Dynamic Object Recognition and Command Detection for Assisting Blind People Using Image Processing Techniques

By

Labeba Tahsin
19241031
Nadia Ahlam Nuba
15301031
Mashiat ferdous Sami
15301094
Sabrina Afrin
15301098

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
December 2019

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Nadia Ahlam Nuba 15301031	**Mashiat Ferdous Sami** 15301094
Sabrina Afrin 15301098	**Labeba Tahsin** 19241031

Approval

The thesis titled “Dynamic Object Detection and Command Detection for Assisting Blind People Using Image Processing Techniques” submitted by

1. Nadia Ahlam Nuba (15301031)
2. Mashiat Ferdous Sami (15301094)
3. Sabrina Afrin (15301098)
4. Labeba Tahsin (19241031)

Of Fall, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 26th December 2019.

Examining Committee:

Supervisor:
(Member)

_____

Dr. Md. Ashraful Alam
Assistant Professor,
Department of Computer Science and Engineering.
BRAC University.

Thesis Coordinator:
(Member)

_____

Dr. Md. Golam Rabiul Alam
Associate Professor,
Department of Computer Science and Engineering.
BRAC University.

Departmental Head:
(Chair)

_____

Mahbubul Alam Majumdar, PhD
Professor and Chairperson,
Department of Computer Science and Engineering.
BRAC University.

Abstract

Human senses are the only medium to communicate with environment. Our brain makes a combination of different senses and convert it into a meaningful mixture. We, humans have five different senses as in two eyes to see, one nose to smell, two ears to hear, one tongue to taste and a skin to feel touch. All of these senses are connected deeply with each other. One sense always has the backup of another one. There are thousands of people in the whole world who are blind. They face many kinds of hindrances in their everyday life. They suffer badly in travelling here and there like shopping, work, educational institutions and so on due to their problem in sight. Hence, many papers have been published on making blind or visually impaired people's navigation easier. Different papers proposed different methods and different ways to navigate their way. Our paper aims to represent a proposed system that helps a visually impaired, which detects various objects and finds right path for them to reach to their destination. Our projected model is mainly based on a well-known object detect and identifier algorithm that needs to be trained by a dataset. Step by step, simulations using dynamic object detection algorithm results in a better accuracy to detect those objects in our proposed model. Hereby, our paper presents the idea of an object detection system based on object extractions; networks for segmentations match with the recognized dataset and locate those objects from the live video-image extractions, which gives an audio output.

Keywords: visually impaired; blind people; objects detection; navigation system; wearable device; path detection.

Dedication

The price of success is hard work, dedication to the job at the hand and the determination that whether we win or lose, we have applied the best of ourselves to the task at hand.

- Vince Lombardi

We dedicate this thesis to Allah for His Fruitful help to finish this thesis, as well as to our family, friends and to our supervisors, classmates and lastly to BRAC University for their help and endless sacrifices.

We wish to express our sincere appreciation and thanks to all that directly and indirectly helped us throughout the whole thesis process.

Acknowledgement

To start, all praise to the Allah for whom our thesis has been completed without any major interruption.

Then, we are mostly thankful to our co-advisor Dr. Md. Ashraful Alam sir for his kind support and advice in our work. He is always there for us whenever we needed help.

And finally gratitude to our parents, without whose throughout support it may not be possible. With their kind support and prayer, we are now on the final stair of our graduation.

Table of Contents

List of Tables

List of Figures

List of Acronyms

RGB	Red, Green, Blue
BGR	Blue, Green, Red
YOLO	You Only Look Once
GPS	Global Positioning System
COCO	Common Object in Context
RCNN	Region-Convolutional Neural Network
GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture
IDE	Integrated Drive Electronics

Chapter 1

Introduction

Every day we bump into different types of disabled people in our life. People fail to realize that they are just like a normal human being like us. However, they feel helpless among us due to their disabilities. From this point of view, my teammates and I have proposed an idea for the blind and visually impaired which will support them in their everyday path way. Mainly our idea is to train a model using faster RCNN and implement through GluonCV library. This model will detect and extract images from live video or show, then measure the distance with the user and gives an audio alarm for the user. Moreover, we are using webcams from three angles, which covers user's front and left-right side area. Now this sums up our complete proposed model to assist blind and visually impaired in their movements.

1.1 Motivation

Eye, one of the most valuable organs of humans that sense the beauty of our surroundings. Only this senses the true meaning of light and dark or bright and dull. It is said that human can sense up to 80 percent of the impressions with the help of our eyesight. "Eyes show the strength of one's soul." – is one of the expressions that we use to describe the deep connection with the mind and eyes. Unfortunately, there are millions of people in the whole world have problems with their eyesight. Some may have been blind since birth, and some other have lost their sight because of any accident or illness. Experts have found that about 750,000 people in Bangladesh are suffering from blindness while approximately 253 million people live with moderate to severe vision impairment and 36 million people are blind in a discussion. [1] At the stage of having no sight, eyes opt to other sense that will take over the responsibilities of eyes in every sphere. Blind people need

to cooperate with other senses like- they can use their hearing sense or smelling sense with proper training. However, only few of the sufferers take the help of existing technological aids to improve their situation. Hence the need for supportive devices for navigation are increasing day by day whereas some of the supportive tools are not available or do not have easy access for all. Till now, the simplest and the most affordable tools for blind or visually impaired are trained dogs and the traditional white cane for their path detection [2]. However, these tools are very popular, but they cannot provide all information they need for safe movement in roads. According to WHO approximately 80% of vision impairment or blindness globally can be cured by Governments aids [3]. As there are different types of devices to prevent and treatment for blind or impaired people. One of the common device of assistive technology is Electronic Travels Aids (ETAs) which collect and transfer information about the surroundings to the user though sensor cameras, sonar scanners or laser scanners. This technology follows the rules of National Research Council [4-6]. Another technology is Electronic Orientation Aids (EOAs) that provide instruction to the user in unknown place [7, 8]. Position Locator Devices (PLD) uses GPS technology, which shows user's precise position. Wahab et al. researched the Smart Cane product development for object detection and developed accurate navigation instructions [9]. Eye Substitution is another embedded device that helps to navigate and direct visually impaired people. The researchers implemented an Android application to implement the proposed algorithms [10] Fusion of Artificial Vision and GPS (FAV&GPS) device for blind people has been developed to enhance the visualization of the location of the user and to locate the surrounding objects using two functions: a map matching method and artificial vision [11].

1.2 Aims and Objectives

After observing all of these papers and so on, we have decided to propose a model for blind that will help them to choose right path to home, work or shop. It is not only handy but also it has easy access for each sphere of people. In our project, it presents a dynamic object and command detection system based on image processing by training faster RCNN that matches the needs of the virtual impaired or blind people in their everyday life. Here we use cameras in three angles- front, left and right to detect the obstacles or objects in their walkway or sideways. As visually impaired are more dependable on their hearing sense, our model will transfer an audio output to the user. That voice will guide him or her to their destination safely. Our aim is to spread the thought that blind people are also capable of doing everything at ease.

Chapter 2

Related work

Past or present, there have been different systems and devices related to the navigation assistance for visually impaired or totally blind people. Each of them has both advantages and drawbacks. Hence, in a paper, they proposed a system that is a sensor module to cover the head area for blind people. This object detector gives two different type output like buzzer mode and vibration mode and modes are controlled by the user. This device is using proximity IR sensor for detection which is placed on a stepper motor [12]. However, this system requires android mobile assistance. Another paper proposed an intelligent walking stick for the blind people. This system is using PCB unit which consists of microcontroller, Bluetooth HC05, MAX232, ADC 0808 and IR sensors [13]. Moreover, they are using RFID sensor and create an android application which gives the

audio output. Whereas, this system cannot cover near to head area. There are other researches on visually impaired people that the walking stick system consists of a GPS that allows the VIP to know about the outdoor area. This system send massage to authorized people when emergency occurs by GSM. Though this system is not user friendly and only cover limited range of outdoor area [14].

Additionally, there is a paper in which they proposed a model where they use Gabor filtering, capture 2D image & convert them into RGB images into Gray scale to detect the staircase and escalator. They also use Microsoft Kinect and faster techniques to assist the blind people. Their system also works in different environment such as foggy, blur, low light, and sunlight with different angle of view point. They only help impaired people to go through the staircase but it cannot deal with the dynamic objects [15]. In another paper, they present an assisting device which can support visually impaired to move safely and easily inside home or any complex corridors at home. They use AR, depth sensor, vision enhancement algorithms to prepare the smart guiding glass. It can only help blind people to move inside but cannot help them to move outside home like busy roads or other places because it cannot deal with the dynamic objects [16]. Furthermore, a project that uses YOLO model to get accurate directional knowledge and the relative distance. They use Microsoft Kinect to estimate the depth and plug-in for Unity-based game program to generate 3D audio output. Their project can detect real time objects and locate a blind person by giving 3D audio messages. However, it is not applicable for outside like roads because it only works with the static objects not the dynamic ones [17].

These different works and researches encouraged us a lot in our work. It also helps us with more and more informative ideas with our model.

Chapter 3

Methodology

3.1 General Description of Proposed Method

This chapter describes the methodology of the model for dynamic object and command detection for assisting blind people. It includes the explanation of dataset to be used in implementing and testing the model, as well as the description of how it detects any dynamic object and gives alarm. It also describes each part of the proposed method including faster RCNN algorithm which was implemented using python GluonCV library. There is different trained model based dataset named COCO, which is also used in our proposed model.

3.2 Model Design

The objective of the proposed dynamic object detection model is to detect all the objects was trained using faster RCNN algorithm which was implemented using python GluonCV library. The model needs a process that takes video as an input data, systematically process frames from the video input, calculate the approximate distance of the object from the blind people. Then it generates audio output according to the measured value. Figure 3.1 provides a block diagram of the model design.

Figure 3.1: Block diagram of the proposed model

Again, Figure 3.2 shows the workflow of the whole proposed model. The model works step by step according to the workflow.

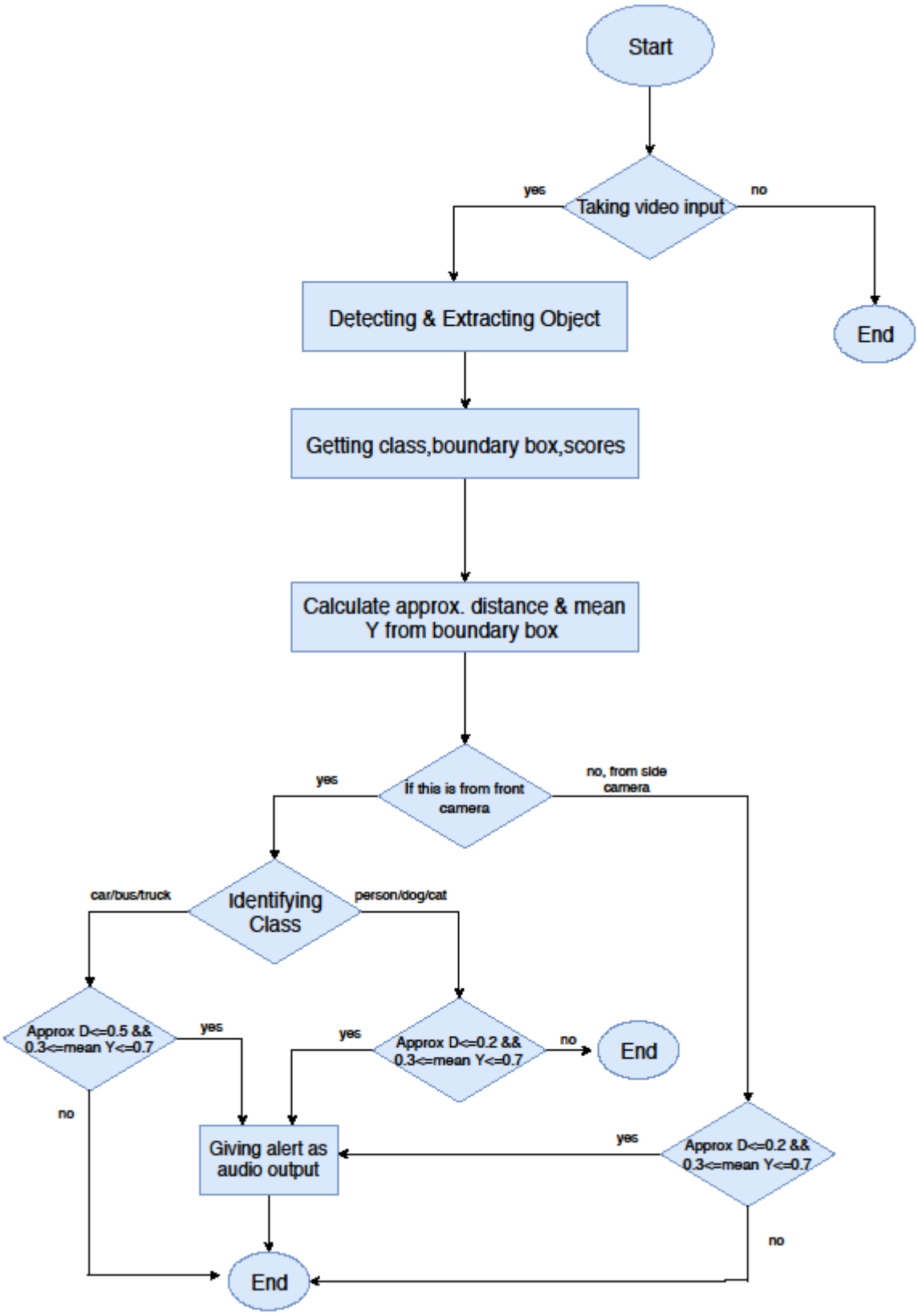

Figure 3.2: The flow chart of the proposed model

3.3 Description of the Flowchart:

The proposed model has been followed some steps so that we can successfully detect our required object and give alert at the accurate time. This model at first takes a video as an input and starts working. When the video input is successfully taken, then by using the faster RCNN algorithm the objects are extracted from individual video frames. After that, we get 3 outputs in return such as class, bounding box and scores.

Here class defines what type of object it can detect. The object can be a car or truck or bus and it can also be a person or dog or cat. From the bounding box we can get Xmin, Xmax, Ymin, Ymax by which we can calculate approx. distance (D) and mean Y.

Scores defines the percentage of an object which is actually looks like that the object from the trained dataset. So that we can confirm about the actual object from the video frame by looking at the percentage.

Next we consider about the camera position, if the object is coming from the front camera and if is either a car or a bus or a truck then we measure the approx. D and mean Y. The model gives alert according to the compared calculation. If approx. D <= 0.5 and 0.3<= mean Y <= 0.7 then the model gives alert. On the other hand, if the detected object is a person or a dog or a cat and approx. D <= 0.2, 0.3 <= mean Y <= 0.7, then the system gives alert to the blind person.

However, if the object is detected from the side camera in that case approx. D <= 0.2 and 0.3 <= mean Y <= 0.7, then the alert is given to the blind person. By following the alerts, a visually impaired person can move easily outside home or crossing the road by himself.

3.4 Faster R-CNN Network:

Region proposal network (RPN) for generating region proposals and a network using these proposals to detect objects are the two networks of Faster R-CNN. The time cost of generating region proposals is much smaller in RPN, when RPN shares the most computation with the object detection network.

An intuitive speedup solution is to integrate the region proposal algorithm into the CNN model. Faster R-CNN is constructing exactly a single, unified model composed of RPN (region proposal network) and fast R-CNN with shared convolutional feature layers.

This is a pre- train CNN network on image classification tasks. So we have to train our model with COCO dataset. Region proposal network is initialized by the pre-train image classifier. It gives two types of samples positive samples have intersection over union is greater 0.7 on the other hand negative samples have less than 0.3.

In MXnet faster RCNN returns raw outputs of class id, bounding box, scores. These three are mostly needed for our model. Calculation is depending on the bounding box. We can calculate approx. distance and mean Y from the bounding box. We get the approx. value of Xmax, Xmin, Ymax, Ymin from bounding box. A lot of intermediate values are generated but we use the approximate values for calculation. By using faster RCNN algorithm which it is implemented using python GluonCV library our proposed model is trained for detecting objects.

3.5 COCO Dataset

When it comes to object detection dataset, we selected COCO dataset. . COCO stands for common object in context [4]. It is a large library which has loads of labeled object. The final version COCO have approximately 164k images in which 118k are trained. It presents 172 classes where

80 thing classes, 91 stuff classes and 1 classs is ‘unlebeled’. This dataset has other highlights like-heavy pixel level annotations, complex context among stuffs and things. Those annotations of COCO are mostly useful for semantic segmentation or object detection.

Here is the display of level hierarchy of COCO stuffing that includes all classes:

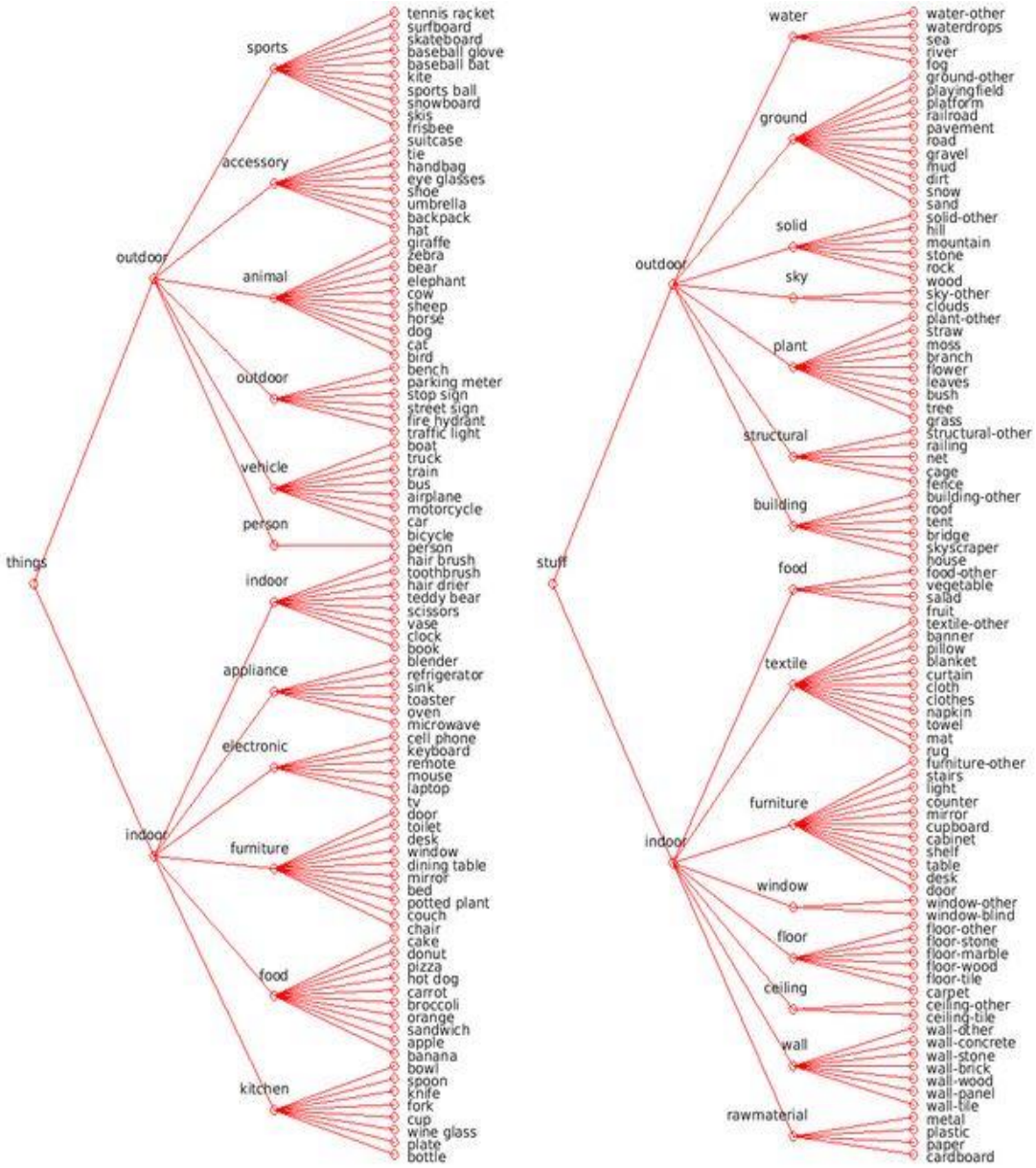

Figure3.3: Level hierarchy of COCO dataset

There's a connection between COCO dataset and GLUON to prepare the dataset. To train data, its necessary to pass through dataset transformation and then load it with GLUON.

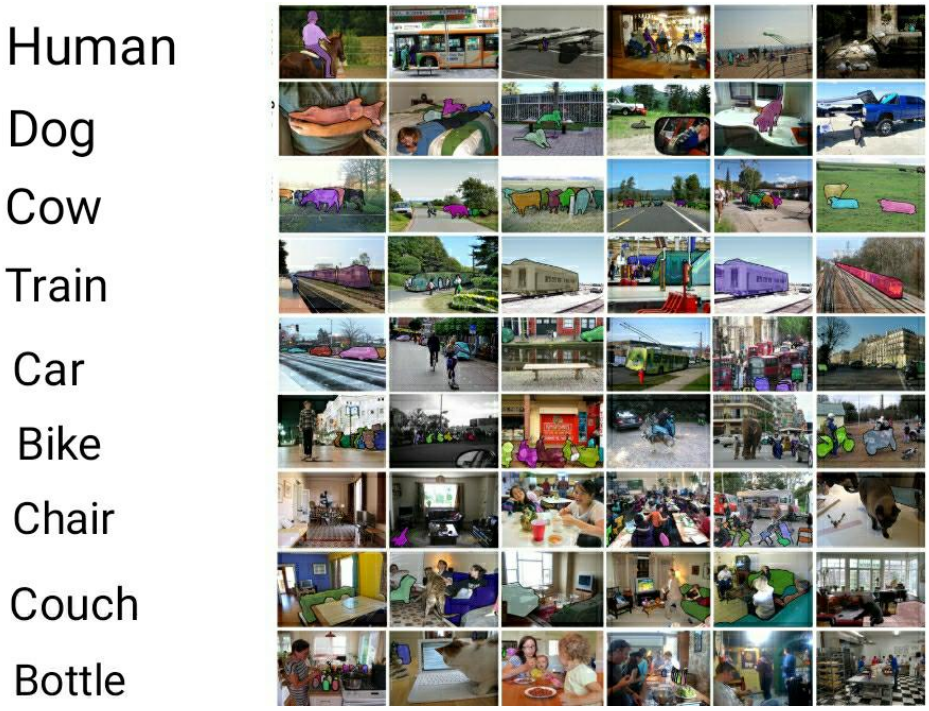

Figure3.4: COCO dataset

3.6 GluonCV

GluonCV is a Deep Learning Toolkit for Computer Vision and natural language processing which provides state-of-the-art pre-trained models and modular APIs with flexible building blocks. Modular APIs are used for enabling efficient customization where user can train, design or interface by using efficient components across different models. This library is based on Apache MXNet. Since Python is an agnostic platform, GluonCV is implemented in Python and available for systems running Linux, OS, and Windows. The advantages of using GluonCV is that have 100 contributors worldwide on GitHub. We are using GluonCV in our project because it can work

without GPU in some cases. Moreover, it can be simulated by different programming languages so that developers can use according to their preferences and it is also a lighted library.

Chapter 4

Implementation

This chapter describes the implementation of the proposed system of dynamic object detection for assisting blind people. The system was implemented in 3 steps. Firstly, a model was trained for detecting objects. Then this model was used for object classification and localization. Finally, this system gave alert in given conditions. The model for detecting objects were trained using faster RCNN algorithm and it was implemented using python GluonCV library [1]. GluonCV is a deep learning toolkit which gives the opportunity to implement deep learning algorithms easily. It is integrated to Apache MXnet which is an open source framework [2]. MXnet is an effective tool for training and deploying neural networks and it supports almost all popular deep learning algorithms. Our model was trained by the train_faster_RCNN.py file provided by gluonCV documentation [3].

This chapter also provides the results of the implemented system. Anaconda Spyder is used to run the whole process.

4.1. Implementation

The proposed system for dynamic object detection for assisting blind people consists of three files.

Table 1: Description of system files used in the model:

File Name	Description
mscoco.py	This script was provided in the official GluonCV website to download cocodataset and unzip them.
train_faster_RCNN.py	The file was provided in the official GluonCV website to train a model using faster RCNN algorithon.
Detection.py	This file contains the code for detecting objects and giving alerts in given conditions.
fRCNN_0000.params	This file contains the trained weights after training process.

The reason for choosing GluonCV library because it is easier to apply than other libraries like tensorflow . It is an open source toolkit. Dataset preprocessing and indexing is less time consuming here and it contains a consistent interface for retrain models.

4.2 Used Libraries

Apache MXNet: The detailed form of MXNet is mixing and maximing network. It is an opensource deep learning framework. It supports almost all popular operating systems. MXNet framework can be used in C++, Python, R, Julia, Perl etc. Therefore, there is no need of changing platforms. Moreover, MXNet models are portable and they require less amount of memory.

GLuonCV: GluonCV is a deep learning toolkit. The main reason behind creating this toolkit was replicating experimental results from published papers as it is the most difficult part. GluonCV is a powerful tool specially for debugging and reproducing experiment results. It supports multiple algorithms and it can be applied on almost all operating system. In our system, we have used GluonCV api, for train our model and get weights.

OpenCV: OpenCV is a python library for real-time computer vision. In a word, it is used for image processing. It is popular for its simplicity and code readability. OpenCV supports a multitude of algorithms and it can do high speed CUDA operation. OpenCV-python is basically the python api for python which combined OpenCV C++ api and python language together. In our system, we have used OpenCV for taking video inputs and showing outputs.

Numpy: Numpy stands for numerical python. It is a popular library of python which is used almost all deep learning projects. It supports large and multidimensional arrays. By using Numpy, it is easy to manipulate data and store them as matrix.

Matplotlib: Matplotlib is a python library which is used for plotting figures and charts. It is basically a 2D plotting visualization library. In our project, we have used Matplotlib to visualize data for multiple times.

FFpyplayer: FFpyplayer is a library for playing and writing media files. In our project, we have used it for giving output as audio form.

4.3 Dataset Preparing

The model for detecting objects was trained using COCO dataset. Its large dataset helps us to detect as much object as possible. GluonCV provides a script to download this large dataset with annotation.

Once downloaded, the files were unzipped. There were three separate folders - train2017, val2017 and annotations. The first two folders contain images for training and validation. The last folder contains json formatted files of annotation. But all of the files of annotation folder were not used. Only instances_train2017.json and instances_val2017.json were used.

After that, gluoncv.data.COCODetection method was used for loading images and labels. No transformation was done in this step. Here is the result of loading data and creating index. This screenshot was captured from Ipython console of Spyder IDE.

```
loading annotations into memory...
Done (t=20.55s)
creating index...
index created!
loading annotations into memory...
Done (t=1.00s)
creating index...
index created!
Num of training images: 117266
Num of validation images: 4952

In [2]:
```

Figure 4.1: A sample of indexed training and validation data

Here after removing unlabeled images the number of total training image is 117266 and the number of validation image is 4952. After indexing, it was checked if the indexing was created correctly. Bounded boxes and class ids were found from train label and it was found that indexing was successful. The 85th image from the train2017 folder was chosen for the test which contained a bus with a person inside it. The result of the test is given below. This figure shows the properly labeled bounded box.

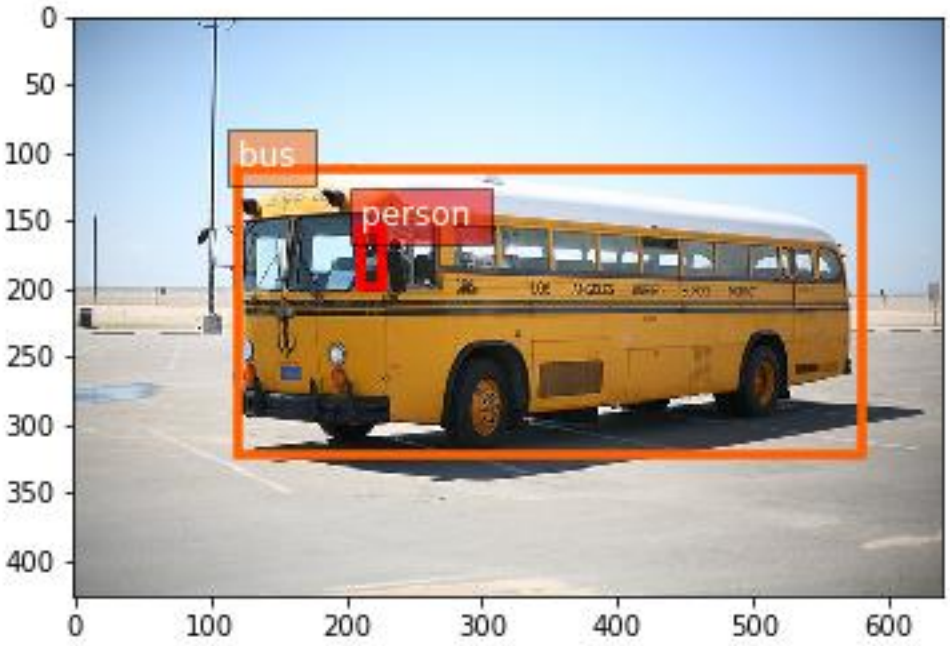

Figure 4.2: A sample of validation of indexing

4.4 Faster RCNN Algorithm Implementation

Faster RCNN algorithm was implemented in train_faster_RCNN.py file. This file was provided by GluonCV official website [3]. This file takes some arguments while running for example epochs, seeds, dataset etc. to train this model, 18 epochs were done and batch-size was 20. By declaring pre trained base false, this model was prevented from using pre trained weights and the weights were updated during the training process.

In this step raw images were turned into tensors. A tensor is a mathematical entity which describes a linear mapping from one set of objects to another. GluonCV is integrated to MXnet which follows BCHW(Batch x Channel x Height x Width) format of tensor. Faster RCNN is a powerful

algorithm which can deal with raw images with various aspect ratios. **gluoncv.data.batchify.Append**() is used for this process. Here images are not stacked or padded and unchanged with their own shape or aspects. For iterating through the whole dataset for multiple times during the training, Dataloader() method was used to load data in small batches.

In MXnet faster RCNN returns class id, scores and boxes. Score means how much perfect is the predicted object. Bounded boxes are co-ordinates of detected object in the frame. During the training process, faster RCNN behaves differently than other algorithms. A huge amount of intermediate values is generated. During training the model, rpn_score and rpn_box was generated which are the raw outputs from RPN's convolutional layer.

To train the model for object detection, four type of losses were involved throughout the training. This loss functions are builtin mxnet gluoncv classes when were built with different equations. RPN class loss was calculated by sigmoid binary cross entropy loss function,RPN box losses was calculated by huberloss function . RCNN class loss was calculated by softmax cross entropy loss and RCNN box loss was calculated by huberloss. To speed up the training process, MXnet let the CPU to pre calculate RPN training targets. RCNN training targets were created from the intermediate outputs with stored target generator.

After calculating the loss function and training targets, the training loop was iterated according to the batch size and epochs. When the iterations were completed, the trained weights were saved as fRCNN_0000.params.

4.5 Detection of Objects

The detection part and relative distance measurement part were done in detection.py file. In this part, fRCNN_0000.params file was loaded in the network. For capturing video and getting the frames, opencv library was used. Here some frame preprocessing was done. For testing purpose, we run the detection process using webcam.

Figure 4.3: Taking input using webcam.

4.5.1 Hardware Setup

Three webcam were used in the whole process. These cameras were attached to a spectacle. One camera was set at the frontal part of the spectacles for getting the front view. Other two were used for getting the view of left and right side. By using these spectacles with three cameras, now it is possible to cross the road safely during a traffic signal.

4.5.2 Detected Outputs

OpenCV captures frames in BGR color format but for detecting objects RGB format is needed. Therefore, each frame was converted into RGB format. Frames which are shorter than 512px, they

are resized into 512px and max size was set to 700px as the longer side of the image remains smaller than 700px.

After that, the frame was fed into the network which returns three values – class id, scores and bounding boxes. cv_plot_bbox()method from gluoncv was used to plot the bounding boxs,scores and classes on the output frame and cv_plot_image() was used to plot the output frame with these values.

4.6 Approximate Distance Measurement and Alerting

In our proposed system, it was not possible to get the real distance of detected object. Because focal length is different for different cameras and it is not a food approach to calculate focal length manually each time while using a different camera. To solve this issue, approximate distance was used instead of real distance which worked perfectly for this system

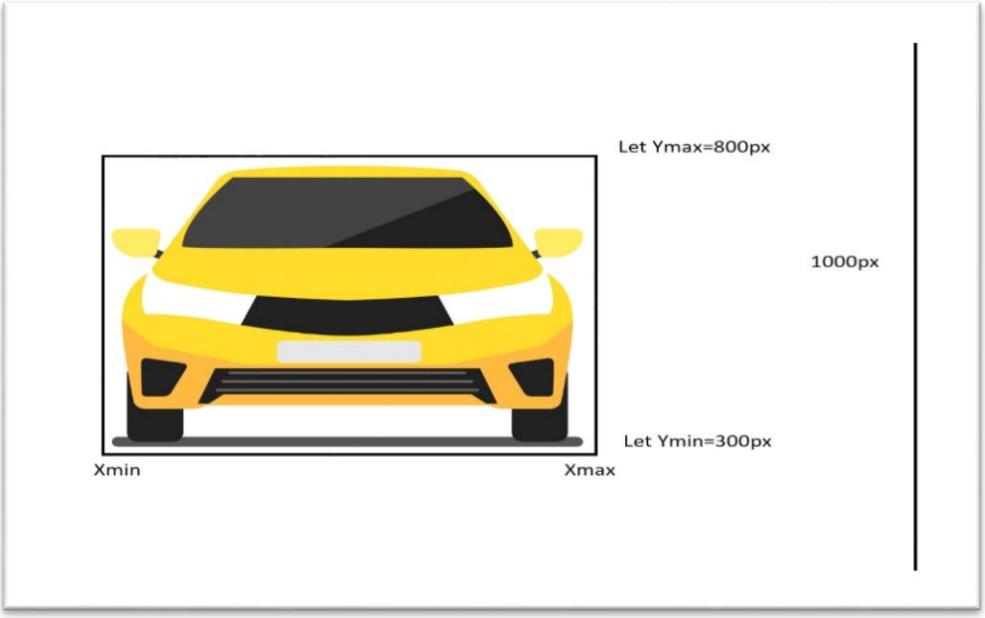

Figure 4.5: Measuring relative distance.

Figure 4.5 refers to a detected object with bounding boxes. Each bounding box contains four values(Ymin,Xmin,Ymax,Ymin) which are the co-ordinate of bounding box on the frame. Here, an equation was proposed for approximate distance –

Approximate distance = frame height – (Ymax –Ymin)

Ymean=(Ymax+Ymin)/2

For the simplicity of the measurement and less memory use, approximate distance and mean of Y were divided by frame height. Now how this process is working with approximate distance and Y mean, that can be cleared with little example.

Lets,

frame height= 1000px

Ymin= 300px; Ymax=800px

Therefore, approximate distance = (1000-(800-300)) px= 700px and MeanY = (800+300)/2= 550 px

After scaling approximate distance= (700/1000) px =0.7px and meanY= (550/1000) = 0.55px

The proposed system is giving alert for four classes (car,bus,truck,person,cat,dog) which means there are two categories here – vehicles and living instances.

If the detected object is car, bus or truck, the system will alert when approximate distance is less than 0.5 but if the detected objected is person, dog or cat the system will alert when approximate distance is less than 0.3. But there was also an issue arisen here. If some objects are close to the camera but there is not any of collision.

Figure 4.6: Detected objected with different localization in frame.

It can be observed from figure 4.6 that, there is not any possibility of collision with the left and right car but they are very close to the camera. Therefore, the system will give wrong alerts. Toovercome this issue, the system will only give alert if MeanY of detected object is in the range of 0.3 to 0.7.

If the detected object is person, cat or dog, the system will alert when approximate distance is less than 0.3 as they are living instances and can avoid collision easily. However, MeanY technique has also been followed to avoid unnecessary objects which are very close to the camera but there is not any change of collision.

Earlier it was mentioned that we are using three cameras – one for front side and the other two are for side views. For left and right camera, the alert is given when approximate distance is less than 0.2 and MeanY of detected object is in the range of 0.3 to 0.7.

Finally, the alert has been created using audio with the help of FFpyplayer library. The whole process continues while getting video frames as input. However, the perfection of the detection process depends on the resolution of the cameras to some extent.

Chapter 5

Results

5.1 Output

The final output for each detected object is given in the audio form and python ffplayer library was used to play the audio. We had made a test for checking accuracy with our trained weight. Then we did the same thing with the weights of pretrained model faster_RCNN_resnet50_v1b_voc from model zoo.

Figure 5.1: Detected objects during the test

Table 2: Output through the test in the model

The output we found through this test is given below:

Item	Frames	Detected Objects	Wrong Prediction
Weights of trained model for the system	30	8	4
Weights of pretrained model	30	5	2

We have run this test for four times with different video frames and specified them as sessions. After that, we created a bar chart for the result. Number of frames in session01, session02, session03, session04 are respectively 30,60,15,24.

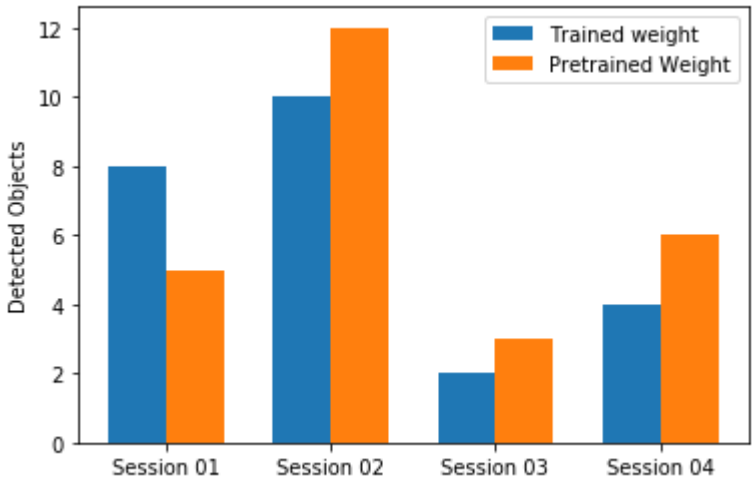

Figure 5.2: Comparison of trained weight with pertained weight for detected objects (Higher is better)

In figure 5.2, session 1 and 2 is higher than the results of session 3,4 as number of frames are higher in session 1 and 2. Pretrained weight (gluonCV model of faster RCNN) worked better in session 2,3,4 and our trained weight worked better in session 1

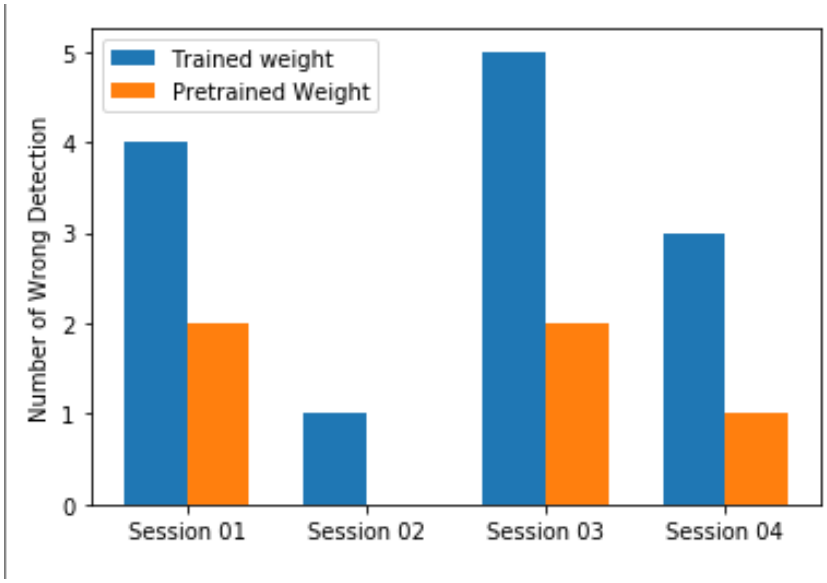

Figure 5.3: Comparison of trained weight with pertained weight for detected objects (Lower is better) (Note: no wrong detection in pretrained weight in session2)

In fig 5.3, there is no wrong detection in pretrained weight (gluonCV model of faster RCNN) in session01, therefore, there is no bar here.

5.2 Accuracy Rate

After running this test more and more, there are some better impacts in our expected output. Each time we get to make the model more accurate. At the end, our model has given 74.65% accuracy with 100 epochs, which is good at this level.

Here, we have given the model accuracy graph between train and validation sets:

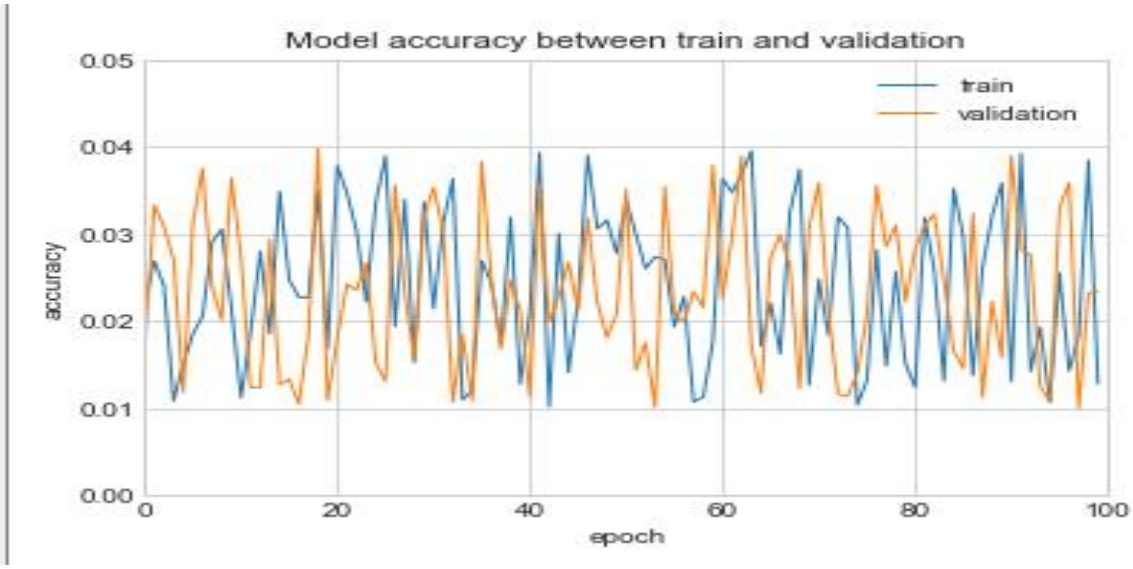

Figure 5.4: Accuracy rate of the proposed model

Chapter 6

Conclusion

We designed and implemented the proposed system that ensures the mobility and safe navigation for blind people. The system must require prior knowledge about the object size, shape or distance. Our work results in a simple, portable system that detects dynamic object and then gives audio command for assisting visually impaired people. It is a travel friendly device based on video and audio feedback. Several experimental results in outdoor and indoor environments showed that this system is very helpful for this user group. Additionally, more and more disabled people are suffering everyday due to the lack of perfect path detector device. Especially there is not much development in this system in Bangladesh. Here GPS may not be available everywhere due to low signal. Hence our paper comes up with a more developed and user-friendly device that assist path detection for blind people. We have implemented the system with different techniques to design our predicted model so that we can present a model that is more efficient and give less error. The proposed system achieved 74.65% accuracy with 100 epochs. Every human being has the right to live with comfort and ease, so it is our duty to help the blind and visually impaired in their daily path detection. It is said that even eyes are useless when the mind is blind. Thus, we have tried to waken their mind through our proposed model with an audio output to waken their eyes to give them a better world.

Chapter 7

Future Scope

Our initial steps are hopefully promising and helpful for visually impaired people. Our future improvements depend on flourishing training dataset. For now, we are notifying the blind person in what distance a car is coming. Actually, for now we are measuring the distance of a car from the person whereas; this system can also be trained into two ways. In order to do so, we will store 1000 pictures of the front side of a car will be stored in one folder and 1000 pictures of other side of a car will be stored in another folder. So that the system will detect not only the distance of a car but also it will tell in which side the car is coming or in which side of the car, the blind person is standing. The more pictures we will store to train the system the most accurately the system can give alert. Furthermore, we will use Raspberry Pi-style Jetson Nano along with the three web cameras for making the prototype of our proposed model. This type of Raspberry Pi is a powerful low-cost AI computer. This developer tool is a board tailored for running machine learning models and using them to carry out tasks such as computer vision. These are some challenges for making this proposed system more efficient in future. Due to lack of time, we are now proposing this model in short but when we will make this device more compatible for business purpose to improve the visually impaired people's life.

References

1. Experts: 750,000 people suffer from blindness in Bangladesh. (2018, October 11). Retrieved from https://www.dhakatribune.com/bangladesh/2018/10/12/experts-750-000-people-suffer-from-blindness-in-bangladesh

2. Blasch B.B., Wiener W.R., Welsh R.L. Foundations of Orientation and Mobility. 2nd ed. AFB Press; New York, NY, USA: 1997

3. Vision impairment and blindness. (n.d.). Retrieved from https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

4. Liu J., Liu J., Xu L., Jin W. Electronic travel aids for the blind based on sensory substitution; Proceedings of the 2010 5th International Conference on Computer Science and Education (ICCSE); Hefei, China. 24–27 August 2010.

5. Sánchez J., Elías M. Guidelines for designing mobility and orientation software for blind children; Proceedings of the IFIP Conference on Human-Computer Interaction; Janeiro, Brazil. 10–14 September 2007.

6. Blasch B.B., Wiener W.R., Welsh R.L. Foundations of Orientation and Mobility. 2nd ed. AFB Press; New York, NY, USA: 1997.

7. Farcy R., Leroux R., Jucha A., Damaschini R., Grégoire C., Zogaghi A. Electronic travel aids and electronic orientation aids for blind people: Technical, rehabilitation and everyday life points of view; Proceedings of the Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments Technology for Inclusion; Los Alamitos, CA, USA. 9–11 July 2006.

8. Kammoun S., Marc J.-M., Oriola B., Christophe J. Toward a better guidance in wearable electronic orientation aids; Proceedings of the IFIP Conference on Human-Computer Interaction; Lisbon, Portugal. 5–9 September 2011.

9. Wahab A., Helmy M., Talib A.A., Kadir H.A., Johari A., Noraziah A., Sidek R.M., Mutalib A.A. Smart Cane: Assistive Cane for Visually-impaired People. Int. J. Comput. Sci. Issues. 2011;8:4.

10. Bharambe S., Thakker R., Patil H., Bhurchandi K.M. Substitute Eyes for Blind with Navigator Using Android; Proceedings of the India Educators Conference (TIIEC); Bangalore, India. 4–6 April 2013; pp. 38–43.

11. White C.E., Bernstein D., Kornhauser A.L. Some map matching algorithms for personal navigation assistants. Trans. Res. C Emerg. Tech. 2000;8:91–108. doi: 10.1016/S0968-090X(00)00026-7.

12. Prof. Seema Udgirkar1, Shivaji Sarokar2, Sujit Gore3, Dinesh Kakuste4, Suraj Chaskar. Object Detection System for Blind People. International Journal of Innovative Research in Computerand Communication Engineering; Vol. 4, Issue 9, September2016.

13. Kher Chaitrali S., Dabhade Yogita A., Kadam Snehal K., Dhamdhere Swati D., Deshpande Aarti V. An Intelligent Walking Stick for the Blind. International Journal of Engineering Research and General Science Volume 3, Issue 1, January-February, 2015.

14. A.Sangami1, M.Kavithra2, K.Rubina3, S.Sivaprakasam. Obstacle Detection and Location Finding For Blind People. International Journal of Innovative Research in Computer and Communication Engineering; Vol. 3, Special Issue 2, March 2015.

15. Zereen, A.N. (2016). *Staircase and Escalator Detection for Visually Impaired* [PDF file]. Bangladesh, DC: Author. Retrieved from http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/8365/14366004_MSc%20in%20CSE.pdf?sequence=1&isAllowed=y

16. Bai, J., & Lian, S., & Liu, Z., & Wang, K., & Liu, D. (2016). *Smart Guiding Glasses for Visually Impaired People in Indoor Environment* [PDF file]. China, DC: Author. Retrieved from https://arxiv.org/ftp/arxiv/papers/1709/1709.09359.pdf

17. Jiang, R., & Lin, Q., & Qu, S. (2016). *Let Blind People See: Real-Time Visual Recognition with Results Converted to 3D Audio* [PDF file]. UK, DC: Author. Retrieved from http://cs231n.stanford.edu/reports/2016/pdfs/218_Report.pdf

18. GLUONCV mxnet. [Online]. https://gluon-cv.mxnet.io/

19. MXNET apache. [Online]. https://mxnet.apache.org/

20. Train Faster RCNN. [Online]. https://gluon-cv.mxnet.io/build/examples_detection/train_faster_RCNN_voc.html

21. COCO Dataset. [Online]. http://cocodataset.org/#home