FAKE CURRENCY DETECTION

A Project report submitted in partial fulfillment of requirements for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND TECHNOLOGY

by

Moham Narendra (209X1A2864)

Sudamalla Sreenivasulu (209X1A2859)

Atla Hemanth Kumar (209X1A2828)

Under the esteemed guidance of

Smt. D. Sowmya
Assistant Professor
Department of C.S.E.

Department of Computer Science and Engineering

G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL

(Affiliated to JNTUA, ANANTAPURAMU)

2023 – 2024

Department of Computer Science and Engineering

G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL

(Affiliated to JNTUA, ANANTAPURAMU)

CERTIFICATE

This is to certify that the Project Work entitled 'Fake Currency Detection' **is a bonafide record of work carried out by**

Moham Narendra (209X1A2864)

Sudamalla Sreenivasulu (209X1A2859)

Atla Hemanth Kumar Reddy (209X1A2828)

Under my guidance and supervision in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & TECHNOLOGY

……..

Smt. D. Sowmya
Assistant Professor,
Department of C.S.E.,
G. Pulla Reddy Engineering College,
Kurnool.

……..

Dr. N. Kasiviswanath
Professor & Head of the Department,
Department of C.S.E.,
G. Pulla Reddy Engineering College,
Kurnool.

DECLARATION

we hereby declare that the project titled "**FAKE CURRENCY DETECTION**" is an authentic work carried out by us as the student of **G. PULLA REDDY ENGINEERING COLLEGE(Autonomous) Kurnool**, during **2023-24** and has not been submitted elsewhere for the award of any degree or diploma in part or in full to any institute.

MOHAM NARENDRA
(209X1A2864)

SUDAMALLA SREENIVASULU
(209X1A2859)

ATLA HEMANTH KUMAR REDDY
(209X1A2828)

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude to our project guide **Smt. D. Sowmya**, **Assistant Professor** of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for her immaculate guidance, constant encouragement and cooperation which have made possible to bring out this project work.

We are grateful to our project in charge **Dr. G. Rajeswarappa**, **Associate Professor** of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for helping us and giving us the required information needed for our project work.

We are thankful to our Head of the Department **Dr. N. Kasiviswanath**, for his whole hearted support and encouragement during the project sessions.

We are grateful to our respected Principal **Dr. B. Sreenivasa Reddy** for providing requisite facilities and helping us in providing such a good environment.

We wish to convey our acknowledgements to all the staff members of the Computer Science and Engineering Department for giving the required information needed for our project work.

Finally, we wish to thank all our friends and well-wishers who have helped us directly or indirectly during the course of this project work.

ABSTRACT

Counterfeiting of currency has become a real threat to the livelihood of people as well as the economy of our country. Though fake currency detectors are available, they are restricted to banks and corporate offices leaving common people and small businesses vulnerable. So, in this project, we will investigate the various security features of Indian currency and then, prepare a software-based system to detect and invalidate fake Indian currency by using advanced image processing and computer vision techniques. This currency authentication system is designed completely using Python language in Jupyter Notebook environment.

Keywords—Fake currency, counterfeit detection, image processing, feature extraction, Bruteforce matcher, ORB detector

CONTENTS

LIST OF FIGURES

INTRODUCTION

1. INTRODUCTION

1.1 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 proposed system is designed to validate Indian currency notes of denomination 500 and 2000 rupees. The system consists of three main algorithms and checks the authenticity of various features in a currency note. The first algorithm consists of several steps including image acquisition, pre-processing, greyscale conversion, feature extraction, image segmentation, comparisons of images and output, and uses advanced image processing methods such as ORB and SSIM. The second algorithm authenticates the bleed lines of the currency notes whereas the third algorithm authenticates the number panel of the currency notes. Finally, the processed output is displayed for each currency note

After demonetization 500 and 2000 are the high esteemed money notes existing till date so there is a most extreme likelihood that this note can be falsified so as to do this we are utilizing programming to recognize the phony notes utilizing picture handling system.

1.2 MOTIVATION

The main motivation behind development of this project was to make system for easy and quick detection of genuine and fake currency notes. This is MATLAB based system for automatic recognition of security features of currency. 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.

1.3 PROBLEMDEFINITION

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.

1.4 OBJECTIVE OF THE PROJECT

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 quicky and easily with low cost.

1.5 LIMITATIONS OF THE PROJECT

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 preprocessing method. In this there are basically 3 to 4 sub stages involved like preprocessing, grayscale conversion, edge detection and segmentation.

1.6 ORAGANIZATION OF THE PROJECT

The first chapter deals with the introduction, motivation for developing this project, objective of the project. The second chapter deals with the system specifications required for developing the project. it includes hardware and software specifications. the third chapter gives you the design and implementation which includes an introduction, source code, description of key parameters and functions, methods of implementation, testing, and validation. finally, the fourth chapter deals with the conclusion and future enhancements.

SYSTEM SPECIFICATIONS

2. SYSTEM SPECIFICATIONS

The software requirement specification is the starting point of the software developing activity. As the system grew complex it became evident that the goal of the entire system can't be easily comprehended. hence the need for the requirement phase arose. The software is initiated by the client's needs. The SRS is the means of translating the ideas of the minds of the clients into a formal document.

2.1 SOFTWARE SPECIFICATION

- Operating system : windows 10/ 64-bit/Version 1909/OS Build 18363.1316
- Programming language : Python
- Open Source IDE : Jupiter Note Book

2.2 HARDWARE SPECIFICATION

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in the case of operating systems.

Requirements:

- Processor : 1 gigahertz **(GHz)** or System on a Chip **(SoC)**
- Speed : 1.1GHZ
- RAM : 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit
- Hard Disk : 16 GB for 32-bit OS 32 GB for 64-bit OS

LITERATURE SURVEY

3. LITERATURE SURVEY

3.1 Introduction

3.1.1 VISVESVARAYA TECHNOLOGICAL UNIVERSITY

3.1.2 Title: Fake currency Detection using Basic Python Programming a

Literature Review

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. So automatic identification of currencies using image processing technique will be helpful to these peoples.it is also be useful at other work places. 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.

.

3.1.3 VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Title: Identification of fake notes and denomination recognition.

Literature Review

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.

3.2 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 genuine currency.

3.3 Disadvantages of Existing System

The existing systems for fake currency detection have certain drawbacks. These include limited detection capabilities against sophisticated counterfeit techniques, high initial and maintenance costs, the risk of false positives and negatives, complexity in operation and training, and potential challenges in adapting to evolving counterfeiting methods. Accessibility issues, speed of detection, integration challenges, privacy concerns, and regulatory barriers further contribute to the limitations of these systems. Continuous advancements in technology may address some of these issues over time, but a comprehensive approach involving various strategies is essential for effective counterfeit detection.

3.4 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 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.

3.5 Advantages of Proposed System

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.

DESIGN AND IMPLEMENTATION

4. DESIGN AND IMPLEMENTATION

4.1 Introduction

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. classifiers will be used to detect fake currency. The proposed system for fake currency detection will be simple, accurate and easy to use.

4.2 Methodology

Preparation of Dataset:

A. Data Collection
B. Image Acquisition
C. Pre-processing
D. Future Extraction
E. Algorithms
F. Detect Fake Currency

A. Data Collection:

The different categories of Indian currencies differ 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.

The first step is the preparation of a dataset containing images of different currency notes (both fake and real) and images of different features of each of the currency notes.

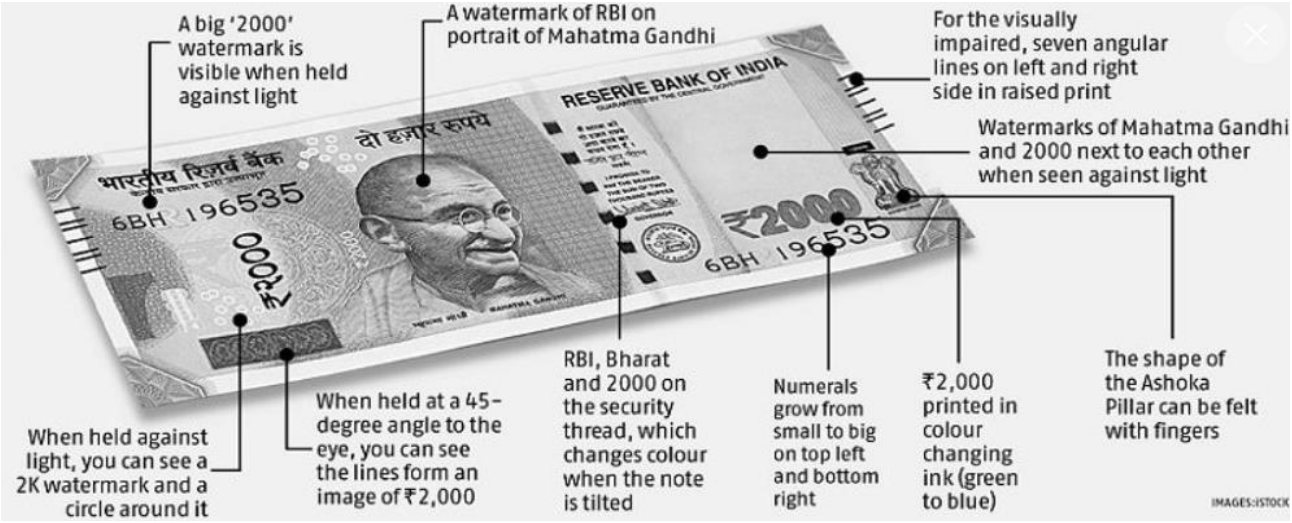

Fig-4.2.1-Data Set

•The dataset will contain the following repositories:

-Sub- dataset for Rs. 500 currency notes.

1) Images of real notes

2) Images of fake notes

3) Multiple images of each security feature.

Sub-dataset of Rs. 2000 currency notes (Similar structure)

•The various security features that we are considering are:

(For Rs. 500 currency notes- Total 10 features)

Rs. 500 in Devanagari and English script (2 features)

– Ashoka pillar Emblem (1 feature)

– RBI symbols in Hindi and English (2 features)

– 500 rupees written in Hindi (1 feature)

– RBI logo (1 feature)

– Bleed Lines on Left and right side (2 features)

– Number Panel (1 feature).

B. Image Acquisition:

Next, the image of the test-currency note is taken as input and fed it into the system. The image should be taken from a digital camera or preferably, using a scanner. The image should have a proper resolution, proper brightness and should not be hazy or unclear. Blurred images and images with less detail may adversely affect the performance of the system.

C. Pre-processing:

Next, the pre-processing of the input image is done. In this step, first the image is resized to a fixed size. A fixed size of image makes a lot of computations simpler. Next up, image smoothening is performed by using Gaussian Blurring method. Gaussian blurring removes a lot of noise present in the image and increases the efficiency of the system.

D. Future Extraction :

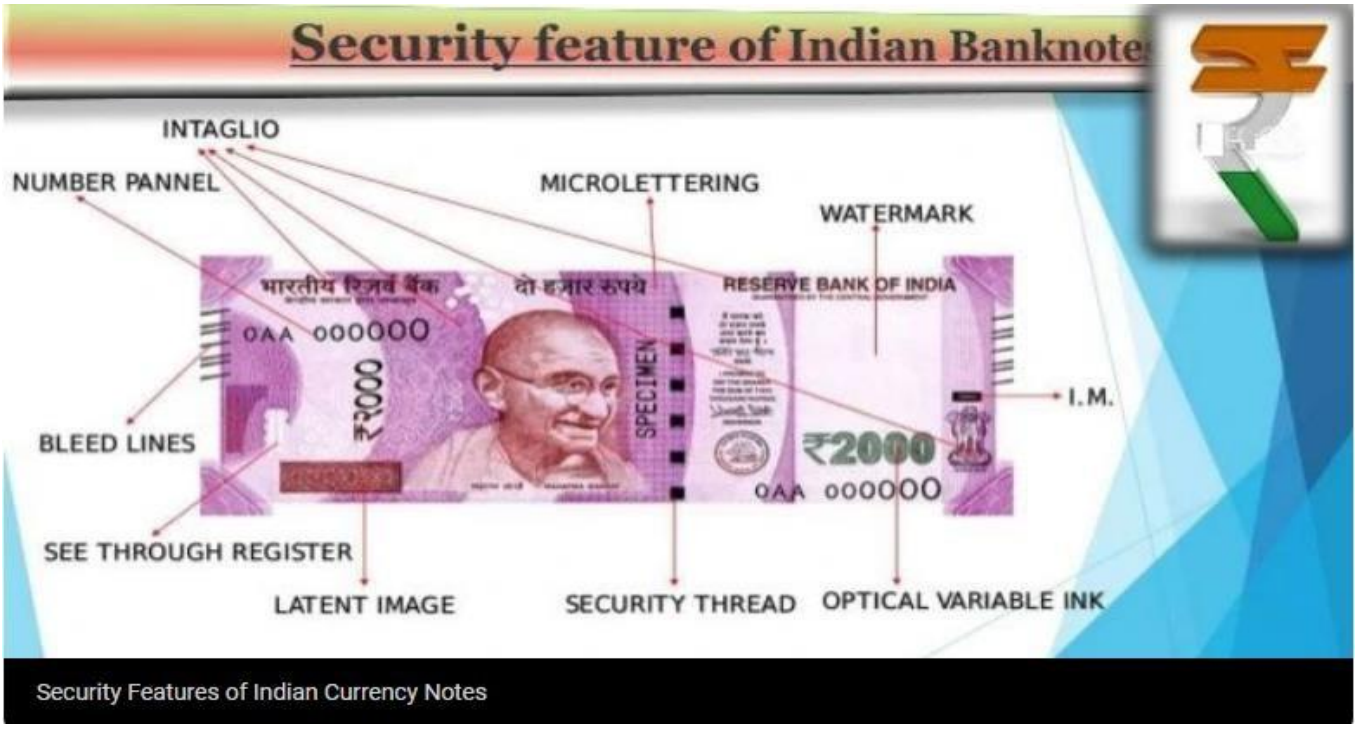

Fig-4.2.2-2000 Banknote

Security Thread:

The security thread appears to the left of the Mahatma's portrait. Security thread has a plain, non-readable fully embedded security thread. All the currency notes have security thread. When notes held against the light, the security thread on Rs.2000, Rs.500 and Rs.100 can be seen as one continuous line.

Fig-4.2.3-security thread

Watermark:

The Mahatma Gandhi Series of banknotes contain the Mahatma Gandhi watermark with a light and shade effect and multi-directional lines in the watermark window.

Micro-Lettering:

Fig-4.2.4-micro lettering

Latent Image:

On the obverse side of Rs.2000, Rs.500, Rs.100, Rs.50 and Rs.20 notes, a vertical band on the right side of the Mahatma Gandhi's portrait contains a latent image showing the respective denominational value in numeral. The latent image is visible only when the note is held horizontally at eye level.

Fig-4.2.5-latent image

Portrait of Mahatma Gandhi at the Center:

The intaglio printing of portrait of Mahatma Gandhi at the center of the currency

Fig-4.2.6-Mahatma Gandhi

See through Register:

The denomination numeral is displayed in the see-through register. Both sides of this register are printed. One side of the two sides is hollow, and the other side is filled with material. The micro lettering has been written horizontally along this register. The note has a latent image on the left side. Moreover, this register is shown above the latent image. When viewed in contrast to the light, this register appears as a single design.

Fig-4.2.7-See Through Register

Ashoka Pillar:
On the right side of the coin there is a picture of the Ashoka pillar.

Fig-4.2.8-Ashoka pillar

Identification Mark:
Just over the Ashoka's pillar symbol, there is an identification mark.

Fig-4.2.9-Identification mark

Bleed Lines:
The oblique lines that protrude from the sides of banknotes are known as bleed lines.

.

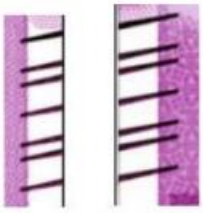

Fig-4.2.10-Bleed Lines

E. Algorithms:

Algorithm- 1: For feature 1- 7

1) Feature detection and matching using ORB:

After completing the necessary processing of the image, feature detection and matching is done using ORB. Our dataset already contains the images of different security features present in a currency note (total 10). Further, we have multiple images of varying brightness and resolutions corresponding to each security feature (6 templates for each feature). Using the ORB algorithm, each security feature is detected in the test image.

2) Feature Extraction:

Now, using ORB location of each template has been detected in the input image within the highlighted area. The highlighted area is then cropped by slicing the 3D pixel matrix of the image. Next, we apply Gray scaling and Gaussian blur to further smoothen the image and now our feature is ready to be compared with the corresponding feature in our trained model.

3) Feature comparison using SSIM:

From the previous step, the part of the test currency image which matches with each of the templates will be generated. In this method, the original template will be compared with the extracted feature and then a score will be given for the similarity between the two images using SSIM.

$SSIM(x, y) = (2\mu x\mu y + c1)(2\sigma xy + c2)/ (\mu 2\ x + \mu 2\ y + c1)(\sigma\ 2\ x + \sigma\ 2\ y + c2)$

Algorithm 2: For feature 8 and 9

]For feature 8 and 9 Every currency note contains bleed lines near the left and right edges. There are 5 lines in case of 500 currency note and 7 lines in case of Rs. 2000 currency near each of the two sides. This algorithm is being used to count and verify the number of bleed lines present in the left and right sides of a currency note. (feature 8 and 9)

Algorithm 3: For feature 10

Every currency note contains a number panel in the bottom right part where the serial number of the currency note is displayed. The number of characters present in the number panel should be equal to 9 (neglecting the space between the characters). This algorithm performs various operations and finally counts the number of characters present in the number panel.

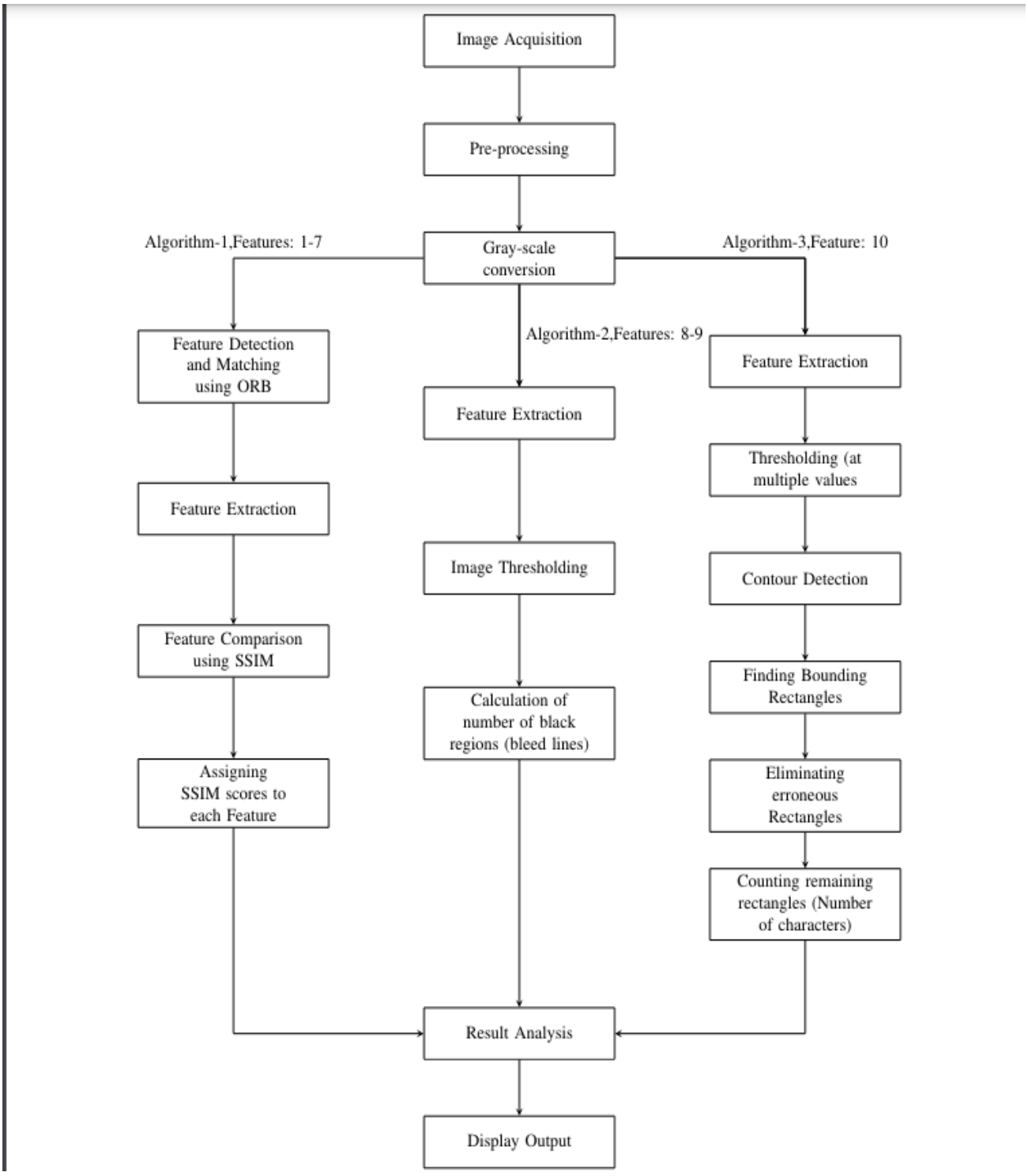

Fig-4.2.11-Flow Chart

4.3UML Diagrams

4.3.1Process flow diagram:

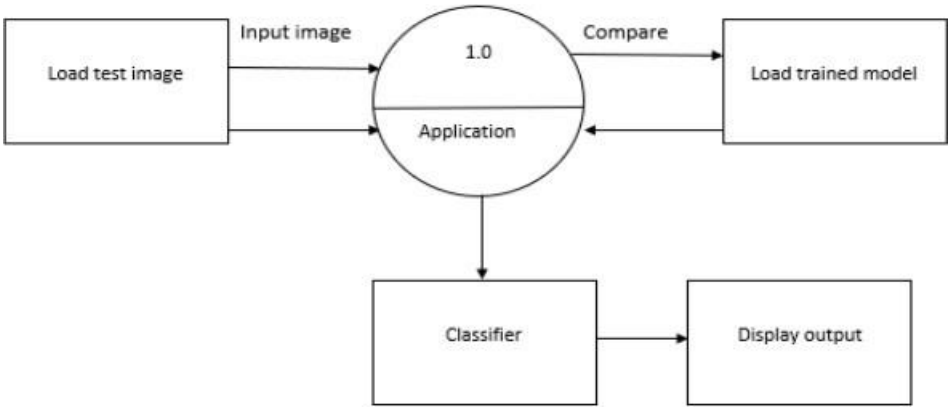

Fig - 4.3.1 – Process Flow Diagram

4.3.2Use Case Diagram

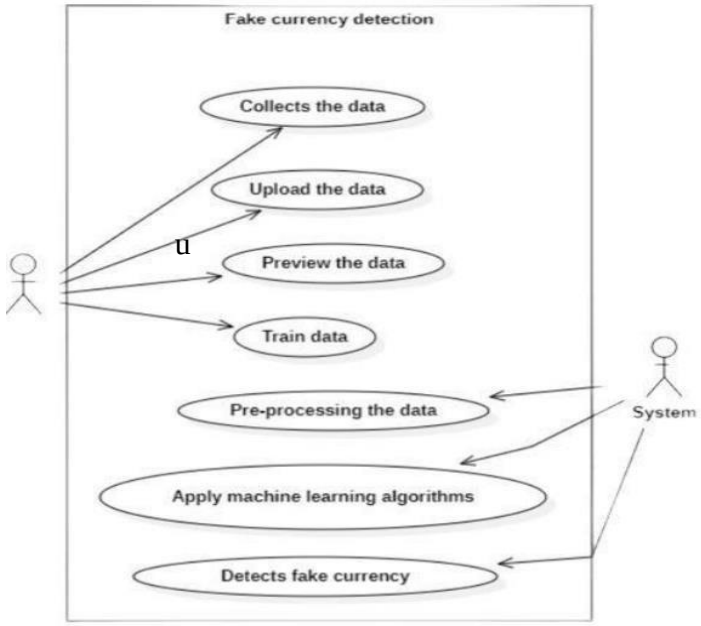

Fig - 4.3.2 - Use Case Diagram

4.3.3 Class Diagram

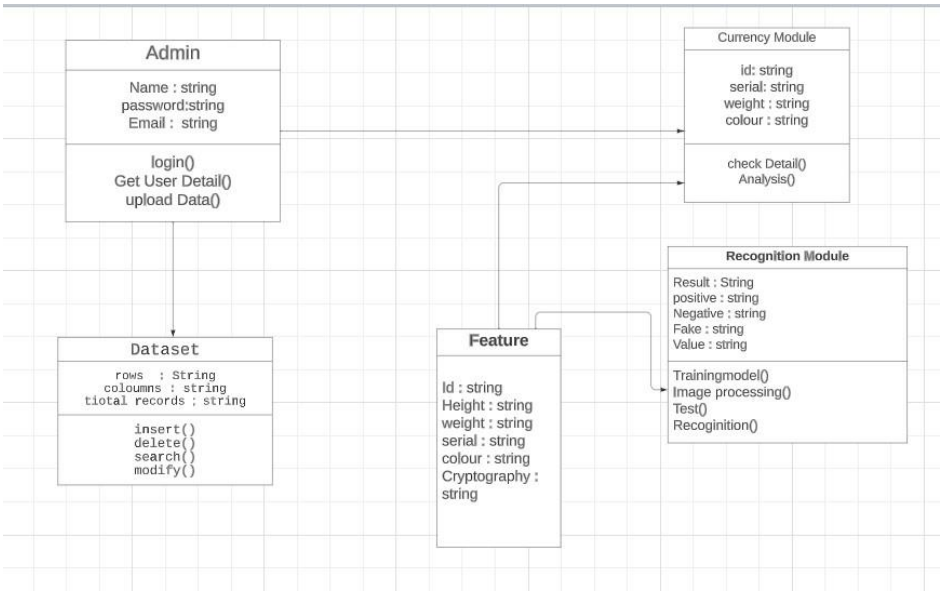

Fig - 4.3.3 - Class Diagram

4.3.4 Deployment Diagram

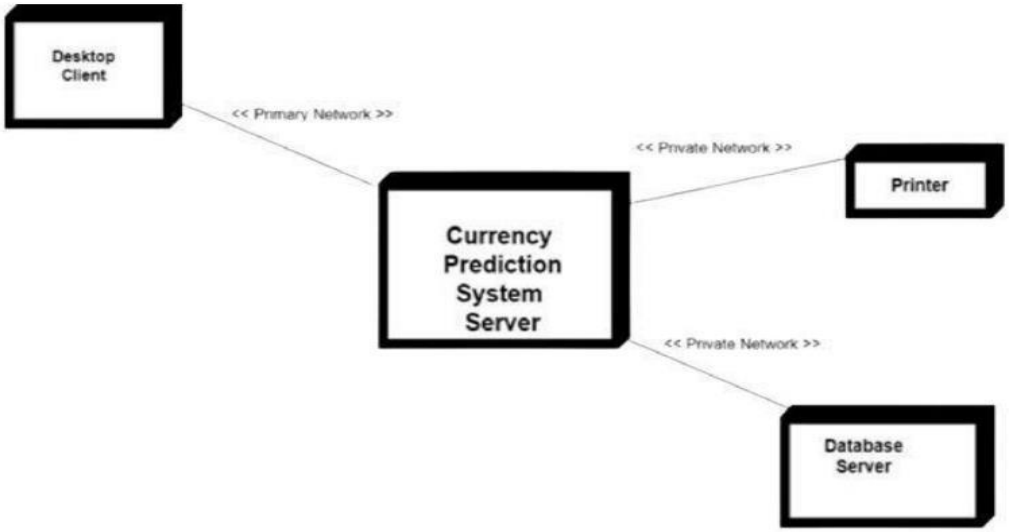

Fig - 4.3.4 - Deployment Diagram

.

4.4 Source Code

Fake Currency Detection

PreProcessing:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage.metrics import structural_similarity as ssim
plt.rcParams["figure.figsize"] = (12, 12)
%store -z
%run ./gui_1.ipynb
%store -r selectedImage
%store -r path
%store -r option
print('Image selected: ', selectedImage)
print('Path: ', path)
print('Currency type: ', option)
if selectedImage == True:
    if option == 1:
        %run ./500_Testing.ipynb
    elif option == 2:
        %run ./2000_Testing.ipynb
if selectedImage == True:
    %store -r result_list
    for x in result_list:
        if x[0] is not None:
            plt.imshow(x[0])
            plt.show()
if selectedImage == True:
  %run ./gui_2.ipynb
%store -z
```

Training:

GUI- Starting:

```
 from tkinter import *
 from PIL import Image
from PIL import ImageTk
import tkinter.filedialog as tkFileDialog
import cv2
from tkinter import messagebox
def currency_type():
    global option
    option = int(var.get())
def select_image():
    global canvas
    global path
    canvas.delete("all")
    path = tkFileDialog.askopenfilename()
        image = cv2.imread(path)
        original_image = image.copy()
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        image = cv2.resize(image, (675, 300))
        image = Image.fromarray(image)
        image = ImageTk.PhotoImage(image)
        canvas.image = image
        canvas.create_image(0, 0, anchor=NW, image=image)
    else:
        messagebox.showinfo("Warning","Please choose an image! (It should be a .jpg file)")
def submit():
    global option
    global path
    global selectedImage
    if len(path) <= 0:
        messagebox.showinfo("Warning","Please choose an image!")
    elif option == -1
```

```
messagebox.showinfo("Warning","Please choose the currency type!")
    else:
        print(path)
        print(option)
        selectedImage = True
        messagebox.showinfo("Info","Image sent for processing!")
        root.destroy()
def exit_window():
    root.destroy()
root = Tk()
root.title("Fake Currency Detection System")
root.resizable(False, False)
window_height = 600
window_width = 1100
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
 x_cordinate = int((screen_width/2) - (window_width/2))
y_cordinate = int((screen_height/2) - (window_height/2))
root.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))
path = ""
option = -1
selectedImage = False
top_frame = Frame(root, bg='black', width=1090, height=50, pady=3)
frame1 = Frame(root, bg='cyan', width=1090, height=80, padx=3, pady=3)
frame2 = Frame(root, bg='brown', width=1090, height=400, pady=5, padx = 5)
frame3 = Frame(root, width=1090, height=50, pady=3)
frame4 = Frame(root, width=1090, height=50, pady=3)
frame5 = Frame(root, bg='white', width=1090, height=20, pady=3)
top_frame.grid(row = 1, column = 1, padx = 5, pady = 5)
frame1.grid(row = 2, column = 1, padx = 5, pady = 5)
frame2.grid(row = 3, column = 1, padx = 5, pady = 5)
frame3.grid(row = 4, column = 1, padx = 5, pady = 5)
frame4.grid(row = 5, column = 1, padx = 5, pady = 5)
```

```
frame5.grid(row = 6, column = 1, padx = 5, pady = 5)

title = Label(master=top_frame, text="FAKE CURRENCY DETECTION SYSTEM", fg = 'dark blue', font = "Verdana 28 bold")

title.pack()

text1 = Label(master=frame1, text="This is a fake currency detection sytem. Select the currency type, browse your image file and get started!", fg = 'blue', font = "Verdana 10")

text1.pack()

canvas = Canvas(master=frame2, width = 675, height = 300)

canvas.pack()

text1 = Label(master=frame3, text="Select currency type: ", fg = 'black', font = "Verdana 12")

text1.pack(side = 'left')

var = IntVar()

R1 = Radiobutton(master=frame3, text="500", variable=var, value=1, command=currency_type, font = "Verdana 15")

R1.pack(anchor =  W)

R2 = Radiobutton(master=frame3, text="2000", variable=var, value=2,command=currency_type, font = "Verdana 15")

R2.pack(anchor =  W)

btn = Button(master = frame4, text="Select an image", command=select_image, font = "Verdana 15 bold", fg='blue')

btn.pack(side = 'left', padx=10, pady=10)

btn = Button(master = frame4, text="Submit", command=submit, font = "Verdana 15 bold", fg='green')

btn.pack(side = 'left', padx=10, pady=10)

btn = Button(master = frame4, text="Exit", command=exit_window, font = "Verdana 15 bold", fg='red')

btn.pack(side = 'left', padx=10, pady=10)

root.mainloop()

%store selectedImage

if selectedImage == True:

    %store path

    %store option
```

GUI- Ending:

```
from tkinter import *
from PIL import Image as PIL_Image
from PIL import ImageTk
import tkinter.filedialog as tkFileDialog
```

```
import cv2
from tkinter import messagebox
import matplotlib.pyplot as plt
%store -r result_list
%store -r path
def display_output():
    # Creating 4 sub frames inside the master_frame
    sub_frame1 = Frame(master_frame, bg='black', pady=5)
    sub_frame2 = Frame(master_frame, bg='brown', pady=5, padx = 5)
    sub_frame3 = Frame(master_frame, pady=5, padx = 5)
    sub_frame4 = Frame(master_frame, pady=5, padx = 5)
    sub_frame1.grid(row = 1, column = 1, padx = 5, pady = 5)
    sub_frame2.grid(row = 2, column = 1, padx = 5, pady = 5)
    sub_frame3.grid(row = 3, column = 1, padx = 5, pady = 5)
    sub_frame4.grid(row = 4, column = 1, padx = 5, pady = 5)
    title = Label(master=sub_frame1, text="FAKE CURRENCY DETECTION SYSTEM", fg = 'dark
blue', font = "Verdana 28 bold")
    title.pack() # Put the label into the window
    canvas_input = Canvas(master=sub_frame2, width = 675, height = 300)
    canvas_input.pack()
    if len(path) > 0 and path[-4:] == '.jpg':
        image = cv2.imread(path)
        original_image = image.copy()
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        image = cv2.resize(image, (675, 300))
        image = PIL_Image.fromarray(image)
        image = ImageTk.PhotoImage(image)
        canvas_input.image = image
        canvas_input.create_image(0, 0, anchor=NW, image=image)
    pass_count = 0
    for i in range(4):
        for j in range(3):
            feature_num = 3*i+j
            if feature_num < 10:
                sub_frame4.grid_rowconfigure(i, weight=1)
                sub_frame4.grid_columnconfigure(j, weight=1)
                feature_frame = Frame(master = sub_frame4, relief = RAISED, borderwidth = 1, bg='light
blue')
                feature_frame.grid(row = i, column = j, padx = 20, pady = 20, sticky="nsew")
                frame1 = Frame(feature_frame, padx=3, pady=3)
                frame2 = Frame(feature_frame, bg='brown', pady=5, padx = 5)
                frame3 = Frame(feature_frame)
                frame4 = Frame(feature_frame)
                frame5 = Frame(feature_frame)
                frame1.grid(row = 1, column = 1, padx = 5, pady = 5, ipadx = 100)
                frame2.grid(row = 2, column = 1, padx = 5, pady = 5)
                frame3.grid(row = 3, column = 1, padx = 5, pady = 5)
                frame4.grid(row = 4, column = 1, padx = 5, pady = 5)
                frame5.grid(row = 5, column = 1, padx = 5, pady = 5)
```

```
                label1 = Label(master = frame1, text = f"Feature {feature_num +1}", fg =
'black', font = "Verdana 12 bold")
        label1.pack()
        canvas = Canvas(master=frame2, width = 200, height = 200)
        canvas.pack()
        image = result_list[feature_num][0].copy()
        h, w = image.shape[:2]
        aspect_ratio = w/h
        resize_height = 0
        resize_width = 0
        img_x = 0
        img_y = 0
        if h > w:
            resize_height = 200
            resize_width = aspect_ratio * resize_height
            img_x = (200 - resize_width)/2
        elif h < w:
            resize_width = 200
            resize_height = resize_width / aspect_ratio
            img_y = (200 - resize_height)/2
        else:
            resize_height = 200
            resize_width = 200
        resize_height = int(resize_height)
        resize_width = int(resize_width)
        img_x = int(img_x)
        img_y = int(img_y)
        image = cv2.resize(image, (resize_width, resize_height))
        image = PIL_Image.fromarray(image)
        image = ImageTk.PhotoImage(image)
        canvas.image = image
        canvas.create_image(img_x, img_y, anchor=NW, image=image)
        if feature_num < 7:
            avg_score = result_list[feature_num][1]
            avg_score = "{:.3f}".format(avg_score)
            text2 = "Avg. SSIM Score: " + avg_score
        elif feature_num < 9:
            line_count = result_list[feature_num][1]
            line_count = "{:.3f}".format(line_count)
            text2 = "Avg. Number of lines: " + line_count
        else:
            status = result_list[feature_num][1]
            if status == True:
                text2 = "9 characters detected!"
            else:
                text2 = "Less than 9 characters detected!"
        label2 = Label(master = frame3, text = text2, fg = 'dark blue', font = "Verdana 11",
bg='light blue')
        label2.pack()
```

```
            if feature_num < 7:
                max_score = result_list[feature_num][2]
                max_score = "{:.3f}".format(max_score)
                text3 = "Max. SSIM Score: " + max_score
            elif feature_num < 9:
                text3 = ""
            else:
                text3 = ""
            label3 = Label(master = frame4, text = text3, fg = 'dark blue', font = "Verdana 11",
bg='light blue')
            label3.pack()
            if feature_num < 7:
                status = result_list[feature_num][3]
            elif feature_num < 9:
                status = result_list[feature_num][2]
            else:
                status = result_list[feature_num][1]
            if status == True:
                pass_count += 1
                label4 = Label(master = frame5, text = "Status: PASS!", fg = 'green', font = "Verdana
11 bold", bg='light blue')
                label4.pack()
            else:
                label4 = Label(master = frame5, text = "Status: FAIL!", fg = 'red', font = "Verdana 11
bold", bg='light blue')
                label4.pack()
        result = Label(master=sub_frame3, text= f"RESULT: {pass_count} / 10 features PASSED!", fg
= 'green', font = "Verdana 24 bold")
    result.pack()
def scrollbar_function(event):
    canvas.configure(scrollregion=canvas.bbox("all"),width=1050,height=550)
root=Tk()
root.title('Fake Currency Detection - Result Analysis')
root.resizable(False, False)  # This code helps to disable windows from resizing
window_height = 600
window_width = 1100
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
x_cordinate = int((screen_width/2) - (window_width/2))
y_cordinate = int((screen_height/2) - (window_height/2))
root.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))
main_frame=Frame(root,relief=GROOVE, bd=1)
main_frame.place(x=10,y=10) # Placing the frame at (10, 10)
canvas=Canvas(main_frame)
master_frame=Frame(canvas)  # Creating master_frame inside the canvas
myscrollbar=Scrollbar(main_frame,orient="vertical",command=canvas.yview)
canvas.configure(yscrollcommand=myscrollbar.set)
myscrollbar.pack(side="right",fill="y")
canvas.pack(side="left")
```

```
canvas.create_window((0,0),window=master_frame,anchor='nw')
master_frame.bind("<Configure>",scrollbar_function)
display_output()
root.mainloop()
```

Testing:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage.metrics import structural_similarity as ssim
from skimage library
from tkinter import *
from tkinter.ttk import Progressbar
import time
plt.rcParams["figure.figsize"] = (12, 12)
%store -r path
print('Path of input image: ', path)
test_img = cv2.imread(path)
test_img = cv2.resize(test_img, (1167, 519))
blur_test_img = cv2.GaussianBlur(test_img, (5,5), 0)
gray_test_image = cv2.cvtColor(blur_test_img, cv2.COLOR_BGR2GRAY)
def preprocessing():
    plt.imshow(gray_test_image, 'gray')
    plt.title('Input image after pre- processing')
    plt.show()
    progress['value']=5
    ProgressWin.update_idletasks()
    progress['value']=5
    ProgressWin.update_idletasks()
def calculateSSIM(template_img, query_img):
    min_w = min(template_img.shape[1], query_img.shape[1])
    min_h = min(template_img.shape[0], query_img.shape[0])
    img1 = cv2.resize(template_img, (min_w, min_h))
    img2 = cv2.resize(query_img, (min_w, min_h))
    img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
    plt.subplot(1, 2, 1)
    plt.imshow(img1, 'gray')
    plt.subplot(1, 2, 2)
    plt.imshow(img2, 'gray')
    plt.show()
    score = ssim(img1, img2)
    return score
def computeORB(template_img, query_img)
    nfeatures=700;
    scaleFactor=1.2;
    nlevels=8;
    edgeThreshold=15; # Changed default (31);
    orb = cv2.ORB_create(
```

```
        nfeatures,
        scaleFactor,
        nlevels,
        edgeThreshold)
    kpts1, descs1 = orb.detectAndCompute(template_img,None)
    kpts2, descs2 = orb.detectAndCompute(query_img,None)
    bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
    matches = bf.match(descs1, descs2)
    dmatches = sorted(matches, key = lambda x:x.distance)
    src_pts  = np.float32([kpts1[m.queryIdx].pt for m in dmatches]).reshape(-1,1,2)
    dst_pts  = np.float32([kpts2[m.trainIdx].pt for m in dmatches]).reshape(-1,1,2)
    M, mask = cv2.findHomography(src_pts, dst_pts, cv2.RANSAC,5.0)
    h,w = template_img.shape[:2]
    pts = np.float32([ [0,0],[0,h-1],[w-1,h-1],[w-1,0] ]).reshape(-1,1,2)
    if M is not None:
        dst = cv2.perspectiveTransform(pts,M)
    else:
        dst = None
    return dst, dst_pts, kpts1, kpts2, dmatches
def testResult():
    plt.rcParams["figure.figsize"] = (3, 3)
    print('\n\nRESULT ANALYSIS\n')
    min_ssim_score_list = [0.4, 0.4, 0.5, 0.4, 0.5, 0.45, 0.5]
    global result_list
    result_list = []              # To store details of each feature
    successful_features_count = 0  # To store number of features which passed the test
    for i in range(NUM_OF_FEATURES):
        avg_score = avg_ssim_list[i]
        img, max_score = best_extracted_img_list[i]
        status = False
        min_allowed_score = min_ssim_score_list[i]
        if avg_score >= min_allowed_score or max_score >= 0.79:
            status = True
            successful_features_count += 1
            print('Feature ' + str(i+1) + ': Successful')
        else:
            status = False
            print('Feature ' + str(i+1) + ': Unsuccessful')
        if img is None:
            img = cv2.imread('Image_not_found.jpg')
        result_list.append([img, avg_score, max_score, status])
    img, line_count = left_BL_result[:]
    if line_count >= 4.7 and line_count <= 5.6:
        status = True
        successful_features_count += 1
        print('Feature 8: Successful- 5 bleed lines found in left part of currency note')
    else:
        status = False
        print('Feature 8: Unsuccessful!')
```

```
    if img is None:
        img = cv2.imread('Image_not_found.jpg')
    result_list.append([img, line_count, status])
    img, line_count = right_BL_result[:]
    if line_count >= 4.7 and line_count <= 5.6:
        status = True
        successful_features_count += 1
        print('Feature 9: Successful- 5 bleed lines found in right part of currency note')
    else:
        status = False
        print('Feature 9: Unsuccessful!')
    if img is None:
        img = cv2.imread('Image_not_found.jpg')
    result_list.append([img, line_count, status])
    img, status = number_panel_result[:]
    if status:
        successful_features_count += 1
        print('Feature 10: Successful- 9 characters found in number panel of currency note')
    else:
        print('Feature 10: Unsuccessful!')
    if img is None:
        img = cv2.imread('Image_not_found.jpg')
    result_list.append([img, status])
    print('\nResult Summary:')
    print(str(successful_features_count) + ' out of 10 features are VERIFIED!')
    global myProgress
    progress['value']=97
    ProgressWin.update_idletasks()
%store result_list
```

4.5Outputs:

Fig.4.5.4- Intially no image is dislayed and user is asked to insert image

Fig.4.5.2-Browsing image

Fig.4.5.3-Input image of currency n

Fig.4.5.4-Image sent for processing...

Fig.4.5.5-GUI showing final result(Real note)

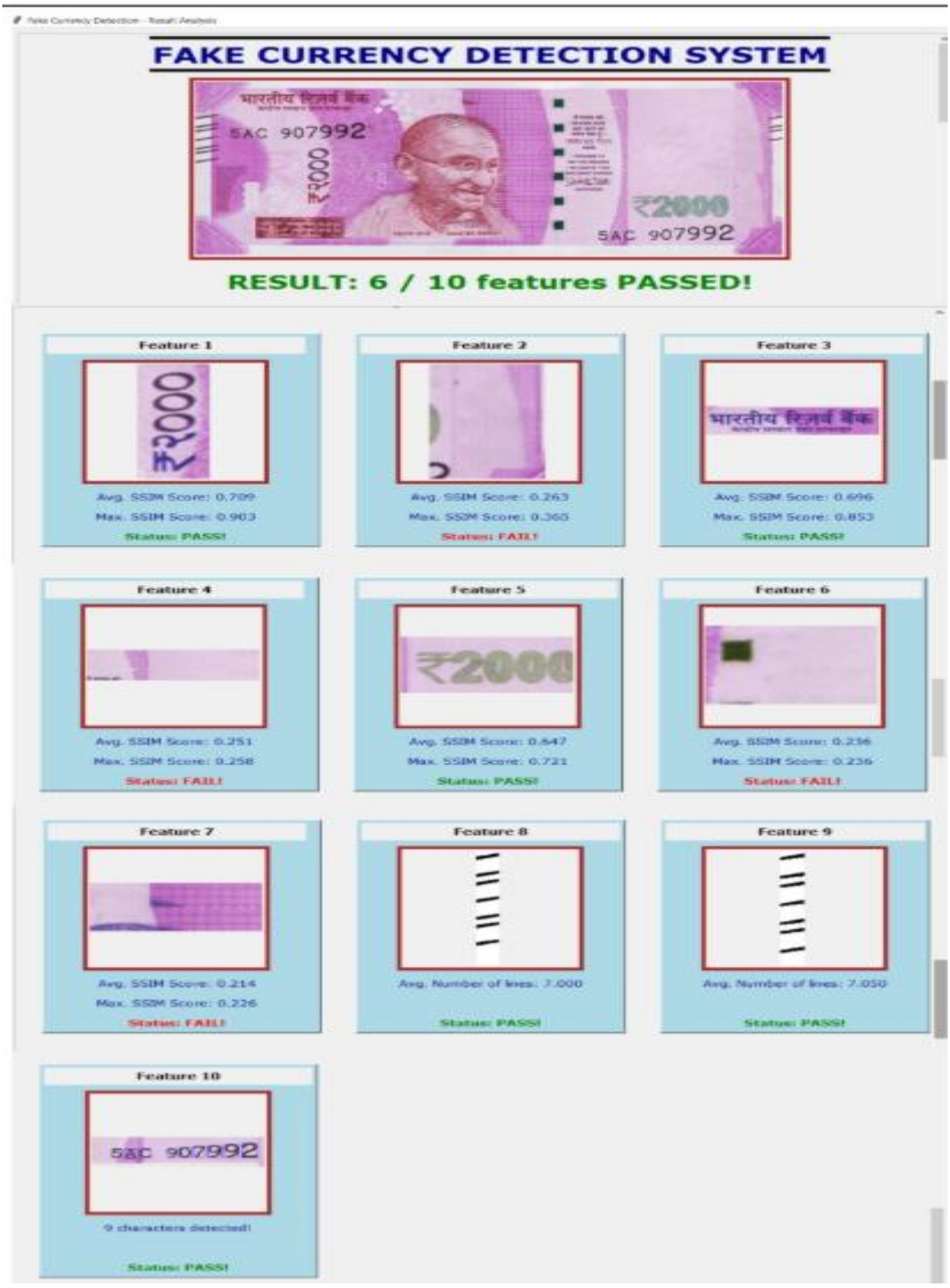

Fig.4.5.6-GUI showing final result(Fake note)

CONCLUSION

1. CONCLUSION

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 not.

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.

REFERENCES

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. 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.

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

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