AI based Object Detection Robot through Live Streaming

Keerthankumar K V[1], Prateek Shiggavi[2], Praveen Kumar G[3], Vinayak kamaraddi[4], Kavitha S S[5]

[1]*Student, Department. of ECE, National Institute of Engineering, Mysuru, India*
[2]*Student, Department. of ECE, National Institute of Engineering, Mysuru, India*
[3]*Student, Department. of ECE, National Institute of Engineering, Mysuru, India*
[4]*Student, Department. of ECE, National Institute of Engineering, Mysuru, India*
[5]*Assistant Prof, Dept. of ECE, National Institute of Engineering, Mysuru, India*

Abstract— The need for Artificial Intelligence is increasing due to the increase in the complexity of modern world. The amount of data that is generated, by both humans and machines, far outperforms humans' ability to captivate, infer and make complex decisions based on that data. Artificial intelligence is the foundation to all computer learning and elucidates all complex decisions. AI based Object detection is one of the solutions that can be implemented in face detection, pedestrian detection, vehicle detection, military and etc. Object detection is one of the most basic and central tasks in computer vision. Its task is to find all the concerned objects in the image, and determine the category and location of the objects. In recent times, with the development of convolutional neural network, significant advances have been made in object detection.

AI combined with Robotics has made us reach far more than normal in solving these problems. Robotics include design, assembly, operation, and use of robots. The objective of robotics is to design machines that can benefit humans. In future mankind would be mainly dependent on robots. In this project we have used the hybrid technology of AI and Robotics. Here AI based Object detection is done using a Robot, as the robot is mounted with the hardware module for object detection, which include microcontroller, camera for streaming and circuit components, and can be programmed for the purpose, i.e. robot manoeuvring and object detection enactments.

Keywords: Artificial Intelligence, Convolutional Neural Network, Microcontroller, Robot Manoeuvring, Object Detection/Recognition.

I. INTRODUCTION

Object detection is the task of detecting multiple objects in an image that comprehenses both object localization and object classification. In this, our task is to identify and detect multiple objects from an image given through the robot on field through live streaming. In this application the detection of objects commonly includes pedestrian, car and motorcycle. Gesture Controlled Robot is the one that can be controlled by simple gestures. The user can control or navigate the robot by using gestures of his/her hands or palms. The command signals are generated from these gestures using an accelerometer which drives according to its position. These signals are then sent to the robot to navigate it in the specified directions. It doesn't require any complex joysticks or switches. Gesture Based Robotics involves human-machine interaction.

In Object detection, Representing an object or human is a complex function and various algorithms are being implemented to analyse these complex functions. Matching features by algorithms usually fail to identify most of the object those don't have texture, hence deep learning methods have been employed for better performance. There are different types of techniques for object detection, which can classifies as algorithms based on Classifications and algorithms based on Regression. RNN and CNN represents the algorithm related to Classification, here Image classification takes an image and predicts the object in an image using Convolutional Neural Network. This method is very slow because we have to run through entire network for prediction for every selected region. YOLO algorithm is based on Regression, here instead of selecting the particular part of an image, we predict classes and bounding boxes for the entire image in single run of the algorithm. YOLO (You Only Look Once) provides a vigorous result on existing object recognition datasets. For example, OverFeat model is eight layers deep whereas VGG16 model is 16-19 layers deep, quick R-CNN model is understood for its hybrid ability to capture the accuracy of deep layer models furthermore as up their speed at constant time. YOLO, that may be a 12-layer model, is understood for its superb prediction speed because it will predict up to forty five frames per second Here the paper follow the step by step implementation where Part II explains the implementation of Robotics, Part III explains in depth implementation and analysis of different Deep Learning Algorithms for object Detection and Part IV explains the celebration of Wi-Fi module. And thus this project provides real time operating model with the combinedimplementation of YOLO and Robotics.

II. ROBOTICS

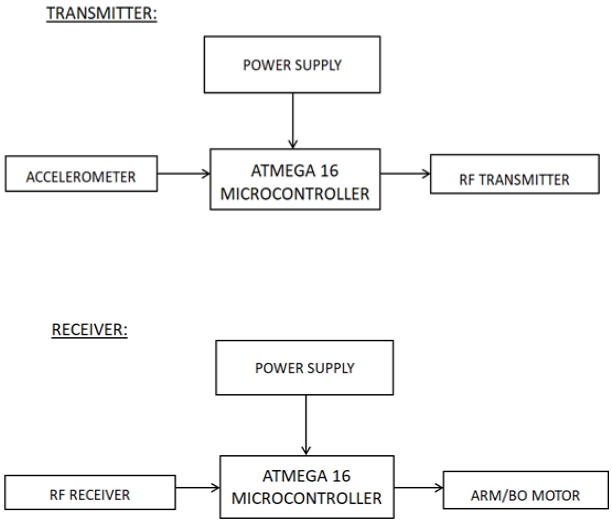
A. Working

Different technologies have been implemented for hand based recognition systems and few of them have shown good results. A definitive objective is to control certain frameworks like cell phones, forced air systems utilizing these procedures.

Quite possibly the most widely recognized methodologies is information glove based methodology. In this methodology there are sensors joined to glove which secure the motions and the signs produced by sensors are handled and comparing guidelines are performed.

Here the robot is also based on the same approach i.e., gesture controlled, where the input gestures given to the robot are hand made gestures. The command signals are generated from these gestures using accelerometer which drives according to its position. These signals are then sent to the robot to navigate it in the specified directions.

The below block diagrams represent the connections at transmitter and receiver:



A signal controlled robot utilizing an accelerometer is one sort of robot which can be worked by the development of hand by putting an accelerometer on it. This framework is partitioned into two sections transmitter gadget and beneficiary gadget. Where a motion gadget alongside switch interfaced fills in as a transmitter gadget and a robot functions as a recipient gadget. At the point when a communicating gadget (accelerometer) is put on the hand, then, at that point it will convey messages to the robot for the necessary activity. An accelerometer is a one sort of sensor and it gives a simple information while moving toward X, Y and Z. These bearings rely upon the sort of sensor. This sensor comprises of bolt bearings, in the event that we slant the sensor one way, the information at the specific pin will change as simple. The simple sign is communicated signals Microcontroller where it is changed over to advanced sign and afterward gets sent through RF transmitter.

The RF recipient will get the information which is moved by the communicating gadget. Furthermore, communicated to Microcontroller which in this way controls the DC BO engine.

The coordinates for the robot manoeuvring are given in the table below:

	X axes	***X axes***	***Y axes***	***Y axes***
	X min	*X max*	*Y min*	*Y max*
Stop	*318*	*324*	*315*	*308*
Forward	*318*	*321*	*348*	*355*
Reverse	*318*	*325*	*275*	*267*
Left	*280*	*270*	*310*	*320*
Right	*360*	*365*	*310*	*318*

B. Implementation Results:

Following are some of the hardware implementation of Robotics, involves the gesture and robot which are implemented on breadboard with Atmega16 microcontroller using Embedded C.

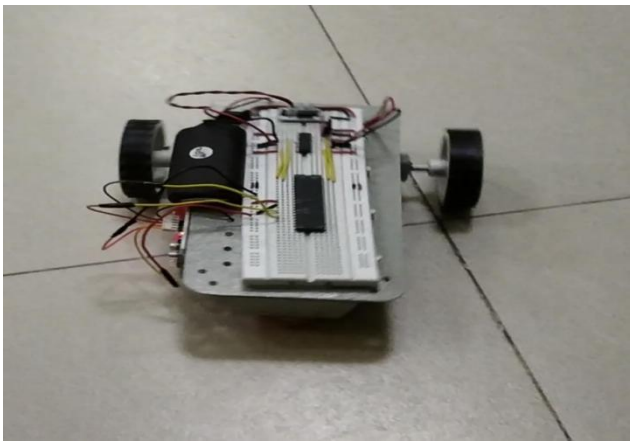


Fig. 1.1 Shows the picture of Robot

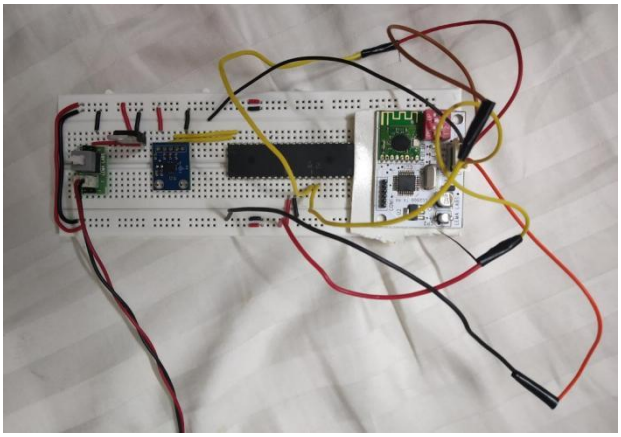


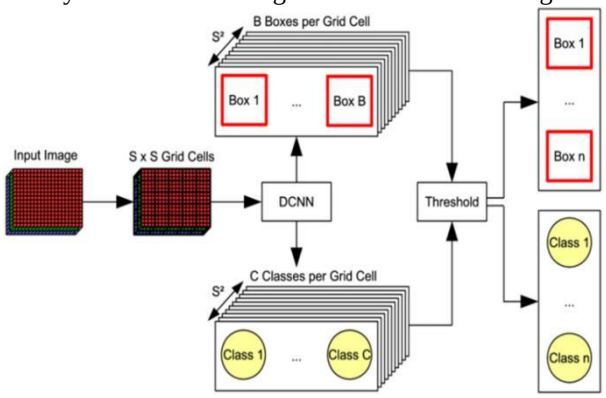
Fig. 1.2 Shows the picture of Gesture.

III. DEEP LEARNING ALGORITHM

A. YOLOv3

The YOLOv3 technique considers object discovery as a relapse issue. It straight forwardly predicts class probabilities and jumping box offsets from full pictures with a solitary feed forward convolution neural organization. It totally takes out area proposition age and highlight resampling, and typifies all stages in a solitary organization to shape a genuine start to finish discovery framework.

The YOLOv3 strategy partitions the info picture into S×S little lattice cells. In the event that the focal point of an item falls into a framework cell, the lattice cell is liable for recognizing the article. Every matrix cell predicts the position data of B jumping boxes and processes the misery scores relating to these bouncing boxes.



Each abjectness score can be obtained as follows:

$$C_i^j = P_{i,j}(\text{Object}) * IOU_{Pred}^{Truth} \quad \ldots.(1)$$

Whereby C_i^j is the abjectness score of the jth bounding box in the ith grid cell. $P_{i,j}(\text{Object})$ is merely a function of the object. The IOU_{Pred}^{Truth} represents the intersection over union (IOU) between the predicted box and ground truth box.

The YOLOv3 method uses binary cross-entropy of predicted abjectness scores and truth abjectness scores as one part of loss function. It can be expressed as follows: (2)

$$E_1 = \sum_{i-0}^{s^2} \sum_{j-0}^{B} W_{ij}^{obj} \left[\hat{C}_i^j \log\left(C_i^j\right) - \left(1 - \hat{C}_i^j\right) \log\left(1 - C_i^j\right)\right]$$

whereby S^2 is the number of grid cells of the image, and B is the number of bounding boxes. The C_i^j

And $\hat{C}_i^j$ are the predicted abjectness score and truth abjectness score, respectively. The position of each bounding box is based on four predictions: t_x, t_y, t_w, t_h on the assumption that (c_x,c_y) is the offset of the grid cell from the top left corner of the image. The centre position of final predicted bounding boxes is offset from the top left corner of the image by (b_x,b_y). Those are computed as follows: (3)

$$b_x = \sigma(t_x) + C_x$$
$$b_y = \sigma(t_y) + C_y$$

Whereby $\sigma()$ is a sigmoid function. The width and height of the predicted bounding box are calculated thus: (4)

$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$

whereby p_w, p_h are the width and height of the bounding box prior. They are obtained by
dimensional clustering.

The ground truth box consists of four parameters (g_x,g_y,g_w and g_h), which correspond to the predicted parameters b_x,b_y,t_w and t_h, respectively. Based on (3) and (4), the truth values of $\hat{t_x}$ $\hat{t_y}$ $\hat{t_w}$ $\hat{t_h}$ can be obtained as: (5)

$$\sigma(\hat{t_x}) = g_x - C_x$$
$$\sigma(\hat{t_y}) = g_y - C_y$$
$$\hat{t_w} = \log\left(\frac{g_w}{p_w}\right)$$
$$\hat{t_h} = \log\left(\frac{g_h}{p_h}\right)$$

The YOLOv3 method uses the square error of coordinate prediction as one part of loss function. It
can be expressed as follows:

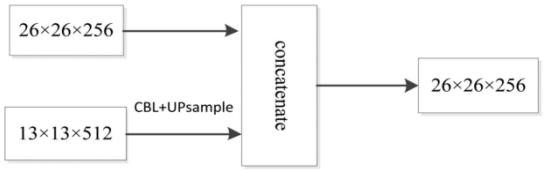
$$E_2 = \sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\left(\sigma(t_x)_i^j - \sigma(\hat{t}_x)_i^j\right)^2 + \left(\sigma(t_y)_i^j - \sigma(\hat{t}_y)_i^j\right)^2\right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\left((t_w)_i^j - (\hat{t}_w)_i^j\right)^2 + \left((t_h)_i^j - (\hat{t}_h)_i^j\right)^2\right]$$

B. Tiny YOLOv3

1. Tiny YOLOv3 Algorithm

The YOLOv3 technique isolates the information picture into S×S little framework cells. In the event that the focal point of an article falls into a framework cell, the lattice cell is answerable for identifying the item. Every network cell

predicts the position data of The Tiny YOLOv3 is utilized for the constant discovery. The Tiny YOLOv3 trunk highlight extraction network has seven convolution layers with 3 × 3 convolution pieces and one convolution layer with 1 × 1 convolution bits, six layers of maxpooling are utilized to decrease the boundaries. The item is anticipated by utilizing a two-scale expectation network with the yield include guide of 13 × 13 and 26 × 26. In the forecast organization, the Tiny YOLOv3 utilizes the upsampling to separate element and reinforce the component combination. In Figure 1, the 13 × 13 component map passes the convolution layer and upsampling layer. This transforms the 13 × 13 × 512 element map into 26 × 26 × 256. The element guide of 26 × 26 additionally is taken from the before in the organization and converged with the upsampling highlight by connection. At long last, the yield include guide of 26 × 26 is framed.



Minuscule YOLOv3 separates the information picture into N × N networks and predicts the jumping boxes inside every matrix cell, and the objective is recognized. At long last, the bouncing boxes and certainty for every arrangement of targets are proposed.(1) represents the formula.

The intersection over union (IoU) is defined as follows:(7)

$$\mathbf{IoU} = \frac{\mathbf{inter\_area}}{\mathbf{union\_area}} = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Bgt=(xgt,ygt,wgt,hgt) represents the position of the ground-truth, and

B=(x,y,w,h) represents the position of the predict box. Therefore, the IoU loss function is suggested to be adopted for the IoU metric. (8)

$$L_{\mathbf{IoU}} = 1 - \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}$$

The misfortune capacity of the Tiny YOLOv3 is characterized from three angles: the jumping box position blunder, the bouncing box certainty mistake and the characterization expectation mistake between the ground truth and the anticipated boxes. (9)

$$\begin{aligned}\text{Loss} = {} & \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{\text{obj}} \left[(x_i + \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{\text{obj}} \\ & \times \left[\left(\sqrt{w_i} + \sqrt{\widehat{w_i}} \right)^2 + \left(\sqrt{h_i} + \sqrt{\widehat{h_i}} \right)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{\text{noobj}} \left(C_i - \widehat{C}_i \right)^2 \\ & + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{\text{obj}} \left(C_i - \widehat{C}_i \right)^2 \\ & + \sum_{i=0}^{s^2} l_{ij}^{\text{obj}} \sum_{(c \in \text{class})} \left(P_i(c) - \widehat{P}_i(c) \right)^2\end{aligned}$$

Where λcoord,λnoobj are the weight parameters and given in advance, the latter is much smaller than the former. (xi,yi,wi,hi) is the predicted bounding boxes after normalization. $(\widehat{x_i}, \widehat{y_i}, \widehat{w_i}, \widehat{h_i})$ is the ground truth boxes after normalization. l_j^{obj} determines whether the j bounding box in the i cell grid contains an object. In addition, l_j^{obj} determines that any object central point falls in the cell grid (Fig. 3.2).

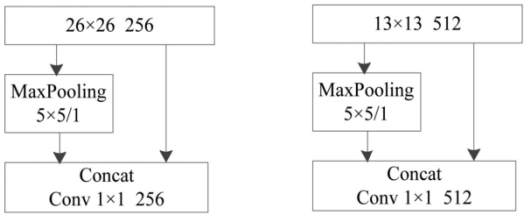


Fig. 3.2 The maxpooling and concatenation of the two scale.

2. Enhance Feature Fusion

In the maxpooling, the fundamental highlights are protected while the boundaries and estimation sum are decreased to forestall overfitting and further develop the speculation capacity of the model. SPP-net uses three unique scales for maxpooling. Along these lines, the further developed Tiny YOLOv3 depends on the SPP-net, it cuts the pooling scale decreases the information preparing. Yet, just the maxpooling layer of 5 × 5 is held. It can extraordinarily expand the responsive field and separate the main logical highlights. So a methodology is proposed in the worked on Tiny YOLOv3 to successfully extricate highlights. The two element maps are independently made by joining the maxpooling layer. Furthermore, the pooling portion is 5 × 5 with the convolution layer with the convolution piece is 1 × 1. As displayed in Figure 3.3, the maxpooling with 5 × 5 channels and 1 step. It can extraordinarily expand the open field and disconnect the main logical highlights.

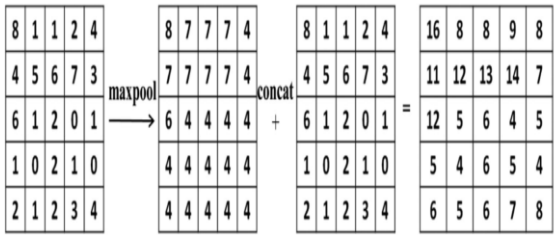


Fig. 3.3 The specific implementation of the maxpooling and concatenation.

In any case, the Tiny YOLOv3 just passes the yield highlight guide of 26 × 26 by convolution and connection to improve the element combination. Here another design, which adds a yield include guide of 13 × 13 by upsample to upgrade the element combination. It proposes the component map size of 13 × 13 to upsample and downsample the element map size of 26 × 26, to improve the element combination. 26 × 26 element map passes the zeropadding layer and the convolution layer. This turns the 26 × 26 × 256 element map into 13 × 13 × 512. We likewise take a component guide of 13 × 13 from prior in the organize and consolidation it with the zeropadding highlights utilizing connection. At last, the yield include guide of 13 × 13 is shaped. Figure 3.4 shows the above cycle.

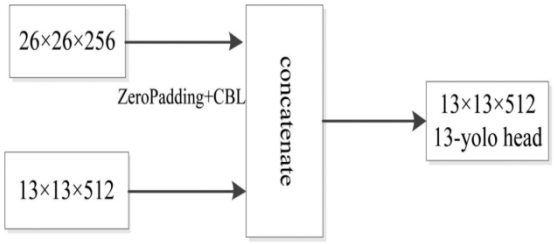


Fig. 3.4 The output feature map of the 13 × 13 feature scale.

3. Implementation Results and Analysis

Following the completion of the training process of the algorithm, a video of busy street has been used to evaluate the performance of the algorithm.

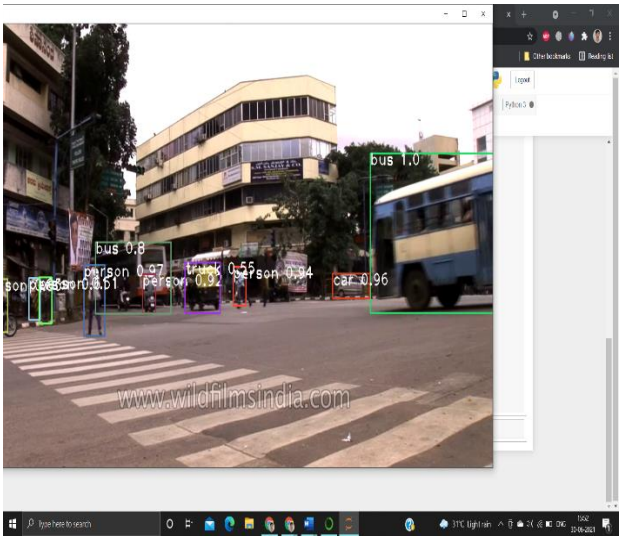


Fig 3.5 Predictions made by YOLOv3

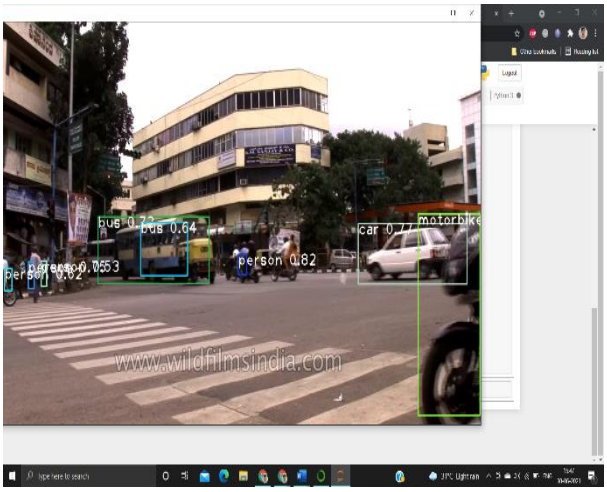


Fig 3.6 Predictions made by Tiny-YOLOv3

It is hard to make an unmistakable examination between various article identification strategies. Along these lines, we can't give a straight choice on the best model. For some genuine applications, we settle on decisions to make a harmony of precision with speed. Consequently, we should know about different qualities that altogether affect execution. For instance, coordinating with procedure and IOU edge, proportion of positive anchor and negative anchor, preparing dataset, usage of differing scale and edited pictures for preparing, area misfortune work, speed of learning, and learning rate rot and so forth.

The number of true positives, true negatives, false positives and false negatives that have been obtained by the models on the test video have been presented in the Table 3.1.

Algorithm	YOLOV3	Tiny YOLO V3	Faster R-CNN
True Positive	1214	986	1133
True Negative	195	184	192
False Positive	39	56	192
False Negative	126	354	207
Precision	0.9688	0.9462	0.9626
Recall	0.9059	0.7358	0.8455
F1 Score	0.9362	0.8278	0.90

- The accuracy of the models has been calculated to be 84.07%, 89.51%, 74.05% respectively for Faster R-CNN, YOLOv3 and tiny YOLOv3.
- The reason for the accuracy score of YOLOv3 being higher is because of its architecture where the object detections are performed at three different scales, making YOLOv3 more efficient in detecting smaller objects or detecting objects in difficult scenarios such as objects appearing partly in a certain frame
- The precision of YOLOv3 is higher that Faster R-CNN and TinyYOLOv3 is because, its predictions are very precise as it can detect at three different scales, whereas Faster R-CNN and Tiny-YOLOv3 struggled to show correct prediction where the size of the object is considerably small. Therefore, it can be concluded that the precision of YOLOv3 in real-time detection and tracking of the construction vehicles is really good.
- The recall values for Faster R-CNN and Tiny-YOLOv3 is lower than YOLOv3 as they have shown incorrect detections in many frames where the object is farther away or the size of the object is smaller, while YOLOv3 provided better results. . Therefore, it can be concluded that the recall of YOLOv3 in real-time detection and tracking of the construction vehicles is really good compared to other.
- Since the performance of the model is directly proportional to the F1 score and the upper limit of the F1 score being 1, it can be said that the performance of YOLOv3 model in real-time detection.

IV. Wi-Fi MODULE

A. Raspberry Pi 3 Model B+

The Raspberry Pi 3 Model B+ is the latest thing in the Raspberry Pi 3 region, displaying a 64-digit quad focus processor running at 1.4GHz, twofold band 2.4GHz and 5GHz distant LAN, Bluetooth 4.2/BLE, faster Ethernet, PoE limit through an alternate PoE HAT and an astounding Video Core IV GPU close by Pi camera of 5MP which get unrivaled quality video/picture. The twofold band distant LAN goes with disconnected consistence affirmation, allowing the board to be arranged into eventual outcomes with inside and out lessened far off LAN consistence testing.

B. Working:

The image / video which captured using Pi camera connected to raspberry Pi board is converted in to frames and sent into the YOLO model which will running on Raspberry Pi 3 using Jupyter notebook installed in the Raspberry pi 3.

Fig 4.1 Image Captured by Raspberry Pi Camera

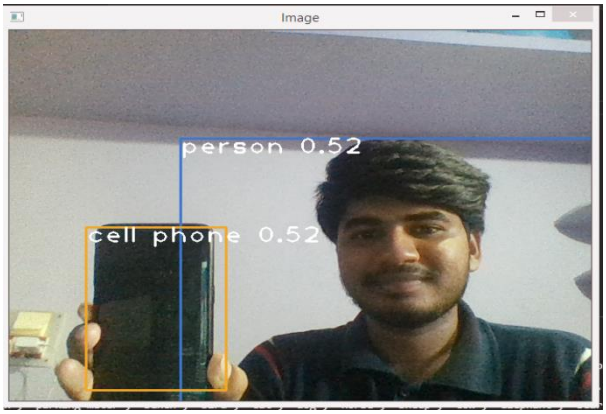


Fig 4.2 Working of YOLO model on Raspberry pi

V. CONCLUSIONS AND FUTURE WORK

The model is basically implemented with the collaboration of Robotics and Deep Learning Techniques, where the Gesture Controlled Robot is implemented using ATmega16 Microcontroller and Accelerometer for controlling the direction of Robot. The robot is mounted with the Raspberry Pi 3 along with Raspberry Pi camera module for live streaming which is connected to the system and the trained YOLO model is dumped into Raspberry Pi for further processing in the Deep Learning model. The object detection is implemented using Convolution Neural Networks to identify different objects, using a pre-trained YOLO model. Training of Deep Neural Network to fit the complex dataset will be done using Keras.

In the working model, as per the direction of the Gesture the coordinates are oriented by the accelerometer, which thereby processed by the Microcontroller and thus sent through RF Transmitter. The RF device being coupled, RF Receiver collects thedata of the coordinated and is passed to Microcontroller which determines the direction of the DC motor based on the values obtained. As the Robot is mounted with the Raspberry Pi Camera along with Raspberry Pi 3B+ model, there will be live streaming video along with desired movement of robot controlled by the gesture. This video is fed into YOLO model which is dumped on Raspberry Pi with the pre-installed Jupyter Notebook is able to detect various object in live streaming video. Thus the model with Gesture Controlled Robot, the video is shoot and thereby this video is subjected to YOLO model for the detection of object on its own for projecting video.

VI. REFERENCES

[1] Wang, B., and Yuan, T., "Traffic Police Gesture Recognition using Accelerometer", IEEE SENSORS Conference, Lecce-Italy,pp. 1080-1083, Oct. 2008.

[2] ATmeGA16-8-bit AVR Microcontroller

[3] Wendong Gai, Yakun Liu, Jing Zhang & Gang Jing (2021) An improved Tiny YOLOv3 for real-time object detection, Systems Science & Control Engineering, 9:1, 314-321, DOI: 10.1080/21642583.2021.1901156

[4] Zhao, Liquan & Li, Shuaiyang. (2020). Object Detection Algorithm Based on Improved YOLOv3. Electronics. 9. 537. 10.3390/electronics9030537.

[5] Real-Time Object Detection with Yolo Geethapriya.S, N.Duraimurugan, S.P. Chokkalingam https://www.ijeat.org/wpcontent/uploads/pape rs/v8i3S/C11240283S19.pdf

[6] Fabian Sachara, Thomas Kopinski, AlexanderGepperth, UweHandmann. Free-hand Gesture Recognition with 3D-CNNs for In-car Infotainment Control in Real-time. Proceedings of 2017 IEEE.

[7] Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf ProcessSYST25.

[8] https://intellica-ai.medium.com/a- comparative-study-of-custom-object- detection-algorithms-9e7ddf6e765e

[9] D. Agarwal, A. Rastogi, P. Rustagi and V. Nijhawan, "Real Time RF Based Gesture Controlled Robotic Vehicle," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021,pp.848-852,doi: 10.1109/INDIACom51348.2021.00152.

[10] "Real-Time Robotic Hand Control Using Hand Gestures" by Jagdish Lal Raheja, Radhey Shyam, G. Arun Rajsekhar and P. BhanuPrasad.

[11]https://paperswithcode.com/sota/object-detection-on-coco

[12] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017,pp.1-6, doi:10.1109/ICEngTechnol.2017.8308186.