INDIANA UNIVERISTY BLOOMINGTON

Final Project: Currency Reader for People with Visual Impairments

B657 Computer Vision

REPORT

Submitted by

Sagar Bhandare
Masters Candidate in Computer Science
School of Informatics and Computing
Indiana University
sagabhan@indiana.edu

Sanket Mhaiskar
Masters Candidate in Computer Science
School of Informatics and Computing
Indiana University
smhaiska@iu.edu

Under the Guidance of:

Professor David Crandall
Asst. Professor, School of Informatics and Computing
Indiana University

Problem and Motivation

A large number of people are unfortunately affected by visual impairments. India is home to around 30% of the world's population as per the Huffingtonpost statistics. Visual Impairments drastically affect the quality of life and cause a lot of hardships for these populations. They also limit the daily activities. One of the daily activities is management of Currency. Currencies have been designed to also help the visually impaired recognize the denominations. But it is still somewhat difficult for the visually impaired to handle cash currency.

1. Background and Related Work

Dedicated devices such as "LookTel Money Reader" have been introduced to help reading currency, but they are often bulky and expensive. There is also a free application EyeNote by U.S. Bureau of Engraving and Printing for the Apple iOS platform for detecting US currency. There is no such version for Android mobile phones. The combined imaging and computational power of new devices has inspired us to embed image processing and computer vision algorithms into the devices. Although for this project we have not ported the model files to the Android Application but used a client server architecture with a good response time. The model files can be easily ported to the mobile phone and the interaction with a remote server can be avoided.

2. Approach

Since our application tries to cater to the visually impaired people, the interface is a single click view which uses the inbuilt camera functionality and sends data to a hosted server. The response is then received in a text format and then the application uses Text to Speech API to voice out the detected currency.

To achieve a high accuracy, we have used pixel values as features. We first thought of using features from Gray Scale images, but that would be a drawback of the trained model since it would also classify copies or fake currencies as real. Even though the system is not fool-proof, it tries to accurately predict the denomination of the Indian currency.

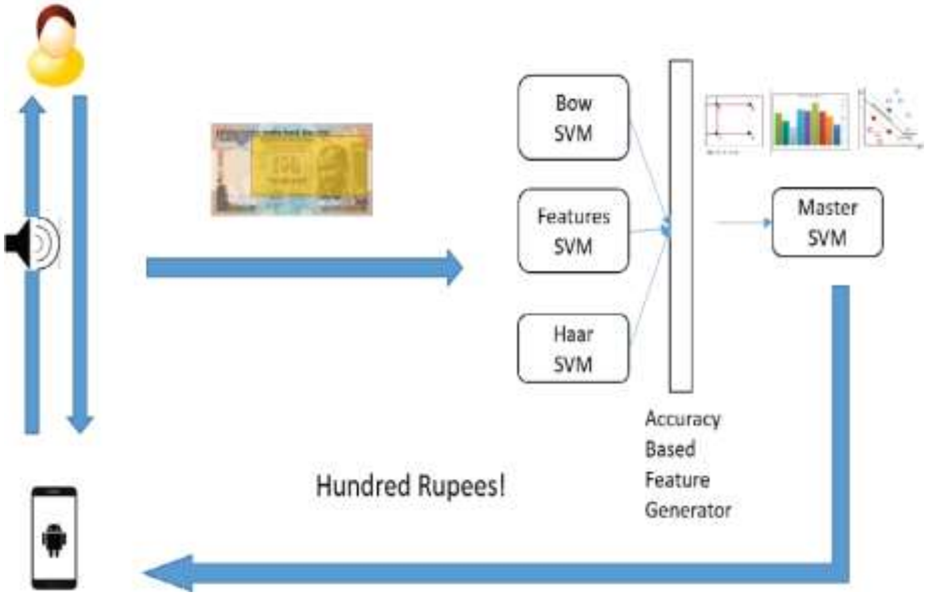
Implementation Details

We identified following challenges for this project,

1. Getting currency information from visually impaired person
2. Identify currency and non-currency
3. After identifying it is a currency, try to find the denomination of currency
4. Provide feedback to visually impaired person

Proposed approach to each challenge:

1. Getting currency information from visually impaired person
    a. Since most of the people have android phones with them we decided to use mobile app as a way to get the image of currency.
    b. That is, we decided to build android app which can take picture of a currency in hand
    c. An image of currency without any sense of orientation
2. Identify currency and non-currency
    a. We can do this by using bag of words approach.
    b. There are few features in each currency which are common and there are few which are unique to that currency.
    c. The features of image can be identified by performing SIFT analysis and getting the key points.
    d. Now we, cluster them using k-means to form visual vocabulary.
    e. Each cluster represent some feature of image.
    f. Now, for each image from train data we can find out the clusters which receive maximum SIFT vectors for each denomination. Create group of these clusters for each denomination.
    g. For a new image if any of these clusters groups get SIFT vectors below 60% of SIFT vectors, it is identified as non-currency.
3. After identifying it is a currency, try to find the denomination of currency
    a. We used combination approach for this challenge.
    b. We created a 4 SVM model that decides the denomination of currency together. Following diagram explains the model in brief,



We created set of classifiers which detect type of currency

c. The first is Bag of words classifier with SIFT detection on selected areas to get key points and generate visual vocabulary. Train SVM based on all images using generated vocabulary.

d. Second is, feature based which considers pixel attributes including color, relative contrast to generate features vector.

e. Last is Haar like feature classifier, which generate feature vector from most significant filtered areas.

f. Accuracy based feature generator generates feature vector for master SVM based on accuracy of each classifier for each class of currency.

4. Provide feedback to visually impaired person

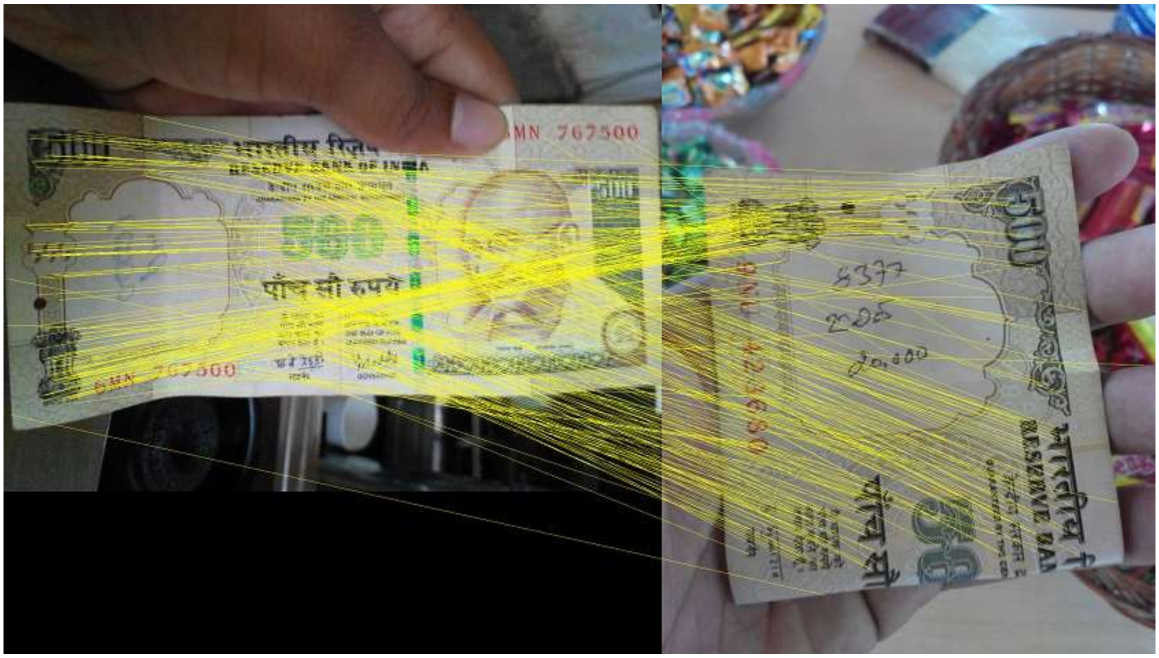
   a. We chose to provide the feedback in terms of speech to the visually impaired person.

   b. We decided to use built in android text to speech API to provide feedback to user.

IMPLEMENTATION DETAILS

1. Identifying currency features

- We have used SIFT (Scale-invariant feature transform) algorithm to detect and describe local features in images. We have used the Lightweight C++ Implementation of David Lowe's SIFT by Andrea Vedaldi (http://www.robots.ox.ac.uk/~vedaldi/).
- As shown in the image below the SIFT algorithm detects key features and generates a feature vector of 128 Dimensions.



2. Generating visual vocabulary (Bag of Words)

- We iterate over every image in the dataset, and extract the SIFT features from it. Each SIFT feature is a 128 dimensional vector.
- We take all the SIFT vectors of every image, put it together, and run Kmeans on it. Once we run clustering, we have visual words. Next, for every image in the training dataset, we iterate over each the SIFT vector and try to find the nearest visual word using Euclidean distance.
- Since each SIFT vector can be associated with a visual word, we can create a histogram among these visual words, and return the frequency (histogram) as the feature vector for SVM classification. The visual words is written out to a file. This would be the core model for Bag of Words.

3. Haar Like Features

- We generated random points and random window sizes to calculate the Haar like Wavelets.
- Calculated the Summed Area Tables for every image i.e Integral Images.
- We then calculated Hx and Hy i.e two Haar projects with different window orientations. Every feature value was the magnitude of Hx and Hy values.
- We then trained the SVM using the feature vector generated and tested it using the same random points and window size values generated for training.

4. Training

- For training we have used the Indian National Rupee Data Set. (Link in References)
- We have used SVM Multiclass Project (https://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html) executables to learn and classify currencies.
- The input file example_file contains the training examples. The file format is the same as for SVMlight, just that the target value is now a positive integer that indicates the class. The first lines may contain comments and are ignored if they start with #. Each of the following lines represents one training example and is of the following format:

  <line> .=. <target> <feature>:<value> <feature>:<value> ... <feature>:<value> # <info>
  <target> .=. <integer>
  <feature> .=. <integer>
  <value> .=. <float>
  <info> .=. <string>
- The target value and each of the feature/value pairs are separated by a space character. Feature/value pairs MUST be ordered by increasing feature number. Features with value zero can be skipped. The target value denotes the class of the example via a positive (non-zero) integer. So, for example, the line

  3 1:0.43 3:0.12 9284:0.2 # abcdef

  specifies an example of class 3 for which feature number 1 has the value 0.43, feature number 3 has the value 0.12, feature number 9284 has the value 0.2, and all the other features have value 0. In addition, the string abcdef is stored with the vector, which can serve as a way of providing additional information when adding user defined kernels.

5. **Testing**

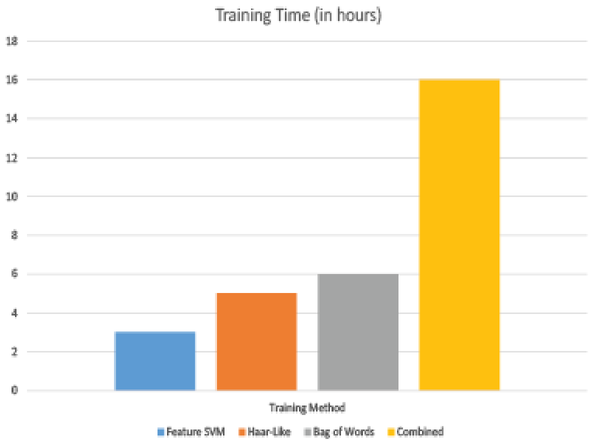
   The model files generated on training set are used to predict the denomination of currency whenever new image is fed to the classifier.

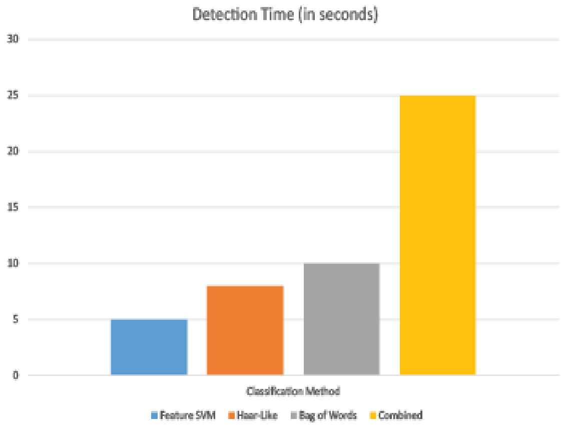
Deployment of Solution

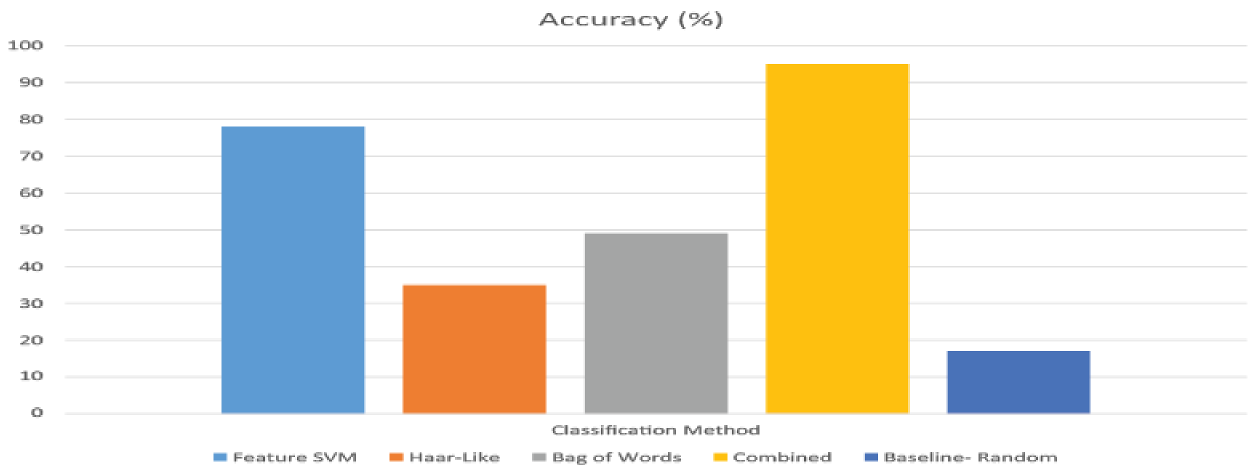
- We tried to port the android libraries like CImg, SIFT to the android mobile application. After spending considerable time of one week in this efforts and realizing we have reached dead end, we decided to change the deployment mechanism. We deployed the solution as server client approach.
- **Server:** The trained models are deployed as web server on a linux ubantu box with tomcat as hosting server. The server is developed using java spring and REST. The rest web service API accepts image file upload and returns currency denomination in JSON format.
- **Client:** Ideally any client having web service facility can query this service but here in this case mobile app is our client. The app takes the photo of currency using android inbuilt camera and sends it to the deployed server. The server URL is configurable in the application. The app gets back the JSON reply and gives speech output based on it.

Results:

On a test dataset of 50 images of each denomination, we achieved following results. The baseline chosen is random selection which gives 17% accuracy.







FUTURE WORK AND IMPROVEMENTS:

- This system can be further enhanced to detect fake currencies by training the SVM on negative and fake samples.
- In future the model files can be uploaded to android app and detection can be made there itself instead of deploying as client server.
- The accuracy can be further improved by training the SVM on currency images under various conditions of light and illuminations.

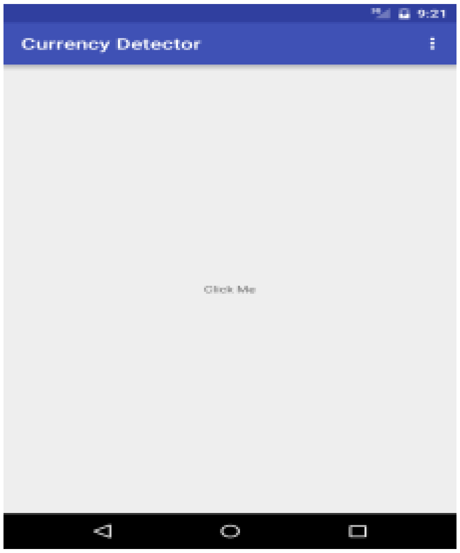
References

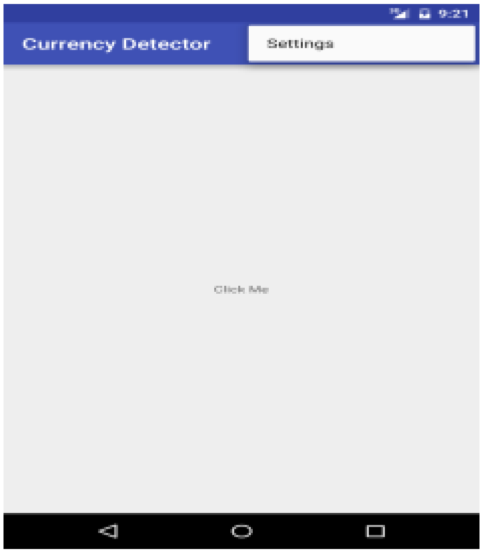
1. C.T. Baker Massof, R.W. Hsu and F.H. Barnett. Visual Disability Variables. I,II: The Importance and Difficulty of Activity Goals for a Sample of Low-Vision Patients. Archives of Physical Medicine and Rehabilitation, 86(5):946-953, 2005.
2. D.G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91-110, 2004.
3. 2005.6. M. zuysal, P. Fua, V. Lepetit. Fast keypoint recognition in ten lines of code. Computer Vision and Pattern Recognition, 2007. CVPR 07. IEEE Conference on, pages 1-8, 2007.h;8, 2007
4. P. Viola and M. Jones. Robust real-time face detection. International Jounrnal on Computer Vision, 57(2):137-154, 2004
5. LookTel Money Reader (http://www.looktel.com/moneyreader)
6. Indian National Rupee Data Set : https://drive.google.com/file/d/0B7Am6-nOVeP7N1lQOUVUYlBuc0E/edit?usp=sharing

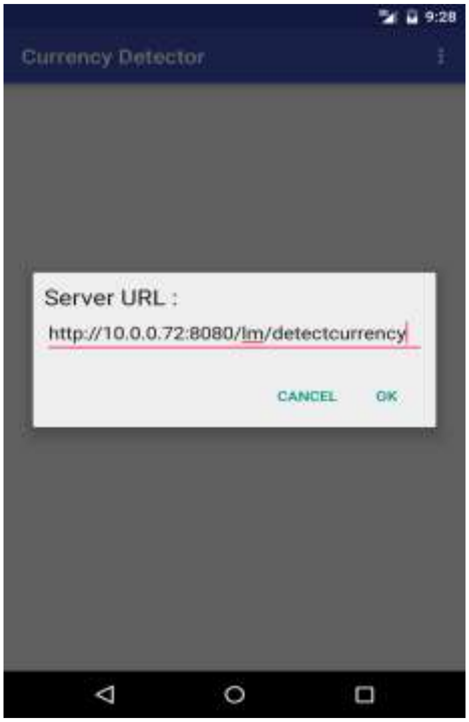
7. Mobile Currency Reader (http://src.acm.org/liu/liu.html)

Instructions for Using the Android Application:

Configure the server settings as shown. Click on Click Me to Open the camera and click an Image. The Image is then sent to the server and then a response is received as text. This response is then converted to a Voice Message with the denomination detection sound.







Server URL :

http://10.0.0.72:8080/lm/detectcurrency