Smart Navigation System for the Visually Impaired Using Tensorflow

Mrs.R.Rajalakshmi[1], Ms.K.Vishnupriya[2], Ms.M.S.Sathyapriya[3],Ms.G.R.Vishvaardhini[4]

[1] *Asst. Professor, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

[2] *Student, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

[3] *Student, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

[4] *Student, Dept. of Electronics and Communication Engineering, Panimalar Engineering College, Tamil Nadu, India*

ABSTRACT

The project aids the visually impaired to navigate independently using real time object detection and identification. The proposed system consists of a Raspberry Pi-3 processor enabled with cloud features and loaded with a pre-trained Convolutional Neural Network model(CNN) developed using tensorflow. The processor was connected to a camera and an ultrasonic sensor. The processor was coded in python, a high level programming language, needed to process images in real time. The camera captures an image in real time which was processed by the Raspberry Pi-3 processor. The python code uses the CNN model to identify the obstacle with boxes and category index. The recognized image category was extracted and stored in a text file. The ultrasonic sensor finds the distance at which the obstacle is present. This distance was also appended into the text file. The contents of the text file are converted to voice using the Google Text To Speech module. This system is extremely flexible and can be used in any environment, without any priming.

Keyword: *tensorflow, Raspberry Pi 3, Convolutional Neural Network, ultrasonic sensor, Python*

1.Introduction

The ability to navigate from place to place is a significant part of daily life. Human beings process the world around them mostly via the sense of sound and vision. It is general belief that vision plays a critical role, but many would have great difficulty in identifying the visual information they use, or when they use it. We find it easy to navigate in extremely familiar places without the sense of vision. This is possible mostly due to muscle memory. This can be experienced in examples such as going to the bathroom from your bedroom in the middle of the night. But only a small minority of people have experienced navigating large-scale, unfamiliar environments without the aid of their eyes. Consider trying to catch a train in Bandhra railway station blindfolded at peak hours. Yet, the visually challenged travel independently on a daily basis. To facilitate safe and efficient navigation, blind individuals must acquire travel skills and use sources of nonvisual environmental information that are rarely considered by their sighted peers. Their sense of smell and their hearing are very sharp, as they rely a lot upon these senses. They also take to feeling the environment around them. This is harmless in the confines of a home. However in an unfamiliar surrounding this could be quite hazardous. How does one avoid running into the low-hanging branch over the sidewalk, or falling into the open manhole? When one walks down the street, how do they know when they have reached the medical store, the cafe, or their friend’s house? The purpose of this chapter is to highlight some of the navigational technologies available to blind individuals to support independent travel. With advancement in technology a lot of navigation systems have come up to aid the visually impaired. However, they remain inaccessible to many. In many cases, they are simply too expensive.

In this context, we propose a smart system which will help the blind navigate easily. The system is low cost and would be focusing on blind navigation in large-scale, unfamiliar environments. The technology discussed can also be used in well-known spaces and is also useful to those with low vision. Additionally, it can also be used by people with both low vision and low hearing ability.

2. Ease of Use

The use of the suggested system is very simple and easy. The system consists of a raspberry pi, a camera and an ultrasonic sensor. The total weight of the system, when made into prototype form will weigh less than a kilogram. The camera will be mounted on a wearable glass, the ultrasonic sensor could be attached to the belt and the Raspberry Pi can be placed in any convenient location on the user. Once the system is switched on it would continuously provide information about the surrounding.

3.Related Work

As mentioned earlier there are a lot of research and work going towards making navigation easier for the visually challenged. The use of bionic eyes are revolutionary. Despite so much effort being put into it, many in the blind community still rely on their memory or the kindness of others for their day to day locomotion.

All the existing navigation systems have ultrasonic sensors as a part of the suggested modules. The ultrasonic sensor proves to be more useful than the IR sensor, as it has a wide range and is much more accurate. In the 1990s, Echolocation [1], was designed. It consisted of two ultrasonic sensors placed upon a glass, which was worn by the user. Objects were differentiated with the help of a different ultrasonic frequencies. An audio output was given directly to the user via head phones. The processing was carried out by a microprocessor. The system was very simple and portable which was very advantageous. However, there were many blind spots, since it used only sensors. This could prove very dangerous to the user, relying solely on it. The Navbelt [2], was an advancement over the echolocation. It used an obstacle sensor robot, eight ultrasonic sensors, instead of two, and a computer. In 1992, this was a complete guidance system. This proved to be a very good improvement. The computer generated a map of angles to locate the obstacles. It produced a series of output audio beeps. The disadvantage in this system was the rigorous training the user had to undergo to understand each audio beep associated with a type of object. Recently this kind of system has been used in canes. They also provide an alternate path to avoid the obstacle. But all these systems are heavy to carry and have limits on the scanning range.

The UCSC [3] and the CyARM [4] are quite similar to each other. The CyARM makes use of the concept of producing tension in a string. Again the use of ultrasonic sensors is taken to detect the presence of obstacles and the distance of them from the user. Low tension is provided when obstacle is close by and high tension is provided when obstacle is far off. The UCSC project was very popular and a lot of companies produced commercial navigation systems using this idea. The computer used an extended version of Kalman filter tracker to detect areas which were crucial for mobility. But they were not very reliable and the cost was very high.

With the development of Digital Image Processing, cameras were considered to be the next big thing. There were numerous possibilities and researches began to exploit them. The NAVI [5], Navigation Assistance for the visually impaired could differentiate between background and the obstacle. This was achieved by capturing the image via a camera and using complex algorithms. In VOICE [6], the system converts the image into audio, so that the user can identify an obstacle. The TVS (Tactile Vision System) [7] and Tyflos [8] systems have a device at the user's abdomen. It converts the depth into vibrations. Various others combine pressure sensors along with the ultrasonic sensors to achieve the obstacle detection goal. But these systems are quite slow and not practically viable. The latest technology seems to use a combination of GPS, sensors and cameras. The cameras have algorithms which compare a captured image to its database. This has provided good results in an indoor environment [12].

The technology which seems to appeal to the visually impaired community is the RFID based systems. The RFID is used in almost all offices to It does not consume a lot of energy to set these RFID tags. This may seem advantageous; however, it is not possible to equip all areas with these tags. Another important advancement is the depth perception. With the help of these depth perception systems, potholes are identified. This is very helpful as uneven surfaces can be found, where even the sighted people trip when they do not pay attention.

The text to speech conversion has also been worked upon extensively. Text to speech (TTS) which is an integral part of this image detection and recognition has its working scenario as follows: the recordings are sliced and organized into acoustic database and the voice reference in database are segmented into syllable, morph and hemp etc. Some of the profitable text to speech converters include select and speak, neospeech, ivona, natural reader and Linguatec voice recorder studio15.The text to speech technology here is implemented with the aid of assorted components such as webcam, raspberry pi and a speaker. The webcam senses the object to be detected and characterized and provides the image as input to the raspberry pi. The image to be sensed is found by the raspberry pi using the python code which is running at the backend. The status of the on -going processes are updated in the LCD and the final output is delivered through speech.

The systems are all quite same when analyzed deeply. They sense the obstacle, give an audio output and a vibration. The use of computer vision is rare and even if it is used, it is used only in indoors. When we consider for outdoor navigation, the database to be considered becomes huge. When we head out, the possibility of the obstacles is quite random. To be able to detect and identify an obstacle seems impossible. In this in this aspect that the prospect becomes daunting. But the advanced in the field of artificial intelligence has reduced the complexity of this task. The system suggested in this module makes use of this artificial intelligence and can be used anywhere and doesn’t impose any prerequisites on the surrounding.

4. Proposed System Approach

It is necessary to discuss about the software and modules which will be used in this smart navigation system.

4.1 TensorFlow

The TensorFlow API is widely used in the field of object detection. In the era of facial recognition, it seems imperative to make use of the advanced technology to recognize objects as well. TensorFlow APIs can be used to detect with bounding boxes, objects in images and or videos using either some of the pre-trained models made available or through models which you can train on your own which the API also makes it easier.

TensorFlow, an open source machine library with a branch of machine learning called deep learning. It has led to proficient improvements in many zones mainly image classification and recognition. Deep learning has major advantage when working in the field of any texture of images. It is also done by classifiers. Classifiers are nothing but a set of codes or modules (Functions). The classifiers we use are preferably a high type of classifier namely neural network which could learn more complex functions. In order to make image classification and recognition at a comparatively easier pace, miscellaneous datasets have been created by the TensorFlow open source environment. Some illustrations include Coco, Kitti and Open image. Here comes in the coco dataset which has a collection of pre-trained detection models.

They are used for initializing models when training on novel datasets and out - of - box inference. COCO[9] (Common Objects in Context) has several features such as object segmentation, recognition in context, super pixel stuff segmentation, 1.5 million object instances, 330 K images, 80 object categories, 91 stuff categories, 5 captions per image, and 250,000 people with key points. COCO 2017 train/val browser has over 123,287 images and 886,284 instances.

Fig-1: COCO[9] Dataset

Deep learning is the basics of the object detection model used here. The concept of deep learning is a new area in machine learning. We are basically training the system to learn on its own. Usually there are three types of machine learning, supervised, semi-supervised and unsupervised learning. In this suggested system we will be using the pre-trained model which will be working on the 90 classes of COCO's dataset. Some of the models developed which uses COCO are:

Model name	Speed	COCO mAP	Outputs
ssd_mobilenet_v1_coco	fast	21	Boxes
ssd_inception_v2_coco	fast	24	Boxes
rfcn_resnet101_coco	medium	30	Boxes
faster_rcnn_resnet101_coco	medium	32	Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	slow	37	Boxes

Fig-2: Object Detection Models[11]

One of the major competencies in the field of computer vision is the object detection. The R-CNNSs could be classified into three advancements namely R-CNN, Fast R-CNN, and Faster R-CNN. The discrimination among these three could be done using parameters such as test time per image, speed up and map. The COCO dataset is used in Faster RCNN. The Regional Convolutional Neural Network plays a significant role in image segmentation and helps in the identification of the object.

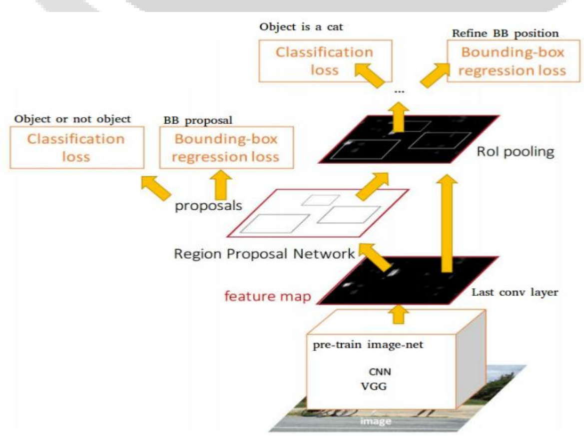


Fig-3: Faster RCNN model [13]

The above diagram explains the process of segmentation and classification of objects based on the faster RCNN model.

From the list given, we see that the faster_rcnn_inception_resnet_v2_atrous_coco has the best mAP, but it is quite slow. Thus, for testing purposes, we used the faster_rcnn_resnet101_coco which has medium speed, but higher accuracy of 32 mAP.

The Tensorflow Object Detection API has many models to offer, as mentioned above. We use the SSD(Single Shot Detection) Mobilenet model, which is a specifically designed light weight model for mobile applications. The Convolutional Neural Network has the two basic parameters: weight and bias. The regularizer class has a default weight of 0.00004. The activation function used by the model is Sigmoid function.

Sigmoid functions have the advantage of being in the range of 0 and 1. Thus they come extremely handy when the Output needs to be predicted as a probability.

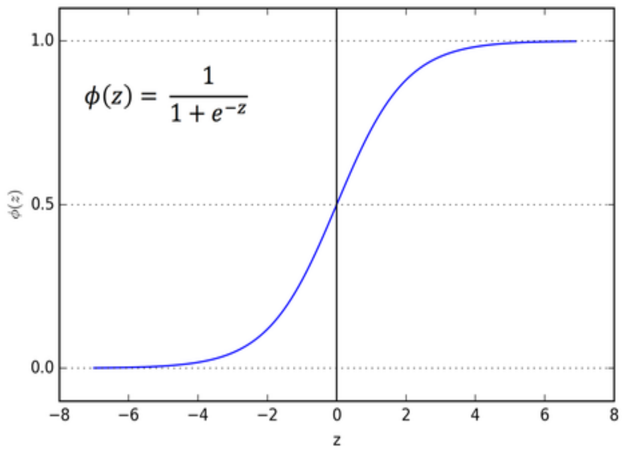


Fig-4: Sigmoid Function

In the convolutional layer, the neurons apply convolutional operation to input. In this layer, dot product between the image chunk and our filter(w) is done. This results in a single number output to which bias(b) is added. If out input is of size N*N, filter size is F, stride is S and input is added with a 0 pad of size P, then the output size will be:

(N-F+2P)/S+1

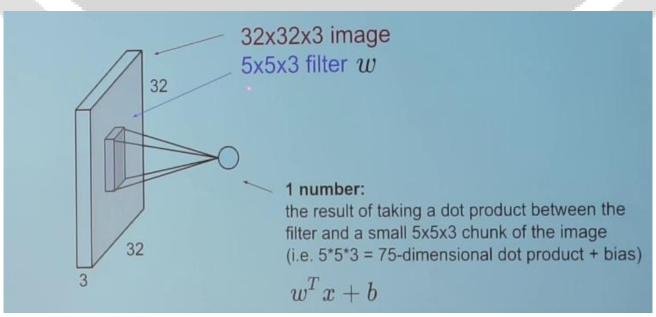


Fig-5: Convolutional Layer [13]

The next layer after convolution is pooling layer. It is used to reduce the spatial size. Max pooling is applied in this case. Filter size F*F is taken and maximum operation is applied over the F*F part of the image.

If we have width w, height h and depth d, then after pooling the image having w1*h1*d1 will be changed as:

$$w2=(w1-f)/S+1$$

$$h2=(h1-f)/S+1$$

$$d2=d1$$

where f*f is the filter size and S is the stride.

4.2 OpenCV

The OpenCV software is required for handling the operation of the camera used in the system. The function CaptureVideo() is important for beginning the use of the camera. The ffmpeg file is necessary for the separation of each frame. This is important as we need to process each frame separately. The other functions which come into use are:

- read()
- waitKey()
- imshow()

The functions are used to display a window upon which the detected objects boxed up with their names coming up on the top along with the percentage up to which the match is made.

4.3 Ultrasonic Sensor

The ultrasonic sensors are cost-effective, reliable and are used for object sensing and distance measurement. Ultrasonic sensors lay between proximity sensors and laser sensors in terms of measurement. The proximity sensors sense up to a low scale whereas the laser sensors have the capability of sensing up to 100 meters.

Here comes into limelight, the ultrasonic sensors which could detect up to three and a half meters and are unaffected by environmental factors such as fog, dust, mist and snow. Besides it is not even affected by color and texture of the objects. Some typical instances of advent in ultrasonic fields (ultrasonic sensors) could be witnessed in the measurement of diameter of rolled goods, monitor slack between machines, fluid and bulk material levels. The key benefits of ultrasonic are numerous. They are able to detect small objects over a large distance. It works independent of object surface and texture. The state of the obstacle is not a matter. It works on both solids and liquids. The color of the target does not affect the sensor. It can be used in any climatic and whether condition. The longevity and ease of maintenance. The ultrasonic sensors are classified into NPN and PNP types and used in object detection (discrete outputs) and distance measurement (analog outputs).

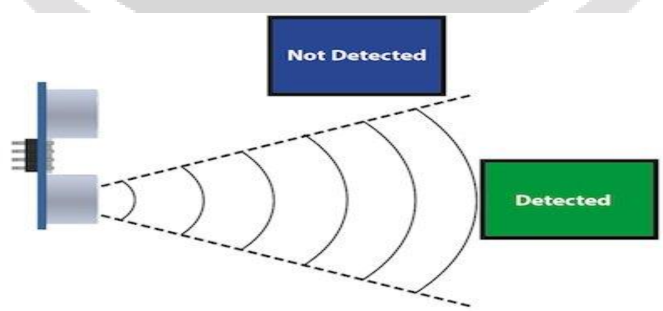


Fig-6: Ultrasonic Sensor[10]

The ultrasonic used here consists of two parts. One part will 'chirp' the ultrasonic pulse while the other will receive and process the echo. The distance is calculated based on the time required to get back the echo. It is a simple logic but quite effective. The ultrasonic sensor is not easily triggered, thus making it safe for the blind user.

4.4 Flow of Work

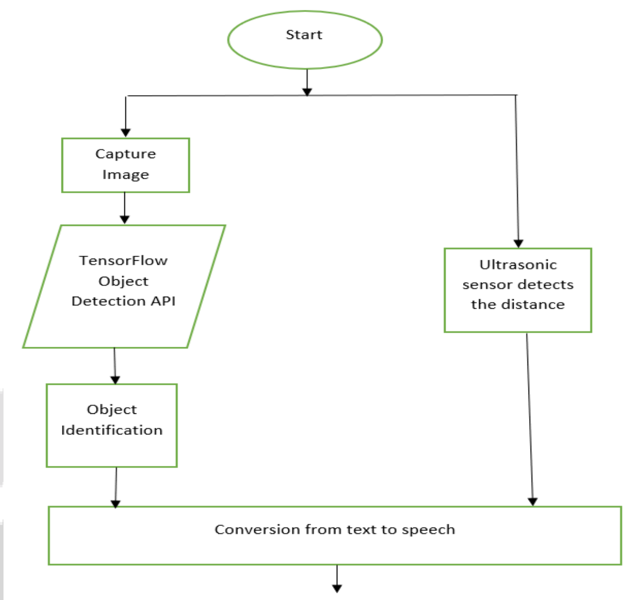


Fig-7: For single iteration

The flow chart shows how the proposed model works. The Raspberry Pi is connected with a camera. This camera is responsible for continuously capturing the images. This is a single iteration flow. After the image is captured we then proceed to use the TensorFlow algorithm for object detection and identification. Simultaneously, the ultrasonic sensor detects the distance of the obstacle. Both these outputs are converted to audio and given to the user via headphones.

Along with this we add the vibration feature. We understand that though the system boasts real time object detection, there is some delay in identifying the type of object. This short processing delay might prove very dangerous to the user. Hence, we add the vibration feature. As soon as the ultrasonic sensor detects the obstacle, a vibration is given to the blind user. Thus, they are immediately alerted and can take necessary precautions while the algorithm comes up with what sort of obstacle it is. The vibration can be given as a wrist band, which could be felt without the vibrator causing any hindrance to the user.

4.4.1 TensorFlow Object Detection API

For real time object detection, the Raspberry Pi needs to have the following:

- OpenCV 3.3.0
- TensorFlow Object Detection module
- Python 3.6
- Protobuf 3.4

The TensorFlow modules depends on the lxml and Protobuf libraries. These can be easily imported on a system. Once all the necessary compilation is done, we can upload the code onto a Raspberry Pi.

A brief glimpse on some of the code:

```
detection_graph = tf.Graph()
with detection_graph.as_default():
  od_graph_def = tf.GraphDef()
```

```
  with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
    serialized_graph = fid.read()
    od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name='')
```

In this part of the code[11], we are loading a frozen TensorFlow model. The word frozen is used because, there are many TensorFlow models and each model keeps training to get better. When a model gives accepted results, we immediately want to go for production. For this, the model and its weights need to be packaged and kept separately. This way, it can be easily updated and distributed. Here, we take the already pre-trained model which has been frozen. We use this frozen model for object detection and identification. The models are created based on Convolutional Neural Networks as we have seen above.

```
language = 'en-uk'
                with open("Ouput.txt", encoding="utf-8") as file:
                    file=file.read()
                speak = gTTS(text="I think its "+file, lang=language,
slow=False)
                file=("sold%s.mp3" %z)
                speak.save(file)
                print(z)
                z=z+1
```

The above code converts text to speech using Google Text To Speech services.

5.Experimental Output

The experiments were conducted using the light weight model ssd_mobilenet_coco_v1

CASE 1:

Fig-8: BOXING AND IDENTIFICATION OF OBJECTS

CASE 2:

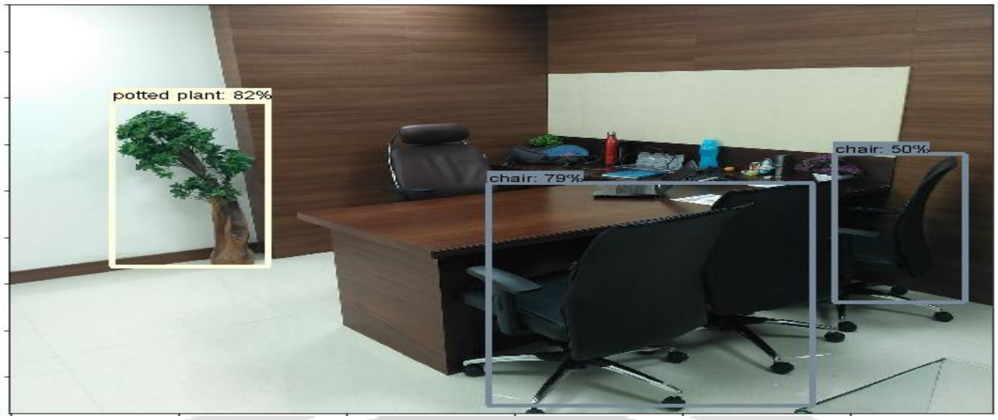


Fig-9: OBJECT DETECTION

CASE 3:

Fig-10: OBJECT(BAG) DETECTION

CASE 4:

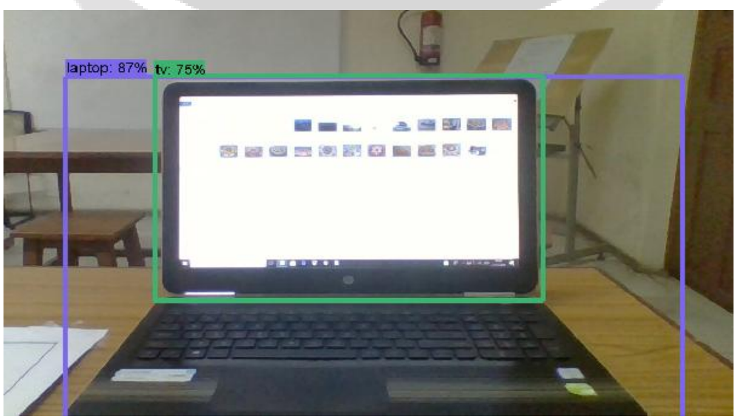


Fig-11: OBJECT DETECTION(LAPTOP)

CASE 5:

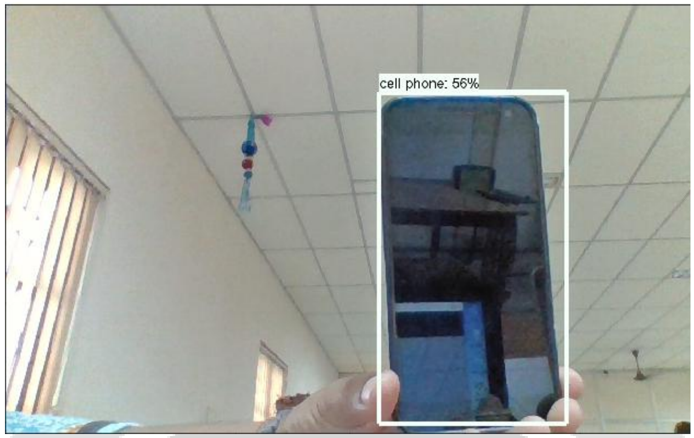


Fig-12: OBJECT DETECTION(PHONE)

CASE 6:

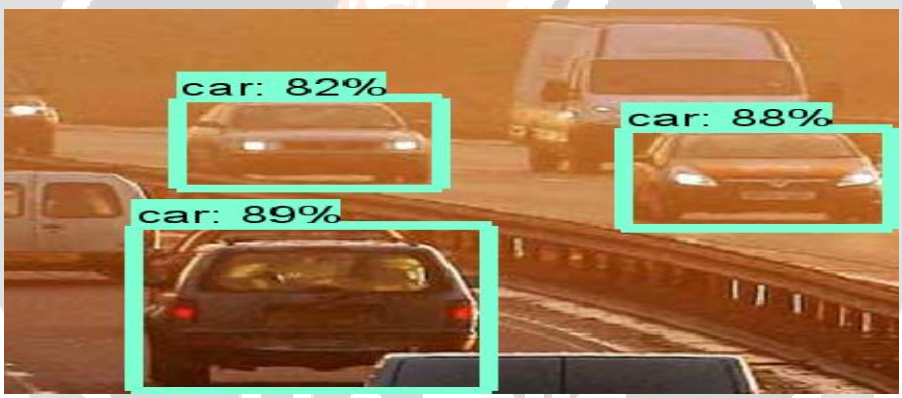


Fig-13: OBJECT DETECTION (MULTIPLE CARS)

CASE 7:

Fig-14: OBJECT DETECTION (MULTIPLE CHAIRS)

CASE 8:

Fig-15: OBJECT DETECTION (MULTIPLE ENTITIES)

CASE 9:

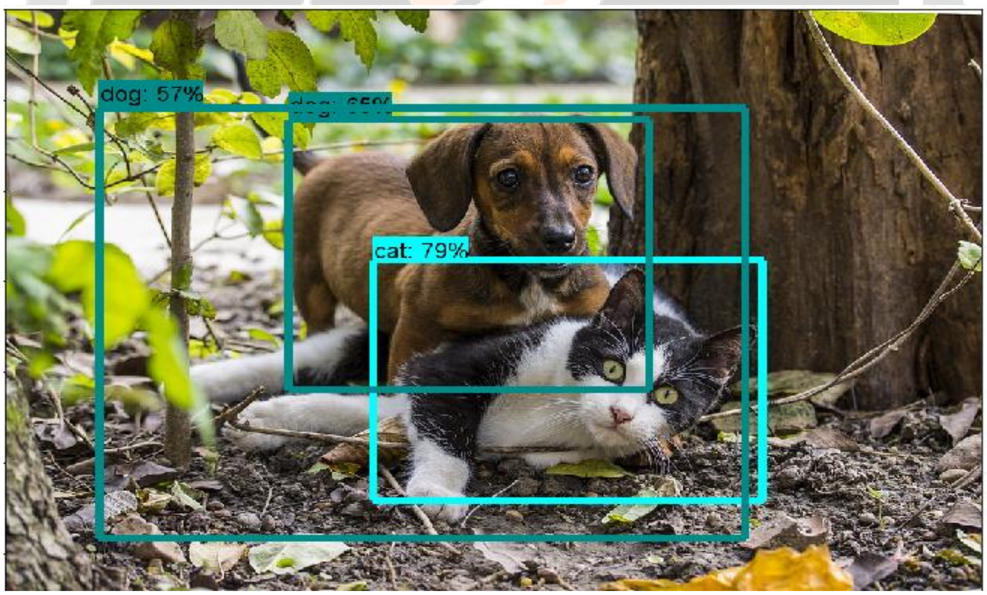


Fig-16: OBJECT DETECTION (MULTIPLE ENTITIES)

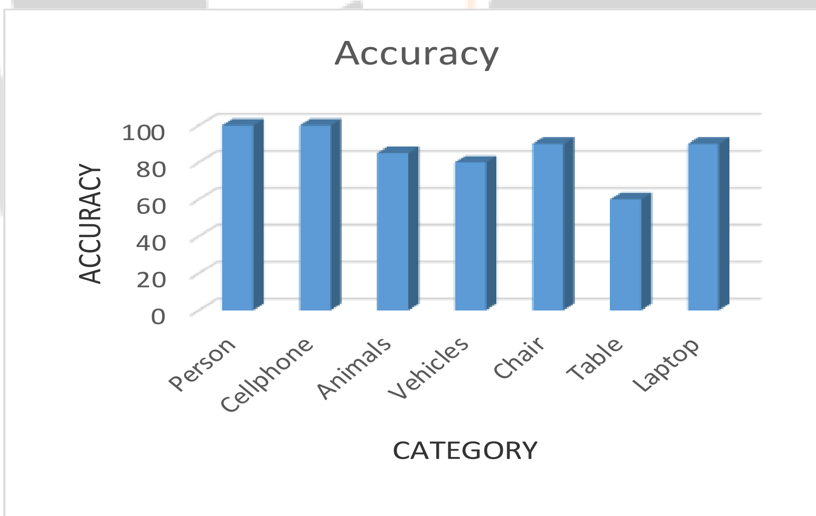
A sample of nine outputs are shown above. By analyzing the results, we can conclude that some categories like chair, cellphone, person are identified accurately in all cases. These category objects were placed in different positions and the model predicted the categories with varying probabilities. The Single Shot Detection MobileNet model used here predicts people with hundred percent accuracy. If the frame contains person/people, the model seems to focus on the person and does not detect other objects in the frame as seen in case 3. If two objects overlap each other, the category is identified correctly but the number of objects detected is not accurate as seen in case 9.

Table-1: Tabulation of Results

S.No	Captured Image	Actual Category	Classified Category	Evaluation	
				T/F	Accuracy %
1.		[‘person’, ‘bag’]	['person']	T	50
2.		[‘bus’]	['bus', 'person', 'person', 'person', 'person']	T	100
3.		[‘car’, ‘car’, ‘car’, ‘car’, ‘car’, ‘car’]	['car', 'car', 'car', 'car']	T	80
4.		[‘cellphone’]	[‘cellphone’]	T	100
5.		[‘chair’, ‘chair’, ‘chair’, ‘chair’]	[‘chair’, ‘chair’, ‘chair’, ‘chair’]	T	100
6.		[‘cow’, ‘calf’]	[‘cow’, ‘sheep’]	F	50
7.		[‘person’, ‘car’, ‘motorcycle’, ‘motorcycle’, ‘motorcycle’, ‘person’, ‘person’]	['car', 'motorcycle', 'person', 'person', 'motorcycle', 'motorcycle']	T	95

8.		[‘dog’, ‘cat’]	[‘dog’, ‘dog’, ‘cat’]	F	95
9.		[‘potted plant’, ‘chair’, ‘table’, ‘lamp’]	[‘potted plant’, ‘chair’]	T	50
10		[‘truck’]	[‘truck’, ‘bus’]	T	90

Chart 1- Graph Depicting Accuracy



6. Conclusion

The system proposed here is a novel method for obstacle detection and identification. It can be easily commercialized and be made to benefit the visually impaired community. Unlike other existing models, it does not require a large database because of the pre-trained Cognitive Neural Network model. The Single Shot Detection MobileNet model has been trained using the COCO[9] dataset which contains almost 300 thousand images. Hence, it can recognize any object without needing a database. From the experiments it was concluded that the system works extremely accurate in identifying people. It has 21 mAP (Mean Average Precision). Common place objects are also identified with satisfactory accuracy, provided they are sharply defined.

7.Future Scope

The future scope of the project determines to recognize any kind of object irrespective of its nature and scope. With the aid of more complex processors and improvement in profound technology, the extension of this module could identify any kind of entity with even more faster frame rate. The text to speech part could also be developed according to the futuristic pace. Instead of using the pre-trained models we can train the model by ourselves. The model can be trained to recognize objects which are frequently encountered by the user. Thus, it can be customized for the specific needs of the user and ensure safer navigation. The faster RCNN model which is already available, though very accurate is still slow. Improvements in the model will be a huge boost towards accurate and quick object detection.

8.References

[1]T. S. T. Ifukube and C. Peng. A blind mobility aid modeled after echolocation of bats. IEEE Transactions on Biomedical Engineering, 38(5):461–465, 1991.

[2]I. Ulrich and J. Borenstein. The guidecane - applying mobile robot technologies to assist the visually impaired. IEEE Transactions on Systems, Man, and Cybernetics, -Part A: Systems and Humans, 31:131– 136, 2001.

[3]D. Yuan and R. Manduchi. A tool for range sensing and environment discovery for the blind. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 3:39, 2004.

[4] K. Ito, M. Okamoto, J. Akita, T. Ono, I. Gyobu, T. Takagi, T. Hoshi, and Y. Mishima. Cyarm: an alternative aid device for blind persons. In Extended Abstracts Proceedings of the 2005 Conference on Human Factors in Computing Systems, CHI 2005, Portland, Oregon, USA, April 2-7, 2005, pages 1483–1488, 2005.

[5]G. Sainarayanan, R. Nagarajan, and S. Yaacob. Fuzzy image processing scheme for autonomous navigation of human blind. Appl. Soft Comput., 7(1):257–264, 2007.

[6]P. B. Meijer. An experimental system for auditory image representations. IEEE Trans Biomed Eng, 39(2):112–121, 1992 Feb.

[7] L. A. Johnson and C. M. Higgins. A navigation aid for the blind using tactile-visual sensory substitution. Conf Proc IEEE Eng Med Biol Soc, 1:6289–6292, 2006.

[8] D. Dakopoulos, S. K. Boddhu, and N. G. Bourbakis. A 2d vibration array as an assistive device for visually impaired. pages 930–937. IEEE Computer Society, 2007.

[9]information on coco from www.tensorflow.org

[10]arcbotics.com/products/sparki/parts/ultrasonic-range-finder

[11] TensorFlow Object Detection API tutorial
Tensorflow Object Detection API https://github.com/tensorflow/models/tree/master/research/object_detection

[12] Computer vision guidance system for indoor navigation of visually impaired people, 2016 IEEE 8th International Conference on Intelligent Systems.

[13] https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9