Currency Detection using ORB Feature

Kashish Miglani*,Priyanshu Sharma[†],Utsav Rai[‡] ,Dr. Graceline J.[§]
School of Computing Sciences and Engineering,
VIT Chennai, Tamilnadu, India 600127

kashish.miglani2015@vit.ac.in, priyanshu.sharma2015@vit.ac.in[†],utsav.rai2015@vit.ac.in[‡]*
graceline.jasmine@vit.ac.in[§]

Abstract—**With the advancement in the Color Image Processing Technology in a past two decades, there has been a huge increase in the fake currency printing in our economic system. Today, with the help of such technology, people can even print the currency in their own house also. Detecting the fake currency is always a big problem for the existing technology. Currency detection system using the ORB feature can solve this problem of detecting the fake currency just by making the use of Fast key point Descriptor and Brief Descriptor. ORB feature is a better alternative for the feature such as SURF, SIFT. This system is successful in detecting the currency using the ORB Feature. This system is having the precision about 70%.**

Keywords - ORB Feature, Image Segmentation, KNN Clustering, SURF Feature, SIFT Feature.

1. Introduction

This system is going to detect the currency. It is going to make the use of ORB feature which is a better alternative to the SIFT Feature. SIFT Feature works on the local feature extracted from the images. This feature is mostly used in the applications related to object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving. SIFT Feature needs a lot of computation power as compare to the ORB Features.

ORB Feature make the use of FAST Descriptor and the Brief Descriptor. FAST in FAST Descriptor stands for Features from accelerated segment test. This is basically a corner detection method, which could be used to extract feature points and later used to track and map objects in many computer vision tasks.The most promising advantage of the FAST corner detector is its computational efficiency. Referring to its name, it is fast and indeed it is faster than many other well-known feature extraction methods, such as difference of Gaussian (DoG) used by the SIFT, SUSAN and Harris detectors. Moreover, when machine learning techniques are applied, superior performance in terms of computation time and resources can be realized. The FAST corner detector is very suitable for real-time video processing application because of this high-speed performance.

BRIEF Descriptor stands for Binary Robust Independent Elementary Features. SIFT uses 128-dim vector for descriptors. Since it is using floating point numbers, it takes basically 512 bytes. Similarly SURF also takes minimum of 256 bytes (for 64-dim). Creating such a vector for thousands of features takes a lot of memory which are not feasible for resource-constraint applications especially for embedded systems. Larger the memory, longer the time it takes for matching.

But all these dimensions may not be needed for actual matching. We can compress it using several methods like PCA, LDA etc. Even other methods like hashing using LSH (Locality Sensitive Hashing) is used to convert these SIFT descriptors in floating point numbers to binary strings. These binary strings are used to match features using Hamming distance. This provides better speed-up because finding hamming distance is just applying XOR and bit count, which are very fast in modern CPUs with SSE instructions. But here, we need to find the descriptors first, then only we can apply hashing, which doesnt solve our initial problem on memory.

BRIEF comes into picture at this moment. It provides a shortcut to find the binary strings directly without finding descriptors. It take input as smoothened image patch. One important point is that BRIEF is a feature descriptor, it doesnt provide any method to find the features. So you will have to use any other feature detectors like SIFT, SURF etc. The paper recommends to use CenSurE which is a fast detector and BRIEF works even slightly better for CenSurE points than for SURF points. In short, BRIEF is a faster method feature descriptor calculation and matching. It also provides high recognition rate unless there is large in-plane rotation.

2. Related Works

People are doing a lot of work in this area using different different feature such as SURF Feature, SIFT Feature, Local Binary Pattern, etc. SURF Feature is also one of the most popular method for extracting the key point feature. SURF Feature make the use of approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a precomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. These can also be computed with the aid of the integral image.

The work of Sahana Murthy [2] proposed a system that can classify and subsequently verify Indian paper currency using fundamental image processing techniques. It uses the comparison between the input banknote and the calculated reference values for different parameters of original banknotes in a similar environment. This system maintains its simplicity while still having high accuracy of 100% for classification and 90% for validity verification.

The work of Mohammad H Alshayeji [3] proposed a system works on the basis of Bit-Plane Slicing Technique. The proposed technique consists of decomposing original images of 256 gray levels into their equivalent 8 binary images. This is useful in analyzing the relative importance contributed by each bit of the original image. Higher order bit levels are evaluated for gray-scale banknote images with the application of Canny edge detection algorithm. The results are then compared with genuine banknotes and with other existing techniques used for detecting counterfeit notes. Unlike existing research, it was observed that the edges obtained using bit-plane sliced images are more accurate and can be detected faster than obtaining them from the original image without being sliced. The detection of counterfeit currency was also achieved by following the process of using Canny edge detection, image segmentation, and feature extraction.

3. Proposed Work

3.1. Feature Extraction using ORB Feature

In this paper, proposed system make the extensive use of the color image processing and feature extraction techniques in order to detect the currency. This proposed system make the use of the ORB feature inspite of making the use of the features such as SURF, SIFT, Local Binary Patterns, etc. For this system, collection of the images is also a big problem. More number of images means more number of the feature extracted from the each image in the database. Using of features such as SURF or SIFT feature with such a large number of the images is big problem computation wise. That is why, our proposed system is going to make the use of ORB Feature which is very light and can be easily computed with such a large number of images.

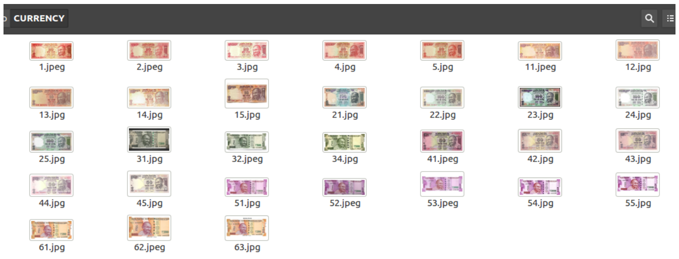
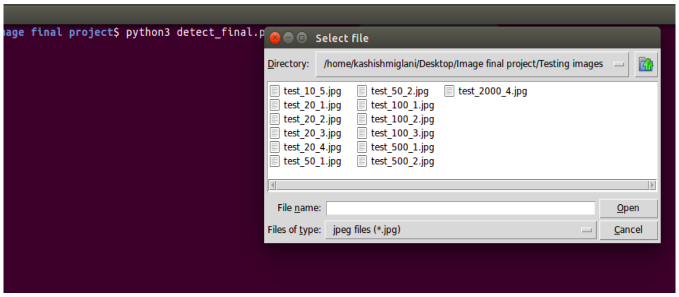


Figure 1. Image Database

In our proposed system, firstly we are going to make the use of the images that are presented in our image database for image segmentation part. For selecting a particular image from the image database, we are going to make use of the Tkinter library of python for building a proper GUI. This GUI is user friendly in nature and help the user in selecting the proper image from the image database.


```
age final project$ python3 detect_final.p
```

Directory: /home/kashishmiglani/Desktop/Image final project/Testing images

```
test_10_5.jpg   test_50_2.jpg    test_2000_4.jpg
test_20_1.jpg   test_100_1.jpg
test_20_2.jpg   test_100_2.jpg
test_20_3.jpg   test_100_3.jpg
test_20_4.jpg   test_500_1.jpg
test_50_1.jpg   test_500_2.jpg
```

Files of type: jpeg files (*.jpg)

Figure 2. GUI for Image Selection

Image segmentation means we are going to divide the complete images into many part in order to simplify and change the representation of an image into something that is more meaningful and easier to analyze. This part of the system comes under the image pre-processing part. This part of the system will help us in increasing the efficiency of our complete proposed system.

After the image segmentation part of our proposed system, next we are going to extracted the keypoint features such as FAST Feature, and the BRIEF Feature. ORB is basically a fusion of FAST key-point descriptor and BRIEF descriptor with many modifications to enhance the performance. First it use FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also use pyramid to produce multiscale-features. But one problem is that, FAST doesnt compute the orientation.

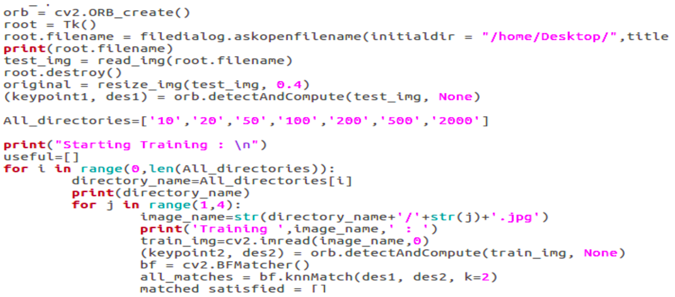

```
orb = cv2.ORB_create()
root = Tk()
root.filename = filedialog.askopenfilename(initialdir = "/home/Desktop/",title
print(root.filename)
test_img = read_img(root.filename)
root.destroy()
original = resize_img(test_img, 0.4)
(keypoint1, des1) = orb.detectAndCompute(test_img, None)

All_directories=['10','20','50','100','200','500','2000']

print("Starting Training : \n")
useful=[]
for i in range(0,len(All_directories)):
        directory_name=All_directories[i]
        print(directory_name)
        for j in range(1,4):
                image_name=str(directory_name+'/'+str(j)+'.jpg')
                print('Training ',image_name,' : ')
                train_img=cv2.imread(image_name,0)
                (keypoint2, des2) = orb.detectAndCompute(train_img, None)
                bf = cv2.BFMatcher()
                all_matches = bf.knnMatch(des1, des2, k=2)
                matched satisfied = []
```

Figure 3. Extracting ORB Feature from the Images

It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch.

Now for descriptors, ORB use BRIEF descriptors. But we have already seen that BRIEF performs poorly with rotation. So what ORB does is to steer BRIEF according to the orientation of keypoints. For any feature set of n binary tests at location (x_i, y_i), define a 2 x n matrix, S which contains the coordinates of these pixels. Then using the orientation of patch, theta, its rotation matrix is found and rotates the S to get steered(rotated) version Stheta.

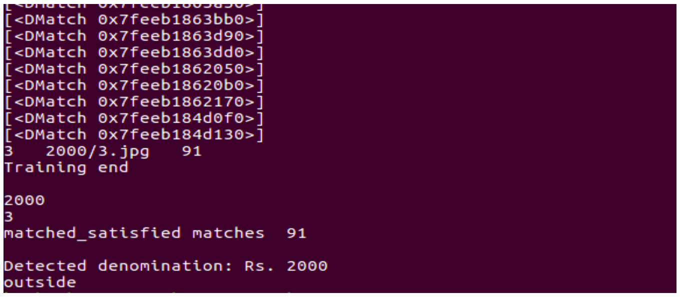
ORB discretize the angle to increments of (2 x pi) /30 (12 degrees), and construct a lookup table of precomputed BRIEF patterns. As long as the keypoint orientation theta is consistent across views, the correct set of points Stheta will be used to compute its descriptor.

BRIEF has an important property that each bit feature has a large variance and a mean near 0.5. But once it is oriented along keypoint direction, it loses this property and become more distributed. High variance makes a feature more discriminative, since it responds differentially to inputs. Another desirable property is to have the tests uncorrelated, since then each test will contribute to the result. To resolve all these, ORB runs a greedy search among all possible binary tests to find the ones that have both high variance and means close to 0.5, as well as being uncorrelated. The result is called rBRIEF.

For descriptor matching, multi-probe LSH which improves on the traditional LSH, is used. The paper says ORB is much faster than SURF and SIFT and ORB descriptor works better than SURF. ORB is a good choice in low-power devices for panorama stitching etc.

After extracting the key features from the image, then we are going to make the use of machine learning algorithms such as KNN Clustering in order to train our model. This train model will help us in detecting the currency with help of similarity metric such as Euclidean Distance, Manhattan Distance, Chessboard Distance, etc.

3.2. Analysis based on KNN - Clustering

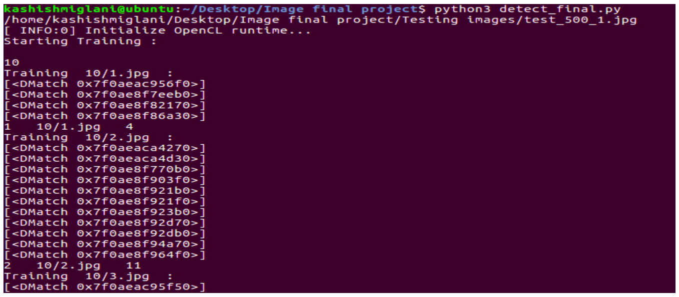


```
[<DMatch 0x7feeb1863bb0>]
[<DMatch 0x7feeb1863d90>]
[<DMatch 0x7feeb1863dd0>]
[<DMatch 0x7feeb1862050>]
[<DMatch 0x7feeb18620b0>]
[<DMatch 0x7feeb1862170>]
[<DMatch 0x7feeb184d0f0>]
[<DMatch 0x7feeb184d130>]
3   2000/3.jpg   91
Training end

2000
3
matched_satisfied matches  91

Detected denomination: Rs. 2000
outside
```

Figure 4. Training of our Model

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). The simple version of the K-nearest neighbor classifier algorithms is to predict the target label(here in this case detecting the currency) by finding the nearest neighbor class. The closest class will be identified using the distance measures like Euclidean distance.

KNN - Classifier is used to find minimum needed neighbours to get the correct prediction from the dataset. After training and finding the minimum needed neighbours our proposed system is able to produce the good results with the precision around 70%.



```
kashishmiglani@ubuntu:~/Desktop/Image final project$ python3 detect_final.py
/home/kashishmiglani/Desktop/Image final project/Testing images/test_500_1.jpg
[ INFO:0] Initialize OpenCL runtime...
Starting Training :

10
Training  10/1.jpg  :
[<DMatch 0x7f0aeac956f0>]
[<DMatch 0x7f0ae8f7eeb0>]
[<DMatch 0x7f0ae8f82170>]
[<DMatch 0x7f0ae8f86a30>]
1   10/1.jpg   4
Training  10/2.jpg  :
[<DMatch 0x7f0aeaca4270>]
[<DMatch 0x7f0aeaca4d30>]
[<DMatch 0x7f0ae8f770b0>]
[<DMatch 0x7f0ae8f903f0>]
[<DMatch 0x7f0ae8f921b0>]
[<DMatch 0x7f0ae8f921f0>]
[<DMatch 0x7f0ae8f923b0>]
[<DMatch 0x7f0ae8f92d70>]
[<DMatch 0x7f0ae8f92db0>]
[<DMatch 0x7f0ae8f94a70>]
[<DMatch 0x7f0ae8f964f0>]
2   10/2.jpg   11
Training  10/3.jpg  :
[<DMatch 0x7f0aeac95f50>]
```

Figure 5. Training our Proposed System

3.3. Conclusion

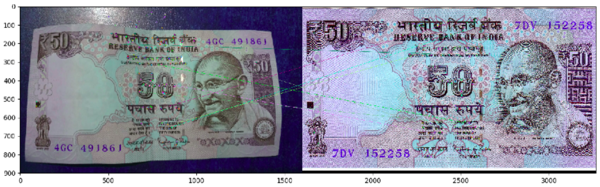


Figure 6. Input And Output Image from our Proposed System

This paper proposed a Content Based Image Retrieval System which make the use of ORB Feature in order to extracted the feature from the input image. And then it make the use of KNN Clustering Algorithm in order to form the clusters of the images and then it make the use of the similarity metric such as Euclidean Distance, Manhattan Distance to detect the currency. Our proposed system is having the precision around 70%.

4. References

[1] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, BRIEF: Binary Robust Independent Elementary Features, 11th European Conference on Computer Vision (ECCV), Heraklion, Crete.LNCS Springer, September 2010.

[2] Sahana Murthy, Jayanta Kurumathur, B Roja Reddy: Design and implementation of paper currency recognition with counterfeit detection, DOI: 10.1109/GET.2016.7916838 IEEE Computer Society (2017)

[3] Mohammad H Alshayeji, Mohammad Al-Rousan, Dunya T. Hassoun: Detection Method for Counterfeit Currency Based on Bit-Plane Slicing Technique, (2015)