Currency Recognition System Using Image Processing

Dr. Ritu Sindhu, Pranshu Verma
Department of Computer Science and Engineering
Galgotias University
Greater Noida, U. P., India
ritusindhu@galgotiasuniversity.edu.in
pranshuv00@gmail.com

***Abstract*--In this paper, we proposed an automatic currency recognition system using digital image processing methodology. The following project mainly focuses on the recognition of currency by its image or photograph. It will help users to recognize details about currency like Currency Value, Currency Name, the value in INR, EURO and US Dollar. It works using the main characteristics of currency notes such as size colour or printed text on it and also depends on differ in currency notes within the same country. We have considered INDIAN Rupee and US Dollar, the most used currencies in our domain with their denominations. This system works accurately and also able to quickly identify the currency notes.**

Keywords—Image Processing, BGR to Greyscale, Binary Threshold, Currency Name, Currency Denomination (Value), Trained Dataset, Current Exchange Rate.

I. INTRODUCTION

According to the UN charter there are around 195 countries around the globe. In which 193 countries are members of the UN and two are observing states. According to The U.N., worldwide there are 180 currencies. All these currencies are different in characteristics such as size, colour and texture. In the era of rapidly growing levels of trade between countries and also tourism all over the world, it becomes necessary to recognize each currency note correctly. Now a days people travel to different countries, they use their native country currency in paying bills or buying stuffs and because most of the local people are not familiar with the currency other than their own country currency and also not familiar with the exchange rate of that currency in their own currency, it becomes necessary to develop an automated system that helps in recognition notes easily, faster with efficiency.

The proposed system is based on image processing and makes the process robust and automatic. We used INR and USD as an example to illustrate the technique. This system is based on our knowledge about computer science technologies like Digital Image Processing, python and also a small step to implement in a system that is most important for industrial development.

We had considered INDIAN Rupee and US Dollar for this project. Project can expand for more currencies inclusion according to use.

The block diagram as in Fig. 1 shows the used methods in the system to obtain the actual output same as the expected output. First of all input a currency note image, which needs to be checked. Then system performs basic image processing techniques on input image and refined to convert it into a usable input for matching with present image in dataset. The system extracts the information from image based on features such as colour and texture. With the help of these features the system determines currency name and denomination of the currency note. After this the exchange rate of currency will be extracted from internet with the help of online exchange rate api url. At the end output displayed on the screen.

A. RELEVANCE

The proposed system is built with the motive to help common people for currency recognition at anytime and anywhere. When people see a new currency that doesn’t belong to their native country then they face many issues in recognition that currency’s value, origin country and also face difficulties in finding the exchange rate. It is need to develop a system which will help people to identify currency value, currency name and exchange rate only by providing currency image.

In this paper we proposed an automation system for easily recognition of currency. The system uses digital image processing techniques to extract information from inserted image and match with the template images.

To implement image processing techniques we used python as programming language because it is much easier to implement digital image processing by using python because of its pre-defined packages and frameworks. NumPy and OpenCv are the two frameworks which are used to perform image processing functionalities and Tkinter is used to design applet of the application.

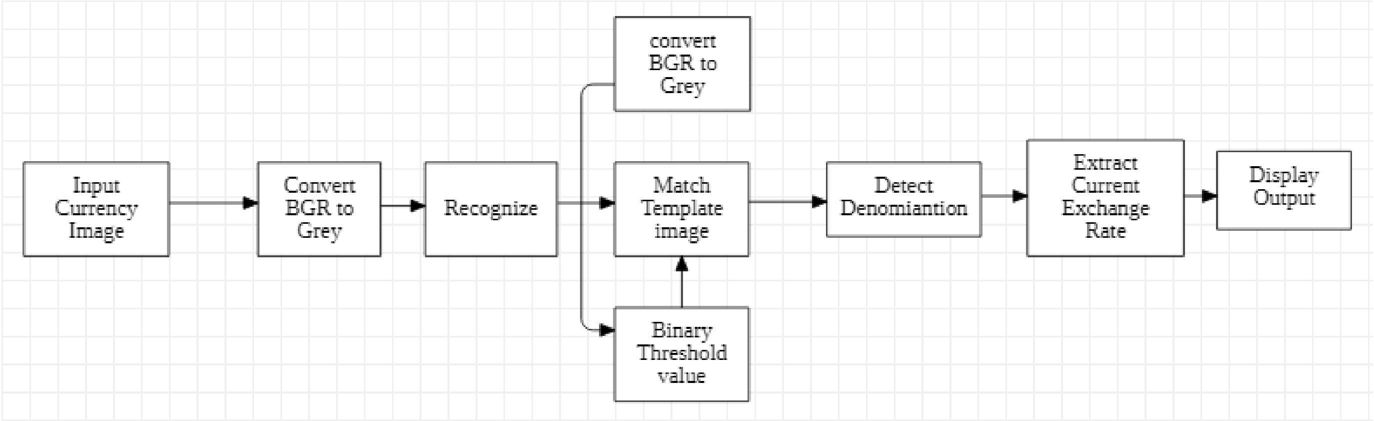


Fig. 1 Block diagram of the Proposed System

II. LITERATURE SURVEY

[1] In 2010, 'K. K. Debnath et al' proposed a system for currency recognition using Ensemble Neural Network (ENN). In this model, negative correlation learning is used to train the Individual Neural Networks (NNs) in an ENN. They used different types of notes such as noisy and old notes because the machine doesn't easily recognize such notes. That's why ENN is used to develop a system to identify them easily and correctly. For testing, they used different dominations notes, which are of 2, 5, 10, 20, 50, 100 and 500 TAKA. The system firstly converts the note image into greyscale and then the image is compressed. After compression, the compressed image is processed to the system as an input for recognition. Due to ENN, the system can easily identify the currency with noise as well as old currency notes. The system has fewer chances of misclassification because of independent training.

[2] In 2010, 'J. Guo et al' proposed a model that uses the Local Binary Pattern (LBP) algorithm for paper currency recognition. It extracts currency features with good quality for efficient recognition of currency notes. LBP algorithm is used for the extraction of characteristics of a currency. LBP algorithm is based on the LBP method and the LBP method has the advantage of simplicity and high speed.

[3] In 2011, 'Jahangir and Raja' proposed a system to recognize Bangladeshi currency with the help of the neural network recognition method. This method had implemented on cost-effective hardware so that can be used in different places. The system takes a currency note image as input and notes are scanned using sensors. The trained data set is created using the backpropagation algorithm. Axis symmetric mask is used in the pre-processing stage for correct recognition of currency notes. They used eight TAKA notes for testing the system and the result was successfully recognition of all notes.

[4] In 2013, 'Qing and Xun' proposed a model which was based on Artificial Neural Networks and Gene Algorithm. ANN and Gene algorithm are problem-solving techniques. Gene Algorithm had used because of slow convergence and indeterminate initial weights for backpropagation neural networks. The Gene Algorithm: Back Propagation takes short training time with faster recognition speed and provides the appropriate result of connection weights and network connection.

[5] In 2000, 'Y. Mitsukura et al' proposed a model that depends on Artificial Intelligence methods and adaptive systems, combining Neural Network methods with the help of the Simulated Annealing Method. The proposed system was low cost with a recognition rate result of 99.68%. The result was achieved by developing small size neural network for coin recognition with the help of a genetic algorithm and simulated annealing.

III. OBJECTIVE of the SYSTEM

- The main objective of this system is to recognize currency using image processing technique.
- The system performs image processing on inserted image and match with trained data set.
- To save user time it needs device should be connected to internet so that it can extract current

exchange rate of that currency into other mainly used currencies.

- For using this no need for high specs hardware and easy to use.

IV. PROPOSED MODEL

Flowchart for proposed system Fig. 2:

1. Start
2. Give input as image (JPG/PNG/GIF format).
3. Conversion of image from BGR to Greyscale.
4. The Binary Threshold value will be assigned with the pixels of greyscale image.
5. Match the greyscale image with template image.
6. If image matched with the trained data set then show the defined output. And if not matched with any image in data set then return error with message.
7. Stop the process

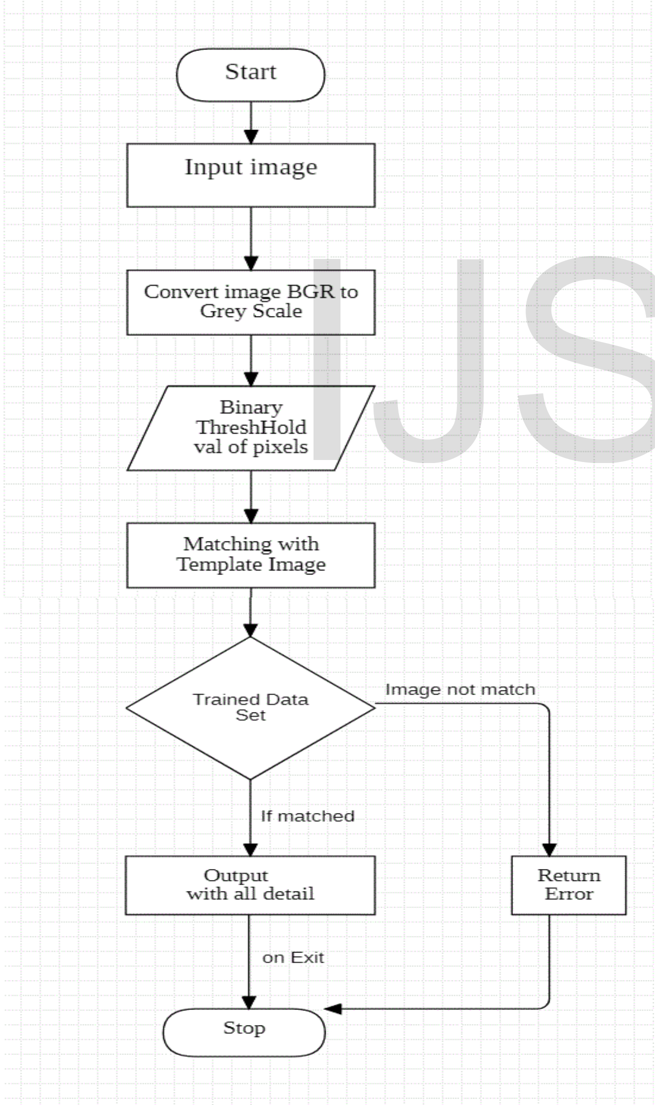


Fig. 2 Flowchart of the system

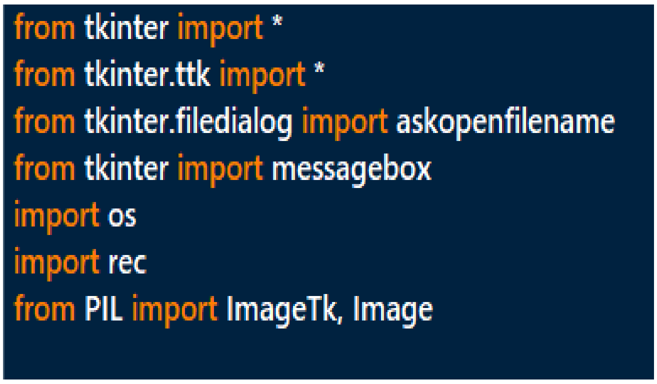
A. METHODOLOGY

For Image processing and Template matching:

```
1. img = cv2.imread(filename) /*for taking input image and read that image*/
2. gray_img=cv2.cvtColor (img,cv2.COLOR_BGR2GRAY) /*conversion of input image from BGR to Greyscale.*/
3. temp ="0" /*initializing a variable temp with 0 value*/
   3.1. entries = os.listdir("trained_data/") /*obtaining list of all files/directories from trained dataset.*/
4. for entry in entries:
   4.1. template = cv2.imread("trained_data/"+entry, cv2.THRESH_BINARY) /*Assigning binary threshold value to the entered greyscale image pixels, 1 for white pixels and 0 for black pixels.*/
   4.2. w, h = template.shape[::-1] /* w= width and h=height and template.shape used to find and search location of a template image in greyscale image.*/
   4.3. result = cv2.matchTemplate(gray_img, template, cv2.TM_CCOEFF_NORMED) /*compare a template with the overlapped image regions.*/
   4.4. loc = np.where(result >= 0.4) /* returns indices of image where matching satisfies.*/
   4.5. for pt in zip(*loc[::-1]):
        cv2.rectangle(img, pt, (pt[0] + w, pt[1] + h), (0, 255, 0), 0)
   4.6. temp=entry
        break
   4.7. imageid = temp.replace(".jpg","")
        print(">>> Recognized with Image Id : " + imageid)
5. if imageid is None: /* if image doesn’t match with any image in trained dataset.*/
   5.1. window.windows.error("Sorry Image Not Recognised") /* gives error with message in applet window.*/
6. else:
   6.1. imageid = int(imageid) – 1 /* and when image match with template image.*/
   6.2. if currencydata[int(imageid)]['currency'] == "Indian Rupee": /* if inserted image matched with Indian Currency template image.*/
        url = https://api.exchangeratesapi.io/latest?base=INR
        /* url for getting live exchange rate in USD and Euro.*/
        6.2.1. response = requests.get(url) /* requesting exchange rate from url server.*/
        6.2.2.  data = response.text /* reading response data text from server.*/
        6.2.3.  parsed = json.loads(data) /* Loading parsed for using json data into python.*/
        6.2.4.  date = parsed["date"] /*parsing of date.*/
        6.2.5.  eur_rate = parsed["rates"]["EUR"] /* parsing of Euro exchange rate w r t INR.*/
        6.2.6.  usd_rate = parsed["rates"]["USD"] /* parsing of USD exchange rate w r t INR.*/
        6.2.7.  print("Currency Value : " , currencydata[int(imageid)]['value']) /*for printing currency value on output screen.*/
        6.2.8.  print("Currency Name : " , currencydata[int(imageid)]['currency']) /*for printing currency name on output screen.*/
        6.2.9.  print("Currency value In EUR:" , eur_rate*currencydata[int(imageid)]['value']) /* for printing exchange rate in EURO on output screen.*/
        6.2.10. print("Currency value In USD:" , usd_rate*currencydata[int(imageid)]['value'] ) /* for printing exchange rate in US Dollar on output screen.*/
        6.2.11. currvalue = currencydata[int(imageid)]['value'] /* obtaining currency value from trained dataset.*/
        6.2.12. currname=currencydata[int(imageid)]['currency'] /*obtaining currency name from trained dataset.*/
        6.2.13. curreur=eur_rate*currencydata[int(imageid)]['value'] /*For getting currency exchange rate with respect to euro. */
        6.2.14. currusd = usd_rate * currencydata[int(imageid)]['value'] /*printing current exchange rate with respect to USD.*/
        6.2.15. window.windows.getind(currvalue, currname, curreur, currusd,self) /* for displaying output on application window.*/
    6.3. else:
        6.3.1.  url="https://api.exchangeratesapi.io/latest?
        6.3.2.  base=USD" /* url for getting live exchange rate in INR and Euro.*/
        6.3.3.  response = requests.get(url) /* requesting exchange rate from url server.*/
        6.3.4.  data = response.text /* reading response data text from server.*/
        6.3.5.  parsed = json.loads(data) ) /* Loading parsed function for using json data into python.*/
        6.3.6.  eur_rate = parsed["rates"]["EUR"] /*parsing of Euro exchange rate w r t USD.*/
        6.3.7.  inr_rate = parsed["rates"]["INR"] /*parsing of USD exchange rate w r t INR.*/
        6.3.8.  print("Currency Value : " , currencydata[int(imageid)]['value'])/*for printing currency value on output screen.*/
        6.3.9.  print("Currency Name : " , currencydata[int(imageid)]['currency']) /*for printing currency name on output screen.*/
        6.3.10. print("Currency value In EUR:" , currencydata[int(imageid)]['value'] * eur_rate) /* for printing exchange rate in EURO on output screen.*/
        6.3.11. print("Currency value In INR:" , currencydata[int(imageid)]['value'] * inr_rate ) /* for printing exchange rate in INR on output screen.*/
        6.3.12. currvalue=currencydata[int(imageid)]['value'] /* obtaining currency value from trained dataset.*/
        6.3.13. currname=currencydata[int(imageid)]['currency'] /* obtaining currency name from trained dataset.*/
        6.3.14. curreur=currencydata[int(imageid)]['value'] * eur_rate /*printing current exchange rate with respect to USD.*/
        6.3.15. currinr=currencydata[int(imageid)]['value'] * inr_rate /*printing current exchange rate with respect to INR.*/
        6.3.16. window.windows.getusd(currvalue, currname, curreur, currinr,self) /*for displaying output on application window.*/
```

For applet:

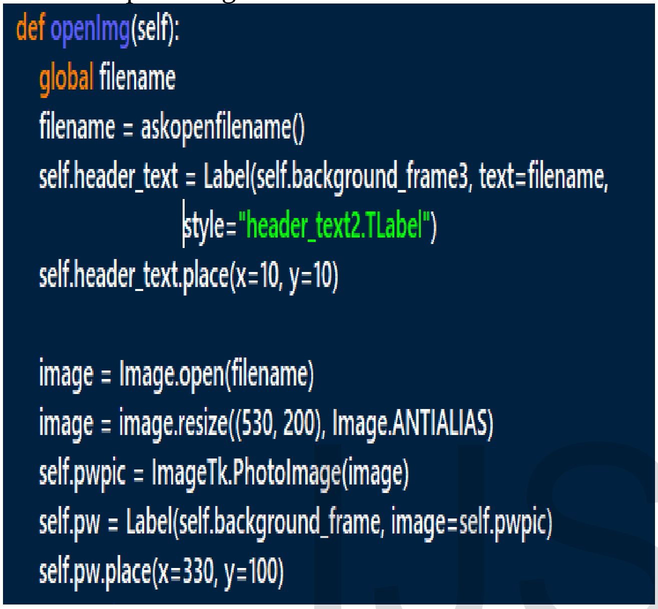
1. Imported frameworks and modules Fig. 3:



```
from tkinter import *
from tkinter.ttk import *
from tkinter.filedialog import askopenfilename
from tkinter import messagebox
import os
import rec
from PIL import ImageTk, Image
```

Fig. 3 Frameworks and Modules

2. ‘Open Image’ button:

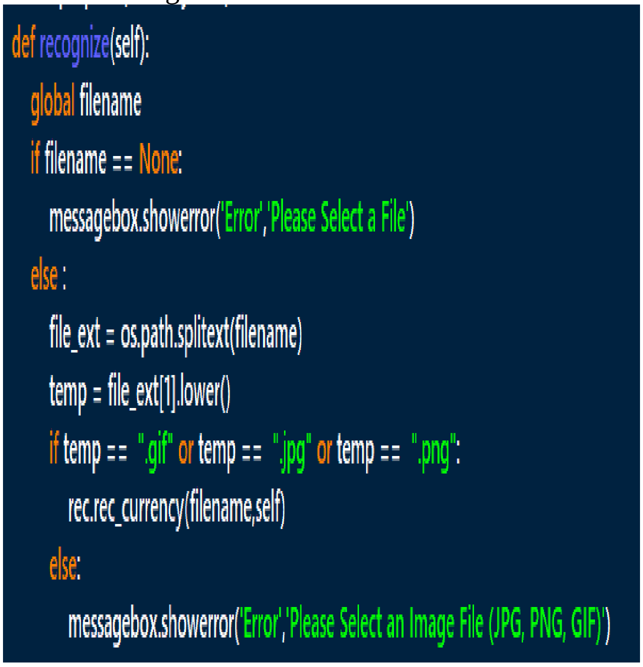


```
def openImg(self):
    global filename
    filename = askopenfilename()
    self.header_text = Label(self.background_frame3, text=filename,
                    style="header_text2.TLabel")
    self.header_text.place(x=10, y=10)

    image = Image.open(filename)
    image = image.resize((530, 200), Image.ANTIALIAS)
    self.pwpic = ImageTk.PhotoImage(image)
    self.pw = Label(self.background_frame, image=self.pwpic)
    self.pw.place(x=330, y=100)
```

Fig. 4 Open Image button

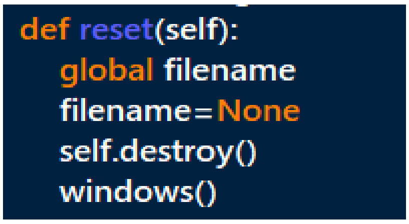
3. ‘Recognize’ button:



```
def recognize(self):
  global filename
  if filename == None:
    messagebox.showerror('Error','Please Select a File')
  else :
    file_ext = os.path.splitext(filename)
    temp = file_ext[1].lower()
    if temp == ".gif" or temp == ".jpg" or temp == ".png":
      rec.rec_currency(filename,self)
    else:
      messagebox.showerror('Error','Please Select an Image File (JPG, PNG, GIF)')
```

Fig. 5 Recognize button

4. ‘Reset’ button:



```
def reset(self):
    global filename
    filename=None
    self.destroy()
    windows()
```

Fig. 6 Reset button

B. SYSTEM DESCRIPTION

Input Image: An image file having extension JPG/PNG/GIF will be provided initially. The image should be of currency notes (INR, USD). It can take USD (1, 2, 5, 10, 20, 50, 100) and INR (1, 2, 5, 10, 20, 100, 200, 500, 2000).

Image Conversion: To convert image from BGR to greyscale we used OpenCv and numpy modules of python. It performs BGR to greyscale conversion using digital image processing. It performs the conversion and sets the binary threshold value for each pixel in greyscale image to perform template matching. Binary Threshold assigns values to every pixel in binary form. Value 1 is assigned to white pixels and 0 to black pixels.

Trained data set: We have trained data set for all currency notes which we have considered in our system. We have around 2 to 4 samples of each currency note in our trained data set which we have considered.

Template Matching: It is the step where converted image matched with the trained data set to recognize the currency and show its value. If data doesn’t match with any template then it will give error and show a message.

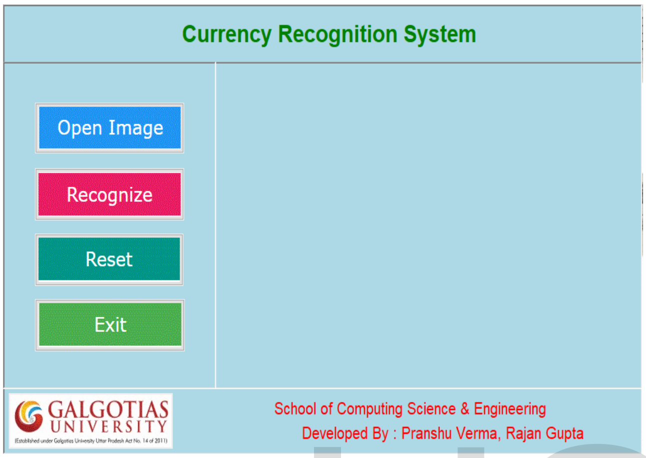
Current Exchange Rate: We used an url to show the current exchange rate in INR, USD/EURO. It will make currency recognition much easier for people with exchange denomination and also will save time. But the essential condition is that we need an active internet connection and the device should be connected to the internet. Exchange rate works with url server to extract current exchange rate and multiply with the detected currency denomination. Output will be print on application window.

V. RESULT

Firstly we run the application, welcome window will open as in Figure 7. Welcome window have 4 buttons ‘Open Image’, ‘Recognize’, ‘Result’ and ‘Exit’ button.

Open Image: When you click on this button then it will direct you to memory and ask you to select image of

currency. After selecting an image you will automatically return to the window and your selected image will be visible in application window as in Fig. 9.
Recognize: This button is used for currency recognition. After pressing this button currency recognition will start and you will get output in few seconds on the application window as in Fig. 12.
Reset: This button is used to clear the application window.
Exit: This works as simple exit button, you will come out of the application and all processes of application will be terminated.


Currency Recognition System

School of Computing Science & Engineering
Developed By : Pranshu Verma, Rajan Gupta

Fig. 7 Welcome window

Input Image: It is the currency image which you want to recognize as in Fig. 8.

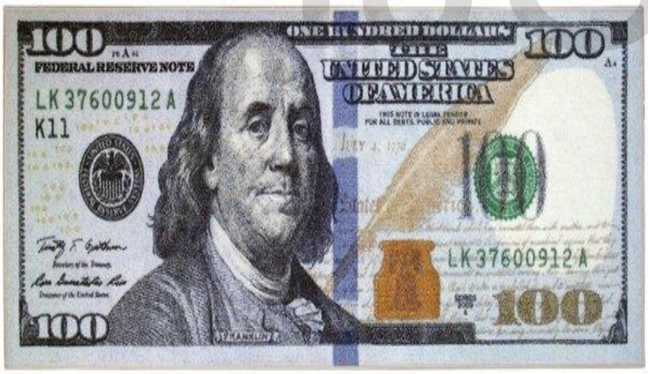
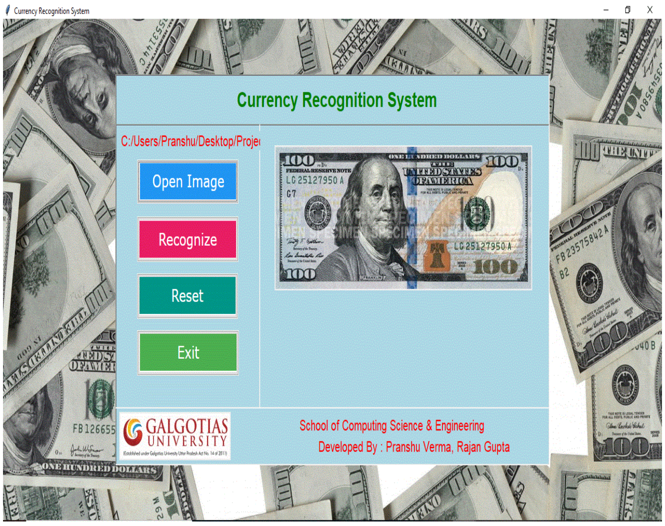


Fig. 8 Input Image

Input Screen as in Fig. 9 is application screen where your select image will appear and allow you to perform actions defined in system.


Currency Recognition System

C:/Users/Pranshu/Desktop/Proje

School of Computing Science & Engineering
Developed By : Pranshu Verma, Rajan Gupta

Fig. 9 Input Screen

During image processing, firstly it will convert inserted image from BGR to Greyscale as in Fig. 10.

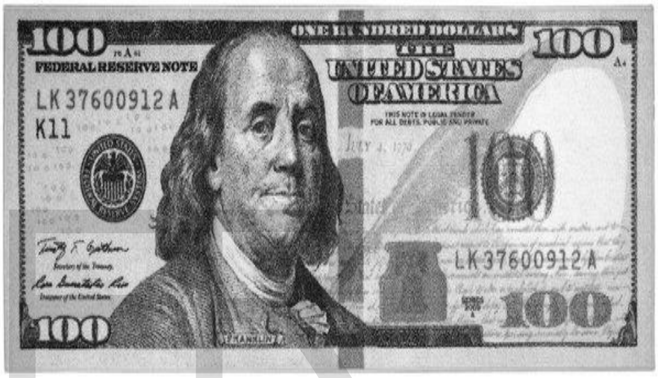


Fig. 10 Greyscale Image

After conversion from BGR to Greyscale it will perform template matching as in Fig. 11 for correctly recognition of the currency with currency name and currency value (Denomination).

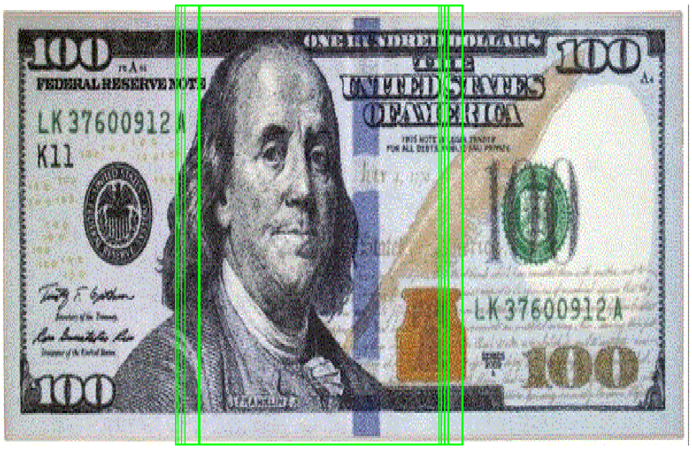
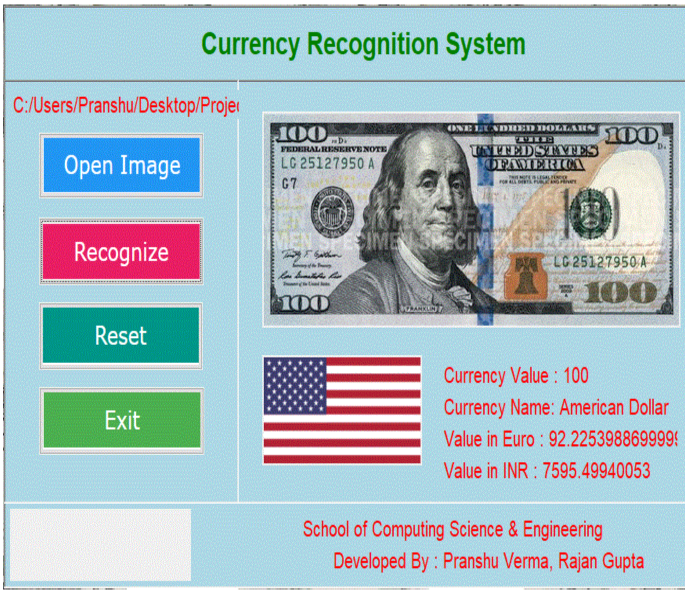


Fig. 11 Template Matching

The final output will contain Currency Value, Currency Name, Value in Euro and Value in INR/Value in USD. And output will also show the country flag.



C:/Users/Pranshu/Desktop/Proje

Currency Value : 100
Currency Name: American Dollar
Value in Euro : 92.225398869999
Value in INR : 7595.49940053

School of Computing Science & Engineering
Developed By : Pranshu Verma, Rajan Gupta

Fig. 12 Output Screen

REFERENCES

1. Pearson Book on Digital Image Processing 4th Edition By Gonzalez (2018).
2. Programming and Problem Solving with Python 1st Edition by ‘Ashok Namdev Kamthane’ and ‘Amit Ashok Kamthane’ 2017
3. Identification of Paper Currency Techniques: A Survey by S. Sahu and T. Verma International Journal of Science Technology & Engineering (2016)
4. Recognition system for real time paper currency. IEEE 2012.
5. Extraction of serial number on bank notes. IEEE.
6. Digital Image Processing And Analysis: Human And Computer Vision Applications With Cviptools, 2nd Edition by Scott E Umbaugh, T and F India.
7. An intelligent system for paper currency recognition - robust features Journal of Intelligent & Fuzzy Systems - A. Sargano, M. Sarfrazb, and N. Haq, 2014.
8. Paper Currency Recognition - Journal of Advanced Computer Science and Technology Research , ‘Althafiri E, Sarfraz M’, ‘Alfarras M. Bahraini’.
9. Study of Printing Identification Based on Multi-spectrum Imaging Analysis, Proceedings of the International Conference on Computer Science and Software Engineering, By Liu Q, Tang L. p. 229 – 232; 2008.
10. https://www.googlescholar.com
11. https://www.geeksforgeeks.org/digital-image-processing/
12. https://www.elprocus.com
13. https://www.worldometers.info/geography/
14. https://www.worldatlas.com/articles/how-many-currencies-are-in-the-world.html