Real-Time Object Detection Using Various Yolo Algorithms with Audio Feedback

Naganjaneyulu Satuluri[1],G. Asritha[2],V. Mounika[3],R. Shalini[4]
[1] Professor, Department of Information Technology, Lakireddy Bali Reddy College of Engineering,Mylavaram,Andhra Pradesh,India.
[2,3,4]LakireddyBaliReddyCollegeofEngineering,Mylavaram,AndhraPradesh,India
svna2198@gmail.com

Article Info
Page Number: 774 – 783
Publication Issue:
Vol. 71 No. 3 (2022)

Article History
Article Received: *12 January 2022*
Revised: *25 February 2022*
Accepted: *20 April 2022*
Publication: *09 June 2022*

Abstract
In Computer Vision, object recognition is a challenging application. It is used in many applications like security tracking, guiding visually impaired people, robotics, traffic signals and autonomous cars. Video analysis and image understanding have been improved through deep learning algorithms that work uniquely with different network architectures, that with the aim of identifying numerous objects from composite images. A large aerial image dataset called MS COCO was used to train YOLO algorithms. Raspberry Pi can be used as an external hardware source. The input is taken from the Pi camera when the button is clicked and then processed by NodeMCU and sent to Raspberry Pi which trains the data using Scaled-YOLOv4 algorithm which has high accuracy and high speed i.e. produces the accurate output in less time compared to any YOLO algorithm. When it detects the objects it generates output as audio which has object name. So it becomes easy for the user to understand the objects around them. This application is used for traffic signals, autonomous cars, security, blind people, robotics and many more things to detect objects around them.

Keywords: Object Detection, Scaled-YOLOv4, MS-COCO, Neural Networks, Darknet.

I. Introduction:

In computer vision, Object Detection is one of the most challenging application as it requires understanding of images in depth. For understanding the data which is small in size the visual system of human will be precise and accurate. For large data we need the system which is more accurate in order to identify the objects and localize them. Hereby the existence of machines come into picture, where computers are trained with the help of accuracy and preciseness. In object detection it's not just classifying the objects, along with this positions of various objects are located which varies from one image to other which further results in achieving good accuracy. One of the most challenging task is to develop the real-time object tracking algorithm which is more effective. Deep Learning has altered the domain of computer vision by working in these kinds of problems. You Only Look Once (YOLO) Algorithms are the ones that we use to detect the real-time objects using Neural Networks. These algorithms gained importance based on their accuracy, speed and learning capabilities. As the algorithm name implies it includes only one forward propagation through the neural networks to recognize the objects. Therefore, the whole image can be predicted in one go. As Raspberry Pi serves various purposes like home automation, edge computing, implementing Kubernetes clusters and many more. We are using it as a hardware source so that we can embed our algorithm in that minicomputer and use it for processing our data. We use an external Raspberry Pi camera in order to capture the real time objects in form of image. The buttons are present in the system in order to capture the image. The user can click the button whenever needed and the data is sent to Raspberry Pi using NodeMCU. These real time objects are from NodeMCU are processed using “Scaled YOLOv4” algorithm in order to detect the objects. Scaled YOLOv4 algorithm gives more accurate output in less time. These identified objects can be known with the help of a voice over. The gTTS which is a Google API is used to detect the text from the output of the algorithms and convert that into speech. We added three different languages of audio namely English, Hindi, Telugu in it using google translator. The user can the audio output in the required format specified by the user. This becomes easy for the user to identify the real time objects around them more easily. This process of taking the real time images and identifying them using Scaled YOLOv4 algorithms while giving the audio output of the detected objects that are present in the real time image can be used for various purposes. Like traffic signals in order to detect the vehicle number, blind people to detect objects around them, etc.

II. Literature Review:

Mansi Mahendru et al, proposed a system for object detection and recognition in real-time. The recognition of objects is one of the challenging applications of computer vision. The techniques used by the author was YOLO and YOLO_v3. In this research study, the precision value of YOLO is in the range of 65-85 % and YOLO_v3 is in the range of 65-98 % [1]. Joseph Redmon et al, proposed a system that detects real time objects using neural networks. To develop the current system, a single unified convolutional neural network is used. In this paper research study was done on different datasets using YOLO. Among all the datasets, VOC 2007 dataset has best average precision of 59.2 [2]. Shuo wang et al, proposed a system that deals with the analysis of objection detection models for real time Unmanned Aerial Vehicles (UAV) applications. In this paper, different YOLO series models are used and simulates them to XTDrone platform. Among all YOLOv4 has the best detection results with mean Average Precision (mAP) of 87.48 [3].

Chandan G et al, proposed a system that deals with the object detection and tracking in video sequence. To develop this system the concept of deep learning and computer vision are used. To implement Single Shot Detector (SSD) algorithm OpenCV library has been used. In this research study, the model was trained to detect 21 objects class with accuracy of 99% [4]. Kedar Potdar et al, proposed a system for live object detection that serves as a blind aid. To implement this system, convolutional neural network is used. In this system, a neural network is used for recognition of pre-trained objects on the ImageNet dataset. In this research YOLO model is tested on the ImageNet dataset which achieves 50 mAP in 200 object categories [5]. Chengji Liu et al, proposed a system that deals with the object detection using traffic signs as a research object for degraded models. In this research, developed a model which is evaluated in two stages. At first stage the model is trained on the academic datasets like ImageNet, COCO, VOC etc and in second stage, neural network model is tested on degraded image sets. To conclude that, the best detection accuracy on degraded image sets was achieved through this model [6].

Aleksa corovic et al, introduced a system for detection of traffic participants in real-time. In this research, the solution is specializing YOLOv3 neural network for real-time object detection and tracking of traffic participants on multiple classes [7]. Adwitiya Arora et al, proposed a system for object detection using image segmentation and deep neural network in real-time applications. In this research, it combines Single Shot Detection (SSD) and MobileNet for object detection. It improves accuracy and latency of real time detection [8]. Jeonghun Lee et al, proposed a system that deals with object detection using YOLO with Adaptive Frame Control (AFC) on AI embedded systems. AFC shows superior performance on various hardware platforms that are used in artificial intelligence applications for object detection [9]. Tanvir Ahmed et al, proposed a system using neural network for object detection. In this paper, YOLOv1 neural network was modified by including SPP and inception module with convolutional kernels. This modified YOLO network evaluated on two different datasets- Pascal VOC 2007/2012 with accuracy of 65.6% and 58.7% respectively [10].

III. Methodology:

A. System Design:

The proposed system automates the object detection in real time operations. The automation takes place by linking OpenCV with IoT. The system design is as shown in Fig 1.

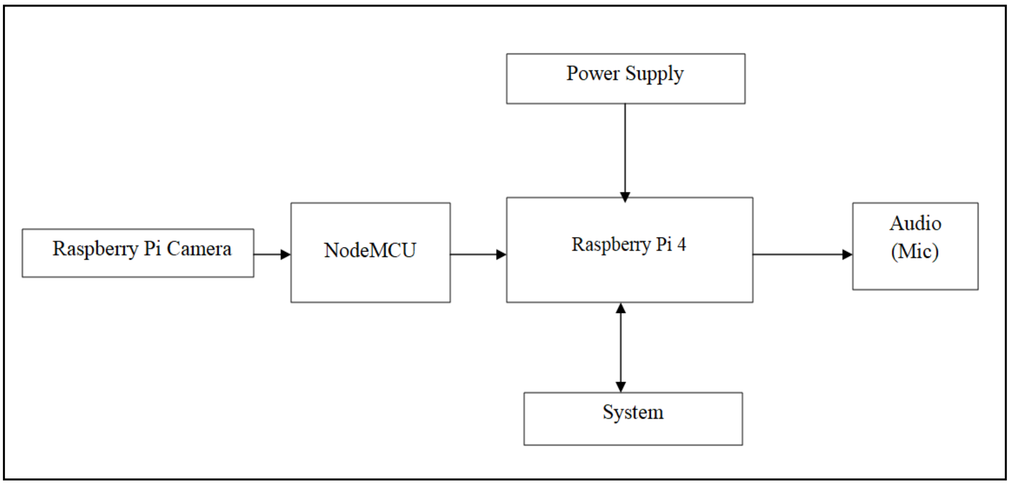

Fig 1: Block diagram of system

This system is installed with Pi camera, when the streaming of video or image gets started then, Pi camera receives that streamed video or image and that input gets converted into frames i.e., objects. An object detection model helps to detect objects in video or image.

Here, we are using Darknet neural network framework. To develop the model of real time object detection, we are using Scaled-YOLOv4 algorithm. Whenever an object is detected, convert that object image into OpenCV bounding box image then to PIL image. Based on the loaded object detection model, predict the class of an object in the image or video. If an object detected class label is predicted then the model converts that text label to audio output in three different languages. The system workflow is shown in Fig 2.

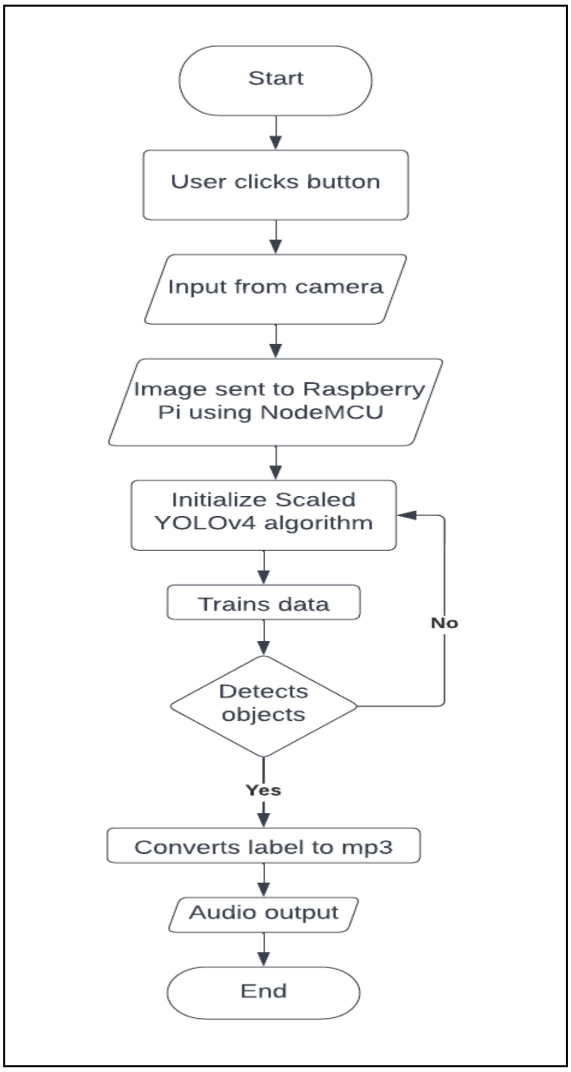

Fig 2: System Workflow

B. Scaled-YOLOv4 Architecture:

It is one of the best neural network algorithm for object detection. It outperforms among all neural networks with the best accuracy and speed. It approaches the best in both accuracy and speed. It has the ratio of speed to accuracy in the range from 15 FPS to 1774 FPS. It is the top neural network model for object detection. To detect large objects in large images, it is important to increase the depth and number of stages in the CNN backbone and neck which allows the dynamic adjustment of depth and width relative to real time inference speed requirements. Scaled-YOLOv4 is more accurate than EfficientDet, SpineNet, Paddle- Paddle

PP YOLO, etc. The YOLO and Cross Stage Partial network (CSP) outperforms well in both terms of accuracy and speed. The working of Scaled-YOLOv4 is shown in Fig 3.

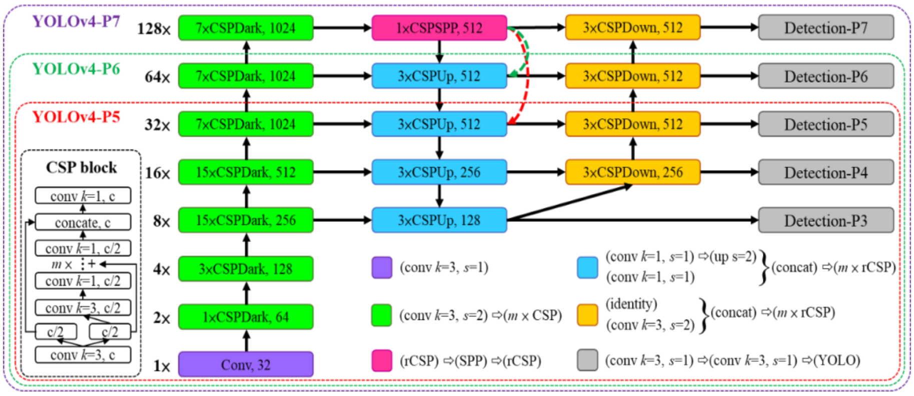

Fig 3: Working of Scaled-YOLOv4

C. Dataset:

For training the model, the dataset is taken from Kaggle Repository. MS-COCO (MicroSoft Common objects in Context) image dataset contains image data which is suitable for object segmentation and detection. In this dataset, the number of images are 3,30,000 while more than 2,00,000 images are labeled in which they are equal halves for training and validation+test. This dataset consists the classes of 80 object categories with image resolution 640x480. All object instances in the dataset are annotated with a detailed segmentation mask. The sample images of dataset are shown in Fig 4 as follows.

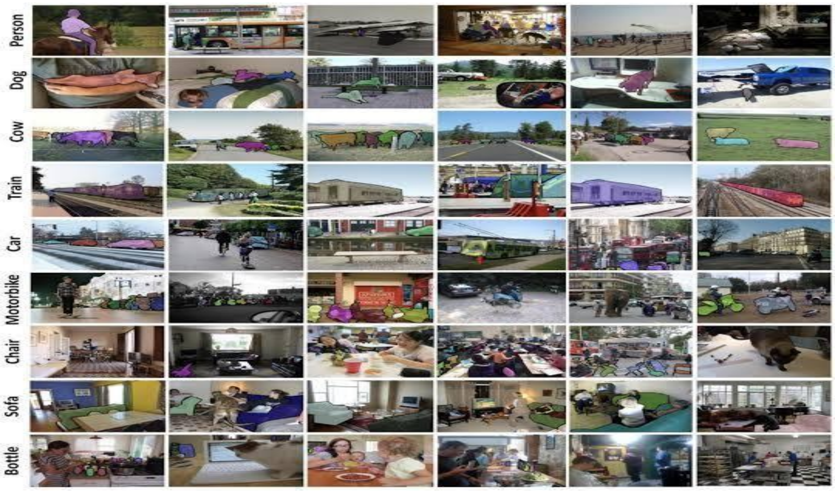

Fig 4: Annotated images samples in MS-COCO Dataset

D. Implementation:

The development of an introduced system was done in the following way:

1. Developing, training, and testing an Object Detection model:

Before running an object detection, at first it is necessary to train a model in order to identify an object in an image. Development of this detection model is done by using Scaled-

YOLOv4 with Darknet framework. We use pre-trained weights of Scaled-YOLOv4 model rather than initializing random weights which helps to take less training time.

To run detection on image, get image ratios to convert it into proper size bounding boxes. Convert the different type images to OpenCV bounding box image by converting the image bytes to numpy array using NumPy library, then decode numpy array to OpenCV image, then convert array to PIL image using PIL library, and format bounding box into png the return class label as in string format.

In order to train an object detection model, we need to have a dataset of labeled images that helps our model what it needs to detect and how. The training image data should be in a specific format that each image should have a file that contains the coordinates of the objects that are present in the image.

Now to develop the model, we are using the free GPU available with Google Colab for training our Scaled-YOLOv4 model for object detection. Then, upload the MS-COCO dataset to your google drive and you have to mount the drive into Colab.

In order to use Scaled-YOLOv4, we need to clone it using darknet repository which belongs to Alexey Bochkovskiy is one of the creator of YOLO. OpenCV is installed with CUDA and GPU in order to make fast computation and then build the Darknet framework.

To develop this model there are three configuration files such as .txt file, .names file, .data file, and Cfg file. In order to start the training process, we have to provide YOLO with a text file that contains the paths of all the images in the training set, .names file consists the names of your classes, .data file contains all the information about training path configurations, and Cfg file contains the values of all parameters used in training process as in the following three sections: 1. [net] section hold values like batch size, subdivisions, no. of steps, learning rate, etc., 2. [yolo] layers actual layers which performs object detection they contain loss types, coordinates of anchors, and non-max suppression values, 3. [convolution] layers are basic with activation functions and batch normalization.

Then, evaluate the model on test images and video to get the accurate result of AP (Average Precision).

2. Developing the system:

The evolution of our system should be done with the necessary components, before testing with dynamic inputs on the model. The kit representation of hardware components is as shown in Fig 5. Ensure that all the connections are secure and correct. At first, Raspberry Pi is connected to external monitor by using HDMI cable. Then Wi-Fi connection is to be done on the Raspberry Pi environment, such that we can interact with the Google Colab environment through our google drive. And enable the camera interface before the model deployment was done, which sequels the Pi camera working to detect the objects.

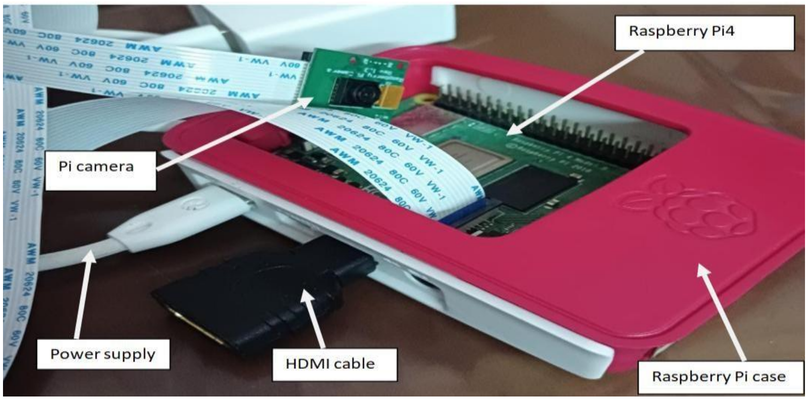

Fig 5: Kit representation

3. Testing with static and dynamic input:

The static input testcase is used to check the working of system when static image is provided as input to the model. Based on the code written, if an object is identified within the image then object detection takes place. From which it displays the output as labeled text of object class, confidence and an object within the bounding box then audio output of object class.

The dynamic input testcase is used to check the working of system when dynamic input such as image or video is provided through Pi camera. Based on the code written, if the image is captured then it detects the objects within that image and then it displays the output as labeled text of object class, confidence and an object within the bounding box then audio output of object class.

IV. Results And Discussion:

The development and training of the model has done as discussed in the earlier section of implementation. In Fig 6, the graph depicting the accuracy (AP) and latency (ms). Here the values of AP related to the various models are in Table 1.

Table 1: Values of performance metrics obtained during our model training

Algorithm	Average Precision (AP)
Scaled-YOLOv4	55.8
EfficientDet	55.1
SpineNet	50.3
YOLOv4	44.5
YOLOv3	34.3
PP-YOLO	45.2

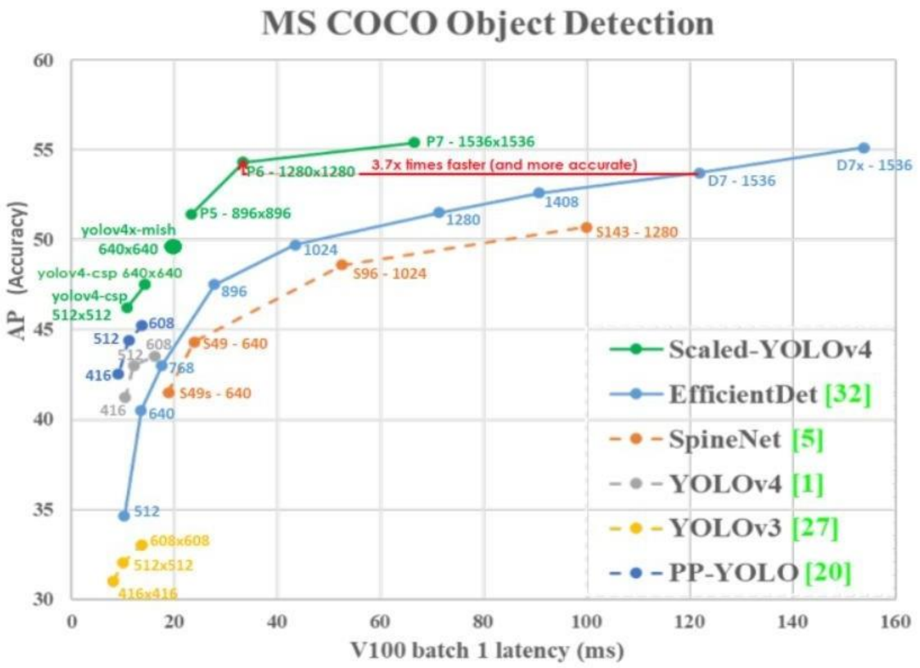

Fig 6: Graph showing various models training

While the model development, performance metrics that are considered are Average Precision (AP), and Latency (ms).

Average Precision (AP) is a popular performance metric in measuring the accuracy of object detection algorithms like MobileNet SSD, Faster R-CNN, YOLO, etc. It is the computation of average precision value for recall value over 0 to 1. AP is about finding the area under Precision-recall curve. AP is about the average over Intersection over Union i.e., the measure to know how much our predicted boundary overlaps the real object boundary.

$$AP= \int_{0}^{1} p(r)\, dr$$

Latency is a performance metric which is the amount of time taken by a system to process request for data and receiving the data requested across a network. Usually measure as round trip time, in milliseconds (ms).

After model training and kit development, the system is tested using static input and also with dynamic input through Pi camera. The system was developed as shown in Fig 5. The obtained results are shown in Fig 7, Fig 8, and Fig 9 respectively. The result shown in Fig 7, is that object detection with audio output taking static image as input, and in Fig 8 and Fig 9 showing that object detection with audio output taking dynamic image and video as input from Pi camera respectively.

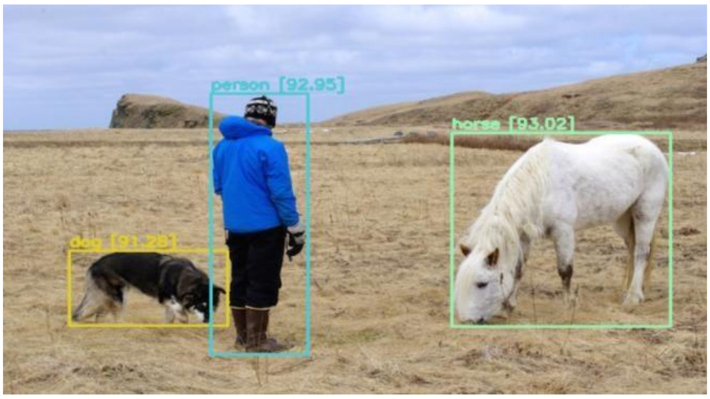

Fig 7: Result showing with static image as input

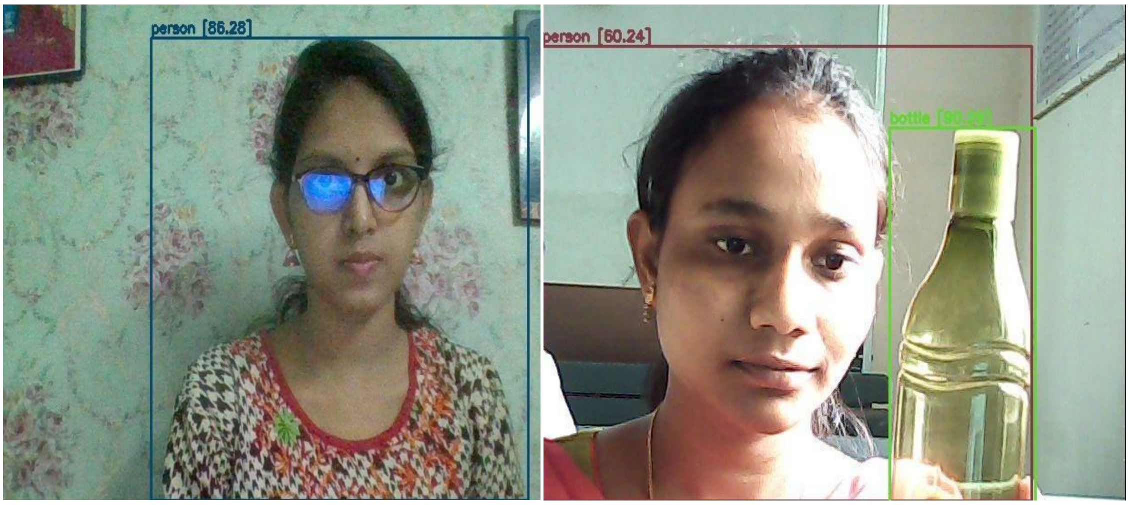

Fig 8: Results showing with dynamic images as input

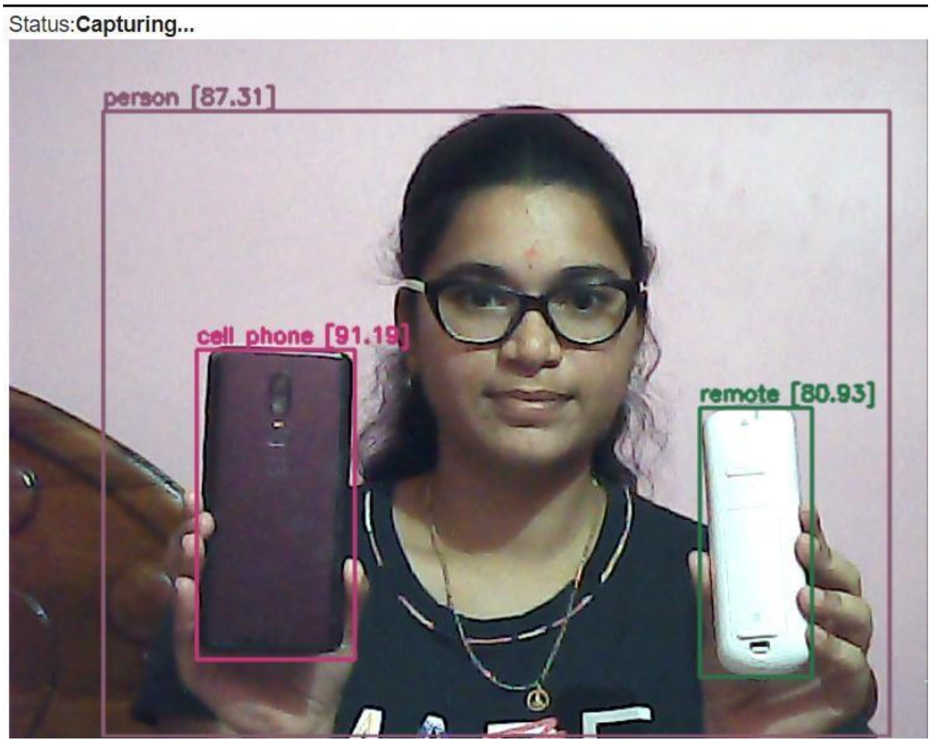

Fig 9: Result showing with video as input

V. Conclusion and Future work:

The development of an automated object detection model has been done in this paper. From the video or an image captured by a Raspberry Pi camera, by using the Scaled-YOLOv4 algorithm which uses Darknet framework the detection of objects takes place. We identified that when an object is detected within the image or video the label text of that object class which is detected is converted to audio output using gTTS module. The results have shown that the high accurate model is Scaled-YOLOv4 with average precision of 55.8. In future, we can extend the model by including the feature that finds the object distance relative to the person position, which helps the visually impaired person to restrict going near to the dangerous objects. Also, the model can be extended by adding the feature that identifies in which direction the object is present to the position of a visually impaired person.

References:

[1]. Mansi Mahendru, “Comparitive Results of Yolo and Yolo_v3 for Object Detection With Audio Feedback”, 2021.

[2]. Joseph Redmon, “Unified, Real Time Object Detection: You Look Only Once”, 2016.

[3]. Shuo Wang, “Comparitive Analysis of Object Detection YOLO-series Models for UAV Applications”, 2021.

[4]. Chandan G, “Object Recognition and tracking with use of Deep Learning and OpenCV”, 2018.

[5]. Kedar Potdar, “Live Object Recognition System as Blind Aid using Convolutional Neural Network”, 2018.

[6]. Chengji Liu, “Object Detection Based on YOLO Network”, 2018.

[7]. Aleksa Corovic, “YOLO Algorithm: Detection of Traffic Participants in real-time”, 2018.

[8]. Adwitiya Arora, “Object Detection for Blind With use of Single Shot Multibox Detector”, 2019.

[9]. Jeonghun Lee, “YOLO with AFC for detection of objects in real-time applications”, 2020.

[10]. Tanvir Ahmad, “Modified YOLO Neural Network Object Detection”, 2020.

[11]. Ashwani Kumar, “Object detection using improved single shot multi-box detector algorithm”, 2020.