International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

CURRENCY RECOGNITION SYSTEM USING IMAGE PROCESSING

M.CHANDRIKA [1], Mrs.R. Madhuri Devi [2]
[1] PG Scholars, Department of CSE, ***PRIYADARSHINI INSTITUTE OF TECHNOLOGY AND MANAGEMENT***, Guntur Dist., Andhra Pradesh, India.
[2], Associate Professor, Department of CSE, ***PRIYADARSHINI INSTITUTE OF TECHNOLOGY AND MANAGEMENT***, Guntur Dist., Andhra Pradesh, India.

ABSTRACT:

In this paper, we propose a system for automated currency recognition using image processing techniques. The proposed method can be used for recognizing both the country or origin as well as the denomination or value of a given banknote. Only paper currencies have been considered. This method works by first identifying the country of origin using certain predefined areas of interest, and then extracting the denomination value using characteristics such as size, color, or text on the note, depending on how much the notes within the same country differ. We have considered 20 of the most traded currencies, as well as their denominations. Our system is able to accurately and quickly identify test notes.

INTRODUCTION

According to the survey conducted by the CIA, there are around 180+ currencies presently circulating in the world. Each of these currencies differs greatly in features such as size, color and texture. Unlike the olden times, the trade and commerce between countries has increased in all sorts of levels. The need for acquiring knowledge about all the currencies by the banks has been extremely important. However for any human teller to recognize each note correctly is something that is not feasible. Thus the need for an efficient automated system that helps in recognizing notes is pivotal for the future. In this paper, we propose an automated system for currency recognition using Image processing techniques.As previously discussed, there have been various systems that have been proposed in various papers.

However one can learn from their results that none of the systems proposed is completely efficient and that taking into account a single parameter for this problem statement is not helpful. Hence we propose a system that takes into account various different features understanding the differences in each note. We aim to build our system in a way that it is easily scalable and gives an accurate result.

LITERATURE SURVEY

Character recognition is not a new problem but its roots can be traced back to systems before the inventions of computers. The earliest OCR systems were not computers but mechanical devices that were able to recognize characters, but very slow speed and low accuracy. In 1951, M. Sheppard invented a reading and robot GISMO that can be considered as the earliest work on modern OCR [1]. GISMO can read musical notations as well as words on a printed page one by one. However, it can only recognize 23 characters. The machine also has the capability to could copy a typewritten page. J. Rainbow, in 1954, devised a machine that can read uppercase typewritten English characters, one per minute. The early OCR systems were criticized due to errors and slow recognition speed. Hence, not much research efforts were put on the topic during 60's and 70's. The only developments were done on government agencies and large corporations like banks, newspapers and airlines etc.

Because of the complexities associated with recognition, it was felt that three should be standardized OCR fonts for easing the task of recognition for OCR. Hence, OCRA and OCRB were developed by ANSI and EMCA in 1970, that provided comparatively acceptable recognition rates[2] .

During the past thirty years, substantial research has been done on OCR. This has lead to the emergence of document image analysis (DIA), multi-lingual, handwritten and omni-font OCRs [2]. Despite these extensive research efforts, the machine's ability to reliably read text is still far below the human. Hence, current OCR research is being done on improving accuracy and speed of OCR for diverse style documents printed/ written in unconstrained environments. There has not been availability of any open source or commercial software available for complex languages like Urdu or Sindhi etc.

One of the most important steps of offline character recognition system is skew detection and correction which has to be used in scanned documents as a pre-processing stage in almost all document analysis and recognition systems. This paper describes the skew detection and correction of scanned document images written in Assamese language using the horizontal and vertical projection profile analysis [5]. Documents with background images in OCR cause an error. A non-linear transformation is used to enhance the contrast of each channel image. The experimental results show that the recognition accuracies are improved significantly after removing background images [7]. For pre-processing Fourier Transform is used which decomposes an image into sine and cosine components with increasing frequencies. Fourier transform converts spatial domain onto frequency domain which is easily used for further processing [1]. Reading text from photographs is a challenging problem. They applied recently developed machine learning algorithms for learning the features automatically from unlabeled data. They proposed text detection and recognition system based on a scalable feature learning algorithm and applied it to images of text in natural scenes [8]. Since past few years, research has been performed to develop machine printed Chinese/English characters. In this paper, they described the search and fast match techniques. High-performance Chinese/English OCR engine is used to construct a large vocabulary. They have collected 1862 text lines from varied sources such as newspapers, magazines, journals, books, etc [9]. H. Wang and J. Kangas [10] proposed a method of identifying character- like regions in order to extract and recognize characters in natural color scene images automatically. Connected component extraction is used to check the block candidates. Priority adaptive segmentation (PAS) is implemented to obtain accurate foreground pixels of the character in each block. Paper [11] presented a system for text extraction based on the open-source OCR algorithm. The system is used for functional verification of TV sets. J. Diaz-Escobar [12] proposed a new method for recognition of content-less characters in degraded images using the phase congruency and local energy model. The suggested phase features are invariant to non-uniform illumination and slight geometric distortions. Degraded images were compared with that of the SIFT method in terms of recognition metrics. Another approach in the paper [13] Hauling the scene text from image and video is challenging due to the complex background, changeable font size, dissimilar style, unknown layout, poor resolution and blurring, position, viewing angle and so on. For text extraction region and connected component based methods are used. Artificial Neural network (ANN) is used as the classifier to filter out the text and non-text component

The main purpose of Optical Character Recognition (OCR) system based on a grid infrastructure is to perform Document Image Analysis, document processing of electronic document formats converted from paper formats more effectively and efficiently. This improves the accuracy of recognizing the characters during document processing compared to various existing available character recognition methods. Here OCR technique derives the meaning of the characters, their font properties from their bit-mapped images. The primary objective is to speed up the process of character recognition in document processing. As a result the system can process huge number of documents with-in less time and hence saves the time. Since our character recognition is based on a grid infrastructure, it aims to recognize multiple heterogeneous characters that belong to different universal languages with 3 different font properties and alignments.

Features :-

Despite of the significant amount of research in OCR, recognition of characters for language such as Arabic, Sindhi and Urdu still remains an open challenge. An overview of OCR techniques for these languages has been planned as a future work. Another important area of research is multi-lingual character recognition system. Finally, the employment of OCR systems in practical applications remains an active are of research.

EXISTING SYSTEM:

Around 180+ currencies are available around the world and the need for an automated system related to currencies has been increasing exponentially recently. The need for developing systems that process notes without human intervention for various different uses has been pivotal for the development of systems that help in detecting and recognizing currency notes. However the varying features in each notes and the security aspects involved in different currencies make this task extremely difficult. Various systems have been proposed in the past that take into account different features such as aspect ratio and HSV values . Methods such as pattern matching have been proposed to develop a system that uses a single algorithm for all the currencies. However not a single method has proven to be efficient enough for actual development thereby making this problem statement an interesting area of research.Once the pre-processing steps have been done, we can identify which regions of the note are relatively empty (black pixels in the binary image). This is done based on certain predefined areas. All the currencies are clustered into groups based on which regions of the note are relatively empty. We have chosen to divide them into 3 groups – left side empty, right side empty, and center empty, although if the number of currencies were larger, we could possibly use a larger set of groups (top empty, bottom empty, etc.). Grouping is done by finding out the ratio of black to white pixels for the required region, and then classifying the note based on this ratio. The values chosen to classify the notes have been found experimentally. Note that some notes have no significant empty space, and therefore don't fall into any of the groups. These notes are classified into another group.

PROPOSED SYSTEM

In this paper, we propose an automated system for currency recognition using Image processing techniques. Our system works for 20 of the most commonly used currencies.One of the first methods proposed to identify the currency notes using image processing techniques was in the early 90's. However their algorithm does not take aspects of authentication of the notes into account. Thus it has been assumed that the notes are in good condition and images as desired are obtained. It is noteworthy to mention that the system proposed requires the input images to be taken in a predefined angle and distance. The system proposed then applies a series of pre-processing steps on the input images and then extracts certain features such as hue, saturation and value parameters in order to compute a Euclidean distance using these values and compare them with the values that are used as standards. Though this method tries to propose an overall algorithm for all the currencies, it is not an efficient method to identify the notes as certain notes across countries have similar features.The image of the banknote must first be pre-processed to remove any extraneous noise. This is done by applying a simple de-noising filter. The image is then converted to a binary image using adaptive threesholding.Once the banknote has been segregated into one of the predefined groups, we can check the image against templates for each of the countries within that group. Note that this is requires less comparisons than checking the image against all templates of every country in the system, and is the reason we have chosen to segregate the countries into such groups. The templates are chosen such that they are small (thus requiring less computation) but still uniquely identify the country of origin.

METHODLOGY

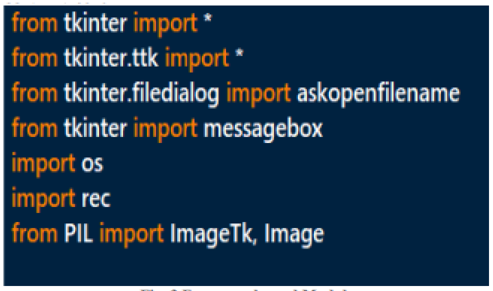
For Image processing and Template matching: 1. img = cv2.imread(filename) /*for taking input image and read that image*/ 2. gray_img=cv2.cvtColor (img,cv2.COLOR_BGR2GRAY) /*conversion of input image from BGR to Greyscale.*/ 3. temp ="0" /*initializing a variable temp with 0 value*/ 3.1. entries = os.listdir("trained_data/") /*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)]['val ue']) /*for printing currency value on output screen.*/ 6.2.8. print("Currency Name : " , currencydata[int(imageid)]['cur rency']) /*for printing currency name on output screen.*/ 6.2.9. print("Currency value In EUR:" , eur_rate*currencydata[int(imag eid)]['val ue']) /* for printing exchange rate in EURO on output screen.*/ 6.2.10. print("Currency value In USD:" , usd_rate*currencydata[int(imag eid)]['va lue']) /* for printing exchange rate in US Dollar on output screen.*/ 6.2.11. currvalue = currencydata[int(imageid)]['val ue'] /* obtaining currency value from trained dataset.*/ 6.2.12. currname=currencydata[int(ima geid)]['c urrency'] /*obtaining currency name from trained dataset.*/ 6.2.13. curreur=eur_rate*currencydata[int(imag eid)]['value'] /*For getting currency exchange rate with respect to euro. */ 6.2.14. currusd = usd_rate * currencydata[int(imageid)]['val ue'] /*printing current exchange rate with respect to USD.*/ 6.2.15. window.windows.getind(currva lue, currname, curreur, currusd,self) /* for displaying output on application window.*/ 6.3. else: 6.3.1. url="https://api.exchangeratesa pi.io/late st? 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)]['val ue'])/*for printing currency value on output screen.*/ 6.3.9. print("Currency Name : " , currencydata[int(imageid)]['cur rency']) /*for printing currency name on output screen.*/ 6.3.10. print("Currency value In EUR:" , currencydata[int(imageid)]['val ue'] * eur_rate) /* for printing exchange rate in EURO on output screen.*/ 6.3.11. print("Currency value In INR:" , currencydata[int(imageid)]['val ue'] * inr_rate) /* for printing exchange rate in INR on output screen.*/ 6.3.12. currvalue=currencydata[int(imageid)]['v alue'] /* obtaining currency value from trained dataset.*/ 6.3.13. currname=currencydata[int(imageid)]['c urrency'] /* obtaining currency name from trained dataset.*/ 6.3.14. curreur=currencydata[int(image id)]['val ue'] * eur_rate /*printing current exchange rate with respect to USD.*/ 6.3.15. currinr=currencydata[int(image id)]['val ue'] * 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:

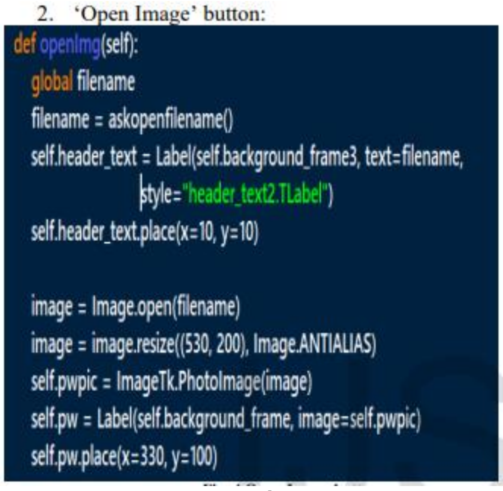
1. Imported frameworks and modules Fig. 3



```
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:

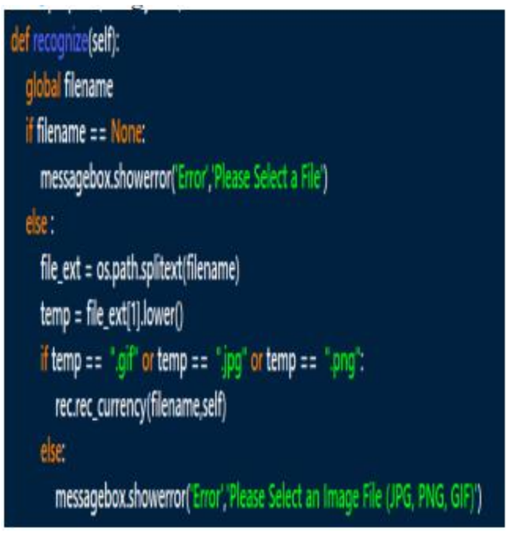


```
def openImg(self):
  global filename
  filename = askopenfilename()
  self.header_text = Label(self.background_frame3, text=filename,
            style="header_text2.TLabel")
  self.header_text.place(x=10, y=10)

  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

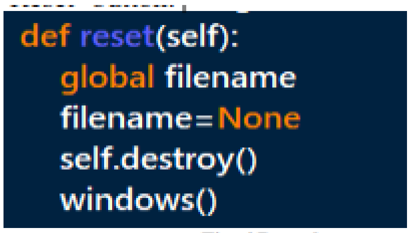
3.”Recognize” button



```
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

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.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

IntegrationTesting

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION & FUTURE SCOPE:

In conclusion, we have designed a system that accurately identifies both the country of origin and the denomination of a given banknote. Our system currently supports twenty of the most common currencies, but can easily be extended to more countries based on the method we have previously described. When compared with the crude algorithm of pixel by pixel comparison, our algorithm is considerably more accurate, and takes less time. We have thus learned that our proposed algorithm is able to identify currency and denomination in an average of 5.3 seconds, which is a considerable improvement over the crude algorithm. However, our proposed system only

Test Case 1	
Test Case Name	Empty login fields testing
Description	In the login screen if the username and password fields are empty
Output	Login fails showing an alert box asking to enter username and password.

Test Case 3	
Test Case Name	User application success
Description	User login need to provide all data.
Output	Enter valid application

considers a limited number of currencies. There are 180+ currencies that can be included in the system, and we have chosen to only do for 20 of the most common ones. Also, the system should be effective in identifying notes that are mutilated. Our system is not effective under this consideration. This can be worked on in the future.

Despite of the significant amount of research in OCR, recognition of characters for language such as Arabic, Sindhi and Urdu still remains an open challenge. An overview of OCR techniques for these languages has been planned as a future work. Another important area of research is multi-lingual character recognition system. Finally, the employment of OCR systems in practical applications remains an active are of research**.**

REFERENCES

Satti, D.A., 2013, Offline Urdu Nastaliq OCR for Printed Text using Analytical Approach. MS thesis report Quaid-i-Azam University: Islamabad, Pakistan. p. 141.

[2] Mahmoud, S.A., & Al-Badr, B., 1995, Survey and bibliography of Arabic optical text recognition. Signal processing, 41(1), 49-77.

[3] Bhansali, M., & Kumar, P, 2013, An Alternative Method for Facilitating Cheque Clearance Using Smart Phones Application. International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2(1), 211-217.