CICERONE- A Real Time Object Detection for Visually Impaired People

Therese Yamuna Mahesh¹, Parvathy S S ², Shibin Thomas³, Shilpa Rachel Thomas ⁴ and Thomas Sebastian⁵

¹Associate Professor, Dept. of Electronics & Communication Engineering, Amal Jyothi College of Engineering, Kanjirappally, Kerala, India
²³⁴,5B-Tech Electronics and Communication Engineering, Amal Jyothi College of Engineering, Kanjirappally, Kerala, India
thereseyamunamahesh@amaljyothi.ac.in
parvathyss2020@ec.ajce.in
shibin0598@gmail.com
shilparachelthomas2020@ec.ajce.in
thomassebastian2020@ec.ajce.in

Abstract. Our work provides a solution to people who are suffering from partial blindness due to diseases of the eye or due to accidents. Unlike people who are born blind, they depend a lot on other people for doing their day to day activities. Our project is a boon to such people as our end product; a smart walking stick detects the trained objects that are used daily by the person and intimates the person using audio messages. This project changes the visual world into an audio world by informing the visually impaired of the objects in their environment. These people can use this particular prototype for self-navigating their way. A YOLO (You Look only once) algorithm is used in our project. Real-time objects in an image are detected with their names represented on a bounding box and these names are converted to speech signals. The conversion to audio signals is done by using an e-Speak tool which forms Google’s Text to Speech (gTTS) system. The prototype consists of several modules. The Raspberry Pi camera module takes the image and transfers it to the Raspberry Pi desktop. Then, real-time object detection is carried out by using YOLO network. The detected image is converted to speech by using the gTTS module and the audio result is provided to the user through a headset. For achieving the required portability, the battery backup is being used.

1. INTRODUCTION

Millions of people living in the world today suffer from visual impairment, and the main reasons are, age-related issues, diabetes and accidents. Blindness or vision impairment, one of the top ten disabilities among men and women. Challenges faced by blind and partially sighted people include loss of flexibility & spontaneity, loss of employment, inability to carry out household task, inability to shop alone, time & effort required to accomplish simple task. The new technologies in the market are now trying to help them lead a normal life. Even though new products are rapidly being developed, they suffer from several self-navigation issues due to which social awkwardness follows. Some of the technologies before are NAVI, Navigation Assistance for the Visually Impaired could differentiate between background and the obstacle, VOICE, the system converts the image into audio, so that the user can identify an obstacle, TVS (Tactile Vision System) and Tyflos systems have a device at the user’s abdomen. It converts the depth into vibrations. These systems are quite slow and not practically viable.

Here, the proposed prototype is connected on top of a walking stick and an image is captured by using the pi camera connected to a Raspberry Pi mounted onto the stick. Intelligent object detection is performed by using YOLO (You Look Only Once) [9], [10] algorithm, a new and state-of-the-art method to detect objects by regression and classification. There are different object detection techniques but are not as accurate as YOLO. The image passes through a neural network only once and provides the output of all detected objects [11]. The
next phase involves text to speech conversion which is implemented using Google’s gTTS module (Google text-to-speech) and thus artificially producing human speech. With the aid of a headset the visually impaired person receives the required information. As a result, a camera based effective self-navigation can be provided.

2. LITERATURE SURVEY

Much like guide dogs, that are trained to guide visually impaired people in a particular locality, use of a smart walking stick would help them to avoid obstacles in their path. The camera-based system in [1] helps to recognize text on the hand held objects. The proposed system consists of camera, to capture the object of interest and an algorithm to extract the text from the background. The optical character recognition (OCR) separates each letters of the detected text. The detected text is given as an audio output with the help of software development kit. The wearable device in [2] is a system that accepts input from the user and detects the objects. The device contains ultrasonic sensor that helps to alert the user about the object in his/her path. Haar cascade algorithm is used to detect the objects. It is a wearable device and placed on the chest of the user. The detected object is given back to the user as audio.

The object detection system in [3] is used to detect objects in the traffic scenes. Here they have used the combination of optimized you only look once (OYOLO), which is 1.18 times faster than YOLO and R-FCN (Regression based Full Convolution Network). It is used to detect and classify the images such as cars, cyclist and pedestrian. Use of YOLO makes location errors, to avoid that we use OYOLO. Paper [4] presents a prototype that extracts the text from image and is converted to speech. Extraction of text is done by using the Tesseract Optical Character Recognition (OCR). This method is carried out by using Raspberry Pi. Text recognition is done by using Open Computer Vision (Open CV), considering the large library of functions when compared to MATLAB. Capturing of image is done by using a portable camera and the image is converted to gray scale and filtered by Gaussian filtering. Then it is binarized and cropped. The cropped image is given to Tesseract OCR. The e-Speak creates an analog signal of the text and is given to the headset.

3. METHODOLOGY

The aim of our project is to design a system at a low cost to enable a person who’s vision is not clear to lead a normal life without depending on others in a selected environment like our house, workplace etc. All the things that are used for the day to day activities can be detected using a walking stick with the raspberry pi attached. The process is as shown in the flowchart in Figure 1.

![Figure 1. Flow chart of the proposed system.](image)

Raspberry pi 4 model B is used for acquiring the images. The acquired images are transferred to the Google drive using rclone. YOLO network is used for object detection and classification. The test to speech conversion of the detected class is done using gTTS (Google Text to Speech module) to be sent to the users earphones.

4. RASPBERRY PI CONFIGURATION

Initially, to get started with the raspberry pi we need the operating system. NOOBS (New Out of the Box Software) is a preconfigured and an easy OS installer manager for the Raspberry Pi. NOOBS include the
following operating system

• Raspbian,
• Pidora,
• OpenELEC,
• RaspBMC,
• RISC OS

Installation procedures for NOOBS are as follows:

1. NOOBS is available for download on the official Raspberry Pi website [5].
2. Next, we have to download and install SD card formatter software into windows OS in order to make the SD card compatible for installing NOOBS.
3. Format the SD card using the above software in FA format. Now, micro SD card is ready for use with the Raspberry Pi.
4. Next, we have to upgrade and update the operating system with the latest version of Raspberry Pi. For that the following two commands are initiated:
   i. sudo apt-get update;
   ii. sudo apt-get upgrade
5. After the above procedures, we have successfully configured the Raspbian operating system

![Raspberry pi camera setup.](image)

Figure 2. Raspberry pi camera setup.

5. YOLO ALGORITHM

YOLO algorithm [6],[8] is primarily used for the prediction of bounding boxes accurately from an image. Images are divided into N x N grids and for each grid the prediction of the bounding boxes are done as well as the class probabilities. After performing object localization and image classification for each grid of the input image, each grid is given a different label. The following project is designed as a YOLO algorithm applied separately on each grid and the objects in it are marked with their particular label and corresponding bounding boxes are also highlighted. The grids having no object are labeled as 0.

Initially, YOLO algorithm is applied to the received image. In our project, the real time image is divided into grids of matrices. As the image complexity varies the image can be split into any number of grids. After division of the images, both classification and the localization process are done on each grid containing the object. The confidence score is computed for all the grids. The confidence score and also the bounding box for each of the grid will differ based on whether the object is detected or not. For no object it displays the value as 0 and displays the value 1 if object exist. Bounding box value will show how confident the network is, that is, how much the detected object matches to the object under observation. The prediction of bounding box is illustrated below.

6. EXPERIMENTAL PROCEDURE

The experimental procedure started with collecting the database, training the YOLO network and then testing the output by giving test images. The network was trained to detect 5 classes namely C1 (class 1) for bottles, C2 (class 2) for watches, C3 (class 3) for keys, C4 (class 4) for pens and C5 (class 5) for glasses.

6.1. Collection of Data Base

The example of dataset collected for class 4 pen is shown in figure 3. Similar datasets were collected for other classes.
6.2. Network Output Analysis

Consider for example, input images being processed in batches. The output must be considered as a vector onto which we are mapping the features or simply a feature map. If $N$ is the number of bounding boxes and $C$ is the number of classes the detector is detecting on, then, these $N$ bounding boxes are responsible for detecting different objects. Several bounding boxes are output with corresponding class names and each bounding box consists of six parameters $(p_c, bx, by, bw, c)$. Therefore, each of these bounding boxes contain $(C+5)$ characteristics that provide relevant information about the bounding boxes such as presence/absence of objects, dimensions of anchors, detected class and corresponding confidence scores also. Since the custom object detector in the following project is trained on only 5 classes, each bounding box is represented with 10 parameters as $(C=5)$ classes.

In YOLO v3, 3 anchor boxes are computed for every cell. The expectation behind feature maps is to predict the desired objects using anchor boxes provided. The object center point coincides with that particular cell. The defining concept of YOLO v3 is that only a single bounding box must be responsible for detecting a particular object. For detection, the center point of the object is evaluated and is mapped to the feature map. The grid cell to which the center point falls on is responsible for detecting the object. Next, for each anchor box a probability is computed that indicates the certainty that the box contains a certain class. It can be done as shown in figure 5.

The generated bounding box dimensions are evaluated using simple mathematical formulas. The anchor box width and height are predicted from neighboring clustering centroids and the center coordinates can be obtained using a sigmoid function. Transition from network output to the bounding box generation is done as follows:

$$b_x = \sigma(a_x) + i_x = \sigma(a_x) + i_x$$  \hspace{1cm} (1)

$$b_y = \sigma(a_y) + i_y = \sigma(a_y) + i_y$$  \hspace{1cm} (2)

$$b_w = (q_w) \cdot e^{aw} = (q_w) \cdot e^{aw}$$  \hspace{1cm} (3)

$$b_z = (q_z) \cdot e^{az} = (q_z) \cdot e^{az}$$  \hspace{1cm} (4)
Here, $b_x$ and $b_y$ are the (x, y) center coordinates and $b_w$ and $b_z$ are the bounding box width and height respectively. The parameters $a_x$, $a_y$, $a_w$, and $a_z$ are the output of the network. The top left most cell coordinates are denoted as $i_x$ and $i_y$, $q_w$ and $q_z$ are the anchor box dimensions.

### 6.3. Bounding Box Dimensions

The sigmoid function is deployed to normalize the output values between 0 and 1. The actual coordinates of the bounding box are not computed by YOLO, only the offset values are determined relative to the top left most corner of the grid cell responsible for detecting the object.

6.4. Output Filtering

The process described above yields a large number of bounding boxes which in turn may cause multiple detections of the same objects in an image. To overcome such issues, the technique of Non-Maximum Suppression (NMS) is deployed which neglects those generated anchor boxes having low confidence scores based upon a pre-defined threshold value (cut off value). That is, if the threshold value is chosen as 0.5 then all generated anchor boxes having a confidence score below 0.5 are discarded. Despite filtering the bounding boxes using threshold values, the network might still end up with the condition of overlapping boxes, as a result, an additional function of NMS known as Intersection over Union (IoU) is used which acts as a second filter for selecting the right boxes. IoU can be implemented as follows:

- An anchor box is defined with two corners, that is the top left most corner and the bottom right most corner ($a_1$, $b_1$, $a_2$, $b_2$)
- Area of the box is computed by multiplying the height ($b_2 - b_1$) and the width of the box ($a_2 - a_1$).
- Next step is locating the coordinates of the intersection of two anchor boxes. ($a_1'$, $b_1'$, $a_2'$, $b_2'$) as follows:
  \[
  a_1' = \text{Min } \{a_1\} \text{ coordinates of box1 and box2} \\
  b_1' = \text{Min } \{b_1\} \text{ coordinates of box1 and box2} \\
  a_2' = \text{Min } \{a_2\} \text{ coordinates of box1 and box2} \\
  b_2' = \text{Min } \{b_2\} \text{ coordinates of box1 and box2}
  \]
- IoU can be illustrated as follows:
  \[
  \text{Intersection Area} = (a_2' - a_1') \times (b_2' - b_1') - - - - -(5)
  \]
Therefore, \[ \text{Union Area} = [\text{box1 area} - \text{box2 area}] - \text{Intersection Area} \] (6)

Therefore,

\[ \text{IoU} = \frac{\text{Intersection Area}}{\text{Union Area}} \] (7)

Figure 8. Input image before NMS and after NMS

6.5. Training

- The resolution of the training images was pre-adjusted in order to keep a constant stride value.
- The entire network was trained for 10000 epochs (2000*no. of classes) with the following specifications:

| Training Specifications | Value |
|--------------------------|-------|
| Batch size               | 64    |
| Subdivisions             | 16    |
| Channels                 | 3     |
| Momentum                 | 0.9   |
| Decay                    | 0.0005|
| Angle                    | 0     |
| Saturation               | 1.5   |
| Exposure                 | 1.5   |
| Hue                      | 1     |
| learning rate            | .001  |
| Policy                   | steps |
| Steps                    | 8000,9000 |

Training was performed on the collected dataset by using open source software known as LabelImg [11] which is a graphical image annotation tool. The software is written in Python and uses python libraries such as Qt for its graphical interface. The annotations are saved as XML files in both the PASCAL VOC format and YOLO format. For this project, we have opted it to be saved in the YOLO format.
7. TEXT TO SPEECH
The main aim of this project is to get the audio output of the detected text. For recognition of the text an optical character recognition (OCR) engine is used which is called the Tesseract-OCR. So, the text file extracted from the Tesseract, needs to be converted to audio which is performed by Python gTTS module, which is a text to speech engine for converting the text words that is extracted by using Tesseract-OCR to speech form. The complete process is executed on the Raspberry Pi as a compact hardware design. The detection of text from the image and text to audio conversion has an accuracy of 85% by taking random samples. The Python interface for Google’s Text to Speech API (Application Package Interface), Google Text to Speech (gTTS) is used. gTTS module produces an mp3 file. The audio conversion by making use of gTTS is much faster. The mp3 file obtained using gTTS can be heard using the play sound, which is an open source - based sound player compatible with python. Play sound is a sound player for the Raspberry Pi which is command line-based.

8. RESULT
The YOLO network was trained to detect five customized objects. The input scenario is as shown in the figure 9 and the objects detected by the YOLO network are shown in figure 10.

![Figure 9. Input image containing several objects](image1)

![Figure 10. Desired objects detected from several objects](image2)

As seen from the figures 10, the custom Yolo v3 architecture was able to detect the 5 classes it had been trained on. Since the network had been trained for an epoch of 10000, the confidence score of the detected object is relatively high.

| Classes | Confidence score |
|---------|------------------|
| Pen     | 99%              |
| Watch   | 98%              |
| Key     | 96%              |

Table 2. Confidence score for figure 19 (three objects detected).

Higher confidence value resulted from exponential decrease of the loss curve where an ideal detector model is required to have a loss value of less than two 2.

After running the code for gTTS we obtain the audio file which is given to the user.

9. CONCLUSION
The main objective of this project is to develop an object detector that is capable of detecting customized classes. This is implemented for five classes of different objects which the user is likely to use daily. The proposed system was implemented using a camera, which captures image, raspberry pi for interface and Bluetooth for the audio output. Ultrasonic sensor is connected to the walking stick for the detection of obstacles. YOLO leads to faster run time [7] and the performance measure using IoU is above 90%. The future scope includes complete independence to the partially blind person both indoors and outdoors by incorporating cards with GPU in the hardware and cloud based execution of the network.
REFERENCES

[1] Samruddhi Deshpande and Revati Shriram “Real time text detection and recognition on hand held objects to assist blind people” International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT) International Institute of Information Technology (IIT), Pune, 2016

[2] Deepthi Jain B, Shwetha M Thakur and K V Suresh “Visual assistance for blind using image processing” International Conference on Communication and Signal Processing, April 3-5, India, 2018

[3] Jing Tao, Hongbo Wang, Xinyu Zhang, Xiaoyu Li and Huawei Yang, “An object detection system based on YOLO in traffic scene”, 6th International Conference on Computer Science and Network Technology (ICCSNT), 2017

[4] Rajesh M, Bindhu K Rajan, Ajay Roy, Almaria Thomas K, Ancy Thomas, Bincy Tharakan T and Dinesh C, “Text recognition and face detection aid for visually impaired person using raspberry pi”, International Conference on Circuits Power and Computing Technologies (ICCPCT), 2017

[5] Installing operating system in Raspberry pi available at https://www.raspberrypi.org/downloads

[6] Geethapriya. S, N. Duraimurugan and S.P. Chokkalingam “Real-Time Object Detection with Yolo”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8, Issue-3S, February 2019

[7] Deepa,Tamilselvan, Abrar and Shrinivas Sampath“Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for Action Decision Networks”, International Conference on Advances in Computing and Communication Engineering (ICACCE), 2019

[8] Joseph Redmon*, Santosh Divvala*, Ross Girshick and Ali Farhadi “You Only Look Once: Unified, Real-Time Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition, 2016

[9] R.Rajalakshmi,K.Vishnupriy,M.S.Sathyapriya and G.R.Vishva-ardhini “Smart Navigation System for the Visually Impaired Using Tensorflow”, LIARIIE-ISSN(O)-2395-4396, Vol-4 Issue-2 2018

[10] Prof. Seema Udgir , Shivaji Sarokar2 , Sujit Gore , Dinesh Kakuste and Suraj Chaskar “Object Detection System for Blind People”, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 9, September 2016

[11] Installation process of labelImg is available at https://github.com/tzutalin/labelImg