GPS SUBSTITUTE FOR BUSES BASED ON IMAGE RECOGNITION

FINAL REPORT

APRIL 2020

# 

# 

# Acknowledgement

# ABSTRACT

The GPS technology is the fastest and most efficient method to pinpoint the live location of a device on a map, as it detects typically accurate within a 4.9m (16ft.) radius under the open sky. However, when a device is surrounded by man-made or natural obstructions decreases their accuracy. In this project, an alternative approach is used to determine the location of moving buses on a route with the assistance of a camera on the bus. The primary idea of this project is to compare two images of the same place while keeping the fact that there can be temporary objects like moving cars, pedestrians, and parked vehicles which were present in the first picture may not be visible in the second image due to time-scale difference, To achieve this goal, the project is divided into four stages to set the ground for the procedural process starting from gathering dataset of a predetermined bus route from using the first video to take out every frame on a video once the dataset is saved each frame is divided in 4 regions and processed to identify the region of interest (ROI) in each division using OpenCV ORB algorithm to detect salient features in each frame and finally the region with highest number of features is considered as ROI and saved with the name of the location of that frame, from this point when the bus sends an image it will only look for that particular region of interest in a frame using a technique called template matching, to verify that ROI is matched correctly structural similarity is used which return the matching ratio, if the ratio is more than 80% then successful match takes place.

Table of content

# About Project

In this project the main task is to determine the location of buses through the pictures captured from the front camera on the Bus and then compare the images from the stored images of the same place.

Further in the project I will be using modern techniques to detect the objects on the image and will apply modern algorithms like **Hear Cascade** is basically a **classifier** which is used to detect the object for which it has been trained for, from the source, the project will be implemented using python programming language and a modern library will be used OpenCV ( a library especially written for Image processing) .

## Development Strategy As per requirement of the project the flow of work is logged on Jira, this project has adapted Kanban environment rather than scrum which suits if working in a team, and also because Kanban methodology allows due to its flexibility in a development by limiting the amount of in a progressive manner, in this way changes in the project are made regular basis on the basis of the feedback which is the reason this methodology is adopted as this project will be done with regular supervision.

## OpenCV

OpenCV is a library written in different languages including Python, the library mainly focusses on image processing, video capture and analysis, it offers a variety of different features which includes object detection and face detection. In the development process of this program, OpenCV will play an important role, as it will be used to read images, capture frames, manipulate images, perform detections such as sign board, bus stops, faces, differentiate between the objects, it can be also used for image processing operations such as image filtering, image transformation and etc.

## Basic Scenario

The Bus will send the video which will be recorded from the camera of the bus through a secure connection between the bus and the backend system, the system will detect the incoming data and perform a data integrity check, after that video will processed by comparing the frames which have been previously stored in the system’s database, if the object in the frame will match from the image in the database it will determine the location of the bus through the data tags attached with pictures, In this way the system will be able to perceive the current location of the bus.

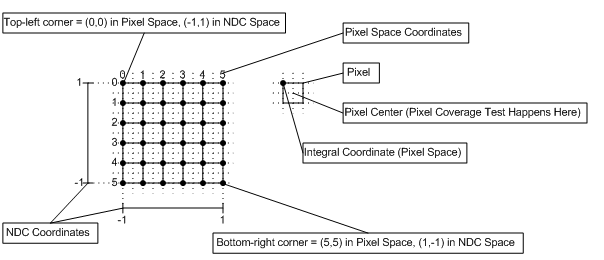
## Approach

Once the video is transmitted to the system, the system will disintegrate the video into frames, the video will be disintegrated in timely manner so that we can capture frames on every 10th sec or straight route it can be every 1min, which can be edited according to the requirement. A GUI will be  made to analyse each image with the frame already available in the database to make it more user friendly with the many other features to perform the image processing operation, even though the whole project can be made without the interaction from GUI, however to make it more convenient and to perform a reliable analysis the project will consist of a GUI.

## Implementation

The OpenCV allows us to capture frames in by importing the library in the program, which will save the images in a system by creating a  unique directory in order to avoid conflicts in a filesystem, further after that each frame is opened in a GUI which will be made from a library called Tkinter, although OpenCV can read images into GUI however it does not offer many choices to fulfil the design requirements, such as adding buttons or display two images simultaneously in the same window for better analysis, to resolve this problem Tkinter is a very popular library which will be integrated with the program.

After loading the image into the GUI, the image processing will be applied to manipulate the images by drawing a rectangle into the picture and extract the pixel coordinate of the picture, the Figure 1.1 describes the clear understanding this implementation.



OpenCV drawing functions allows us to make the rectangles and circles in the pictures, for example: [cv.rectangle](https://docs.opencv.org/master/d6/d6e/group__imgproc__draw.html#gac865734d137287c0afb7682ff7b3db23)(img,(384,0),(510,128),(0,255,0),3) top-right corner of image.

Furthermore, in  a next step an attempt to detect the objects in the images will be made by identifying the existence of objects in the image, OpenCV have introduced a wide range of algorithms for detecting objects in the image, Algorithms like **scale-invariant feature transform** (**SIFT**)  algorithm can be used to detect the SIFT key points of objects are first extracted from a set of reference images[[2]](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform#cite_note-Lowe1999-2) and stored in a database, **FAST Algorithm for Corner Detection** is another algorithm which is feature detection algorithm which is used to match the features in the images, It is not necessary that these algorithms will be used in the project, these are mentioned as part of the background reading can be used  depending on the requirement and efficiency of the algorithm.

# 