Multicultural Education

Volume 7, Issue 6, 2021

# You Only Look Once (YOLOv3): Object Detection and Recognition for Indoor Environment

**Hassan Salam , Hassan Jaleel , Salma Hameedi**

**Article Info**

***Article History***

Received:
April 07, 2021

Accepted:
June 07, 2021

***Keywords:***
Computer Vision, Convolutional Neural Network (CNN), Object detection, OpenCV,

**DOI:**
10.5281/zenodo.4906284

**Abstract**

*Computer Vision (CV) is a study field that is responsible for developing techniques to perform tasks that the human visual system can do. Object detection is a technique used for detecting objects in videos and images. The research aims at detecting objects for indoor environment such as offices or rooms in different conditions of lighting by using YOLOv3 and generating a voice message for each detected object. YOLOv3 outperforms the other deep learning algorithms such as CNN because it looks at the entire image by predicting the bounding boxes using Convolutional Neural Network and finding class probabilities for these bounding boxes. However, CNN does not look at the image completely; it splits the image into regions that sequentially enter the neural network for performing the object detection and recognition process. This makes YOLOv3 faster than other deep learning algorithms. Open-source Computer Vision (OpenCV) was used for capturing the video frames. Then YOLOv3 was used to detect the objects in each frame and determine their location. Finally, the sound in the Arabic language was generated for the detected objects. The proposed method can detect six objects with an accuracy of 99% in the overall performance.*

## Introduction

Computer vision (CV) is a technology that uses machines for processing and understanding videos and photos. There are many tasks that belong to Computer Vision; one of the main tasks is object detection. Object detection is a technique that is used for detecting the objects in videos and images [1, 2, 3]. Object detection has many applications such as the applications that assist blind and visually impaired people in recognizing the objects in their environment, self-driving cars, car plate detection, face detection and automated parking systems [4, 5]. There are many algorithms used for performing object detection such as YOLO, CNN, R-CNN, etc. [6]. The proposed algorithm is YOLOv3, which is a fast and accurate method for object detection.YOLOv3 consists of Convolution Neural Network (CNN) and an algorithm for processing outputs from neural networks [7]. The Convolution neural network (CNN) is a Deep Neural Network which is similar to the human visual cortex. CNN has one input layer, two or more hidden layers, and one output layer. The input layer is the first layer in CNN through which the image enters the neural network. The number of neurons in the input layer is equal to the number of features. The hidden layer is consisted of convolution, activation, pooling, and fully connected layers. CNN has at least one convolution layer that computes a dot product between the connected region in the input and the weights for producing a feature map. The activation layer (ReLU activation function) is used with the convolution layer to accelerate the training process by removing the negative values. The result of the activation layer is then downsampled (pooled) by using the pooling layer for simplifying the feature map. The output of these layers is connected using a fully connected layer. The fully connected layer is a one dimensional layer having all the labels to be classified, and the score for each label of classification is produced by this layer. The output layer is the last layer of CNN where the number of classes is equal to the number of neurons of this layer [8, 9, 10, 11]. Figure 1 explains the structure of Convolution Neural Network [11].

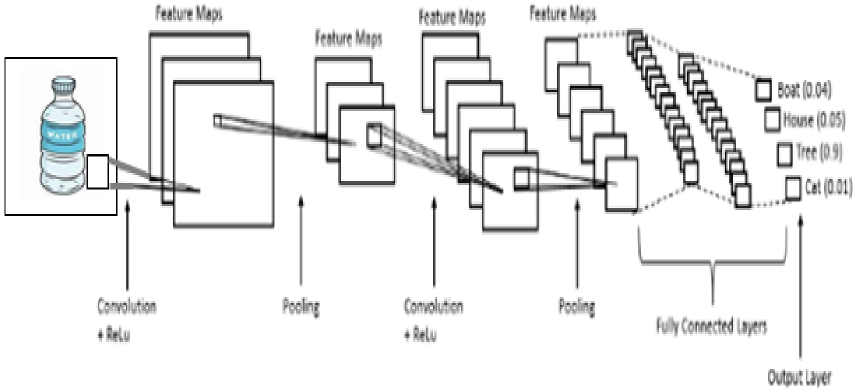


Figure 1. The structure of CNN[11]

The remaining of this paper is organized as follows: Section 2 gives some brief background of the research efforts related to the proposed method. Section 3 describes the proposed object detection and recognition system. Section 4 explains the training process for the Deep Neural Network. The experimental results and discussions of the proposed method are presented in Section 5. Finally, the conclusions are given in Section 6.

**Literature review**

Many research attempts have been made for object detection and recognition by using deep learning algorithms such as CNN, RCNN, YOLO, etc. To understand some of these algorithms, a literature review is conducted in this research [14].

Aleksa Ćorović, et al. (2018) implemented a system for detecting the traffic participants using YOLOv3 and Berkley Deep Drive dataset. The number of object classes which can be detected by this system is five (truck, car, traffic signs, pedestrian and lights) in different conditions of the driving (snow, overcast and bright sky, fog, and night). The accuracy was 63% [7].

Rohini Bharti, et al. (2019) implemented smart glasses to help visually impaired people. CNN with Tensorflow, custom Dataset, OpenCV, and Raspberry Pi were used to implement this system. Sixteen classes can be detected by this system. The system achieves an accuracy of 90% [15].

Omkar Masurekar, et al. (2020) created a model for object detection to help visually impaired people. YOLOv3 and Custom dataset that consists of three classes (bottle, bus and mobile) were used. Google Text To Speech (gtts) was used for sound generation. The authors found that the required time for detecting the objects in each frame was eight seconds and the achieved accuracy was 98% [16].

Sunit Vaidya, et al. (2020) implemented a web application and an android application for object detection. YOLOv3 and coco dataset were used in these systems. The authors found that the maximum accuracy in web applications is 89 % and 85.5% in mobile phones. The required time for detecting the objects was two seconds and this time increased by increasing the number of objects [17].

Shifa Shaikh, et al. (2020) used Raspberry Pi, YOLOv3, and coco dataset to implement a system for object detection. The accuracy was 100% (for a chair, person, cell phone and clock) and 95% on the overall performance [18].

A. S. Mahmoud, et al. (2020) implemented a model for object detection in optical remote sensing images. Mask RCNN and NWPU-VHR-10 dataset were used. The model can detect ten types of objects. The highest accuracy was 95% and the required time for detection was 7.1 seconds [19].

Deep learning algorithms are used for implementing other types of applications such as monitoring systems, sign language translation, etc.

Azher Atallah, et al. (2020) created a system for sign language translation. CNN with Tensorflow and custom dataset were used. This system converted the sign language into voice. The system can recognize 40 hand gestures. The accuracy of this system was 98% [20].

E. K. Elsayed, et al. (2021) created a semantic translation system for dynamic hand gestures. Three-dimensional CNN followed by convolutional long short-term memory for improving the recognition accuracy were used. Three datasets were used, the first dataset was LSA64, LSA was the second dataset and the third one was a custom dataset, which contains 11 dynamic Arabic gestures. The recognition accuracy was 97.4% [21].

Abdulwahab A. Abdulhussein, et al. (2020) implemented a Hand Gesture Recognition system using CNN and the Custom dataset. Twenty-four letters were recognized by this system and the accuracy was 99.3% [22].

Sajidah S. Mahmood, et al. (2020) implemented a monitoring system for detecting and classifying moving vehicles in a video using CNN and the Custom dataset. The author found that the accuracy was 92% [23].

I. A. M. Zin, et al. (2020) created an application for herbal plant recognition using CNN with the Custom dataset. The system can recognize twelve types of plants with an accuracy of 99% [24].
M. Anandhalli, et al. (2021) implemented a model for vehicle detection and tracking using CNN with the Tensorflow and Custom dataset. The accuracy of this model was 90.88% [25].
A. A. Naufal, et al. (2020) developed a smart parking system. Mask RCNN was used to detect the parking space. Moreover, mAlexNet was used to determine if the parking space is empty or not. Two datasets were used, the first dataset contains images of the parking area and the other dataset contains videos from a CCTV record to detect parking space availability in the parking area. The first dataset was CNRPark, which was used for detecting the parking space. The IOU for marking the parking positions using Mask RCNN was 85.80% while the accuracy for parking space availability using mAlexNet was 73.73% [26].
R. F. Rachmadi, et al. (2020) implemented a model for kinship verification depending on faces' images. Family-aware CNN and the FIW (Families In the Wild) dataset were used. The accuracy of this model was 68.84% [27].
R. Parmar, et al. (2017) created a model for detecting diabetic retinopathy from retinal images using CNN with the Custom dataset. The accuracy of this model was 85% [28].
Our proposed system uses YOLOv3 with the Custom dataset. It produces a high accuracy, which is 99%, for object detection and recognition.

## The Proposed Method

In this paper, a proposed object detection and recognition system is presented. The proposed system uses You Only Look Once (YOLOv3) based on the Custom dataset for detecting and recognizing the objects in the indoor environment such as offices or rooms. OpenCV library is used for capturing and processing the images. YOLOv3 detects and recognizes the objects in each image. In addition, the Playsound library is used for playing the sound to tell the user about the objects in each frame with their locations (Center, Left, or Right). The proposed system is implemented on a personal computer with the Python programming language. Figure 2 shows the steps of the proposed object detection and recognition system.

The following steps are employed for detecting and recognizing the objects by using the proposed system:
**Step 1:** OpenCV library was used to operate the webcam for the purpose of capturing the frames (images).
**Step 2:** OpenCV was used to resize each frame (image) to 416x416.
**Step 3**: If there are objects in the frame, YOLOv3 will detect and recognize these objects depending on the weight file. However, if the frame has no objects, then the next frame is selected. The weight file is generated from the training process.
**Step 4:** When YOLOv3 detect and recognize the objects, their locations are calculated depending on the width of the frame and the center point of these objects.
**Step 5:** Playsound library is used to play the sound from sounds dataset, to allow the user to know about the objects that are found in the frame with their locations (Right, Center or Left).

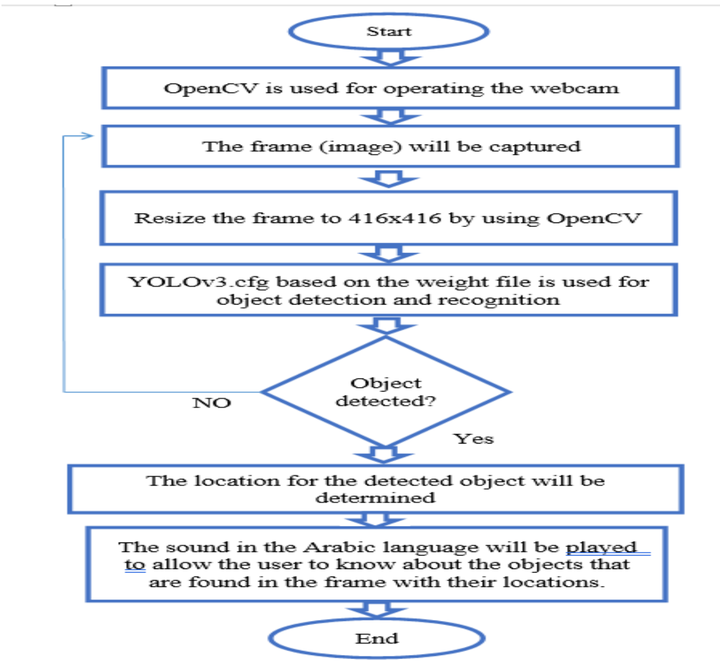


Figure 2. The proposed system's flowchart

**OpenCV**

OpenCV which means Open source Computer Vision is a library that is used for image processing. Image processing is a type of signal processing in which an image is the input and the output is also an image or set of characteristics related to the image. OpenCV was started as a research project by Intel. It combinesdifferent tools to solve the problems of computer vision. [29].

**You Only Look Once (YOLOv3)**

YOLOv3 consists of Convolution neural network (CNN) and an algorithm for processing outputs from the network [7]. YOLOv3 is a real-time, multi-object detection and fast method. YOLOv3 outperforms the other algorithms because of itshigh processing. YOLOv3 applies a single CNN to an entire image, divides the image into S x S grid, predicts the bounding boxes, and finds the probabilities for these boxes [13]. YOLOv3 is consisted of 106 layers.The objects can be detected at 3 different scales (small, medium, and large). Figure3 shows the layers of YOLOv3.

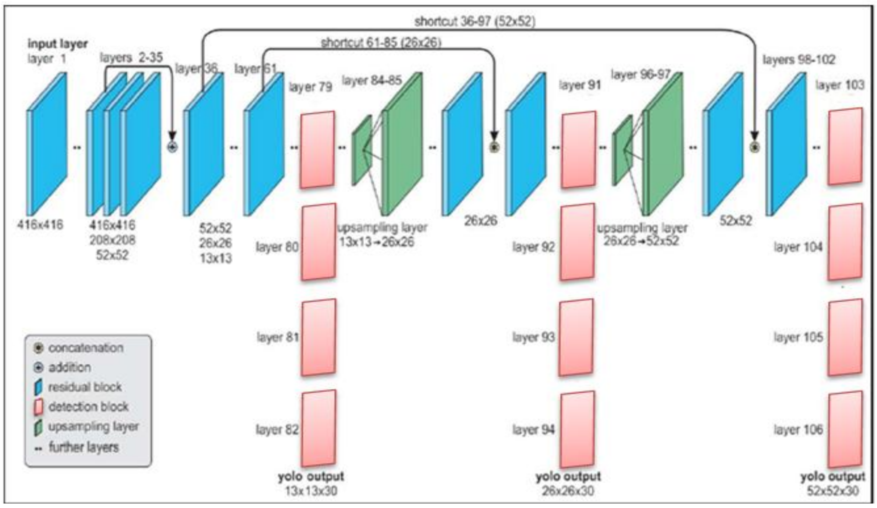


Figure 3. YOLOv3

The YOLOv3 algorithm works as follows:

- YOLOv3 takes the image (frame) from the camera and analyzes this frame for detecting and recognizing the objects. Then, it divides the entire frame into S x S grids. Each grid may have single or multiple objects. These objects are to be bounded by a bounding box within each grid. So, there may be B bounding boxes for each grid.
- A confidence score is given (40% or above) depending on which bounding boxes are predicted against the probability of C classes. The predictions with an invalid confidence score are not projected.

- There are five values: x, y, w, h, and confidence for each boundary box prediction. The center of the box is represented by (x, y). w represents the width of the box and h represents the height. The x, y, w, and h are between [0, 1]. There are six class probabilities for each cell but only one class probability is predicted per cell. The final prediction is of the form S * S * (B * 5 + C). Only one object can be detected per cell of the grid.
- Only one object can be detected by a grid cell. Therefore, YOLOv3 uses an anchor box to detect multiple objects. Consider the image in Figure 4. In this image, the midpoints of the human and the car are in the same grid cell. Therefore, an anchor box was used. The purple color grid cells denote the two anchor boxes for those objects. Any number of anchor boxes can be used to detect multiple objects in a single image. In this image, two anchor boxes are used [16].

Figure 4. Anchor Boxes

- If the same object is included in two or more grid cells, then the center point of the object is determined and the grid that has the center point of the object is selected. Multiple bounding boxes are generated around the objects; there are two methods that can be used to solve this problem. These methods are Intersection over Union (IoU) and Non-Max Suppression (NMS). In the IoU method, when the value of intersection over union is equal to or more than a threshold value then the prediction is good. By increasing the threshold value, the accuracy will be increased [16]. In the Non-Max Suppression, the boxes that have a high probability are taken and the boxes with a high IoU are suppressed. Repeat this process until a box is selected and consider that as the bounding box for that object [6].

**Determining the Locations of Objects**

After applying the NMS method, there is one bounding box for each detected object in the image. Each bounding box has five values x, y, w, h, and confidence. (x,y) is the center point of the bounding box. The 416x416 image has a width of 416. Depending on the width of the image and the center point of the bounding box, the location of the object can be determined if it is Right, Center or Left.

**Custom Dataset (Images Dataset)**

For training the deep learning model, a lot images data are required [30]. The prepared dataset of the proposed system is consisting of 180 labeled images for six objects (Table, Person, TV, Bottle, Chair and Laptop). There are 30 images for each object. These images have different sizes and they are in the .jpg format. Figure 5 illustrates some of these images.

Figure 5. The Custom Dataset

**Sounds Dataset**

The set of sounds in the Arabic language iscreated and stored in PC. If the object is detected and recognized, the Playsound library will play the sound to allow the user to know about the objects in each frame with their locations. The sounds are in MP3 format. In this method, the text will be converted into sound at high speed and without using the Internet.

**The Training Process**

The training process for the proposed neural network was executed on the GPU of Google Colab. The following are the steps for the training process.

**Step 1:** A set of high-resolution and colored images of different sizes is collected.

**Step 2:** LabelImg was used to give label for each object in the image. The LabelImg is an application used for labeling the objects in the image.

**Step 3:** The prepared dataset is ready to train the neural network. The dataset that consists of 180 color images was split into two groups. The first group comprises 85% of the total images as the training images and the second group includes the remainder, which is 15% of the total images as the testing images. The training process was executed on Google Colab and it took nearly three hours. The neural network was trained for 3000 iterations.

**Step 4:** the weight file is created at the end of the training process.

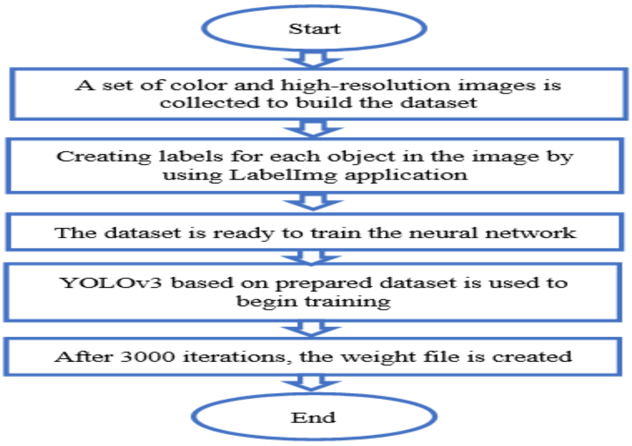


Figure 6. The steps of the training

**RESULTS AND ANALYSIS**

YOLOv3 is a fast and accurate algorithm, used to achieve good accuracy and high speed. The proposed system can be applied to the indoor environment such as rooms or offices and it can detect multiple objects. In addition, it can detect the objects even if the distance between the objects and the webcam is greater than three meters. The system can detect six objects categories. The mean Average Precision (mAP), which was used to evaluate the performance of the proposed system, was 100%. Figure 7 explains the loss and mAP for the proposed system. Other metrics were used to evaluate the performance of the proposed system such as IoU (intersection over Union), recall, precision, F1-score, TP, FP, and FN. Figure 8 shows some of the performance metrics obtained. Figure 9 illustrates the object detection through the webcam.

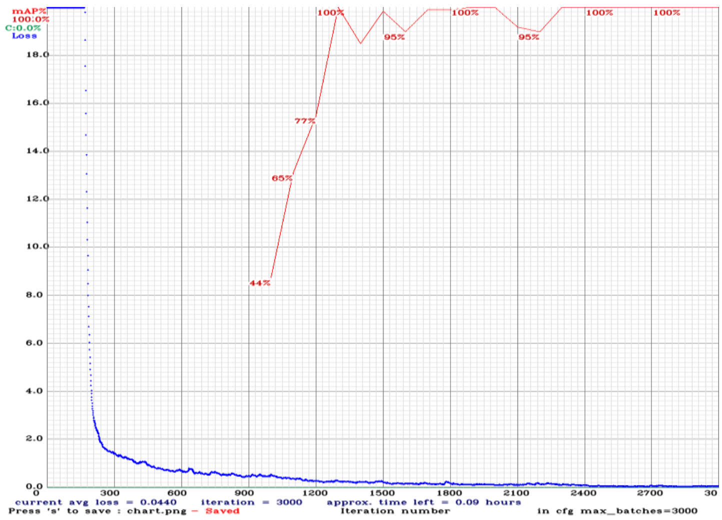


Figure 7. Loss and mAP (mean Average Precision)

```
class_id = 0, name = TV, ap = 100.00%          (TP = 5, FP = 0)
class_id = 1, name = person, ap = 100.00%      (TP = 4, FP = 0)
class_id = 2, name = bottle, ap = 100.00%      (TP = 4, FP = 0)
class_id = 3, name = chair, ap = 100.00%       (TP = 4, FP = 0)
class_id = 4, name = Table, ap = 100.00%       (TP = 7, FP = 0)
class_id = 5, name = Laptop, ap = 100.00%      (TP = 4, FP = 0)

 for conf_thresh = 0.25, precision = 1.00, recall = 1.00, F1-score = 1.00
 for conf_thresh = 0.25, TP = 28, FP = 0, FN = 0, average IoU = 87.42 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 1.000000, or 100.00 %
```

Figure 8. Performance metrics obtained

## 1.1 The Confusion Matrix

The confusion matrix is a summary that gives the results of the prediction on the classification problem. The numbers of correct and incorrect predictions are summarized with counted values and broken down class by class. The confusion matrix shows how your model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but also more importantly the types of errors that are being made.

For TV (Class_Id = 0) TP = 5 and FP = 0
For Person (Class_Id = 1) TP = 4 and FP = 0
For bottle (Class_Id = 2) TP = 4 and FP = 0
For Chair (Class_Id = 3) TP = 4 and FP = 0
For Table (Class_Id = 4) TP = 7 and FP = 0
For Laptop (Class_Id = 5) TP = 4 and FP = 0

The total TP =28
The total FP = 0

True Positive (TP)isthe number of correctly detected objects.False Positive (FP) is the number of incorrect detections.False Negative (FN) is the number of missed detections.

**Precision**

Precision measures how accurate your predictions are. It is calculated as:

The number of True Positive (TP) divided by the sum of True Positive (TP) and False Positive (FP), as given in Eq. (1).

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \qquad (1)$$

The precision value obtained was 1.00.

**Recall**

Recall is used to calculate the true predictions from all correctly predicted data.It is calculated as: The number of True Positive (TP) divided by the sum of True Positives (TP) and False Negative (FN), as illustrated in Eq. (2)

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \qquad (2)$$

The Recall value obtained was 1.00.

**F1-score**

F1-score is the HM (Harmonic Mean) of precision and recall. The value of the F1-score obtained was 1.00.

**Average Intersection over Union (IoU)**

The area of overlapping (intersection) is divided by the area of union between the ground truth bounding box and the detection bounding box for a certain threshold. Eq. (3) explains the average IoU.

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \times 100\% (3)$$

The average IoU for the proposed system was 87.42%.

**Mean Average Precision (mAP)**

The mAP is the average of the Average Precision (AP) calculated for all the classes, as indicated in Eq. (4).

$$\text{mAP} = \frac{\text{sum of AP for the total classes}}{\text{no.of total classes}} \times 100\% \qquad (4)$$

In particular, the mAP for the proposed system was 100%.

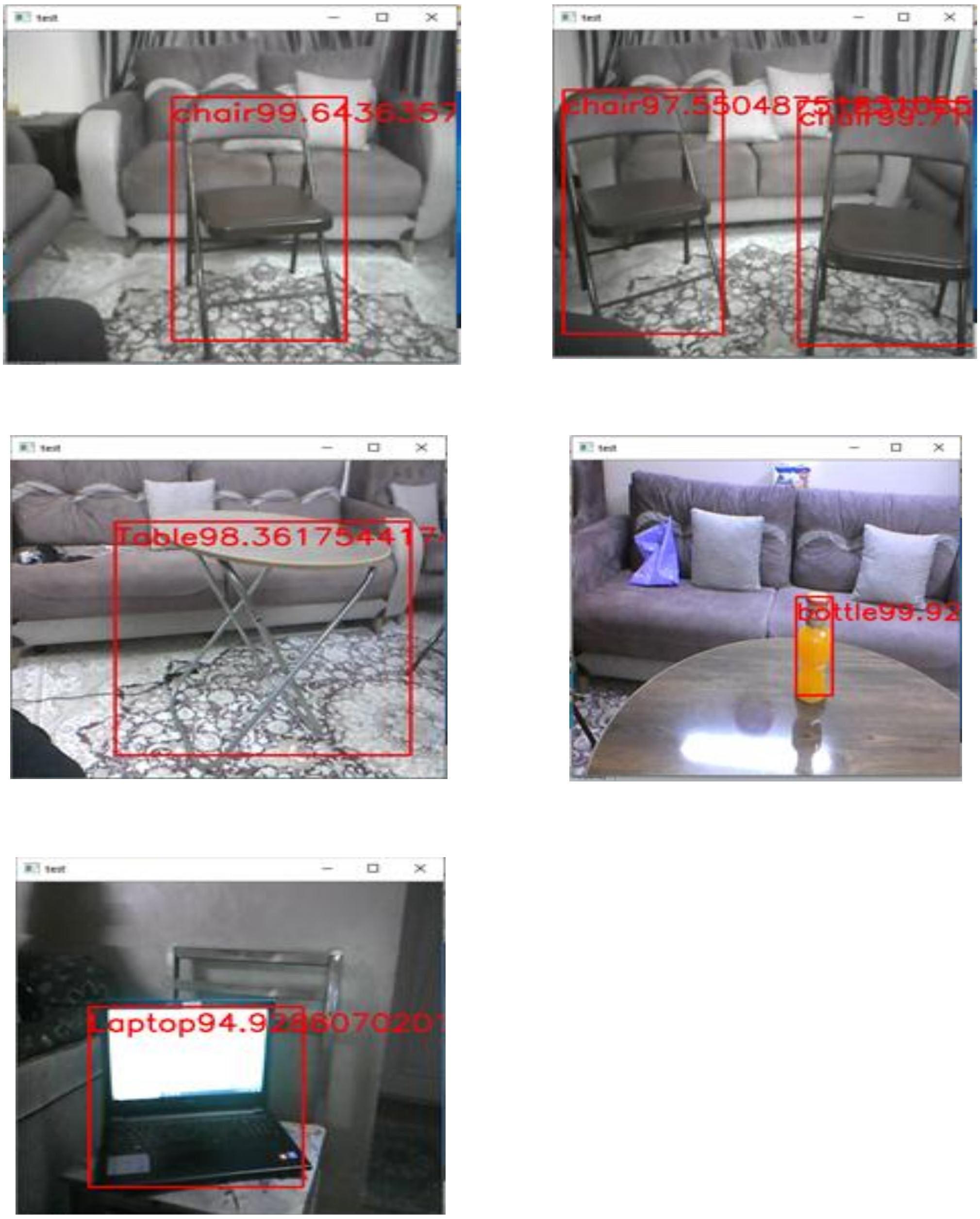


Figure 9. Object detection through the webcam

**CONCLUSION**

Many algorithms are used for detecting and recognizing the objects such as YOLO, CNN, Fast R-CNN and R-CNN. YOLOv3 was used because it is an accurate and fast and method, which can detect and determine the location of objects in real-time. The proposed system does not need the connection to the internet for text-to-sound conversion as it uses the Playsound to play the sound from sounds dataset. The achieved accuracy for the proposed system is 99%. The required time for detection on the PC was nearly two seconds. In the future, the system can be implemented on Raspberry Pi to build smart eyeglasses for assisting blind people in detecting and recognizing the objects in their environment. Moreover, the age and gender prediction techniques can be added to this system to predict the age and gender. The face recognition technique can also be added to the system to recognize the person.

**Table 1:** The comparison between some methods from the literature review and the proposed method

| Paper | Method | Accuracy | No. of objects | Detection Time |
|---|---|---|---|---|
| Aleksa Ćorović, et al. (2018) [7] | YOLOv3 based on Berkley Deep Drive dataset | 63% | 5 | Not Reported |
| O. Masurekar, et al. (2020) [16] | YOLOv3 based on custom dataset | 98% | 3 | 8 sec |
| Sunit Vaidya, et al. (2020) [17] | YOLOv3 based coco dataset | 85.5% in Android<br>89 % in web | 80 | 2 sec |
| A. S. Mahmoud, et al. (2019) [19] | Mask RCNN and NWPU-VHR-10 dataset are used | 95% | 10 | 7.1 sec |
| Sajidah S. Mahmood, et al. (2020) [23] | CNN based on custom dataset | 92% | 1<br>vehicle | Not Reported |
| I. A. M. Zin, et al. (2020) [24] | Convolutional Neural Network with custom dataset | 99% | 12 | Not Reported |
| M. Anandhalli, et al. (2021) [25] | CNN based on custom dataset | 90.88% | 1<br>(vehicle) | Not Reported |
| Our proposed system | YOLOv3 based on custom dataset | 99% | 6 | 2 sec |

## References


V. Kharchenko and I. Chyrka, "Detection of Airplanes on the Ground Using YOLO Neural Network," *Int. Conf. Math. Methods Electromagn. Theory, MMET*, vol. 2018-July, pp. 294–297, 2018, doi: 10.1109/MMET.2018.8460392.

Z. Cheng, J. Lv, A. Wu, and N. Qu, "YOLOv3 Object Detection Algorithm with Feature Pyramid Attention for Remote Sensing Images," *Sensors Mater.*, vol. 32, no. 12, p. 4537, 2020, doi: 10.18494/sam.2020.3130.

H. Jabnoun, F. Benzarti, and H. Amiri, "Object detection and identification for blind people in video scene," *Int. Conf. Intell. Syst. Des. Appl. ISDA*, vol. 2016-June, pp. 363–367, 2016, doi: 10.1109/ISDA.2015.7489256.

S. N. Srivatsa, G. Sreevathsa, G. Vinay, and P. Elaiyaraja, "Object Detection using Deep Learning with OpenCV and Python,", International Research Journal of Engineering and Technology (IRJET), Volume 8, Issue 1, pp. 227–230, 2021.

M. Shah and R. Kapdi, "Object detection using deep neural networks," *Proc. 2017 Int. Conf. Intell. Comput. Control Syst. ICICCS 2017*, vol. 2018-Janua, pp. 787–790, 2017, doi: 10.1109/ICCONS.2017.8250570.

S. Geethapriya, N. Duraimurugan, and S. P. Chokkalingam, "Real time object detection with yolo," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 3 Special Issue, pp. 578–581, 2019.

A. Corovic, V. Ilic, S. Duric, M. Marijan, and B. Pavkovic, "The Real-Time Detection of Traffic Participants Using YOLO Algorithm," *2018 26th Telecommun. Forum, TELFOR 2018 - Proc.*, 2018, doi: 10.1109/TELFOR.2018.8611986.

K. Potdar, C. D. Pai, and S. Akolkar, "A Convolutional Neural Network based Live Object Recognition System as Blind Aid," *arXiv*, 2018.

P. S. Bhairnallykar, A. Prajapati, A. Rajbhar, and S. Mujawar, "Convolutional Neural Network ( CNN ) for Image Detection," *International Research Journal of Engineering and Technology (IRJET)*, Vol. 7, no.11, pp. 1239–1243, 2020.

T. S. Gunawan *et al.*, "Development of video-based emotion recognition using deep learning with Google Colab," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 5, pp. 2463–2471, 2020, doi: 10.12928/TELKOMNIKA.v18i5.16717.

N. Mittal, A. Vaidya, and A. Shreya Kapoor, "Object Detection and Classification Using Yolo," *International Journal of Scientific Research & Engineering Trends*, vol. 5, no. 2, 2019.

C. Tung *et al.*, "Large-Scale Object Detection of Images from Network Cameras in Variable Ambient Lighting

Conditions," *Proc. - 2nd Int. Conf. Multimed. Inf. Process. Retrieval, MIPR 2019*, pp. 393–398, 2019, doi: 10.1109/MIPR.2019.00080.

M. Burić, M. Pobar, and M. Ivašić-Kos, "Adapting yolo network for ball and player detection," *ICPRAM 2019 - Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, no. Icpram, pp. 845–851, 2019, doi: 10.5220/0007582008450851.

A. Abdurrasyid, I. Indrianto, and R. Arianto, "Detection of immovable objects on visually impaired people walking aids," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 2, pp. 580–585, 2019, doi: 10.12928/TELKOMNIKA.V17I2.9933.

R. Bharti, K. Bhadane, P. Bhadane, and A. Gadhe, "Object Detection and Recognition for Blind Assistance," International Research Journal of Engineering and Technology (IRJET), Volume: 06, Issue: 05, pp. 7085–7087, May 2019.

O. Masurekar, O. Jadhav, P. Kulkarni, and S. Patil, "Real Time Object Detection Using YOLOv3," *Int. Res. J. Eng. Technol.*, vol. 07, no. 03, pp. 3764–3768, 2020.

S. Vaidya, N. Shah, N. Shah, and R. Shankarmani, "Real-Time Object Detection for Visually Challenged People," *Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020*, no. Iciccs, pp. 311–316, 2020, doi: 10.1109/ICICCS48265.2020.9121085.

Shifa Shaikh, Vrushali Karale and Gaurav Tawde, "Assistive Object Recognition System for Visually Impaired," *Int. J. Eng. Res.*, vol. V9, no. 09, pp. 736–740, 2020, doi: 10.17577/ijertv9is090382.

A. S. Mahmoud, S. A. Mohamed, R. A. El-Khoribi, and H. M. AbdelSalam, "Object detection using adaptive mask RCNN in optical remote sensing images," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 1, pp. 65–76, 2020, doi: 10.22266/ijies2020.0229.07.

A. A. Fahad, H. J. Hassan, and S. H. Abdullah, "Deep Learning-based Deaf & Mute Gesture Translation System," vol. 9, no.5 , pp. 288–292, May 2020, doi: 10.21275/SR20503031800.

E. K. Elsayed, D. R. Fathy "Semantic Deep Learning to Translate Dynamic Sign Language," vol. 14, no. 1, pp. 316–325, 2021, doi: 10.22266/ijies2021.0228.30.

A. A. Abdulhussein and F. A. Raheem, "Hand Gesture Recognition of Static Letters American Sign Language (ASL) Using Deep Learning," *Eng. Technol. J.*, vol. 38, no. 6A, pp. 926–937, 2020, doi: 10.30684/etj.v38i6a.533.

S. S. Mahmood and L. J. Saud, "An Efficient Approach for Detecting and Classifying Moving Vehicles in a Video Based Monitoring System," *Eng. Technol. J.*, vol. 38, no. 6A, pp. 832–845, 2020, doi: 10.30684/etj.v38i6a.438.

I. A. M. Zin, Z. Ibrahim, D. Isa, S. Aliman, N. Sabri, and N. N. A. Mangshor, "Herbal plant recognition using deep convolutional neural network," *Bull. Electr. Eng. Informatics*, vol. 9, no. 5, pp. 2198–2205, 2020, doi: 10.11591/eei.v9i5.2250.

M. Anandhalli, V. P. Baligar, P. Baligar, P. Deepsir, and M. Iti, "Vehicle detection and tracking for traffic management," *IAES Int. J. Artif. Intell.*, vol. 10, no. 1, p. 66, 2021, doi: 10.11591/ijai.v10.i1.pp66-73.

A. A. Naufal, C. Fatichah, and N. Suciati, "Preprocessed Mask RCNN for Parking Space Detection in Smart Parking Systems," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 6, pp. 255–265, 2020, doi: 10.22266/ijies2020.1231.23.

R. F. Rachmadi, I. K. E. Purnama, S. M. S. Nugroho, Y. K. Suprapto "Family-Aware Convolutional Neural Network for Image-based Kinship Verification," vol. 13, no. 6, pp. 20–30, 2020, doi: 10.22266/ijies2020.1231.03.

R. Parmar and R. Lakshmanan, "Detecting diabetic retinopathy from retinal images using CUDA deep neural network," *Int. J. Intell. Eng. Syst.*, vol. 10, no. 4, pp. 284–292, 2017, doi: 10.22266/ijies2017.0831.30.

M. Naveenkumar and V. Ayyasamy, "OpenCV for Computer Vision Applications," *Proc. Natl. Conf. Big Data Cloud Comput.*, no. March 2015, pp. 52–56, 2016.

F. Guo, K. Hao, M. Xia, L. Zhao, L. Wang, and Q. Liu, "*Detection of insulator defects based on YOLO V3"*, vol. 287. Springer International Publishing, 2019.

## Author Information

**Hassan Salam**
University of Technology
Baghdad, Iraq

**Hassan Jaleel**
University of Technology
Baghdad, Iraq

**Salma Hameedi**
University of Technology
Baghdad, Iraq