IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Design and Development of Android Application for Visually Impaired People Using Object Detection and Direction prediction

[1]Dr. Kalyan Bamane, [2]Simran Khaparde, [3]Harshita Totala, [4]Yuvraj Singh, [5]Dnyaneshwar Ghule

[1]Associate Professor, [2,3,4,5]Student
[1]Information Technology,
[1]D Y Patil College of Engineering, Akurdi, Pune, India

Abstract: Vision is one of the most important senses that a human possesses. It helps humans to understand and analyze their surroundings and take the appropriate actions depending on their surroundings. This task might seem easy; however, visually impaired people find it very difficult to understand their surroundings and move around them. Therefore, this project focuses on the design and development of an android application for the visually impaired people that will make their lives comparatively easier than what it is today. In addition to object detection, the application will also provide the feature of direction prediction, which uses sensor data and machine learning algorithms to guide users with audio cues on the most appropriate path or direction to take.

Index Terms - Object Detection, Deep Learning, Visually Impaired, YOLO, OpenCV, Image Processing

INTRODUCTION

With the increase in the digital technology increased the potential to empower individuals with visual impairments and enhance their daily lives. This project introduces an Android application that leverages the YOLO (You only look once) item detection version and OpenCV (Open-source pc vision Library) to deal with the particular demanding situations confronted by visually impaired individuals. The utility is designed to offer real-time item detection and direction prediction skills, improving visually impaired people’s independence and mobility. The YOLO model, renowned for its performance and accuracy in object detection, is seamlessly integrated with OpenCV to create a powerful device.

This Android Software utilizes the device's digicam to capture the person's surroundings, processing the stay video feed to perceive and classify items in actual-time. Auditory or haptic comments are then provided to the person, enabling them to understand and navigate their surroundings higher. Moreover, this software is going a step similarly by incorporating a path prediction characteristic. by way of analyzing the positions of detected items and the use of sensor facts, the application courses users through audio cues to take the most suitable route, making sure safe and green navigation.

This comprehensive solution is aimed toward promoting inclusivity and autonomy for visually impaired people by means of harnessing deep getting to know, pc vision, and real-time sensor records. The Android software provided on this venture serves as an effective device to bridge the gap between visible impairment and an international increasingly reliant on technology, in the end empowering customers to navigate and engage with their surroundings with a bit of luck and independently.

LITERATURE SURVEY

1. **Real-Time Object Detection Using Yolo Algorithm for Blind People**
   Prof. Pradnya Kasture, Akshay Tangade, Aditya Pole, Aishwarya Kumkar, Yash Jagtap

2. **A study on Real Time Object Detection using Deep Learning**
   Pradyuman Tomar, Sagar, Sameer Haider

3. **Real Time Object Detection with YOLO**
   Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam

4. **Object Detection System for Blind with Voice Guidance**
   Miss Rajeshvaree Ravindra Karmarkar, Prof.V.N. Honmane

5. **Real Time Deep Learning Based Object Detection Framework**
   William Tarimo, Moustafa M.Sabra, Shonan Hendre

6. **Real Time Object Detection Using YOLOv3**
   Omkar Masurekar, Omkar Jadhav, Prateek Kulkarni, Shubham Patil

7. **Object Detection Method Based on YOLOv3 using Deep Learning Networks**
   Vidyavani , K. Dheeraj, M. Rama Mohan Reddy, KH.Naveen Kumar

8. **Real-Time Object Detection for Visually Challenged People**
   Sunit Vaidya, Naisha Shah, Niti Shah, Prof. Radha Shankarmani

9. **Real Time Object Detection with Audio Feedback using YOLO vs YOLO v3**
   Mansi Mahendru1, Sanjay Kumar Dubey

10. **You Only Look Once: Unified, Real-Time Object Detection**
    Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi.

11. **Object Detection and Recognition in Real Time using Deep Learning for Visually Impaired People**
    M.I. Thariq Hussan, Saidulu Dorepalli

PROPOSED WORK

3.1 Object Detection Algorithm

Object detection is the principal objective of this system being developed. It consists of object classification and object localization. Object detection is the process of sorting the objects into predefined categories. In other words, object classification assigns a label to an entire image. That label is the name of the object currently in that image. For example, give the computer a picture of a horse and it will try to split it and output "horse". We easily identify objects in an image, but classifying objects is a difficult task for computers. In the imaging domain, the computer attempts to separate objects from the image by drawing a rectangular box (also called a bounding box) around the object.

Thus, object detection is the combination of object classification and object localization in which we try to classify and isolate multiple objects present in the image. The output of this module will give us the name of the object and the coordinates of the bounding box. Next part of the model, the direction of the object will be determined using the coordinates of these bounding boxes and the distance between the object and the user will be calculated.

Algorithm selection is most important for effective use. Thus, different object detection algorithms such as R-CNN, Fast R-CNN and YOLO are compared. R-CNN uses a region based proposed method. R-CNN does not take the whole image, instead, it only takes the part of the image that has a higher chance of containing the object. The training time of the network is very large.
Moreover, it cannot be used in real time as its speed is very slow, it takes 47 seconds for each image to get detected. The speed and accuracy of Fast R-CNN are better than R-CNN. In Fast R-CNN, there is no need to feed 2000 regions every time to the convolution layer; instead, it is passed only once per image which provides a convolutional feature map.

In this system, the YOLO (You Only Look Once) algorithm is used which is the best fit for Real-time detection applications. YOLO, when compared to different detection algorithms, works differently. The YOLO algorithm takes the entire image (frame) in a single turn and processes it. The feature that makes YOLO different from other algorithms is its excellent processing speed where it can process 45 frames per second. Figure 1 shows us comparison of various object detection algorithm with respect to speed.

1. Comparison between different object detection algorithms

Algorithm	**Speed**	
R-CNN	.05FPS	20 s/img
Fast-RCNN	.5FPS	2 s/img
Faster-RCNN	.7FPS	140 ms/img
YOLO	45FPS	22 ms/img

Table.1 Comparison of Algorithms
[*Note:* Reprinted from "Real-time object detection for visually challenged people." by Vaidya, Sunit, et al, 2020 4th (ICICCS) IEEE, 2020.]

Challenges for Object Detection Algorithm:

Speed-Accuracy Trade-off: Achieving real-time performance while maintaining high detection accuracy is a significant challenge in object detection. Faster algorithms tend to sacrifice accuracy, while more accurate algorithms are often slower.

Hardware Constraints: Real-time object detection systems must run efficiently on hardware with limited computational resources, such as CPUs, GPUs, or specialized accelerators like TPUs (Tensor Processing Units) or FPGAs (Field Programmable Gate Arrays).

Robustness to Environmental Variability: Real- world environments present various challenges, including changes in lighting conditions, occlusions, cluttered backgrounds, and object deformations. Object detection models need to be robust to these factors to perform reliably in diverse settings.

3.2 YOLO Algorithm

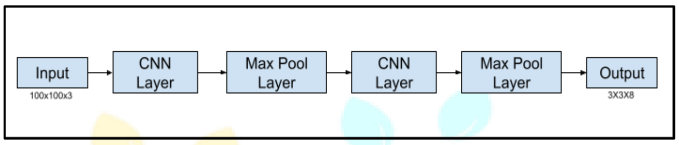


Fig 1. YOLO Architecture

YOLO is a new approach to detect multiple objects present in an image in real-time whilst drawing bounding boxes around them. It passes the image through the CNN algorithm only once to get the output, thus it is named YOLO. Although comparatively it is similar to R-CNN, YOLO practically runs a lot faster than Faster R-CNN because of its simpler architecture. Unlike Faster R-CNN, YOLO can classify and perform bounding box regression simultaneously. With YOLO, the class label containing objects, their location can be predicted in one glance. Completely deviating from the typical CNN pipeline, YOLO treats the object detection task as a regression problem by dimensionally separating bounding boxes and their related class probabilities, which are predicted using a single neural network. The process of performing bounding box prediction and class probability calculations is a unified network architecture that was initially introduced by YOLO. YOLO's architecture is similar to a typical convolutional neural network inspired by the Google Net model for image classification.
Figure 2 describes us the architecture of YOLO. The operation of this model can be described as follows: It begins by resizing the input image to a 448x448 resolution prior to being fed into its interface. Initially, a 1x1 convolution is employed to decrease the channel count, followed by the utilization of a 3x3 convolution to further reduce the channel dimensions.

The model ultimately generates a cubic output. Across the network, the primary activation function utilized is Rectified Linear Unit (ReLU), introducing non-linearity to capture intricate data patterns. However, in the final layer of the model, a linear activation function is used, typically suited for tasks involving regression or numerical output.

To enhance the model's training and prevent it from overfitting, it includes additional techniques such as batch normalization and dropout. These methods contribute to stabilizing the model's performance and preventing overfitting on the training data.

3.3 Working of YOLO Algorithm

YOLO stands for You Only Look Once and is one of the most popular object detection algorithms. YOLO combines a multi-step process, using a single neural network to perform both classification and prediction of bounding boxes for detected objects in an image. As such, it is better in terms of search performance and faster than running two different neural networks to identify and classify objects separately. It does this by re modelling traditional image classifiers to be used for the regression task of identifying bounding boxes for objects. YOLO divides an input image into an N × N grid. If the center portion of an object falls into the boundary of one of the grid cells, then that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those particular boxes. These confidence scores show how accurate the model is about the box containing an object and how accurate it assumes the predicted box to be.

The YOLO algorithm predicts many bounding boxes per grid cell. During the training time, we want only one bounding box to be responsible for predicting an object. YOLO appoints one predictor of the bounding box to be "responsible" for predicting an object based on the highest current IOU along with the ground truth. This leads to segregation between the bounding box predictors. Each predictor gets better at predicting certain sizes, aspect ratios, or classes of objects, improving the overall recall score overtime.

The main technique used in the YOLO model is non-maximum likelihood (NMS). NMS is a post-processing step used to improve the accuracy and performance of target detection. In object detection, multiple checkboxes are usually created for an object in an image. These boxes may overlap or be in different locations, but they all represent the same thing. NMS is used to identify and remove empty or invalid boxes and display one box for each object in the image.

The YOLO algorithm processes 45 frames per second so it is incredibly fast and it is also efficient because it can predict more than one object from an image.

The YOLO algorithm is designed to estimate the class of an object and the bounding box that defines the object's position in the input image. As shown in figure 3, it represents the bounding box for a particular object.

It recognizes each bounding box using four parameters:

- Centre of a bounding box denoted by (bx ,by)
- Width denoted by (bw)
- Height denoted by (by)
- Value c corresponds to a class of an object (i.e. Person, cell phone, cup, bicycle, etc.)
- Predicted value pc is a probability of an object in the bounding box.

$$Y = (P_C, B_X, B_Y, B_H, B_W, C)$$

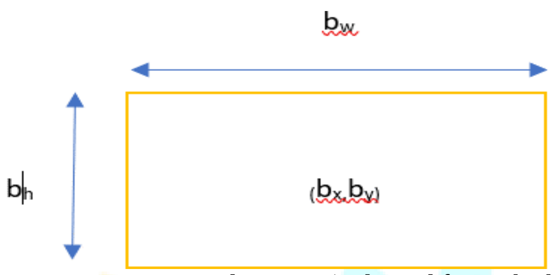


Fig. 2 Boundary Box (Indicated for each object)

[*Note*: Reprinted from "Real-time object detection for visually challenged people." Vaidya Sunit, 2020 4th (ICICCS). IEEE, 2020]

The confidence score can be calculated using the formula: C = Pr(object) * IoU
IoU: Intersection over Union between the predicted box and the ground truth. If no object exists in a cell, its confidence score should be zero. Each bounding box predicts a set of four values(x,y,w,h) in addition to their corresponding confidence value, with each grid predicting a category which is recorded by C, thus the result is S*S*(5*B+C) dimension for an S*S image.
When testing the category that each grid predicts is multiplied by a confidence value, which is predicted by the bounding box corresponding to final category of bounding box, represented by:

$$\Pr(\text{Class}(i) \mid \text{object}) * \Pr(\text{object}) * IOU$$

Once the final class score is obtained, the threshold is determined, subregions are filtered, and additional frames are processed by the non-limiting (NMS) algorithm to obtain the final result.

If we have a picture with two boxes representing a cat and a dog. The first step YOLO takes is to split the image into grids. For example, in figure 4 we have passed a sample image to our model to get a 3×3 grid:

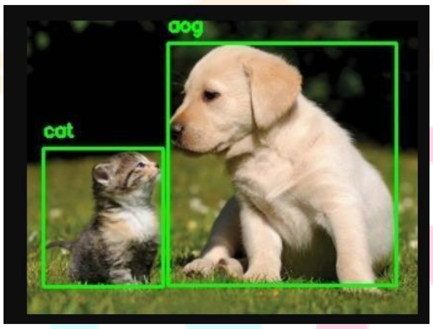


Fig. 3 Boundary Box (As detected by our prototype)

The presence of a grid enables us to detect an object per grid rather than detecting one object per image. For each grid cell present we can encode a vector which describes the cell. For example, the first cell in the above image from the top-left doesn't have any object,

$$C_{1,1} = (P_c, Bx, By, Bw, Bh, C_1, C_2) = (0,?,?,?,?,?,?)$$

where Pc is the probability of the object class, Bx and By are coordinates of the center of the bounding box, relative to the cell, Bx and By are width and height of the bounding box relative to the whole image, C1 and C2 are 0 or 1 depending on which class represents the bounding box (C1 for cat and C2 for dog). Vector (C1,1) consists of symbols? because if the first component Pc is equal to zero, then the rest of the components can have random numbers are they are not taken into consideration. YOLO is capable of predicting all objects at once and therefore it is named You Only Look Once.

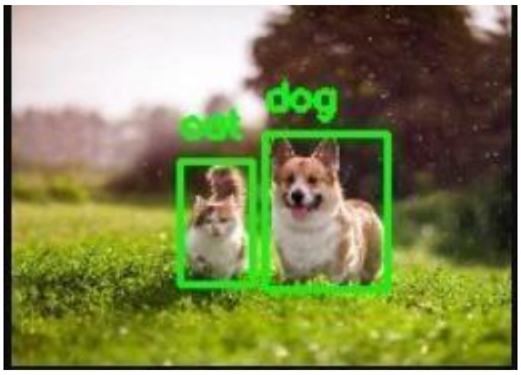


Fig. 4 Boundary Box (As detected by our prototype)

In figure 5, this is an issue that can occur if the algorithm predicts multiple connected boxes for a class. We can only choose one box with the highest probability for each class, but what if there are multiple objects in the same class in the picture (like multiple cats)? First, we select the box with the highest possible value. We then use Intersection Over Union (IoU) to compare the box to all other boxes in that group. In general, this index is also called the Jaccard index, but in computer vision the name IoU is used. The formula of IoU is:

$$IoU = \frac{\text{area of the intersection between } B_1 \text{ and } B_2}{\text{area of the union between } B_1 \text{ and } B_2},$$

where B1 and B2 are two bounding boxes.
If the value of IoU is higher than a predefined threshold (e.g. 0.5), then boxes with lower probability are excluded. This means that two boxes that have higher IoU values may represent the same object in the image, so we remove the box with the lower value. Repeat this process until all of the boxes are estimated to be occupied or excluded.

3.4 Direction Calculation

The direction of the object will be informed to the visually impaired person in the following way: “to the left”, “to the right”, “at the center”. The direction will help the user to determine the exact location of the object, and will help the user to navigate in a better way. Let’s take a look at figure 6 n order to determine the accurate direction, we will divide the image into 3 parts along the length of the screen. The ratio of these lengths are 3:4:3. The portion of the image in which the center of the object lies will determine the accurate direction of the object. For instance, if the center of the object lies in the first portion of the screen, its direction will be to the LEFT. The coordinates of the center can be calculated using the coordinates of the bounding box. Only the x coordinates of the bounding box are required to determine the x coordinate of the center.

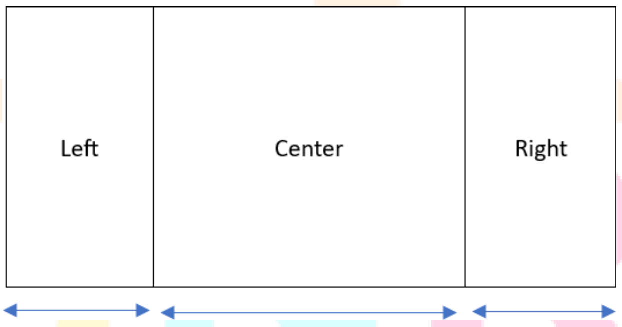


Fig. 5 Direction Determination Method

[*Note*: Reprinted from “Android based object detection system for visually impaired” by Badave, A., Jagtap, R., Kaovasia, R., Rahatwad, S., & Kulkarni, S. (2020, February (I4Tech) (pp. 34- 38). IEEE.]

3.5 Audio Output

Providing the audio output will be the last stage of the system. This audio output will be providing:

1] The name of the object

2] The direction of the object

For instance, the audio output message is as follows: “A bike is at your right”. The information provided to the visually impaired user through the audio message will help him to identify the objects, the directions and also the obstacles around him. It can be conveyed through the phone’s speakers or earphones. In order to generate the audio output, we will be using the Text to Speech library provided in Android Studio. The audio output can also be provided to the user in other local languages such as Hindi and English. The output provided by different modules in this application is discrete. For instance, the object detection module will give the object name as its output; the direction calculation module will give the direction. It is crucial to combine these discrete values into one meaningful sentence that the user can interpret easily. We need to ensure that the audio output is not provided for the same object twice in a row. This will ensure that the user gets to know about all the objects around him.

DATASET

COCO Dataset stands for Common Objects In Context. It is a predefined dataset by Microsoft. The COCO dataset is a collection of demanding, high-quality datasets for computer vision, mostly using state-of-the-art neural networks. This name is also used to refer to the format in which the datasets are stored. It is an object detection, segmentation, and captioning dataset. It contains around 330k images in which more than 200k images are labelled, which makes it even easier to recognize the class (category) of detected objects. It has around 1.5 million object instances and 80 object categories. COCO annotations employ the JSON file format, which has a top value of dictionary (key-value pairs inside braces). It can also have nested dictionaries or lists (ordered collections of objects inside brackets), as shown below:

```
{"info": {…},
"licenses": […],
"images": […],
"categories": […],
"annotations": […]}
```

In figure 6, we have a graph of MS COCO dataset on various object detection algorithm, calculated with respect to frames per second.

Info Section: It contains metadata about the dataset like description, url, version etc.

Licenses section: It contains the links to the licenses for the images present in the dataset. All the license contains the id field which is used to recognize the license.

Image: It is the second most important dictionary of the dataset. It has the fields like licence, file_name, coco_url, height, width and date_captured.

Categories Section: It contains classes of the objects that may be detected on images.

Annotations Section: This is the most important section of the dataset, which contains information vital for each task for specific COCO dataset.

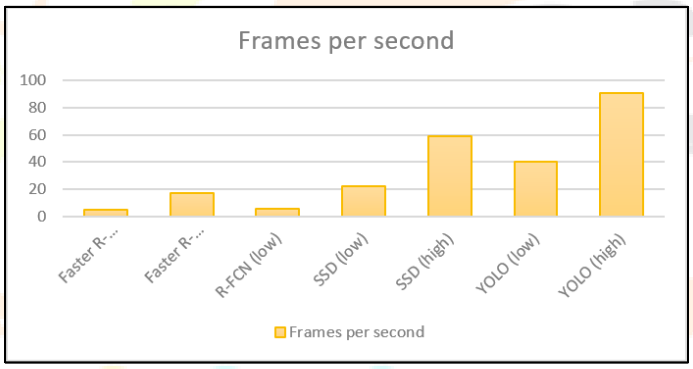


Fig. 6 MS COCO Dataset Performance on Various Object Detector Algorithms

[*Note*: Reprinted from “A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework”]

DESIGN DIAGRAM

The system architecture of our proposed work is depicted in Figure 7, showcasing various modules dedicated to implementing the model.

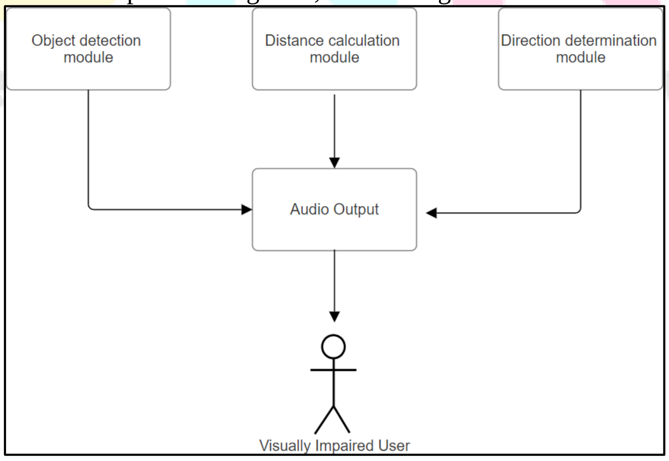


Fig. 7 System Architecture

[*Note:* Reprinted from “Android based object detection system for visually impaired” by Badave, A., Jagtap, R., Kaovasia, R., Rahatwad, S., & Kulkarni, S. (2020, February), International Conference on Industry 4.0 Technology (I4Tech) (pp. 34-38). IEEE.

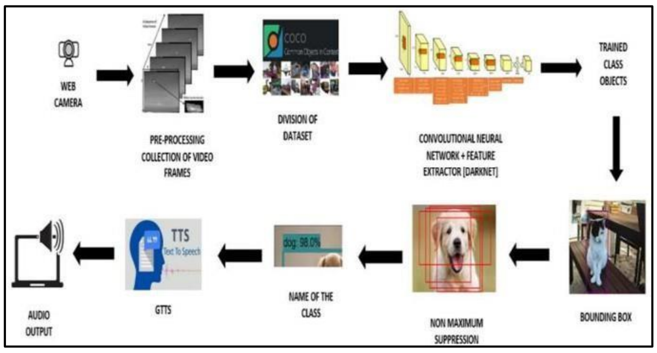


Fig. 8 Design Diagram

[*Note*: Reprinted from "Object Detection with Voice Guidance to Assist Visually Impaired Using Yolov7" by Boobalan, Dr & S, Bhuvanikha & M, Sivapriya & R, Sivakumar. (2023). International Journal for Research in Applied Science and Engineering Technology. 11. 764-768. 10.22214/ijraset.2023.50182]

APPLICATION SCREENSHOTS

Following figure, no 9, no 10, no 11, no 12 shows the real time object detection of various objects from our YOLO model.

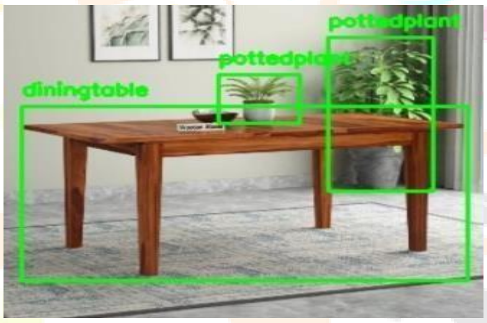


Fig. 9 Objects Detected by Algorithm

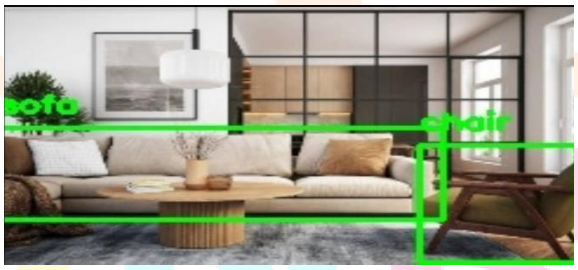


Fig. 10 Objects Detected by Algorithm

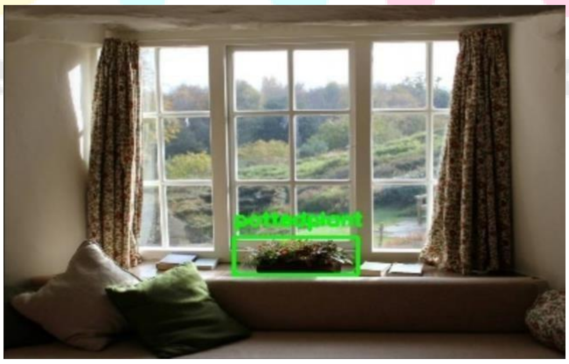
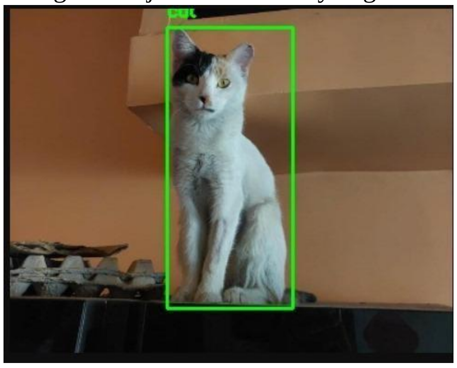


Fig. 11 Objects Detected by Algorithm

Fig. 12 Objects Detected by Algorithm



COMPARISON OF YOLO VERSIONS

Currently, for the implementation of android application for visually impaired people we have chosen YOLO v4 version. Previously existing applications or tools have mostly chosen other versions of YOLO and majorly YOLO v3. Figure 12 shows the comparison of YOLO V4 with other algorithms in real time. Also figure 15 summaries the comparison of YOLO V4 with other version of YOLO with respect to mAP.\

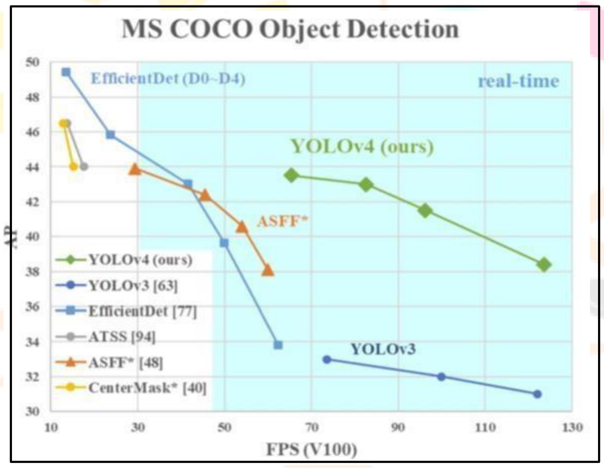


Fig. 13 Comparison of YOLO v4 and other algorithms

[Note: Reprinted from "YOLOv4: Optimal Speed and Accuracy of Object Detection" by Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao, arXiv preprint arXiv:2004.10934 (2020)]

2. Comparison between different object detection algorithms

YOLO	Backbone	Neck	Head	mAP
V1	24-layered CNN	NM Suppression	SoftMax	0.37
V2	DarkNet-19	NM Suppression	SoftMax	0.83
V3	DarkNet-53	FPN	Log. Regr	0.93
V4	CSPDarkNet53	PANET	Yolo v3	0.97
V5	CSPDarkNet53	PANET	Yolo v3	0.97

Table 2 Comparison of YOLO v4 and other algorithms based on mAP

[*Note:* Reprinted from "Melanoma Lesion Detection and Segmentation Using YOLOv4-DarkNet and Active Contour" by Albahli, S., Nida, N., Irtaza, A., Yousaf, M. H., & Mahmood, M.T. (2020), *EEE access*, *8*, 198403-198414.]

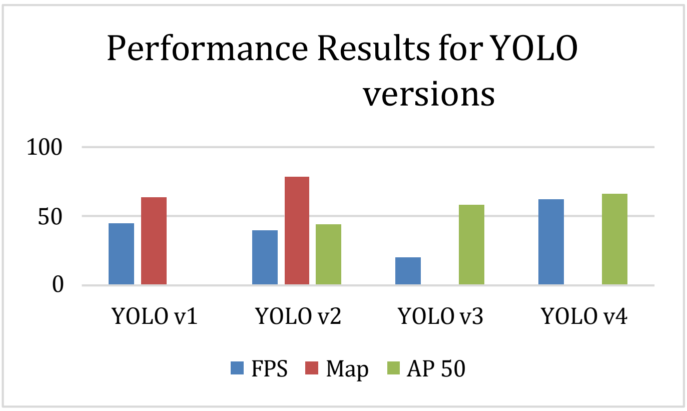


Fig. 14 Performance results for different YOLO Versions based on existing models

FUTURE SCOPE

The application that we are developing is very easy to navigate and interpret for the visually impaired people. When the user turns on the application the camera will automatically start capturing the real time video. As soon as the user presses a button, the server- side backend algorithm will start processing it and notify the user accordingly as output audio. The Yolo algorithm can be stopped by pressing the same button again. This is the way in which the objects around the visually impaired people will be detected along with the objects' direction and distance. This output will be provided as an audio output to the people. However, the accuracy can be increased. Also, the current system works on android Operating system which can be further developed in order to be compatible on ever convenience device.

The model will work accurately for the MS COCO dataset. However, for future work, this model shall be aimed at giving the same and more accurate result for self-explored datasets and many other customizable datasets.

The positions of the objects only have 3 different criteria for height and width which gives 9 different position possibilities. This can be further improved to give a more accurate positioning of the object in the future stages. If the objects are hidden by obstacles in front of them, then they are not captured in the camera and not detected. This will also be considered in future stages. The accuracy of detection in darkness should also be improved. The distance of the object from the camera is also a feature that can be incorporated in the next stage.

Outdoor navigation assist can also be implemented. The system can use the phone's GPS to determine the location during the travel and the directions can be obtained with the help of Google Maps API. Currently, our system provides output in the English language. This feature can be extended to include multiple other languages that will enable users from all the counties to use this system.

CONCLUSION

In the past few years, many solutions have been devised to help visually impaired people orientate and recognize objects in their surroundings. However, most of these systems cannot be used in the daily lives of visually impaired people due to the high cost of the infrastructure required to implement these systems. Our goal is to develop a system that will help the visually impaired to recognize objects around them indoors and to move independently outdoors. In order for a user to use our system, a visually impaired user does not need to learn any specific skills. Moreover, the cost of the proposed system is minimal, as only a smartphone is needed to implement it. Hardware such as sensors are not used in our object detection system, unlike other proposed systems, thus reducing the overall system cost. A prototype application has been developed that currently has all the features listed in this article except for the distance calculation feature. Together, with the distance calculation function, we will retrain the TensorFlow object detection model to detect more objects.

REFERENCES

1. Mahendru, Mansi, and Sanjay Kumar Dubey. "Real time object detection with audio feedback using Yolo vs. Yolo_v3." 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2021

2. 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.

3. Tarimo, William, Moustafa M. Sabra, and Shonan Hendre. "Real-time deep learning based object detection framework." 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2020.

4. Redmon, Joseph, et al. "You only look once: Unified, real- time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

5. Kasture, Prof & Tangade, Akshay & Pole, Aditya & Kumkar, Aishwarya. (2021). RealTime Object Detection Using Yolo Algorithm for Blind People. International Journal of Advanced Research in Science, Communication and Technology. 152-156. 10.48175/IJARSCT-1191.

6. Karmarkar, Rajeshvaree Ravindra, and V. N. Hommane. "Object detection system for the blind with voice guidance." Int. J. Eng. Appl. Sciences Technol 6.2 (2021): 67-70.

7. Masurekar, Omkar, et al. "Real time object detection using YOLOv3." International Research Journal of Engineering and Technology (IRJET) 7.03 (2020): 3764-3768

8. Geethapriya, S., N. Duraimurugan, and S. P. Chokkalingam. "Real-time object detection with Yolo." International Journal of Engineering and Advanced Technology (IJEAT) 8.3S (2019): 578- 581.

9. Tomar, Pradyuman, and Sameer Haider. "A Study on Real Time Object Detection using Deep Learning.". International Journal of Engineering Research Technology (IJERT) 11.05 (2022).

10. Badave, Ajinkya, et al. "Android based object detection system for the visually impaired." 2020 International Conference on Industry 4.0 Technology (I4Tech). IEEE, 2020.

11. Alsultan, Omar Kanaan Taha, and Mohammad Tarik Mohammad. "A Deep Learning-Based Assistive System for the Visually Impaired Using YOLO-V7." Revue d'Intelligence Artificielle 37.4 (2023): 901-906.

12. International Journal of Computational Intelligence Systems

13. Android based object detection system for visually impaired. In 2020 International Conference on Industry 4.0 Technology (I4Tech) (pp. 34-38). IEEE. Badave, A., Jagtap, R., Kaovasia, R., Rahatwad, S., & Kulkarni, S. (2020, February).

14. Object Detection with Voice Guidance to Assist Visually Impaired Using Yolov7. International Journal for Research in Applied Science and Engineering Technology. 11. 764-768. Boobalan, Dr & S, Bhuvanikha & M, Sivapriya & R, Sivakumar. (2023)

15. Vidyavani, A., et al. "Object detection method based on YOLOv3 using deep learning networks." International Journal ofInnovative Technology and Exploring Engineering (IJITEE) 9.1

16. YOLOv4: Optimal Speed and Accuracy of Object Detection
Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan.(2020).

DATA AVAILABILITY

The data supporting this research are openly available from International Journal for Research in Applied Science & Engineering Technology (IJRASET) at https://doi.org/10.22214/ijraset.2023.50182. Additionally, any supplementary material or relevant information needed to reproduce the reported results is available at the URL https://towardsdatascience.com/yolo-v4-optimal-speed-accuracy-for-object-detection-79896ed47b50] or can be obtained from the corresponding author upon request.

COMPETING INTERESTS

The authors declare that they have no financial interests or personal relationships that could have influenced the research reported in this paper. All other authors declare no conflicts of interest.

AUTHOR'S CONTRIBUTIONS

Dr. Kalyan D Bamane contributed to helping with the research and development of application. Simran Khaparde conceived and designed the study and created the required content and research data. Harshita Totala and Yuvraj Singh collected and analyzed the data, Dnyaneshwar Ghule interpreted the results.