ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 5, Special Issue 6, July 2016

4th National Conference on Design Innovations for 3Cs "Compute-Communicate-Control"

Organized by

Dept. of ECE, MVJ College of Engineering, Bangalore-560067, India

Indian Currency Note Denomination Recognition and Validation

Sushma R G[1], Nandeesha K[2], Shreeharsha S[3], Srinidhi G[4], Vijay Raj Gopal R[5]

Assistant Professor, Dept. of ECE, MVJ College of Engineering, Bengaluru, India[1]

UG Student, Dept. of ECE, MVJ College of Engineering, Bengaluru, India [2],[3],[4],[5]

ABSTRACT: Counterfeit notes are one of the biggest problem occurring in cash transactions. For a country like India, it is becoming a big hurdle. Over the past few years, as results of the great technological advances in color printing, duplicating and scanning, counterfeiting problem has become more and more serious.At present the Currency denomination recognition is becoming a dynamic topic for researchers in different potential applications.It is very difficult to count different denomination notes in a bunch. This paper proposes an image processing technique to identify paper currency denomination and fake note recognition.It also includes designing a system that helps in identification of Indian currency notes to check whether it is a valid or invalid according to the RBI rules and regulations.

KEYWORDS: counterfeit notes, feature extraction, image processing, MATLAB

I. INTRODUCTION

The currency system is prevalent in India since a very long time. The Government of India introduced its first paper money issuing 10 rupee notes in 1861. It was followed by 20 rupee notes in 1864, 5 rupees in 1872, 10,000 rupees in 1899, 100 rupees in 1900, 50 rupees in 1905, 500 rupees in 1907 and 1000 rupees in 1909. In 1917, 1 and 2½ rupees notes were introduced.The Reserve Bank of India (RBI) started note generation in 1938, issuing 1000, 100, 50,20,10,5,2 rupee notes, while the Government kept on issuing 1 rupee notes. At present the currency system in India has INR 1,2,5,10,50,100,500,1000. But these are unique in one way or the other. These characteristics may be shade, size or some distinguishing proof imprints and so on.

Manual testing of all notes in transactions is very time consuming and untidy process and also there is a chance of tearing while handling notes. Therefore Automatic methods for bank note recognition are required in many applications such as automatic selling-goods and vending machines. Every year RBI (Reserve bank of India) face the problem of counterfeit currency notes or destroyed notes. Handling a large volume of counterfeit notes imposes additional problems.Reserve bank of India given useful tips on some features to detect a fake Indian rupee note such as Optical Variable Ink, Latent Image, Security Thread, Micro lettering, Watermark, etc..

The best way to identify a note is the silver bromide thread that runs vertically through a currency note. Fake currency notes tend to have silver-colored band painted in place of the silver thread. A real note has a prominent thread with raised 'RBI' markings made on it in English and Hindi. Also, in a real note, the color of the thread shifts from green to blue when viewed from different angles.

The identification of objects in an image is called recognition. This process would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures. A camera with a minimum resolution of 640X480 is used to capture the images of the Indian currency notes. These images are then given as an input to the feature extraction code implemented in MATLAB and the currency note is featured in the image.

II.LITERATURE SURVEY

1. Indian Currency Note Denomination and Recognition in Color images.

This was proposed by Hanish Agarwal and Padam Kumar.The method of their implementation is based on Localization and Recognition using color matching.The difficulties faced during processing is that the color of the note may be degraded with constant usage and if anew currency note is released then the algorithm has to be updated accordingly which is not feasible.

2. **Image Processing Based Feature Extraction of Indian Currency Notes**

This was proposed by Krishnan G.The method of implementation is based on Thresholding, Morphological filtering and Word segmentation.The limitations of this technique is two words are sometimes joined together and it is difficult to distinguish between the characters. So Word segmentation becomes a major task.

3. **Recognition and classification of currency notes using discrete wavelet transforms**

This method was proposed by SS Sannakki and Pallavi J Gunjale.The method used was PNN (Probabilistic Neural Networks) technique for classification. As this model required more memory for image storage and processing it was not feasible. It also offered less accuracy and was slow.

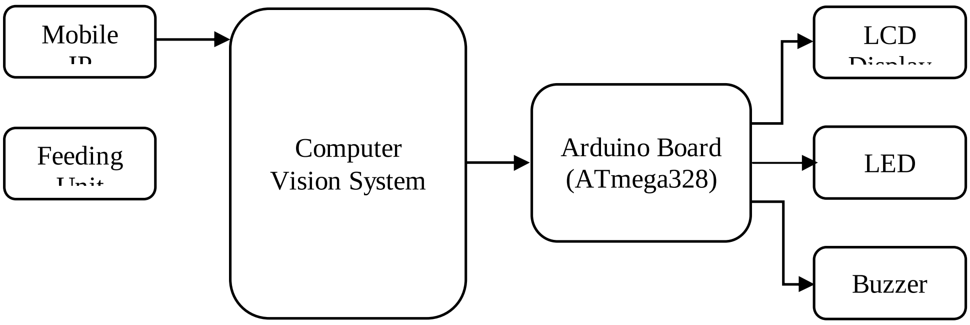
4. **An automated recognition of fake or destroyed Indian currency notes**

This was proposed by Sanjana, Manoj Diwakar and Anand Sharma.The method used based on SVM (Support Vector machines), ANN (Artificial Neural Networks) and heuristicanalysis.The major limitation caused was complexity in training the networks and recognizing characters.

5. **Image processing based heuristic analysis for enhanced currency recognition**

This was proposed by Parminder Singh Reel, Gopal Krishnan and Smriti Kotwal.The method is Heuristic analysis and Serial number recognition.As with the previous methods the problems were based on the color degradation with usage and inefficient character recognition.

III. PROPOSED SYSTEM



IV.METHODOLOGY

1. Image acquisition: It is the action of retrieving an image from the source, usually a hardware based source. A high resolution image is captured from a special mobile application called IP Webcam. IP Webcam: It is a special type of mobile application which is used to acquire an image. Here, desired resolution can be set capture an image. In order to aquire an image through this application, the mobile IP camera and the PC should share the same network. On starting the video stream, mobile is assigned a local IP address using which the PC can link to the mobile camera. The communication is achieved between the mobile and MATLAB. A sample code to acquire a single frame is given below,

```
url = 'http://192.168.1.86:8080/shot.jpg';
img = imread(url);
```

The advantage of using this mobile application is that we can capture high resolution images without using any additional light sources.

2. Image pre-processing: Here various image processing techniques such as filtering, histogram equalization, and image conversions are applied on the captured image.

- Conversions: Two types of conversions are used in this system are RGB to grayscale and RGB to YC_bC_r.. In RGB colour model, each colour appears in its primary spectral components of Red, Green and Blue. The colour of a pixel is made up of three components; Red, Green and Blue (RGB), described by there corresponding intensities. Colour components are also known as color channels or planes (components). In Indian currency notes, metal strip is not visible in RGB colour model so we are converting RGB to YC_bC_r. Y′ is the luma component and C_b and C_r are the blue-difference and red-difference chroma components. Y′ (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.

- Median filtering: The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later process. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighbouring entries. The pattern of neighbours is called the 'window', which slides, entry by entry, over the entire signal.

- Histogram equalization: This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. This method is useful in images with backgrounds and foregrounds that are both bright or both dark.

3. Segmentation: It is the method of extracting the region of interest from the currency note. Here, the intensity values that resides within our region of interest in both row wise and column wise are identified using a specialized MATLAB command known as imtool. Based on those intensity values we will segment the image.

4. Feature extraction: This is nothing but isolating the desired characters from the currency note. After this Thresholding is performed.

5. Thresholding: The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity $I_{i,i}$ is less than some fixed constant T (that is, $I_{i,j} < T$), or a white pixel if the image intensity is greater than that constant T.
After thresholding, there is a chance of getting additional objects along with our extracted feature, for this we are going to label each and every object. After this, the area of each object is calculated and a constant threshold is set based on those array of areas. Then using specialized MATLAB command known as 'bwareaopen' the additional components are removed.

V. SOME IMPORTANT MATLAB FUNCTIONS USED

1) bwlabel - Label connected components in 2-D binary image.
Usage: [L num] = bwlabel(BW);
The above command returns a label matrix L that contains labels for the 8-connected objects found in BW. The label matrix, L is of the same size as BW. Also returns num, the number of connected objects found in BW.

2) nnz – number of nonzero matrix elements (can be considered as the area of an object).
Usage: AREA(i) = nnz(L==i);
The above command returns the area of ith labelled image by counting the number of non-zero pixels in the image. This command is used in a for loop to calculate the area of a set of labeled images. AREA(i) will contain the number of non-zero pixels in ith image.

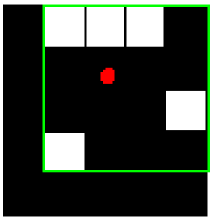
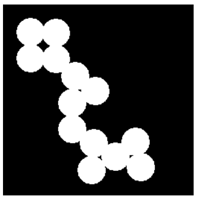
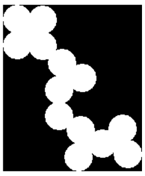
3) bwareaopen - remove small objects from binary image.
Usage: BW2 = bwareaopen(BW,P);
The above command removes all connected components (objects) that have fewer than P pixels from the binary image BW, producing another binary image, BW2. This operation is known as an area opening.
4)
5) bwboundaries – trace region boundaries in binary image.
Usage: [B L] = bwboundaries(BW,'noholes');
The above command traces the exterior boundaries of objects excluding boundaries of holes inside those objects, in the

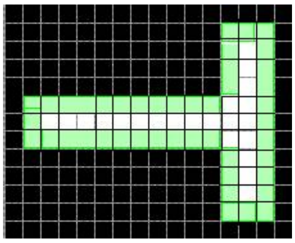
binary image BW. Also returns B, a cell array of boundary pixel locations along with a label matrix L where objects are labeled.

6) rectangle – create rectangle with sharp or curved corners.

Usage: rectangle('Position', [BB(1),BB(2),BB(3)],'ÉdgeColor','r','LineWidth',2);
The above command creates a rectangle in 2-D coordinates. Position/space and dimensions for plotting rectangle are determined by the argument BB(i) which contains a four element vector of the form [x y w h] in data units for ith image/object. The x and y elements determine the location and the w and h elements determine the size.
regionprops – measure properties of image regions.

Usage: stats = regionprops(BW,'all');
The above command returns the measurement or co-ordinates of all the properties for each connected component (object) in the binary image, BW. stats will be loaded with a struct array containing a struct for each object in the image.

Property Name	**Description**
Area	Returns a scalar that specifies the actual number of pixels in the region.
BoundingBox	Returns smallest rectangle containing the region, specified as a 1-by-Q*2 vector, where Q is the number of image dimensions. Usually returns a four element vector of the form [x y w h] for each object in case of a 2D image. The x and y are the co-ordinates whereas w and h are the width and height of the bounding box respectively.
Centroid	Returns a 1-by-*Q* vector that specifies the center of mass of the region. The first element of Centroid is the horizontal coordinate (or *x*-coordinate) of the center of mass, and the second element is the vertical coordinate (or *y*-coordinate). All other elements of Centroid are in order of dimension. This figure illustrates the Centroid and bounding box for a non-contiguous region. The region consists of the white pixels; the green box is the bounding box, and the red dot is the centroid. 
Eccentricity	Returns a scalar that specifies the eccentricity of the ellipse that has the same second-moments as the region. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. The value is between 0 and 1. (0 and 1 are degenerate cases. An ellipse whose eccentricity is 0 is actually a circle, while an ellipse whose eccentricity is 1 is a line segment.)
Extent	Returns a scalar that specifies the ratio of pixels in the region to pixels in the total bounding box. Computed as the Area divided by the area of the bounding box.
FilledArea	Returns a scalar that specifies the number of on pixels in FilledImage.
FilledImage	Returns a binary image (logical) of the same size as the bounding box of the region. The on pixels correspond to the region, with all holes filled in, as shown in this figure.  Original Image, Containing a Single Region  Image Returned
Image	Returns a binary image (logical) of the same size as the bounding box of the region. The on pixels correspond to the region, and all other pixels are off.

Property Name	Description
Perimeter	Returns a scalar that specifies the distance around the boundary of the region. *regionprops* computes the perimeter by calculating the distance between each adjoining pair of pixels around the border of the region. If the image contains non-contiguous regions, *regionprops* returns unexpected results. This figure illustrates the pixels included in the perimeter calculation for this object. 

VI. RESULTS

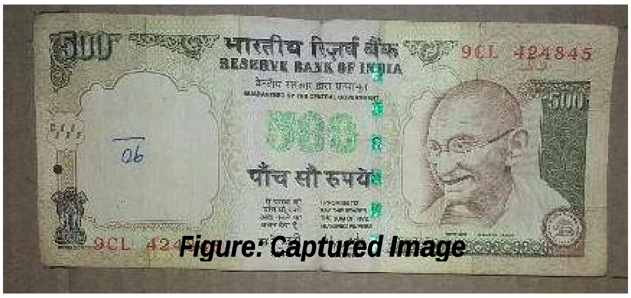
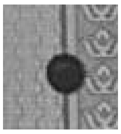


Figure: Captured Image



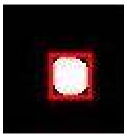
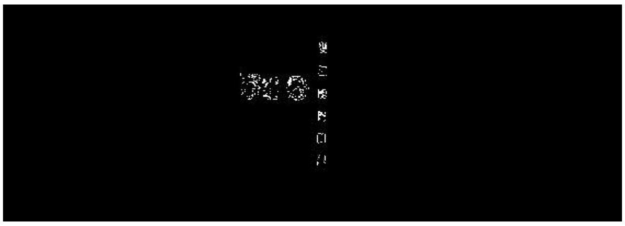
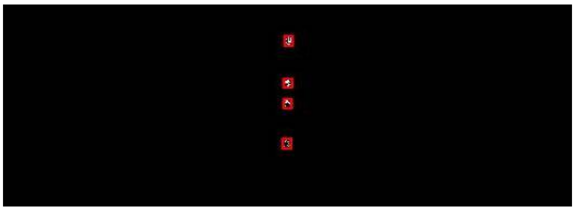


Figure: Highlight Identification Mark

Figure: Cropped Image





VII. CONCLUSION

When compared to all the above processes our proposal is based on very simple techniques which uses very less algorithms which are efficient in identifying counterfeit currency notes. As well as the ideas listed before were inefficient in identifying the denomination or the originality.Our proposal stands very different compared to all these by integrating the parameters like denomination, originality and validation as well. Our system works on the assumption that there is only one currency note placed in the field of view taken by the camera. This can be implemented in large scale where a bunch on notes of different denominations are placed.

REFERENCES

1. Hanish Agarwal, Padam Kumar, "Indian currency note denomination recognition in color images", IJACEC, Vol. 1, ISSN 2278–5140.
2. Krishan, G.2010.Image Processing Based Feature Extraction of Indian Currency Notes. PhD diss., Thapar University
3. Sannakki S. S, and Gunjale, P. J 2014, "Recognition and Classification of Currency Notes using
4. Discrete Wavelet Transform", International Journal of Emerging Technology and Advanced Engineering, Vol.4, Iss.
5. Sanjana, Manoj Diwakar, Anand Sharma, "An Automated recognition of Fake or Destroyed Indian currency notes in Machine vision", IJCSMS, Vol. 12, April 2012.