International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

INTELLIGENT HELMET DETECTION USING OPENCV AND MACHINE LEARNING

K S P V Siva Lalith[#1], Mokirala Sudhamshu[#2],Gunna Kamal Abhishek[#3], Dr. T. Rama Swamy[#4], Dr. V.Jayaprakasan[#5], Dr. G. Prasad Acharya[#6]

[#1] UG student, ECE, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India
[#2] UG student, ECE, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India
[#3]UG student, ECE, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India
[#4]Associate Professor,Dept.of ECE,Sreenidhi institute of science and technology,Telangana,India
[#5] Professor,Dept.of ECE,Sreenidhi institute of science and technology,Telangana,India
[#6]Associate Professor,Dept.of ECE,Sreenidhi institute of science and technology,Telangana,India

---

Abstract—
In daily life, the role of a helmet is vital for motorists. The human brain is an important organ, which is protected by the skull. So the head is to be protected by a helmet, in case of an accident. From our literature survey we found that in India, the majority of motorists do not wear a helmet. This negligence causes fatal injuries. We want to minimize this risk. Our project uses ML and OPENCV tools for Helmet Detection. In this project we use a camera module to detect the face of the person. The preprocessed input is fed to the Machine Learning model. This model processes it and transports the output to Arduino, which 'outputs' the real-time status of the helmet.

Keywords:- **MachineLearning, OpenCV, Helmet Detection, Arduino.**

I. INTRODUCTION

This paper is about the detection of a helmet. Based on our Literature survey we came to know that, an alarming percentage of motorists lack common traffic knowledge and also do not equip themselves with a helmet. This leads to fatal injuries which could have been preventable. A motorist helmet hence plays a vital role. Similarly the usage of helmets has equal importance in the mines, construction sites etc. The main theme of our project is to minimize or eliminate the fatalities. This can be done by detecting if a motorist is wearing a helmet or not. Since this process is automated, it reduces the workload on Traffic Police who otherwise would need to manually capture the images of such motorists. In our project there is a camera which captures the live-video of the person. This video is pre processed frame by frame using OpenCV tools. Each frame is of the resolution 1920x1080p. Further it is fed to the data models which make use of YOLO machine learning algorithm. The predefined data models will process and generate the output, which provides the real-time status of the helmet. A rectangular box encloses the face of the motorist while providing the status- Helmet or No_Helmet along with the percentage of confidence. Next we make use of Arduino. The sensed output is given to the Arduino. It is processed using PyFirmata software equipped on the Arduino. PyFirmata is used to interface with the Python source code and the Arduino code. For easy monitoring, we configure the Arduino to generate an output in the form of an LED status and Buzzer.. If the input to the Arduino contains the presence of the helmet then there is no sound generated by the buzzer and the LED is off. Otherwise the buzzer generates a beep sound and the LED starts glowing.

II. HARDWARE IMPLEMENTATION

To maximize the utilisation of the system a perfect hardware is to be developed. Every component has many functions. So, as implementers, we need to choose the best and least components which produce our desired function. The below are various components that are used by us.

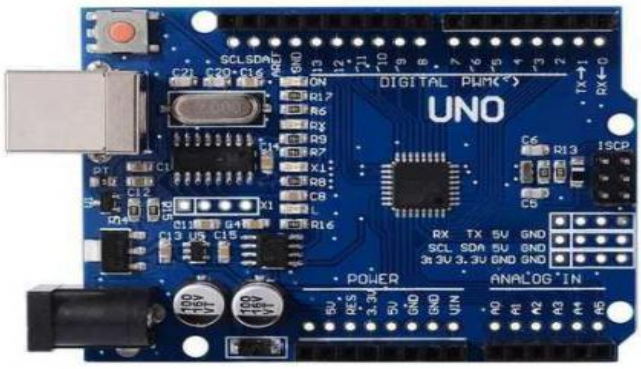
A. Camera Module OV7670

Power is provided from 3.3V supply. It is a low price image sensor. It is a DSP that can operate at a maximum of 30 fps and 640 x 480 resolutions which is equivalent to 0.3 Megapixels. Working Temperature is 0-50 Degree Celsius and All Glass Lenses, Lens is made of Magnesium Alloy material.y. An external oscillator provides the source of the clock for the camera module. In our design the camera continuously captures the video. This video is important data that needs to be analysed. Here video is given to the data model. So the

camera is used for inputting the data to the machine learning model.

ARDUINO UNO

It is based on the ATmega328p microcontroller board. It has 14 digital pins which can be used as input or output and among those 6 pins can be used for PWM outputs. It also has 6 analog pins. A 16MHz Quartz crystal is used for the clock. It has a USB connection, Reset button, ICSP header, And Power jack. It operates with 5V supply. 2KB SRAM, 1KB EEPROM, 32KB Flash memory available with microcontroller. A built in Led connected to the 13th pin. pinMode(), digitalWrite(), digitalRead() are the functions to program the pins as input or output. Serial communication is available from the SoftwareSerial library. I2C and SPI communications are supported by the ATmega328p microcontroller. Arduino supports the serial monitor feature. In this mode serial communication can be achieved.



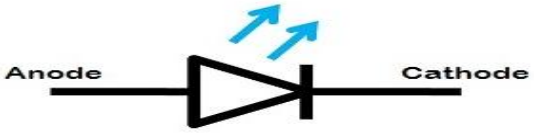
B. Buzzer

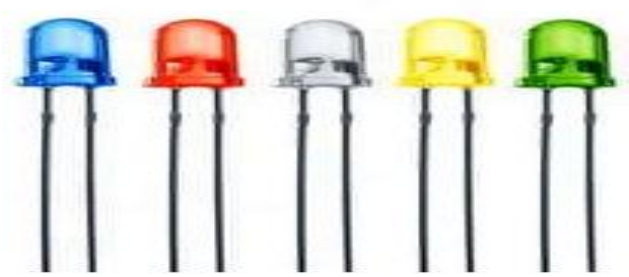
DC voltage is used for powering the buzzer. In general there are various types of buzzers available. Buzzer has two pins as shown in the figure below.

Operating specifications of buzzer are as listed below.
Frequency:- 3Hz to 300 Hz.
Voltage:- 3V to 24 V (DC).
Temperature:- – 20° C to +60°C.
In general we can understand that buzzers are used for alert. In our design also the purpose is similar. The arduino sends the control signal to the buzzer. Based on that signal the sound of the buzzer varies. Actually if there is no helmet detected then the buzzer produces sound. Else no sound from the buzzer.

C. LED

LED's are Indicators. It is a 2 terminal semiconductor device which emits light. The operation of the Led is quite similar to the normal PN diode. That is it conducts in one direction and does not allow current flow in reverse direction. Symbol of LED is given under. Two terminals are anode and cathode. In this design the output from arduino makes LED ON/OFF. For our understanding LED On implies Helmet absence and LED Off implies Helmet presence.





III. SOFTWARE IMPLEMENTATION

A. TENSOR FLOW

It is a machine learning open source platform. The Google Brain team developed the Tensorflow. It provides various levels of abstraction, which helps to build a model easily. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. Provides an easy way to deploy and train models in the Cloud, Servers or devices. Provides solutions to various ML problems like recommendation systems for movies, classification of images like clothes, handwritten text detection etc. TensorFlow provides stable Python and C++ APIs, as well as non-guaranteed backward compatible API for other languages. It is powered by a vast forthcoming community comprising students, developers etc, where you can discuss your ideas and approach solutions to your problems. You can install the Tensorflow package into your system using the 'pip' package available to be used from your Command Prompt.

B. OPEN CV

OpenCV *Open Source Computer Vision Library*. It is a library of programming functions mainly aimed at real-time computer vision. It was originally developed by Intel. It is a huge Open Source cross-platform library for machine learning, computer vision, and image processing. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. It helps to process images, videos and identify objects. OpenCV has a large community of more than 47 thousand users. OpenCV has many applications as listed below:- Counting number of people, Face recognition, Detecting speed of vehicles on the highways, Driver less car controlling,Video/image search. In our project the capturing and preprocessing of the live-video of the motorist is done by the OpenCV tools.

C. DARKFLOW

DarkFlow is a python implementation of YOLOv2 using Tensorflow. A framework is created either- YOLO or YOLOv2. It is an amazing tool that can be used to start detecting common objects in images or videos "out of the box". Next the anchor boxes are created, using cpython which is where python can use C data types and call c functions and API's. If something is wrong with your data or configuration DarkFlow halts the program and displays an appropriate error message.

IV. DESIGN METHODOLOGY

1. Video Capturing

The input to our system is the real-time video capturing of the motorist. This is captured at an FPS rate of 6-12 FPS here. We can boost the FPS by simply making the source code run on the GPU rather than the CPU. Convenient background lighting while capturing the video is appreciated. The video capturing can be turned OFF by exiting the system using the key 'Q.'

2. PreProcessing and Feeding video to ML model using OpenCV

The captured input is processed frame by frame. This is done with the help of OpenCV tools. Firstly we resize the frames into the required resolution of 1920x1080p. Then the frame is fed as input to the Neural network created using the TFNet function available with Darkflow.

3. Video Processing using Machine Learning

The frames are processed using the YOLO You Only Look Once algorithm. Initially we are required to download the pre-trained weights for the Neural Network. The frame is divided into equal sized 'blocks.' They are processed by the Neural Network. It generates a 6x1 matrix weights which defines the parameters required for the 'object localization' here, 'Helmet.' The parameters include 'bounding box' dimensions i.e. object center coordinates, box width,box height, probability of class i.e. 'helmet' detected etc. A bounding box is usually a rectangular box surrounding the object detected. We can also show the probability with which the object has been detected. Thousands of similar images are to be used to train the model to increase its efficiency. For this, COCO dataset is used. It's an open source dataset consisting of various images of various objects like Helmet, animals, people, vehicles etc. All such images are labelled and provided with well-defined weights too.

4. Analysing ML output using Arduino

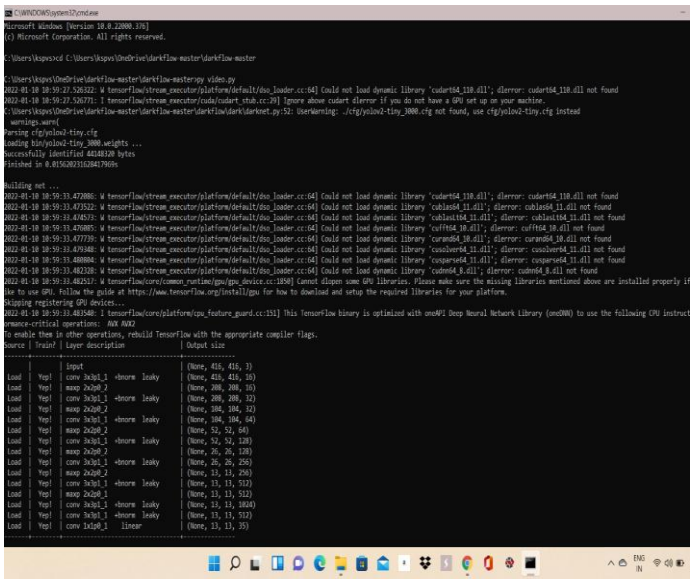
Visual data can be quickly understood and analyzed. Hence we output the data i.e. 'helmet detected' or not in the form of real time video and also using an LED and Buzzer to present the output. This is done by configuring the Arduino. Here, Pyfirmata Python library is used for giving input to Arduino. The Pyfirmata package can be used to give access to the write and read pins of the Arduino. Generally Arduino is to be synchronized. Based on the input that the Arduino receives we will control other pins' status.

Displaying output using Arduino

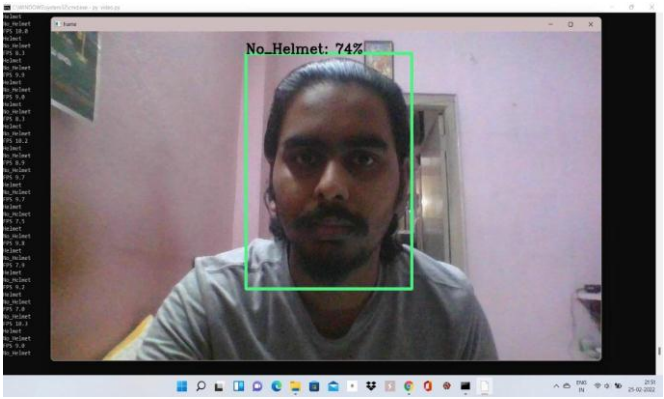
The basic displaying units are LEDs and Buzzer. Arduino has various methods to control the working of pins. pinMode function is used to set the particular pin as output or input. The digitalWrite function is used to send a particular voltage level on the pin. LED and Buzzer are two terminal components. The cathode of the LED and the negative pin of the buzzer are connected to the ground. Anode of LED, positive terminal of the buzzer are connected to digital pins of the Arduino. Based on the input to the Arduino, the output is displayed. If the Helmet is detected from the input given to the Arduino, then the LED doesn't glow and the Buzzer is silent. Else, if Helmet is absent then LED glows and buzzer will give a beep sound.

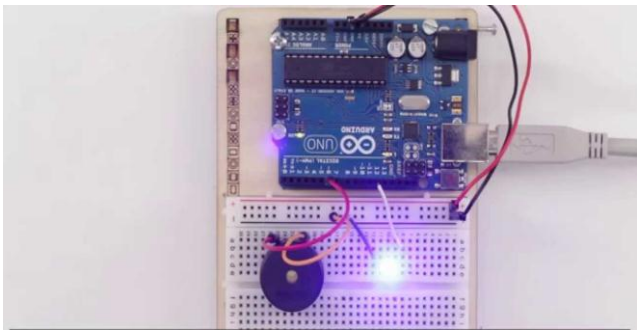
V. RESULTS

Capturing Video

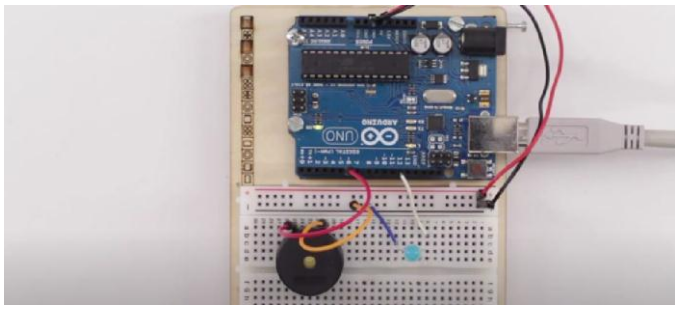


1. Output without Helmet

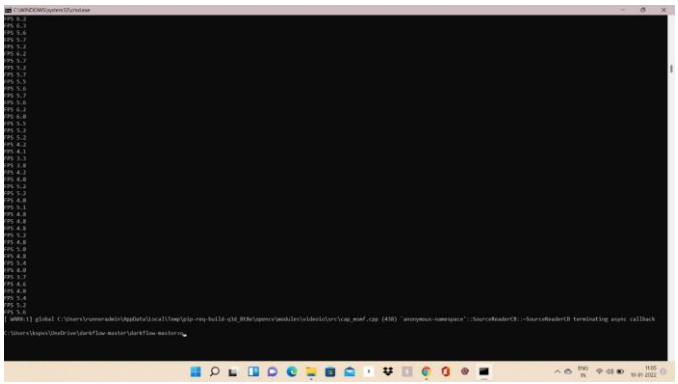




2. Output with Helmet



3. Terminating the Video Capturing



VI. CONCLUSION

Wearing a helmet is of utmost importance to reduce the number of fatal accidents and to reduce the impact itself. Hence we approached the solution using the Object Detection and Localization concept. For this we captured the real-time video feed of the motorist and fed it as input, frame by frame, to the Neural Network. We used the YOLO algorithm to analyze the frame, generate a weight matrix and to generate the 'bounding box' over the Helmet. YOLO does this so by breaking down the frame into equal sized blocks. Each block is checked to detect the central coordinate of the object here, Helmet. The block which contains this is identified and a bounding box is generated. If the Helmet is not detected, then a bounding box with the label 'No_Helmet' is shown.

VII. FUTURE SCOPE

This project can be used to implement an automatic challan system. It can be done by interfacing the Bluetooth module to arduino and based on helmet status for the amount of time automatic challan can be generated.

The Helmet Detection system can be interfaced to motorcycles. Doing this we can control the Bike condition based on Helmet Detection. If Helmet is detected initially then the bike can turn ON. Else the bike cannot turn ON.

VIII. REFERENCES

1. https://ieeexplore.ieee.org/document/8938394
2. https://turkjphysiotherrehabil.org/pub/pdf/322/32-2-1.pdf
3. https://www.irjet.net/archives/V6/i12/IRJET-V6I1214.pd
4. https://arxiv.org/abs/1506.02640
5. https://arxiv.org/abs/1612.08242