

Fake Currency Detector

Aditya Bhardwaj¹, Arya Tyagi², Ruchika Pandey³, Vrinda Batra⁴, Mr. Jagbeer Singh⁵
^{1,2,3,4,5}Dept. of CSE, MIET, Meerut

Abstract: -The proliferation of counterfeit currency poses a significant threat to both individuals and the national economy. Although fake currency detectors are available, their accessibility is confined to banks and corporate offices, leaving regular individuals and small businesses vulnerable to this threat. This initiative aims to investigate the security features of Indian currency and create a system which is based on software that employs advanced image processing and computer vision techniques for the identification and elimination of counterfeit Indian currency. The entire currency authentication system is crafted using the Python language within the Jupyter Notebook environment.

Keywords: fakeness of currency, counterfeiting detection, image processing, ORB detector, extracting features, brute force matcher.

1. Introduction

The widespread issue of duplicating currency or illegally producing counterfeit notes by mimicking legitimate manufacturing processes poses a significant challenge for every nation. Counterfeit currency has the potential to devalue genuine money, leading to inflation through an unauthorized and unnatural expansion of the money supply. While manual authentication of currency notes is a viable solution, it is a time-consuming, imprecise, and challenging process. Consequently, automatic testing of currency notes becomes essential to efficiently handle large volumes of currency and obtain accurate results within a short timeframe. In this project, we advocate for a system designed to detect counterfeit currency with the help of diverse image-processing techniques and algorithms.

The developed system is designed to authenticate Indian currency notes in 500 and 2000 rupee denominations. With three primary algorithms, the system thoroughly examines distinct features within the currency notes. The first algorithm involves a series of steps, such as image capturing, preprocessing, greyscale conversion, extraction of features, segmentation of image, and comparison using advanced image processing techniques such as ORB and SSIM. This system offers a straightforward method for swiftly and accurately authenticating currency notes. It serves as an automated alternative to current manual methods, accessible for easy use by anyone to identify counterfeit currency.

A. Security Features commonly used to Detect Fake Notes:

- 1) **Bleed lines:** There are angular bleed lines on 500 and 2000 note on left and right corners of the note in raised print. In 500Rs. note there are 5 bleed lines and In 2000Rs. note, 7 bleed lines.
- 2) **Security Feature:** A security thread with a color-shifting effect bears the inscription "Bharat" in Hindi, alongside RBI and 2000 (or 500 for the 500-rupee note). The thread's color transitions from green to blue upon tilting.
- 3) **Latent Image:** Latent image of numbers 2000 or 500 can be seen when note is held at 45 degrees angle.
- 4) **Water mark:** A watermark of Mahatma Gandhi and electrotype of numeral 2000/500.
- 5) **Denominational Numeral:** A see-through register featuring the denominational numeral 2000 becomes visible when the note is held against the light.
- 6) **Portrait of Mahatma Gandhi:** Portrait of Mahatma Gandhi with RBI written on his spectacle, which can be read using a magnifying glass.
- 7) **Number panel:** Sequential numerals, increasing in size, are printed on both the upper left and lower right sides.
- 8) **Denominational numeral:** On left side of Mahatma Gandhi, there is a 500/2000 in Devnagari script.

-
- 9) *Ashoka Pillar*: On the bottom right side of the note, there is Ashoka Pillar.
- 10) *Guarantee and Commitment clause*: The guaranteed and promised clause of the Reserve Bank of India is displayed in both Devnagri and English, positioned in the top left and top right corners of the currency notes, respectively.
- 11) *RBI seal*: The Reserve Bank of India's seal is located just beneath the Governor's signature. Both the seal and the guarantee clause, among other elements, are produced using 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.

2. Literature Survey

This paper presents the development of an automated system to distinguish between genuine and counterfeit Indian currency notes. The automatic system holds significant utility not only within the banking sector but also across various other fields. India has witnessed a rise in counterfeit currency notes, particularly in denominations of 100, 500, and 1000 rupees, with advancements in technology such as scanning, color printing, and duplication contributing to this issue. In this proposed model, the process begins with image acquisition, followed by pre-processing steps involving cropping, smoothing, and adjustment. Subsequently, the image is converted to grayscale, and image segmentation is applied, leading to feature extraction and reduction. Finally, the model compares the processed image for authentication.

This paper details the implementation of an automated system for the detection of counterfeit Indian paper currency notes using MATLAB. The system employs feature extraction with the HSV color space and various image processing techniques. The proposed architecture consists of the following stages: Image Acquisition, Gray-Scale Conversion, Edge Detection, Image Segmentation, Characteristic Extraction, Comparison, and Output. In the project setup, a currency note is positioned in front of a camera to determine its authenticity. MATLAB, installed on a computer, analyzes the captured images of the notes. This algorithm is specifically designed for Indian denominations of 100, 500, and 1000 rupees. If the note is found to be genuine, a corresponding message is displayed on the screen, and vice versa.

This paper introduces a hybrid model for the detection of counterfeit currency, specifically focused on Bangladeshi notes, and it has been implemented using MATLAB. The model integrates three image processing algorithms: Optical Character Recognition (OCR), Hough Transformation, and Face Recognition (MSD), aiming to achieve improved results. The proposed algorithm consists of six essential steps: data collection, pre-processing of the collected data, edge detection, feature extraction, identification, and output results. Despite a slightly longer processing time, the proposed model demonstrated an impressive accuracy of 93.33%, surpassing the individual algorithms applied in isolation.

This paper employs two approaches: the first one is, by employing analysis through hyperspectral imaging, and the second one, by extracting distinct features in counterfeit and genuine currency notes. By comparing these features, we can effectively differentiate between fake notes and real notes. Various colored lights, including Ultraviolet (UV) light, Normal LED Bulb, Red LED light, Green LED light, and Blue LED light, each with different wavelengths ranging from 360 nm to 800 nm, are utilized for the hyperspectral imaging note detection unit with image processing algorithms. The experimental results indicate that the results achieved are nearly accurate.

This paper elucidates the recognition and verification of paper currency utilizing image processing techniques. The proposed approach encompasses various stages, including Image Acquisition, Feature Extraction and Comparison, Texture Features, and Voice Output. The system is bifurcated into two components. The initial segment focuses on identifying the currency denomination through image processing, while the second part involves providing an oral output to inform visually impaired individuals about the denomination of the note they possess. The intended outcomes encompass both text and voice outputs indicating the recognized and verified currency.

3. 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 primary goal of the project is to automatically detect counterfeit Indian currency notes 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 a User-friendly interface, to make it convenient to use and understand.

4. 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 are to be considered 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 1: Summary of Literature Survey

Authors	Methodology	Merits	Limitations
Sonali R. Darade[1]	Feature extraction and image 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 image processing	Whole setup required
Adiba Zarin, Jia Uddin[3]	Optical Character Recognition (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 extraction	Cost and time efficient	If note is dirty and torn than it will give wrong answer

5. Image Acquisition

Afterwards, the system accepts the image of the test currency note, which ideally should be captured with a digital camera or, preferably, scanned. 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.

6. Pre-processing

Next, the input image undergoes pre-processing, starting with resizing it to a fixed size for streamlined computations. 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.

7. 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 simplifies computation and processing for grayscale images.

8. 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 enhance the accuracy and ease of searching for the security feature (template image), a defined search area will be specified on the test currency image where the template is anticipated to be located. Subsequently, ORB will be employed to detect the template within the test image, and the outcome will be appropriately highlighted with a marker. This procedure will be repeated for each image of every security feature in the dataset, ensuring that the identified section of the test image is consistently highlighted using appropriate 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 region is subsequently extracted by slicing the 3D pixel matrix of the image. Next, we apply Grayscale 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.

9. Gray-scale conversion



Fig.2: Features in 500 currency bill

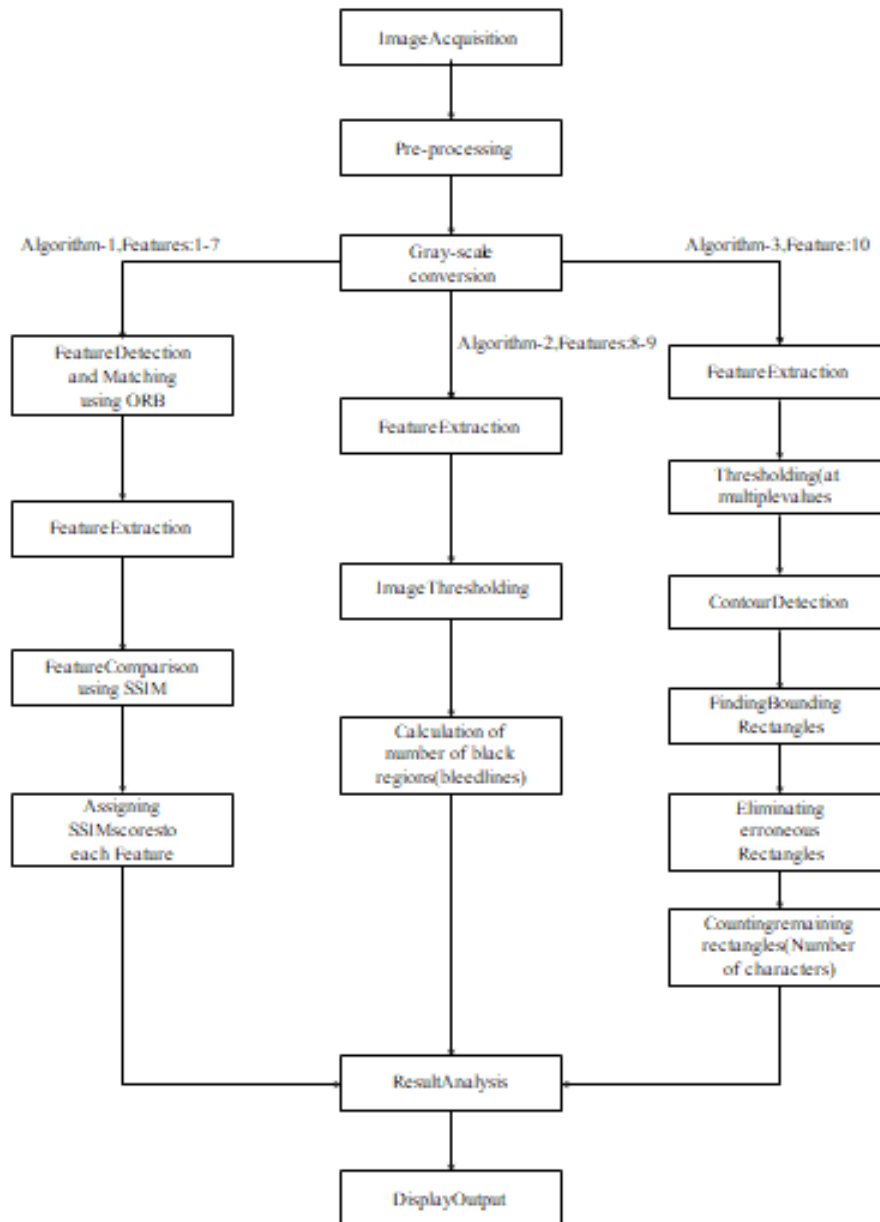


Fig.4: Flow Diagram



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 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.

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

The Structural Similarity Index (SSIM) serves as a scoring system to quantify the degradation in image quality resulting from processes like data compression or losses during data transmission. Essentially, it assesses the similarity between two images. Implemented within the image library, it utilizes the mentioned formula to compute similarity, yielding a value between -1 and 1. A higher SSIM value, closer to 1, indicates greater similarity. Consequently, for each security feature, the SSIM value between each image of that feature and the corresponding extracted feature from the test image is computed. Subsequently, the mean SSIM for each security feature is calculated and stored.

10. 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. The algorithm is utilized to tally and authenticate the count of bleed lines located on both the left and right sides of a currency note (features 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. Therefore, a section adjacent to the left and right edges of the input currency note image is meticulously isolated.
- 2) *Image Thresholding*: In the second step, the image undergoes thresholding with an appropriate value. This guarantees that only the black bleed lines are retained against a white background, simplifying subsequent processing.

Calculation of number of bleed lines: The third step involves determining the number of bleed lines. Initially, we iterate over each column of the threshold image. Subsequently, for each pixel in every column, the count of black regions is calculated by incrementing a counter whenever the current pixel in the column is white and the immediately following pixel is black. Simultaneously, we count the number of black regions for each column. However, if the count of black regions is excessively large (≥ 10), indicating an error, that particular column is disregarded. Ultimately, the average count of black regions is computed by considering only the non-erroneous columns. The result is then presented as the number of bleed lines. For Rs 500 currency notes, this count should be approximately 5, and for Rs 2000 currency notes, it should be around 7.

11. 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.

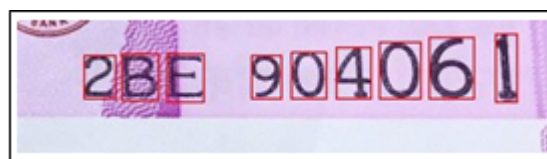


Fig.5: Detecting Number Panel

- 1) *Image Thresholding (with multiple values)*: In the first step of this algorithm, thresholding is performed using a suitable value to isolate black characters within the number panel against a white background, facilitating their easy detection. However, unlike previous approaches, thresholding here is carried out using multiple values. Initially, the image is thresholded at the starting value (90), and the character count is calculated following all subsequent steps mentioned below. Then, the threshold value is iteratively incremented by 5, and the process of character count calculation is repeated until either the final value (150, in our case) is reached, or evidence of 9 characters in the number panel is sufficient.
- 2) *Contour Detection*: In the next step, contours are identified within the thresholded image of the number panel.
- 3) *Finding Bounding Rectangles*: Next, the algorithm establishes the bounding rectangle for each contour, with the details of each rectangle being stored in a list.
- 4) *Eliminating erroneous rectangles*: The collection of rectangles calculated in the preceding step might include various inaccurate and unnecessary rectangles caused by noise 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. Subsequently, rectangles enclosed by a larger rectangle are excluded. Ultimately, rectangles positioned excessively high within the number panel are also removed.
- 5) *Calculation of number of characters*: The rectangles that persist following the preceding elimination step are specifically those that enclose individual characters within the number panel. The algorithm then calculates the remaining number of rectangles, providing the count of characters detected in that particular thresholded image. The process is iteratively performed with multiple threshold values (starting from 90 or 95, incrementing by 5 in each iteration). The algorithm halts if it identifies 9 characters in three consecutive iterations or if the threshold value reaches the maximum (150 in our case)

12. 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 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.



Fig. 6: At the start, no image is exhibited, prompting the user to insert an image



Fig.7: Browsing image



Fig.8: Input image of currency note

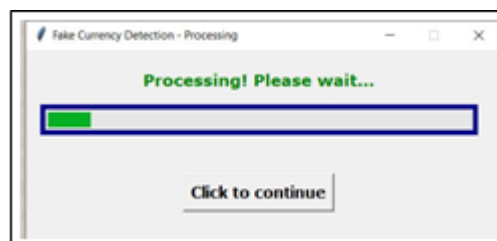


Fig.9: Image has been submitted for processing

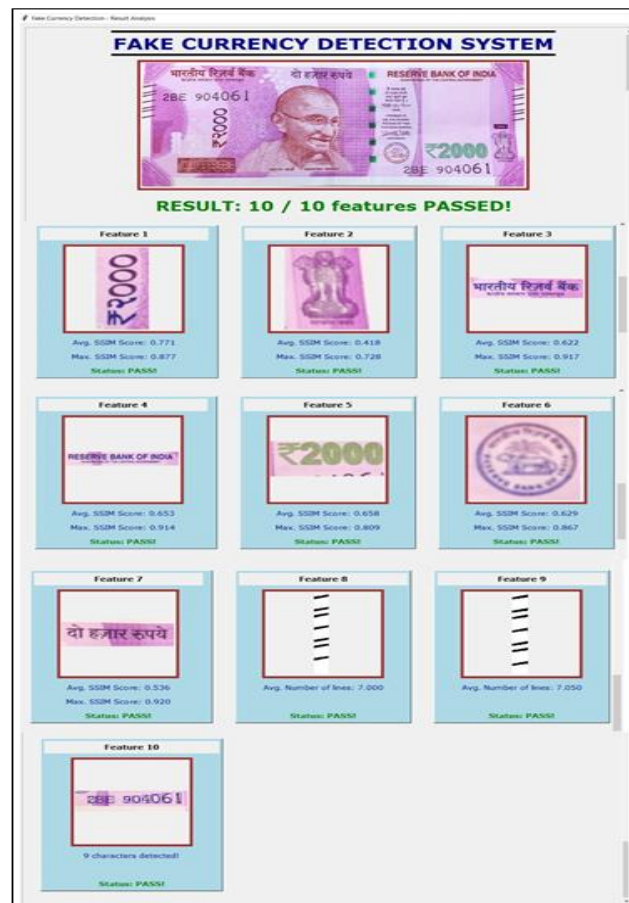


Fig.10: Final result shown by GUI (Real note)

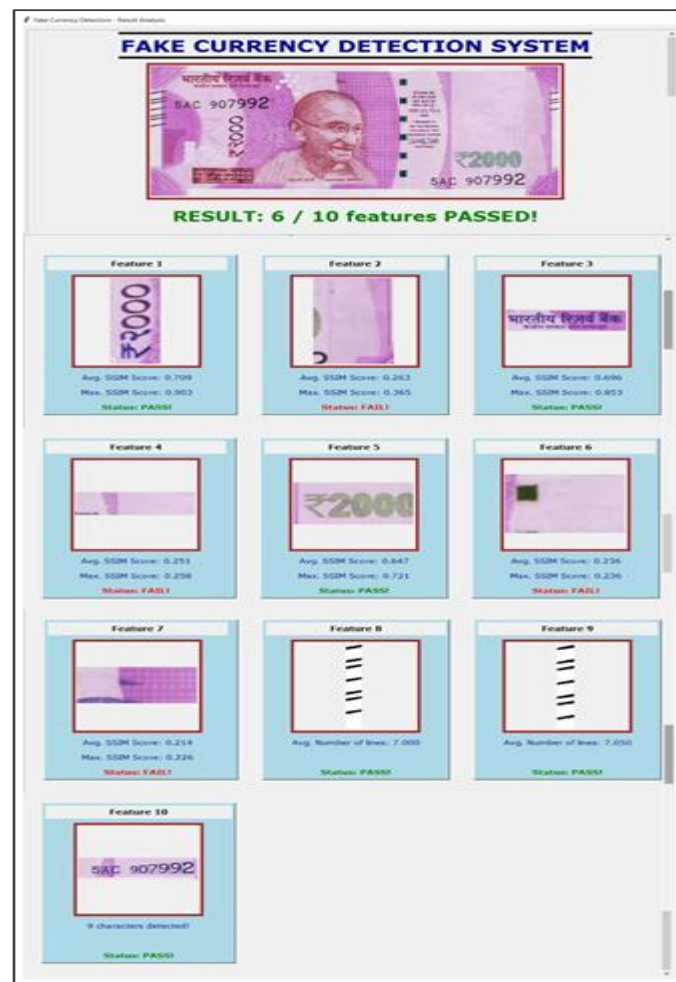


Fig.11:Final result shown by GUI (Fakenote)

13. 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 gathers the average SSIM score and the maximum SSIM score for each feature. A feature is considered genuine if its average SSIM score exceeds a determined minimum value (to be established through thorough testing). Additionally, a feature is validated if its maximum SSIM score is notably high, likely surpassing 0.8.
- Algorithm 2 (Feature 8-9): This algorithm finally calculates the average number of bleed lines on both the left and right sides of a currency note. Each feature is deemed successful if the average number of bleed lines approximates 5 for the Rs 500 currency note and 7 for the Rs 2000 currency note.
- Algorithm 3 (Feature 10): This algorithm ultimately provides the count of characters found in the number panel of the currency note. Success is achieved if the detected character count equals 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 proposed system underwent performance analysis using a diverse set of currency note images. Utilizing a dataset comprising both fake and real currency notes with denominations of 500 and 2000, all notes were tested,

and the accuracy was subsequently calculated. To assess accuracy, it was presumed that a currency note is genuine if it passes at least 9 features out of 10; otherwise, it is considered fake. The testing process was conducted independently for both real and fake notes.

- In the testing of genuine notes, 9 Rs. 2000 notes and 10 Rs. 500 notes were examined, resulting in 15 out of the total 19 notes producing the correct desired results. The accuracy achieved was 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 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.

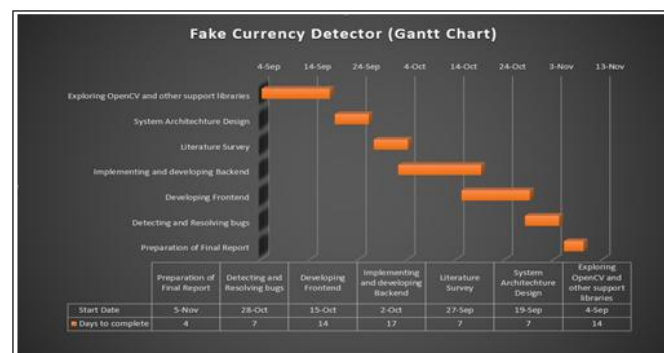


Fig.12: Gantt Chart

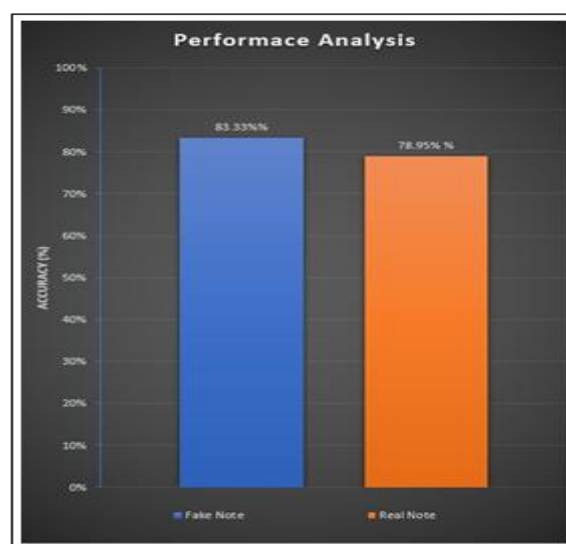


Fig.13: Performance Analysis Graph

14. Conclusion

In this paper, a fake currency detection model has been proposed for authentication of Indian currency notes of de- nomination 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. Subsequently, the model's results are calculated, and a detailed analysis of each feature is presented through a graphical user interface (GUI) generated using the Tkinter GUI library.

This 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.

Acknowledgment

We are immensely grateful to Mr. Jagbeer Singh for his invaluable contributions, for sharing his expertise as a guide every week and for his kind co-operation and encouragement in completing this project and their valuable guidance during the development of this project.

Implemented/BasePaper

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 indiancurrencynote," 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
- [5] M. A. L. Sawant, "Fake indian currency recognition system by using matlab."
- [6] 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>