International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhancement in Real Time Deep Learning Object Detection and Direction Prediction for Visually Impaired using YOLO and OpenCV

Kalyan Devappa Bamane[1], Nitisha Rajgure[2], Vinod Wadne[3], Simran Khaparde[4], Preeti Patil[5], Rutuja Vivek Tikait[6], Abhijit J Patankar[7], Aarti S Gaikwad[8]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: Millions of individuals around the globe have permanent visual impairment, underscoring the importance of facilitating their understanding of people and the identification of essential daily-use products. To address this need, we propose the system to recognize such items within their daily routines. Numerous initiatives are underway in this field to aid the visually impaired without end to end deployment. The objective is to identify objects and translate them into auditory cues to inform individuals with visual impairment about these items with the system comprises a camera, a speaker, and an image processing system. The primary focus of this study is the amalgamation of real-time object detection and recognition using advanced deep learning techniques. The aim is to detect and label the position and names of multiple objects captured by the camera through an object detection algorithm.

Keywords: *Object Detection, Deep Learning, Visually Impaired, YOLO, OpenCV, Image Processing.*

1. Introduction

Vision stands as a pivotal sense through which individuals effortlessly engage with their environment, recognizing objects and people in real-time. The visual context aids in determining the proximity and interaction strategies with the surrounding objects, enabling smooth communication for individuals with typical sight. As per the WHO survey more than 1 million people across the globe suffer with permanent visual impairment [1]. Conversely, the visually impaired encounter significant challenges in their daily tasks due to the absence of visual cues. Hence, it becomes imperative for individuals with visual impairments to gain insights into their environment and acquire object-related information. Employing object recognition algorithms founded on the You Only Look Once (YOLO) architecture—a deep learning model—allows for object detection via camera-based systems. This study focuses on analysing precise object information and locating those using advanced deep learning techniques in object recognition.

Object detection techniques are categorized into two primary groups. The first involves classification-based algorithms, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), which necessitate region selection and individual prediction, resulting in slower processing. The second category involves regression-based algorithms, where YOLO is a notable example. Here, predictions encompass classes and bounding boxes in a single algorithm run, enabling the detection of multiple objects using a unified neural network.

YOLO excels in speed compared to other classification algorithms, despite potential localization errors, presenting fewer false positives in the background. The proposed web application integrates the system's camera to capture environmental objects and continually retrieve video frames. These frames undergo analysis using the YOLO algorithm to identify and categorize objects by creating bounding boxes. The application then generates an audio output based on the most reliably detected object in a frame. To minimize audio output interference, specific frames are selected at predetermined intervals, reducing noise in the output.

2. Literature Review

The system to detect real time images using two distinct algorithms, Yolo and Yolo_v3 is proposed [2] which aims to detect multiple everyday objects and provide voice prompts to alert individuals about nearby and distant objects. The system is evaluated under similar criteria to assess accuracy and performance. Yolo employs TensorFlow's SSD Mobile Net model, while Yolo_v3 uses the Darknet model. To generate audio feedback, the gTTS Python library, which converts text to speech, is employed. The audio is played through the pygame Python module. Testing involves both algorithms using the MS-COCO Dataset comprising over 200K images, employing a webcam in various scenarios to evaluate the algorithms' accuracy in diverse situations.

[1,4,5,6,7,8] *D Y Patil College of Engineering, Akurdi, Pune, India*
[2]*Zeal Education Society*
[3]*JSPM's Imperial College of Engineering and Research, Wagholi , Pune*

The technique presented in the work [3] transforms the visual environment into an auditory landscape, enabling the visually impaired to detect and recognize objects in their surroundings independently. This real-time object detection system, powered by image processing and machine learning, uses a camera to identify objects and communicates their presence and location through audio output. The system limits in accuracy of object classification.

The study aimed at benefiting the individuals with visual impairment [4] endeavours to merge real-time object detection and recognition utilizing appropriate deep learning methodologies. The work is focused on detecting, identifying the position, and name multiple objects captured by a camera using an object detection algorithm, Image and video processing algorithms were devised to gather real-time data from the camera, employing Deep Neural Networks for object prediction. The system utilizes Google's renowned Text-To-Speech (GTTS) API module for precise voice output, accurately recognizing object categories and locations. Notably, the system demonstrates the recognition of 91 outdoor object categories, delivering speech-based output, even when confronted with limited spectral information in the data.

The project [5] offers a revolutionary solution—a smart walking stick that identifies everyday objects and notifies users through audio messages. This innovation transforms the visual world into an auditory experience, providing vital information to the visually impaired for independent navigation. Leveraging the YOLO (You Only Look Once) algorithm, our system detects real-time objects in images, displaying their names within bounding boxes, which are then converted into speech signals. The e-Speak tool, integrating Google's Text-to-Speech (gTTS) system, translates these names into audio signals. Comprising various modules, the prototype employs a Raspberry Pi camera module to capture images and transmit them to the Raspberry Pi desktop for real-time object detection using the YOLO network. The detected objects are then converted to speech using the gTTS module, and the audio output is delivered to the user via a headset, ensuring portability with the aid of a battery backup.

The paper [6], introduces a model that performs real-time detection of brightness and primary colors in images using the RGB method via an external camera. It also involves identifying essential objects and conducting facial recognition from a personal dataset. Object identification utilizes the YOLO Algorithm, while facial recognition involves MTCNN Networking. Object detection processing achieves a speed of 6-7 frames per second with an accuracy rate ranging between 63-80% and the face identification process attains an accuracy rate of 80-100%.

3. Overview of the proposed model

The figure 1 present's detection network comprises 24 convolution layers along with 2 fully connected layers. It employs alternating 1 X 1 convolution layers to diminish the feature space from previous layers. Initially, the convolution layers are pretrained on the ImageNet classification task at half resolution (224 X 224 input image), and later, the resolution is doubled for detection. The model operates as follows: The input image is converted to 448x448 before passing through the interface. It begins by using 1x1 convolutions to reduce the number of channels, followed by 3x3 convolutions to further decrease channel numbers, culminating in a cubic output. The primary activation function employed is ReLU, except for the last layer, which uses a linear activation. To prevent overfitting, the model applies additional techniques such as batch normalization and regularization.

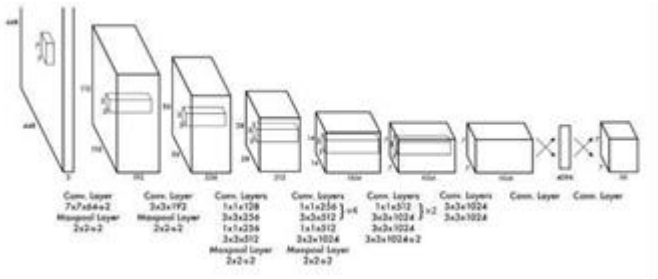

Fig 1: Detection architecture

Working of YOLO algorithm

Fig 2: Example on YOLO algorithm

The algorithm operates based on four fundamental approaches:

Residual Blocks: Initially, the original image (A) is divided into an NxN grid of identical images, typically with N being 4 for optimal image quality. Each cell within this grid is responsible for detecting and predicting the category as well as the probability/belief value associated with the item it contains.

Bounding Box Regression: Following the grid division, the algorithm proceeds to determine the bounding boxes outlining the rectangles that encompass all objects in the image. These bounding boxes correspond to each element present in the image. YOLO employs a variant of the given formula to define the properties of these bounding boxes. The final vector representation of each bounding box (Y) is

defined as: Y = [pc, bx, by, bh, bw, c1, c2]. This step is particularly crucial during the model's training phase. Here, 'pc' denotes the network's efficiency in containing the object; for instance, the probability that all red lines exceed zero. The image on the right simplifies this process, as each yellow cell holds minimal significance or impact.

Intersection Over Unions (IOU): This measure evaluates the overlap between predicted bounding boxes and ground-truth bounding boxes, aiding in fine-tuning and improving the accuracy of object localization.

Non-Maximum Suppression: This technique eliminates redundant or overlapping bounding boxes, retaining only the most probable and accurate detections, thereby refining the final output.

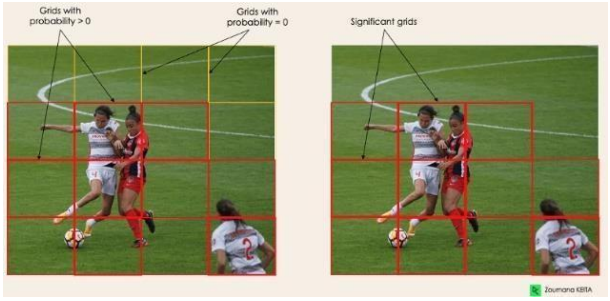

Fig 3: Example on YOLO algorithm

- bx, by are the x and y coordinates of the centre of the bounding box with respect to the enveloping grid cell.
- bh, bw correspond to the height and the width of the bounding box with respect to the enveloping grid cell.
- c1 and c2 correspond to the two classes Player and Ball. We can have as many classes as your use case requires.

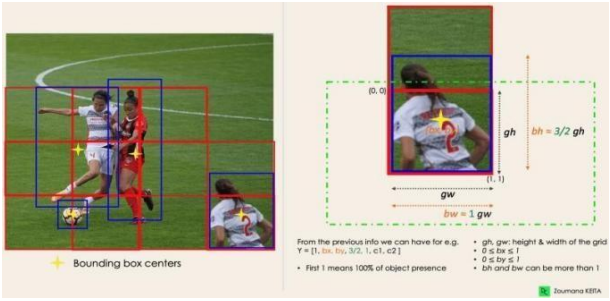

Figure 4: Example on YOLO algorithm

Intersection of organisations or IOU: Oftentimes, an object in a picture may have several boxes competing for prediction, even though not all boxes are connected to each other. The purpose of IOU (value between 0 and 1) is to discard grid boxes and keep only relevant ones. The logic behind this is as follows: Users define their initial IOU selection, which could be for example. YOLO then calculates the IOU, which is the intersection area of each cell divided by its association area. Finally, it ignores predictions for rows with threshold IOU ≤ and calculates rows with threshold IOU >. The following is an example of using the mesh selection process for low-level objects. We can see that the starting object has two candidate meshes and, in the end, only “Grid 2” is selected.

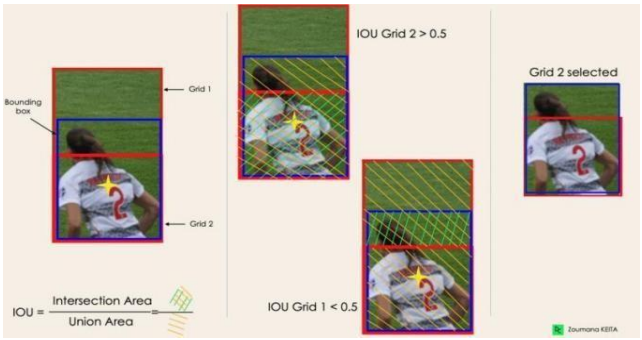

Figure 5: Example on YOLO algorithm

Setting the maximum limit or NMS: IOU threshold is not always sufficient because a product may have many bins with IOUs exceeding the threshold and all these bins will be loud. Here we can use NMS to store only the frames with the highest probability score.

4. COCO Dataset

The COCO (Common Objects in Context) dataset is a large- scale image database for object detection, segmentation, and aggregation tasks. It has over 330,000 images, each filled with a 5-line description of 80 items and locations. The COCO database is widely used in computer science and has been used to train and evaluate many state-of-the-art detection and segmentation models. This file has two main parts: images and descriptions. Complete list of training, validation and testing links. The description is in JSON format and each file corresponds to an image.

The COCO (Common Objects in Context) dataset group is divided into two main groups: "Objects" and "Objects". The "Objects" class includes things that are easy to store or manipulate, such as animals, cars, and items. materials found at home. Examples of "things" categories in COCO are: People, Bicycles, Houses, Bicycles.

"Stuff" classes include background or environmental items such as sky, water, and road. Examples of "stuff" classes in COCO are: Sky, Tree, Road.

Fig 4: COCO Dataset Class List

Object detection is the most popular application of computer vision. Explores objects with boxes to divide and divide into pictures. The COCO file can be used to demonstrate sample detection. This data provides bounding boxes for 80 different objects and can be used to identify bounding boxes and train models to classify objects in images.

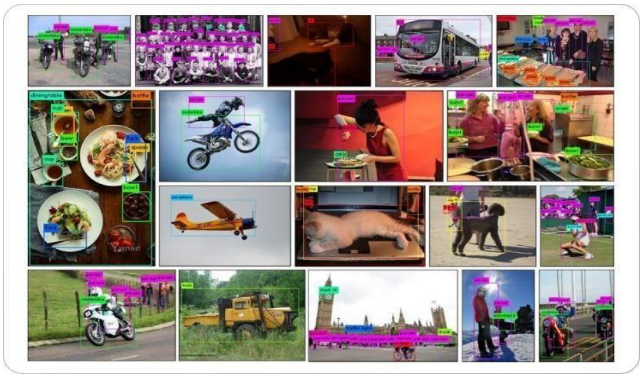

Fig.7 Results of YOLO_v3 in the COCO Dataset

5. OpenCV Library

OpenCV, a massive open-source library for computer vision, machine learning and graphics, now plays an important role in business processes that are crucial in today's systems. Using it, people can process images and videos to identify objects, faces, and even human writing. When integrated with many libraries (such as NumPy), Python can process OpenCV sequence models for analysis. We use vector space to describe image patterns and their different features and perform mathematical operations on these features.

OpenCV Applications: Many applications have been solved by OpenCV, some of them are listed below-

- Facial Recognition
- Automatic Detection and Tracking People - in shopping malls etc. counting (flow of people) in places.)
- Retrieval Robot and Driverless Vehicle
- Navigation and Control Object
- Recognition Medical Image
- Analysis Movies - 3D Structures in Motion TV Channel Advertising Recognition

6. Speech Output Using Gtts Api

A new audio system has been added that uses speakers or headphones or a Bluetooth headset, especially for alarms or to keep the user aware of things around or in front of them. In addition, voice output during navigation will be more effective and faster, especially when the user is walking on the road. So, poor visibility will give the necessary warning or stop for a while until the object in front comes out. The module works using Google's Text-to-Speech (GTTS) API, which is widely used on Android smartphones. It is a screen reader program developed by Google for the Android operating system. GTTS can read on-screen text aloud with the help of multiple languages including German, French, Tamil, Hindi, English and more. All of these languages are supported in the GTTS API. It was released on November 6, 2013. The sermon is delivered in a fast or slow voice. However, according to the latest update, the sound design cannot be changed. The best thing about this API is that it sounds great.

7. Comparison of Yolo with Other Object Detection Algorithms

Faster R-CNN: The Faster R-CNN model was developed by a research team at Microsoft. Faster R-CNN is a deep neural network for detecting objects that appears to users as a unified, end-to-end network. This network can estimate the location of different objects. To truly understand Faster R-CNN, we also need to know about R-CNN and the networks it evolves from, such as Fast R-CNN. Faster R-CNN is an extension of Fast R-CNN. As the name suggests, Faster R-CNN is faster than R-CNN due to the Region Proposal Network (RPN). R-CNN is an integrated model, the model consists of two modules:

RPN (Region Proposal Network): Used to specify regions and regions in order to determine the convolutional neural network for the object type.

Fast R-CNN: Convolutional neural network for extracting features from desired regions and outputting bounding box and class labels.

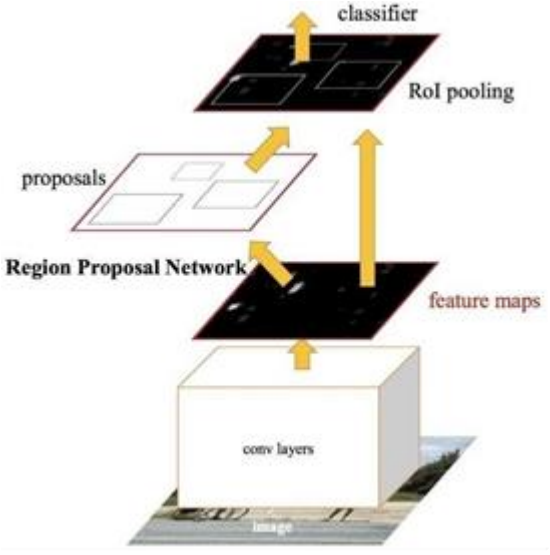

Fig.8 Faster R-CNN Architecture

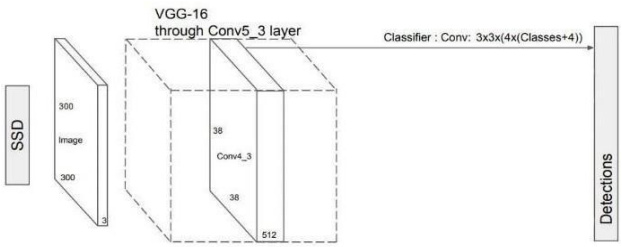

Fig.9 SSD Architecture

SSD: SSD is an anti-virus. It does not refer to a network application domain and predicts boxes and classes directly from maps in one go. Object detection of SSD consists of 2 parts: extracting the map and using convolution filters to detect objects.

SSD can be trained end-to-end for better accuracy. SSD can be more predictable and better cover location, scale and aspect ratio. By eliminating the fine-grained region and using lower resolution images, the model can run faster and still maintain state-of-the-art accuracy faster than R-CNN.

Algorithm for proposed method:

- Consider an image and we need to create a grid that will give us the features of an object.
- We will use OpenCV which will read the input image and data points and specify the file path to an image in a NumPy array.
- Next, we will detect an image in a grid view after the process of reading the image by OpenCV and NumPy and converting the grid to rectangular boxes.
- The last step involves displaying the image with a rectangle and the name of the viewport. This will be done using the YOLO and COCO dataset. Here are the steps to navigate through the app:
- Open the app on your smartphone.
- When you start the application, the camera will be displayed.
- The camera then captures real-time images.
- After pressing the button that says "Start/Stop" the live view will be via OpenCV.
- At each point, the detected product will appear in the box along with its product, tag and trust score.
- The exact location of the detected object and the relative position of the object. Visually impaired people are warned with sound output.

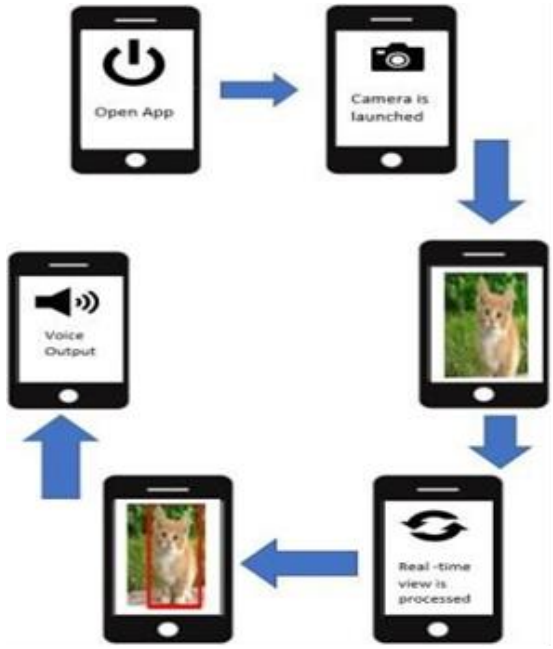

Fig. 10 Sequence of events for the Android App

Expected Real-Time Accuracy Metric:

From the YOLO algorithm, we expect to detect objects in an image. Comparing the 'time' metric for object detection of Fast-RNN, Faster-RNN, and YOLO; we conclude that YOLO is faster than the other two algorithms.

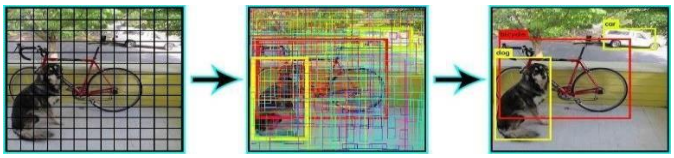

Fig.11 Image before and after object detection

Method	Fast-RCNN	Faster-RCNN	YOLO
Time (Sec)	55.769754	48.45637	7.366887

Table.1 Time Taken to detect objects

As we see in the graph (Figure 12), YOLO is 7 times faster than Fast R-CNN and 6 times faster than Faster R-CNN and also Faster R-CNN is faster than Fast R-CNN but not faster than YOLO.

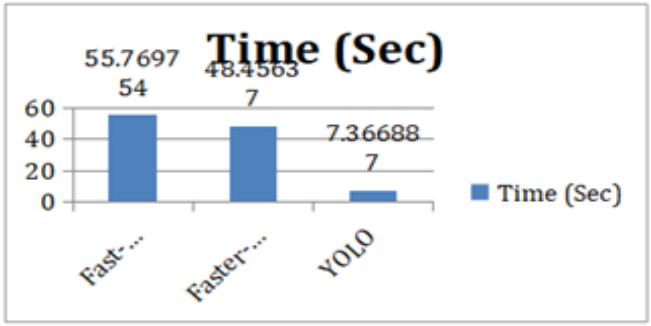

Fig.12 Time Taken to detect objects

Performance Comparison:

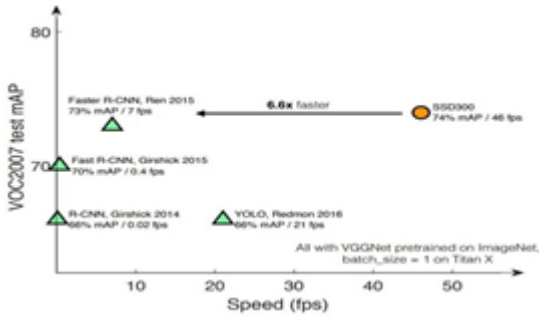

Fig.13 SSD vs Faster R-CNN vs YOLO

The following is a scatter plot of speed and accuracy of the major object detection methods (R-CNN, Fast R-CNN, Faster R-CNN, YOLO and SSD300). Note that YOLO and SSD300 are the only single shot detectors, while the others are two stage detectors

8. Conclusion

In this article, we propose and recommend the YOLO algorithm for target detection due to its advantages. This algorithm can be used in many areas to solve some real-life problems such as safety, monitoring traffic lanes, and even helping blind people with movement. The algorithm is generalizable and outperforms different ideas when generalized from natural images to different domains. This algorithm is easy to design and can be trained directly on full images. YOLO Enter the entire image while guessing the border. It also predicts less negativity in the background area. Compared to other classifier algorithms, this algorithm is more efficient and faster for immediate real time use.

9. Future Work

Create an object detection system with the help of deep learning like YOLO and use this method to estimate the location of objects. The system would provide voice guidance for the blind. This system will be designed to help the visually impaired people. The accuracy of the system can be increased by using YOLO Algorithm. Further, existing systems only contain the object detection feature. Our proposed system will also consist of a direction prediction feature that will indicate on which direction of the user the object is actually placed. The YOLO Algorithm will be chosen since it provides a faster and better result as compared to other algorithms. Python is the preferred programming language for developing models and rapid application development due to its simplicity, simplicity, wide selection of libraries and frameworks, and freedom platform. OpenCV is a computer vision and machine learning software library for graphics. OpenCV is the most popular library for real-time applications as it increases computational performance and provides excellent libraries. We would also be creating an android application for object detection.

Acknowledgements

This research was supported/partially supported by [Name of Foundation, Grant maker, Donor]. We thank our colleagues from [Name of the supporting institution] who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper. We thank [Name Surname, title] for assistance with [particular technique, methodology], and [Name Surname, position, institution name] for comments that greatly improved the manuscript.

Author contributions

Name1 Surname1: Conceptualization, Methodology, Software, Field study **Name2 Surname2:** Data curation, Writing-Original draft preparation, Software, Validation., Field study **Name3 Surname3:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] M. Mahendru and S. K. Dubey, "Real Time Object Detection with Audio Feedback using Yolo vs. Yolo_v3," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 734-740, doi: 10.1109/Confluence51648.2021.9377064.

[3] Vaidya, Sunit, et al. "Real-time object detection for visually challenged people." 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2020.

[4] M. I. Thariq Hussan, D. Saidulu, P. T. Anitha, A. Manikandan and P. Naresh (2022), Object Detection and Recognition in Real Time Using Deep Learning for Visually Impaired People. IJEER 10(2), 80-86. DOI: 10.37391/IJEER.100205.

[5] Therese Yamuna Mahesh et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1085 012006.

[6] Ferdousi Rahman, Israt Jahan Ritun, Nafisa Farhin, and Jia Uddin. 2019. An assistive model for visually impaired people using YOLO and MTCNN. In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy (ICCSP '19). Association for Computing Machinery, New York, NY, USA, 225–230. https://doi.org/10.1145/3309074.3309114