

MINI PROJECT REPORT

On

OMNIBUS

B.E (IT) – IV / VI Sem

By

Thalla Pavan(160117737045)

Pradyumna(160117737046)

Under the guidance of

KNVS Sridevi

Assistant Professor



DEPARTMENT OF INFORMATION TECHNOLOGY

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)

(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)

KOKAPET(V),GANDIPET(M), RR District HYDERABAD -75

Website: www.cbit.ac.in

2019-2020



**CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY (A)**
Kokapet(Village), Gandipet, Hyderabad, Telangana-500075. / www.cbit.ac.in
ISO 9001:2015 Certified Institute



COMMITTED TO
RESEARCH,
INNOVATION AND
EDUCATION

41
years



CERTIFICATE

This is to certify that the project work entitled “ **OMNIBUS** ” submitted to CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY, in partial fulfillment of the requirements for the award of the completion of VI semester of B.E in Information Technology, during the academic year 2019-2020, is a record of original work done by **Thalla Pavan(160117737045) Pradyumna(160117737046)** during the period of study in Department of IT, CBIT, HYDERABAD, under our guidance.

Project Guide

KNVS Sridevi

Asst.Prof, Dept. of IT,
CBIT, Hyderabad.

Head of the Department

Dr.Suresh Pabboju

Professor, Dept. of IT,
CBIT, Hyderabad.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Knvs.Sridevi, our project guide, for her invaluable guidance and constant support, along with her capable instruction and persistent encouragement.

We are grateful to our Head of Department, Dr. Suresh Pabboju, for his steady support and for the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our Principal, Dr.P. Ravinder Reddy, as well as the management of the institute, for having designed an excellent learning atmosphere.

Our thanks are due to all member of the staff and our labassistants for providing me with the help required to carry out the groundwork of this project.

ABSTRACT

While trains get a lot of attention, buses can also deliver successful public transport services – if we can overcome some common problems. The main issues arise because buses share roads with cars. This tends to reduce the efficiency of bus travel, unless given a dedicated corridor like trains. Road congestion leads to frequent stopping, long journey times and delays in passenger pick-up and drop-off. Other issues include limited off-peak service, poor-quality shelters and unreliable timetables. In rural areas buses will reach bus-stops on time. So people won't face any problems in rural areas. But in urban areas bus schedule is not reliable because of fluctuations in bus timings.. Sometimes buses will reach on time and sometimes they will reach the stops 10 minutes or 20 minutes later.

Omnibus is developed using python in visual studio code. Omnibus is neither an app nor an website and also it doesn't have an interface. Public transport that is safe, efficient and effective is a core priority when it comes to city-building. So omnibus software helps in building a cost effective public transportation system. This software uses openCV module for image processing and twilio module for sending messages.

CONTENTS

| | |
|--|-----|
| • DECLARATION | ii |
| • AKNOWLEDGEMENT | iii |
| • ABSTARCT | iv |
| | |
| 1. INTRODUCTION | |
| 1.1. OVERVIEW | 1 |
| 1.2. MOTIVATION | 1 |
| 1.3. OBJECTIVE OF PROJECT | 1 |
| | |
| 2. EXISTING SYSTEM | 2 |
| | |
| 3. PROPOSED SYSTEM | |
| 3.1. METHODOLOGY | 3 |
| 3.2. ARCHITECTURE OF PROPOSED SYSTEM | 4 |
| | |
| 4. SOFTWARE AND HARDWARE REQUIREMENTS | |
| 4.1. VISUAL STUDIO CODE | 5 |
| 4.2. PYTHON | 6 |
| 4.3. OPENCV MODULE | 7 |
| 4.4. TWILIO MODULE | 9 |
| | |
| 5. IMPLEMENTATION | 11 |
| 6. RESULTS | 13 |
| 7. CONCLUSION | 14 |
| 8. FUTURE SCOPE | 14 |
| 9. BIBLIOGRAPHY | 15 |

LIST OF FIGURES

| | |
|--|----|
| Fig 3.1 Flow chart of Ominbus | 4 |
| Fig 5.1 Image processing code | 11 |
| Fig 5.2 Twilio module for sending messages | 11 |
| Fig 5.3 main code for processing the data | 12 |
| Fig 6.1 Message to bus driver | 13 |

1.INTRODUCTION

1.1 OVERVIEW

Omnibus is developed using python in visual studio code. Omnibus is neither an app nor an website and also it doesn't have an interface. Public transport that is safe, efficient and effective is a core priority when it comes to city-building. So omnibus software helps in building a cost effective public transportation system. This software uses openCV module for image processing and twilio module for sending messages.

1.2 PROBLEM STATEMENT

While trains get a lot of attention, buses can also deliver successful public transport services – if we can overcome some common problems. The main issues arise because buses share roads with cars. This tends to reduce the efficiency of bus travel, unless given a dedicated corridor like trains. Road congestion leads to frequent stopping, long journey times and delays in passenger pick-up and drop-off. Other issues include limited off-peak service, poor-quality shelters and unreliable timetables. In rural areas buses will reach bus-stops on time. So people won't face any problems in rural areas. But in urban areas bus schedule is not reliable because of fluctuations in bus timings.. Sometimes buses will reach on time and sometimes they will reach the stops 10 minutes or 20 minutes later.

.

1.3 OBJECTIVE OF PROJECT

The main objective of the project is to build effective transportation system and make people to take public transport without any difficulty. It also aims to decrease government expenses on fuel and also traffic problems by effectively controlling the buses.

2.EXISTING SYSTEMS

This is a new project for providing effective transportation. There is no effective existing system as such for providing effective public transportation. Governments have been using the schedule for buses since the existence of public transport. Buses use to start from depos at particular timings as per the schedule. This schedule for buses is being used even today. But as the population increased and people who take public transport increased mainly in urban areas it is difficult to maintain proper timings for buses.

3.PRPOSED SYSTEM

3.1 Methodology

This project provides effective public transportation. At each and every route in bus-stop a camera is installed which takes pictures of people for every 1 minute. These images are then stored in cloud and a one main computer retrieves images from the cloud and perform image processing on these images. After processing these images it generates a count which tells how many persons are present on the image. Based on the count at specified route in each and every bus-stop if specified number of people are present at that route then a message will be sent to bus driver to take the bus in that route. If minimum number of people are not present then bus is not assigned to that route. In this way we provide effective transportation. In this method there is no schedule for buses and they don't need to follow the timings.

3.1 ARCHITECTURE OF PROPOSED SYSTEM

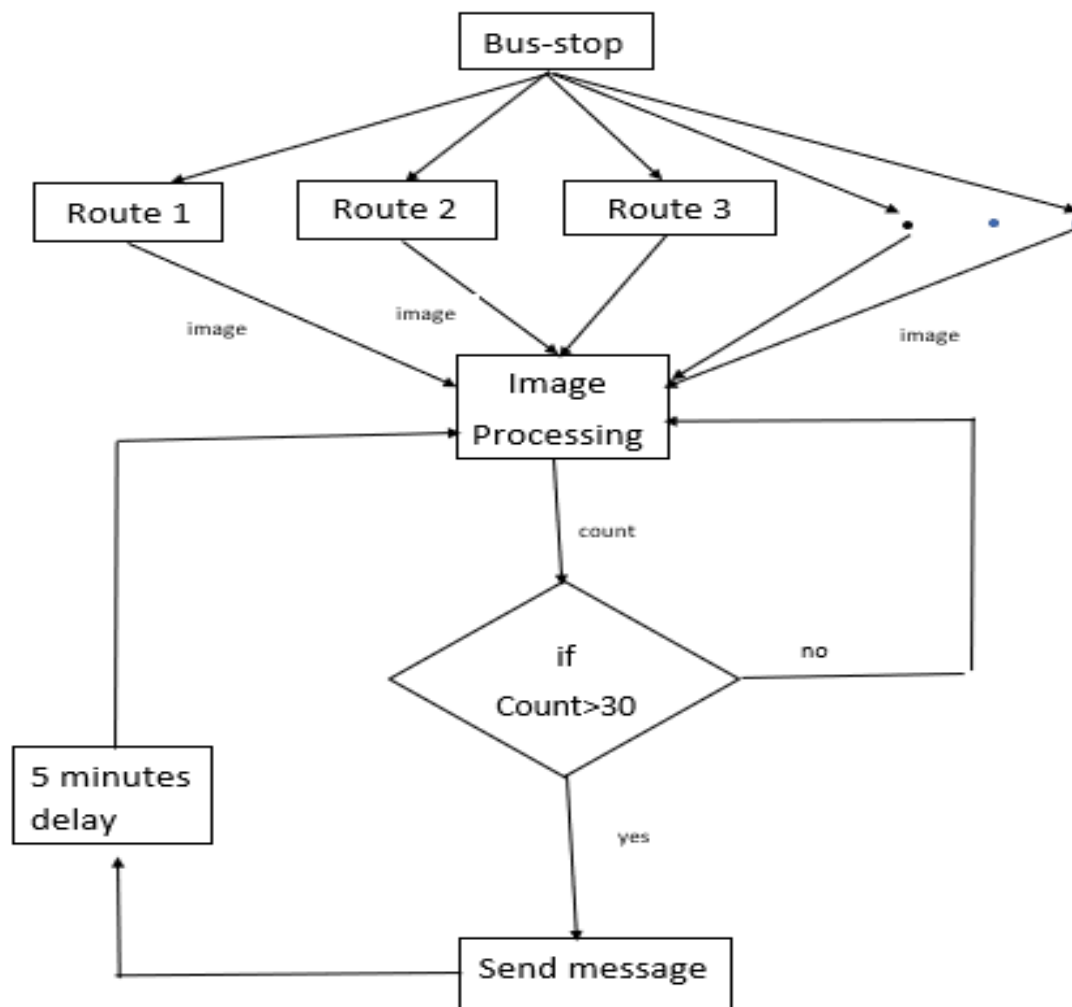


Fig 3.2.1 Architecture of proposed system

4. SOFTWARE AND HARDWARE REQUIREMENTS

This project is developed in python using visual studio code.

Other requirements include:

- openCV module and twilio module
- twilio account
- haarcascade_fronalface_default.xml classifier

4.1 VISUAL STUDIO CODE

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

4.2 PYTHON

Python is a general-purpose language, which means it can be used to build just about anything, which will be made easy with the right tools/libraries. Python is great for backend web development, data analysis, artificial intelligence, and scientific computing. Many developers have also used python to build productivity tools, games and desktop apps, so there are plenty of resources to help you learn how to do those as well.

Python was designed to be easy to understand and fun to. Fun is a great motivator, and since you'll be able to build prototypes and tools quickly with Python, many find coding in python a satisfying experience. Thus, Python has gained popularity for being a beginner-friendly language, and it has replaced Java as the most popular introductory language at top U.S. Universities.

Being a very high-level language, Python reads like English, which takes a lot of syntax-learning stress off coding beginners. Python handles a lot of complexity for you, so it is very beginner-friendly in that it allows beginners to focus on learning programming concepts and not have to worry about too much details.

As a dynamically typed language, Python is really flexible. This means there are no hard rules on how to build features, and you'll have more flexibility solving problems using different methods (though the Python philosophy encourages using the obvious way to solve things). Furthermore, Python is also more forgiving of errors, so you'll still be able to compile and run your program until you hit the problematic part.

Because Python is a dynamically typed language, the same thing can easily mean something different depending on the context. As a Python app grows larger and more complex, this may get difficult to maintain as errors will become difficult to track down and fix, so it will take experience and insight to know how to design your code or write unit tests to ease maintainability.

As a dynamically typed language, Python is slow because it is too flexible and the machine would need to do a lot of referencing to make sure what the definition of something is, and this slows Python performance down.

At any rate, there are alternatives such as PyPy that are faster implementations of Python. While they might still not be as fast as Java, for example, it certainly improves the speed greatly.

4.3 OPENCV MODULE

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

COMPUTER VISION

Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modelling and replicating human vision using computer software and hardware.

Computer Vision overlaps significantly with the following fields –

- Image Processing – It focuses on image manipulation.
- Pattern Recognition – It explains various techniques to classify patterns.
- Photogrammetry – It is concerned with obtaining accurate measurements from images.

FEATURES OF OPENCV LIBRARY

Using OpenCV library, you can –

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyse the video, i.e., estimate the motion in it, subtract the background, and track objects in it.
- OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc.
- This tutorial explains the concepts of OpenCV with examples using Java bindings.

OPENCV LIBRARY MODULES

Following are the main library modules of the OpenCV library.

Core Functionality

This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array Mat, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.core`.

Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, colour space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.imgproc`.

Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.video`.

Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.videoio`.

Objdetect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name `org.opencv.objdetect`.

Highgui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely `org.opencv.imgcodecs` and `org.opencv.videoio`.

4.4 TWILIO MODULE

Twilio is an SMS API Gateway. Twilio is a developer platform for communications. Software teams use Twilio APIs to add capabilities like voice, video, and messaging to their applications. This enables businesses to provide the right communications experience for their customers. Behind Twilio APIs is a Super Network, a software layer that connects and optimizes communications networks around the world. This is what allows your users to reliably call and message anyone anywhere.

With Twilio, you can reach customers in the ways they prefer, and engage with them effectively using context related to that interaction. As customer experience can increasingly make or break your brand, programmable communications has become more crucial than ever to the success of businesses today.

SMS API

A SMS API is well-defined software interface which enables code to send short messages via a SMS Gateway.

As the infrastructures for SMS communications and the internet are mostly divided, SMS APIs are often used to 'bridge the gap' between telecommunications carrier networks and the wider web. SMS APIs are used to allow web applications to easily send and receive text messages through logic written for standard web frameworks.

4.5 HAARCASCADE_FRONALFACE_DEFAULT.XML CLASSIFIER

Haarcascade_fronalface_default.xml is a haarcascade classifier which is used to detect the faces in an image.

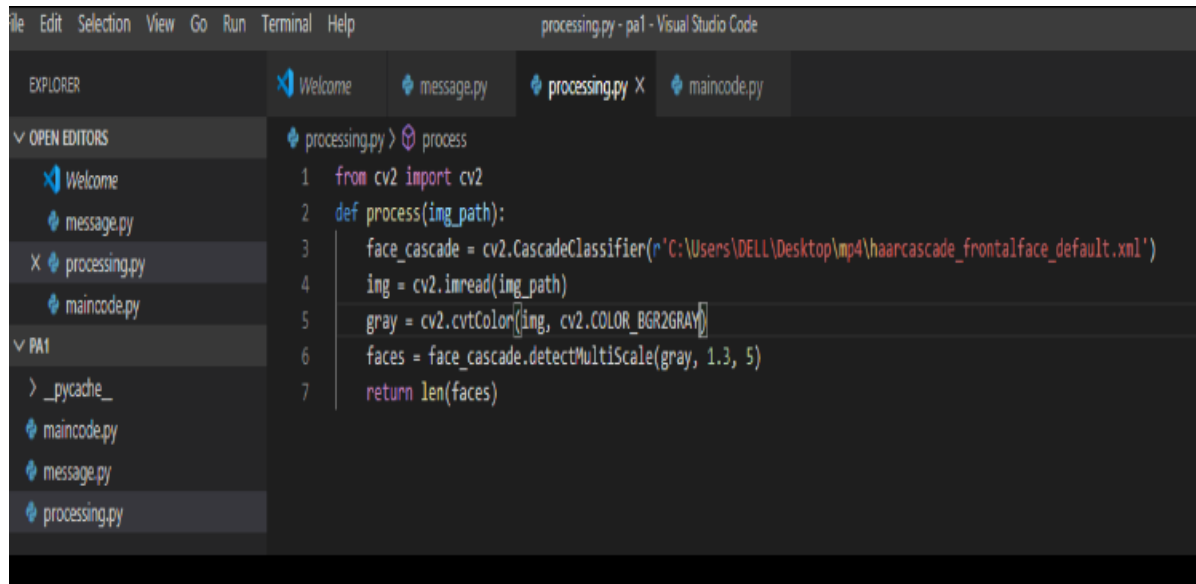
Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. OpenCV already contains many pre-trained classifiers for face, eyes, smiles, etc. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. The algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

A few things to note:

- The detection works only on grayscale images. So, it is important to convert the colour image to grayscale. (line 8)
- Detect Multiscale function (line 10) is used to detect the faces. It takes 3 arguments — the input image, scaleFactor and minNeighbours. scaleFactor specifies how much the image size is reduced with each scale. minNeighbours specifies how many neighbors each candidate rectangle should have to retain it. You can read about it in detail [here](#). You may have to tweak these values to get the best results.
- faces contain a list of coordinates for the rectangular regions where faces were found. We use these coordinates to draw the rectangles in our image.

5.IMPLEMENTATION

5.1 By using opencv's cv2 module we use haarcascade classifier to count the number of people in given image.



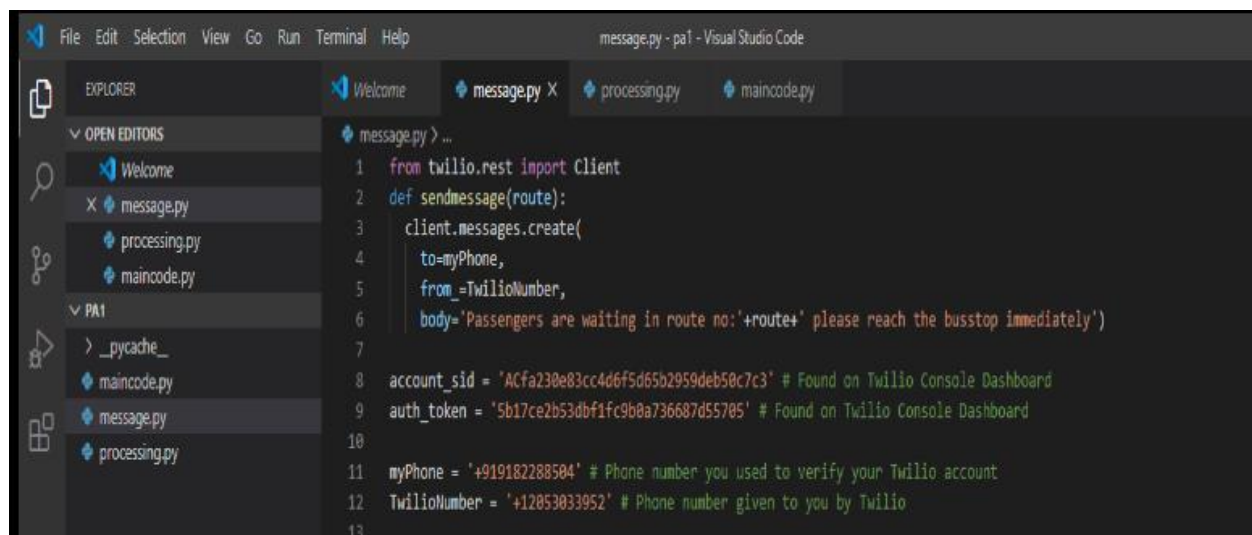
```
processing.py - pa1 - Visual Studio Code

EXPLORER
  Welcome
  message.py
  processing.py X
  maincode.py
  PA1
    _pycache_
    maincode.py
    message.py
    processing.py

processing.py
1 from cv2 import cv2
2 def process(img_path):
3     face_cascade = cv2.CascadeClassifier('C:\Users\DELL\Desktop\mp4\haarcascade_frontalface_default.xml')
4     img = cv2.imread(img_path)
5     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
6     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
7     return len(faces)
```

Fig 5.1 Haarcascade classifier

5.2 By using twilio.rest from twilio module and registering in twilio site we mention numbers in the code so that it will send the messages to that number.



```
message.py - pa1 - Visual Studio Code

EXPLORER
  Welcome
  message.py X
  processing.py
  maincode.py
  PA1
    _pycache_
    maincode.py
    message.py
    processing.py

message.py
1 from twilio.rest import Client
2 def sendmessage(route):
3     client.messages.create(
4         to=myPhone,
5         from_=TwilioNumber,
6         body='Passengers are waiting in route no: '+route+' please reach the busstop immediately')
7
8 account_sid = 'ACfa230e83cc4d6f5d65b2959deb50c7c3' # Found on Twilio Console Dashboard
9 auth_token = '5b17ce2b53dbf1fc9b0a736687d55705' # Found on Twilio Console Dashboard
10
11 myPhone = '+919182288504' # Phone number you used to verify your Twilio account
12 TwilioNumber = '+12053033952' # Phone number given to you by Twilio
13
```

Fig 5.2 Twilio code for sending messages

5.3 Finally by executing the main code it will first do image processing and then according to the threshold given, it will send the messages to the employees.

```

EXPLORER
  OPEN EDITORS
    Welcome
    message.py
    processing.py
    X maincode.py
  PAI
    > _pycache_
    maincode.py
    message.py
    processing.py
  OUTLINE
    {} message
    {} os
    {} processing
    {} time
    v assign
    def t

maincode.py > main
1  import os
2  import message
3  import processing
4  import time
5  def assign(route_path):
6      num=0
7      for f in os.listdir(route_path):
8          img_path=os.path.join(route_path,f)
9          num += processing.process(img_path)
10         # os.remove(img_path)
11     return num
12 def main():
13     dict1={}
14     path=r'C:\Users\DELL\Desktop\omnibus'
15     for l in os.listdir(path):
16         dict1[l]=0
17     while(1):
18         for l in os.listdir(path):
19             elapse_time=time.time()-dict1[l]
20             if(dict1[l]>0 and elapse_time>10):
21                 dict1[l]=0
22                 num=assign(route_path)
23                 elif(dict1[l]==0):
24                     route_path=os.path.join(path,l)
25                     num=assign(route_path)
26                     print(l)
27                     print(num)
28                     if(num>=2):
29                         message.sendmessage(l[5:])
30                         dict1[l]=time.time()
31                     break
32     break
33 main()

```

Fig 5.3 main code for processing the data

6.RESULTS

6.1 Message to the bus driver if specified number of people are present in the route.

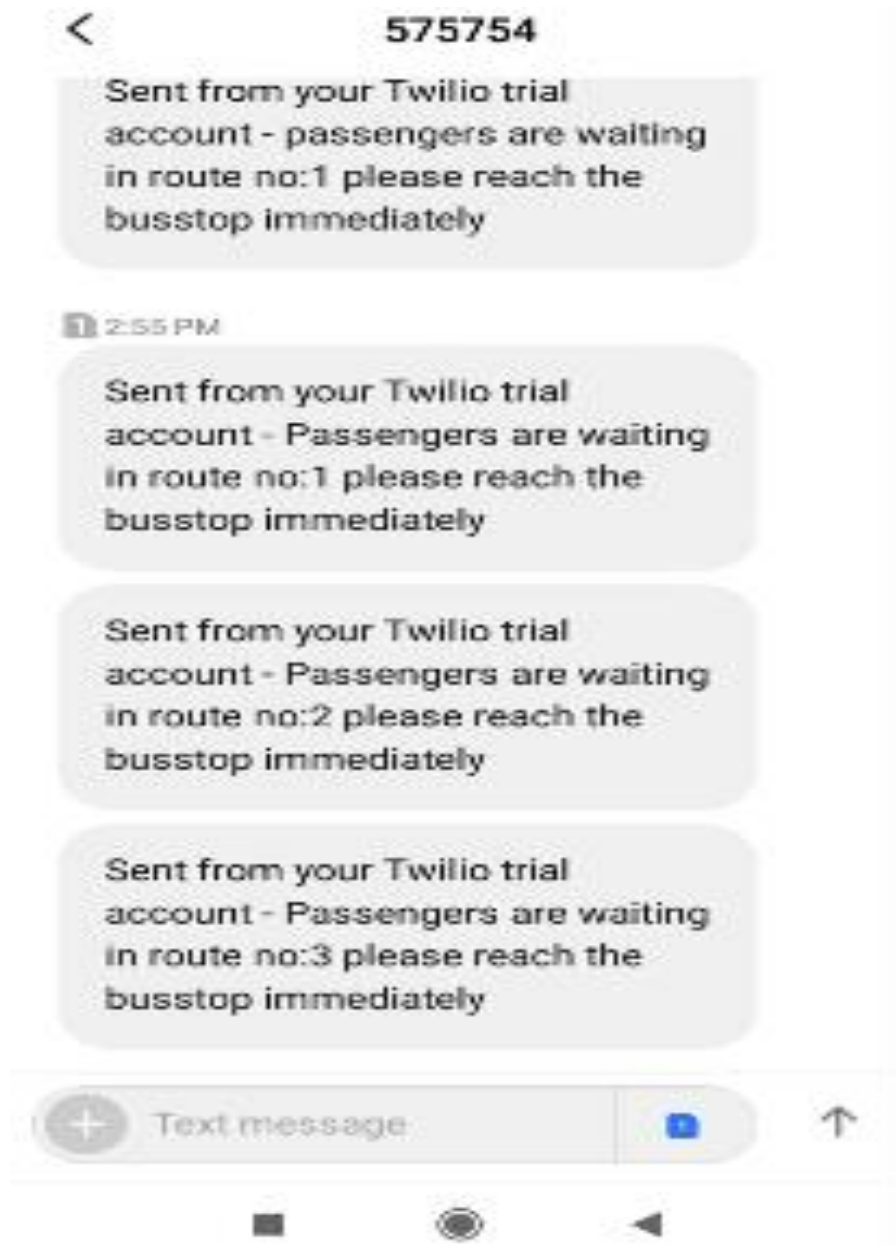


Fig 6.1 Message to bus driver

7.CONCLUSION AND FUTURE SCOPE

Advantages of using this software are:-

- It saves time for people who want to take public transport.
- It saves fuel and hence it reduces government expenses.
- It reduces traffic to some extent.

We can develop this project further by collecting the count each and every day and each and every hour and prepare timetable for the next month accordingly. We can have a mixture of both schedule for buses as well dynamic allocation of buses for providing effective transportation.

8.BIBLIOGRAPHY

- [1] <https://stackoverflow.com>
- [2] <https://www.tutorialspoint.com>
- [3] <https://www.twilio.com>
- [4] <https://docs.opencv.org/master>