FAKE CURRENCY DETECTOR

***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

I. Introduction

Currency duplication or production of counterfeit currency notes illegally by imitating the actual manufacturing process is a huge problem that every country is facing. Fake currency can reduce the value of real money and cause inflation due to an unauthorized and unnatural increase in the money supply. Manual authentication of currency notes is a solution but it is a very time-consuming, inaccurate, and difficult process. Automatic testing of currency notes is, therefore, necessary for handling large volumes of currency notes and then, getting accurate results in a very short time span. In this project, we propose a fake currency note detection system using various image processing techniques and algorithms.

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. This system provides a hassle-free way to authenticate currency notes quickly and accurately. This automated system can replace the existing manual methods and can be used by anyone easily to detect fake currency.

A. Commonly Used Security Features to Detect Fake Notes

1) Bleed lines: There are angular bleed lines on 500 and 2000 note on left and right corner of note in raised print.In 500Rs. note there are 5 bleed lines and In 2000Rs. note, 7 bleed lines.

2) Security Thread: A colour-shifting security thread with the inscription Bharat (in Hindi).RBI and 2000 (500 for 500 note). The colour changes from green to blue when it is tilted.

3) Latent Image : Latent images of number 2000 / 500 can be seen when note is held at 45 degrees angle.

4) Water mark: Watermark of Mahatma Gandhi and electrotype of numeral 2000/500.

5) Denominational Numeral: See-through register, with denominational numeral 2000, can be seen when you hold the note against the light.

6) Portrait of Mahatma Gandhi: Portrait of Mahatma Gandhi wiht RBI written on his spectacle which can be read using a magnifying glass

7) Number panel: Numerals growing from small to big size, is printed on the top left side and bottom right side.

8) Denominational numeral: On left side of Mahatma Gandhi there is a 500/2000 in Devnagari script.

9) Ashoka Pillar: There is Ashoka Pillar on right bottom side.

10) Guarantee and promise clause: The guarantee and promise clause of RBI is present in devnagri as well as in english in the top left and top right corner of the currency notes respectively.

11) RBI seal: The seal of RBI is present just below the Governor’s signature. This seal as well as the guarantee clause, etc are in intaglio printing.

12) Denominational value in words: The denominational value of the currency note is written in devnagiri script in the top central region of the note.

II. Literature Survey

1. In this paper an automatic system is designed for identification of Indian currency note is fake or original. The automatic system is very useful in the banking system and other fields also. In India increase in the fake currency notes of 100, 500, and 1000 rupees. As there are increases in technology, like scanning, colour printing, and duplicating, there is an increase in the counterfeit problem. In this model first the image acquisition is done and applies to pre-process to the image. In pre-processing crop, smooth and adjust then convert the image into grey colour after conversion applies the image segmentation then extract features and reduce, finally comparing image.

2. In this paper, An automatic recognition of fake Indian paper currency note using MATLAB is implemented and feature extraction with HSV color space and other applications of image processing. The proposed architecture is as follows Image Acquisition, Gray-Scale Conversion, Edge Detection, Image Segmentation, Characteristic Extraction, Comparison, Output. In the project setup, a note is placed in front of the camera to check whether it is fake or genuine. The clicked pictures of notes are analyzed by MATLAB program installed on the computer. This algorithm works for Indian denomination 100, 500, 1000. If the note is genuine, the respective message is appeared on the screen and vice-versa.

3. In this paper, a hybrid fake currency detection model was proposed and implemented on MATLAB. The model was designed to detect fake Bangladeshi notes. In the proposed model, three image processing algorithms were chosen namely Optical Character Recognition(OCR), Hough Transformation and Face recognition(MSD) to get enhanced results. Then, the results of the proposed model were compared with each model used. The algorithm followed six basic steps: data collection, pre-processing of the collected data, edge detection, feature extraction, identification and output results. In addition, although the proposed model took slightly longer to process, it displayed 93.33% accuracy which was significantly higher compared to the individual algorithms applied.

4. In this paper two ways are used: one by using analysis through hyper spectral imaging and the second one is the Extraction of different features in fake and real currency notes and through comparing with each other we can able to differentiate the fake note from the real note. The different colour lights used for hyper spectral imaging are Ultraviolet (UV) light, Normal LED Bulb, Red LED light, Green LED light and Blue LED light with different wavelengths ranging from 360 nm to 800nm respectively. This all modules are implemented in MATLAB. They have implemented a fake note detection unit with image processing algorithms. The experimental results indicate that the results achieved are nearly accurate.

5. In this paper the recognition and verification of the paper currency is explained with the use of image processing techniques. The proposed approach consists of multiple element transactions like Image Acquisition, Feature extraction and comparison, Texture features, and Voice output.This system is divided into two parts. The first part is to identify the currency denomination through image processing. The second part is the oral output to notify the visually impaired person about the denomination of the note that he/she is currently having. The desired results will be in text and voice output of the currency recognized and verified.

III. Problem Statement

To test the authenticity of Indian currency notes by preparing a system which takes the image of currency bill as input and gives the final result by applying various image processing and computer vision techniques and algorithms.

A. Objectives

- The main objective of the project is to identify the fake Indian currency notes through an automated system by using Image processing and computer vision techniques.
- The system should have high accuracy.
- The system should be able to give the final results in a short time.
- The system should have an User-friendly interface, to make it convenient to use and understand.

IV. Methodology

*A. **Preparation of Dataset***

- 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
- 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 (template)
  - 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)

TABLE I: Summary of Literature Survey

Authors	Methodology	Merits	Limitations
Sonali R. Darade [1]	Feature extraction and im- age processing	Detection of note is good Cost is low	External camera is used
Binod Prasad Yadav, P.H Patil [2]	Feature extraction with HSV image processing	effective and efficient im- age processing	Whole setup required
Adiba Zarin ,Jia Uddin [3]	Optical Character recogni- tion (OCR)	93.33 % accuracy	Hard method
Shripad veling [4]	Hyperspectral Imaging	Two ways to get result	Cost is high ,complicated
Dr. P. Mangayarkarasi, Akhilendu, Anakha A S [5], Meghashree K, Faris A B	Image processing, Image Acquisition, Feature ex- traction	cost and time efficient	If note is dirty and torn than it will give wrong an- swer

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. *Gray- scale conversion*

Gray scale conversion is mainly used because an RGB image has 3 channels whereas a gray image has only one channel. This makes the computation and processing on images much more easier in the case of gray scaled images.

E. *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. To make the searching of the security feature (template image) easier and more accurate, a search area will be defined on the test currency image where that template is most likely to be present. Then, ORB will be used to detect the template in the test image and the result will be highlighted properly with a marker. This process will be applied for every image of each security feature present in the data-set and every time the detected part of the test image will be highlighted properly using proper markers.

Fig. 1: ORB Feature detection and Matching

2) Feature Extraction: Now, using ORB location of each template has been detected in the input image within the high- lighted 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.

Fig. 2: Features in 500 | currency bill

Fig. 3: Features in 2000 | currency bill

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

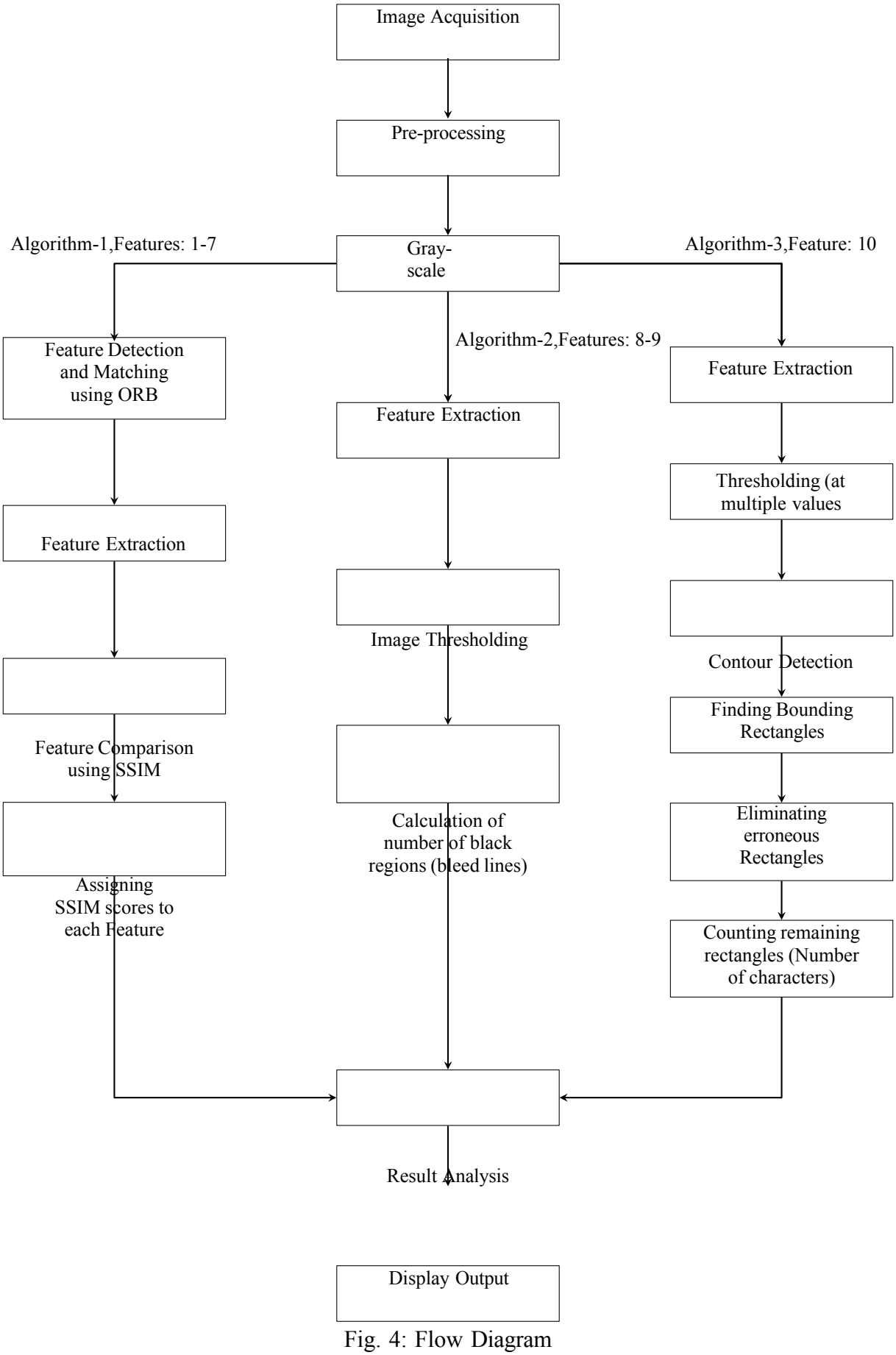


Fig. 4: Flow Diagram

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.

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The Structural Similarity Index (SSIM) is a scoring system that quantifies the image quality degradation that is caused by processing such as data compression or by losses in data transmission. Basically it looks for similarity between two images. It is a part of skimage library and uses the above mentioned formula to calculate similarity. It returns a value between -1 and 1. Closer the SSIM to 1, higher is the similarity. So, for every security feature, the SSIM value between each image of that security feature and the corresponding extracted feature from the test image will be calculated. Then, the mean SSIM for each security feature is calculated and stored.

F. ***Algorithm 2: 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)

1) Feature Extraction: In the first step, the region in which the bleed lines are present are extracted by cropping the image. So, a part near the left and right edges of the input currency note image is carefully extracted.

2) Image Thresholding: In the 2nd step, the image is thresholded using a suitable value. This ensures that only the black bleed lines remain on a white background and makes further processing quite easy.

3) Calculation of number of bleed lines: The 3rd step involves calculation of number of bleed lines. In this step, first we iterate over each column of the thresholded image. Then we iterate over each pixel of each column. Then, we calculate the number of black regions in each column by increasing a counter whenever current pixel of the column is white and the immediate next pixel is black. Similarly we, count number of black regions for each column, but, if the number of black regions is too large (>= 10), then that column is erroneous and it is discarded. Finally, the average count of black regions is calculated by considering the non- erroneous columns only and the result is displayed as the number of bleed lines. This count should be approximately 5 for Rs 500 currency notes and 7 for Rs 2000 currency notes.

G. ***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.

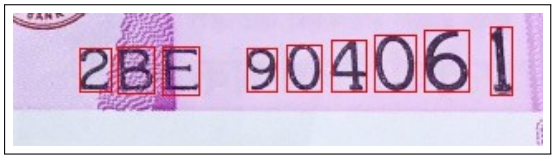


Fig. 5: Number Panel Detection

1) Image Thresholding (with multiple values): The 1st step in this algorithm is again thresholding with suitable value so that only the black characters remain in the number panel on a white background and become easy to detect. But in this algorithm, thresholding is done using multiple values, i.e. first the image is thresholded at the initial value (90), then all other steps mentioned below are done and number of characters are calculated. After that, the threshold value is increased by 5 every time and the process of calculation of number of characters is repeated till either we reach the final value (150, in our case) or we find enough proof that 9 characters are present in the number panel.

2) Contour Detection: In the 2nd step, contour detection of the thresholded image of number panel is done.

3) Finding Bounding Rectangles: In the 3rd step, the bounding rectangle for each contour is found. The details of each rectangle is put inside a list.

4) Eliminating erroneous rectangles: The list of rectangles computed in the previous step may contain a number of erroneous and unnecessary rectangles due to noise present in the image. These erroneous rectangles need to be eliminated. So, in this step, all rectangles whose area is either too big or too small are eliminated. Then, the rectangles which are bound by a bigger rectangle are also eliminated. Finally, those rectangles which are positioned completely too high in the number panel are also eliminated.

5) Calculation of number of characters: The rectangles remaining after the previous elimination step are those rectangles which bound only each character of the number panel. The number of rectangles still remaining is calculated and this gives us the number of characters detected in that particular thresholded image.

The above process is repeated for multiple threshold values (starting form 90 or 95 and increasing by 5 in each iteration). The algorithm stops if either it detects 9 characters in three consecutive iterations or if the threshold value reaches the maximum value (150 in our case).

H. ***Displaying Output***

Finally, the result of all algorithms is displayed to the user. The extracted image of each feature and the various important data collected for each feature is displayed properly in a GUI window. Further, the status (Pass/ Fail) of each feature is displayed along with the details. Finally the total number of features that have passed successfully for the input image of

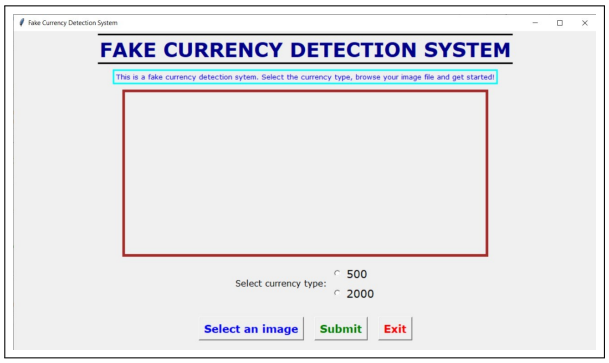
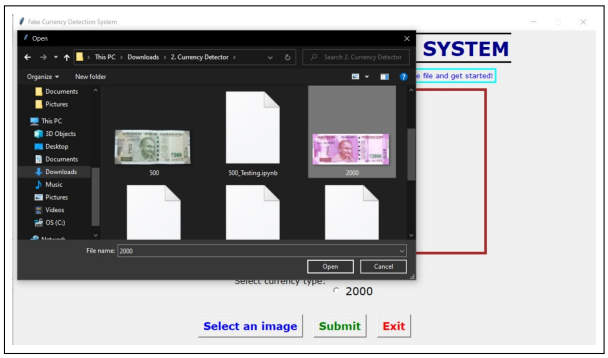


Fig. 6: Intially no image is dislayed



and user is asked to insert image

Fig. 7: Browsing image

Fig. 8: Input image of currency note

Fig. 9: Image sent for processing...

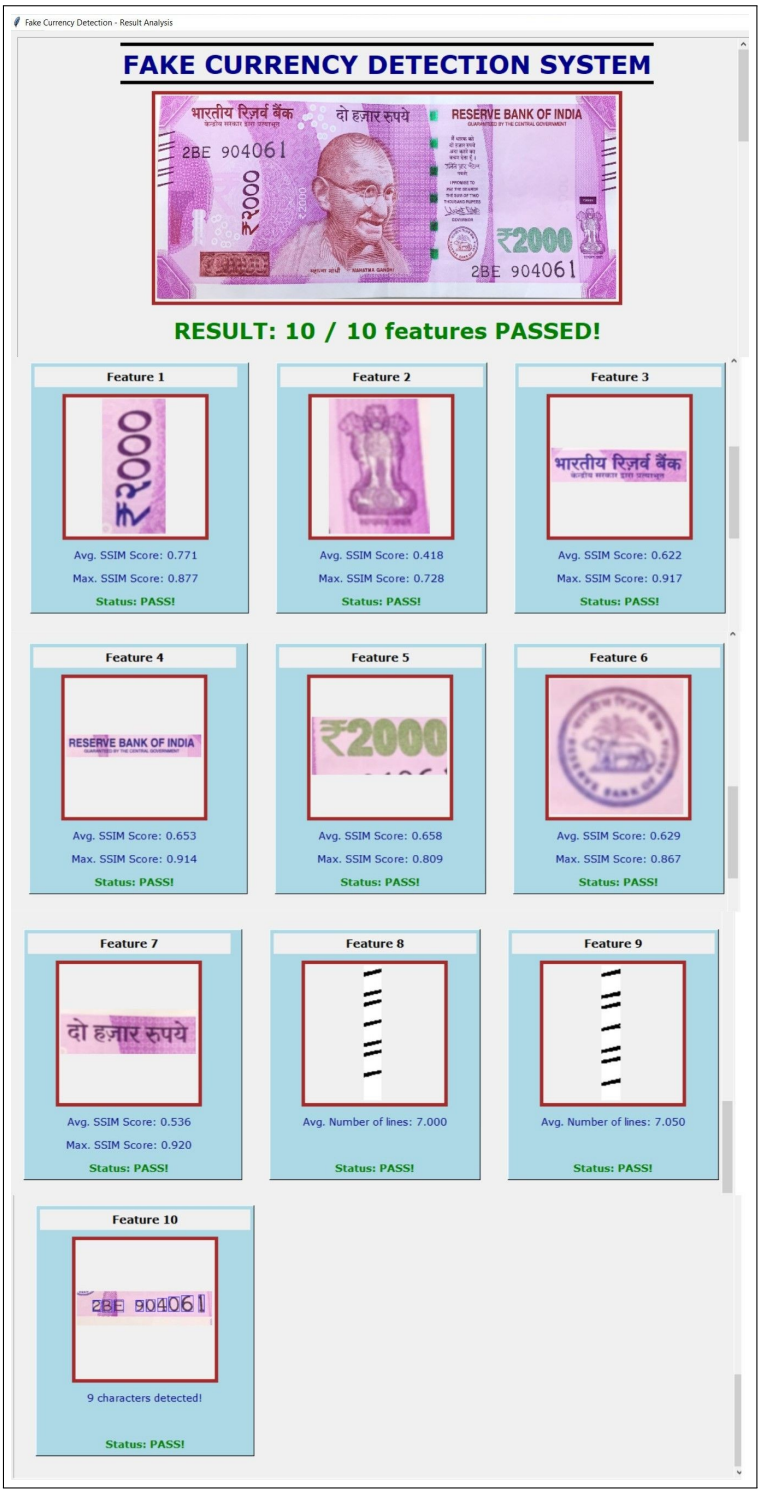


Fig. 10: GUI showing final result(Real note)

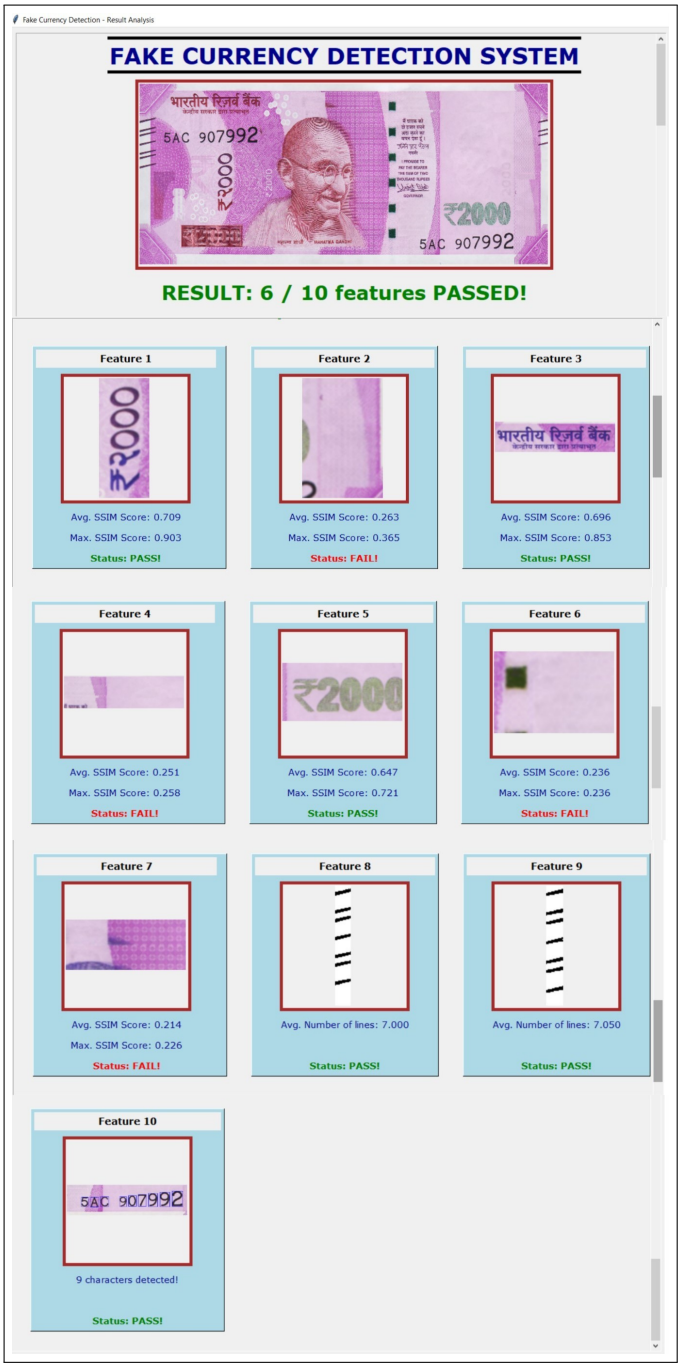


Fig. 11: GUI showing final result(Fake note)

currency note is displayed and based upon that it is decided whether the note is fake or not. The entire GUI is programmed in python itself using tkinter library.

V. Results and Analysis

The proposed system authenticates the input image of currency note through image processing. The input image passes through various algorithms in which the image is processed and each extracted feature is thoroughly examined. The results are calculated in the following manner:

- **Algorithm 1 (Feature 1-7):** This algorithm finally collects the average SSIM score and the max. SSIM score for each feature. A feature passes the test and is real if the average SSIM score is greater than a minimum value (this value has to be decided after proper testing). A feature also passes the test if the max. SSIM score is too high (probably greater than 0.8).
- **Algorithm 2 (Feature 8-9):** This algorithm finally returns the average number of bleed lines present in the left and right sides of a currency note. Each feature passes the test if the average number of bleed lines is closer to 5, in case of Rs 500 currency note, and 7, in case of 2000 currency note.
- **Algorithm 3 (Feature 10):** This algorithm finally returns the number of characters present in the number panel of the currency note. This feature passes the test if the number of characters detected is equal to 9 (for at least one threshold value).

Fig 6 to Fig 11 are snapshots of the GUI used in our implementation.

A. *Performance Analysis*

The performance analysis of the proposed system was carried out using various images of currency notes. As we had already created a dataset for both fake and real currency notes of denominations of 500 and 2000, all the notes were tested and then the accuracy was calculated. For the sake of calculating the accuracy, it was assumed that if the currency note passed at least 9 features out of 10 then the note is real otherwise it is fake. Testing of both real and fake notes was done separately.

- For testing of real notes, 9 real notes for Rs.2000 and 10 real notes for Rs. 500 were considered, out of which 15 of the total 19 notes gave the correct desired results. Accuracy: 79%
- Similarly, for testing fake notes, 6 fake notes were taken into consideration for each denomination (12 notes in total), for which 10 of the 12 notes gave the correct required output. Accuracy: 83%

The accuracy for both real and fake currency notes was calculated separately and the result is shown in the a bar graph (Fig 13).

B. *Time Analysis*

The proposed system has been implemented using Python programming language in Jupyter Notebook environment. Along with the final results, a lot of images and other analysis related data is being printed by our system. If all the data (consisting of 100+ images regarding the examination of each feature) is printed, then the system takes about 35 seconds to process, print all the data and give the final results. If only the final results are displayed after processing of input image, the system takes only 5 seconds for each input image.
So, practically the model takes about 5 seconds to give the results of each input image.

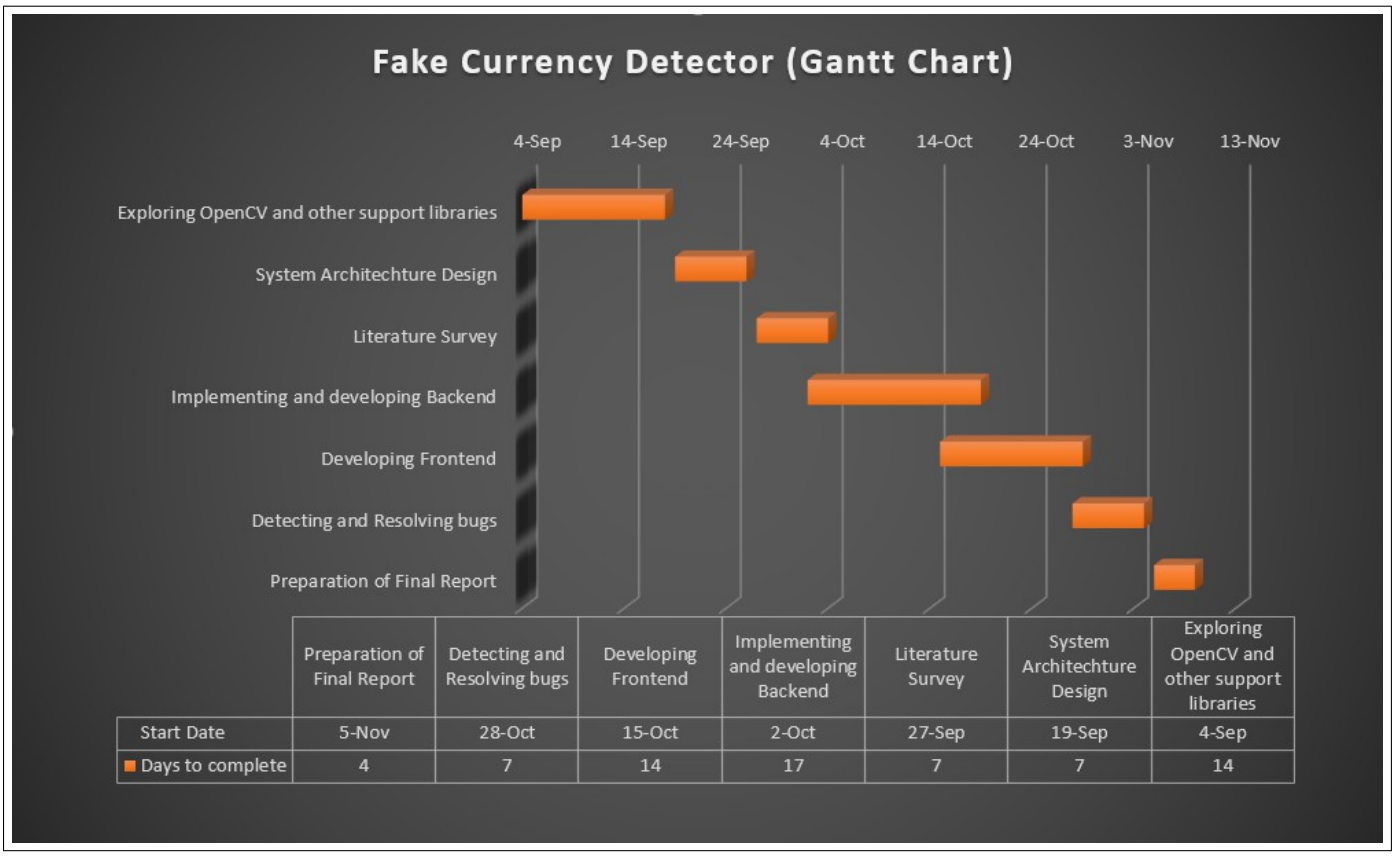


Fig. 12: Gantt Chart

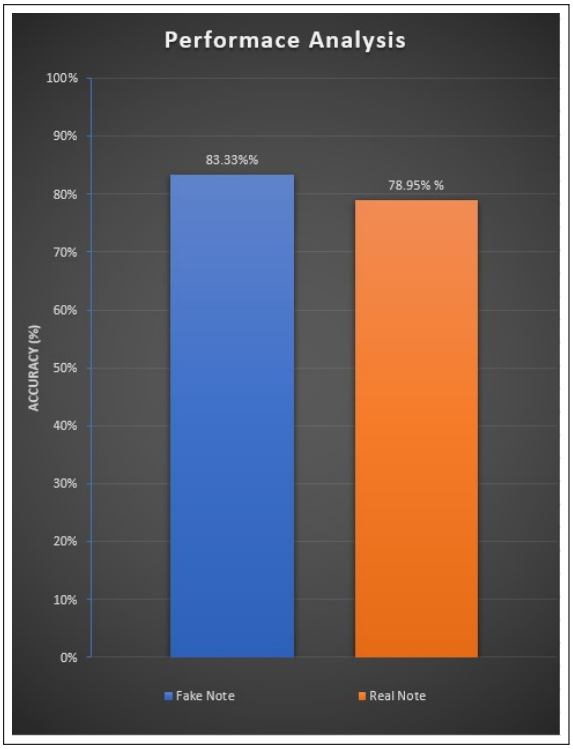


Fig. 13: Performance Analysis Graph

VI. Conclusion

In this paper, a fake currency detection model has been proposed for authentication of Indian currency notes of denomination 500 and 2000 and implemented using OpenCV image processing library in Python3. In this model, 10 features of the input currency note are considered and then analyzed using 3 different algorithms. The input image is taken through a GUI which allows the user to browse the image in his/her system. Then the results of the implemented model are computed and the analysis of each feature is displayed in detail through a graphical user interface (GUI) created using Tkinter GUI library.

The model takes less time (about 5 sec- when only final results are shown leaving unnecessary details) for processing an input image. The results are also quite decent giving almost 79% accuracy in detecting genuine currency and 83% accuracy in detecting counterfeit currency.

IMPLEMENTED/BASE PAPER

The initial idea and inspiration for our proposed system has been taken from the paper mentioned below. However, the methodology of this paper is quite different from our proposed system.

Paper name: Fake Indian Currency Note Recognition
Authors: Mangayarkarasi, P and Akhilendu and A S, Anakha and K, Meghashree and A B, and Faris
Published year: 2020

REFERENCES

1. S. R. Darade and G. Gidveer, "Automatic recognition of fake indian currency note," in *2016 international conference on Electrical Power and Energy Systems (ICEPES)*. IEEE, 2016, pp. 290–294.
2. B. P. Yadav, C. Patil, R. Karhe, and P. Patil, "An automatic recognition of fake indian paper currency note using matlab," *Int. J. Eng. Sci. Innov. Technol*, vol. 3, pp. 560–566, 2014.
3. A. Zarin and J. Uddin, "A hybrid fake banknote detection model using ocr, face recognition and hough features," in *2019 Cybersecurity and Cyberforensics Conference (CCC)*. IEEE, 2019, pp. 91–95.
4. M. S. Veling, M. J. P. Sawal, M. S. A. Bandekar, M. T. C. Patil, and M. A. L. Sawant, "Fake indian currency recognition system by using matlab."
5. F. A. B, P. Mangayarkarasi, Akhilendu, A. A. S, and M. K, "Fake indian currency note recognition," vol. 7, pp. 4766–4770, 2020. [Online]. Available: https://www.irjet.net/archives/V7/i5/IRJET-V7I5915.pdf