



Identification of currency denomination using image processing

Gokul Ramasamy

gokulramkavi@gmail.com

Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu

Sakthi Subramanian

sakthisuburamaniyan@gmail.com

Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu

ABSTRACT

This paper aims to solve the problem of detecting a paper currency and output the result as speech. The developed model accepts an image from a webcam (RasPi Camera). The Raspberry Pi's image processing algorithm then extracts certain features from this image and matches them against a set of training data. When an appropriate match of the acceptable threshold is found, the device outputs the denomination value through speech synthesis. Moreover, this paper also compares two methods of image processing namely, Brute Force Matcher Algorithm and Convolution Neural Network Algorithm. Finally, the pros and cons of both the methods are evaluated and the best algorithm is ascertained.

Keywords— Image processing, Brute force matcher, CNN algorithm, Raspberry Pi, Visually challenged, Currency denomination

1. INTRODUCTION

The assistive technology is one of the most basic and important system that helps a person with a disability to work around his challenges. This project presents progressive efforts for developing an assistive technology for visually impaired so that they can lead their life independently both socially and financially.

1.1 Problem statement

Post demonetization, the sizes of the Indian currency notes have drastically changed. For example, the new Rs.100 and Rs.200 notes similar physical dimensions. Though the colour of such notes is very contrasting, this difference is beneficial only to those blessed with eyesight. The population of the visually challenged in India is a staggering 36 million. These people have a hard time identifying these new notes (even the Braille and small dots and holes on these notes seem to fade away with prolonged usage). This project aims to relieve some of their problems using assistive technology.

1.2 Objective

Development of Real Time Paper Currency Identification and Audio Output System using open source hardware(Raspberry Pi) and software(Python), for the visually challenged, thus allowing them to autonomously deal with new Indian banknotes, particularly while accepting their money back during their day to day activities.

2. LITERATURE SURVEY

The need for systems that processes currency without human intervention has become inevitable for various applications. 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 as mentioned in papers [1] [2] [3].

An approach to the Portable camera based assistive text and product label reading from hand-held objects for blind persons is performed in [4]. Another approach to the efficient banknote recognition based on selection of discriminative regions with one-dimensional visible-light line sensor is mentioned in [5]. In another approach [6], Currency recognition using a smartphone and Comparison between colour-SIFT and grayscale SIFT algorithms.

As previously discussed, there have been various systems with image processing that have been proposed in various papers [7] [8] [9]. 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. In the papers [10] [11] the classification model is created using artificial neural network. Our approach is based on [3] but the differentiation of currency starts with first confirming that it is original Indian currency and then identifying the denomination. First various features are extracted as specified in the above papers and finally machine learning classifier algorithms are employed with these obtained features.

3. THEORETICAL BACKGROUND

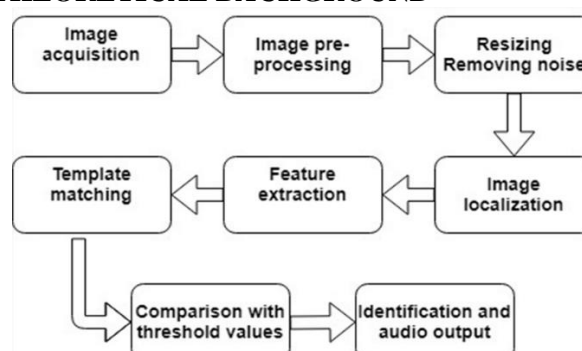


Fig. 1: Basics image processing

3.1 Basics of image processing

The figure 1 details the flow of any basic image processing algorithm. These basic steps are used in both Brute Force Matcher as well as Convolution Neural Network algorithm. However, the CNN algorithm further does some classification which makes it suitable for applications where a very large size of training set is required.

3.2 Feature extraction

The SIFT and SURF key point detector and descriptor, although comparatively old, have proven remarkably successful in a number of applications using visual features, including object recognition, image stitching, visual mapping, etc. However, it imposes a large computational burden, especially for real-time systems. A computationally efficient replacement to SIFT and SURF that has similar matching performance, less affected by image noise and capable of being used for real-time performance is called ORB.

ORB is a good alternative to SIFT and SURF in computation cost and matching performance. ORB is basically a fusion of FAST key point detector and BRIEF descriptor with many modifications to enhance the performance. First it uses FAST to find key points then applies Harris corner measure to find top N points among them. It also uses pyramids to produce multiscale-features.

3.3 FAST key points

Feature Detection using FAST - Features from Accelerated Segment Test algorithm

- Select a pixel P in the image which is to be identified as an interest point or not. Let its intensity be I_p .
- Select appropriate threshold value t .
- Consider a circle of 16 pixels around the pixel under test.
- Now the pixel P is a corner if there exists a set of n contiguous pixels in the circle (of 16 pixels) which are all brighter than I_p+t , or all darker than I_p-t . (Shown as white dash lines in the below image). n was chosen to be 12.

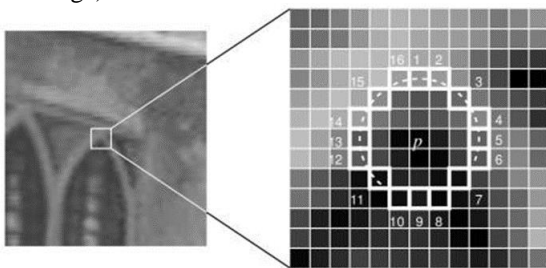


Fig. 2: Features from Accelerated Segment Test algorithm

FAST does not compute the orientation. To account for this drawback, a new modification to the existing algorithm was made. It computes the intensity weighted centroid of the patch with located corner at centre. The direction of the vector from this corner point to centroid gives the orientation.

3.4 BRIEF descriptors

BRIEF reduces the memory consumption by providing a shortcut to find the binary strings directly without finding descriptors. It takes smoothed image patch and selects a set of nd (x, y) location pairs in a unique way. Then some pixel intensity comparisons are done on these location pairs. For e.g., let first location pairs be p and q . If

$$I(p) < I(q)$$

Then its result is 1, else it is 0. This is applied for all the nd location pairs to get nd -dimensional bit string. Since BRIEF is

a feature descriptor, it does not provide any method to find the features. So some other feature detectors like SIFT, SURF, ORB etc. have to be used.

4. IMAGE CLASSIFICATION ALGORITHMS USED IN THE PROJECT

4.1 Brute Force Classification

The descriptor of one feature in first set is matched with all other features in second set using a distance calculation and the closest one is returned as the most matched one. For any two images it calculates the hamming distance using the descriptors and returns the point with minimum hamming distance. The following key point/descriptor mapping was obtained.



Fig. 4: Brute Force Classification

4.2 Convolution neural network

In neural networks, Convolutional Neural Network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications, objects detection, face recognition etc. CNN image classifications takes an input image, process it and classify it under certain categories. Computer sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see $h \times w \times x \times d$ (h = Height, w = Width, d = Dimension). Technically, deep learning CNN models to train and test each input image will pass it through a series of convolution layers with filters, pooling fully connected layers and applies activation function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

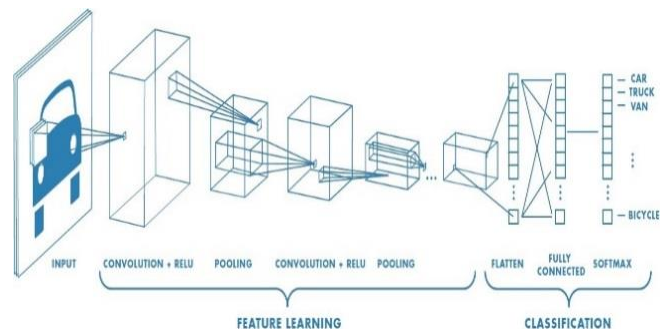


Fig. 5: Convolution neural network

4.2.1 Convolution layer: Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

4.2.2 Padding: To fit the input image, two options are available as follows:

- Pad the picture with zeros (zero-padding)
- Drop the part of the image where the filter did not fit.

This is called valid padding which keeps only valid part of the image.

4.2.3 Non Linearity (ReLU)

- ReLU stands for Rectified Linear Unit for a non-linear operation. The output is

$$f(x) = \max(0,x)$$

- ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values
- There are other non-linear functions such as tanh or sigmoid can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than other two.

4.2.4 Pooling layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called sub sampling or down sampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types. In this project, Maxpooling is used which takes the largest element from the rectified feature map.

4.2.5 Fully connected layer: The developed matrix was flattened into vector and fed into a fully connected layer like neural network. This model created was used to classify the image.

5. METHODOLGY

The system is divided into two parts. The first part is to identify the currency denomination through image processing. The second part is the oral output to notify the visually impaired person about the denomination of the note that he/she is currently having.

The development of this device is based on a webcam integrated with Raspberry Pi microcontroller and a speaker for sound output. The real time bank notes are captured and processed through different image processing techniques like edge detection, segmentation, and feature extraction and classification.

Here Raspberry Pi is used as a processor which processes the image of the currency note captured by the web camera. The controlling code for web camera is written and stored in processor. Captured image is stored in memory. Now Raspberry Pi will process the image to identify the denomination of the currency. The processing algorithms and codes are written in PYTHON OpenCV. The reason for selection of the said hardware and software is that, this paper intends to make this product as a cost efficient model using open source hardware such as Raspberry Pi, so that it may favour future advanced improvements from people all over the world thus benefitting the end user

6. BLOCK DIAGRAM

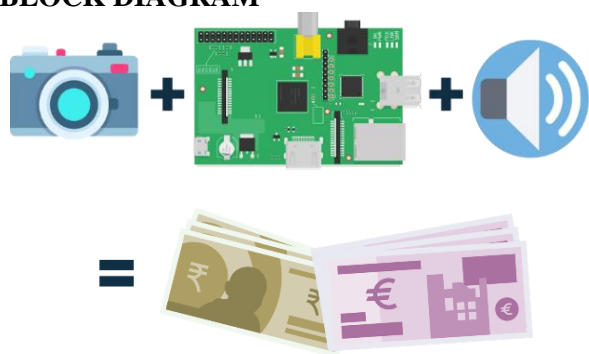


Fig. 3: Block diagram

7. TIME SCHEDULE

Table 1: Time schedule

Week	Work Description
1	Data Collection for development of the image processing code.
2	Development of the image processing code using Brute Force Matcher algorithm in Python.
3	Development of the image processing code using CNN algorithm in Python.
4	Development of Python Code for interaction with the webcam and external speaker.
5	Integrating Raspberry Pi circuit with the Webcam-Speaker circuit and running the system as a whole.
6	Integrating the power supply unit and testing the whole circuit running independently.
7	Testing and debugging (if any) of the project.

8. HARDWARE IMPLEMENTATION

8.1 Specification

Table 2: Hardware Specification

S. no	Component Required	Quantity	Amount	Justify the requirement
1.	Webcam	1	Rs.449	To scan the note for its denomination.
2.	Raspberry Pi	1	Rs.2800	To do all the image processing activities.
3.	16X2 LCD Display	1	Rs.154	To display the status of each stage (that is. scanning, processing, connecting etc.)
4.	Speaker	1	Rs.200	To give the oral output of the currency note's originality and denomination.
5.	Computer with Matlab Simulink and Python software	1		Used for developing the image processing python code for Raspberry Pi.
6.	Power Supply Board	1	Rs.50	For powering up the external circuits like Raspberry Pi, Speaker and the Webcam.

9. A. PYTHON OpenCV CODE: USING BRUTE FORCE MATCHER ALGORITHM (COMPLETE WORKING CODE)

```
#Calling the necessary library
from preprocess import *
from matplotlib import pyplot as plt
from picamera import PiCamera
from time import sleep
#Importing the necessary library functions
import subprocess
import pyttsx3
#Image acquisition using RasPi camera
camera = PiCamera()
camera.start_preview()
sleep(5)
camera.capture('test.jpg')
camera.stop_preview()

max_val = 8
max_pt = -1
max_kp = 0
```



```
orb = cv2.ORB_create()
#Importing the captured image into this program
test_img = read_img('files/test.jpg')
original = resize_img(test_img, 0.4)
display('original', original)
(kp1, des1) = orb.detectAndCompute(test_img, None)
#Declaring the training set
training_set=['files/Fiftyrupees.jpg',
'files/Fiftyrupees2.jpg',
'files/Fiftyrupeesnew.jpg',
'files/Fiftyrupeesnew2.jpg',
'files/Hundredrupees.jpg',
'files/Hundredrupeesnew.jpg',
'files/Hundredrupeesnew2.jpg',
'files/Hundredrupees2.jpg',
'files/TenRupees.jpg',
'files/TenRupees2.jpg'...]
for i in range(0, len(training_set)):
    # train image
    train_img = cv2.imread(training_set[i])
    (kp2, des2) = orb.detectAndCompute(train_img, None)
# brute force matcher
bf = cv2.BFMatcher()
all_matches = bf.knnMatch(des1, des2, k=2)
good = []
# if good then append to list of good matches
for (m, n) in all_matches:
    if m.distance < 0.789 * n.distance:
        good.append([m])
if len(good) > max_val:
    max_val = len(good)
    max_pt = i
    max_kp = kp2

    print(i, ' ', training_set[i], ' ', len(good))
if max_val != 8:
    print(training_set[max_pt])
    print('good matches ', max_val)
    train_img = cv2.imread(training_set[max_pt])
    img3 = cv2.drawMatchesKnn(test_img, kp1,
train_img, max_kp, good, 4)

    note = str(training_set[max_pt])[6:-4]
    print("\nDetected denomination: Rs. ", note)
#Speech synthesis
engine = pyttsx3.init()
engine.say("Detected denomination is ")
engine.say(note)
engine.setProperty('volume', 0.9)
engine.runAndWait()
else:
    print("No Matches")
#Speech synthesis
engine = pyttsx3.init()
engine.say("No match found.Try again.")
engine.setProperty('volume', 0.9)
engine.runAndWait()
```

10. B. PYTHON OpenCV CODE: USING CNN ALGORITHM (CODE SNIPPET)

```
import cv2
import numpy as np
from random import shuffle
import os
from tqdm import tqdm
MODEL_NAME='currency_
```

```
recognition.model'.format
(LR,'6conv-basic-video')
def label_img(img):
    word_label=img.split('.')[0]
    if word_label=='20':return [1,0,0,0]
    elif word_label=='50':return [0,1,0,0]
    elif word_label=='100':return [0,0,1,0]
    elif word_label=='200':return [0,0,0,1]
    elif word_label=='200':return [0,0,0,0,1,0,0]
    elif word_label=='500':return [0,0,0,0,0,1,0]
    elif word_label=='2000':return [0,0,0,0,0,0,1]
def create_train_data():
    training_data=[]
    for img in tqdm(os.listdir(TRAIN_DIR)):
        label=label_img(img)
        path=os.path.join(TRAIN_DIR,img)
        img=cv2.resize(cv2.imread
(path,cv2.IMREAD_GRAYSCALE),
(IMG_SIZE,IMG_SIZE))
        training_data.append([np.array(img)
,np.array(label)])
    shuffle(training_data)
    print(training_data)
    np.save('train_data.npy',training_data)
    return training_data
def process_test_data():
    testing_data=[]
    for img in tqdm(os.listdir(TEST_DIR)):
        path=os.path.join(TEST_DIR,img)
        img_num=img.split('.')[0]
        img=cv2.resize(cv2.imread(path,cv2.
convnet = fully_connected(convnet, 4, activation='softmax')
convnet = regression(convnet
IMREAD_GRAYSCALE),(IMG_SIZE, IMG_SIZE))
        testing_data.append([np.array(img), img_num])
        np.save('test_data.npy',testing_data)
        return testing_data
train_data=create_train_data
import tflearn
import tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout,
fully_connected
from tflearn.layers.estimator
import regression
convnet = input_data(shape= [None,IMG_SIZE, IMG_SIZE, 1],
name='input')
convnet = conv_2d(convnet, 32, 2, activation='relu')
convnet = max_pool_2d(convnet, 2)
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)
elif np.argmax(model_out)==3: str_label='200' elif
np.argmax(model_out)==4: str_label='200'
optimizer='adam', learning_rate=LR,
loss='categorical_crossentropy', name='targets')
model = tflearn.DNN(convnet,
tensorboard_dir='log')
if os.path.exists('{ }.meta'.format
(MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded')
train=train_data
test=train_data
X=np.array([i[0] for i in train]).reshape(-
1,IMG_SIZE,IMG_SIZE,1)
```

11. HARDWARE SETUP



Fig. 4: Hardware setup

12. RESULTS OBTAINED

Two methods of image processing have been carried out and following results were obtained:

Average time for processing,

- Using Brute Force matcher-9 seconds
- Using CNN algorithm-5.4 seconds

CNN algorithm is considerably faster than Brute Force matcher.

13. CONCLUSION AND SCOPE OF FUTURE WORK

The developed model detects the denomination of Indian currency with the help of basic image processing algorithm. CNN algorithm is more efficient than the Brute Force Matcher in terms of time and resources used.

The developed product is a Multi-Purpose Module and can be implemented in Real Time Applications such as automating vending machines, automatic ticket counters etc. This module can further be developed into identifying Counterfeit Currency also. This requires an addition of few more pre-processing techniques and a slight modification to the existing hardware to incorporate UV light.

Moreover the current system has a poor accuracy for identifying coins due to the reflective nature of the material. A suitable lighting source like UV can be used to prevent this problem in the future.

14. REFERENCES

- [1] Binod Prasad Yadav. "Indian Currency Recognition and Verification System Using Image Processing". In: International Journal of Advanced Research in Computer Science and Software Engineering 4.12 (2014).
- [2] H. Hassanpour, A. Yaseri, G. Ardeshiri, "Feature Extraction for Paper Currency Recognition" 2007 IEEE conference.
- [3] Vishnu R, Bini Omman, "Principal Features for Indian Currency Recognition" 2014 Annual IEEE India Conference (INDICON).
- [4] Yi C, Tian Y, Arditi A. Portable camera based assistive text and product label reading from hand-held objects for blind persons. IEEE/ASME Transaction on Mechatronics. 2014 Jun; 19(3):808–17.
- [5] Pham T.D., Park Y.H., Kwon S.Y., Park K.R., Jeong D.S., Yoon S. Efficient banknote recognition based on selection of discriminative regions with one-dimensional visible-light line sensor. Sensors. 2016;16:328 doi: 10.3390/s16030328
- [6] Doush I.A., Al-Btoush S. Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms. J. King Saud Univ. Comput. Inform. Sci. 2016 doi: 10.1016/j.jksuci.2016.06.003.
- [7] Vipin Kumar Jain and Ritu Vijay. "Indian currency denomination identification using image processing technique". In: (2013).
- [8] Smarti Kotwal. "Image processing based heuristic analysis for enhanced currency recognition". In: International Journal of Advancements in Technology 2.1 (2011), pp. 82–89.
- [9] Muhammad Sarfraz. "An Intelligent Paper Currency Recognition System". In: Procedia Computer Science 65 (2015), pp. 538–545.
- [10] Mriganka Gogoi, Syed Ejaz Ali, Subra Mukherjee, "Automatic Indian Currency Denomination Recognition System based on Artificial Neural Network", 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN).
- [11] Sahana Murthy, Jayanta Kurumathur, B Roja Reddy, "Design and Implementation of Paper Currency with Counterfeit Detection", 2016 Online International Conference on Green Engineering and Technologies (IC-GET).
- [12] "Design and Development of real time paper currency recognition system" Anilkumar B Department of Electronics and Communications Engineering GMR Institute of Technology, Rajam, Andhra Pradesh, India KRJ Srikanth Department of Electronics and Communications Engineering MITS, Rayagada, odissa, India
- [13] Identification of Counterfeit currency and denomination using Raspberry pi Swami Gururaj M.1, Naveen J.2 Student, Department of EEE, NMAMIT, Nitte, Udupi, India Assistant Professor, Department of EEE, NMAMIT, Nitte, Udupi, India
- [14] Snigdha Kamal, Simarpreet Singh Chawla, Nidhi Goel, and Balasubramanian Raman, Feature Extraction and Identification of Indian Currency Notes Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2015.
- [15] Feature point extraction algorithm from <http://in.mathworks.com/matlabcentral/fileexcha>. Template Matching is at <http://in.mathworks.com/help/vision/ref/matchfeatures.html>

BIOGRAPHIES



Gokul Ramasamy

Student

Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India

Sakthi Subramanian

Student

Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India