REAL TIME OBJECT DETECTION AND DISTANCE APPROXIMATION

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted By
ASHISH SINGH
2K19/CSE/04

Under the supervision of

Dr. ROHIT BENIWAL
(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2021
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

DECLARATION

I, Ashish Singh, Roll No. 2K19/CSE/04 student of M.Tech (Computer Science & Engineering), hereby declare that the Project Dissertation titled "**REAL TIME OBJECT DETECTION AND DISTANCE APPROXIMATION**" which is submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: DTU, Delhi

Ashish Singh
(2K19/CSE/04)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I, hereby certify that the Project Dissertation titled ***"REAL TIME OBJECT DETECTION AND DISTANCE APPROXIMATION"*** which is submitted by Ashish Singh, Roll No. 2K19/CSE/04, Department of computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of degree of Master of Technology (Computer Science and Engineering) is a record of a project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:

(Dr. Rohit Beniwal)

SUPERVISOR

Assistant Professor

Department of Computer Engineering Delhi Technological University

ABSTRACT

Ordinary diurnal tasks can be very strenuous for people with visual defects. Over the years many contemporary technologies have been developed to help visually impaired persons. However, these technologies are only able to narrate the contents on a mobile screen, and does not help in describing the real-world objects around a visually impaired person. The purpose of this research work is therefore, to provide a system that can detect an object and predict its distance and direction from an individual in real-time. The proposed system is a combination of an object detection model and a novel algorithm to approximate the distance and direction of objects called distance approximation algorithm. The detection and localization of objects is carried out by MobileNet and Single Shot Detector which are deep neural networks and are pretrained on the COCO dataset. The detection model highlights the identified objects by means of labelled bounding boxes. The coordinates of these bounding boxes are then used by the distance approximation algorithm to predict an object's distance and direction. The system is tested using different images and live video feed from a camera, however in order to determine the efficiency of the system images of a single object taken from various distances is used. Findings indicate that the system achieves an average accuracy of 96% in predicting the distance and thus, would be able to be effective in aiding visually impaired or blind persons.

ACKNOWLEDGEMENT

I am extremely grateful to **Dr. Rohit Beniwal** Assistant Professor, Department of Computer Science Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to his for the extensive support and encouragement he provided.

I also convey my heartfelt gratitude to all the research scholars of the web Research Group at Delhi Technological University, for their valuable suggestions and helpful discussions throughout the course of this research work.

Ashish Singh

Roll-No. 2K19/CSE/04

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Image processing refers to the process where some operations are performed on an image, which makes it fit for extracting useful information. This useful information is mainly features or characteristics of the image provided as an input to the image processor. The output of the image processor is further used to detect various objects in the image using a computer vision technique known as object detection. Object detection algorithms rely on machine learning to identify various patterns from feature extraction of an image and based on them, they distinguish between different objects. Therefore, we also use the application of image processing and machine learning to provide a model for detecting an object in the blind person's surrounding and predicting how far it is from the concerned person and what is its related direction. This concept could help people who are suffering from blindness, Glaucoma, Diabetic retinopathy, etc. By sensing the surrounding environment and assisting them to know the things around them with an approximate distance could help them in finding a path to walk through. Although there are some apps which help blinds or visually impaired people in navigating through the public places by using some other personal assistance like the app Be My Eyes, Be My Eyes is all about contributing and benefiting from small acts of kindness. Blind or visually impaired users can request help from a sighted volunteer, who will be notified on their phone. As soon as the first sighted user accepts the request for help, a live audio-video connection will be set up between the two parts. The sighted helper can now assist the blind or visually impaired, through the video connection from the blind or visually impaired user's rear-facing camera. But in this project, we have tried to build a system which will work independently, without any other person's involvement.

1.1 MOTIVATION

As per the work of Bourne et al. in Lancet Global Health [1], an estimated 217 million people suffer from moderate to severe visual impairment and 36 million are blind. Functional presbyopia affects an estimate of 1094.7 million people, out of which 666.7 million people are aged 50 years and above. The rise in the number of elderlies will increase the percentage of the population who are at risk of visual impairment [1]. Further, these visual defects are not limited to just the elderly, children aged below 15 who are in the prime of their lives are also suffering from these defects. Moreover, according to the World Health Organization's (WHO) data, globally around 2.2 billion people suffer from visual

impairment. Therefore, these people are in dire need of aid to enhance their vision and impede the progression of their disability to sense the world better [2].

Amidst this, there are 15 million people, who are suffering from blindness, thus making India, home to the biggest blind population on the globe [3]. Globally, it is estimated that overall, 40 to 45 million people are blind and cannot walk without any assistance [4]. There are few apps, which help blind or visually impaired people in navigating through public places using assistance from volunteers of these apps. However, these apps have a limitation that they are always dependent on the assistance of volunteers. However, in this research work, we provide a system, which will work independently of any volunteer i.e., without any other person's assistance.

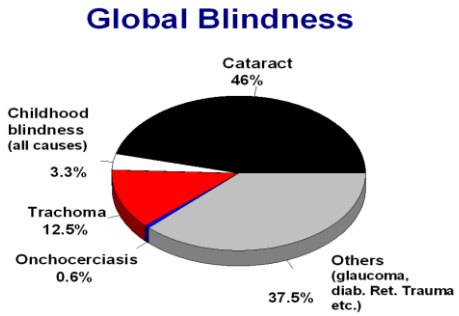

Fig. 1. Global Data of eye-related diseases

1.2 CONTRIBUTIONS

For the purpose of object detection, a pre-trained MobileNet SSD model is used which is trained over the COCO dataset which includes 90 different classes. The model successfully detects the multiple objects from the image. For implementing object detection from live streaming OpenCV module has been used which feeds the frames from the live video stream. Objects in the image captured through webcam are detected by the MobileNet model and a novel approach has been used to estimate the distance of the object from the webcam. A simple observation is implemented to detect the object from the webcam that is the size of the object is inversely proportional to the distance, if the distance increases the size of the object decreases. We believe that this study will inspire more people towards the welfare of the blind people and this study will work as the starting steps to many more future studies in this field with the development of existing technology. With this study, we hope that we could provide enough description of the main features that need to be included in any system that serves this group of people and make their life a bit easier.

CHAPTER-2

LITERATURE REVIEW

2.1 IMAGE CLASSIFICATION

Image Classification is wherever a computer will analyze a picture and determine the 'category' the image falls below. (Or the chance for the image to be a part of a 'category'.) A class is primarily a label, as an example, 'car', 'animal', 'structure' and then on. For example, you embody an image of a cat. Image segregation may be a program that analyzes a picture and tells you that it's a cat. (Or it can be a cat.)

For us, separating pictures isn't an enormous deal. However, it's an ideal example of Moravec's contradiction once it involves technology. (That is, the items we discover straightforward square measure troublesome in AI.)

Original image classification relies on raw pixel information. This suggests that computers can break down pictures into individual pixels. The matter is that 2 pictures of identical object might seem to be terribly completely different. They will have completely different backgrounds, angles, shapes, etc.

This has created it terribly troublesome for computers to 'see' properly and separate pictures. To beat these difficulties deep learning is employed.

Deep learning may be a style of machine learning; a set of computing (AI) that enables machines to be told from information. Deep learning involves the utilization of pc programs called neural networks.

In neural networks, input filters through hidden layers of areas. These nodes perform every input method and transmit their results to consequent layer of nodes. This is often perennial till it reaches the output layer, so the machine offers its response.

There square measure differing kinds of neural networks supported however the hidden layers work. Image classification by typical reading typically involves convolutional neural networks, or CNN. At CNN, nodes within these hidden layers don't perpetually share their output with all nodes in the next layer (known as convolutional layers).

Deep learning permits machines to spot and extract options from pictures. This suggests that they will learn the weather to appear at in photos by analyzing multiple pictures. Therefore, program planners ought not to install these filters manually.

2.2 MOBILENET

MobileNet could be a CNN design model for Image Classification and MobiIe Vision. There are different models moreover however what makes MobiIeNet special that it terribly less computation power to run or apply transfer learning to. This makes it an ideal appropriate Mobile devices, embedded systems and computers while not GPU or low procedure potency with compromising considerably with the accuracy of the results. It's additionally best suited to net browsers as browsers have limitation over computation, graphic process and storage

2.2.1 MobileNet Architecture

- MobileNets for mobiIe and embedded vision applications is planned, that are supported by efficient that uses depthwise separable convolutions to build light weight deep neural networks.
- Two easy world hyper-parameters that efficiently exchange between latency and accuracy are introduced.

The core layer of MobiIeNet is depthwise separable filters, named as Depthwlse SeparabIe Convolution. The network structure is another issue to spice up the performance. Finally, the dimension and resoIution can be tuned to exchange between latency and accuracy.

2.2.2 Depthwise Separable Convolution

Depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×11×1 convolution called a pointwise convolution. In MobileNet, the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×11×1 convolution to combine the outputs the depthwise convolution. The following figure illustrates the difference between standard convolution and depthwise separable convolution.

CHAPTER-3

RELATED WORK

Although many researchers have worked on object detection and image processing, however, to the best of our knowledge, no work has been found so far to detect an object in real-time and predict its distance using neural networks and image processing techniques. Although there are various tools, from a simple cane to the complex system using software and hardware, to assist a blind person. However, their feasibility varies from indoor to outdoor along with dynamic surroundings. In the 1990s, Golledge et al. were the first ones to propose a conceptual model that intended the use of Geographic Information System (GIS), sonic sensor components, Global positioning system (GPS), and speech for helping the visually impaired [5]. An example of an implemented model is a system called MOBIC, which is based on GPS for the aid of the less visually privileged. It also uses a voice command to dictate the path and direction to the blind user [6]. Similarly, Drishti, which is a wireless pedestrian navigation system, devises a path for the user that incorporates various technologies such as mobile computers, wireless networks, voice recognition, GIS, and GPS. Drishti System imbibes all the surrounding information and then evaluates an optimized path for the destination of the concerned blind user [7]. In addition, technologies such as RFID chips use a lot of hardware infrastructure, being highly static in nature, require advanced positioning of chips, and are only limited to indoor systems [8]. Another example of a device that can sense its environment and can walk avoiding obstacles is a robot known as Lola, which is a human-sized biped robot that uses onboard sensing and 3D point cloud processing techniques [9]. With the outbreak of coronavirus disease, several researchers have worked on measuring the social distance between the two objects [10]. However, their work is still lagging in providing any assistance to a visually impaired person. Therefore, in this research work, we provide a system that can fulfill the gaps of the earlier works where a system has been designed and implemented, which works independently without requiring any person's assistance. Moreover, the system detects the objects in real-time and predicts their approximate distance from the blind person.

CHAPTER-4

RESEARCH APPROACH

The research approach for real-time object detection and distance approximation is divided into two phases namely object detection and distance approximation, which are as follows:

4.1 OBJECT DETECTION

We use a combination of MobileNet and Single Shot Detector (SSD) which is trained on the COCO dataset. The conjunction of both allows us to detect, recognize and localize multiple objects in an image. This combination is much more computationally economical and doesn't sacrifice much accuracy. For the classification of objects, we have used MobileNet Architecture. MobileNet is lightweight neural networks which use depth wise separable convolutions. The MobileNet uses the standard convolution in which the values of all the input channels is combined by the convolution operation. The standard convolution is only used as the first layer where an output image with only 1 channel per pixel is obtained by running a single convolution kernel across an image that has 3 input channels. The rest of the layers do "depth wise separable" convolution. Two different convolution operations are combined here which are a point wise convolution a depth wise convolution. Convolution is performed on each channel separately in a depth wise convolution where as in a regular convolution, the input channels are combined. A depth wise convolution creates an output image with 3 channels for an image with 3 channels. A unique set of weights is assigned to each channel. Filtering the input channels, edge detection, color filtering, etc. are performed by the depth wise convolution. The depth wise convolution is followed by a point wise convolution [11-12]. It is almost similar to a regular convolution instead there is a 1×1 kernel as shown in Fig 2 [11].

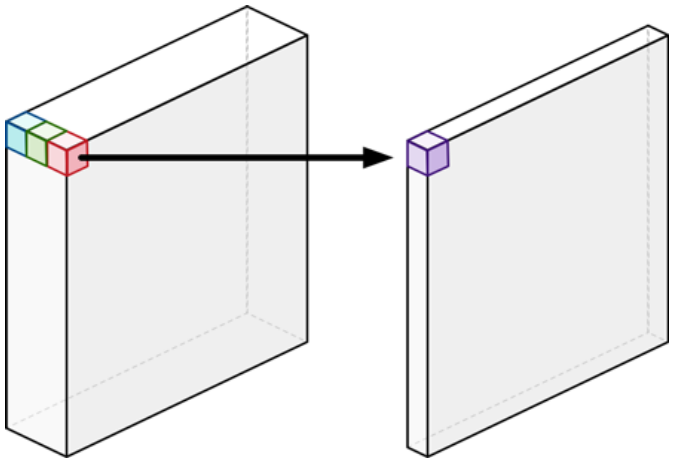

Fig. 2. A pointwise convolution

All the channels are simply added up (as a weighted sum) to be precise. In a normal convolution, many of these point wise kernels are usually stacked up together with many channels to create an output image. The purpose of this point wise convolution to create new features by combining the output channels of the depth wise convolution. The resultant convolution is called a depthwise separable convolution made by putting together these two convolutions— a depthwise convolution followed by a pointwise convolution The task of filtering and combining is done in a single step in a regular convolution, but these are performed at different steps with a depthwise separable convolution [11-12].

Through this process we bring down the computation cost of the process from "D1 · D1 · X · Y · D2 · D2" to "D1 · D1 · X · D2 · D2" where number of input channels is X, the number of output channels is Y, the kernel size is D1 × D1 and the feature map size is D2 × D2 [11].

There are 30 layers in a full MobileNet [12]. The network design is as follows:

i. convolutional layer with stride 2
ii. depthwise layer
iii. pointwise layer that doubles the number of channels
iv. depthwise layer with stride 2
v. pointwise layer that doubles the number of channels
vi. depthwise layer
vii. pointwise layer
viii. depthwise layer with stride 2
ix. pointwise layer that doubles the number of channels
x. and so, on up to total 30 layers

SSD is used to localize the objects in an image. SSD is intended to be free of the bottom network, it can run above just about everything, together with MobileNet [12] shown in Fig 3[11]. MobileNet + SSD shown in Fig 4[13] shows that instead of regular convolutions depthwise separable layers are used for the network's object detection part. With SSD sandwiched above MobileNet, we get results that are real-time. The conversion of pixels from the input image into features is carried out by the MobileNet layers which describes the objects within the image and the next layers receives them as a forward. MobileNet is employed here as a feature extractor for the SSD [13].

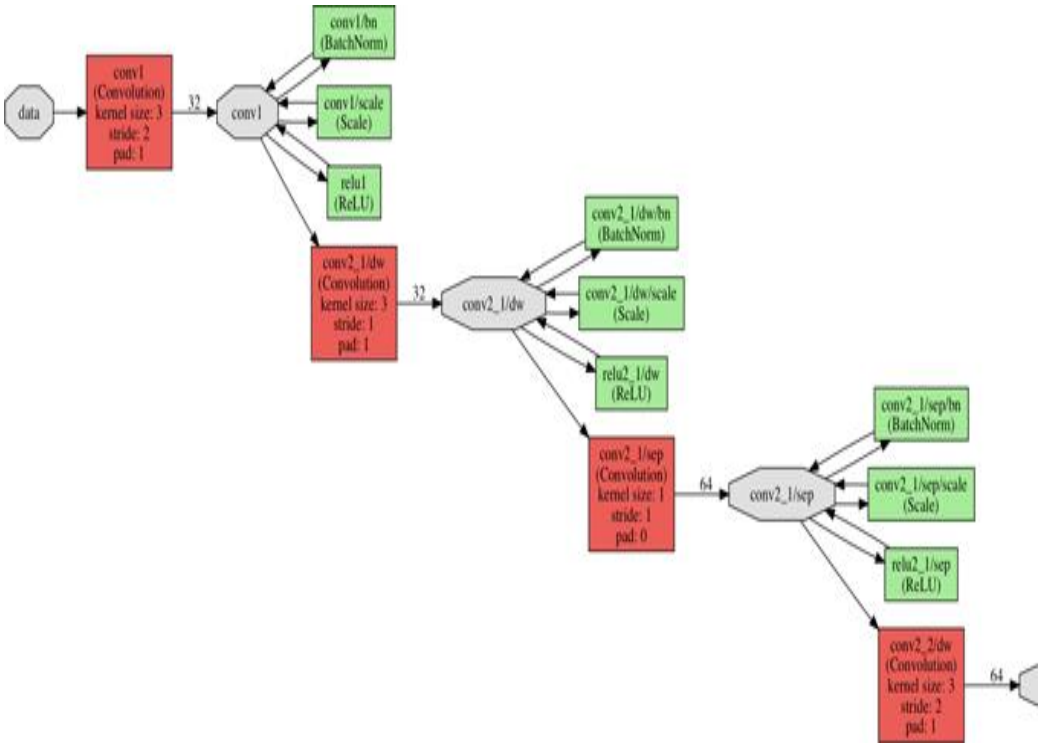

Fig. 3. MobileNets architecture [11]

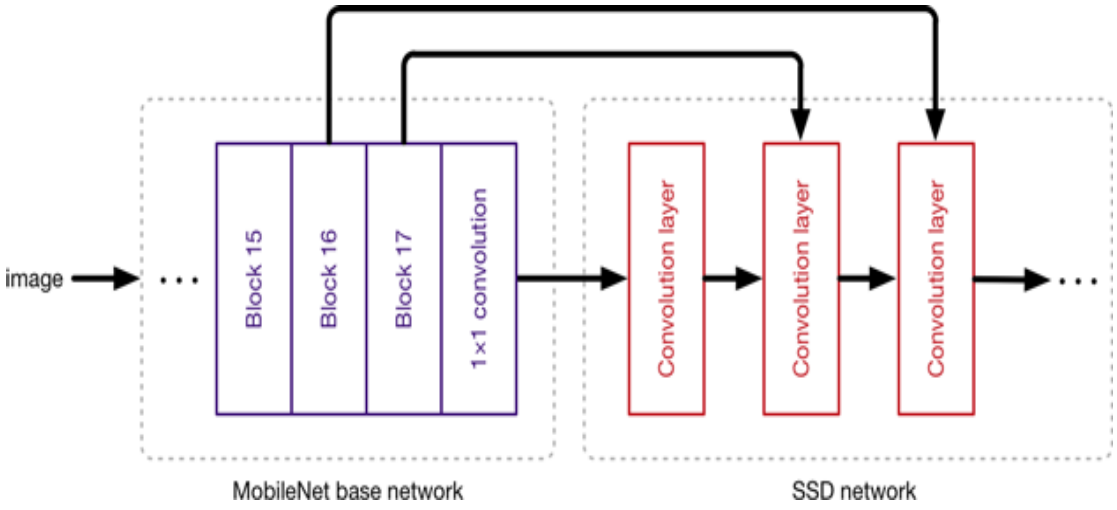

Fig. 4. MobileNet SSD Architecture [13]

We feed forward the low-level features of MobileNet to SSD convolution layers because we do not want only the classification of objects but also the location of the object in each image. So, for this, we not only connect the high-level features of the MobileNet Network but also the feed forwards the low-level features to localize the object. Fig 5 depicts the control and data flow of the object detection process.

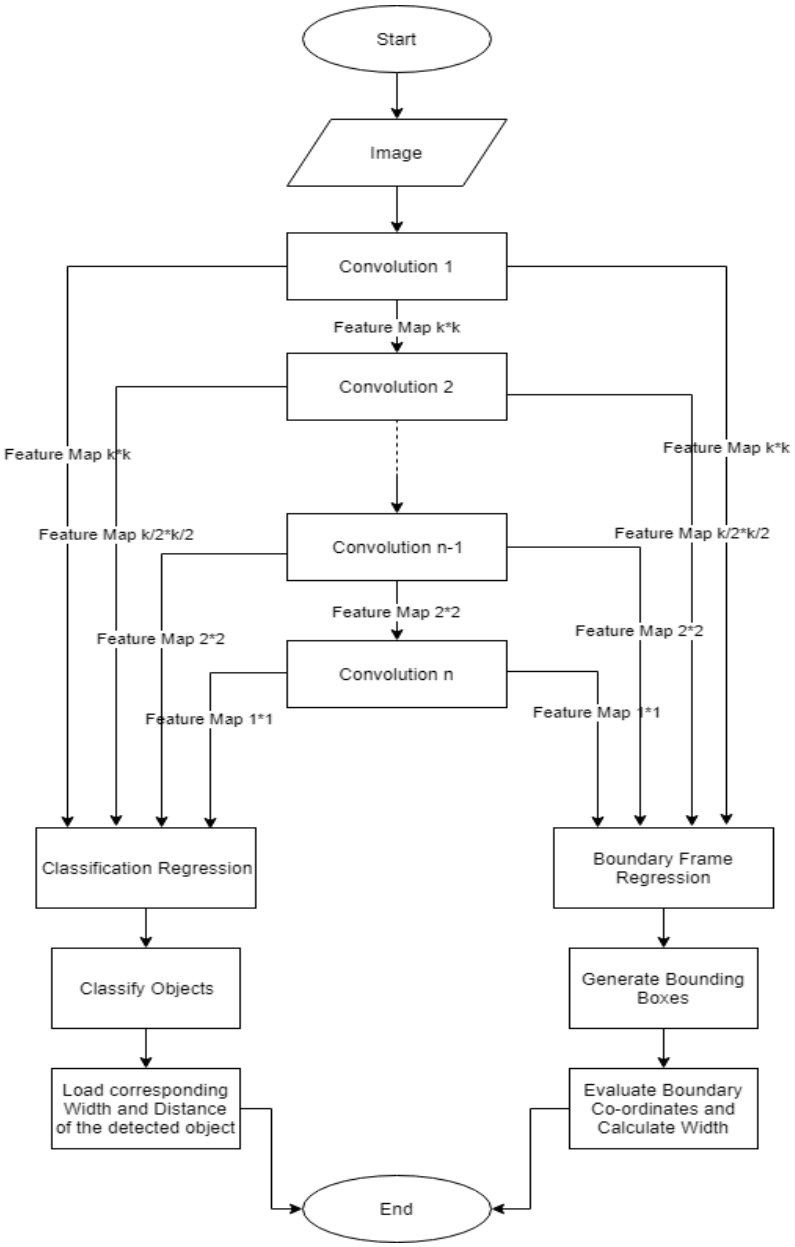

Fig. 5. Object Detection Process

4.2 Distance Approximation

First, we need to detect the objects from the image or videos. For object detection, TensorFlow Object Detection API has been used, which will be able to recognize different objects from the images. It can detect 79 different class of objects that are: “person, bicycle, car, motorcycle, airplane, bus, train, truck, boat, traffic light, fire hydrant, stop sign, parking meter, bench, bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, backpack, umbrella, handbag, tie, suitcase, Frisbee, skis, snowboard, sports ball, kite,

baseball bat, baseball glove, skateboard, surfboard, tennis racket, bottle, wine glass, cup, fork, knife, spoon, bowl, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake, chair, couch, potted plant, bed, dining table, toilet, TV, laptop, mouse, remote, keyboard, cell phone, microwave, oven." For videos, OpenCV has been used to feed the video frames to the object detection model. After detecting the object and classifying its class the coordinates of the bounding box is evaluated which is further used for the estimation of object's distance from the camera. For the purpose of distance prediction, we need to pre-store one instance of the object of every class from a distance and the original size of the object must be known. That is just like the training of the neural network models we need to specify or let's say train the system using instance and predicting the distance of object for other. By using this instance, a formula is defined and that will be used to predict the distance for other instances. Now, for the direction of object coordinates of the bounding box has been used. By using the coordinates, quadrant can be defined in which the center of the object lies.

This method of distance estimation will be implemented on images as well as on videos. For distance estimation, we considered that the distance and size of the object are inversely proportional to each other. Assume that we place the object at a distance 'd' inches from the camera and the size of the object in the image is 'S' units. Now, we store these values once and for all, when this object is detected in some other image, the relation would be $S * d = s * D$, where 's' is the new size of the image and 'D' is the new distance. Hence, the formulae for the distance will be $D = \frac{(S*d)}{s}$. The algorithm for distance approximation is as follows:

1. Detect an object and its class using object detection API (TensorFlow).
2. Find out the coordinates for the bounding box which are the coordinates of the principal diagonal of the rectangular box (x_1, y_1) and (x_2, y_2).
3. Now, get the values of 'S' and 'd', previously stored for the object, which are training values.
4. Calculate new size of object by the Equation1: $s = x_2 - x_1$.
5. Distance $D = \frac{S*d}{s}$
6. To find out direction of object
   - Calculate the center of the object that is $\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}$.
   - Now, by examining where this point lies in the coordinate system of the camera screen or the window that is used, we can tell its direction.

Methodology for Distance Approximation: Fig 6 presents the methodology used for the distance estimation process.

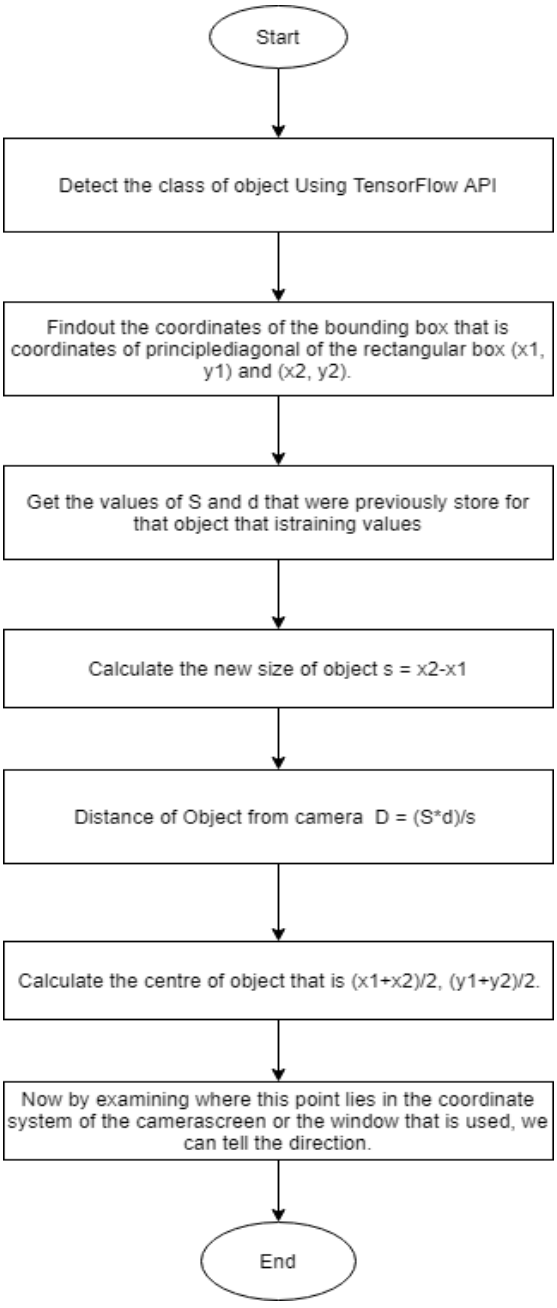

Fig. 6. Methodology for Distance Approximation

CHAPTER-5

IMPLEMENTATION, RESULTS, AND ANALYSIS

To implement the research approach, there are two alternatives available. One is to build and train the model from scratch for object detection and then implement the distance approximation algorithm on top of the new model. The other alternative is transfer learning, where an already trained, model is pulled to achieve the task. However, we chose the second alternative since it is a more efficient method as compared to the first one.

To proceed with the second alternative, we used the TensorFlow object detection API to pull "ssd_mobilenet_v1_coco_2017_11_17," which is an already trained model on the COCO dataset. Now, with the model in hand, we used OpenCV to capture live frames via camera and then fed them to the model for classification. After classification is done on a frame, bounding boxes are drawn around every recognized object and the coordinates of those objects are then fed to the distance approximation algorithm, which ultimately gives us the distance and the direction of the object.

Through this module, we were able to identify and localize the prominent objects in the image. Along with that, we were also able to extract the coordinates of the bounding box, which with respect to the camera or user helps in finding the relative direction and distance of the classified object to aid the visually impaired.

5.1 OBJECT DETECTION

The results of the implementation of the object detection approach are represented by the following figures. Fig 7 shows the image of two street dogs and Fig 8 presents the output of the model that has successfully detected the two dogs with 93% confidence and with their coordinates in Fig 9.

Fig. 7. Image of 2 street dogs

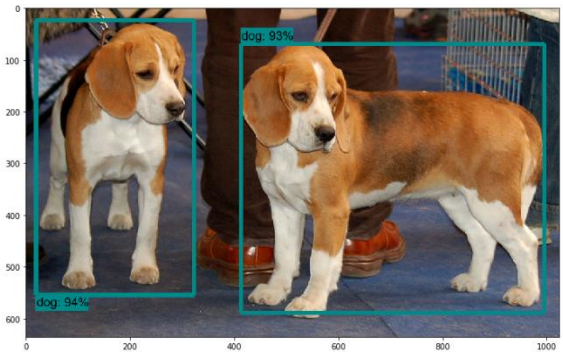

Fig. 8. Two street dogs predicted from image

```
dog
dog
24.85745394229889 19.676193237304688 554.6577959060669 323.35205078125
69.65154719352722 412.503662109375 588.0749065876007 996.4010009765625
```

Fig. 9. Predicted Dogs Coordinates in image

Fig 10 is the image of the laptop which was taken using a camera and Fig 11 is the result generated for Fig 10 which shows that laptop has been correctly detected in the image with 99% confidence and coordinates of the laptop are generated in Fig 12.

Fig. 10. Laptop on table

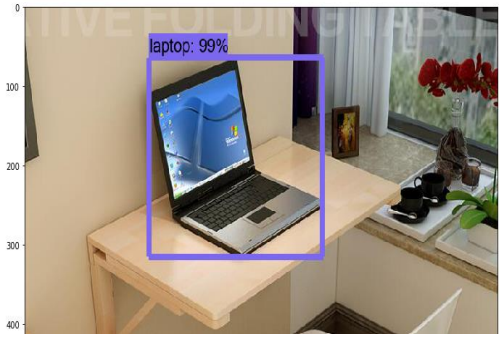

Fig. 11. Laptop detected from the image

```
laptop
64.592145383358 196.0929036140442 314.406156539917 472.2016006708145
```

Fig. 12. Coordinates of predicted laptop

After analyzing the above results with a live webcam, it can be said that they are satisfactory and this module can recognize various objects in an image, which can assist a visually impaired person. Since this module is computationally economical and light, it can provide real-time assistance to the user by using a live feed camera and voice commands.

5.2 DISTANCE APPROXIMATION

We implement and test this approach by using images of a bicycle taken from different distances. The results in this module are obtained using the formulae as defined in section 3.2. According to the formulae, first, we pre-stored one instance of the bicycle image as shown in Fig. 13(a). For this image we already knew: Distance d = 60 inches and Size of the image from 60 inches, S = 3300

Accordingly, these pre-stored are provided as a base for all other images. Distance prediction is then carried out based on these two parameters *'S'* and *'d'*. The first case that is considered is for an image where the bicycle distance from the camera is 92 inches as shown in Fig. 13(b).

(a)

(b)

Fig. 13. (a)Bicycle at a distance of 60 inches, (b)Bicycle at distance 92 inches.

The distance of the bicycle from the camera will be, $D = \frac{(S*d)}{s}$

Where, '*D'* is the new Distance of the bicycle from the camera, *'S'* is the size of the bicycle from a predefined distance, *'d'* is the distance of object from a predefined distance and *'s'* is the new size of the object

'S' and *'d'* are pre-stored whereas *'s'* is calculated using the object coordinates as obtained in section 4.1. The formulae for the calculation of *'s'* is as discussed in Equation1 of Section 3.2. Here s = 3129.74 – 913.75 = 2215.99.

Consequently, D = (3300*60)/2215.99 = 89.35 inches.

Similarly, distance approximation is carried out for various bicycle images taken from different distances and a table is compiled for their analysis. Table 1 shows a comparison of predicted and original distances of objects in inches along with their percentage error.

Table 1 Comparison of predicted and original distances of the object.

Original Distance (inches)	Predicted Distance (inches)	Percentage Error
52	54	3.8462
68	65	4.412
76	73	3.947
84	79	5.952
92	89	3.261
100	97	3
108	101	6.481
82	79	3.659
125	120	4
Average Percentage Error		4

After analyzing the above table, it is found that the average error margin is 4% and thus resulting in accuracy of 96% in predicting the distances as compared to the original ones. Therefore, it can be said that this model can help a blind person to sense and be aware of his surroundings.

5.3 REAL-TIME OBJECT DETECTION USING WEBCAM

Result as depicted by Fig. 14 shows the real-time feasibility and proficiency of the proposed system in detecting multiple objects of the same as well as of different types. This system has also been tested using a webcam and is perfectly detecting as well as predicting the distance of objects in live streaming. Also, the direction of the objects has been correctly determined by the system as depicted by Fig. 15.

Fig. 14. Object detection using Webcam.

```
Location of  cell phone is  Right
FOUND..... cell phone at
Distance of  cell phone is  179.01865230499862
Location of  cell phone is  Right
50
FOUND..... person at
Distance of  person is  26.463277887138606
Location of  person is  Right
51
FOUND..... cell phone at
Distance of  cell phone is  29.42487469607771
Location of  cell phone is  Right
```

Fig. 15. Direction of Objects.

CHAPTER-6

CONCLUSION

The purpose of doing this research work was to develop modules that can be integrated together to construct a more sophisticated system of object detection and distance approximation because such systems can be used to solve many real-world problems. Therefore, in this research work, we proposed a system for detecting the objects as well as finding out their approximate distances. To the best of our knowledge this is the first attempt that combines detection and distance approximation of objects. MobileNet and SSD are used to detect and localize objects from input sources. The localized objects have bounding boxes drawn around them by the detection model. The coordinated of these bounding boxes are then used by distance approximation algorithm to estimate the distance and direction of objects. We then tested this system on some real-life objects as obtained from images and live streaming videos. As a result, the system is able to detect, recognize, and find the localized position of the objects in a given image or in the live streaming videos. Furthermore, the system is capable of approximating an object's distance with an average accuracy of 96%. Also, the system is able to find the object's relative direction from the camera. This unique integration of the techniques as used in the proposed system will assist people with visual impairment in their day-to-day life activities. Moreover, as part of future work, the system can be implemented using voice prompts, which will assist disabled persons.

REFERENCES

[1] Bourne R, Flaxman S, Braithwaite T et al. (2017) Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. The Lancet Global Health 5:e888-e897. doi: 10.1016/s2214-109x(17)30293-0.

[2] World Health Organization (WHO). https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

[3] India has the largest blind population (Kounteya Sinha, Oct 11, 2007). https://timesofindia.indiatimes.com/india/India-has-largest-blind-population/articleshow/2447603.cms

[4] World Health Organization (WHO). https://www.who.int/blindness/Vision2020%20-report.pdf.

[5] GOLLEDGE R, LOOMIS J, KLATZKY R et al. (1991) Designing a personal guidance system to aid navigation without sight: progress on the GIS component. International journal of geographical information systems 5:373-395. doi: 10.1080/02693799108927864

[6] Petrie H, Johnson V, Strothotte T et al. (1996) MOBIC: Designing a Travel Aid for Blind and Elderly People. Journal of Navigation 49:45-52. doi: 10.1017/s0373463300013084

[7] L. Ran, S. Helal and S. Moore, "Drishti: an integrated indoor/outdoor blind navigation system and service," Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the, 2004, pp. 23-30, doi: 10.1109/PERCOM.2004.1276842.

[8] S. Chumkamon, P. Tuvaphanthaphiphat and P. Keeratiwintakorn, "A Blind Navigation System Using RFID for Indoor Environments," 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008, pp. 765-768, doi: 10.1109/ECTICON.2008.4600543.

[9] D. Wahrmann, A. Hildebrandt, R. Wittmann, F. Sygulla, D. Rixen and T. Buschmann, "Fast object approximation for real-time 3D obstacle avoidance with biped robots," 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2016, pp. 38-45, doi: 10.1109/AIM.2016.7576740.

[10] Imran Ahmed, Misbah Ahmad, Joel J.P.C. Rodrigues, Gwanggil Jeon, Sadia Din, A deep learning-based social distance monitoring framework for COVID-19, Sustainable Cities and Society, Volume 65, 2021, 102571, ISSN 2210-6707, https://doi.org/10.1016/j.scs.2020.102571.

[11 Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*

[12] Google's MobileNets on the iPhone (14 JUNE 2017). http://machinethink.net/blog/googles-mobile-net-architecture-on-iphone/

[13] MobileNet version 2 (22 APRIL 2018). http://machinethink.net/blog/mobilenet-v2/