As per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology, 7(04): 146-150, 2021

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0704023
Available online at:http://www.ijmtst.com/vol7issue04.html

Bus Monitoring System using AI

Krupansh Shah[1] | Sharan Kanchan[1] | Rithvik Yermal[1] | Sawankumar Naik[2]

[1]UG Student, Mechatronics Department, Mukesh Patel College of Technology Management and Engineering, Mumbai, India.
[2]Asst. Professor, Mechatronics Department, Mukesh Patel College of Technology Management and Engineering, Mumbai, India.

To Cite this Article
Krupansh Shah, Sharan Kanchan, Rithvik Yermal and Sawankumar Naik, "Bus Monitoring System using AI", *International Journal for Modern Trends in Science and Technology*, Vol. 07, Issue 04, April 2021, pp.:146-150.

Article Info
Received on 22-March-2021, Revised on 11-April-2021, Accepted on 17-April-2021, Published on 21-April-2021.

ABSTRACT

In this present world, the one suffering from global pandemic, maintaining social distance is a must. Covid 19 has changed the perception of the whole world. India is also going through the same situation and population is making it more challenging to keep the pandemic in check. Also, due to the high population, we are all aware that traffic situation is also a big issue in India. To avoid this traffic jams and constant hikes in gasoline prices most of the people are using public vehicles but unnecessarily end up wasting hours standing at bus stops.

So in this project, we are going to help these people by letting them know the situation of bus whether it is crowded or empty by notifying the passenger count. Our system will also be able to detect the emotions of the passengers so as to detect any potential threat travelling amidst the crowd .The system is designed by using image processing technique. Raspberry pi camera is connected with Raspberry Pi 3B in the bus for detecting images of the people on the bus and sending the data to the server via 4G communication. This system uses open source computer vision (open CV) real time object detection to analyze and process the data then calculate the count of the passenger by using the maximum face detection data.

KEYWORDS: *Open CV, V2 camera, Raspberry Pi, image processing, Artificial Intelligence*

I. INTRODUCTION

Public transport are getting awfully crowded, with increasing demand we need to find the balance between space and resources allocated to meet the needs of the population. BEST is the second-largest mass transport system in Mumbai after the suburban train network and ferries 30 lakh commuters daily. To ease the travel of commuters we proposed this idea of a seat occupancy detection system. Some of the BEST buses are so crowded and travelers have to wait a lot for the bus but with this system they can easily check which bus would be more accessible and also comfortable for them thus saving them abundant time.

Also in this pandemic maintaining social distance is the only solution we have right now. This is also one of the reasons to propose this idea so that commuters can travel without breaking social distancing norms

Covid 19 which has affected each and every person in one way or the other. Each and every human, whether employed or owners, students or teachers, all are facing issue of traveling to their respective destination by maintaining social distancing to take care of their own health. Also due to traffic jams and increasing petrol and diesel prices people are opting public transport.

II. SYSTEM ARCHITECTURE

A. Hardware

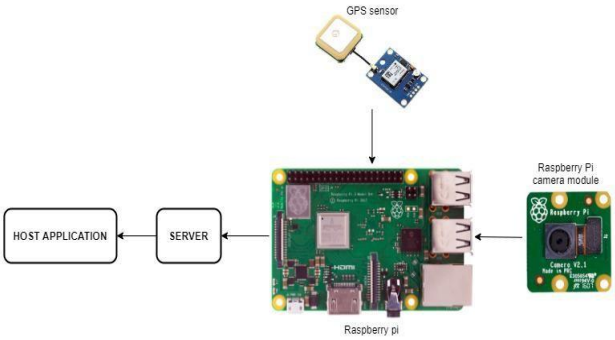


Fig. 1: System Block Diagram

1) Raspberry pi :

It is a credit-card-sized single-board computer developed by Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. Raspberry pi has a Broadcom BCM2853 System on Chip (SoC) which includes ARM1176JF-S 700 MHz processor [7].

Raspberry Pi comes with 2 USB ports and a 10/100 Ethernet controller. The Raspberry Pi does not come with a real-time clock, so an OS must use a network time server, or ask the user for time information at boot time to get access to time and date for file time and date stamping. For video output Raspberry use Composite RCA (PAL and NTSC), HDMI. Current support for the CSI (Camera Serial Interface) and DSI (Display Serial Interface) is limited at the time of writing. The GPIO (General Purpose Input Output) connector allows for a variety of external devices to be connected.

2) Raspberry Pi V2 Camera :

The Raspberry Pi Camera Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras.

3) *USB 2.0* :

Universal Serial Bus (USB) is a master- slave device which is made up of many slaves and a single master. The slaves are called the peripherals and the master is the host. Only the host has the ability to begin data transfer. The slaves cannot initiate data transfer. They only respond to the master's instructions.

The bidirectional communication between the USB is dependent on logical channels – called as 'pipes'. It connects the host controller to the device endpoint [3]. The endpoint is a logical entity which resides onto the device. The connections established has the pipe like structure which we used in connection oriented data transfer. A USB device can have 32 connections- two of which are reserved for giving Ack. So a total of 30 are present for normal use for the data transfer in bidirectional manner.

B. Software

Python:

Python is an object-oriented programming language that is designed in C. By nature, it is a high-level programming language that allows for the creation of both simple as well as complex operations. Along with this Python comes inbuilt with a wide array of modules as well as libraries which allows it to support many different programming languages like Java, C, C++, and JSON.

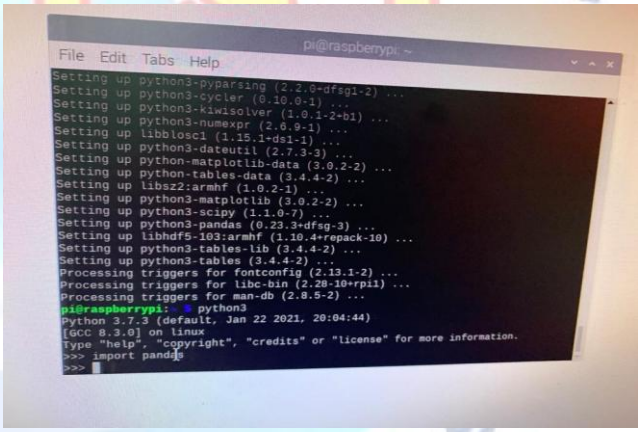


Fig.2 Setting up Python and installing Python libraries

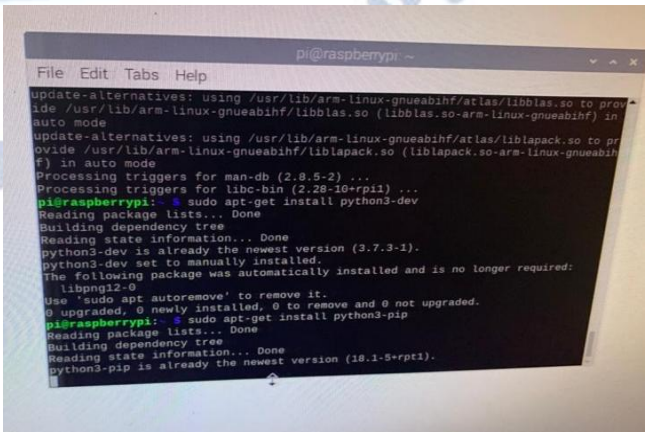


Fig.3 Installing Python3-Developer

Open CV :

We have to import this and several other libraries such as the numpy , keras, tensor Flow and some small libraries to make the open CV work efficiently .Open CV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. Open CV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, using flash, follow eye movements, recognize scenery.

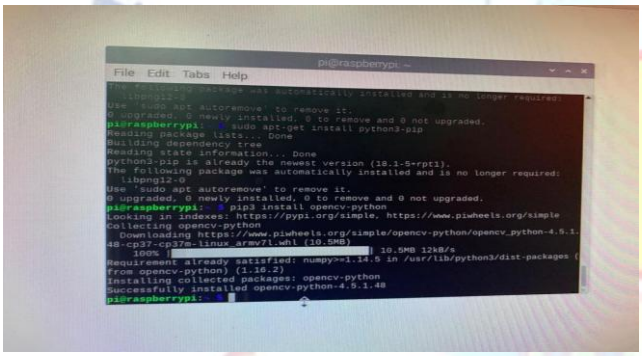
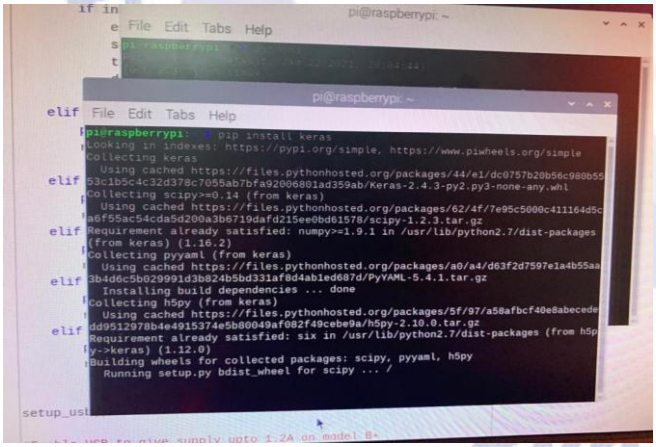


Fig.4 Installing OpenCV



. *Fig.5 Installation of keras*

III. SYSTEM WORKING

Firstly, the camera captures a video footage which is directly fed to the processor which is the Raspberry pi. The Raspberry pi compares them with the dataset loaded in the SD card. We have used the fer2013 dataset here as it is the best dataset for image detection and processing. Thus when we write the code we train the dataset and test it for predicting the outcome for a test image. On comparing the continuous feed from the camera with the dataset and code there are rectangles formed around faces of individuals depicting that there are these many people currently travelling on the bus. Thus this process is continuously repeated until the ordinance key is pressed and we exit from the output screen.

IV. IMPLEMENTATION ALGORITHM

A. The following two scenarios can halt the face detection;

The person is not sitting appropriately as in their full face frame is not visible evidently thus leading to the rectangle not being formed around the face and person going undetected.

Raspberry pi Malfunctioning: If the Raspberry pi is lagging a lot or corrupted then the code will not run eventually leading to face not being detected.

B. Termination:

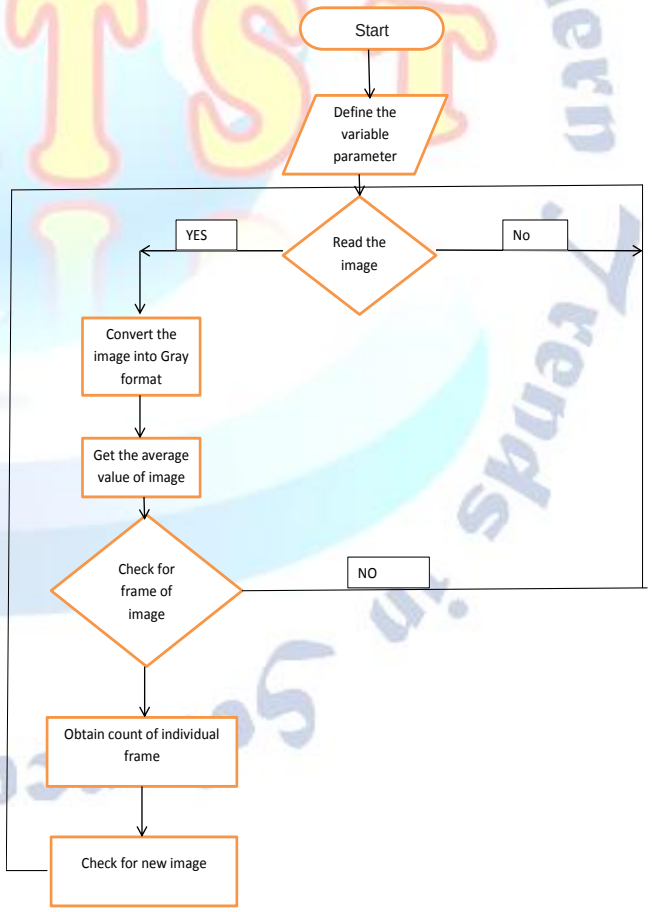


Fig. 6: Flowchart of the algorithm

In the above flowchart we firstly let the system read the image which is the direct input from V2 camera. If the image is detected, then it is

converted to grey scale format or else start over. After converting in gray image average value of gray image is calculated and then number of frames in the image is checked. If frames are not detected than start over again by reading the image. Once the frames are checked individual frame count is obtained. These individual frames signify the number of individual currently travelling in the bus. If one frame is detected means one person is travelling. New image is checked from camera and input in processor. The above process is repeated again and again until terminated.

V. RESULTS

Fig. 7: Actual implementation on prototype

Firstly we try and formulate the code in a python related application such as the spyder or even Pycharm .I have used spyder .The concepts used in the process of writing the code are OpenCV , image processing and Machine Learning. We developed a small code for testing purposes which was able to read the video input from the raspberry pi v2 camera and send the images continuously to the processor which is the Raspberry pi. If the code has no errors then we try and run the code. But firstly since the v2 camera is an output device we will have to enable it by activating the terminal and by selecting the"enable Camera" option in system settings as shown in the below image.

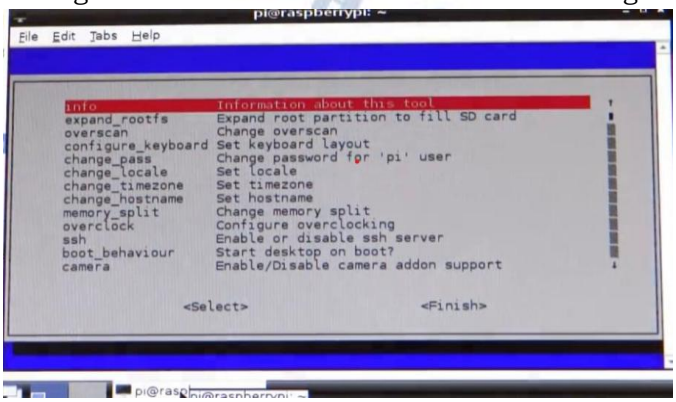


Fig 8 Camera Activation

After enabling the V2 camera we import the camera libraries the Terminal so that we can see the output on the screen when we run the code.

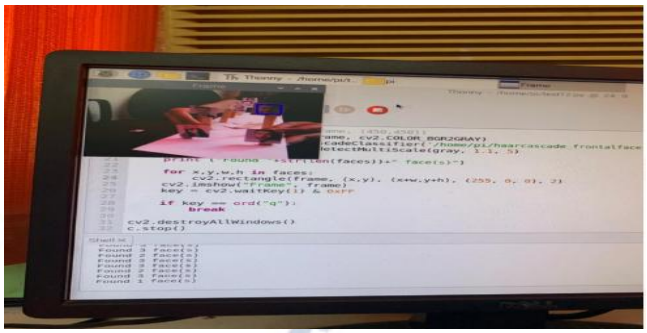


Fig 9 Output from Testing

From the above prototype we were able to conclude that :

1)The pictures in the bus were overlapping each other and were not organized.

2)The pictures were too small in frame size.

3)The V2 camera was easily able to detect the actual human faces than faces in the images.

Hence we modified the code a bit and changed the prototype where the faces were occupying most of the image and the seating of the passengers was much more organized like the actual bus.

But here the load on the raspberry pi was increasing and thus there were constant malfunctions happening. Thus a "Watchdog" had to be installed so that the raspberry pi stopped malfunctioning and freezing after capturing a few frames

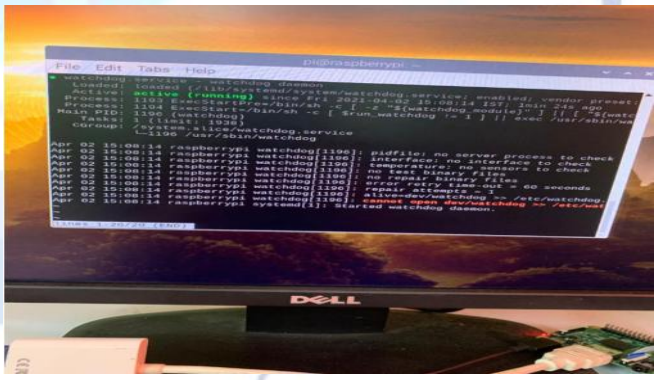


Fig 10 Watchdog

Now we ran the code and got an output as shown:

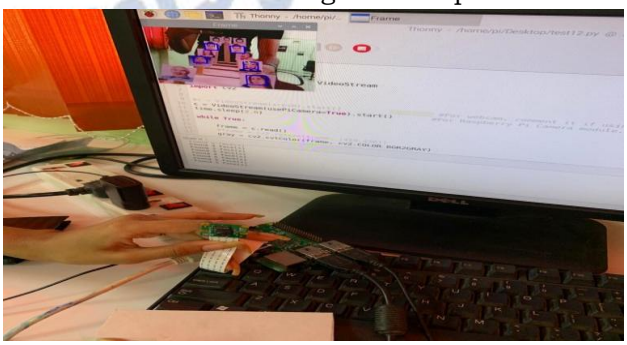


Fig 11 Final Output

We got a much better output where out of 9 images we were able to detect 7 images at once.
Only 2 images were not detectable as one was making a very odd face and the other image was I think too far for the camera.
Since we have trained the dataset for many images we can find out the emotions of a person through a test code by comparing it with the vast number of images from the dataset. Here we have included 7 basic emotions which can be detected such as Anger, Happy, Sad , Anxious , Fear , Disgust and Surprise.

Fig 12 Input test image for emotion detection

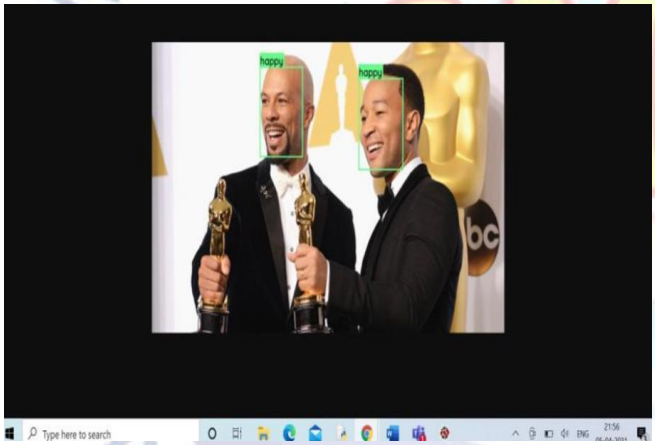


Fig 13 output image for emotion detection

Over here we are trying to test images for Emotion Detection. So after writing the code for emotion detection we input the fig. 12 into the code for testing. So when we run the code we get an output where frames are formed around the faces of individuals present in the image which is depicted by the green rectangles around the faces and the emotion present on the faces are shown. Here both individuals look happy. This is prediction through testing and is only achievable because we had trained our dataset.

VI. CONCLUSION

A new transport system can be developed on the basis of this project which can be implemented in Mumbai as the numbers of passengers are increasing day by day. This system also gives the count of number of passengers travelling in the bus. The implementing of this project will make the life of the common people using public transport much comfortable. The passenger can plan his travel schedule effectively and in turn save time by avoiding the waiting for the bus which can be utilize in a productive manner. The comfort and ease will attract more people to use public transport which will increase the revenue of the government. Due to ease and comfort many people will avoid their own transport and opt for public transport which will reduce the traffic on the road.

ACKNOWLEDGEMENT

We hereby take this opportunity to thank our institution Mukesh Patel School of Technology Management and Engineering for giving us this opportunity.

REFERENCES

[1] Cyrel O., et al. "Real-time integrated CCTV using face and pedestrian detection image processing algorithm for automatic traffic light transitions." 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). IEEE, 2015.

[2] Himanshu Sharma, Sumeet Saurav, Sanjay Singh, Anil K Jain, and Ravi Jain, “Analyzing Impact of Image Scaling Algorithms on Viola-Jones Face Detection Framework”, Advanced Electronics System, Academy of Scientifc and Industrial Research , CSIR-Central Mechatronics Research Institute, 2015.

[3] Guan-Chun Luh , “Face Detection system using Skin Color Pixel Detection and Viola-Jones Face DEtector ”, Department of Mechanical Engineering, Tatung University, 2014

[4] Paul Viola, and Michael Jones, “Rapid Object Detection using a Bosted Cascade of Simple Features”, Mitsubishi Electric Research Labs, Compaq Cambridge Research Lab, 201

[5] K.Sreedhar , and B.Panlal , “Enhancement of Images using Morphological Transformations”, International Journal of Computer Science & Information Technology (IJCSIT), Vol 4, No 1, p. 3-50, Feb 2012