**Project 3 Proposal**

**Name: Parshwa Rajyaguru**

**Class: CS6825 Computer Vision**

**Section 1: Goal Statement**

Problem is Stop traffic sign detection and recognition for blind-low vision pedestrian.

**Section 1.1:**

Input Image is like:

Conditions in which my real time system work is:

* Good Lighting and Not working at night.
* No shadow and tree cover sign board.
* Sign board should not be covered by snow or any other things.
* Camera should be within +/-20 degree range from traffic sign board.
* No glare and bulb light or sun rays hits in such a way that it cover traffic sign board.
* Traffic sign board should not be curved or tilt.

**Section 1.2:**

When stop traffic sign detected, it output “STOP” to speech as audio and draw boundary in blue color around stop traffic sign in that.

**Section 2: Development Platform and Target AVD and Device**

I am using Android Studio 2.3 as my real time system development platform and my target device is

API 25 Android 7.1.1 Nougat, using OpenCV 3.1.0 and I will test it on Nexus 5 (Android 6.0.1) with API 23 as my physical device.

**Section 3: Algorithms**

**Section3.1: Overview**

I will have 3 main component into my Traffic sign detection system. In first component, I will detect whether stop sign into image or not using color thresholding and edge detection feature.

Then second component is stop sign recognition using contour detection and pattern matching algorithm. The third component is final result stage in which if system find stop sign, it draw outline around stop sign and through text speech it alert user.

**Section 3.2: Stop Sign Detection**

In this component, I will do color thresholding(red color) and convert it to HSV color space.

Reference- <https://solarianprogrammer.com/2015/05/08/detect-red-circles-image-using-opencv/>

<http://stackoverflow.com/questions/32522989/opencv-better-detection-of-red-color>

<http://docs.opencv.org/3.2.0/df/d9d/tutorial_py_colorspaces.html>

Then I will perform edge detection on that. I will use ‘Canny Edge’ Algorithm for edge detection.

Reference- <http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html>

**Section 3.3: Stop Sign Recognition**

At recognition stage, two basic method I will implement.

First is Contour detection. In which I will basically do whether any octagonal shape detected in image.

Reference- <http://stackoverflow.com/questions/20176768/opencv4android-detect-shape-and-color-hsv>

<https://github.com/michaeltroger/shape-detection>

<http://www.androidpeak.com/shapesimage-android-image-api-shape-imageviews/>

<http://opencv-srf.blogspot.com/2011/09/object-detection-tracking-using-contours.html>

Second stage is template matching. In which, I will take one stop sign template as input and compare it with stop sign detected in image.

Reference- <http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html>

<http://answers.opencv.org/question/94744/opencv-with-android-template-matching-using-camera-images/>

<http://answers.opencv.org/question/22262/opencv-with-android-template-matching/>

<http://stackoverflow.com/questions/17001083/opencv-template-matching-example-in-android>

<http://stackoverflow.com/questions/18336673/opencv-android-template-matching-from-camera>

**Section 3.4: Result**

I will alert user with outline stop sign with blue color and through text to speech. The first part will show image on ‘ImageView’.

Reference- <https://developer.android.com/reference/android/widget/ImageView.html>

<https://www.mkyong.com/android/android-imageview-example/>

The second text to speech will be done using standard android TextToSpeech class.

Reference- <https://developer.android.com/reference/android/speech/tts/TextToSpeech.html>

**Section 4: GUI Interface**

Button – “Take Photo” for Capturing Image

Image View where processed image display.

If System found stop traffic sign, it will use ‘TextToSpeech’ android class to alert user through audio that stop sign detected.

