A Project Report

On

REAL TIME OBJECT DETECTION FOR EFFECTIVE INTELLIGENT TRANSPORT SYSTEM USING RASPBERRY PI

Submitted in partial fulfillment for the award of the degree

of

Bachelor of Technology

in

Electronics and Communication Engineering

by

THUPAKULA JAANVI	19F61A0468
MOOLINTI JITHENDRA	19F61A0476
PINJARI KHAJA	19F61A0483
V K KARTHIK	19F61A0482
CHENNIKAYALA GURUPRASAD	19F61A0455

Under the esteemed guidance of

Mrs. D. SAKUNTHALA M.Tech., (Ph.D)

Assistant Professor, Department of ECE

Department of Electronics and Communication Engineering

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE & Affiliated to JNTUA, Ananthapuramu)

(Accredited by NBA for Civil, EEE, ECE, MECH and CSE, New Delhi)

(Accredited by NAAC with ‘A+’ Grade, an ISO 9001:2008 Certified Institution)

Siddharth Nagar, Narayanavanam road, Puttur-517583, A.P

2023

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE & Affiliated to JNTUA, Ananthapuramu)
(Accredited by NBA for Civil, EEE, ECE, MECH and CSE, New Delhi)
(Accredited by NAAC with 'A+' Grade, an ISO 9001:2008 Certified Institution)
Siddharth Nagar, Narayanavanam road, Puttur-517583, A.P

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Project entitled " REAL TIME OBJECT DETECTION FOR EFFECTIVE INTELLIGENT TRANSPORT SYSTEM USING RASPBERRY PI" that is being submitted by

THUPAKULA JAANVI	**19F61A0468**
MOOLINTI JITHENDRA	**19F61A0476**
PINJARI KHAJA	**19F61A0483**
V K KARTHIK	**19F61A0482**
CHENNIKAYALA GURUPRASAD	**19F61A0455**

is in partial fulfillment of the requirements for the award of BACHELOR OF *TECHNOLOGY in* ELECTRONICS AND COMMUNICATION ENGINEERING *to* JNTUA , ANANTHAPURAMU. *The results embodied in this Project report have not been submitted to any other University or Institute for the award of any degree.*

Internal Guide **Head of the Department**

Submitted for the project report viva-voce examination held on ______________

Internal Examiner **External Examiner**

Acknowledgement

We wish to express our profound and sincere gratitude to Mrs. ***D. Sakunthala,*** *Assistant Professor of Electronics and Communication Engineering,* ***Siddharth Institute of Engineering & Technology, Puttur****, who guided us into the intricacies of this project with utmost clarity.*

We would also like to extend our gratitude to ***Dr. P Ratna Kamala****, Head of the Department of Electronics and Communication Engineering for her encouragement and for providing the facilities to carry out the work in a successful manner.*

We are thankful to ***Dr. K. Chandrasekhar Reddy****, Principal for his encouragement and support.*

We wish to express our sincere thanks to ***Dr. K. Indiraveni,Vice-Chairman,*** *and* ***Dr. K. Ashok Raju, Chairman*** *of Siddharth Group of Institutions, Puttur, for providing ample facilities to complete the project work.*

We would also like to thank all the faculty and staff of the Electronics and Communication Engineering Department, for helping us to complete the project work.

Very importantly, we would like to place on record our profound indebtedness to our parents and families for their substantial moral support and encouragement given throughout our studies.

TABLE OF CONTENTS

CHAPTER NO.	NAME	PAGE NO.
	ABSTRACT	**i**
	LIST OF FIGURES	**ii**
	LIST OF TABLES	**iii**
CHAPTER 1	**INTRODUCTION**	**1-4**
	1.1 Existing System	**4**
CHAPTER 2	**LITERATURE SURVEY**	**5-6**
CHAPTER 3	**PROPOSED SYSTEM**	**7-9**
	3.1 Proposed System	**7**
	3.2 Deep Learning System Model	**8**
	3.3 Block Diagram	**9**
CHAPTER 4	**HARDWARE DESCRIPTION**	**10-12**
	4.1 Raspberry Pi	**10**
	4.2 Web Camera	**11**
	4.3 Speaker	**11**
	4.4 Power Supply	**12**
CHAPTER 5	**SOFTWARE DESCRIPTION**	**13-14**
	5.1 Python Software	**13**
	5.2 Open CV	**13-14**
	5.3 Tensor Flow	**14**
CHAPTER 6	**APPLICATIONS**	**15**
CHAPTER 7	**RESULTS**	**16-17**
CHAPTER 8	**8.1 CONCLUSION**	**18**
	8.2 FUTURE SCORE	**18**
	REFERENCES	**19-21**
ANNEXURE – A	**SOURCE CODE**	
ANNEXURE – B	**PROJECT BUDGET**	
ANNEXURE – C	**JOURNAL CERTIFICATIONS**	

ABSTRACT

Intelligent transport system(ITS) is not just limited to traffic congestion control and information, but also for road safety and efficient infrastructure usage. The problem related to traffic is a complex one requiring proper design and planning for developing a solution .For safety and harmony in the flow of traffic ,drivers are expected to pay attention to identify,interpret and follow certain rules while driving. Misinterpretation of traffic signs signals lead to catastrophes. An automatic system in a car which detects,recognizes.interprets traffic signs,traffic signals and pedestrians effectively and gives warning to the driver would be of great help in reducing the misinterpretations and accidents.In the present study,traffic sign and signal recognition system has been developed to increase the safety of the road users by installing the developed hardware system inside the car which gives an audio output for alerting the driver.Tensor Flow algorithm was used for the real time object detection through deep learning due to its high accuracy.The algorithm is embedded into Raspberry Pi 4 prototype for processing and analysis to detect the traffic signs,signals and pedestrians from the real time video which will be recorded by a camera placed in the system and produces an audio output to alert the driver. This work aims to study the accuracy,dealy and reliability of the developed system using a Raspberry Pi 4 processor considering several scenarios related to the state of the situation and the condition of the traffic signs and movement of humans. A real-time tested hardware implementation has been conducted for different classes of traffic signs,traffic signals and pedestrians. The result showed more than 98% accuracy and is reliable with an acceptable delay.

Keywords:Open CV,Tensor flow,Rapberri pi 4

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
Fig 3.1	Block Diagram of Proposed Method	9
Fig 4.1	Raspberry Pi Controller	10
Fig 4.2	Web Camera	11
Fig 4.3	Speaker	12
Fig 7.1	Object Detection	17
Fig 7.2	Object Detection Unit	17

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
Table 1.1	Mandatory Signs	2
Table 1.2	Some Examples Of Cautionary Road Signs	3
Table 1.3	Some Examples Of Informatory Road Signs	3

CHAPTER 1

INTRODUCTION

Road accidents are multi-causal and are often the result of interplay of various factors like driver not able to identify traffic signs and signals in new routes, misinterpretation of pedestrian movement, over speed, use of mobiles, jumping signals etc. Every person, whether a passenger, driver, pedestrian would have noticed various sign boards and signals along the roadside that serve important purposes. These important road paraphernalia help us as route guides, warnings and traffic regulators. As control devices for traffic signs and signals it needs full attention and appropriate driver's response. These road Signs were around us date long back in history and the earliest road signs were milestones, giving distance or direction. In the middle ages, multidirectional signs at intersections became common, giving directions to cities and towns. With the advent of motorized traffic and its increasing pressure on road, many have adopted pictorial signs and standardized their signs to facilitate international travel, where language differences would create barriers. In general it is used to help enhance traffic safety through appropriate caution, regulation and informatory signs. Most of them use symbols in place of words and have international recognition and acceptance. These signs were primarily evolved in Europe, and have been adapted by most countries. In India, Motor Vehicle Act 1988 has laid down the uniform road Signs in its Schedule I which comprehensively explains the shape and sizes of these road signs. Article 5 of Chapter II of the Convention on Road Signs and Signals held on 8th November 1968 lays down the classes of Road Signs, which were broadly categorized into: a) Mandatory signs b) Cautionary signs c) Informatory signs. A further guide to the function of a sign is its color. Mandatory Road signs are generally round in shape with red border and some in Blue circles which gives a mandatory instruction such as "Compulsory Turn Left" etc. Blue Rectangles are used for information signs. All triangular signs are red. There are few exceptions to the shape and color rules, to give certain sign greater prominence. Examples are the "STOP" and "GIVE WAY"" that are octagonal and triangular, in shape. These signs are obligatory on the traffic which is used by a specific area of road that indicates what must one do, rather than must not do. Violation of these signs attracts heavy fines and punishments. Importantly, violation of these could lead to major accidents also. This Section gives few road sign examples for Mandatory, Cautionary and Informatory categories and a brief description of each sign is presented (Table no 1).

S.N	Sign	Meaning
1		The Most important and prominent road sign. This sign indicates that driver should immediately stop. Usually Police, Traffic and Toll Authorities use this sign at check posts.
2		This sign is used at roundabouts where a specific lane discipline is to be followed. This sign directs the traffic to give way to the fellow traffic on the right side of the driver.
3		This sign notifies that entry is prohibited for all vehicles. Certain portions of an area or road are demarcated as 'NO ENTRY' areas for traffic. This could be entry to a restricted area or no traffic zone. So the driver should obey it and divert his route.
4		This sign designates the speed of traffic on road. The Limit specified must be invariably followed to avoid penal action and accidents on the road.
5		This sign indicates that the driver should drive in left lane for smooth traffic flow. The sign is installed mainly on the roads which do not have divider in between any two way traffic flows on the same road.

TABLE 1.1-MANDATORY SIGNS

1.1-Traffic Signs:

1.1.1 Cautionary Signs:

These signs are meant to caution the driver about the hazards/situation lying ahead on the road. The driver should obey these for his safety. Though violation of these road signs do not attract any legal action, they are very important for the fact that avoiding them could result in major accidents. Cautionary signs are triangular in shape with red border.

Sl. No.	Sign	Meaning
1.		This sign cautions about a sharp left turn on the road ahead. These are essentially erected on hilly roads. It gives time to reduce the speed to manage the turn and also sets eyes of the driver on turn.
2.		This road sign indicates that ther is stee ascent ahead and driver should get ready to climb and put the vehicle in relevant gear. Most of the times, these signs are found on hilly road where steep ascents and descents are normal part of travel.

TABLE 1.1.1 - SOME EXAMPLES OF CAUTIONARY ROAD SIGNS

1.2 Informatory signs:

These signs are meant to provide information on direction, destination, roadside facilities, etc. to the road user. Following informative road signs helps a driver in saving time, reaching destination without wandering around. These signs are generally facilitators to the driver and signs are normally blue in color. The sign may have direction arrow and also the distance facility from the sign.

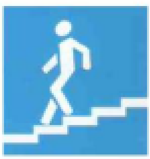
Sl. No.	Sign	Meaning
1.		This informatory sign indicates that there is a Petrol Pump ahead. Sometimes distance is also indicated on this sign which gives an idea about its location from the sign post.
2.		This sign indicates that there is Hospital nearby. The driver should be careful while driving through this stretch and should not honk unnecessarily.
3.		This sign indicates entry to a pedestrian underpass/subway. Pedestrians should invariably use these underpass/subway to cross the road.

TABLE 1.2.1 - SOME EXAMPLES OF INFORMATORY ROAD SIGNS

1.1 Existing System :

In the existing system, These systems use cameras mounted on vehicles or at fixed locations to detect objects such as vehicles, pedestrians, and cyclists. The images captured by the cameras are analyzed using computer vision algorithms to identify and track objects in real-time.It is very low accuracy.

1.1.1 *Drawbacks:*

- Real-time object detection systems may not always accurately detect and track objects, especially in challenging weather conditions or complex environments.
- The range of some object detection systems may be limited, which can be a problem for detecting objects that are far away from the sensor.
- Object detection systems require regular maintenance and calibration to ensure they are functioning properly. This can be time-consuming and costly.

CHAPTER 2

LITERATURE SURVEY

2.1 T. H. M. A. Y. K. K. K. Isamu Takai, "Optical Vehicleto-Vehicle Communication System Using LED Transmitter and Camera Receiver," IEEE Photonics Journal, Vol. 6, No. 5, pp. 7902513-7902513; October 2014.

T. H. M. A. Y. K. K. K. Isamu Takai et.al proposed a IEEE 802.11p describe how communication takes place with an individual DSRC spectrum channel, which imposes a new section set of requirement on communication system by introducing operating mode of WAVE and IEEE 802.11 in BSS. Advantages of multichannel operations, advanced security and other applications on upper layer.

2.2 H. Elgala, R. Mesleh, and H. Haas, "Indoor Broadcasting via White LEDs and OFDM," IEEETrans. On Consumer Electronics, Vol. 55, No. 3, pp. 1127-1134, Aug. 2009.

H. Elgala, R. Mesleh, and H. Haas et.al proposed a promising technology for vehicular communication for safety measures, evaluation of the performance impact of varying channel conditions done, impacts of energy efficient packet error rate, rate of collision and successful packet transmission with respect to throughput performance.

2.3 J. Zhang et al., "Data-Driven Intelligent Transportation Systems: A Survey". IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624-1639, 2011.

J. Zhang et.al proposed in capable of tackling the severe interferences present in the open based road to developing wireless technology has properly chosen which is a complete DSRC system for Intelligent System using combined digital technology along with reception diversity like spread spectrum. This type of DSRC system is operational in its basic form with several mobile users over a distance of 500 meters and for more efficiency radio channel is categorized. A new method in, which describes cloud-based computing in traffic managing system for metropolitan areas, thus increasing the performance, travelers safety, and to reduce consumption of energy. For routing geographical addressing and cloud-based service discovery mechanism used, and throughput improves by this method.

2.4 Papadimitratos,A.LaFortelle ,K .Evenssen, R.Bringolo,S.Cosenza, ”Vehicular communication systems: Enabling technologies, applications , and future outlook on intelligent transportation “in IEEE Comm.Mag.,vo 47,no.11,pp.84-95,Nov.2009.

Papadimitratos,A.LaFortelle ,K .Evenssen, R.Bringolo,S.Cosenza et.al proposed paper[30] widely developed cellular network along with communications having device to device(D2D), is a promising technique to support reliable and efficient vehicular communications. Power allocation and Spectrum sharing concentrated on slowly varying information largescale fading of channels. Across all V2I links uniform capacity performance is available and its major drawback is high mobility. Vehicular networks like VANET which is one of the wireless networks used for vehicular communications on roads.

2.5 W. Jia-yuan, Z. Nian-yu, W. Dong, I. Kentaro, I. Zensei and N. Yoshinori, “Experimental study on visible light Communication based on LED,” The Journal of China Universities of Posts and Telecommunications, Vol.19, No. 2, pp. 197 200, October 2012.

W. Jia-yuan, Z. Nian-yu, W. Dong, I. Kentaro, I. Zensei and N. Yoshinori et.al proposed this has more reliability but drawback is conventional routing is not possible. But in [29] overcome the demerit thus by determine reliable routes for this mechanism to find vehicle information from the source vehicle to destination, but for this latency is a problem which is low.

CHAPTER 3

PROPOSED SYSTEM

3.1 Proposed System

Raspberry Pi is a mini-computer that is capable of running applications as a computer [28]. It is one of the most popular single-board controllers and is affordable with number of resources available. In the present study, Raspberry Pi 4 module is used to run the Tensor Flow algorithm and is connected to a camera mounted at the mirror level of the car to record the real-time video. A Raspberry Pi display module displays the related information of the detected traffic signs, signals, and the pedestrians if they suddenly come across the road and a speaker module to give an audio output as an alert to the driver. Fig.1 shows the complete hardware of the developed system. The system will be deployed in a car on the dashboard and is powered using power car adapter.

LabelImg also known as label image, which is an open-source image labelling tool that is used to label the size of the traffic signs and signals image dataset. Each traffic sign and signal image from the video recording needs to be labelled before starting the training process with TensorFlow software library. The purpose of labelling the images is to identify the traffic signs, traffic signals and pedestrians precisely. LabelImg carries out the process of segmentation of the images and then continues with the process of annotation and interpretation for the each class ofl images. Annotation of these images is important as the bounding boxes will be shown on the traffic sign, traffic signal and pedestrian images, which thus can be easily recognized, as shown in Figure 6. The coding used for LabelImg is Python programming language. After LabelImg annotates the sample set of traffic sign, traffic signal and pedestrian images, they are saved in 'XML' file format, and are ready to be trained and tested by TensorFlow models.

As stated above, in this work, the TensorFlow algorithm is used to train a dataset consisting of five different classes of traffic signs, traffic signals and pedestrians including the Stop Sign, Speed Limit 25,Speed Limit 40, Traffic signals(Red and Green) and pedestrians (Humans). These five classes of objects are considered as they are commonly found on the road and most of them are not correctly interpreted from a distance due to various factors which leads to mishap. For each class, there are 100 sample images with different angles and size, thus give a total of 500 sample images. The training process which is crucial in deep learning model is important to prepare the pre-trained model before the implementation of the real-time recognition system. The pre-trained model segments the size, colour, shape and boundary of the sample traffic regulator and pedestrian images that is used for the recognition.

3.2 Deep Learning System Model

As mentioned above, deep learning can be introduced to improve system performance for typical classification problem. Currently, CNN's (Convolution Neural Network) are used with good prospects in the field of deep learning as they are pre-trained networks and can be used for image classification which shows better performance compared with traditional approaches also good in feature extraction. CNN models are based on convolution, activation, pooling, and full-connected neural networks which consequently recognizes the significant features with next to no human management. Among them, convolution is an effective method to extract object features [4], while pooling is used to reduce the dimension of features [5]. The MobileNet-v2 model created by Google often called as light weight neural network which can solve the classification problem quite well. Hence the proposed framework in this paper is mainly based on this model. In our model, multiple captured images captured through a real-time video pass through three convolution layers and an average pooling layer followed by batch normalization. The activation function used here will be ReLU6.Then the extracted features are fed into a full-connected layer using softmax activation function. Finally, three different types of output results are obtained. The model has three modules: pre-processing images, forward propagation and back propagation. Pre-processing images mainly includes decoding, resizing, and standardizing. Forward propagation defines the detailed CNN structure. Back propagation is responsible for training and optimizing network parameters, while improving the accuracy of the model. All weights in our model are initialized by the normal function and the biases are initialized as zero.

3.3 Block Diagram

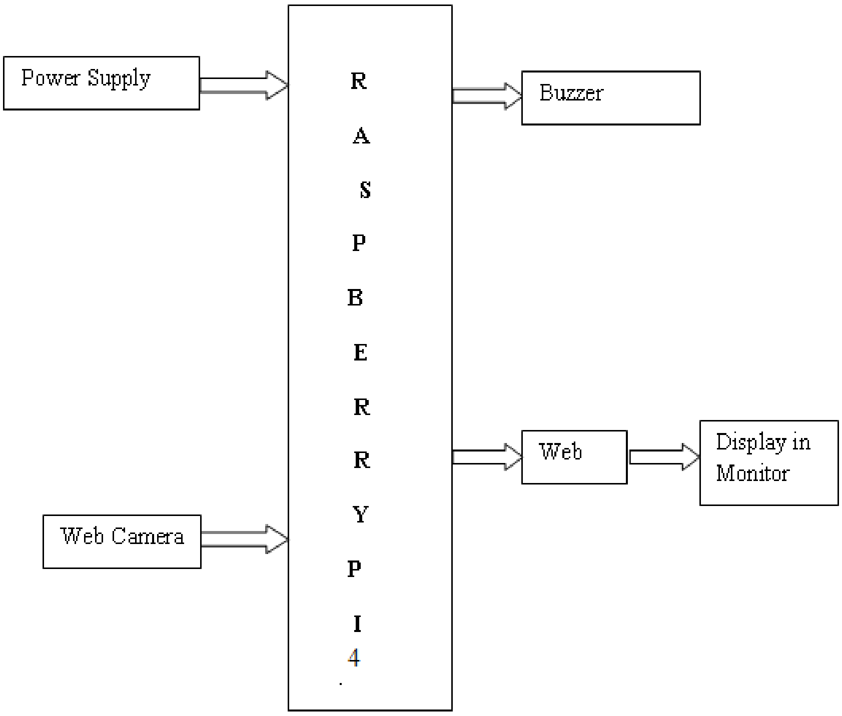


Figure 3.3: Block Diagram of Proposed Method

For autonomous vehicles, the complex processing of video images is one of the main hurdles which can be possible by using our Raspberry pi 4 module. It is portable, easy to integrate and needs 5volts of power supply that can get from battery supply of the vehicle. It has 40 IO related pins and supports all protocols mainly CAN protocol which is used mainly in automobiles. It also supports I2C, serial, Wi-Fi Ethernet, USB. In the present work, the main objective is to build intelligent transport system with effective traffic recognition. The raspberry system is capable of recognizing multiple objects through a camera by plugging a webcam into one of the Raspberry Pi 4’s USB slots as in Fig.1. The webcamera which is connected through USB is configured by selecting the port to read the image from the video stream by using Tensorflow and OPENCV models. After detecting the images it produces a voice output of the detected sign which uses eSPEAK software to synthesize speech as an alert message.

CHAPTER 4

HARDWRE DESCRIPTION

4.1 Raspberry Pi

Raspberry pi is a small chip of single board computer. There are various model of raspberry available in the market i.e. the Raspberry Pi1 Model B, Raspberry Pi1 Model B+, Raspberry pi2, Raspberry Pi3 Model B. These all are differ in memory capacity and hardware features like Raspberry pi3 has inbuilt Bluetooth and Wi-Fi modules whereas in previous versions these modules were not available .It has 1.2 GHz 64-bit quad core ARMv8 CPU with 1 GB of RAM as displayed in the Fig.3.

FIG.4.1.1: RASPBERRY PI CONTROLLER

The Foundation provides Raspier, a Debi an-based Linux distribution for download, as well as third-party Ubuntu, Windows 10 IOT Core, RISC OS, and specialized media center distributions. It promotes Python and Scratch as the main programming language, with support for many other languages.The default firmware is closed source, while an unofficial open-source is available

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support andshown in Figure 4.1.1

4.2. Web Camera

Web Camera is used to take the continuous images to get the traffic signs and signals from the real world that looks like in Fig.4. According to the images available through the camera we can send these images to the raspberry pi to perform car's control action.

Fig.4.2.1: Web Camera

A webcam is a video camera that feeds or streams its image in real-time to or through a computer to a computer network. When "captured" by the computer, the video stream may be saved, viewed, or sent on to other networks via systems such as the internet, and emailed as an attachment. When sent to a remote location, the video stream may be saved, viewed, or sent there. Unlike an IP camera (which connects using Ethernet or Wi-Fi), a webcam is generally connected by a USB cable, or similar cable, or built into computer hardware, such as laptops.

4.3. Speaker

Speakers are one of the most common output devices used with computer systems and the most common look like Fig.5. Some speakers are designed to work specifically with computers, while others can be hooked up to any type of sound system. Regardless of their design, the purpose of speakers is to produce audio output that can be heard by the listener.

Speakers are one of the most common output devices used with computer systems. Some speakers are designed to work specifically with computers, while others can be hooked up to any type of sound system. Regardless of their design, the purpose of speakers is to produce audio output that can be heard by the listener.

Fig.4.3.1: Speaker

4.4 Power Supply

The information to the circuit is related to the arranged power supply. The A.C information i.e., 230v from the mains gives wind around the transformer to 12v and is lively to a rectifier.

The yield grabbed from the rectifier is a sore D.C voltage. Remembering the true objective to get an unmodified DC voltage, the yield voltage from the rectifier is supported by a channel to leave any AC parts to introduce widely after refresh. Inevitably, this voltage is given to a voltage controller to get an immaculate unfaltering dc voltage.

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 Python Software

Python is a high level programming language used widely in industries and research work. Different versions of python IDLE is available for programming the python language.

5.2 Open CV

Open CV is an open-source computer vision library. The library is written in C and C++ and runs under Linux, Windows, and Mac OS X. There is active development on interfaces for Python, Ruby, MATLAB, and other languages.

Open CV was designed for computational efficiency and with a strong focus on real-time applications. Open CV is written in optimized C and can take advantage of multi-core processors. If you desire further automatic optimization on Intel architectures [Intel], we can buy Intel's Integrated Performance Primitives (IPP) libraries [IPP], which consist of low-level optimized routines in many different algorithmic areas. Open CV automatically uses the appropriate IPP library at runtime if that library is installed. One of Open CV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The Open CV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. Because computer vision and machine learning often go hand-in-hand, Open CV also contains a full, general-purpose Machine Learning Library (MLL).

This sub-library is focused on statistical pattern recognition and clustering. The MLL is highly useful for the visual tasks that are at the core of OpenCV's mission, but it is general enough to be used for any machine learning problem.

The Open CV-Library is mainly aimed at real-time computer vision. Some example areas would be Human-Computer Interaction (HCI); Object Identification, Segmentation, and Recognition; Face Recognition; Gesture Recognition; Motion Tracking, Ego Motion, and Motion Understanding; Structure from Motion (SFM); and Mobile Robotics. The Open CV Library software package supports many functions whose performance can be significantly enhanced on the Intel architecture (IA), particularly. The Open CV-Library is a collection of low-overhead, high-performance operations performed on images. This manual explains the Open CV Library concepts as well as specific data type definitions and operation models used in the image processing domain. The manual also provides detailed descriptions of the functions included in the Open CV Library software.

It stands for Open Source Computer Vision. It has a library of programming function mainly for real time computer visions. It has over more than 2500 optimize algorithms for set of classical algorithm as well as for the state of art algorithms in the computer visions. It is basically used for image processing in which in the present study it is used for the face detection, object detections, image recognition, traces and also for other functions.

5.3 TensorFlow

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of Tensor Flow.

TensorFlow is an open-source library for fast numerical computing. It was created and is maintained by Google and was released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API.

Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems, not least of which is RankBrain in Google search and the fun DeepDream project.

It can run on single CPU systems and GPUs, as well as mobile devices and large-scale distributed systems of hundreds of machines.

5.3.1 How to Install TensorFlow

Installation of TensorFlow is straightforward if you already have a Python SciPy environment.

TensorFlow works with Python 3.3+. You can follow the on the TensorFlow website. Installation is probably simplest via PyPI, and specific instructions of the pip command to use for your Linux or Mac OS X platform are on the Download and Setup webpage. In the simplest case, you just need to enter the following in your command line.

CHAPETR 6

APPLICATIONS

6.1 Vehicle and Pedestrian Detection:

Real-time object detection can help detect vehicles, pedestrians, and other objects on the road, providing drivers with real-time information about the traffic conditions ahead. This can help improve road safety and reduce the risk of accidents.

6.2 Traffic Management:

Real-time object detection can be used to monitor traffic flow and manage traffic congestion. The system can provide information about the number of vehicles on the road and their speed, allowing authorities to make informed decisions about traffic management.

6.3 Parking Management:

Real-time object detection can help detect available parking spaces and provide information to drivers. This can reduce the time drivers spend looking for parking spaces, leading to reduced traffic congestion and improved air quality.

6.4 Road Maintenance:

Real-time object detection can help detect potholes, cracks, and other road defects, allowing authorities to take corrective measures before they become major safety hazards.

6.5 Autonomous Vehicles:

Real-time object detection can be used in autonomous vehicles to detect objects on the road and make decisions based on that information. This can help improve the safety of autonomous vehicles and reduce the risk of accidents.

CHAPTER 7

RESULTS

In the present work, we have a assumed a car with the prototype developed which is placed at the dashboard of the vehicle moving ahead and the images are captured and are divided into frames through a real time video stream recorded from the camera mounted. The Raspberry pi is used to read the frames with a minimum resolution of the pixels and frame rate set and the FPS rate can be increased using python and Open CV. The FPS is calculated by using the formula

time1 = (t2-t1)/freq

frame_rate_calc= 1/time1

For the object detection, Tensorflow model is used for training and testing where it will grab the frame from the video and resize it to expected shape. It will perform the actual detection by running the model with this image as input and retrieve the detection results. It will loop over all the directions and draw a detection box if confidence is above the minimum threshold. The object detected will be shown with a bounding box labelled with object name and accuracy of the detection in terms of percentage as shown in the figures Fig.6,7,8,9. A real-time alert system is developed using the Raspberry Pi 4 processor. The MobileNetv2 pre-trained model is used to train the system to detect certain traffic signs, traffic signals and pedestrians which are captured through the real-time recording video using Raspberry Pi camera to alert the drivers. Five different classes of images that includes Stop Sign, Speed Limit 25, Speed Limit 40, Traffic signals (Red and Green) and pedestrians (Humans) have been considered for the real-time tested implementation. The performance of the developed system has been evaluated in terms of accuracy, delay and reliability. The results show that the good performance of detecting and identifying traffic sign images which takes 4 images per second and compares the test data for object detection with 1 frames detection per second.

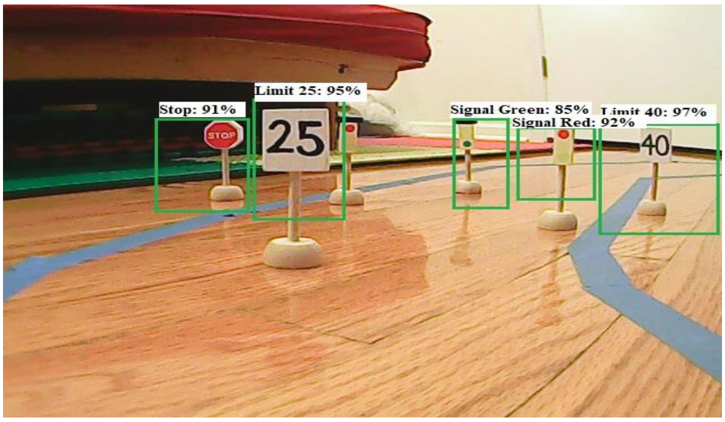


FIG.7.1: OBJECT DETECTION

FIG.7.2: OBJECT DETECTION UNIT

CHAPTER 8

8.1 Conclusion

This project's implementation was focused on traffic signs, traffic signals and pedestrian detection, thereby alerting the driver through voice message. The work described in this paper is based on Real time object detection system using Raspberry Pi hardware similar to other applications in the field. For the detection part, Tensor Flow and CNN model MobileNetV2 was used. Experimental results run on these classes of objects shows that proposed structure works on the detecting, identifying and locating traffic signs, signals and pedestrians with better performance and reliability under driving circumstances and meets the continuous prerequisite of a high level advanced driver assistant system.

8.2 Future Scope

VLC has extraordinary potential for future ITS environments, as deep learning technique can be acquainted with it to get to the next level framework execution. A real time VLC prototype using Raspberry Pi for detection of traffic sign based on light communication for controlling the vehicle's status and also transmitting the safety messages to the vehicle at the back with an alerting system. We carried out experimental measurements in terms of the eye diagram, beam profile at different link spans and light intensity of LED array for a range of offset angles in between two vehicles for the successful communication.

REFERENCES

1. Qinghua Xiang, Yingxiu Li, Wenting Huang, Wenxuan Ye, and Jianhua Shen," Deep Learning assisted Visible Light Communication based Intelligent Transport System", 2019 18th International Conference on Optical Communications and Networks.
2. W.-L. Jin, "SPIVC: A Smartphone-based inter-vehicle communication system," Proceedings of Transportation Research Board Annual Meeting, 2012.
3. A. Boukerche et al., "Vehicular Ad Hoc Networks: a new challenge for localization-based systems," Computer Communications, ScienceDirect, pp. 1-12, 2008.
4. N. M. Husain Fidvi, "Car to Car Communication System," source:car communication system,[AvailableOnline:http://www.engineersgara ge.com/contribution/car-to-carcommunicationsystem?page=1]
5. FCC,[AvailableOnline:http:/www.fcc.gov/Bureaus/E ngineeringTechnology/News_Releases/1999/nret9 00 6.html], October, 1999.
6. T. H. M. A. Y. K. K. K. Isamu Takai, "Optical Vehicleto-Vehicle Communication System Using LED Transmitter and Camera Receiver," IEEE Photonics Journal, Vol. 6, No. 5, pp. 7902513-7902513; October 2014.
7. Haoui, A., R. Kavaler and P. Varaiya, 2008. Wireless magnetic sensors for traffic surveillance. Transportation Research Part C: Emerging Technologies, 16(3): 294-306.
8. H. Elgala, R. Mesleh, and H. Haas, "Indoor Broadcasting via White LEDs and OFDM," IEEETrans. On Consumer Electronics, Vol. 55, No. 3, pp. 1127-1134, Aug. 2009.
9. W. Jia-yuan, Z. Nian-yu, W. Dong, I. Kentaro, I. Zensei and N. Yoshinori, "Experimental study on visible light Communication based on LED," The Journal of China Universities of Posts and Telecommunications, Vol.19, No. 2, pp. 197 200, October 2012.
10. 10.H. Elgala, R. Mesleh, H. Haas and B. Pricope, "OFDM Visible Light Wireless Communication Based on VLSI",2013
11. 11.Yiyan, W., et al. 2009. Video Image Vehicle Detection System for Signaled Traffic Intersection. in Hybrid Intelligent Systems, HIS '09. Ninth International Conference on. 2009.
12. 12.White LEDs," In the Vehiclar Technology Conference Proceeding, pp. 2185-2189, 22-25, April, 2007.

13. 13.N. Lourenco et. al, “Visible Light Communication System for Outdoor Applications,” In the 8th International Symposium on Communication Systems, Networks and Digital Signal Processing, pp. 1-6.18-20 July 2012.

14. 14.W.-L. Jin, “SPIVC: A Smartphone-based inter-vehicle communication system,” Proceedings of Transportation Research Board Annual Meeting, 2012.

15. A. Boukerche et al., “Vehicular Ad Hoc Networks: a new challenge for localization-based systems,” Computer Communications, Science Direct, pp. 1-12, 2008.

16. W.H. Organization.(June 2011).Fact Sheet 310 – The 10 causes of death.Availablehttp://www.who.int/mediacentre/f actsheets/fs310/en/W .H. Organization.(September 2011). Fact Sheet 358 Road Traffic Injuries Available: http://www.who.int/mediacentre/factsheets/fs358 /en/ .

17. N. M. Husain Fidvi, “Car to Car Communication System,” source: carcommunicationsystem,AvailableOnlinehttp://w ww.engineersgarage.com/c Contribution/car-tocar-communication-system? Page=]

18. M. Alsabaan, K. Naik,T. Khalifa,A. Nayak, ”Vehicular networks for reduction of fuel consumption and CO2 emission”, Industrial Informatics (INDIN), 2010 8th IEEE International Conference on , pp.671-676, 13-16 July 2010.

19. B.Zhou, J.Cao,H.Wu,Adaptive traffic light control of multiple intersections in WSN-based ITS”,978-1- 4244-8331,IEEE,2011.

20. 3rd Generation Partnership Project: Technical Specification Group Radio Access Network; Study on LTE-based V2X Services (Release 14), 3GPP TR 36.885 V2.0.0, Jun.2016.

21. Scenarios, requirements and KPIs for 5G mobile and wireless system,METISICT317669METIS/D1.1,Apr2013.[Online].Available:https://www.metis2020.com/docum ents/deliverables/.

22. .Papadimitratos,A.LaFortelle ,K .Evenssen, R.Bringolo,S.Cosenza, ”Vehicular communication systems: Enabling technologies, applications , and future outlook on intelligent transportation “in IEEE Comm.Mag.,vo 47,no.11,pp.84-95,Nov.2009.

23. B. Williams, “Intelligent Transport Systems Standards”, Ed. London: Artech House, 2008.

24. J. Harding et al., “Vehicle-to-vehicle communications: readiness of V2V technology for application,” U.S. National Highway Traffic Safety Administration (NHTSA), Tech. Rep. DOT HS 812 014, 2014.

25. J. Zhang et al., “Data-Driven Intelligent Transportation Systems: A Survey”. IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624-1639, 2011.

26. Luo, Q., 2008 .Research on” Intelligent Transportation System Technologies and Applications” in Power Electronics and Intelligent Transportation System, PEITS '08. Workshop on. 2008.

27. Wang, W. and K. Bengler, 2011.”Computational Intelligence for Transportation: Driving Safety and Assistance” International Journal of Computational Intelligence Systems, 4(3): 286-286.

28. Wu, Q. and R. Lan, 2010. The Development of Computational Technology and Its Use in Finance. In Education Technology and Computer Science (ETCS), Second International Workshop on. 2010

29. Messelodi, S., et al., 2009. Intelligent extended floating car data collection. Expert Systems with Applications, 36(3, Part 1): 4213-4227.

30. Yiyan, W., et al. 2009. Video Image Vehicle Detection System for Signaled Traffic Intersection. in Hybrid Intelligent Systems, HIS '09. Ninth International Conference on. 2009.

31. IEEE802 Part15.7: "PHY and MAC standard for short-range wireless optical communication using visible light". (Draft 4), December 2010.

ANNEXURE-A

SOURCE CODE

```
import os
import argparse
import cv2
import numpy as np
import sys
import time
from threading import Thread
from subprocess import call
import importlib.util
# Define VideoStream class to handle streaming of video from webcam in separate processing thread
# Source - Adrian Rosebrock, PyImageSearch: https://www.pyimagesearch.com/2015/12/28/increasing-raspberry-pi-fps-with-python-and-opencv/
class VideoStream:
    """Camera object that controls video streaming from the Picamera"""
    def __init__(self,resolution=(640,480),framerate=30):
        # Initialize the PiCamera and the camera image stream
        self.stream = cv2.VideoCapture(0)
        ret = self.stream.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc(*'MJPG'))
        ret = self.stream.set(3,resolution[0])
        ret = self.stream.set(4,resolution[1])
        # Read first frame from the stream
        (self.grabbed, self.frame) = self.stream.read()
            # Variable to control when the camera is stopped
        self.stopped = False
    def start(self):
            # Start the thread that reads frames from the video stream
        Thread(target=self.update,args=()).start()
        return self
```

```
    def update(self):
        # Keep looping indefinitely until the thread is stopped
        while True:
            # If the camera is stopped, stop the thread
            if self.stopped:
                # Close camera resources
                self.stream.release()
                return
            # Otherwise, grab the next frame from the stream
            (self.grabbed, self.frame) = self.stream.read()
    def read(self):
            # Return the most recent frame
        return self.frame
    def stop(self):
            # Indicate that the camera and thread should be stopped
        self.stopped = True
# Define and parse input arguments
parser = argparse.ArgumentParser()
'''parser.add_argument('--modeldir', help='Folder the .tflite file is located in',
                    required=True)
parser.add_argument('--graph', help='Name of the .tflite file, if different than detect.tflite',
                    default='detect.tflite')
parser.add_argument('--labels', help='Name of the labelmap file, if different than labelmap.txt',
                    default='labelmap.txt') '''
parser.add_argument('--threshold', help='Minimum confidence threshold for displaying detected objects',
                    default=0.5)
parser.add_argument('--resolution', help='Desired webcam resolution in WxH. If the webcam does not support the resolution entered, errors may occur.',
                    default='1280x720')
parser.add_argument('--edgetpu', help='Use Coral Edge TPU Accelerator to speed up detection',
                    action='store_true')
args = parser.parse_args()
```

```
MODEL_NAME = 'project/tflite/ok/'
GRAPH_NAME = 'detect.tflite'
LABELMAP_NAME = 'labelmap.txt'
min_conf_threshold = float(args.threshold)
resW, resH = args.resolution.split('x')
imW, imH = int(resW), int(resH)
use_TPU = args.edgetpu
# Import TensorFlow libraries
# If tflite_runtime is installed, import interpreter from tflite_runtime, else import from regular
tensorflow
# If using Coral Edge TPU, import the load_delegate library
pkg = importlib.util.find_spec('tflite_runtime')
if pkg:
    from tflite_runtime.interpreter import Interpreter
    if use_TPU:
        from tflite_runtime.interpreter import load_delegate
else:
    from tensorflow.lite.python.interpreter import Interpreter
    if use_TPU:
        from tensorflow.lite.python.interpreter import load_delegate
# If using Edge TPU, assign filename for Edge TPU model
if use_TPU:
    # If user has specified the name of the .tflite file, use that name, otherwise use default
'edgetpu.tflite'
    if (GRAPH_NAME == 'detect.tflite'):
        GRAPH_NAME = 'edgetpu.tflite'
# Get path to current working directory
CWD_PATH = os.getcwd()
# Path to .tflite file, which contains the model that is used for object detection
PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,GRAPH_NAME)
# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH,MODEL_NAME,LABELMAP_NAME)
```

```
# Load the label map
with open(PATH_TO_LABELS, 'r') as f:
    labels = [line.strip() for line in f.readlines()]
# Have to do a weird fix for label map if using the COCO "starter model" from
# https://www.tensorflow.org/lite/models/object_detection/overview
# First label is '???', which has to be removed.
if labels[0] == '???':
    del(labels[0])
# Load the Tensorflow Lite model.
# If using Edge TPU, use special load_delegate argument
if use_TPU:
    interpreter = Interpreter(model_path=PATH_TO_CKPT,
                              experimental_delegates=[load_delegate('libedgetpu.so.1.0')])
    print(PATH_TO_CKPT)
else:
    interpreter = Interpreter(model_path=PATH_TO_CKPT)
interpreter.allocate_tensors()
# Get model details
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
height = input_details[0]['shape'][1]
width = input_details[0]['shape'][2]
floating_model = (input_details[0]['dtype'] == np.float32)
input_mean = 127.5
input_std = 127.5
# Initialize frame rate calculation
frame_rate_calc = 1
freq = cv2.getTickFrequency()
# Initialize video stream
videostream = VideoStream(resolution=(imW,imH),framerate=30).start()
time.sleep(1)
```

```
#for frame1 in camera.capture_continuous(rawCapture, format="bgr",use_video_port=True):
while True:
    # Start timer (for calculating frame rate)
    t1 = cv2.getTickCount()
    # Grab frame from video stream
    frame1 = videostream.read()
    # Acquire frame and resize to expected shape [1xHxWx3]
    frame = frame1.copy()
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame_resized = cv2.resize(frame_rgb, (width, height))
    input_data = np.expand_dims(frame_resized, axis=0)
    # Normalize pixel values if using a floating model (i.e. if model is non-quantized)
    if floating_model:
        input_data = (np.float32(input_data) - input_mean) / input_std
    # Perform the actual detection by running the model with the image as input
    interpreter.set_tensor(input_details[0]['index'],input_data)
    interpreter.invoke()
    # Retrieve detection results
    boxes = interpreter.get_tensor(output_details[0]['index'])[0] # Bounding box coordinates of detected objects
    classes = interpreter.get_tensor(output_details[1]['index'])[0] # Class index of detected objects
    scores = interpreter.get_tensor(output_details[2]['index'])[0] # Confidence of detected objects
    #num = interpreter.get_tensor(output_details[3]['index'])[0]  # Total number of detected objects (inaccurate and not needed)
    # Loop over all detections and draw detection box if confidence is above minimum threshold
    for i in range(len(scores)):
        if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):
            # Get bounding box coordinates and draw box
            # Interpreter can return coordinates that are outside of image dimensions, need to force them to be within image using max() and min()
            ymin = int(max(1,(boxes[i][0] * imH)))
            xmin = int(max(1,(boxes[i][1] * imW)))
```

```
            ymax = int(min(imH,(boxes[i][2] * imH)))
            xmax = int(min(imW,(boxes[i][3] * imW)))
            cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)
            # Draw label
            object_name = labels[int(classes[i])] # Look up object name from "labels" array using
class index
            label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: 'person: 72%'
            labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
# Get font size
            label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to
top of window
            cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0],
label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # Draw white box to put label text in
            cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX,
0.7, (0, 0, 0), 2) # Draw label text
            speech =object_name
            call(["espeak",speech])
    # Draw framerate in corner of frame
    cv2.putText(frame,'FPS:
{0:.2f}'.format(frame_rate_calc),(30,50),cv2.FONT_HERSHEY_SIMPLEX,1,(255,255,0),2,cv
2.LINE_AA)
    # All the results have been drawn on the frame, so it's time to display it.
    cv2.imshow('Object detector', frame)
    # Calculate framerate
    t2 = cv2.getTickCount()
    time1 = (t2-t1)/freq
    frame_rate_calc= 1/time1
    # Press 'q' to quit
    if cv2.waitKey(1) == ord('q'):
        break
# Clean up
cv2.destroyAllWindows()
videostream.stop()
```

ANNEXURE-B

PROJECT BUDGET DETAILS

SI NO	COMPONENT NAME	COST (₹)
1	RASPBERRY PI 4	8200
2	CAMERA	2300
3	SPEAKER	500
4	ADAPTER AND USB CABLE	700
	TOTAL	11700

ANNEXURE-C

JOURNAL CERTIFICATIONS

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

P D Selvam

In recognition of the publication of the paper entitled

Real Time Object Detection for Effective Intelligent Transport System using Raspberry Pi

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 10 Issue 3 , March-2023 | Date of Publication: 2023-03-27

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2303616 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2303616 Registration ID : 510754

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Tupakula Jaanvi

In recognition of the publication of the paper entitled

Real Time Object Detection for Effective Intelligent Transport System using Raspberry Pi

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 10 Issue 3 , March-2023 | Date of Publication: 2023-03-27

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2303616 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2303616 Registration ID : 510754

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Moolinti Jithendra

In recognition of the publication of the paper entitled

Real Time Object Detection for Effective Intelligent Transport System using Raspberry Pi

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 10 Issue 3 , March-2023 | Date of Publication: 2023-03-27

Parin P
EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2303616 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2303616 Registration ID : 510754

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Pinjari Khaja

In recognition of the publication of the paper entitled

Real Time Object Detection for Effective Intelligent Transport System using Raspberry Pi

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 10 Issue 3 , March-2023 | Date of Publication: 2023-03-27

Parin P
EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2303616 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2303616 Registration ID : 510754

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

V K Karthik

In recognition of the publication of the paper entitled

Real Time Object Detection for Effective Intelligent Transport System using Raspberry Pi

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 10 Issue 3 , March-2023 | Date of Publication: 2023-03-27

Parin P
EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2303616 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2303616** **Registration ID : 510754**

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Chennikayala Guruprasad

In recognition of the publication of the paper entitled

Real Time Object Detection for Effective Intelligent Transport System using Raspberry Pi

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 10 Issue 3 , March-2023 | Date of Publication: 2023-03-27

Parin P
EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2303616 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2303616** **Registration ID : 510754**

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator