2022

||Jai Sri Gurudev||

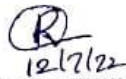
Sri Adichunchanagiri Shikshana Trust®

S.J.C INSTITUTE OF TECHNOLOGY, CHICKBALLAPUR – 562101
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

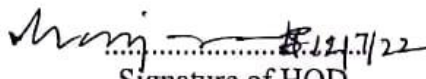


CERTIFICATE

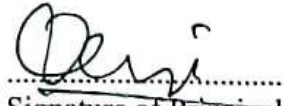
This is to certify that the project work entitled “FAKE CURRENCY DETECTION USING DEEP LEARNING” is a bonafied work carried out by **G V KEERTHANA(1SJ18CS033)**, **GEETHA A(1SJ18CS036)**, **GUTTA SAI SANDHYA(1SJ18CS037)**, **KUSUMA K N (1SJ18CS050)** in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during the year **2021-2022**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements with respect to project work prescribed for the Bachelor of Engineering degree.


12/7/22

Signature of Guide
Prof. Chandana K R
Assistant Professor
Dept. of CSE, SJCIT


12/7/22

Signature of HOD
Dr. Manjunatha Kumar B H
Professor & HOD,
Dept. of CSE, SJCIT
Professor & HOD,
Department of Computer Science & Engg.,
S.J.C. Institute of Technology,
Chickballapur-562 101



Signature of Principal
Dr. G T Raju
Principal, SJCIT
Principal
S J C Institute of Technology
Chickballapur - 562 101

External Examiners:
Name of the Examiners

Signature with Date

1.....

.....

2.....

.....

DECLARATION

We G V Keerthana (1SJ18CS033), Geetha A (1SJ18CS036), Gutta Sai Sandhya (1SJ18CS037), Kusuma K N(1SJ18CS050) Students of VIII semester B.E in Computer Science and Engineering at S J C Institute of Technology, Chickballapur, hereby declare that this dissertation work entitled “FAKE CURRENCY DETECTION USING DEEP LEARNING” has been carried out at B.G.S R&D Centre, Dept. of CSE, SJCIT under the guidance of guide Prof. Chandana K R ,Assistant Professor , Dept. of CSE, SJC Institute of Technology, Chickballapur and submitted in the partial fulfilment for the award of degree Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2021-2022. We further declare that the report had not been submitted to another university for the award of any other degree.

Place: Chikkaballapur

Date:

**G V Keerthana
1SJ18CS033**

**Geetha A
1SJ18CS036**

**Gutta Sai Sandhya
1SJ18CS037**

**Kusuma K N
1SJ18CS050**

ABSTRACT

Fake currency is the money produced without the approval of the government, creation of it is considered as a great Offence. The elevation of colour printing technology has increased the rate of fake currency note printing on a very large scale. Years before, 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. This results in the issue of fake notes instead of the genuine ones has been increased very largely. It is the biggest problem faced by many countries including India. Though Banks and other large organizations have installed Automatic machines to detect fake currency notes, it is really difficult for an average person to distinguish between the two.

ACKNOWLEDGEMENT

With reverential pranam, we express my sincere gratitude and salutations to the feet of his holiness **Paramapoojya Jagadguru Byravaikya Padmabhushana Sri Sri Sri Dr. Balagangadharanatha Maha Swamiji**, his holiness **Paramapoojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji**, and **Sri Sri Mangalnath Swamiji** , Sri Adichunchanagiri Mutt for their unlimited blessings.

First and foremost we wish to express our deep sincere feelings of gratitude to our institution, **Sri Jagadguru Chandrashekaranaatha Swamiji Institute of Technology**, for providing us an opportunity for completing the Final Project successfully.

We extend deep sense of sincere gratitude to **Dr. G T Raju, Principal, S J C Institute of Technology, Chickballapur**, for providing an opportunity to complete the Final Project.

We extend special in-depth, heartfelt, and sincere gratitude to HOD **Dr. Manjunatha Kumar B H, Head of the Department, Computer Science and Engineering, S J C Institute of Technology, Chickballapur**, for his constant support and valuable guidance of the Final Project.

We convey our sincere thanks to our project Guide **Prof. Chanadana K R, Assistant Professor, Department of Computer Science and Engineering, S J C Institute of Technology**, for her constant support, valuable guidance and suggestions of the Final Project.

We also feel immense pleasure to express deep and profound gratitude to Technical Seminar Co-ordinators **Prof. PradeepKumar G M and Prof. Shrihari MR, Assistant Professors, Department of Computer Science and Engineering, S J C Institute of Technology**, for their guidance and suggestions of the Final Project.

Finally, we would like to thank all faculty members of Department of Computer Science and Engineering, S J C Institute of Technology, Chickaballapur for their support. We also thank all those who extended their support and co-operation while bringing out this Final Project

G V Keerthana(1SJ18CS033)

Geetha A(1SJ18CS036)

Gutta Sai Sandhya(1SJ18CS037)

Kusuma K N(1SJ18CS050)

CONTENTS

Declaration	i
Abstract	ii
Acknowledgement	iii
Contents	iv
List of Figures	vi
List of Tables	vii

Chapter No	Chapter Title	Page No
1	INTRODUCTION	01-06
	1.1 Overview	01
	1.2 Problem Statement	01
	1.3 Significance and Relevance of Work	02
	1.4 Objectives	02
	1.5 Methodology	02
	1.6 Organization of the Report	06
2	LITERATURE SURVEY	07-10
3	SYSTEM REQUIREMENTS AND SPECIFICATION	11-17
	3.1 System Requirement Specification	11
	3.1.1 Hardware Specification	11
	3.1.2 Software Specification	11
	3.2 Functional Requirements	17
	3.3 Non-Functional Requirements	17
4	SYSTEM ANALYSIS	18-19
	4.1 Existing System	18
	4.1.1 Limitation	18
	4.2 Proposed System	19
	4.2.1 Advantages	19
5	SYSTEM DESIGN	20-23
	5.1 Activity Diagram	20
	5.2 Use Case Diagram	21

	5.3 Data flow Diagram	22
	5.4 Sequence Diagram	23
6	IMPLEMENTATION	24-30
	6.1 Algorithm/Pseudo code module wise	25
7	TESTING	31-36
	7.1 Methods of Testing	31
	7.1.1 Unit Testing	32
	7.1.2 Validation Testing	33
	7.1.3 Integration Testing	33
	7.1.4 User Acceptance Testing	34
	7.2 Test Cases	34
8	PERFORMANCE ANALYSIS	37
9	CONCLUSION & FUTURE ENHANCEMENT	39
	BIBLIOGRAPHY	39
	APPENDIX	41
	Appendix A: Screen Shots	41
	Appendix B: Abbreviations	46
	PAPER PUBLICATION DETAILS	47

LIST OF FIGURES

Figure No.	Name of the Figure	Page No.
Figure 1.1	Methodology	02
Figure 1.2	Examples of training and testing	04
Figure 5.1	Activity Diagram	20
Figure 5.2	Use Case Diagram	21
Figure 5.3	Data Flow Diagram level 0	22
Figure 5.4	Data Flow Diagram level 1	22
Figure 5.5	Sequence Diagram	23
Figure 6.1	VGG 16 Architecture	25
Figure 6.2	System Architecture	26
Figure 7.1	Testing	32
Figure 8.1	Performance Analysis	37
Figure 8.2	Accuracy of CNN model	38
Figure 8.3	Loss of CNN model	38
Figure A1	Training Datasets	41
Figure A2	GUI	41
Figure A3	50 Rupees Real Note	42
Figure A4	50 Rupees Fake Note	42
Figure A5	100 Rupees Real Note	43
Figure A6	100 Rupees Fake Note	43
Figure A7	200 Rupees Real Note	44
Figure A8	200 Rupees Fake Note	44
Figure A9	500 Rupees Real Note	45
Figure A10	500 Rupees Fake Note	45

LIST OF TABLES

Table No.	Table Name	Page No.
Table 7.1	Unit testing	32
Table 7.2	Input Frame Test Case	34
Table 7.3	Convert to grayscale Test Case	35
Table 7.4	Detect edge Test Case	35
Table 7.5	Output Frame test case	36

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW:

Computers and mobile phones have become an unavoidable part of our lives. There are a lot of things which we can do with these technologies. With the rapid development of mobile phones and technologies come several services like application creation - (refers to the process of making application software for handheld and desktop devices such as personal computers and Personal Digital Assistants. Fake currency Detection is a system that can be used to overcome the limitations most of the people and our institutions of higher learning face with respect to making difference between counterfeit currencies- (is imitation currency produced without the legal sanction of the state or government, usually in a deliberate attempt to imitate that currency and so as to deceive its recipient) and real currencies. The project involves making use of Digital Image Processing Domain - Digital image processing is the use of computer algorithms to perform image processing on digital images.

This has led to the increase of corruption in our country hindering the country's growth. Some of the methods to detect fake currency are watermarking, optically variable ink, security thread, latent image, techniques like counterfeit detection pens. We hereby propose an application system for detecting fake currency where image processing is used to detect fake notes. We will find out dissimilarities between the image under consideration and the prototype. CNN classifiers will be used to detect fake currency. The proposed system for fake currency detection will be simple, accurate and easy to use.

1.2 PROBLEM STATEMENT:

In exiting and ongoing concepts of different adaptations, tests and innovations had been kept for the future due to the lack of time. Few of the major problem are:

- Speed of system is slow.
- Not able to keep track of device's location through which the currency is scanned

- and maintaining the same in the database.

1.3 SIGNIFICANCE AND RELEVANCE OF WORK:

- Significance of work is thus about aiming to find some intrinsic value in one's work- related activities that make them worth doing, it is a general evaluation of “the value or worth of one's work”
- The devices to test currency’s authenticity or originality are costly and take large sum for maintenance as well. System made can be easily used, maintained with low cost and low investment with more accurate results.

1.4 OBJECTIVES:

- The main objective of fake currency detection project is to detect fake currency using deep convolutional neural network.
- To provide cheaper and accurate system to the user, which can be easily accessible and give accurate results.
- To develop a user-friendly application for currency recognition system.
- To make it available for common people quickly and easily with low cost.

1.5 METHODOLOGY:

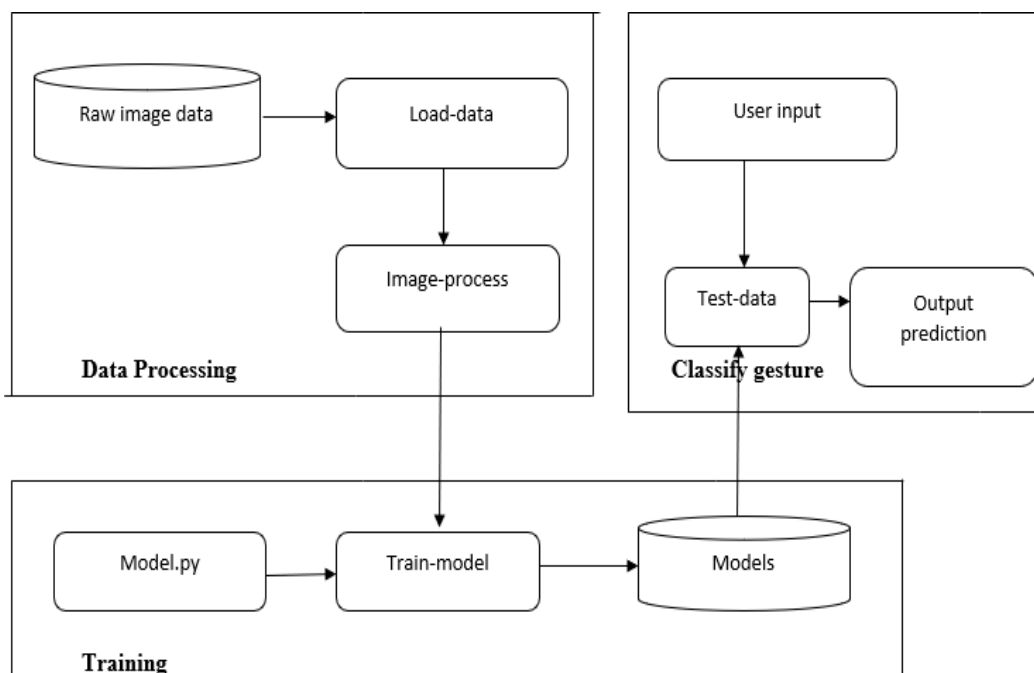


Figure 1.1: Methodology

The main stages involved in this method are Data Collection, Pre-processing, Feature extraction, prediction model and evaluation.

Step 1: Gather Your Dataset

The first component of building a deep learning network is to gather our initial dataset. We need the images themselves as well as the labels associated with each image. These labels should come from a finite set of categories, such as: categories fake and real currency.

Furthermore, the number of images for each category should be approximately uniform (i.e., the same number of examples per category) then our classifier will become naturally biased to overfitting into these heavily-represented categories.

Class imbalance is a common problem in machine learning and there exist a number of ways to overcome it. We'll discuss some of these methods later, but keep in mind the best method to avoid learning problems due to class imbalance is to simply avoid class imbalance entirely.

As our system is mainly focusing on detection of Fake currency, we gathered our data as images. The dataset obtained consisted of several images of real and fake currencies.

Step 2: Split Your Dataset and pre-processing

Now that we have our initial dataset, we need to split it into two parts:

1. A training set
2. A testing set

A training set is used by our classifier to “learn” what each category looks like by making predictions on the input data and then correct itself when predictions are wrong. After the classifier has been trained, we can evaluate the performing on a testing set.

It's extremely important that the training set and testing set are independent of each other and do not overlap! If you use your testing set as part of your training data, then your classifier has an unfair advantage since it has already seen the testing examples before and “learned” from them. Instead, you must keep this testing set entirely separate from your training process and use it only to evaluate your network.

Common split sizes for training and testing sets include 66:6%33:3%, 75%=25%, and 90%=10%, respectively. (Figure 1.2):

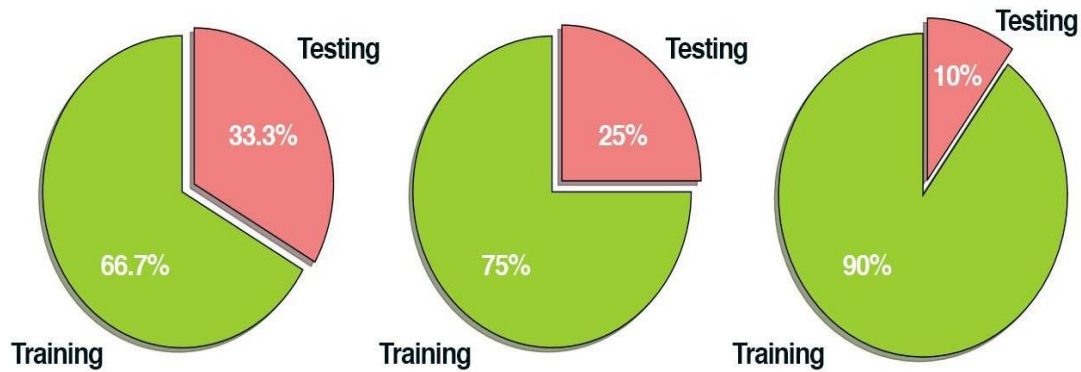


Figure 1.2: Examples of common training and testing data splits.

These data splits make sense, but what if you have parameters to tune? Neural networks have a number of knobs and levers (ex., learning rate, decay, regularization, etc.) that need to be tuned and dialed to obtain optimal performance. We'll call these types of parameters hyperparameters, and it's critical that they get set properly. In practice, we need to test a bunch of these hyperparameters and identify the set of parameters that works the best. You might be tempted to use your testing data to tweak these values, but again, this is a major no-no! The test set is only used in evaluating the performance of your network. Instead, you should create a third data split called the validation set. This set of the data (normally) comes from the training data and is used as "fake test data" so we can tune our hyperparameters. Only after have we determined the hyperparameter values using the validation set do we move on to collecting final accuracy results in the testing data. We normally allocate roughly 10-20% of the training data for validation. If splitting your data into chunks sounds complicated, it's actually not.

Pre-processing

The primary target is to improve image highlights needed for additional processing. Here, the input image is converted into grayscale image for all the further preprocessing purposes. The image is then thresholded and further erosion and dilation is applied to the thresholded image. This image is used to extract the contours and extreme points.

Step 3: Train Your Network

Given our training set of images, we can now train our network. The goal here is for our network to learn how to recognize each of the categories in our labeled data. When the model makes a mistake, it learns from this mistake and improves itself. So, how does the actual “learning” work? In general, we apply a form of gradient descent.

Step 4: Evaluate

Last, we need to evaluate our trained network. For each of the images in our testing set, we present them to the network and ask it to predict what it thinks the label of the image is. We then tabulate the predictions of the model for an image in the testing set.

Finally, these model predictions are compared to the ground-truth labels from our testing set. The ground-truth labels represent what the image category actually is. From there, we can compute the number of predictions our classifier got correct and compute aggregate reports such as precision, recall, and f-measure, which are used to quantify the performance of our network as a whole.

Convolutional Neural Network

CNN is utilized to get better result. The signal convolved with kernels to get include map. Past layers are interconnected with weights of the kernel. To upgrade the qualities of information image by back propagation calculation. Since feature maps of all units shared by the kernels. It will serve to reduces over fitting. Each data of neighbourhood is taken by utilizing kernels. Kernel is a major source of context information. Activation function is applied to the output of neural network

- Convolutional Layers

Objective of the convolution layer is to take or extract the features from the input [image], just the part of picture is link to the following convolution layer.

- Padding

Padding is incorporating a zero layer outside the input volume so the data on border won't be lost and we can get a similar dimension of output as input volume. Here we are using zero padding.

- Activation Function

Non-linear activation function ReLU (Rectifier Activation function) is used to provide accurate results than classical sigmoid function.

- Pooling Layer

It is used for combining spatially nearby features. Max-pooling is generally used to join features. It decreases the dimension of input image and controls overfitting.

1.6 ORGANIZATION OF THE REPORT:

This report is organized into majorly into 5 different sections and each section provides detailed/ brief description about the project. The 5 sections mentioned are:

1. Introduction – This section provides you overview of the project, what's the major problem that is being addressed, objectives, which methodology we are following to implement this project and information about remaining part of the report.

2. Literature Survey – This section provides previous work of this problem and their limitations.

3. SRS – System requirement Specification section provides information about functional and non-functional requirements of this project.

4. Expected outcome – This gives an idea of how the outcome would be.

5. Advantages and Applications – What are the advantages of the approach or framework being developed comparatively to previous existing one and information regarding application of the project in various fields.

CHAPTER 2

LITERATURE SURVEY

2.1 Counterfeit currency detection using deep convolutional neural network (2019) presented by Prof Kiran Kamble, Anuthi Bhansali, Pranali Satalgaonkar, Shruthi Alagundgi

In this relevant paper, many recognition techniques are implemented to recognize images, recognize faces, recognize car license plates, and recognize human behaviours. Currency is the primary average for circulation, and Various countries' currencies have different qualities. However, when the value of currency grows, there will be an increase in counterfeit currency. Counterfeit money might damage these nations' interests. As a result, one of the hottest subjects and a critical issue at the moment is how to use recognition technology to the genuine of money (Zhang, 2018). Visual examination was used in the past to identify and genuine money, particularly currency notes. Our eyesight cannot sense everything; sometimes, it is not easy for humans to distinguish genuine currency from auth genuine entice currency without the aid of technology.

Disadvantages

- Consumes more time to produce output
- Consumes more space

2.2 Fake currency Detection using Basic Python Programming and Web Framework (2020) presented by Prof Chetan More, Monu Kumar, Rupesh Chandra, Raushan Singh

Currency duplication also known as counterfeit currency is a vulnerable threat on economy. Although fake currency is being printed with precision, the Crime Investigation Department (CID) says that they can be detected with some effort. Currency printed by local racketeers can be detected easily as they use the photographic method, hand engraved blocks, lithographic processes and computer colour scanning. In counterfeit notes, the watermark is made by using opaque ink, painting with white solution, stamping with a dye engraved with the picture of

Development of an analytic tool for software-based vehicle condition analysis for resales.

Mahatma Gandhi. Tourists are the most vulnerable people to fake currencies, because they don't know the proper and precise way of finding the difference between fake and real currencies note. So automatic identification of currencies using image processing technique will be helpful to these peoples.it is also be useful at other workplaces. . The system designed to check the Indian currency note with denominations 10, 20, 50, 100, 200, 500 and 2000. It will pre-process the digital pictures and organize the prepared arrangement of information and it will distinguish in monetary forms. This paper proposes a convenient and cheapest method for identifying Indian currencies. At the end of the process user can know whether the currency note is fake or real and its equivalent currency value into more than 150 counties

Disadvantages

- Flask web frameworks are more complex
- high maintenance cost

2.3 Identification of fake notes and denomination recognition presented by Archana MR, Kalpitha C P, Prajwal S K, Pratiksha N

Image processing is a rapidly growing area of research with application to various aspects of business. Image processing is used to convert an image to digital as well as to obtain certain types of information from the same. The image processing and processing modes include analog and digital image processing. Digital image processing techniques helps to manipulate digital images with computers. The system uses computer algorithms for image processing which is better than analog processing and prevents various processing problems such as noise and signal distortion that provides more complex algorithms and implementation of methods that are not possible in analog design. Currency is used as the medium of exchange for goods and services. Human error is a huge concern in cases where large amounts of cash transactions are conducted, leading to a push for increase in automation of transactions in the banking sector. Indian paper currency consists of six major denominations, with each having distinguishing features, such as size,

prominent color, identification mark. With the development of sophisticated printing techniques, counterfeit currency has become a significant concern. Some of the consequence of counterfeit notes on society are a reduction in the value of real money, increase in prices due to more money being circulated in the economy and decrease in acceptability of money. To prevent circulation of counterfeit notes, a system to detect fake notes must be developed. Notes with the legal sanction of the government possess certain security features such as intaglio printing, fluorescence and watermark. So far, many different approaches have been proposed to solve the problem of paper currency recognition and verification.

Disadvantages

- Complex Mechanism
- High cost

2.4 Hidden security features for the recognition of fake currency (2018) - K Lavanya , T B Bhaskara Reddy.

System proposed in this paper the note is checked by using image processing techniques like image acquisition, reprocessing, and image enhancement. The main objective of this is to specify about several security features of the highest denomination note introduced in year 2016.

Disadvantages

- If the object is smaller than the pixel size then it cannot be applied

2.5 M. R. Pujar. “Indian Currency Recognition and Verification using Image Processing”, International Journal of Advance Research, Ideas and Innovations in Technology

In this paper we are using MATLAB which involves extraction of invisible and visible features of Indian currency notes. The image of the currency note is captured through a digital camera. Processing on the image is done on that acquired image using concepts like image segmentation, edge information of image and characteristics feature extraction.

Disadvantage

- MATLAB Takes more time to execute than other compiled languages

2.6 Currency Counting Fake Note Detection presented by Mayank in 2018

The proposed system is implemented the fake note detection unit with MATLAB algorithm. This paper is based on the image processing and UV LED's to give solutions for fake currency problems.

Disadvantages

- MATLAB takes more time to execute
- It is not open source and expensive to purchase

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATION

3.1 SYSTEM REQUIREMENT SPECIFICATION

System Requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and non-functional requirements only.

It doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements.

3.1.1 Hardware Specification

- Processor: intel core i3 or above
- Processor speed: 500Mhz or above
- RAM: 4GB or above

3.1.2 Software Requirement

- Python
- Python IDLE

Software Libraries Required

- OpenCV
- Imutils
- Numpy
- Keras
- Tensorflow
- Pillow
- Tkinter

Software Specification:

Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a “batteries included” language due to its comparative standard library



Python Idle

Every Python installation comes with an Integrated Development and Learning Environment, which you'll see shortened to IDLE or even IDE. These are a class of applications that help you write code more efficiently. While there are

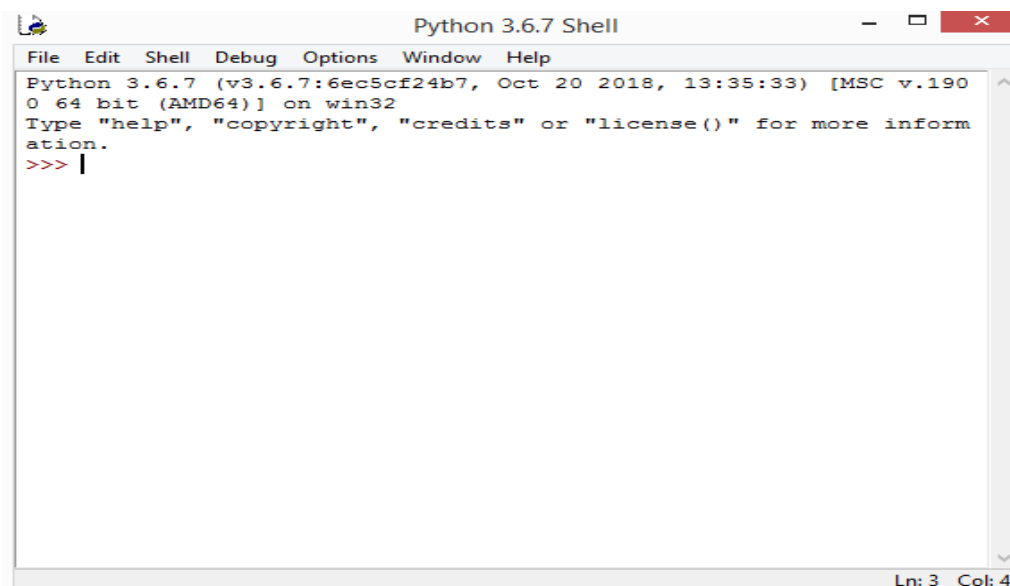
many IDEs for you to choose from, Python IDLE is very bare-bones, which makes it the perfect tool for a beginning programmer.

Python IDLE comes included in Python installations on Windows and Mac. If you're a Linux user, then you should be able to find and download Python IDLE using your package manager. Once you've installed it, you can then use Python IDLE as an interactive interpreter or as a file editor.

An Interactive Interpreter

The best place to experiment with Python code is in the interactive interpreter, otherwise known as a shell. The shell is a basic Read-Eval-Print Loop (REPL). It reads a Python statement, evaluates the result of that statement, and then prints the result on the screen. Then, it loops back to read the next statement.

The Python shell is an excellent place to experiment with small code snippets. You can access it through the terminal or command line app on your machine. You can simplify your workflow with Python IDLE, which will immediately start a Python shell when you open it.

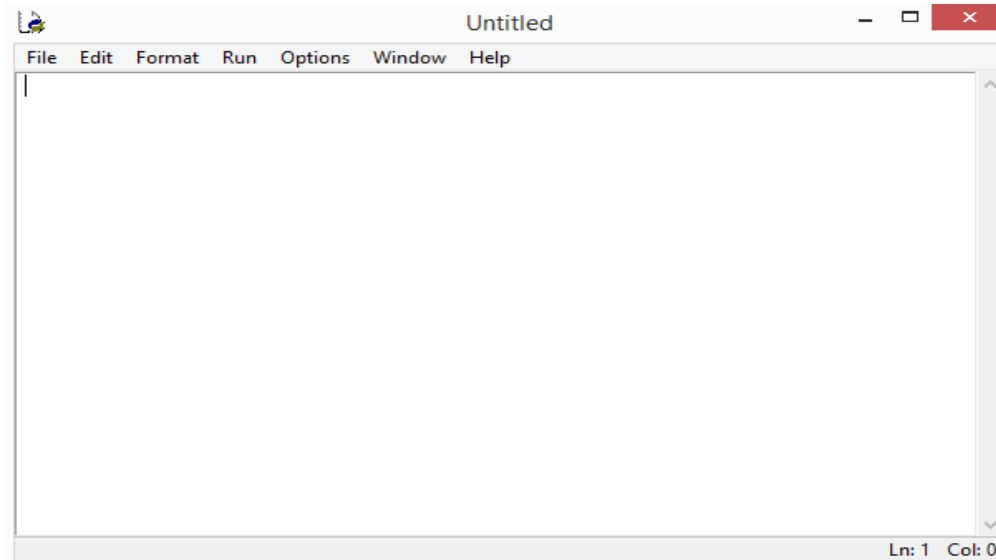


```
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

A File Editor

Every programmer needs to be able to edit and save text files. Python programs are files with the `.py` extension that contain lines of Python code. Python IDLE gives you the ability to create and edit these files with ease.

Python IDLE also provides several useful features that you'll see in professional IDEs, like basic syntax highlighting, code completion, and auto-indentation. Professional IDEs are more robust pieces of software and they have a steep learning curve. If you're just beginning your Python programming journey, then Python IDLE is a great alternative.



Libraries Specification

Opencv

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then

the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV

It is used for:

- Reading an image
- Extracting the RGB values of a pixel
- Extracting the Region of Interest (ROI)
- Resizing the Image
- Rotating the Image
- Drawing a Rectangle
- Displaying text

Imutils

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV

Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

Keras and tensorflow

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation

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 workflows should be quick and easy, while arbitrarily advanced workflows should be *possible*
- **Powerful** -- Keras provides industry-strength performance and scalability

TensorFlow is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.
- Scaling computation to many devices, such as clusters of hundreds of GPUs.
- Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of Tensor Flow: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning.

Tkinter

Python provides various options for developing graphical user interfaces (GUIs).

The most important features are listed below.

- **Tkinter** – Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.
- **wxPython** – This is an open-source Python interface for wxWidgets GUI toolkit. You can find a complete tutorial on WxPython
- **PyQt** – This is also a Python interface for a popular cross-platform Qt GUI library. Tutorialspoint has a very good tutorial on PyQt
- **JPython** – JPython is a Python port for Java, which gives Python scripts seamless access to the Java class libraries on the local machine

There are many other interfaces available, which you can find them on the net.

Pillow

Python pillow library is used to image class within it to show the image. The image modules that belong to the pillow package have a few inbuilt functions such as load images or create new image.

3.2 FUNCTIONAL REQUIREMENTS:

- Device must be enabled or disabled by user.
- Device should be able to extract the features of the image captured by camera.
- Device should be able to tell the real image or fake image notes.
- GUI performance should be stable without severely lagging

3.3 NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, and the need for interoperability with other software and hardware systems or because of external factors such as:

Usability

The system must be easy to learn for both users of the device and helpers who are the users of the GUI interface.

Reliability

The reliability of the device essentially depends on the software tools (OpenCV, NumPy, TensorFlow etc.) and hardware tools (camera, computer etc.) used for the system development.

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

Because bogus denominations in the market and illiteracy among the people in a given country are causing money recognition problems in today's world, an automatic currency recognition system is essential. The two most critical criteria in such a system are accuracy and speed. Furthermore, accuracy takes precedence than speed. By examining the significant qualities, a currency recognizer recognises the currency and determines the denomination. There were numerous ways presented by researchers. Physical attributes (width, length) are used by some, while internal properties are used by only a few (texture, colour). Every problem in the software industry has a solution. We can save both time and energy by utilising such software solutions. There was a system in the early 1990s that used image processing to identify cash notes. Their algorithm, on the other hand, does not take into consideration characteristics of note authentication. It was necessary for such a system to have a picture input, after which it was executing various activities for it.

4.1.1 Limitation

From the observation of the papers, we can say that there are certain stages which are very important in the existing system architecture. 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 taken 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.

4.2 PROPOSED SYSTEM:

These days, technology advances at a breakneck pace. As a result, the banking sector is becoming more up to date by the day. This necessitates the use of automatic currency recognition. Many academics have been urged to create a robust and effective automatic cash identification machine. Automated equipment that can recognize banknotes are now widely utilized in modern product dispensers such as candy, cold drink bottles, and bus or train tickets. Cash recognition technology primarily tries to identify and extract visible and unseen properties of currency notes. To date, a number of methods for identifying the currency note have been proposed. However, the most effective method is to make use of the currency's visual properties. Color and size, for example. If the note is soiled or ripped, however, this procedure will not work. When a note becomes soiled, its color qualities are drastically altered. As a result, it's critical to consider how we extract information from the image of the currency note and apply the appropriate algorithm to increase note recognition accuracy.

4.2.1 Advantages:

The proposed system contains the advantages of the existing system and eliminates the disadvantages of it. The project centres on the design and implementation of Fake Currency Detection. The scope of the project is to provide approaches and strategies, which have proved to be suitable when accessing the image of the desired currency note.

The scope of this project includes:

- Study existing image detection schemes and concern on recognition base types.
- Study the usability features of the existing fake currency detection methods from the general and ISO features.
- Mapping between the recognition-based image detection system methods and the usability features and extract a collection of usability features to be built in the new system prototype.

CHAPTER 5

SYSTEM DESIGN

5.1 ACTIVITY DIAGRAM:

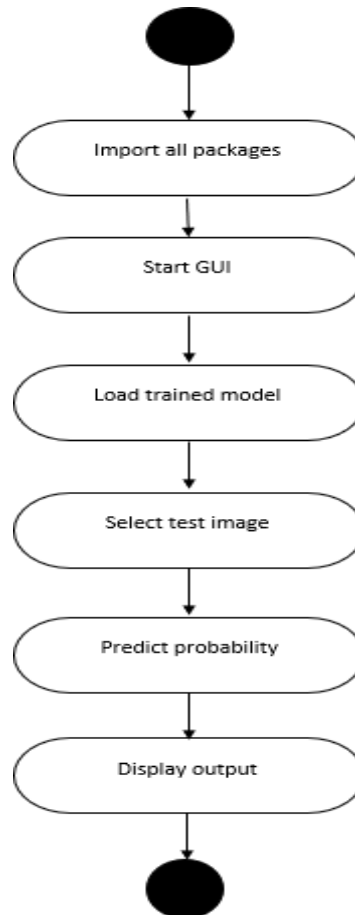


Figure 5.1: Activity diagram

We use **Activity Diagrams** to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behaviour diagrams. An activity diagram is a **behavioural diagram** i.e., it depicts the behaviour of a system. An activity diagram portrays the control flow from a start

point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system. An activity diagram is very **similar to a flowchart**.

5.2 USE CASE DIAGRAM:

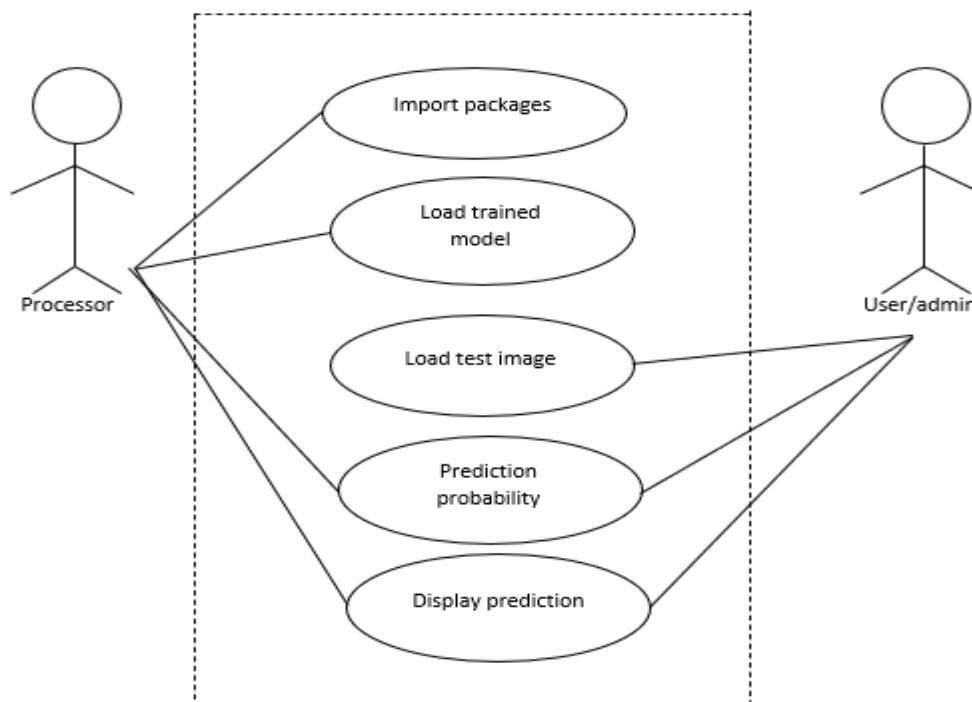


Figure 5.2: Use case diagram

Use case consists of user and processor where user is used to provide the input to the system and processor is used to process the input data and provide output. The flow is shown in the above diagram.

First user as to run the system and run the code, model and library packages are imported and loaded. After the run of code GUI is being displayed and click on select file and load the test image. After loading the image, click in prediction button to analyse the image and to give predicted output and displayed.

5.3 DATA FLOW DIAGRAM:

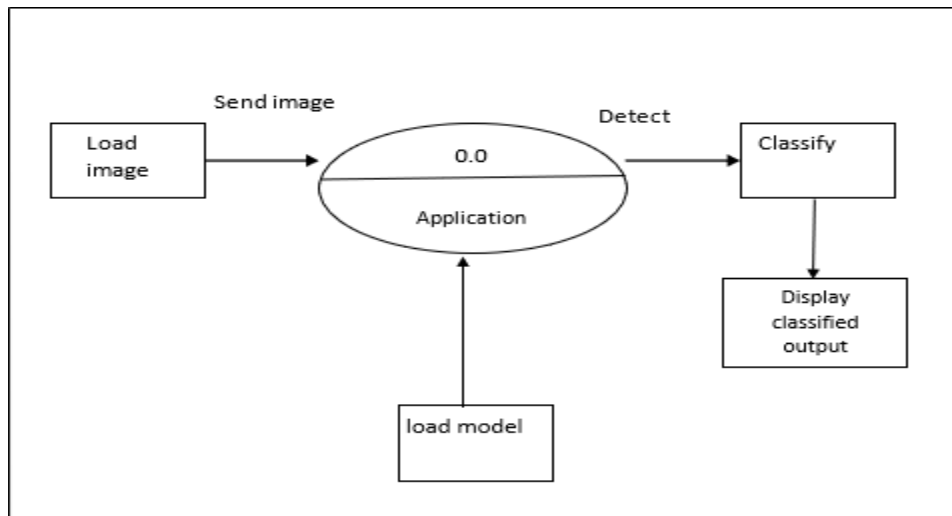


Figure 5.3: data flow diagram level 0

Above mentioned diagram is the representation of DFD0 which provides u the content diagram or say overview of the whole system. It is designed to be an at- a-glance view, showing the system as single high level process. Here from the file image is be loaded to the application where the loaded image is sent to classification unit to predict the result with the help of CNN model file

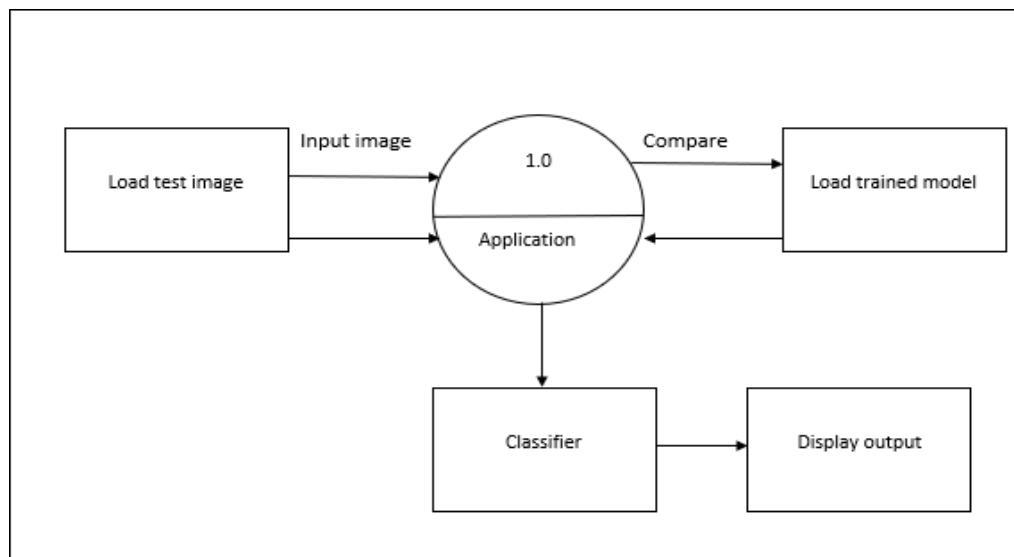


Figure 5.4: dataflow diagram level 1

Above mentioned diagram is the representation of DFD1. The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules

in the system and flow of data among various modules. Here from the file image is be loaded to the application where the loaded image is sent to classification unit to predict the result and classes are classified given a label.

5.4 SEQUENCE DIAGRAM

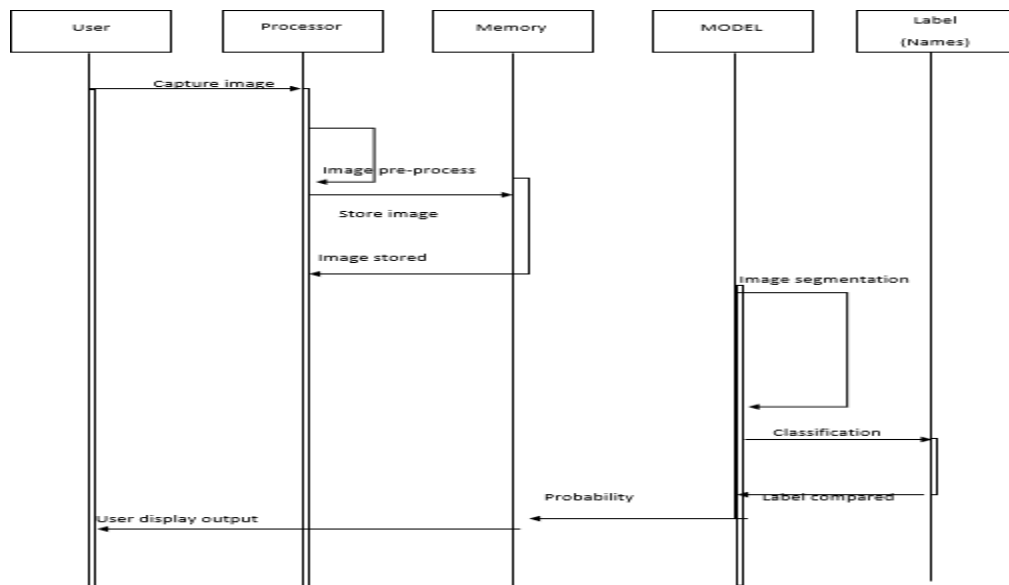


Figure 5.5: sequence diagram

Sequence diagram consists of 5 different blocks namely user, processor, memory, Model and labels as shown in the above figure

User will provide the input image through the file's already saved image is being taken in consideration which is been captured and sent to the processor where pre-processing of data is done which is resizing, reshaping and other parameters and after that those are stored in the memory unit.

After pre-processing and storing of image, cnn trained model file is loaded where the featured of the image is extracted for classifying the output. After classifying the output, label is provided.

CHAPTER 6

IMPLEMENTATION

OVERVIEW OF SYSTEM IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed.

Steps for Implementation

- Write up Installation of Hardware and Software utilities.
- Write up about sample data used.
- Write up about debugging phase.
- Implementation steps

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified and so that debugging testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity clarity and elegance are the hallmarks of good programs and these characteristics have been implemented in each program module.

The goals of implementation are as follows

- Minimize the memory required.
- Maximize output readability.
- Maximize source text readability.
- Minimize the number of source statements
- Minimize development time

6.1 ALGORITHM/PSEUDO CODE MODULE WISE

Convolutional Neural Network

CNN is utilized to get better result. The signal convolved with kernels to get include map. Past layers are interconnected with weights of the kernel. To upgrade the qualities of information image by back propagation calculation. Since feature maps of all units shared by the kernels. It will serve to reduces over fitting. Each data of neighbourhood is taken by utilizing kernels. Kernel is a major source of context information. Activation function is applied to the output of neural network.

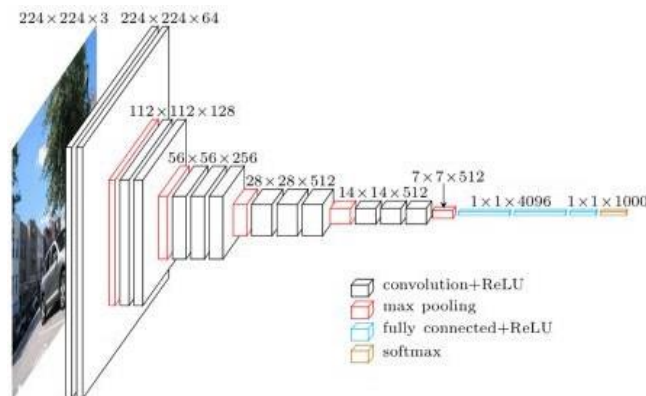


Figure 6.1: VGG 16 Architecture

- Convolutional Layers: Objective of the convolution layer is to take or extract the features from the input [image], just the part of picture is link to the following convolution layer.
- Padding: Padding is incorporating a zero layer outside the input volume so the data on border won't be lost and we can get a similar dimension of output as input volume. Here we are using zero padding.
- Activation Function: Non- linear activation function ReLU (Rectifier Activation function) is used to provide accurate results than classical sigmoid function.
- Pooling Layer: It is used for combining spatially nearby features. Max-pooling is generally used to join features. It decreases the dimension of input image and controls over-fitting.

Each layer in a CNN applies a different set of filters, typically hundreds or thousands of them, and combines the results, feeding the output into the next layer in the network. During training, a

CNN automatically learns the values for these filters.

In the context of image classification, our CNN may learn to:

- Detect edges from raw pixel data in the first layer.
- Use these edges to detect shapes (i.e., “blobs”) in the second layer.
- Use these shapes to detect higher-level features such as facial structures, parts of a car, etc.in the highest layers of the network.

The last layer in a CNN uses these higher-level features to make predictions regarding the contents of the image.

In terms of deep learning, an (image) convolution is an element-wise multiplication of two matrices followed by a sum.

1. Take two matrices (which both have the same dimensions).
2. Multiply them, element-by-element (i.e., not the dot product, just a simple multiplication).
3. Sum the elements together.

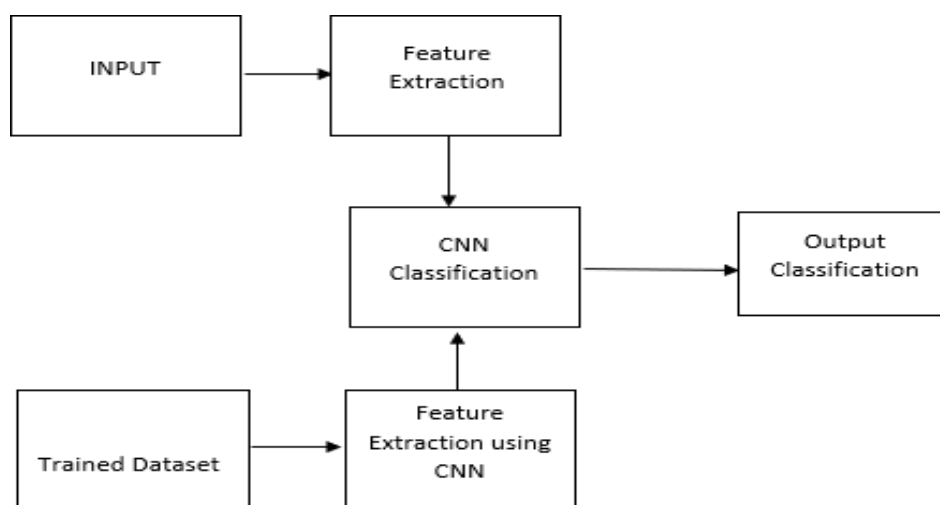


Figure 6.2: System Architecture

The above Figure illustrates the block diagram of the proposed system. The proposed model contains the three stages in order to classify and detect the digits:

- A. Feature Extraction
- B. CNN Classification
- C. Training dataset

Training Dataset

It is initial set of data used to help a program understand how to apply neural networks and produce results. It is then complimented by validation and testing dataset. The initial approach to the training set images that are to be processed in order to reduce the data, by thresholding them into a binary image.

Feature Extraction

It is a process of dimension reduction of initial raw dataset to more manageable groups for processing. In the feature extraction stage redundancy from the data is removed.

CNN Classification

It is a specific type of Artificial Neural Network (ANN) which uses Perceptron, a machine learning unit algorithm for supervised learning to analyse data.

Pseudo code:

```
import tkinter as tk
from PIL import Image, ImageOps, ImageTk
from tkinter.filedialog import askopenfilename
import cv2
import imutils
import numpy as np
```

```
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
import time

window = tk.Tk()
window.title("Detection")

window.geometry('1100x650')
window.configure(background='snow')

message = tk.Label(window, text="Fake Currency Detection", bg="snow",
fg="black", width=48,
                    height=2, font=('times', 30, 'italic bold '))
message.place(x=5, y=10)

def exit():
    window.destroy()

def clear():
    rst.destroy()

def on_closing():

from tkinter import messagebox
    if messagebox.askokcancel("Quit", "Do you want to quit?"):
        window.destroy()

window.protocol("WM_DELETE_WINDOW", on_closing)

#Disable scientific notation for clarity
np.set_printoptions(suppress=True)

classes = {
```

```
1:'100 Fake',
2:'100 Real',
3:'200 Fake',
4:'200 Real',
5:'500 Fake',
6:'500 Real',
7:'50 Fake',
8:'50 Real',
}

print("[INFO] loading network...")
model = load_model("final.model")

def model_predict():
    frame = cv2.imread(path)
    # pre-process the image for classification
    image = cv2.resize(frame, (64, 64))
    image = image.astype("float") / 255.0

    image = img_to_array(image)
    image = np.expand_dims(image, axis=0)

    # load the trained convolutional neural network
    # classify the input image
    label = model.predict(image)[0]
    idx = np.argmax(label)
    final_prediction = classes[idx+1]

    global rst

    rst = tk.Label(text='RESULT\n\n '+ final_prediction.upper(),
background="snow",fg="Black", font=("", 15))

    rst.place(x=800,y=150)
```

```
clrWindow = tk.Button(window, text="Clear", command=clear ,fg="black"
,bg="lawn green" ,width=20 ,height=3, activebackground = "Red" ,font=('times',
15, ' bold '))

clrWindow.place(x=90, y=390)

def openphoto():

    global path

    path=askopenfilename(filetypes=[("Image File",)])

    im = cv2.imread(path)

    cv2image = cv2.cvtColor(im, cv2.COLOR_BGR2RGBA)

    cv2image = imutils.resize(cv2image, width=350)

    img = Image.fromarray(cv2image)

    tkimage = ImageTk.PhotoImage(img)

    myvar=tk.Label(window,image = tkimage, height="350", width="350")

    myvar.image = tkimage

    myvar.place(x=390,y=150)

    preImg = tk.Button(window,
text="Predict",fg="black",command=model_predict ,bg="lawn green" ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))

    preImg.place(x=90, y=270)

    takeImg = tk.Button(window, text="Select
Image",command=openphoto,fg="white" ,bg="blue2" ,width=20 ,height=3,
activebackground = "Red" ,font=('times', 15, ' bold '))

    takeImg.place(x=90, y=150)

    quitWindow = tk.Button(window, text="Quit", command=on_closing
,fg="white" ,bg="Red" ,width=20 ,height=3, activebackground = "Red"
,font=('times', 15, ' bold '))

    quitWindow.place(x=800, y=450)

window.mainloop()
```

CHAPTER 7

TESTING

7.1 METHODS OF TESTING

Software testing is an examination that is carried out to offer information to stakeholders regarding the quality of the product or service being tested. Software testing can also give a corporation with an objective, unbiased picture of the software, allowing them to grasp and comprehend the risks associated with software implementation. The process of executing a programme or application with the goal of detecting software bugs is known as testing (errors or other defects). Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

meets the requirements that guided its design and development,

- responds correctly to all kinds of inputs.
- performs its functions within an acceptable time.
- Is sufficiently usable.
- can be installed and run in its intended environments
- Achieves the general result its stakeholder's desire.

The testing steps are:

- Unit Testing.
- Validation Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.

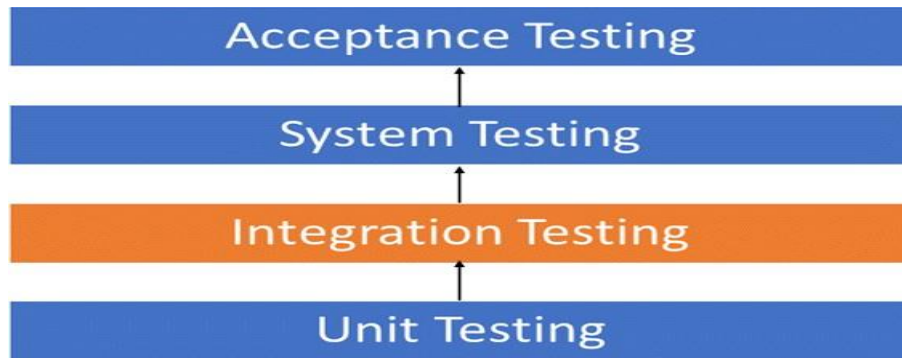


Figure 7.1: Testing

7.1.1 UNIT TESTING

Individual units or components of software are tested in unit testing, which is a sort of software testing. The goal is to ensure that each unit of software code works as intended. Unit Testing is done during the development (coding phase) of an application by the developers. Unit tests are used to isolate a part of code and ensure that it is correct. A singular function, method, procedure, module, or object might be considered a unit. Unit testing is the first step in the testing process before moving on to integration testing.

Because software developers sometimes want to save time by doing limited unit testing, this is a misconception because insufficient unit testing leads to large costs in defect correction during System Testing, Integration Testing, and even Beta Testing after the programme has been constructed. It saves time if proper unit testing is done early in the development process.

Table 7.1: Unit testing

Function Name	Tests Results
Load the model file	Insert the trained model file to the code
Prediction	Predicted values of currency
Display result	Output is displayed successfully.

7.1.2 VALIDATION TESTING

The process of determining if software meets stated business requirements during the development process or at the end of the development process.

Validation testing guarantees that the product fits the needs of the customer. It can also be defined as demonstrating that the product performs as expected when used in the right setting.

7.1.3 INTIGRATION TESTING

Integration testing is a sort of testing in which software modules are conceptually linked and tested as a unit. A typical software project is made up of several software modules written by various programmers. The goal of this level of testing is to find flaws in the way various software modules interact when they're combined.

Integration testing examines data transmission between these units.

Although each software module is unit tested, defects still exist for various reasons like

- In general, a Module is created by a single software developer whose programming logic and understanding may differ from that of other programmers. Integration testing is required to ensure that the software parts work together.
- There's a good risk that the clients' needs will alter throughout module development. These new needs may not be unit tested necessitating system integration testing.
- There's a chance that the software modules database interfaces are incorrect.
- External hardware interfaces, if present, may be incorrect.
- Inadequate exception handling may result in problems.

7.1.4 USER ACCEPTANCE TESTING

User Acceptance Testing (UAT) is a sort of testing in which the end user or customer verifies and accepts the software system before it is moved to the production environment. After functional, integration, and system testing, UAT is performed in the final step of testing.

An acceptance test's performance is essentially the user's show. User motivation and knowledge are essential for the system's proper operation. The aforesaid tests were carried out on the newly designed system, which met all of the requirements. The following test case designs were used for all of the above testing methodologies

7.2 TESTING CASES

Table 7.2: Input Frame Test Case

Test Case	1
Name of Test	Input frame/image
Input	Load the input image / frame
Expected output	Image is been fetched for further process
Actual output	Image is been fetched for further process
Result	Successful

Table 7.3: Convert to grayscale Test Case

Test Case	1
Name of Test	Convert to gray scale
Input	Input image
Expected output	Convert RGB to gray scale image
Actual output	Convert RGB to gray scale image
Result	Successful

Table 7.4: Detect edge Test Case

Test Case	1
Name of Test	Detect edge
Input	Gray scale image
Expected output	Detecting edge
Actual output	Detecting edge
Result	Successful

Table 7.5: Output Frame test case

Test Case	1
Name of Test	Output frame
Input	Image frame and converted image
Expected output	Output frame
Actual output	Output frame
Result	Successful

CHAPTER 8

PERFORMANCE ANALYSIS

Performance: Several performance requirements were established, checking for inputs, outputs and working.

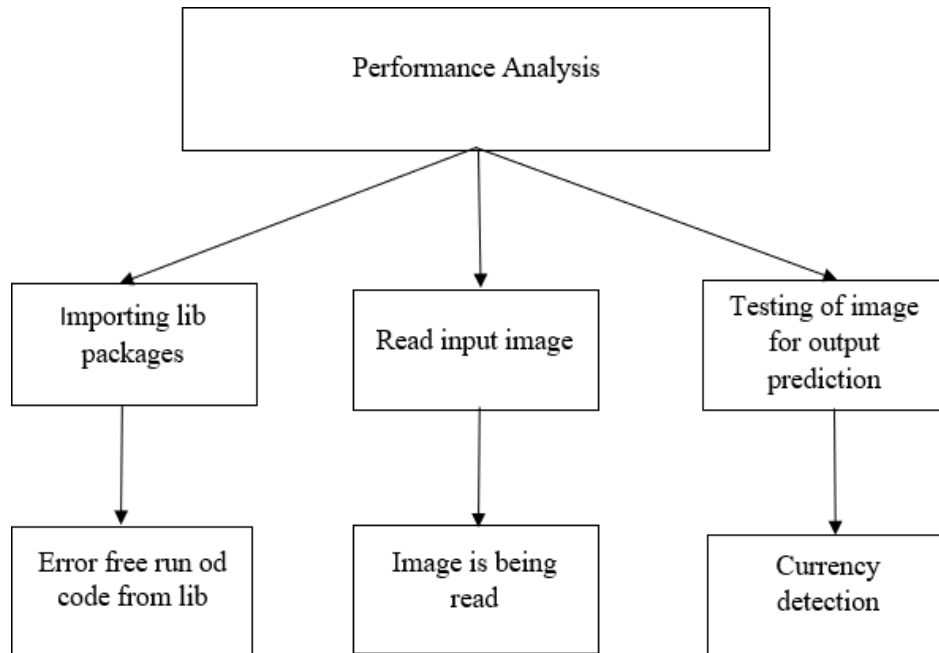


Figure 8.1: Performance Analysis

Environmental: No harm for environmental parameters.

Social: Feasibility for everyone in day-to-day life.

Accuracy: In general, performance data obtained using sampling techniques are less accurate than data obtained by using counters or timers. In the case of timers, the accuracy of the clock must be taken into account.

Simplicity: User friendly.

Flexibility: A flexible can be extended easily to collect additional performance data or to provide different views of the same data. Flexibility and simplicity are often opposing requirements.

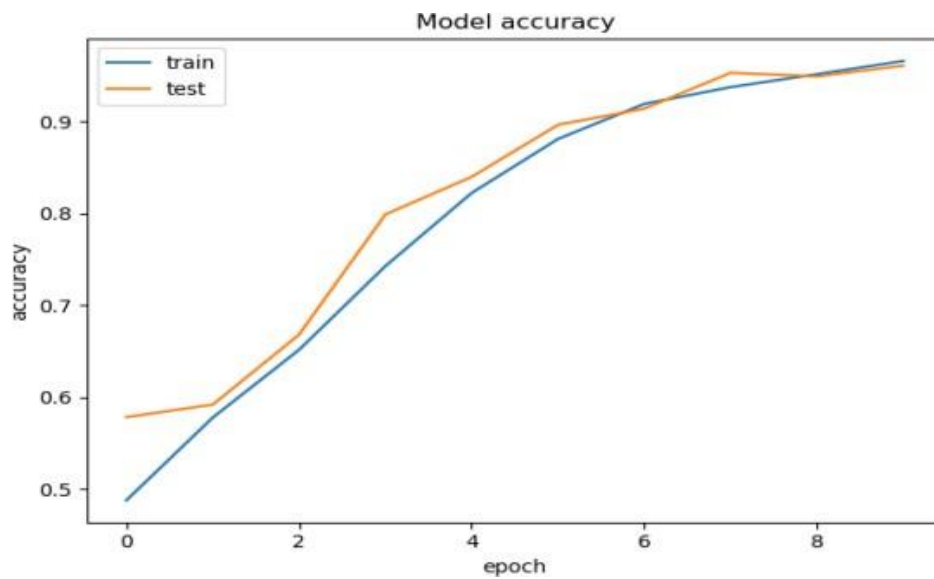


Figure 8.2: Accuracy of CNN model

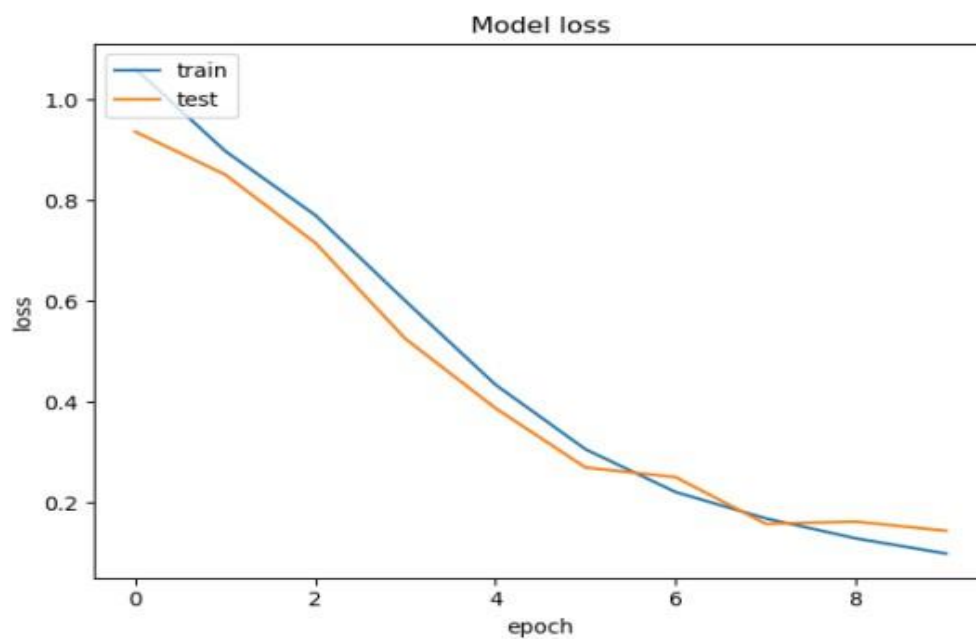


Figure 8.3: Loss of CNN model

Figure 8.2 shows the graph of the accuracy with respect to the epochs. It's observed that as the epochs increase, the accuracy of the model also increases.

Further Figure 8.3 shows the loss function and it can be clearly seen that the loss decreases with increase in the epochs.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION:

Since the monetary property highlights are discovered layer by layer, the discovery precision

is often great. We've looked at the whole image of money so far, but in the future, we'll try to include all of the security features of money by using a fair fundamental structure and providing sufficient preparation information. Furthermore, clamour may be present in the captured image, which must be taken into account as a pre-handling step in the money location procedure. It is also possible to achieve recognition and phoney money recognition by using examples of cash surface as highlights for enhancing the finding precision.

As a result, the various strategies presented in this research were effectively implemented and tested by experiments on the model. Using the modules, CNN was shown to be the optimal feature for performing the approach. By doing model classification, we were able to attain a 95% accuracy rate. In addition, the detection of coins works effectively in this manner.

FUTURE ENHANCEMENT:

Technology is advancing at a rapid pace these days. The proposed technique can be used to detect coins as well as recognise phoney currencies. Other countries' currencies can be added, and a comparison between them can be made. When a picture is loaded into the training folder from the outside, it does not provide 100 percent accuracy. By optimising the system, we can solve this problem.

BIBLIOGRAPHY

- [1] Prof Chetan More, Monu Kumar, Rupesh Chandra, Raushan Singh, “Fake currency Detection using Basic Python Programming and Web Framework” IRJET International Research Journal of Engineering and Technology, Volume: 07 Issue: 04 | Apr 2020 ISSN: 2395- 0056
- [2] Vivek Sharan, Amandeep Kaur,” Detection of Counterfeit Indian Currency Note Using Image Processing” International Journal of Engineering and Advanced Technology (IJEAT), Volume.09, Issue:01, ISSN: 2249-8958 (October 2019)
- [3] Aakash S Patel, “Indian Paper currency detection” International Journal for Scientific Research & Development (IJSRD), Vol. 7, Issue 06, ISSN: 2321-0613 (June 2019)
- [4] Archana M Kalpitha C P, Prajwal S K, Pratiksha N,” Identification of fake notes and denomination recognition” International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume. 6, Issue V, ISSN: 2321-9653, (May2018)
- [5] S. Atchaya, K. Harini, G. Kaviarasi, B. Swathi, “Fake currency detection using Image processing”, International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333 (2017).
- [6] Neural Network and Deep Learning book by Charu C Aggarwal
- [7] Deep Learning Tutorial: <https://www.javapoint.com>

APPENDIX

Appendix A: Screen Shorts

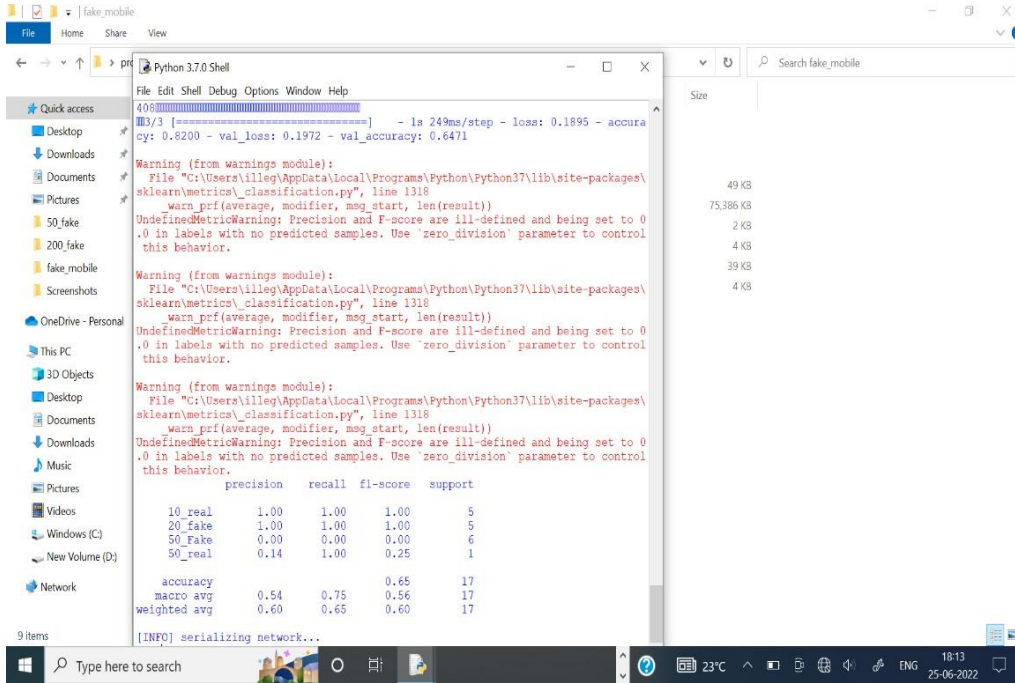


Figure A1: Training Datasets

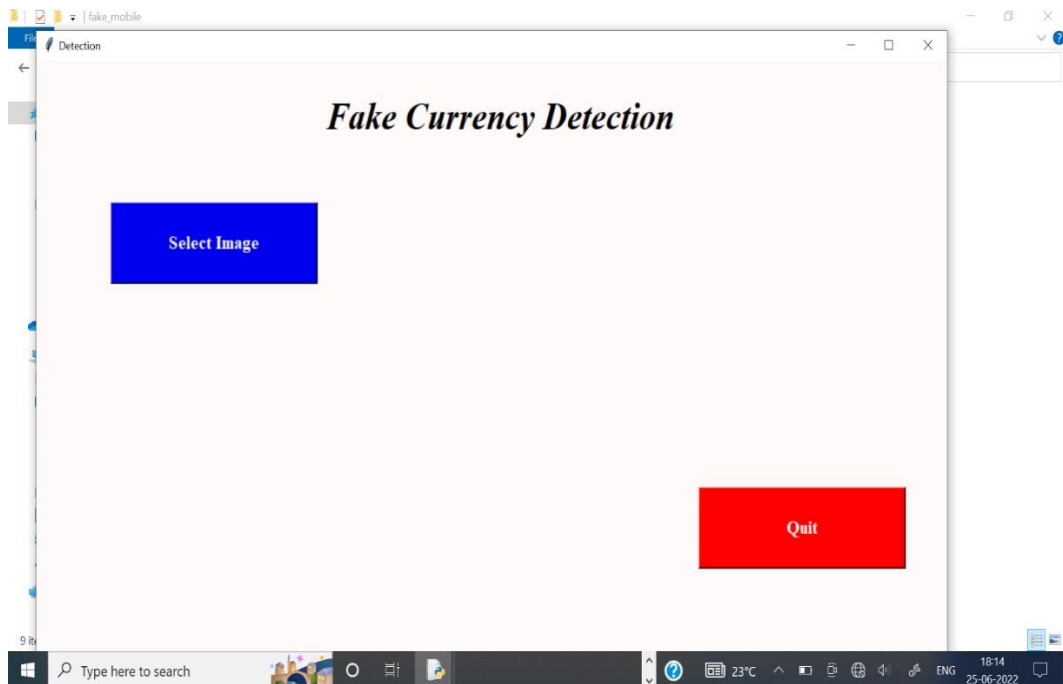


Figure A2: GUI

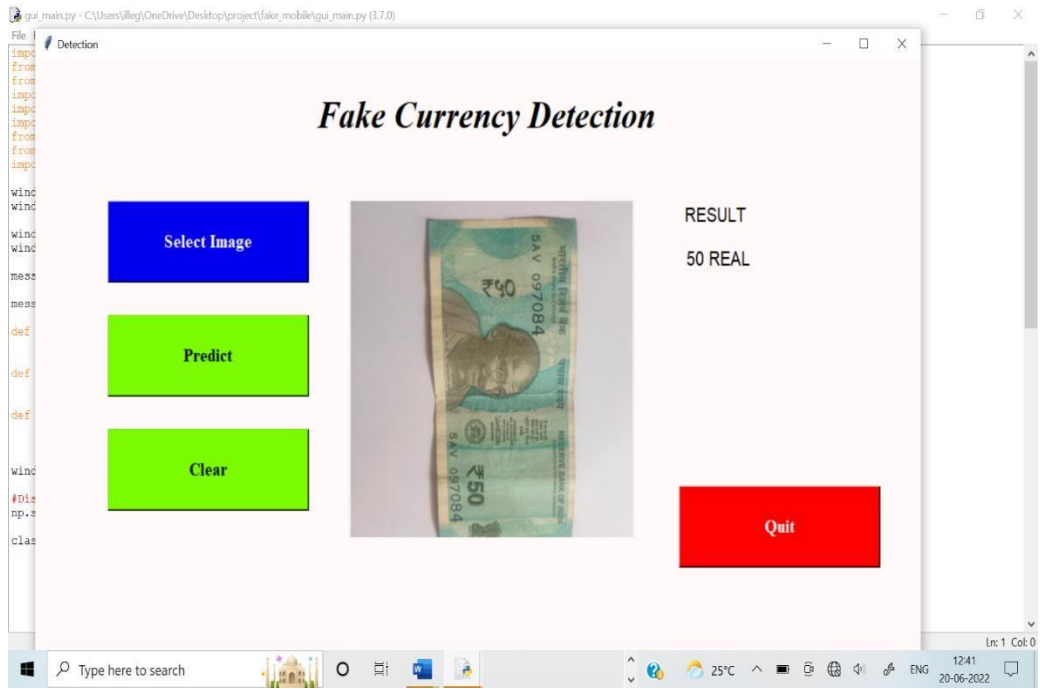


Figure A3: 50 Rupees Real Note

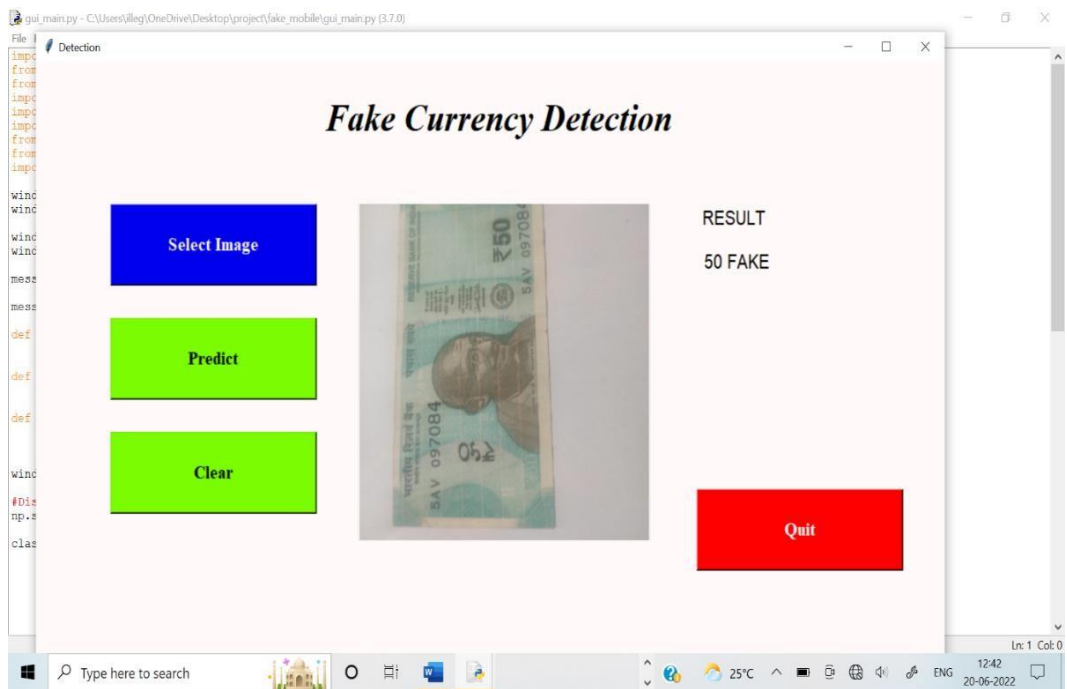


Figure A4: 50 Rupees Fake Note

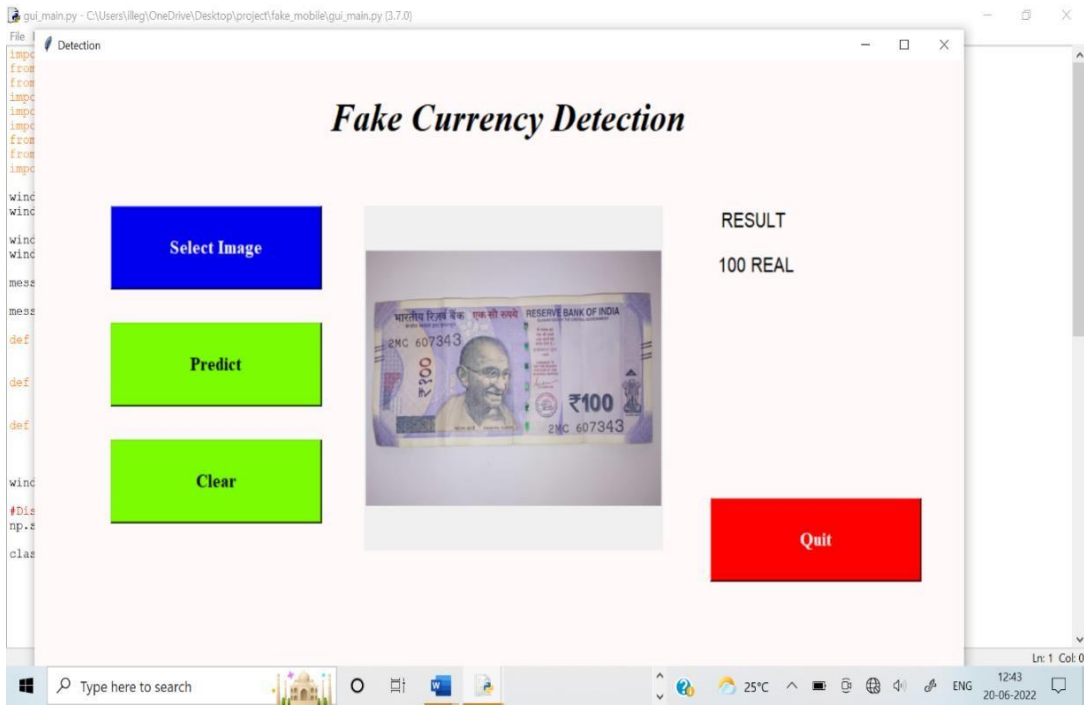


Figure A5: 100 Rupees Real Note

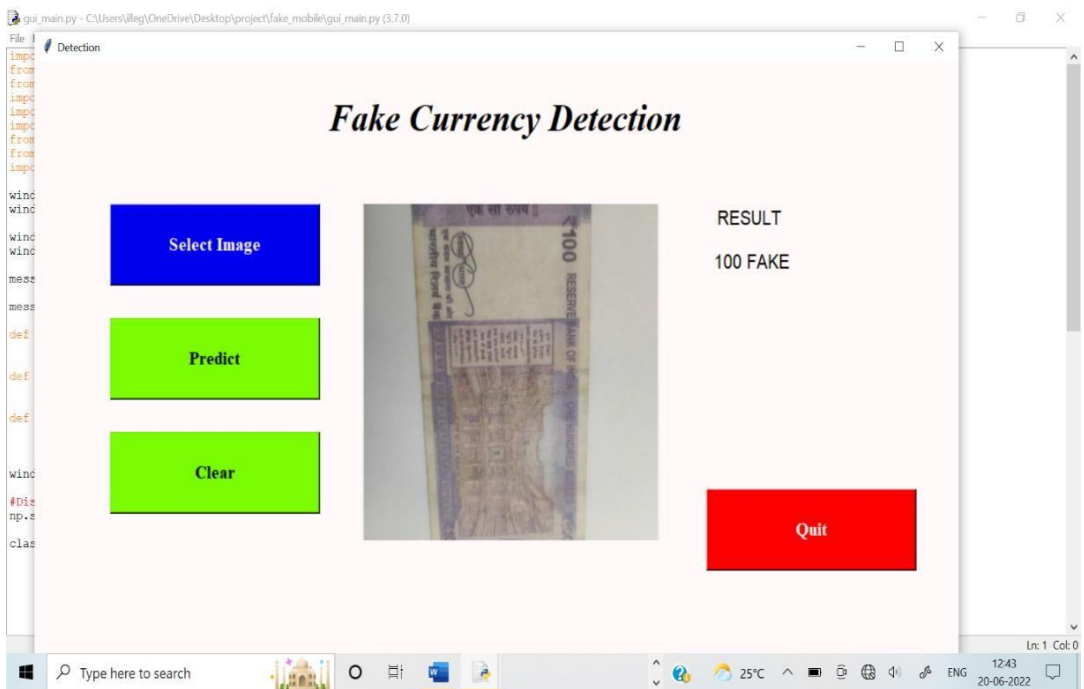


Figure A6: 100 Rupees Fake Note

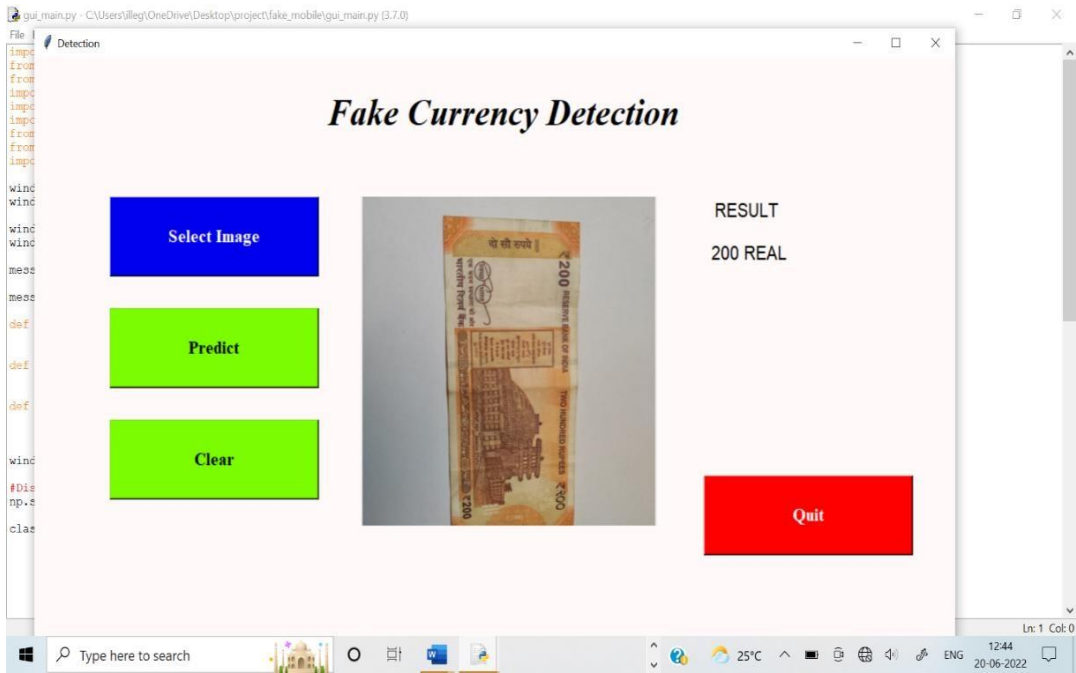


Figure A7: 200 Rupees Real Note

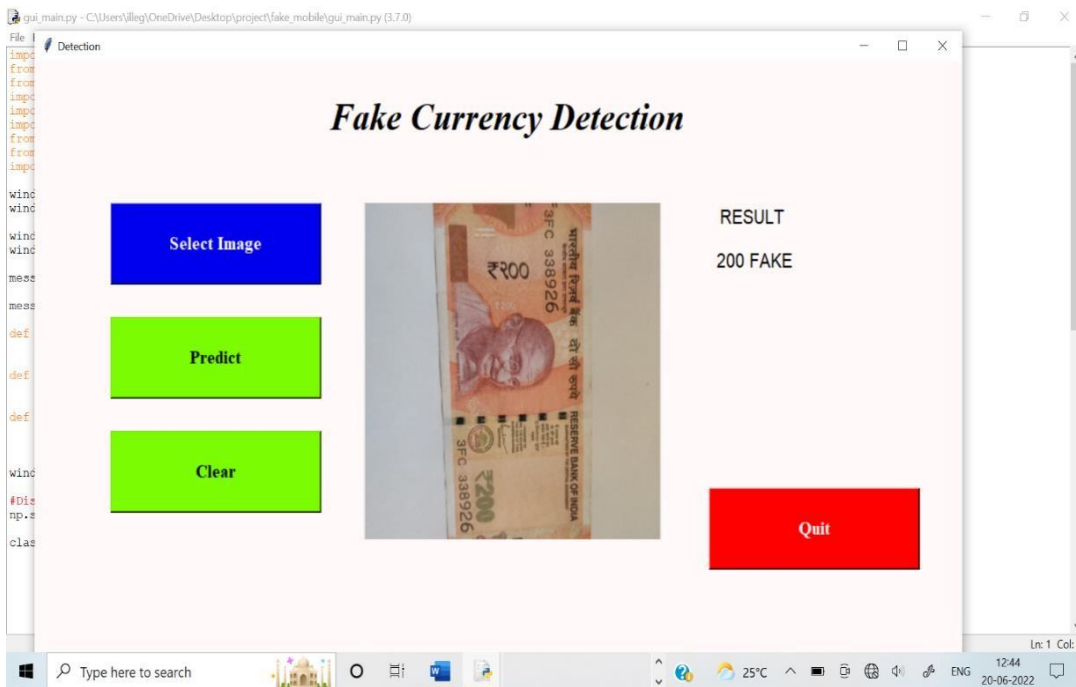


Figure A8: 200 Rupees Fake Note

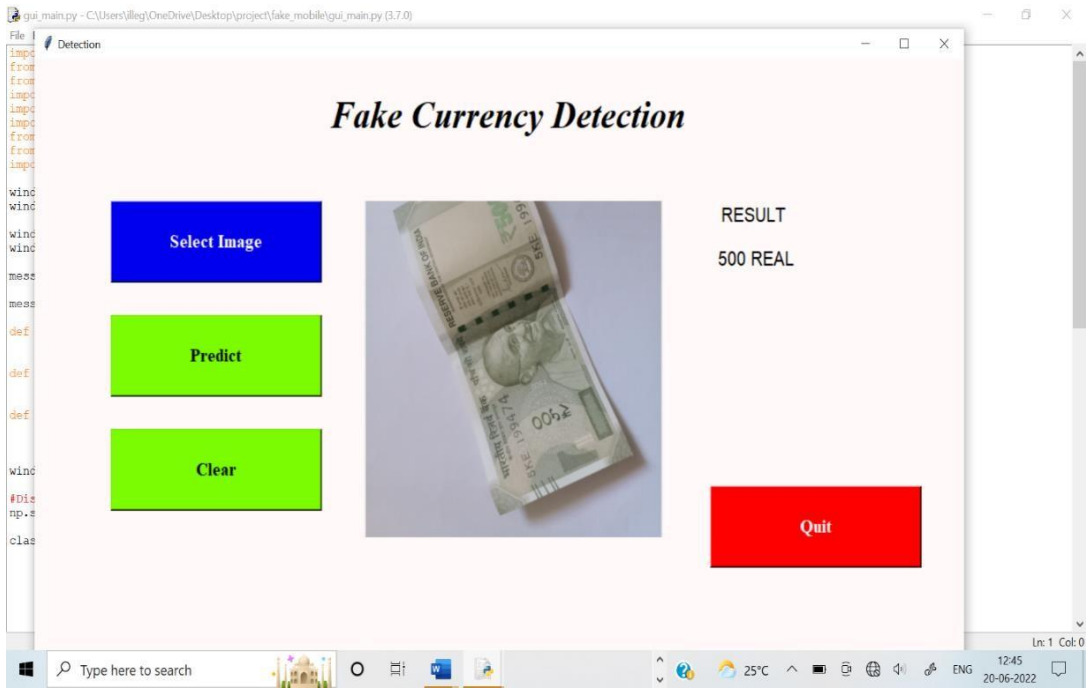


Figure A9: 500 Rupees Real Note

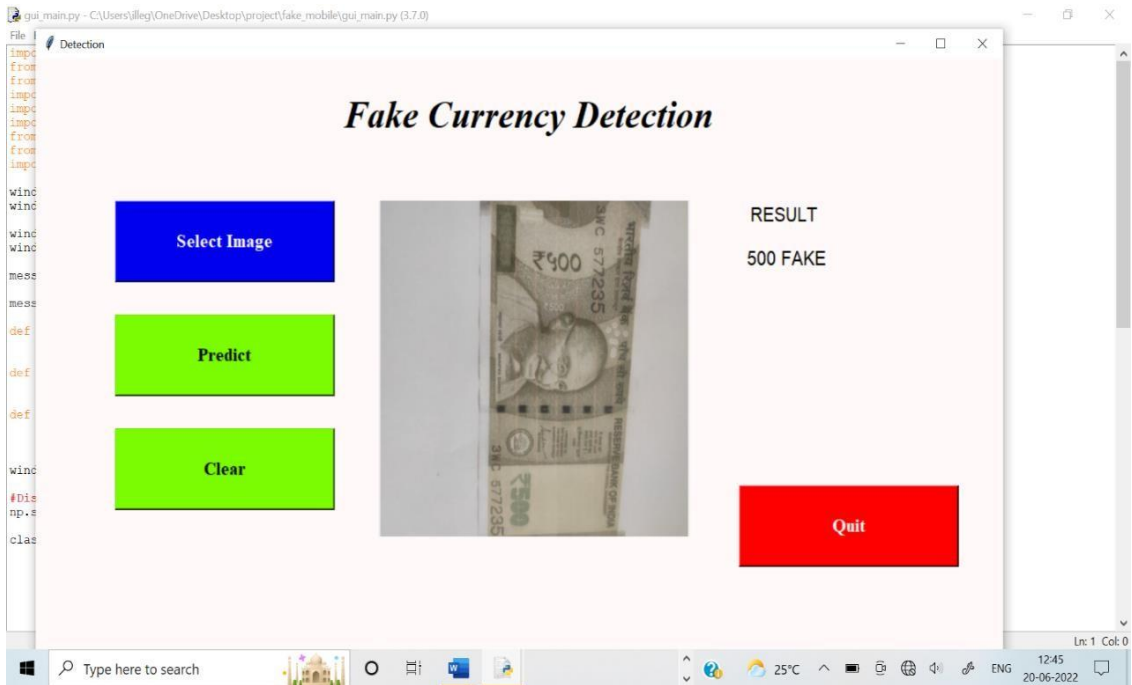


Figure A10: 500 Rupees Fake Note

Appendix B: Abbreviation

CNN: Convolutional Neural network

VGG: Visual Geometry group

ReLU: Rectifier Activation function

MATLAB: Matrix Laboratory

IDE: Integrated Development Environment

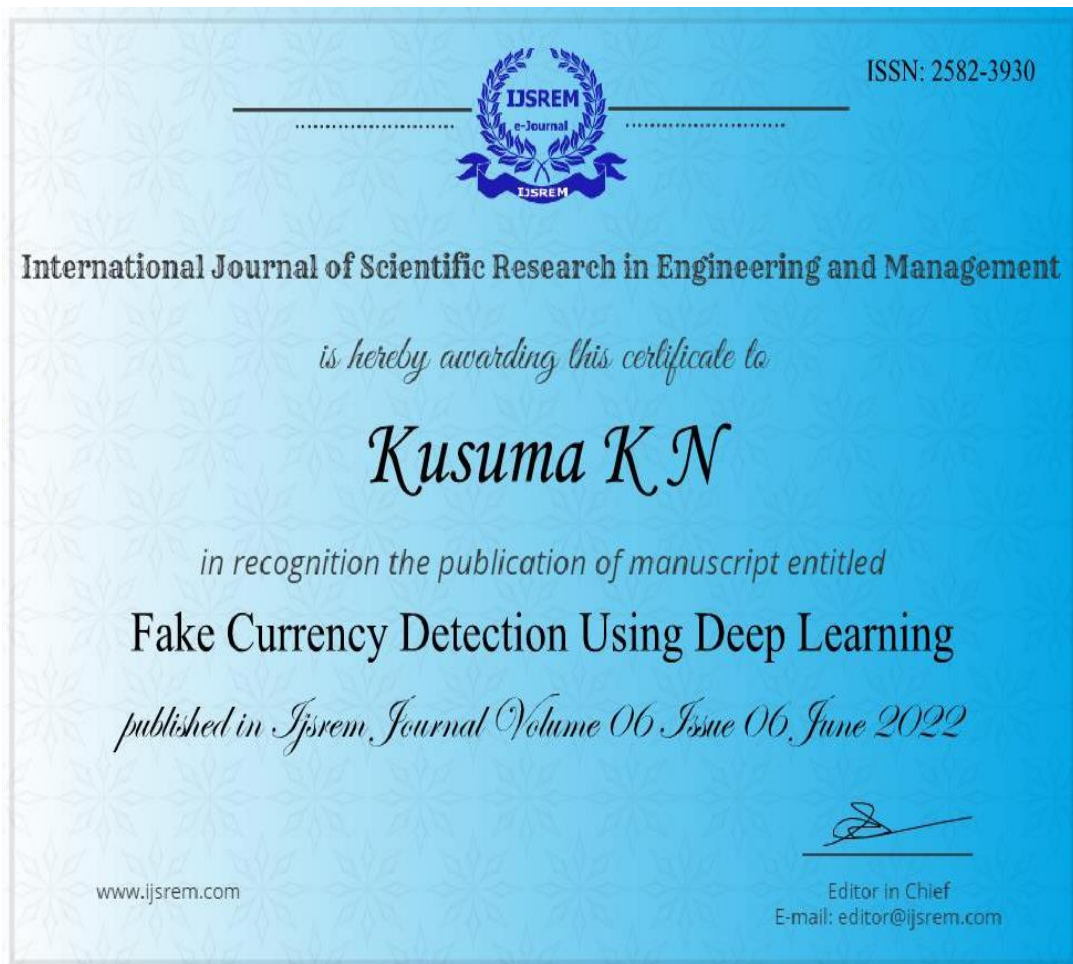
REPL: Read-Eval-Print Loop

GUI: Graphical user interface

PAPER PUBLICATION DETAILS

The paper has been published in International Journal of Scientific Research in Engineering and Management (IJSREM) journal Volume 06, Issue 06 June 2022 and entitled as “Fake Currency Detection using Deep Learning”. The authors of this paper are Kusuma K N, G V Keerthana, Geetha A, Gutta Sai Sandhya and Prof. Chandana K R.

This paper shows the insights about the project which consists of Abstract, Introduction, Literature Review, Methodology, Implementation, Conclusion and so on





ISSN: 2582-3930

International Journal of Scientific Research in Engineering and Management

is hereby awarding this certificate to

G V Keerthana

in recognition the publication of manuscript entitled

Fake Currency Detection Using Deep Learning

published in Ijsrem, Journal Volume 06, Issue 06, June 2022

www.ijsrem.com


Editor in Chief
E-mail: editor@ijsrem.com



ISSN: 2582-3930

International Journal of Scientific Research in Engineering and Management

is hereby awarding this certificate to

Geetha A

in recognition the publication of manuscript entitled

Fake Currency Detection Using Deep Learning

published in Ijsrem, Journal Volume 06, Issue 06, June 2022

www.ijsrem.com


Editor in Chief
E-mail: editor@ijsrem.com



ISSN: 2582-3930

International Journal of Scientific Research in Engineering and Management

is hereby awarding this certificate to

Gutta Sai Sandhya

in recognition the publication of manuscript entitled

Fake Currency Detection Using Deep Learning

published in Ijsrem Journal Volume 06 Issue 06, June 2022

www.ijsrem.com

Editor in Chief
E-mail: editor@ijsrem.com