B. TECH. PROJECT REPORT

On

Currency Recognition on Mobile

for Visually Challenged People

BY

Punit Lakshwani

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

DECEMBER 2018

Currency Recognition on Mobile for Visually Challenged People

Submitted in partial fulfillment of the requirement for the award of the degree

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

by

Punit Lakshwani

Under Guidance of

Dr. Surya Prakash
Assistant Professor
Computer Science and Engineering

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2018

CANDIDATE'S DECLARATION

I hereby declare that the project entitled **"Currency Recognition on Mobile for Visually Challenged People"** submitted in partial fulfillment for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** carried out under the supervision of **Dr. Surya Prakash, Assistant Professor, Discipline of Computer Science and Engineering, IIT Indore** is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student with date

CERTIFICATE BY BTP GUIDES

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of BTP Guides with date and their designation

Preface

This report on "**Currency Recognition on Mobile for Visually Challenged People"** is prepared under the guidance of Dr. Surya Prakash, Assistant Professor, Computer Science & Engineering.

Through this report I have tried to use Computer Vision techniques (using OpenCV) to design an innovative Android application that utilizes the camera of the mobile device to detect currency bills with the intention for robust, practical use by the visually impaired.

Also, my method is generic and scalable to multiple domains including those beyond the currency bills. My solution uses a visual Bag of Words (BoW) based method for recognition. I have tried to the best of our abilities and knowledge to explain the content in a lucid manner. I have also added models and figures to make it more illustrative.

Place: IIT Indore
Date: 1/12/2017

Acknowledgements

I would like to thank my B.Tech Project guide **Dr. Surya Prakash** for his guidance and constant support in structuring the project and his valuable feedback throughout the course of this project. His overseeing the project meant there was a lot that I learnt while working on it. I thank him for his time and efforts.

I am grateful to **Mr. G Iyyakutti Iyapan** without whom this project would have been impossible. He provided valuable guidance and also taught me how to write a thesis.

I am thankful to all my friends who helped in collecting pictures of Indian currency.

Lastly, I offer my sincere thanks to everyone who helped me complete this project, whose name I might have forgotten to mention.

Abstract

In this report, I present an application for recognizing currency bills using computer vision techniques that can run on a low-end smartphone. It is intended for robust and practical use by the visually impaired peoples. Though I have used the paper bills of Indian National Rupee (₹) as a working example but method is generic and scalable to multiple domains including those beyond the currency bills.

Solution uses a visual Bag of Words (BoW) based method for recognition. To enable robust recognition in a cluttered environment, we first segment the bill from the background using an algorithm based on iterative graph cuts. We formulate the recognition problem as an instance retrieval task.

This is an example of fine-grained instance retrieval that can run on mobile devices. I have evaluated the performance on a set of images captured in diverse natural environments, and reported an accuracy of **93.4%** on **4232 images**.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visual object recognition on a mobile phone has many applications. In this report, I focus on the problem of recognition of currency bills on a low-end mobile phone. This is an immediate requirement for the visually impaired individuals. There are around 285 Million people estimated to be visually impaired worldwide, out of which 39 Million are blind and 246 Million have low vision. The differences in texture or length of currency bills are not really sufficient for identification by the visually impaired. Moreover, bills are not as easy to distinguish by touch as coins. Certain unique engravings are printed on the bills of different currencies but they tend to wear away.

I adopt an approach based on computer vision on mobile devices, and develop an application that can run on low-end smartphones. I consider the bills of Indian National Rupee (₹) as a working example, but the method can be extended to a wide variety of settings. Our problem is challenging due to multiple reasons. Since our application is desired to be usable in a wide variety of environments (such as in presence of background clutter, folded bills etc.), we need a robust recognition scheme that can address these challenges. Also, visually impaired users may not be able to cooperate with the imaging process by realizing the environmental parameters (like clutter, pose and illumination).

The problem of currency recognition using computer vision techniques has been studied in the past. Neural networks have been used for recognition. Hidden Markov Model has also been exploited using texture characteristics of the bills as a feature. While most of the above work has shown high accuracy for classification, the test cases have usually consisted of scanned or carefully captured bills. These test cases lack variations in illumination, environment, texture and dimension. Most of the previous methods formulate the solution as one which gets trained offline with enough positive and negative examples. However, our approach is based on the formulation as an instance recognition under clutter.

1.1 Motivation for the work

I have been motivated by a lot of work that has been done recently on computer vision for mobile phones, and its various applications. Robust detection and recognition of objects of interest has been one major area of study. In this architecture, the client mobile system acts as the input/output device while performing minimal tasks. Work that has involved processing solely on the mobile phone primarily focuses on robust detection and recognition of objects, 3D reconstruction and Augmented Reality. Applications have also been targeted for visually impaired users as well.

The target audience being the visually impaired introduces additional challenges. The user is unaware of the condition of the surrounding environment — other objects, lighting, contrast, and even whether the bill is properly placed in the field of view of the camera or not. The system should be robust towards a wide variety of images that are likely to be captured by the target user. Using the application should be simple and intuitive for a person who cannot see. It should have a custom camera that once started requires no input from the user. In short, the

problem at hand requires innovative modules that can recognize the bill in diverse environments reliably, robustly and efficiently.

1.2 Design and Challenges

Working on a mobile platform brings with it a number of unique challenges that need to be taken care of. Primarily, the restrictions are in the memory, the application size, and the processing time. Currently, the average size of an iOS application is 23MB, while the RAM limit for a Windows phone application is 150MB. For an application to run on a mobile phone without affecting the others, it should not use more than 100MB of storage and 50MB of RAM. Our application recognizes the bills in two major steps. First we segment the bill from the clutter. Then we look at the most similar bill in the database. Though both these problems can be solved with good performance using many state-of-the-art computer vision algorithms, they are not really mobile friendly. The recognition model and other necessary information for our application would typically require more than 500MB of storage and 200MB of RAM with a direct implementation. This exceeds practical limits by a large amount.

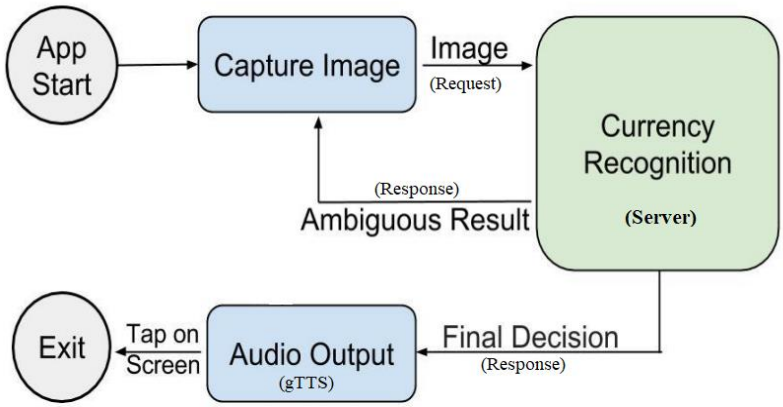

Figure 2: High-level control flow diagram

The target audience being the visually impaired introduces additional challenges. The user is unaware of the condition of the surrounding environment — other objects, lighting, contrast, and even whether the bill is properly placed in the field of view of the camera or not.

The system should be robust towards a wide variety of images that are likely to be captured by the target user. Using the application should be simple and intuitive for a person who cannot see. It should have a custom camera that once started requires no input from the user. In short, the problem at hand requires innovative modules that can recognize the bill in diverse environments reliably, robustly and efficiently.

Chapter 2

Project Work

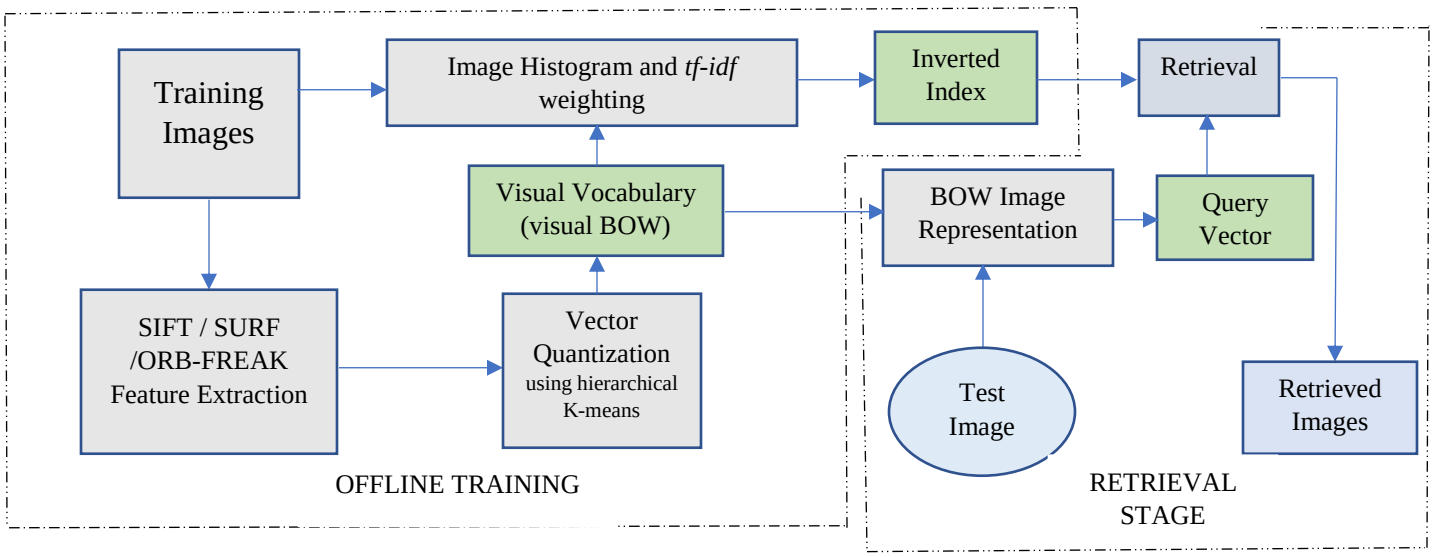

Figure 2: A conceptual schematic of the back-end

2.1 Segmentation (Preprocessing)

The images might be captured in a wide variety of environments, in terms of lighting condition and background while the bill in the image itself could be deformed. Image segmentation is important not just for reducing the data to process but also for reducing irrelevant features (background region) that would affect the decision-making.

We start with a fixed rectangular region of interest (ROI) which is forty pixels smaller from all four sides than the image itself. We assume that a major part of the bill will be present

inside this region. Everything outside this ROI is a probable background. Once this region is obtained, it must be extended to a segmentation of the entire object.

Let x be an image and let y be a partition of the image into foreground (object) and background components. Let $x_i \in R^3$ be the color of the i^{th} pixel and let y_i be equal to +1 if the pixel belongs to the object and to -1, otherwise. For segmentation we use a graph cut based energy minimization formulation. The cost function is given by

$$E(x,y) = -\sum_i \log p(y_i|x_i) + \sum_{(i,j)\in\mathcal{E}} S(y_i, y_j|x)$$

The edge system Ɛ determines the pixel neighborhoods and is the popular eight-way connection. The pairwise potential $S(y_i,y_j|x)$ favours neighbor pixels with similar color to have the same label. Then the segmentation is defined as the minimizer $\arg\min_y E(x, y)$. I use the GrabCut algorithm, which is based on iterative graph cuts, to carry out foreground/background segmentation of the images captured by the user.

The system should be able to segment the foreground object correctly and quickly without any user interaction. Whenever the foreground area is smaller than a pre-decided threshold, a fixed central region of the image is marked as foreground.

2.2 Instance Retrieval

1) Building a Visual Vocabulary:

I first locate keypoints in the foreground region of the image (obtained from segmentation) and describe the keypoint regions, using any descriptor extractor like SIFT, SURF or ORB-FREAK. We obtain a set of clusters of features using hierarchical K-means algorithm.

The distance function between two descriptors x_1 and x_2 is given by

$$d(x_1, x_2) = \sqrt{(x_1 - x_2)^\top \Sigma^{-1} (x_1 - x_2)}$$

where Σ is the covariance matrix of descriptors. As is standard, the descriptor space is affine transformed by the square root of Σ so that Euclidean distance may be used. The set of clusters forms the visual vocabulary of images.

2) Image Indexing Using Text Retrieval Methods:

For every training image, after matching each descriptor to its nearest cluster, we get a vector of frequencies (histogram) of visual words in the image. Instead of directly using visual word frequencies for indexing, we employ a standard 'term frequency - inverse document frequency' (*tf-idf*) weighting. Suppose there is a vocabulary of *k* words, then each image is represented by a k-vector $V_d = (t_1, \ldots, t_i, \ldots, t_k)^T$, of weighted word frequencies with components

$$t_i = (n_{id}/n_d)log(N/n_i)$$

Here n_{id} is the number of occurrences of word *i* in document *d*, n_d is the total number of words in the document *d*, n_i is the total number of occurrences of term *i* in the whole database and *N* is the total number of documents in the whole database. The weighting is a product of two terms: the word *frequency* (n_{id}/n_i), and the *inverse document frequency* $log(N/n_i)$. However, retrieval on this representation is slow and requires lots of memory. This makes it impractical for applications on mobile phones. Therefore, we use an inverted index for instance retrieval.

3) Retrieval Stage:

At the retrieval stage, we obtain a histogram of visual words (query vector) for the test image. Image retrieval is performed by computing the normalized scalar product (cosine of the angle) between the query vector and all *tf-idf* weighted histograms in the database. They

are then ranked according to decreasing scalar product. We select the first 10 images for further processing.

4) Spatial re-ranking:

The Bag of Words (BoW) model fails to incorporate the spatial information into the ranking of retrieved images. In order to confirm image similarity, we check whether the keypoints in the test image are in spatial consistency with the retrieved images. We use the popular method of geometric verification (GV) by fitting fundamental matrix to find out the number of keypoints of the test image that are spatially consistent with those of the retrieved images.

5) Classification:

In the voting mechanism, each retrieved image adds votes to its image class (type of bill) by the number of spatially consistent keypoints it has (computed in the previous step). The class with the highest vote is declared as the result.

Chapter 3

Adaptation to Mobile

We were able to adapt the above solution to a mobile environment by creating an android app and a remote server on flask without sacrificing the effective accuracy. This allows us to achieve the best possible performance, given the severe restrictions in various aspects of the pipeline that we have to contend with. Segmentation using iterative graph cuts is generally slow and typically takes more than 1 second for a 800×600 image on a 2.2 GHz Dual Core processor. The recognition model needed for retrieval cannot be used directly on a mobile phone because of the memory requirement. A vocabulary of size 10K used for instance retrieval along with an inverted index requires approximately 1.4GB of storage space and 1.0 GB RAM. So we created a python flask server and an android app which output denomination amount by using gtts (google text to speech).

In the following section, we provide a description of our user-friendly Android app using screenshots.

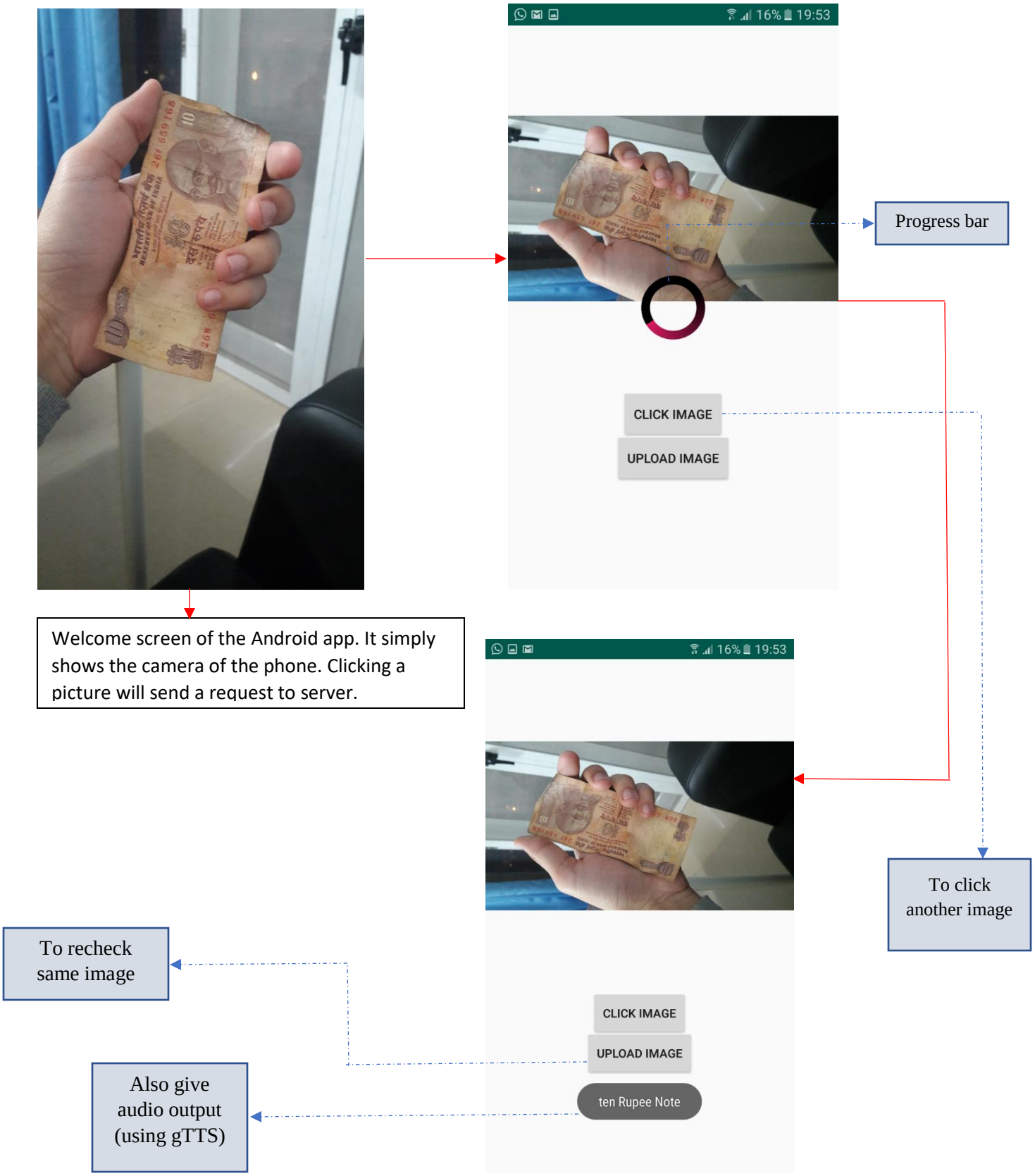

Figure 3: User-friendly Android app

Chapter 4

Dataset

The various denominations of Indian Rupee bills differ in size and color, apart from the printed denomination and other texts which makes for easy visual identification. However, for the visually impaired, text and color do not help at all and size can lead to confusion because of the similar dimensions of the various bills. Therefore, part of our work involved creating such a collection (Figure 4).

Figure 4: Images from the dataset with bills in varying illumination and background

Figure 5 shows statistics of this dataset. The images are captured using popular mobile phone cameras, with different resolutions, 2 megapixels (MP), on default setting. For each bill, there are 4 different half-folds and 2 full-length configurations. For each denomination we consider at least 12 different bills, across 6 different indoor environments and 7 different outdoor environments, while collecting the dataset. This introduces many variations in illumination, background and pose in the dataset. The dataset contains images of both new and worn out bills, as well as bills with scribbles on them.

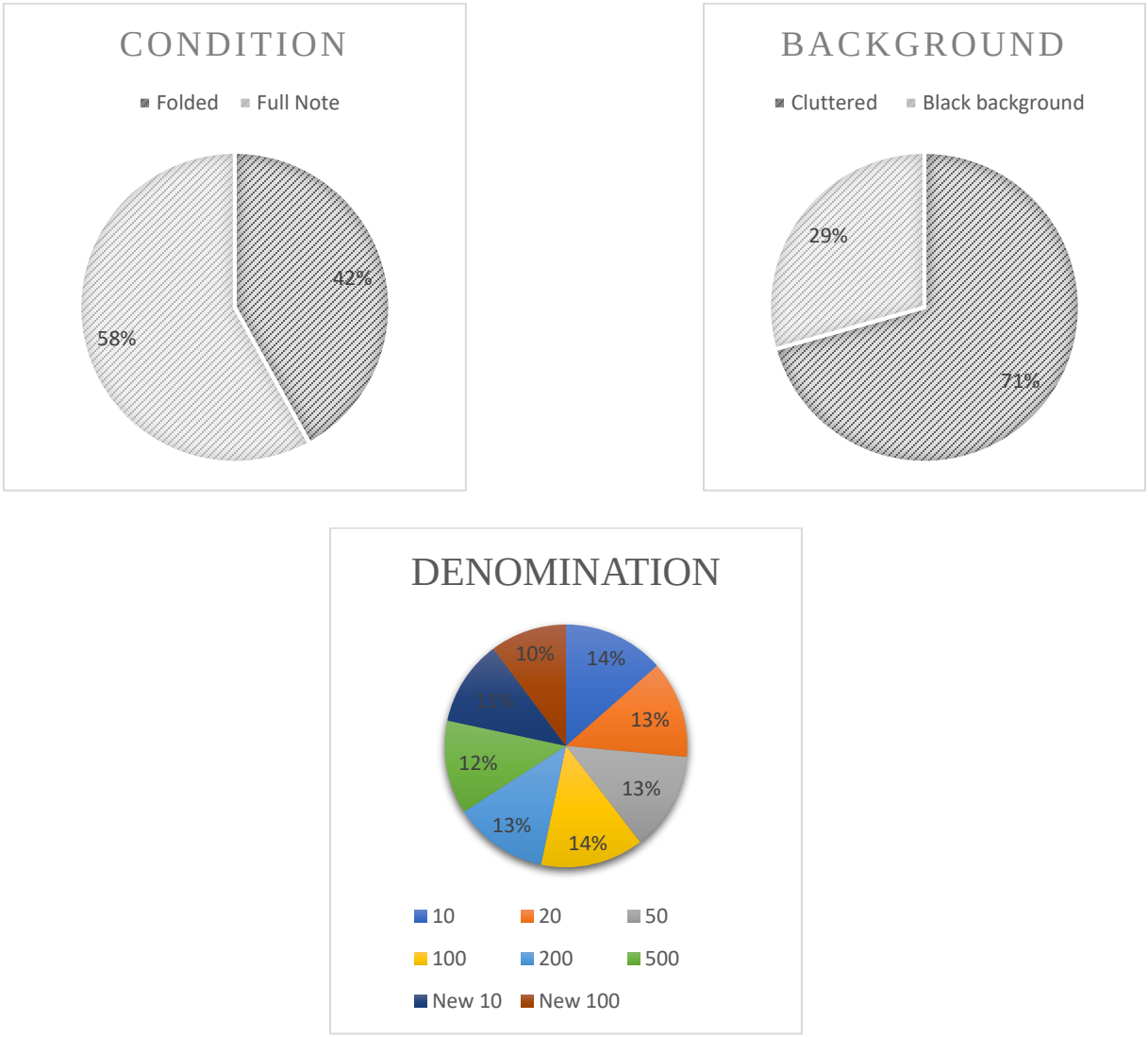

Figure 5: Various statistics that reflects the dataset's comprehensiveness

Chapter 5

Results and Discussion

I have experimented with the accuracy tests for various feature detectors and extractors like SIFT, SURF and ORB-FREAK. However, SIFT is far better in terms of accuracy, with 93.4% of the responses being correct using a vocabulary of size 10K. For the same vocabulary size, SURF was able to correctly recognize 88.7% of the test cases whereas for ORB-FREAK it was only 71.2%.

Using segmentation with instance retrieval improves retrieval performance. In our case, segmentation helps in removing irrelevant keypoints which results in faster retrieval as well as a reduced error rate. However, a wrong segmentation (marking the note as background) may result in a classification error.

Segmentation also helps in reducing the processing time in consecutive steps while increasing the accuracy by some amount. This is due to fewer keypoints being considered for description and geometric verification, which takes longer than segmentation.

With our approach we have been able to report a recognition accuracy of 93.4% on our dataset of Indian Rupee (₹).

When there are bills of multiple denominations in the view of the camera, the result is bound to be ambiguous. Since the user may be unaware of the surroundings, we have no workaround for this situation.

Another case where failure is common, is when the autofocus has not functioned properly, or the phone has been shaken or moved while capturing the image. This results in an image being blurred

or the bill being out of focus. In cases where the image is blurred and the system fails to detect keypoints. So, when the bill is not in focus, the result is either ambiguous or incorrect.

Table 3: Accuracy with dataset of cluttered background images

	Dataset Size for training (tested on 560 images)		
Feature Extraction	744	2254	4232
SIFT	71.2%	88.7%	93.4%
SURF	68.7%	79.4%	84.6%
ORB	57.8%	72.2%	77.4%

Table 4: Accuracy without cluttered background images

	Dataset Size for training (without cluttered background) (tested on 300 images)	
Feature Extraction	542	1228
SIFT	96.4%	98.2%

Table 5: Accuracy of dataset with folded notes only

	Dataset Size for training (with only folded notes) (tested on 400 images)	
Feature Extraction	872	1778
SIFT	85.4%	88.2%

Chapter 6

Conclusion & Future work

I have succeeded in our aim to develop a system that can be used to recognize currency for a visually impaired user and ported the system to a mobile environment. The methods used work well on noisy images captured from a mobile phone. Currency retrieval and thereafter recognition is an example of fine-grained retrieval of instances which are highly similar. This requires segmentation for removal of clutter. Through our experiments, it has been established that segmentation is helpful for the retrieval process as it reduces the chance of reporting erroneously as well as the overall processing time, and also that the instance retrieval method ensures results swiftly. We expect our system to easily adapt to other currencies of the world as well as a collection of various currencies simultaneously while keeping a similar level of accuracy and speed. And we can also create an android without need of remote server by utilizing OpenCV library for computer vision and image processing related tasks. While the camera interface is coded in Java, the language of Android, the image processing in the application to be done in native C++ code.

Bibliography

1. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos : http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic03.pdf
2. C. Rother, V. Kolmogorov, and A. Blake. GrabCut: interactive foreground extraction using iterated graph cuts : https://cvg.ethz.ch/teaching/cvl/2012/grabcut-siggraph04.pdf
3. An introduction to Bag-of-Words in NLP : https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428
4. Rabia Jafri, Syed Abid Ali, and Hamid R. Arabnia. Computer visionbased object recognition for the visually impaired using visual tags. IPCV, 2013.
5. Image Retrieval with Bag of Visual Words: https://in.mathworks.com/help/vision/ug/image-retrieval-with-bag-of-visual-words.html