A

Mini project On

Fake Currency Detection Using Machine Learning

A Dissertation Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

Electronics and Communication Engineering

By

TALLADA SAI SHARAN (Roll No: 20D41A04K5)
SHAIK MOHAMMED ALEEMUDDIN (Roll No: 20D41A04J4)
ALLAMLA SRIKANTH (Roll No: 18D41A0409)
KATTA SRI LAKSHMI PRASANNA (Roll No: 21D45A0420)

Under the esteemed guidance of

Ms. B. Deepika Rathod
Assoc. Prof

Department of Electronics and Communication Engineering

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
Recognized under 2(f) and 12(B) of UGC Act. 1956
Sheriguda village, Ibrahimpatnam, RR District – 501 510, T.S,INDIA

2023 - 2024

CERTIFICATE

This is to certify that the project report entitled **"FAKE CURRENCY DETECTION USING MACHINE LEARNING"** submitted by

Mr. TALLADA SAI SHARAN (Roll No: 20D41A04K5)
Mr. SHAIK MOHAMMED ALEEMUDDIN (Roll No: 20D41A04J4)
Mr. ALLAMLA SRIKANTH (Roll No: 18D41A0409)
Ms. KATTA SRI LAKSHMI PRASANNA (Roll No: 21D45A0420)

In partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** to the Jawaharlal Nehru Technological University, Hyderabad is a record of bonafide work carried out by them under our guidance and supervision

The results presented in this thesis have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other University for the award of any other degree or diploma.

MS. B . DEEPIKA RATHOD
Internal Guide

Dr. N.C. SENDHIL KUMAR
Head of the Department

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is my privilege and pleasure to express my profound sense of gratitude and indebtedness to my Project Guide **MS. B. DEEPIKA RATHOD, Assoc. Prof.** of Electronics and Communication Engineering Department, Sri Indu College of Engineering & Technology, for her guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work.

I take the opportunity to offer my humble thanks to **Dr. N.C. SENDHIL KUMAR, Prof. & Head of the Department,** Electronics & Communication Engineering, Sri Indu College of Engineering & Technology, for his encouragement and constant help.

I also thank **Dr. G.SURESH, Principal**, **SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY,** for his support in this Endeavour.

In addition I would like to thank all the **Faculty members** of Department of Electronics & Communication, & **Management,** who provided us with good lab facilities and helped us in carrying out the project successfully.

I finally thank my family members and friends for giving moral strength and support to complete this dissertation.

Tallada Sai Sharan (Roll No: 20D41A04K5)
Shaik Mohammed Aleemuddin (Roll No: 20D41A04J4)
Allamla Srikanth (Roll No: 18D41A0409)
Katta Sri Lakshmi Prasanna (Roll No: 21D45A0420)

ABSTRACT

The one important asset of our country is Bank currency and to create discrepancies of money miscreants introduce the fake notes which resembles to original note in the financial market. During demonetization time it is seen that so much of fake currency is floating in market. In general by a human being it is very difficult to identify forged note from the genuine not instead of various parameters designed for identification as many features of forged note are similar to original one. To discriminate between fake bank currency and original note is a challenging task. So, there must be an automated system that will be available in banks or in ATM machines. To design such an automated system there is need to design an efficient algorithm which is able to predict weather the banknote is genuine or forged bank currency as fake notes are designed with high precision. In this paper we are using CNN algorithm on dataset available on UCI machine learning repository for detection of Bank currency authentication. To implement this we have applied machine learning algorithms are measured their performance on the basis various quantitative analysis parameter.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

ATM – Automated Teller Machine
CNN – Convolutional Neural Network
DFD – Data Flow Diagram
GA – Genetic Algorithm
GUI – Graphical User Interface
ID – Identity Document
MATLAB – Matrix Laboratory
OO – Object Oriented
OOP – Object Oriented Programming
RBI – Reserve Bank of India
RGB – Red Green Blue
SVM – Support Vector Machine
UML – Unified Modeling Language

CHAPTER 1

INTRODUCTION

Duplicating money represents the unlawful replication of unique money, henceforth fake money is a phony cash that has not been approved by the administration. RBI is the main body which has sole duty to print cash notes in India. Consistently RBI faces the issue of fake money notes, once separated and flowed in the market. Counterfeit note discovery framework is created for perceiving counterfeit note from the certifiable. The main arrangement that is by and by accessible for basic man to recognize fake cash is Fake Note Detector Machine. This machine is for the most part accessible just in banks which aren't reachable each time by normal resident. Every one of these situations needs a sort of answer for average folks to pass judgment on a fashioned monetary certificate and to cease our money from losing its worth. The technique of picture preparing depends on the extraction of the highlights of Indian banknotes. Pictures are handled by utilizing different procedures of picture preparing and assist different highlights are extricated from the pictures. The methodology comprises of various segments including picture handling, trademark extraction, looking at pictures. The essential thing of approach is that we extricate the highlights based on which we will arrange the phony note. Security highlights of money are basic for deciding genuine and phony cash. Regular security highlights incorporate watermarks, idle pictures, security string, and optically factor ink. In the study, a methodology for counterfeit money detection separates the general characteristics dormant pictures and ID mark from the picture of money. Extricating traits from pictures of money notes can get very mind boggling as it includes the extraction of some obvious and undetectable highlights of Indian cash. After demonetization 500 and 2000 are the high esteemed money notes existing till date so there is a most extreme likelihood that this notes can be falsified so as to dodge this we are utilizing programming to recognize the phony notes utilizing picture handling system. Fake currency notes have led to reduction in the value of money and also loss on economic and social front. In the paper we are using Image Processing and Machine Learning to check the authenticity of the currency note. An android application will help people to easily detect the fake note as many people today carry smart phones and hence android application is not a difficult thing for people to handle.

1.1 PROBLEM DEFINITION

During demonetization time it is seen that so much of fake currency is floating in market. In general by a human being it is very difficult to identify forged note from the genuine not instead of various parameters designed for identification as many features of forged note are similar to original one. To discriminate between fake bank currency and original note is a challenging task. So, there must be an automated system that will be available in banks or in ATM machines. To design such an automated system there is need to design an efficient algorithm which is able to predict weather the banknote is genuine or forged bank currency as fake notes are designed with high precision.

SYSTEM REQUIREMENTS:

1.2 HARDWARE REQUIREMENTS

System	: Pentium i3 Processor
Hard Disk	: 500 GB.
Monitor	: 15'' LED
Input Devices	: Keyboard, Mouse
Ram	: 2 GB

1.3 SOFTWARE REQUIREMENTS

Operating system : Windows 10 Coding

Language : Python

1.4 LANGUAGE SPECIFICATION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language.

1.5 HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

1.6 APPLICATION AND FEATURES OF PYTHON

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.

Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Jav

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

1. **A.A.Mandankandy, K.E. Kannammal, Fake currency detection: a survey. Gedrag en Organisatie 33(4), 622–638 (2020).**

   Imitate something authenticate is called as counterfeit. In banking sector, counterfeit currency is a big threat still. There are lots of detection methods are available, but with the advent of freely available image operation tools, it’s a serious issue in banking sector. There are lots of important regions which is present in currencies, finds out those for evaluation is the basic task. Classifiers can find out the extracted features either genuine or fake. Without classifiers we can cross check with the original note’s region with the segmented currency image. But that alone can’t help to identify the authentication of the particular image. . If the number of interruption is zero, if it is real note otherwise it is fake. And also we calculate the entropy of the currency notes for the efficient detection of fake currency note.

Alignment and edges may not be same if we segment the important portions, so fake currency note image(s) may be considered as original in some cases. To avoid that extracted features has to be process by classifier(s) to get better results. In this paper, the comparative study of image segmentation or thresholding, feature extraction, classification and finally selection approaches also included. And also added some analysis work which is possible based on some existing methods.

2. **2S. Arya, M. Sasikumar, Fake currency detection, in 2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC), Feb 2020.**

Fake currency notes are increasing day by day, in order to overcome this we proposes a very helpful and efficient system to detect the fake currency. For detecting the fake currency note is done by counting the number of interruptions in the thread line. For predicting the note is real or fake on the basis of number of interruptions. If the number of interruption is zero, if it is real note otherwise it is fake. And also we calculate the entropy of the currency notes for the efficient detection of fake currency note. MATLAB software is used to detect the fake currency note.

3. **A. Singh, K. Bhoyar, A. Pandey, P. Mankani, A.Tekriwal, Detection of fake currency using image processing. Int. J. Eng. Res. Technol. (IJERT) (2019).**

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. We are using image processing and cloud storage to make this app portable and efficient.

4. **S. Shaker, M.G. Alawan, Paper currency detection based image processing techniques: a review paper**

The currency has a great meaning in everyday life. Thus currency recognition has gained a great interest for many researchers. The researchers have suggested diverse approaches to improve currency recognition. Based on strong literature survey, image processing can be considered as the most widespread and effective technique of currency recognition. This paper introduces some close related works of paper-currency recognition. This paper has explained a variety of different currency recognition systems. The applications have used the power of computing to differentiate between different types of currencies with the appropriate layer. Choosing the proper feature would improve overall system performance. The main goal of this work is to compare previous papers and literatures through reviews these literatures and identify the advantages and disadvantage for each method in these literatures. The results were summarized in a comparison table that presented different ways of reviewing the technology used in image processing to distinguish currency papers.

5. A. Upadhyaya, V. Shokeen, G. Srivastava, Analysis of counterfeit currency detection techniques for classification model (2018).

Counterfeit currency is one of the threats which creates vice to nation's economy and hence impacts the growth worldwide. Producing forge currency or fabricating fake features in the currency considered to be a crime.

Currency crime comes under the criminal law and known to be as Economical crime. Over the past few years many researchers have proposed various techniques to identify and detect forged currency. The serious problem has been come up with variety of solutions in terms of hardware related techniques, Image processing and machine learning methods.

Advancements in printing and scanning technology, trading of material are some of the problems in germinating counterfeit currency. The study presents various fake currency detection techniques proposed by various researchers. The review highlighted the methodology implemented on particular characteristics feature with success rate of each method to detect counterfeited currency. Moreover, the study includes the analysis of widely acceptable statistical classification technique for currency authentication. The comparative analysis of Logistic Regression and Linear Discriminant Analysis (LDA) was performed to realize the better model for currency authentication. It has been found that classification Model using Logistic regression shows better accuracy of 99% then LDA. The study will benefit the reader in identifying most feasible technique to be implemented based on the accuracy rate.

6. T. Kumar, T. Subhash, D. Regan, Fake currency recognition system for Indian notes using image processing techniques (2019).

In the point of economic stability of a nation, circulation and use of the fake currency notes pose major threats. Curbing the use of fake currency notes nowadays becomes digitalized with use of digital image processing algorithms. Counterfeit notes are printed with the utmost precision level to par with the original. So fake currency detection is a difficult task by simple visual inspection and use of digital image processing algorithms come to play a vital role. The conceivable arrangements are there, to utilize either chemical properties of the currency or to utilize its physical appearance for detection. The methodology exhibited in this paper depends on physical appearance of the Indian currency. Image processing algorithms have been embraced to expose the highlights of Indian currency notes, for example, security thread, intaglio printing (RBI logo) and distinguishing proof imprint, which have been received as security highlights of Indian currency. To make the framework increasingly robust and exact, the definitive score of all the three highlights has been intertwined to separate among genuine and fake monetary standards. Another parameter used to quantify the execution of the proposed framework is mean square error, which is roughly 1%. It might be embraced by the everyday citizens too, who frequently face the issue of separating among genuine and fake monetary standards

7. S. Gothe, K. Naik, V. Joshi, Fake currency detection using image processing and machine learning (2018).

With the increase in the technology, the convenience and ease of people to carry out various task is increasing on a large scale. But with the advancement in technology, the amount of crime carried out due to wrong use of these technologies is also increasing on a large scale. Similar thing applies to the currency notes being handled by us on day to day basis. Fake currency notes are made so accurate that finding out which one is real note and which one is fake becoming increasingly difficult. Fake currency note is the imitation of the authentic currency note for wrong purposes and without the permission of the state and central government. Hence with the advancement in the development of fake currency notes, the detection mechanism needs to be developed as well to reduce its flow in the market.

Fake currency notes have led to reduction in the value of money and also loss on economic and social front. In the paper we are using Image Processing and Machine Learning to check the authenticity of the currency note. An android application will help people to easily detect the fake note as many people today carry smart phones and hence android application is not a difficult thing for people to handle. This led us to satisfy our purpose of our application being helpful to common citizen of India. Key wordsFake Currency Detection, Counterfeit Detection

8. V. Lalithendra Nadh, G. Syam Prasad, Support vector machine in the anticipation of currency markets. Int. J. Eng. Technol. (UAE) 7(2), 66–68 (2018).

Various researchers have done an expansive research within the domain of stock market anticipation. The majority of the anticipated models is confronting some pivotal troubles because of the likelihood of the market. Numerous normal models are accurate when the data is linear. In any case, the expectation in view of nonlinear data could be a testing movement. From past twenty years with the progression of innovation and the artificial intelligence, including machine learning approaches like a Support Vector Machine it becomes conceivable to estimate in light of nonlinear data. Modern researchers are combining GA (Genetic Algorithm) with SVM to achieve highly precise outcomes. This analysis compares the SVM and ESVM with other conventional models and other machine learning methods in the domain of currency market prediction. Finally, the consequence of SVM when compared with different models it is demonstrated that SVM is the premier for foreseeing.

9. M. Laavanya, V. Vijayaraghavan, Real time fake currency note detection using deep learning. Int. J. Eng. Adv. Technol. (IJEAT) 9(1S5) (2019). ISSN: 2249-8958.

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. Once the feature map is learnt the network is ready for identifying the fake currency in real time. The proposed approach efficiently identifies the forgery currencies of 2000, 500, 200, and 50 with less time consumption.

10. A.Vidhate, Y. Shah, R. Biyani, H. Keshri, R. Nikhare, Fake currency detection application. Int. Res. J. Eng. Technol. (IRJET) 08(05) (2021). e-ISSN: 2395-0056

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. 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 are going to detect the variation in barcode among the real and fake one and also, we will find out dissimilarities between the image under consideration and the prototype. CNN classifiers will be used to detect fake currency. The proposed app for fake currency detection will be simple, accurate and easy to use.

11. T. Agasti, G. Burand, P. Wade, P. Chitra, Fake currency detection image processing, ICSET-2017. IOP Conf. Ser. Mater. Sci. Eng. 263, 052047 (2017).

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. This article describes extraction of various features of Indian currency notes. MATLAB software is used to extract the features of the note. The proposed system has got advantages like simplicity and high performance speed. The result will predict whether the currency note is fake or not.

CHAPTER 3
ML ALGORITHM

3.1 CNN Algorithm

Convolutional Neural Network is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used.

CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. The computer sees an image as an array of pixels and depends on the resolution of the image. Based on image resolution, it will see as **h * w * d**, where h= height w= width and d= dimension. For example, An RGB image is **6 * 6 * 3** array of the matrix, and the grayscale image is **4 * 4 * 1** array of the matrix.

Convolution Layer

Convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. Convolution is a simple mathematical operation which is fundamental to many common image processing operators.

Convolution provides a way of `multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. Convolution has applications that include probability, statistics, acoustics, spectroscopy, signal processing and image processing, geophysics, engineering, physics, computer vision and differential equations. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

The dimension of the image matrix is $\mathbf{h \times w \times d}$.

The dimension of the filter is $\mathbf{f_h \times f_w \times d}$.

The dimension of the output is $\mathbf{(h-f_h+1) \times (w-f_w+1) \times 1}$.

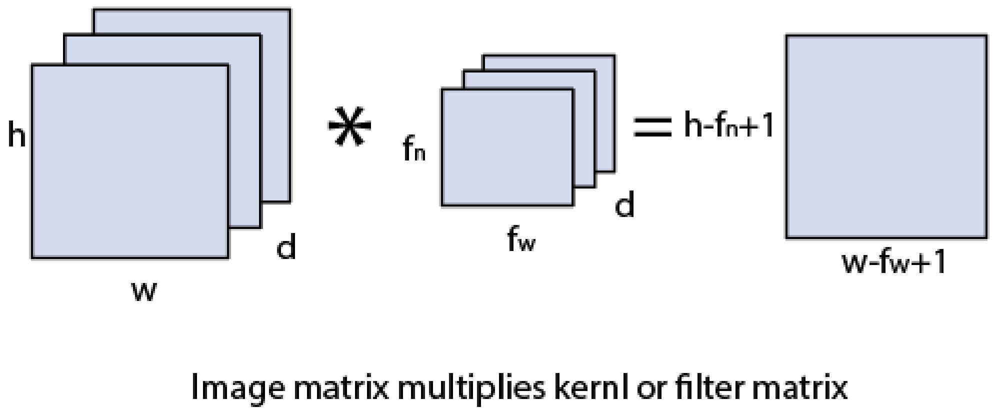

Figure 3.1 Image matrix Multiples Kernel

Let's start with consideration a 5*5 image whose pixel values are 0, 1, and filter matrix 3*3 as:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$5 \times 5 - \textbf{Image Matrix}$ $\quad$ $3 \times 3 - \textbf{Filter Matrix}$

Figure 3.2 Image And Filter Matrix

The convolution of 5*5 image matrix multiplies with 3*3 filter matrix is called "**Features Map**" and show as an output.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Convolved Feature

Figure 3.3 Convolved Feature

Convolution of an image with different filters can perform an operation such as blur, sharpen, and edge detection by applying filters.

Strides

Stride is the number of pixels which are shift over the input matrix. When the stride is equaled to 1, then we move the filters to 1 pixel at a time and similarly, if the stride is equaled to 2, then we move the filters to 2 pixels at a time. The following figure shows that the convolution would work with a stride of 2.

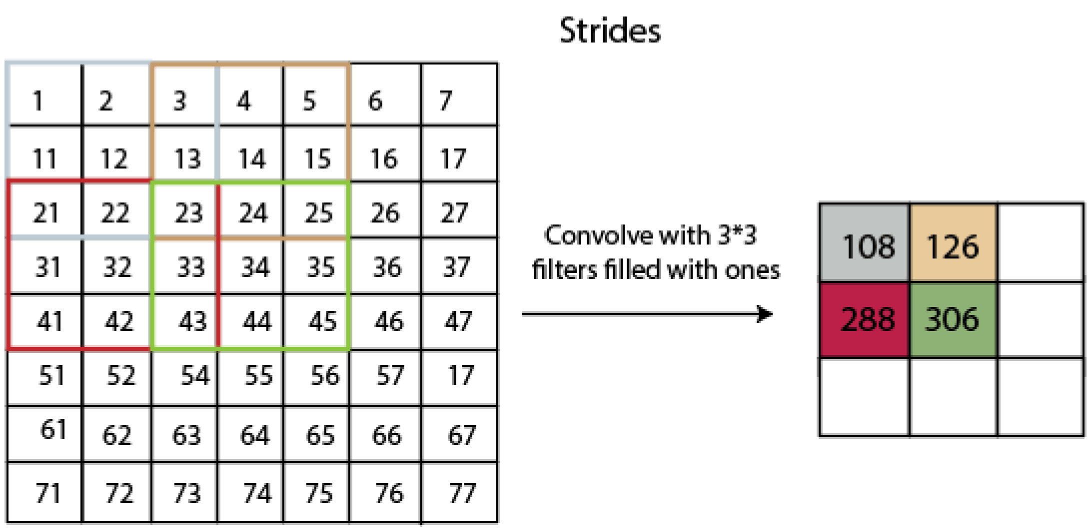

Figure 3.4 Strides

Padding

Padding plays a crucial role in building the convolutional neural network. If the image will get shrink and if we will take a neural network with 100's of layers on it, it will give us a small image after filtered in the end.

If we take a three by three filter on top of a grayscale image and do the convolving then what will happen?

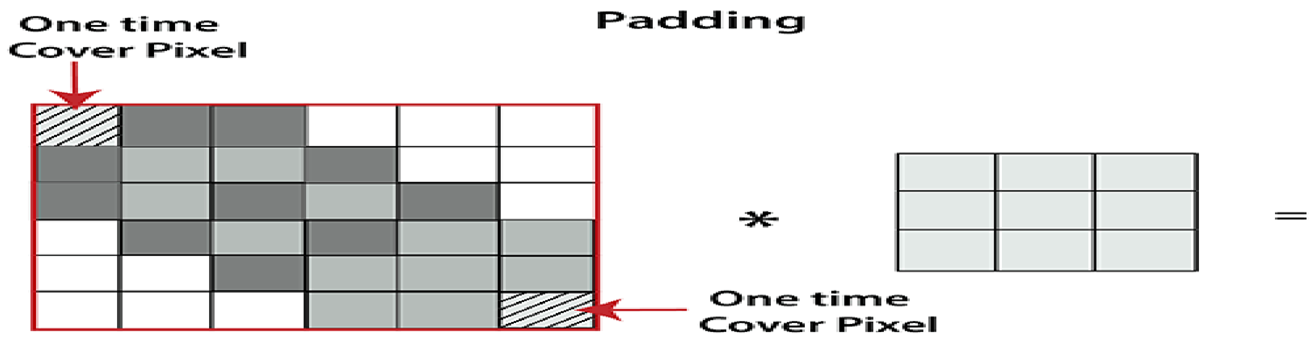

Figure 3.5 Padding

It is clear from the above picture that the pixel in the corner will only get covers one time, but the middle pixel will get covered more than once. It means that we have more information on that middle pixel, so there are two downsides:

- Shrinking outputs
- Losing information on the corner of the image.

To overcome this, we have introduced padding to an image. **"Padding is an additional layer which can add to the border of an image."**

Pooling Layer

Pooling layer plays an important role in pre-processing of an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "**downscaling**" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called down sampling or subsampling, which reduces the dimensionality of

each map but retains the important information. There are the following types of spatial pooling:

Max Pooling

Max pooling is a **sample-based discretization process**. Its main objective is to downscale an input representation, reducing its dimensionality and allowing for the assumption to be made about features contained in the sub-region binned.

Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.

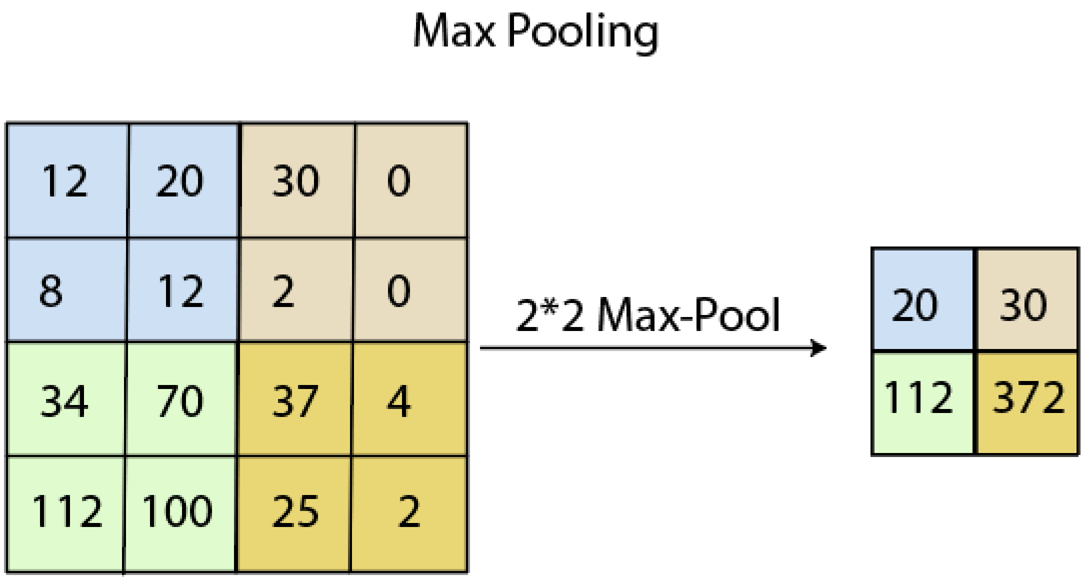

Figure 3.6 Max Pooling

Average Pooling

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

Syntax

layer = averagePooling2dLayer(poolSize)

layer = averagePooling2dLayer(poolSize,Name,Value) Sum

Pooling

The sub-region for **sum pooling** or **mean pooling** are set exactly the same as for **max-pooling** but instead of using the max function we use sum or mean.

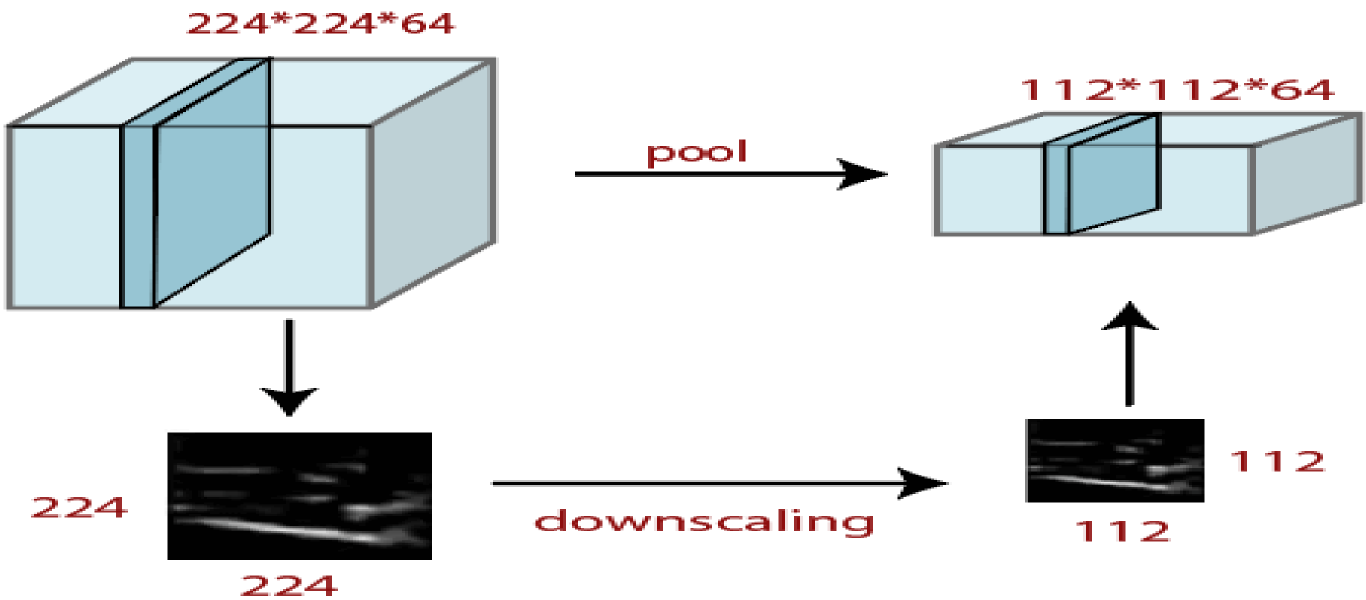

Figure 3.7 Downscaling

Fully Connected Layer

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

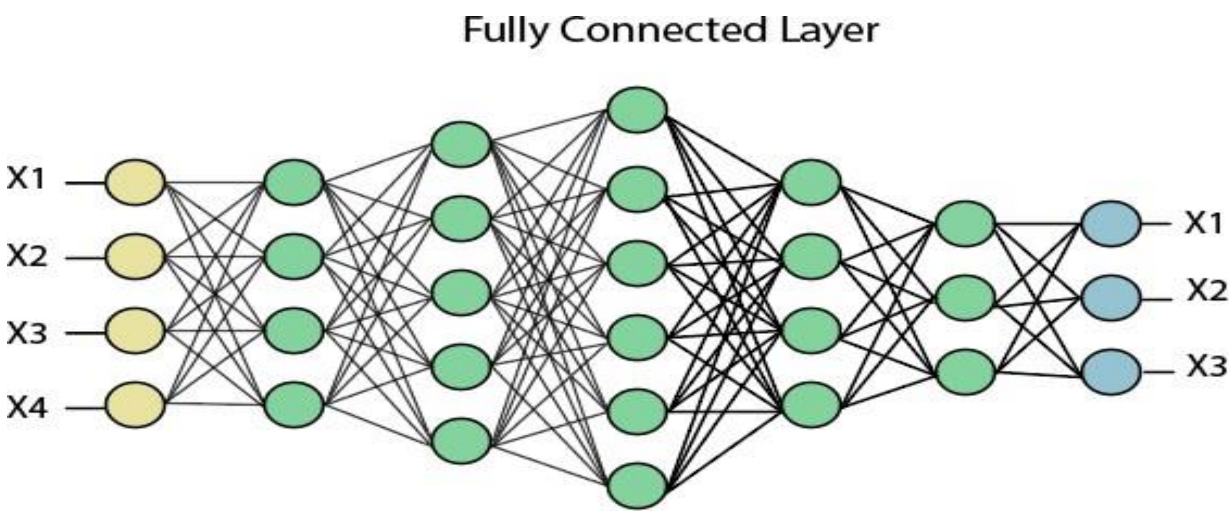

Figure 3.8 Fully Connected Layer

In the figure 3.8, the feature map matrix will be converted into the vector such as **x1, x2, x3... xn** with the help of fully connected layers. We will combine features to create a model and apply the activation function such as **softmax** or **sigmoid** to classify the outputs as a car, dog, truck, etc like in figure 3.9.

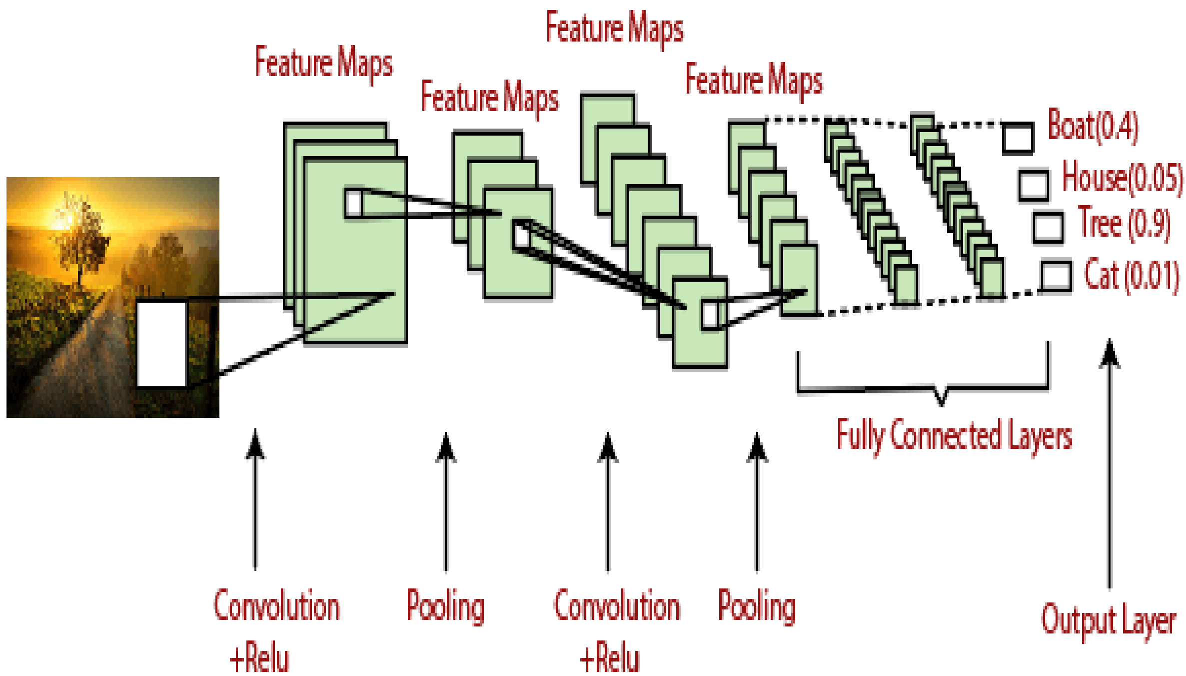

Figure 3.9 Classifications

CHAPTER 4

SYSTEM ANALYSIS, MODULES DESCRIPTION AND TESTING

PURPOSE

The purpose of this work is detection of fake bank currency using machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

4.1 SCOPE

The scope of this SRS document persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRS to the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

4.2 EXISTING SYSTEM

Indian is a developing country, Production and printing of Fake .In this article, recognition of paper currency with the help of digital image processing techniques is described. Around eight characteristics of Indian paper currency is selected for counterfeit detection. The identification marks, optical variable link, see through register and currency color code decides the currency recognition. The security threads, water mark, Latent image and micro-lettering features are used for currency verification. The characteristics extraction is performed on the image of the currency and it is compared with the characteristics of the currency.

.

4.3 PROPOSED SYSTEM

In This System, fake currency detection is a major issue around the world, influencing the economy of pretty much every nation including India. The utilization of fake money is one of the significant issues looked all through the world now days. This paper deals with the matter of identifying the currency that if the given sample of bank currency is fake. Different traditional strategies and methods are available for fake bank currency identification. In general by a humanbeing it is very difficult to identify forged note from the genuine not instead of various parameters designed for identification as many features of forged note are similar to original one. To discriminate between fake bank currency and original note is a challenging task.

4.4 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

1. What data should be given as input?
2. How the data should be arranged or coded?
3. The dialog to guide the operating personnel in providing input.
4. Methods for preparing input validations and steps to follow when error occur.

4.5 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

1. Convey information about past activities, current status or projections of the
2. Future.
3. Signal important events, opportunities, problems, or warnings.
4. Trigger an action.
5. Confirm an action

4.6 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail. As shown in figure 5.3.1.

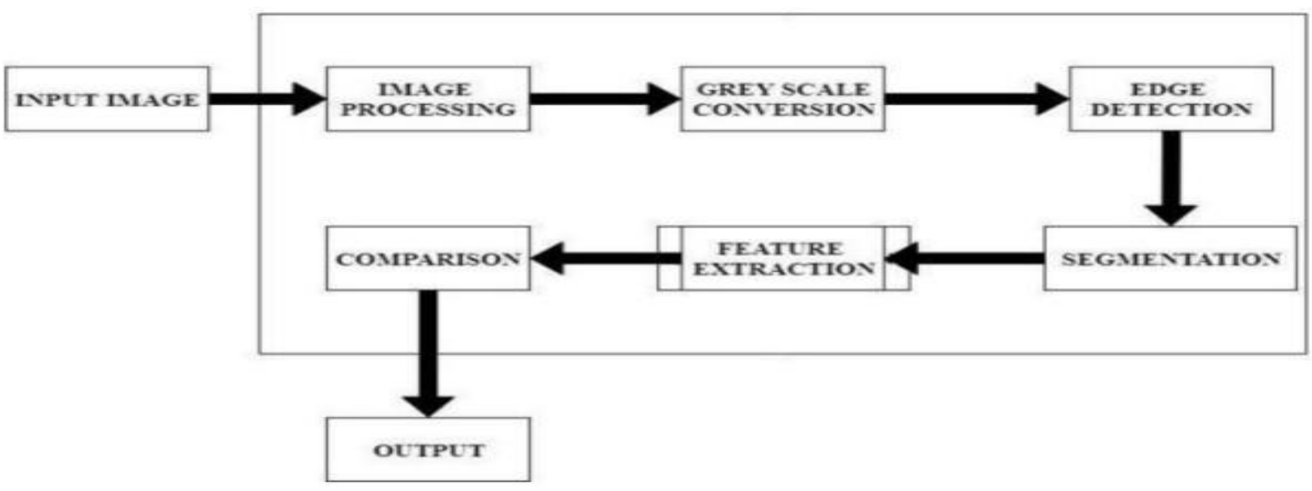

Figure 5.3.1 Data Flow Diagram

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

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. As shown in figure 5.3.2.

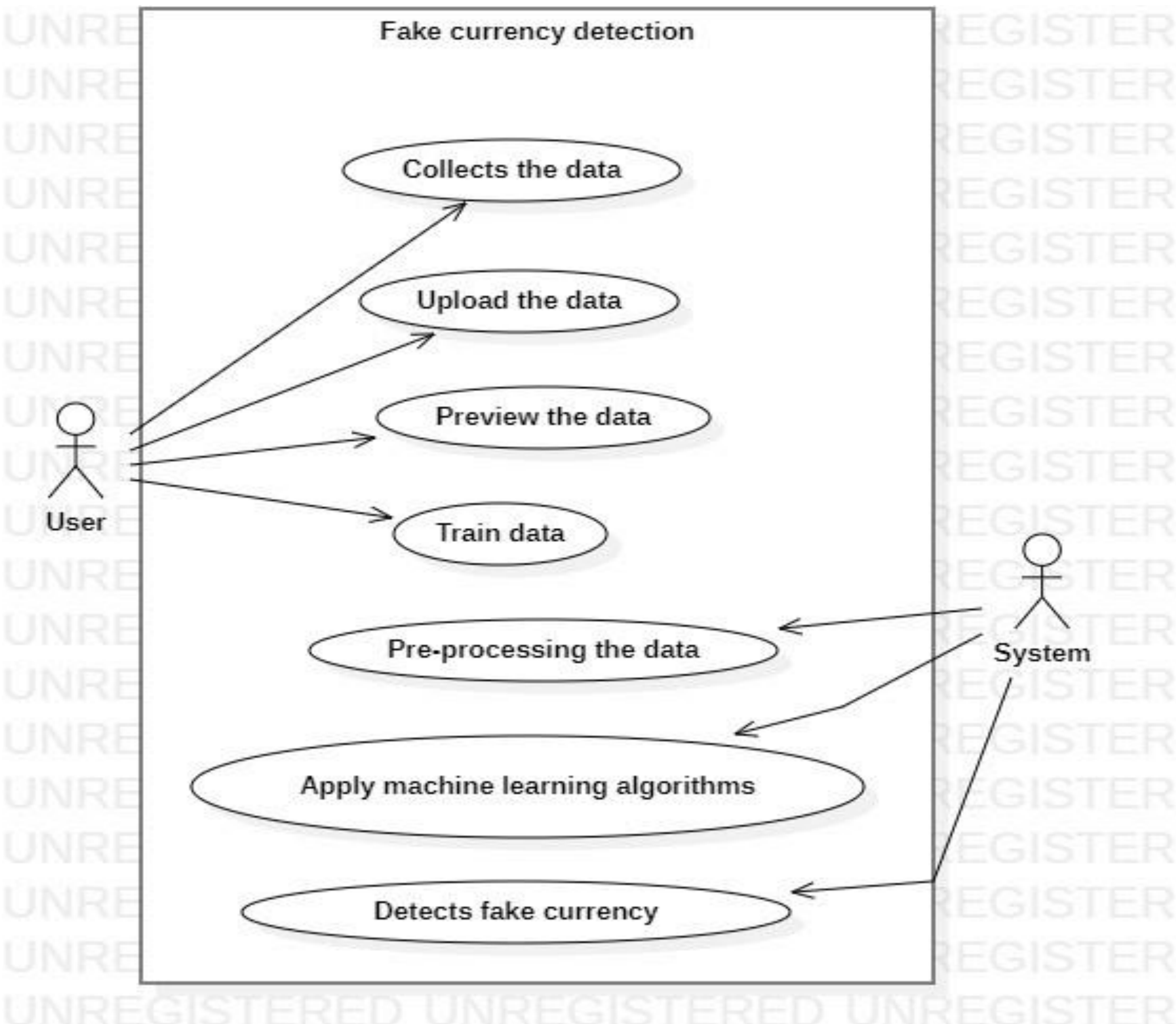

Figure 5.3.2 Use Case Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. As shown in figure 5.3.3.

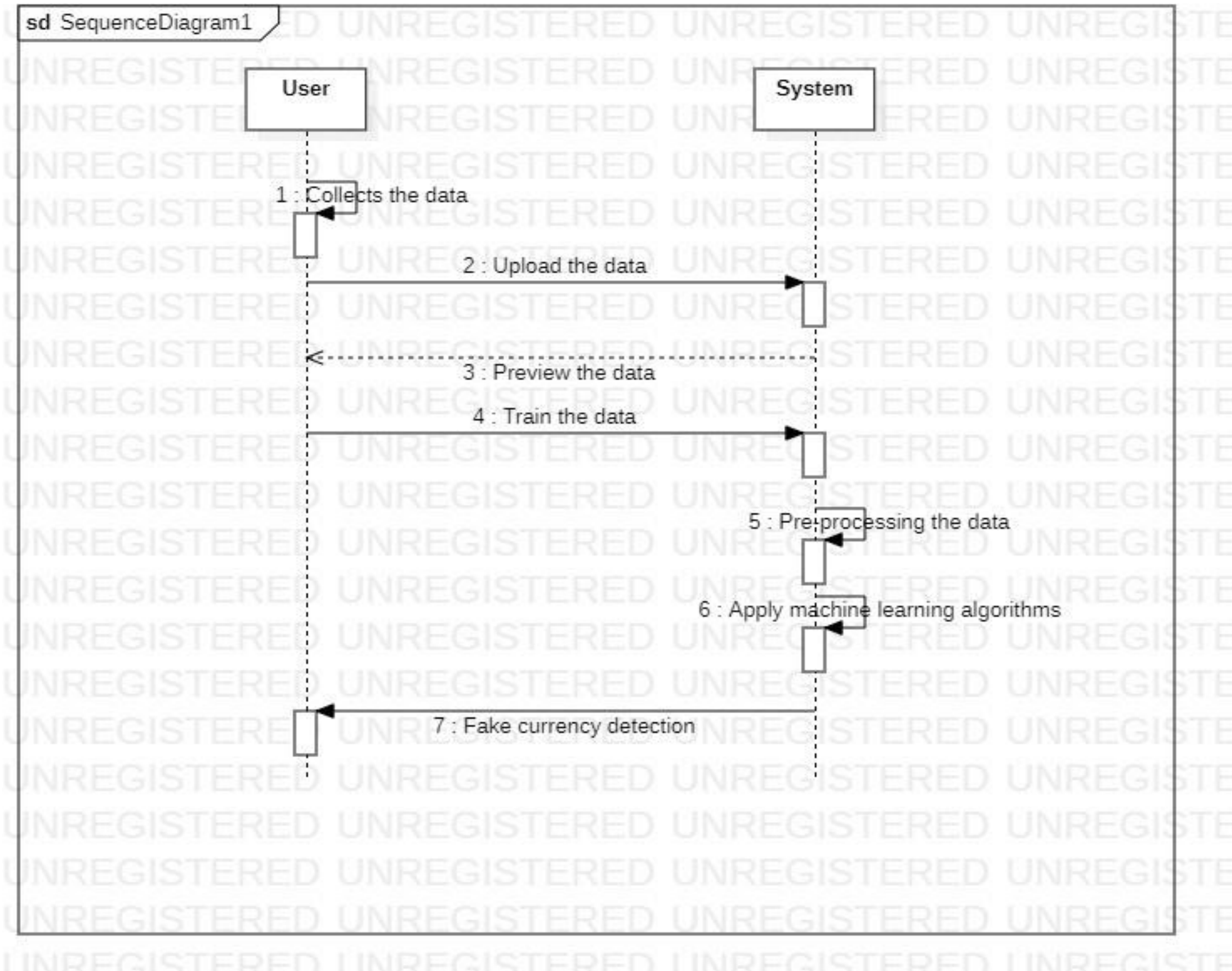

Figure 5.3.3 Sequence Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. As shown in figure 5.3.4.

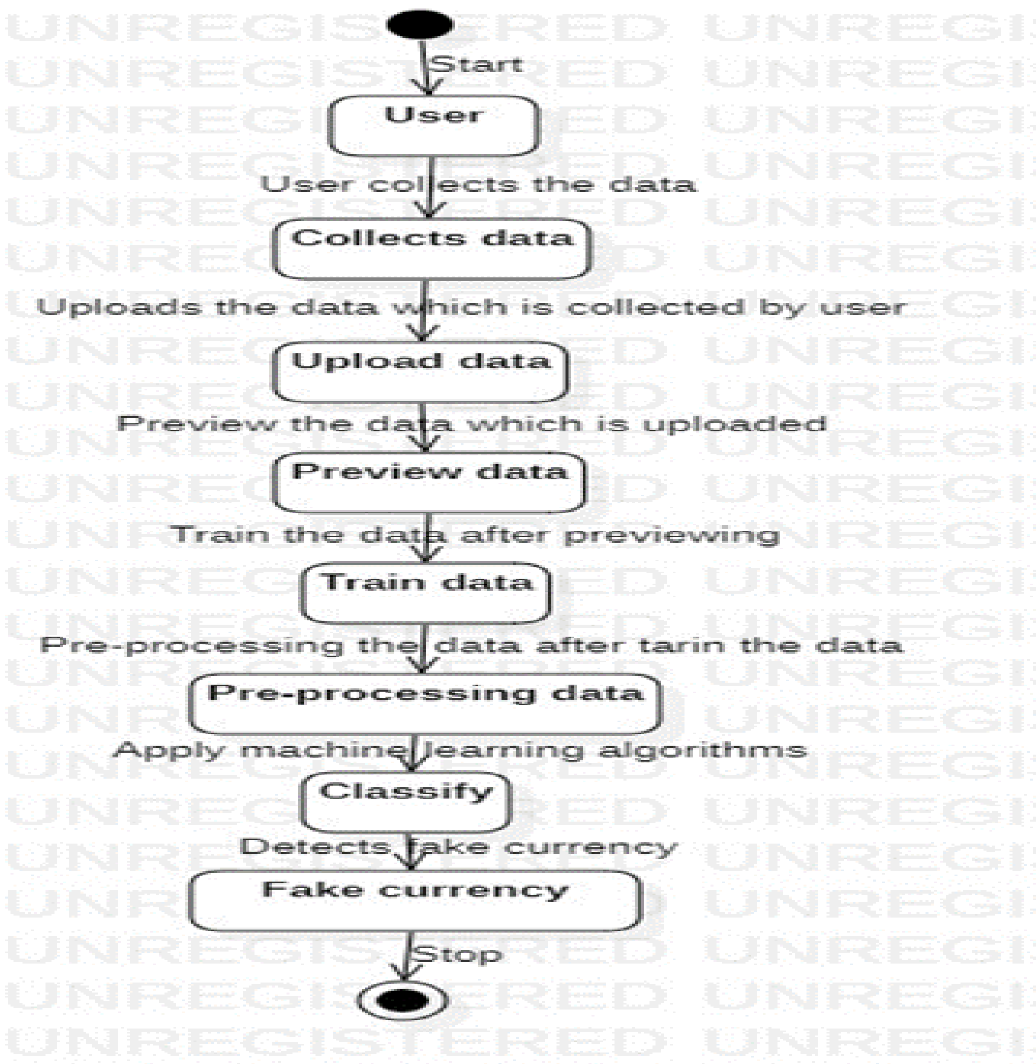

Figure 5.3.4 Activity Diagram

MODULES DESCRIPTION

4.7 MODULES

Data Collection Pre-

Processing Feature

Extraction

Detect Fake Currency

MODULE DESCRIPTION

4.7.1 Data Collection Module

The different categories of Indian currencies differs in value estimation and color usage, separated from the quality of printing ,material used for printing and other which makes for simple visual distinguishing proof. In any case, for the visually disabled person, the content and color will not give the assistance at all and measure can lead to disarray since of the comparable measurements of the different coins.

4.7.2 Pre-Processing Module

In pre-processing the operations normally initial to main data analysis and extraction of information. In this unwanted distortion are suppressed and enhance some image features that are important to further processing. It includes image adjusting and image smoothening. After these two pre-processing steps, the images of the currency were applied for feature extraction.

4.7.3 Feature Extraction

Feature extraction employs the selection and extraction of some of the Effective and important features, among the largest data set of the features which are extremely important for the recognition of fake currency. Some Features of an image are Latent image and Identification Mark. We first create a database of a number of authentic

Indian notes and then extract their features. The extracted features are used for detection of fake currency.

4.7.4 Detect Fake Currency Module

In this work six supervised machine learning algorithms are applied on dataset available on UCI machine learning repository for detection of Bank currency authentication. To implement this we have applied machine learning algorithms are measured their performance on the basis various quantitative analysis parameter. And some of ML algorithm are giving better accuracy for particular train test ratio.Below table gives the information about Advantages and Disadvantages of Machine Learing.

Advantages	**Disadvantages**
1)Machine learning enables automation of complex tasks, reducing manual effort and enhancing efficiency.	1) ML models heavily rely on quality data, and biased or incomplete datasets can result in inaccurate predictions.
2) ML algorithms can analyze vast datasets, leading to more accurate predictions and decision-making.	2) Implementing and maintaining ML systems can be complex, requiring specialized knowledge and resources.

Table Advantages and Disadvantages of ML

SYSTEM TESTING

4.8 Basics of software testing

There are two basics of software testing: black box testing and white box testing.

4.8.1 Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

4.8.2 White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

4.9 Types of testing

There are many types of testing like

1. Unit Testing
2. System Testing
3. Usability Testing
4. Acceptance Testing
5. Regression Testing

4.9.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

4.9.2 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

4.9.3 Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

4.9.4 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

4.9.5 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

CODE

Pre Processing : import

```
numpy as np
import cv2
import matplotlib.pyplot as plt
import joblib
import cvutils
import os
import sys
import tkinter as tk
from tkinter import filedialog


# Extracting features from training images def
trainproc():
    train_imgs1 = cvutils.imlist("train_images\\500")
    train_imgs2 = cvutils.imlist("train_images\\2000") k = 0
    for tr in train_imgs1: pth
        = tr

        # Reading the image out =
        "train\\500"
        img = cv2.imread(pth)

        # resizing
        img = cv2.resize(img, (1200, 512), interpolation=cv2.INTER_LINEAR)

        # Denoising image
        img = cv2.fastNlMeansDenoisingColored(img, None, 10, 10, 7, 21)

        # Converting to grayscale
        img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

        # compute the median of the single channel pixel intensities v =
        np.median(img)
        sigma = 0.33
        # apply automatic Canny edge detection using the computed median lower =
        int(max([ 0, (1.0 - sigma) * v ]))
        upper = int(min([ 255, (1.0 + sigma) * v ])) img =
        cv2.Canny(img, lower, upper)

        # Extracting features
        id1 = img[ 195:195 + 170, 190:190 + 85 ]


        id2 = img[ 330:330 + 105, 720:720 + 105 ]
```

```
    id3 = img[ 320:320 + 90, 865:865 + 205 ]
    id4 = img[ 250:250 + 40, 1120:1120 + 40 ]
    id5 = img[ 5:5 + 405, 660:660 + 40 ]
    id6 = img[ 284:284 + 132, 1090:1090 + 90 ]

    # Saving the features
    ids = [ id1, id2, id3, id4, id5, id6 ] out1
    = "\\demo" + str(k) + ".jpg" d = 1
    for i in ids:
        cv2.imwrite(out + "\\id%d" % d + out1, i) d = d +
        1
    k = k + 1

k = 0
for tr in train_imgs2: pth
    = tr

    # Reading the image out
    = "train\\2000" img =
    cv2.imread(pth)

    # resizing
    img = cv2.resize(img, (1200, 512), interpolation=cv2.INTER_LINEAR)

    # Denoising image
    img = cv2.fastNlMeansDenoisingColored(img, None, 10, 10, 7, 21)

    # Converting to grayscale
    img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

    # compute the median of the single channel pixel intensities v =
    np.median(img)
    sigma = 0.33
    # apply automatic Canny edge detection using the computed median lower =
    int(max([ 0, (1.0 - sigma) * v ]))
    upper = int(min([ 255, (1.0 + sigma) * v ])) img =
    cv2.Canny(img, lower, upper)

    # Extracting features

        id1 = img[ 195:195 + 165, 225:225 + 55 ]
        id2 = img[ 330:330 + 95, 760:760 + 90 ]
        id3 = img[ 335:335 + 80, 890:890 + 205 ]
```

```
        id4 = img[ 255:255 + 25, 1105:1105 + 53 ]
        id5 = img[ 10:10 + 480, 726:726 + 35 ]
        id6 = img[ 280:280 + 140, 1100:1100 + 75 ]

        # Saving the features
        ids = [ id1, id2, id3, id4, id5, id6 ] out1
        = "\\demo" + str(k) + ".jpg" d = 1
        for i in ids:
            cv2.imwrite(out + "\\id%d" % d + out1, i) d = d +
            1
        k = k + 1


# Extracting features from test image# Extracting features from test image def testproc():
    root = tk.Tk()
    pth = tk.filedialog.askopenfilename() root.destroy()
    out1 = "test\\ids1"
    out2 = "test\\ids2"

    # Reading the image img =
    cv2.imread(pth)
    cv2.imshow('qq', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

    # resizing
    img = cv2.resize(img, (1200, 512), interpolation=cv2.INTER_LINEAR) cv2.imshow('qq', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

    # Denoising image
    img = cv2.fastNlMeansDenoisingColored(img, None, 10, 10, 7, 21)

    # Converting to grayscale

    img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    cv2.imshow('qq', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
# compute the median of the single channel pixel intensities v =
```

```
np.median(img)
sigma = 0.33
# apply automatic Canny edge detection using the computed median lower =
int(max([ 0, (1.0 - sigma) * v ]))
upper = int(min([ 255, (1.0 + sigma) * v ])) img =
cv2.Canny(img, lower, upper) cv2.imshow('qq',
img)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Extracting features
id1 = img[ 195:195 + 170, 190:190 + 85 ]
id2 = img[ 330:330 + 105, 720:720 + 105 ]
id3 = img[ 320:320 + 90, 865:865 + 205 ]
id4 = img[ 250:250 + 40, 1120:1120 + 40 ]
id5 = img[ 5:5 + 405, 660:660 + 40 ]
id6 = img[ 284:284 + 132, 1090:1090 + 90 ]
ids1 = [ id1, id2, id3, id4, id5, id6 ] #

Saving the features

id1 = img[ 195:195 + 165, 225:225 + 55 ]
id2 = img[ 330:330 + 95, 760:760 + 90 ]
id3 = img[ 335:335 + 80, 890:890 + 205 ]
id4 = img[ 255:255 + 25, 1105:1105 + 53 ]
id5 = img[ 10:10 + 480, 726:726 + 35 ]
id6 = img[ 280:280 + 140, 1100:1100 + 75 ]
ids2 = [ id1, id2, id3, id4, id5, id6 ]

d = 1
for i in ids1:
    cv2.imwrite(out1 + "\\test%d.jpg" % d, i) d = d +
    1
d = 1
for i in ids2:

    cv2.imwrite(out2 + "\\test%d.jpg" % d, i) d = d +
    1

# Displaying th features for i
in range(6):
    plt.subplot(2, 3, i + 1)
    plt.imshow(ids1[ i ])
    plt.xticks([ ])
plt.yticks([ ])
```

```
        plt.show()

    for i in range(6): plt.subplot(2, 3,
        i + 1) plt.imshow(ids2[ i ])
        plt.xticks([ ])
        plt.yticks([ ])

    plt.show()


# Main procedure while
True:
    x = input("Enter 0 to start extracting features from training images or 1 for testing the
image\n")
    if x == '0': trainproc()
        print("Features extracted!!!!!!")
        os.system("perform-training.py") break
    if x == '1': testproc()
        print("Features          extracted!!!!!!")
        os.system("perform-testing.py")
        break
    if x == 'exit':
        print("Exited!!!") break
    else:
        print("Enter correct key")
        continue
```

Training :

```
#For image processing
import cv2
# To performing path manipulations import
os
# Local Binary Pattern function
from skimage.feature import local_binary_pattern # For
plotting
import matplotlib.pyplot as plt # For
array manipulations import numpy
as np
# For saving histogram values
import joblib
# Utility Package import
cvutils
```

```
# Store the path of training images in train_images
train_images500 = []
for d in range(6): i =
    d+1
    ti = cvutils.imlist("train/500/id%i"%i)
    train_images500.append(ti)
n = len(train_images500[0])

train_images2000 = [] for
d in range(6):
    i = d+1
    ti = cvutils.imlist("train/2000/id%i"%i)
    train_images2000.append(ti)
n = len(train_images2000[0])

X_test500 = [] X_test2000 =
[]

# For each image in the training set calculate the LBP histogram # and
update X_test, X_name and y_test
for train_image in train_images500: #
    Read the image

    X_temp = []
    for i in range(n):
        im = cv2.imread(train_image[i])
        # Convert to grayscale as LBP works on grayscale image im_gray =
        cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)

        radius = 3
        # Number of points to be considered as neighbours
        no_points = 8 * radius
        # Uniform LBP is used
        lbp = local_binary_pattern(im_gray, no_points, radius, method='uniform') # Calculate
        the histogram
        n_bins = int(lbp.max() + 1)
        hist, _ = np.histogram(lbp, density=True, bins=n_bins, range=(0, n_bins)) X_temp.append(hist)
        # Append histogram to X_test
    X_test500.append(X_temp)
```

```
for train_image in train_images2000: #
    Read the image
    X_temp = []
    for i in range(n):
        im = cv2.imread(train_image[i])
        # Convert to grayscale as LBP works on grayscale image im_gray
        = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY) #im_gray =
        cv2.resize(im_gray, (100, 100),
interpolation=cv2.INTER_LINEAR) radius =
        3
        # Number of points to be considered as neighbours
        no_points = 8 * radius
        # Uniform LBP is used
        lbp = local_binary_pattern(im_gray, no_points, radius, method='uniform') # Calculate
        the histogram
        n_bins = int(lbp.max() + 1)
        hist, _ = np.histogram(lbp, density=True, bins=n_bins, range=(0, n_bins)) X_temp.append(hist)
        # Append histogram to X_test
    X_test2000.append(X_temp)
# Dump the data
joblib.dump((X_test500, X_test2000,n), "lbp.pkl", compress=3)

print("Images are been trained") os.system("preprocessing.py")
```

<u>Testing :</u>

```
# For image processing
import cv2
# To performing path manipulations import
os
# Local Binary Pattern function
from skimage.feature import local_binary_pattern # For
plotting
import matplotlib.pyplot as plt # For
array manipulations import numpy
as np
# For saving histogram values
import joblib
# Utility Package import
cvutils

# Displaying the fake result image def
fake_img():
    pth = "fake.jpg"
    img = cv2.imread(pth)
```

```
    cv2.imshow('FAKE!!!!', img)
    cv2.waitKey(0) cv2.destroyAllWindows()


# Displaying the genuine result image def
genuine_img():
    pth = "genuine.jpg" img =
    cv2.imread(pth)
    cv2.imshow('GENUINE!!!!', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()


# Load the List for storing the LBP Histograms, address of images and the corresponding label

X_test500, X_test2000, n = joblib.load("lbp.pkl")

# Store the path of testing images in test_images
test_images500 = cvutils.imlist("test/ids1") test_images2000 =
cvutils.imlist("test/ids2")

# Dict containing scores
results_all500 = {}
results_all2000 = {}

# total scores
tot500 = 0
tot2000 = 0

for i in range(6):
    # Read the image
    im = cv2.imread(test_images500[ i ], 0)

    radius = 3
    # Number of points to be considered as neighbourers
    no_points = 8 * radius
    # Uniform LBP is used
    lbp = local_binary_pattern(im, no_points, radius, method='uniform') # Calculate
    the histogram
    n_bins = int(lbp.max() + 1)
    hist, _ = np.histogram(lbp, density=True, bins=n_bins, range=(0, n_bins)) # Display
    the query image
    results = [ ]
    scores = 0
```

```
    # For each image in the training dataset
    # Calculate the chi-squared distance and the sort the values for index,
    x in enumerate(X_test500[ i ]):
        score = cv2.compareHist(np.array(x, dtype=np.float32), np.array(hist, dtype=np.float32),
cv2.HISTCMP_CHISQR)
        #        print(score)
        scores += score
    scores = scores / 3 results.append(round(scores,
    3)) results_all500[ "id%i" % i ] = results tot500
    += results[ 0 ]
#    print(results_all)

for i in range(6):
    # Read the image
    im = cv2.imread(test_images2000[ i ], 0)

    radius = 3
    # Number of points to be considered as neighbourers
    no_points = 8 * radius
    # Uniform LBP is used
    lbp = local_binary_pattern(im, no_points, radius, method='uniform') # Calculate
    the histogram
    n_bins = int(lbp.max() + 1)
    hist, _ = np.histogram(lbp, density=True, bins=n_bins, range=(0, n_bins)) # hist = x[:,
    1]/sum(x[:, 1])
    # Display the query image results
    = [ ]
    scores = 0
    # For each image in the training dataset
    # Calculate the chi-squared distance and the sort the values for index,
    x in enumerate(X_test2000[ i ]):
        score = cv2.compareHist(np.array(x, dtype=np.float32), np.array(hist, dtype=np.float32),
cv2.HISTCMP_CHISQR)
        #        print(score)
        scores += score
    scores = scores / 3 results.append(round(scores,
    3)) results_all2000[ "id%i" % i ] = results tot2000
    += results[ 0 ]
#    print(results_all)

buff1 = 0
buff2 = 0
```

```
while True:
    if results_all500[ 'id0' ][ 0 ] > 0.006: break
    if results_all500[ 'id1' ][ 0 ] > 0.02: break
    if results_all500[ 'id2' ][ 0 ] > 0.008: break
    if results_all500[ 'id3' ][ 0 ] > 0.06:

        break
    if results_all500[ 'id4' ][ 0 ] > 0.02: break
    if results_all500[ 'id5' ][ 0 ] > 0.07: break
    else:
        buff1 = 1
        break

while True:
    if results_all2000[ 'id0' ][ 0 ] > 0.006: break
    if results_all2000[ 'id1' ][ 0 ] > 0.02: break
    if results_all2000[ 'id2' ][ 0 ] > 0.008: break
    if results_all2000[ 'id3' ][ 0 ] > 0.06: break
    if results_all2000[ 'id4' ][ 0 ] > 0.02: break
    if results_all2000[ 'id5' ][ 0 ] > 0.07: break
    else:
        buff2 = 1
        break

if buff1 == 0 and buff2 == 0:
    fake_img()
    print("1")
elif buff1 and buff2:
    print("2")
    if tot2000 > tot500:
        print("500 CURRENCY NOTE")
    else:
        print("2000 CURRENCY NOTE")
    genuine_img()
elif buff1:
    print("3")
    print("500 CURRENCY NOTE")
    genuine_img() else:

    print("4")
    print("2000 CURRENCY NOTE")
          genuine_img() os.system("preprocessing.py"
```

CHAPTER 5

RESULTS AND DISCUSSION

Output:

The presented output outlines a systematic approach for detecting fake currency using machine learning algorithms. Beginning with the selection of an image , the process proceeds through image acquisition, resizing based on pixels , and subsequent steps include various processes listed below.

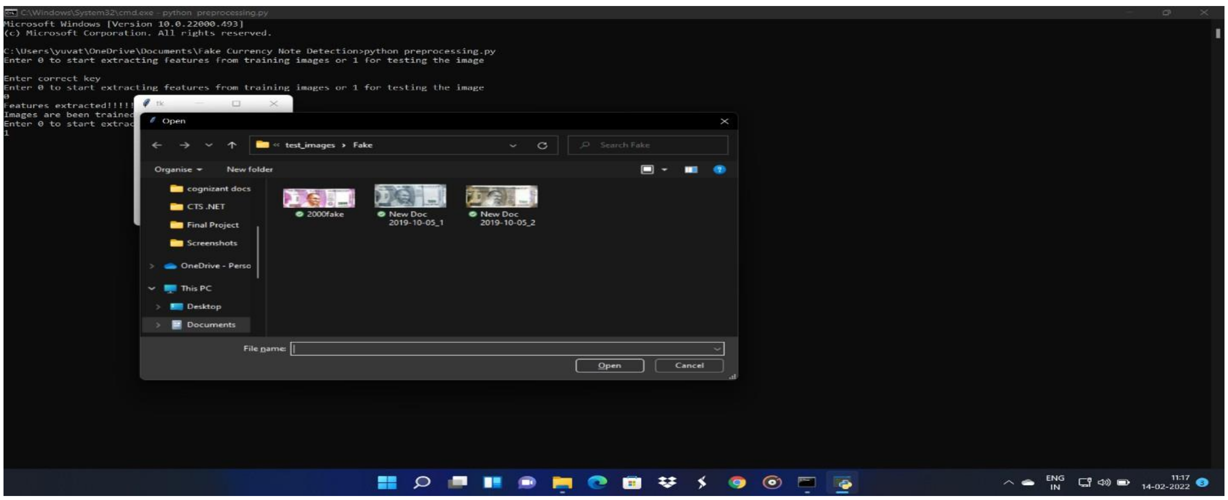

Figure 1 : Here we have to choose image

This figure represents the initial step where an image is chosen for further processing, serving as the input for the entire system.

Figure 2 : Image Acquisition – Here image will get resized based on pixels

Figure 2, the chosen image undergoes the image acquisition process, where it is resized based on pixels to ensure uniformity in subsequent analysis.

Figure 3 : Pre-Processing – Converts RGB to Gray color

The third figure, Pre-Processing, illustrates the conversion of the image from RGB to grayscale, simplifying subsequent processing steps.

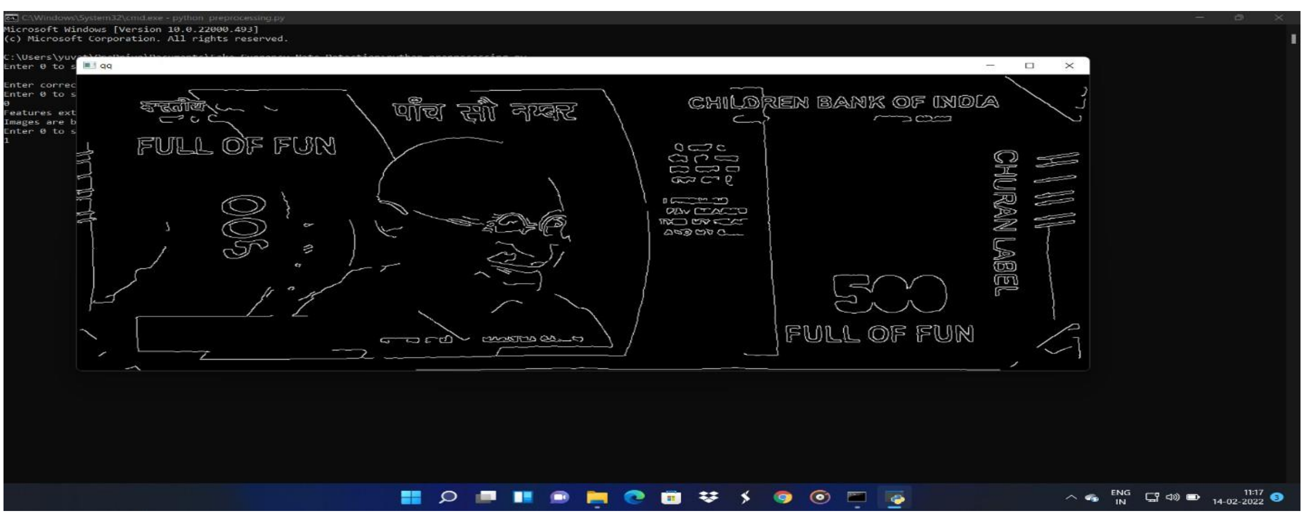

Figure 4 : Edge Detection – Detects Edges in Image

Displayed in Figure 4, Edge Detection is the process of identifying and highlighting edges within the image, crucial for subsequent feature extraction.

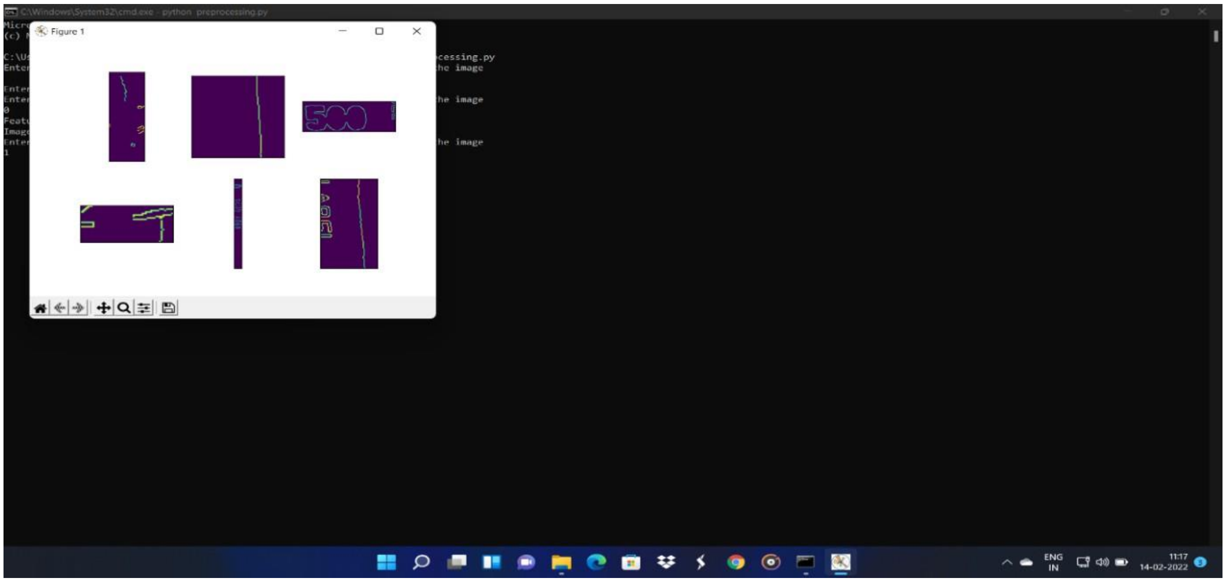

Figure 5 : Segmentation – Here feature extraction technique is Used

Figure 5 showcases the Segmentation step, employing feature extraction techniques to isolate and identify specific components within the image relevant to currency analysis.

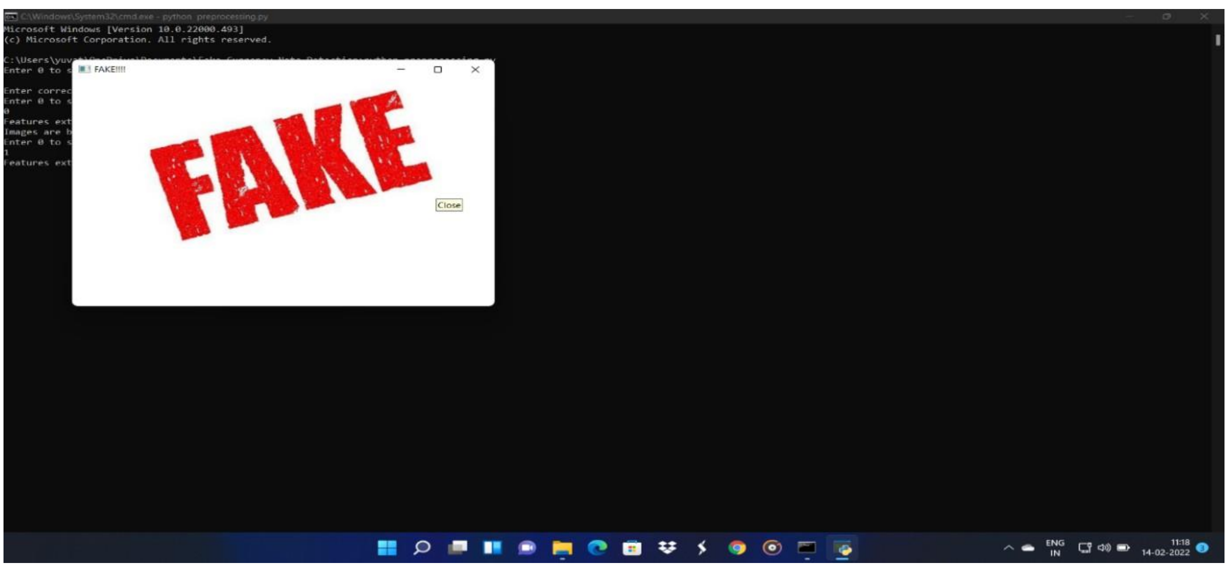

Figure 6 : Classification – Here its shows whether the currency is fake or not.

The final figure, Classification, visually indicates the system's output, determining and displaying whether the currency in the processed image is classified as fake or genuine based on the implemented machine learning classification algorithms.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

In this work, we have discussed that how our proposed system detects the fake bank currency using machine learning algorithms. The proposed system is also scalable for detecting the whether the currency is fake or not by image processing. The system is not having complex process to detect the whether the data contains fake bank currency like the existing system. Proposed system gives genuine and fast result than existing system. Here in this system we use cnn algorithm to detect whether currency is fake or no. Looking ahead, there are several avenues for further development and enhancement of the proposed system. Firstly, continuous updates to the machine learning algorithms, particularly CNN, can improve the system's accuracy and reliability over time. Additionally, incorporating more sophisticated image processing techniques and exploring advanced neural network architectures may contribute to even more robust detection capabilities.

REFERENCES

1. Vidhate, Y. Shah, R. Biyani, H. Keshri, R. Nikhare, Fake currency detection application. Int. Res. J. Eng. Technol. (IRJET) 08(05) (2021). e-ISSN: 2395-0056.

2. A.A. Mandankandy, K.E. Kannammal, Fake currency detection: a survey. Gedrag en Organisatie 33(4), 622–638 (2020).

3. A.A. Mandankandy, K.E. Kannammal, Fake currency detection: a survey. Gedrag en Organisatie 33(4), 622–638 (2020).

4. A.Singh, K. Bhoyar, A. Pandey, P. Mankani, A. Tekriwal, Detection of fake currency using image processing. Int. J. Eng. Res. Technol. (IJERT) 8(12) (2019).

5. G. Navya Krishna, G. Sai Pooja, B. Naga Sri Ram, V. Yamini Radha, P. Rajarajeswari, Recognition of fake currency note using convolutional neural networks. Int. J. Innov. Technol. Exploring Eng. 8(5), 58–63 (2019).

6. K.D. Sudha, P. Kilaru, M.S.R. Chetty, Currency note verification and denomination recognition on Indian currency system. Int. J. Recent Technol. Eng. 7(6S) (2019). ISSN: 2277-3878

7. M. Laavanya, V. Vijayaraghavan, Real time fake currency note detection using deep learning. Int. J. Eng. Adv. Technol. (IJEAT) 9(1S5) (2019). ISSN: 2249-8958.

8. T. Kumar, T. Subhash, D. Regan, Fake currency recognition system for Indian notes using image processing techniques (2019).

9. A.Upadhyaya, V. Shokeen, G. Srivastava, Analysis of counterfeit currency detection techniques for classification model (2018).

10. S. Gothe, K. Naik, V. Joshi, Fake currency detection using image processing and machine learning (2018).

11. S. Shaker, M.G. Alawan, Paper currency detection based image processing techniques: a review paper (2018).

12. V. Lalithendra Nadh, G. Syam Prasad, Support vector machine in the anticipation of currency markets. Int. J. Eng. Technol. (UAE) 7(2), 66–68 (2018).

13. T. Agasti, G. Burand, P. Wade, P. Chitra, Fake currency detection using image processing, ICSET-2017. IOP Conf. Ser. Mater. Sci. Eng. 263, 052047 (2017).