

**Capstone Design (2)**

**Spring, 2019**

**School of Software, CAU**

**Week #2**

**-Proposal-**

**Team list.**

**20166282 양 태 성**

**20165770 김 병 국**

**20166319 윤 종 엽**

Index

[1. Introduction 3](#_Toc3146263)

[2. Goal 3](#_Toc3146264)

[3. Motivation 3](#_Toc3146265)

[4. Development and Implementation contents 3](#_Toc3146266)

[4.1 User case 3](#_Toc3146267)

[4.2 used tools 4](#_Toc3146268)

[5. Team roles & Project schedules 4](#_Toc3146269)

# Introduction

People with color blindness usually experience discomfort in many cases.

In order to solve the problem of the traffic light, we inform users of the current color of the traffic light that is in front of the camera.(smartphone or LCD of Raspberry Pi)

  Through, color blind people recognize the signal correctly and prevent traffic accidents.

In the case of a pedestrian, the camera and the screen of the smartphone application can be used,

and in the case of the driver, the color of the traffic light in front of the vehicle can be accurately recognized by using the camera and the display of Raspberry Pi.

# Goal

For pedestrians who are color-blind, they illuminates traffic lights in front of them by using the smartphone camera. The display shows information about what the current traffic light is.

For drivers who are color-blind, they attach camera to the car to confirms whether there is a traffic light in the front, and if there is a traffic light, information about the present signal(ex. color) is displayed through the display attached to the vehicle.

Because accurate signal recognition is important, we focus on the accuracy of signal recognition.

# Motivation

People with color blindness suffer from many inconveniences, such as driving and walking for color of traffic lights. Sunglasses for color blindness are available now, but it is difficult to commercialize in terms of cost, so we can make it easy for people with color blindness to use common applications, cameras and displays in cars.

# Development and Implementation contents

## User Case

1. **For Pedestrians**

At first, through the application, the user enters the desired traffic light on the screen through the smartphone camera. Through the image processing technique by using openCV, it confirms whether there is a traffic light on the input screen and grasps current signal information of the traffic light. And outputs the detected information to the smartphone screen.

1. **For Drivers**

First, install the camera and display by using Raspberry Pi in the vehicle. A traffic light in front of the vehicle is input to the screen of Raspberry Pi.

And the input screen is transmitted to the server, and information on the current signal is obtained through the image processing technique by using openCV. Additional information is displayed on the display of the Raspberry Pi.

## Used tools

1. **C++ OpenCV 3.0.0**
2. **Firebase**
3. **Android Studio**
4. **Raspbian**

# Team roles & Project schedules

|  |  |  |  |
| --- | --- | --- | --- |
| **Names Weeks** | **양태성** | **김병국** | **윤종엽** |
| **1** | **Team Organization** | | |
| **2** | **Selecting subject, Advance investigation, Producing PPT** | | |
| **3** | **design** | | |
| **4** | **OpenCV Learning and Settings** | **Constructing Server** | **Android ui design** |
| **5** | **Feature extraction for distinguishing traffic lights** | **Application Sever Interworking** | |
| **6** | **OpenCV Imprved accuracy** | **Prepare for Midterm**  **& Write Midterm PPT** | **Write Midterm Report** |
| **7** | **Midterm Demonstration and Presentation** | | |
| **8** | **Mid term** | | |
| **9** | **OpenCV Imprved accuracy** | **Raspberry Learning** | |
| **10** | **OpenCV Imprved accuracy** | **Raspberry development Setting** | |
| **11** | **OpenCV, Server, Paspberry Interworking** | | |
| **12** | Create final demo  **Creating Final Demo** | | |
| **13** | **Final Check** | **Prepare for Final**  **& Write Final PPT** | **Write Final Report** |
| **14** | **Final Demonstration and Presentation** | | |
| **15** | **Final Report and Manual Submission** | | |