North South University

Department of Electrical and Computer Engineering

CSE499B – Senior Design Project – Summer 2020

# Navigational Assistance for the Visually Impaired through Computer Vision Techniques

Group Members:

| | |
|---|---|
| Shadab Hafiz Choudhury | 1631335642 |
| Ishrat Jahan Ananya | 1631636042 |
| Sarah Suad | 1632282642 |
| Nabiul Hoque Khandakar | 1631164642 |

Faculty Advisor:

Tanjila Farah
Senior Lecturer and Lab Coordinator
Department of Electrical and Computer Engineering

# Declaration

We hereby declare that the work presented in this report is the outcome of our eight months work performed under the supervision of Tanjila Farah of the Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh. The work was spread over the span of the final year course, CSE499 – Senior Design Project, in accordance with the course curriculum of the Department for the Bachelor of Science in Electrical and Electronics Engineering Program.

Students' Names and Signatures

| | |
|---|---|
| Shadab Hafiz Choudhury | Ishrat Jahan Ananya |
| Sarah Suad | Nabiul Hoque Khandakar |

# Approval

The Senior Project Report on "Navigational Assistance for the Visually Impaired through Computer Vision Techniques" has been submitted by Shadab Hafiz Choudhury (ID#: 1631335642), Ishrat Jahan Ananya (ID#: 1631636042), Sarah Suad, (ID#: 1632282642) and Nabiul Hoque Khandakar (ID#: 1631164642), students of the Department of Electrical and Computer Engineering, North South University, Bangladesh. This report partially fulfils the requirement for the degree of Bachelor of Science in Computer Science and Engineering in February 2021 and has been accepted as satisfactory.

Supervisor's Signature

Ms. Tanjila Farah

Senior Lecturer and Lab Coordinator

Department of Electrical and Computer Engineering

North South University, Dhaka, Bangladesh

Department Chair's Signature

Dr. Rezaul Bari

Professor and Chair

Department of Electrical and Computer Engineering

North South University, Dhaka, Bangladesh

# Acknowledgments

We are extremely grateful to all the Faculty and Staff of Electrical and Computer Engineering Department of North South University for providing us with the opportunity and platform to carry out this project. We would like to thank Ms. Sumaiya Tasneem Haque, Lecturer of the Department of English and Modern Languages at North South University, for providing insight that helped us narrow the scope of our project.

Finally, we would also particularly like to acknowledge and thank our supervisor, Ms. Tanjila Farah, Senior Lecturer of the Department of Electrical and Computer Engineering. We are indebted to her for her efforts, providing guidance and advice over the course of our senior design project.

# Abstract

Blind people face many difficulties in daily life, one of which is navigation. There are several solutions leveraging the use of computer hardware and artificial intelligence to help guide them. However, most current solutions use complicated hardware and so are not suitable for everyone. This project uses deep learning to implement a semantic segmentation algorithm that recognizes walkable areas in an interior environment in real-time, directing users away from obstacles such as furniture or people. We test ShuffleNet v2 and DeepLab v3 and implement the former into an app that can be used on any android phone. The app is capable of recognizing obstacles within two steps of the user and warning them accordingly.

## Table of Contents

## List of Figures

## List of Tables

# Chapter 1

# Overview

## 1.1 Introduction

Visually impaired, or blind people, face many difficulties in daily life. As they are bereft of visual stimulus, they cannot interact or navigate the world around them easily. Locating and picking up an object is a difficult task, as they have to feel around to find out where the object is. In many cases, they also risk knocking other, potentially valuable, fragile or dangerous objects over. They can avoid this by taking their time and being excessively careful. Similarly, although they cannot read written text, they can use braille for physical texts and text-to-speech software for digital text.

Braille has been available for physical media for almost two hundred years, and there are even braille readers for digital screens (though the latter are quite expensive). Text-to-speech programs are also highly developed and ubiquitous in mobile or computer software.

In recent years, artificial intelligence has also been used to help with the first problem – recognizing and locating objects. The field of live object recognition is fairly developed, and though there are no highly popular consumer solutions, they can be expected in the near future.

However, there is no easy way for them to navigate through a room full of obstacles. For most blind people, their perception is limited to the length of a walking stick, which itself has an extremely narrow ‘field of view’. This causes a large number of inadvertent collisions with various obstacles in the room such as furniture and other objects, leading to injuries or damage to the contents of the room.

A walking stick provides extremely minimal feedback to the user. This feedback is usually in tactile form, and as it is transmitted through a narrow device very little detail can be gleaned. It is not possible to distinguish a chair leg from a table leg using a walking stick. Raising the walking stick up higher to differentiate poses various risks such as hitting other people or damaging objects.

In addition, someone using the stick vigorously may damage furniture or knock over other objects at ground level. It may bump into other people, get snagged, and so on. There is also the possibility of damage to the stick.

There are currently 285 Million people around the world with varying levels of visual impairment, ranging from moderate to severe. Out of this 285 million, 40 million are fully blind and have no sense of sight at all.

In Bangladesh, 6 million people exhibit some form of visual impairment, ranging from mild to severe. Estimates for the number of individuals who are fully blind range from 750,000 to 1 million [1], [2].

A lot of work has been done before on helping visually impaired people navigate. However, most prior research has been focused on a hardware-based solution, whether it is a high-tech walking stick equipped with a myriad of devices, or a multitude of sensors mounted all over the body. They mainly used radio or ultrasound waves to detect obstacles.

Despite being effective solutions, one of the major pitfalls of hardware-based solutions is that hardware is expensive to manufacture, import and maintain. In a developing country like Bangladesh, the majority of the 750 thousand blind people are underprivileged and will not be able to bear those expenses easily.

By contrast, software-based solutions have the advantage in that they have no material costs once development is complete. They can be duplicated and distributed for no cost beyond the initial development costs. Therefore, in order to solve the problem of helping visually impaired people navigate, we decided to turn to a software-based solution using computer vision.

One of the primary goals of Computer Vision research has been to emulate human sight and everything that entails. This includes object detection, object classification, motion analysis, etc. Without going into the details of each application of computer vision, the most important aspect for helping blind people navigate is object detection and classification.

The goal of this research is to use purely computer vision to help a blind person gain a rudimentary understanding of an interior area's layout, allowing them to plan out how to proceed. This would be a significant step in making moving around easier for them.

Computer vision technology has advanced a long way. Depending on the type of neural network used, certain tasks can be accomplished at a very high speed even on limited hardware such as on a mobile phone. In a developing country like Bangladesh, there are hundreds of thousands of blind people who have extreme difficulty navigating indoor spaces due to being unable to see the layout of furniture and clutter in a room.

The aim of this project is to develop a straightforward software-based solution that will help the visually impaired navigate indoor spaces by using just their mobile phones, avoiding the costs associated with hardware solutions.

## 1.2. Motivation

It is an indisputable statement that everyone deserves the best possible standard of living. Part of maintaining this statement and ensuring a high quality of life for everyone involves minimizing the impact a disability has in everyday life.

The human body has five major senses that are all crucial for functioning in daily life: sight, hearing, smell, taste and touch. The lack of any one of them strongly affects an individual's ability to move around and communicate, as well as broader things such as educational and job opportunities. One of the group members working on this project is disabled, lacking one of those major senses. Therefore, helping those with disabilities is not just a matter of social duty but also personal.

We chose to work on this project to help visually impaired people move around more easily in their day-to-day life. Vision impairment usually ranges from near-sightedness, where an individual is unable to focus on objects further from him, to total blindness, where they do not have any vision at all.

There are over 40 million people in the world suffering from blindness, out of which around 1 million are in Bangladesh. As mentioned in the prior section, a great deal of the work done so far is not suitable for the average Bangladeshi, due to costs of hardware and maintenance. However, smartphone penetration is on the rise in Bangladesh, having increased from 45% in 2017 to nearly 55% in 2020 [3]. With many external factors, it seemed prudent to focus on mobile phones as a means of distribution of whatever technology we develop to help the visually impaired.

We finalized the scope of the project after interviewing several people who are afflicted by visual impairments.

## 1.3. Project Description

The project Navigational Assistance for the Visually Impaired through Computer Vision Techniques involves exploring various image segmentation models and datasets currently available and curating a dataset of primarily indoor locations, followed by the design and development of a model that can infer the layout of a room using a deep learning algorithm to segment the image of a room into different sections that denote the floor, walls, furniture, clutter, etc. The app would then convert the segmented image into an audial description that a visually impaired individual can understand.

## 1.4. Project Goals

Overall, the goal of this project is to design and develop a functional substitute, or at the minimum, a complement, for the walking stick. Numerous solutions exist for outdoors navigation over longer distances, so this project aims to specialize in indoors navigation instead.

The final project outcome will contain a mobile phone app that integrates a computer vision algorithm. This algorithm will receive data in the form of a video stream from the phone's camera. This video stream is broken up to individual frames, which allows us to carry out semantic segmentation on them. The results of the segmentation tell the program which areas in front of the user have clear, unobstructed floor, i.e., are walkable, and which areas are not. The app will give output to the user in the form of a voice commands, continuously keeping them apprised as to whether the area in front of them is clear or blocked.

# Chapter 2

# Index Terms

## 2.1. Deep Learning

Deep Learning is one of the major approaches to developing Artificial Intelligence and Pattern Recognition. Originally introduced by Ivanhenko [4] in 1967 and later used by LeCun [5] to solve the problem of recognizing handwritten documents, it has since then been developed to the extent where it can be used to solve complex problems, such as classification, feature clustering, prediction, feature extraction, et cetera. The principal components of a Deep Learning approach are Neural Networks.

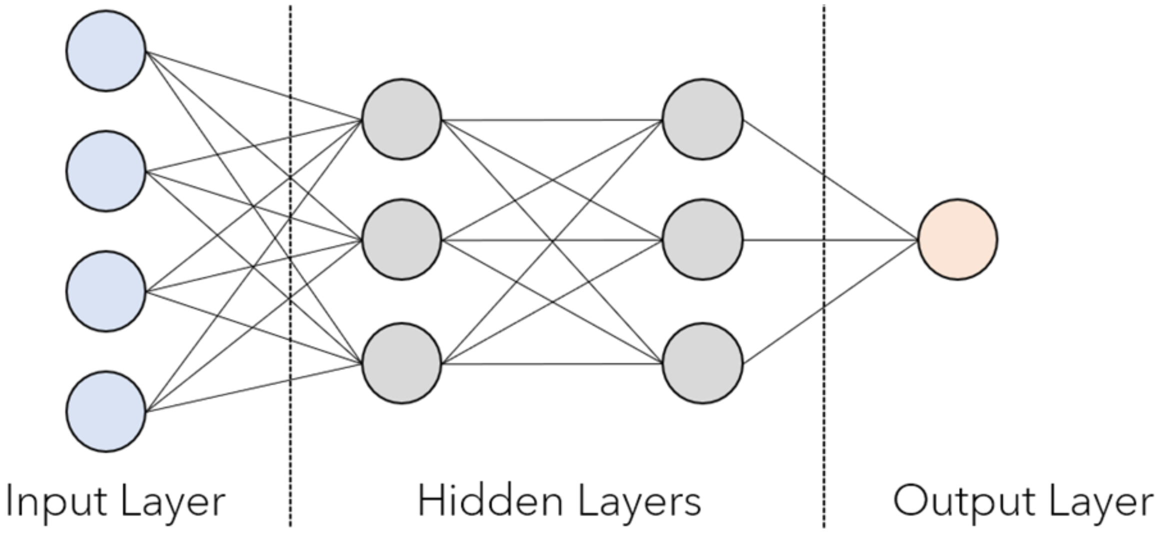


Fig. 1. An example of a Neural Network.

Neural Networks are named thus because they attempt to mimic the structure of a human brain. A neural network is made up of ‘neurons’ or nodes. Inputs are passed through several layers composed of nodes. These layers are typically referred to as hidden layers, as their precise activities are typically not apparent to the developers or users. Simple neural networks tend to use less than 10 layers. Complicated, deep neural networks can consist of hundreds of layers.

With each layer, an input is broken down to individual features. Each neuron receives a parameter or ‘weight’ from the connected neurons, and depending on various other information this particular neuron would decide to pass on its value to the next layer. Information can propagate forwards and backwards based on this, allowing neural networks to break down features from complex inputs such as images easily.

## 2.2. Computer Vision

Computer Vision is an umbrella term for an interdisciplinary field that focuses on algorithms and applications to enable computers to understand, process and output visual information in the same way humans do. In order to do this, initial studies into the field arose from generalizing human vision and cognition [6], before it could be translated to a form that could be computed.

The various uses of computer vision include image editing and enhancement [7], 3D modelling [8], robotic guidance and navigation, et cetera. Most of these applications of computer vision rely on sophisticated artificial intelligence systems.

Using Deep Learning and CNNs, it is possible to do things such as locate an object within a picture and identify it, recognize a person's face and emotion, or divide up a picture into different sections such as 'ground', 'sky', 'person' etc. The last technique is known as segmentation and is the focus of this project.

## 2.3. Convolutional Neural Network

There are several different types of neural networks that can be used in Deep Learning. They include Feed-Forward Networks, Recurrent Neural Networks, Generative Adversarial Networks and many others. However, for image segmentation and classification tasks, the most effective type of network is a Convolutional Neural Network (CNN). CNNs are the most common type of neural network used for visual work such as image and video processing.

CNNs are highly effective for visual processing and pattern recognition because they can break down an image by running filters over it. Each hidden layer involves a filter, or ‘convolution’, being carried out at different points of the image. As an image passes through different layers, the resolution decreases but most of the information is retained.

## 2.4. Semantic Segmentation

There are two types of Segmentation applied in computer vision. Instance Segmentation extracts several unique instances of an object from an image. The different instances are labelled with different colours. Semantic segmentation extracts different sections of an image without regards for unique instances. An overview of the various approaches used to carry out semantic segmentation is given in [9].

The following figure gives a comparison of the two.

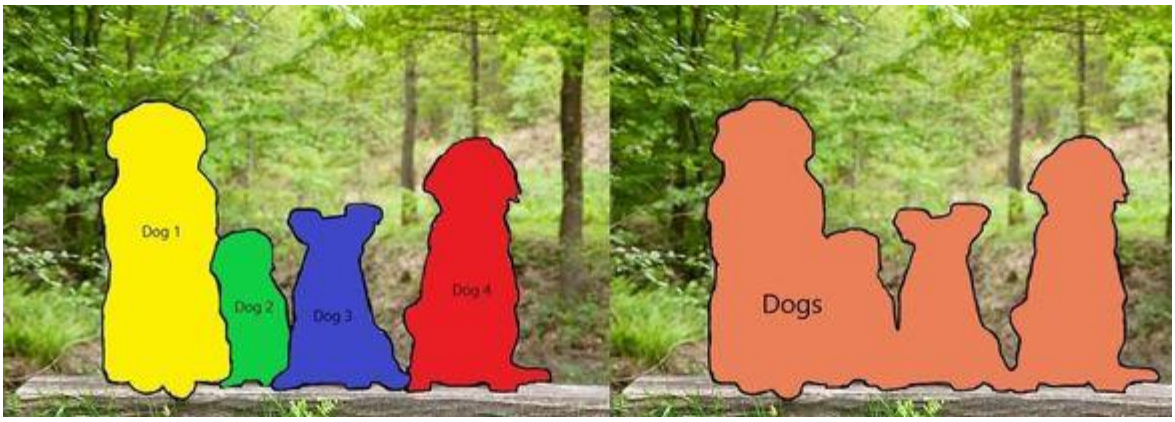


Fig. 2: Instance Segmentation vs Semantic Segmentation

The basic process of semantic segmentation can be abstracted as such: the neural network learns features from a dataset of images that are already annotated (have different sections to be segmented marked in different colours). By extracting features, it learns how to segment images and labels each individual pixel of an image with an appropriate colour.

Semantic segmentation algorithms use encoder-decoder networks, which are essentially a combination of a normal CNN, the encoder, followed by a reversed CNN, the decoder. Initially, the network breaks high-level features down, carrying out object detection. It locates different classes in an image in different positions. In the example of Fig. 2, the first step would be to locate the general location of the dogs. After the general regions have been located, the exact edges are inferred by comparing the pixel colours of each region on either side of the rough contour. This gives the output as a segmented image.

The decoder is necessary because unlike classification problems, segmentation requires an output that is of similar or the same dimensions as the input. The decoder upscales the image in order to achieve this.

# Chapter 3

# Literature Survey

## 3.1. Related Literature on Computer Vision

Computer vision has been a highly researched field of artificial intelligence for a long time. Liu et al's paper [10] gives a good overview of recent work done in the scope of semantic segmentation, including a comparison of several types of neural network for this purpose. Semantic segmentation has been used in a large number of contexts and there are multiple databases available for this. Other applications of semantic segmentation include medical imaging [11] to distinguish different sections of a body image. It has been used in self-driving cars [12] to detect boundaries between the road and the sidewalk, as well as

Detecting interior environments in real-time has been explored before [13], [14], as it is very important in robotics applications. These papers show that it is possible to create accurate segmentation and 3D maps of cluttered interior environments. However, they also emphasize the use of RGB-D input rather than RBG inputs. RGB-D stands for Red-Green-Blue-Depth. In order to measure depth, specialized sensors such as Kinect sensors are needed in addition to the image sensor on a mobile camera.

## 3.2. Related Literature on Navigation for the Blind

Technological solutions have been extremely helpful to blind people [15]. OCR approaches let the visually impaired read texts that are not in braille [16]. Mobile apps have been developed that can recognize some everyday objects [17]. There are even solutions for helping the visually impaired maintain social distancing in public [18].

A lot of work has been done on providing navigational assistance for visually impaired people. However, most prior research had been focused on a hardware-based solution, whether it is a high-tech walking stick equipped with a myriad of devices, or a multitude of sensors mounted all over the body. They mainly used radio or ultrasound waves to detect obstacles. While this is an effective method, it has the same shortcoming of being unable to help visually impaired people perceive anything outside a narrow "field of view". As such, they were not a proper substitute for visual navigation.

A computer-vision based solution would come closer to providing the visually impaired with sight. It could act in a similar way as a human guide who directs the blind person away from obstacles. Using computer vision, we can also find out what type of obstacle it is, whether a chair, table, shelf, et cetera, further helping navigation. Previous research has been done on navigating outdoors using computer vision and on developing CNNs that can decipher the general layout of a room from a single image, determining the shape and size of the room from the positions of the far corners [19], [20]. This project will take notes from the outcomes of previous research in similar fields but attempt to work from scratch to develop a functional device for indoors navigation with the intent of guiding a visually impaired person.

In practice, the final output device could act very much like some Self-Driving or Robot-Guidance devices, albeit calibrated to suit a human. Taking inspiration from solutions used for Self-Driving cars, it was determined that techniques for 'Free Space Detection' [21] could be applied to this use case effectively. In vehicular automation, this approach usually utilizes multiple stereo cameras and LIDAR for 3D mapping. The developed project will instead simply make use of the core concept.

A recent approach for a hardware-based solution for helping visually impaired people navigate was Sahoo el al's design of a Walking Stick Aid [22]. This solution integrated a total of eight different hardware components into a single walking stick. A Raspberry Pi and PIC microcontroller were used for control devices, and a vibration motor and buzzer were used for communicating with the user. The device integrated a discrete power supply and three sensors: Ultrasound, Water Level and GPS.

Though the device proved to be highly effective, especially when paired with a mobile app, there were a few shortcomings. The presence of so many different components on the device increases the weight, expense and risk of failure.

A hardware-based alternative to using a walking stick is to use multiple sensors mounted on the body, as evidenced in Bousbia-Salah, Bettayeb and Larbi's work in developing a navigational aid for blind people [23]. This particular system attached two ultrasonic sensors to an individual's shoulders, an accelerometer to their waist, and several more ultrasonic sensors and vibration motors throughout their body and walking stick.

This system proved quite effective. However, it is a relatively older approach, and more modern approaches are able to fulfil the same goal with fewer sensors thanks to advancements in surface modelling. It also has the same disadvantages as the first paper examined.

Both these two hardware-based approaches have similar downsides. Therefore, a software-based solution is needed.

Saleh et al's work on Outdoor Navigation for the Visually Impaired [24] is a fairly comprehensive approach based on deep learning. Here, a smartphone camera is used to get inputs, which is then segmented into different classes using the Google DeepLabV3 model. It is an effective solution that is actually suitable for use in all environments, not just outdoors.

However, the training and testing of the model was done on a dataset that used a large number of outdoors images. Therefore, it is not as efficient or accurate as it could be for interior areas. As a general computer vision-based solution for helping blind people navigate, this project is clearly effective and shows the strength of the approach.

As one of the weaknesses of image classification problems is having a context that is too broad, specialization should be carried out. It should be possible to develop a similar solution that is optimized for interior environments.

# Chapter 4

# Methodology

## 4.1. Overall Design

The project was designed with two factors in mind: first that it must be able to take a video stream input with minimal delay, and secondly it must be able to carry out semantic segmentation inference without delay, output the results and convert the results into audio for the user in real time at a pace consistent with walking.

## 4.2 App Design Requirements

It should be noted that the mobile phone, in this context, will be mounted at the user's shoulder and aimed at a slight angle downwards. This can be done using a simple system of straps and holsters, or the user can simply hold the phone up at shoulder-height. This position and angle are similar to that of a person with their gaze aimed at the floor a few steps in front of them. From this viewpoint, the bottom third of the image frame will cover the area immediately in front of the user's feet, while the middle third of the image will cover the area the user will reach with a few steps.

In order to determine the best angle for the phone, we examined the field of view of several popular phone cameras. This would allow us to determine what kind of angle the phone should be held at for maximum effectiveness.

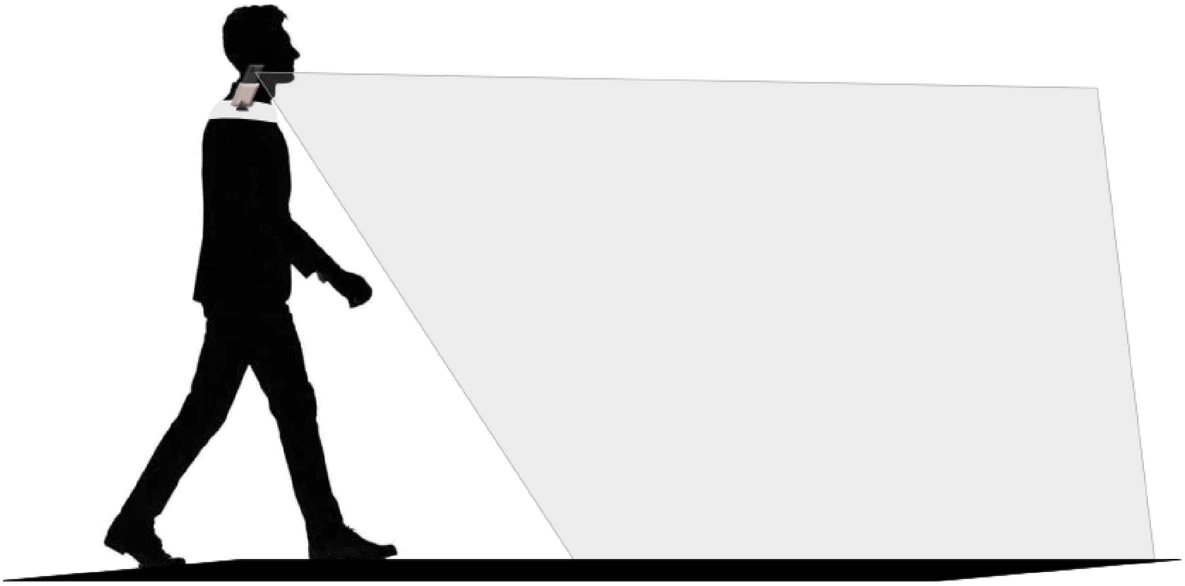


Fig. 3: Arrangement of the Phone Camera.

The field of view of a phone camera depends on the focal length of the sensor used [25]. Most phones only have a single camera. Only high-end flagship devices have multiple sensors, and those devices are not our target. A table of a selection of mobile phones, their focal length and approximate field-of-view is given here.

| Phone Model | Focal Length of Camera | Approximate Field of View |
|---|---|---|
| Samsung Galaxy J5 | 28mm | 75° |
| Samsung Galaxy A21 | 35mm | 63° |
| Xiaomi Redmi Note 4 | 30mm | 71° |
| iPhone 6S | 29mm | 74° |
| iPhone SE | 29mm | 74° |

Table 1. A Selection of Phones and their FoVs

Assuming an average adult height of 1.75 meters and an average stride length of about 0.3m, we should place the camera so that the bottom of it targets approximately 0.5m in front of the user. In order to do that, a camera held at shoulder level must be held at an angle we can calculate using the following equation:

$$angle = 90° - FoV - (\arctan\left(\frac{0.5}{1.75}\right)) \qquad (1)$$

For most phones, this gives a value in the range of 10-20° offset downwards from the horizontal. However, it will require tuning based on the phone, since the display aspect ratio as well as any pre-processing taking place in the sensor may change the field of view.

## 4.3. The Front-End

The front-end of the mobile app includes a camera preview function built using Google's Camera2 API. When the app is launched, the app requests for the camera permission. Once the permission is granted, the app can function normally. Chapter 7 and Fig. 12 provide more details on the user interface.

While the app is running, the preview image from the camera is displayed on the screen at all times. While it will not be of use to a visually impaired person, it may be useful for anyone else accompanying that person, as well as for development and debugging purposes. The front end is

The android app utilizes the Camera2 API's to call the Camera device, which starts in a background thread in order to allow the user to use the phone for other purposes while the app runs in the background. It also gets the size of the preview, which is recorded as it must be passed to the segmentation algorithm later. A TextureView object is used to display the preview on the screen. This preview

Once the camera has been successfully loaded, the bitmap representation of the current frame of the preview is extracted and passed to the segmentation algorithm. It should be noted that the preview frame is passed at a rate of 1 per second. This rate can be increased or decreased as needed. By passing it at this rate, we also get an output at 1 FPS.

## 4.4. Semantic Segmentation and Models

In Semantic Segmentation, each pixel of an image is assigned a colour that labels it as belonging to a particular class. In order to do this, a model breaks down the features and extracts the general contour of a 'segment'. Keeping in mind that shallow networks extract high level features and 'broad strokes' and deep models allow the algorithm to extract fine lines.

In normal convolutional networks, the output is usually much smaller than the input. In a classification problem, the input may include several dimensions and multiple classes, which is then output as a single class.

For segmentation, the output must label individual pixels. If it was carried out in a similar way to a normal convolutional network, the output image would be far too small to use properly. Therefore, semantic segmentation models use an 'Encoder-Decoder' design. The encoder consists of a CNN as described above. The decoder takes the output from the encoder and upscales it back to the original resolution, allowing the segmentations to be mapped accurately to the real image.

A block diagram of the process is given here in Fig. 4:

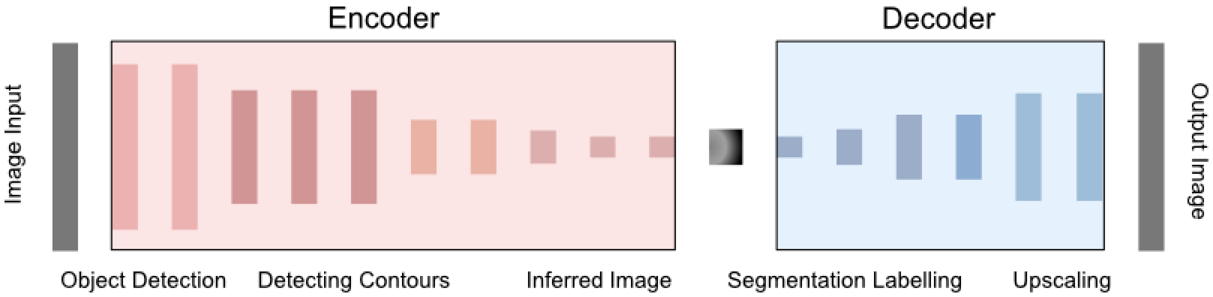


Fig. 4: Block Diagram of Image Segmentation

The primary goal of the algorithm is to detect 'walkable space' versus 'non-walkable space', so semantic segmentation is the most suitable. There are many architectures for semantic segmentation available, such as R-CNN, Fast-CNN, FCN, etc. Each architecture has its advantages and disadvantages in regards to processing speed and accuracy.

For example, R-CNN [26] is relatively inaccurate compared to newer models, achieving only about 53% accuracy in datasets such as PASCAL VOC. Additionally, it is relatively slow, taking more than 10 seconds per inference. Fast-CNN is a faster version of R-CNN [27], cutting down the inference time to about 0.5 to 2 seconds per image. However, that is still relatively slow for the purposes of real-time segmentation.

Two models were chosen for further work: Shufflenet and DeepLabV3

### 4.4.1. ShuffleNetv2 Architecture

Shufflenet is a CNN architecture that was developed from the ground-up to be extremely efficient using limited computing resources [28].

One of the goals of the project was to minimize hardware costs. While the assumption can be made that even underprivileged people can afford a basic, low-end smartphone, it cannot be assumed that the device will have a lot of power. Therefore, ShuffleNet sacrifices accuracy for processing speed and limited resource use.

In ShuffleNet, group convolutions are utilized. In grouped convolutions, filters are separated into several different groups. Each group applies filters in a parallel manner, and at the end the final output layer is composed by combining the outputs of each group. Grouped convolutions were first proposed in AlexNet [29] and has been further developed since then.

Most of the computation cost is minimized by taking advantage of grouped convolutions. This method has one weakness – some information can get exaggerated or marginalized due to how grouped convolutions work. Therefore, a 'Channel Shuffle' function is carried out which flattens the output layers and minimizes the effects of information loss.

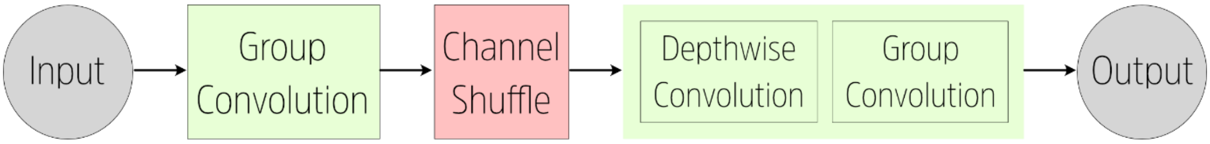


Fig. 5: ShuffleNet Architecture

### 4.4.2. DeepLabV3 Architecture

Developed by Google, DeepLabV3 is one of the most accurate and powerful architectures for semantic segmentation available today [30]. In a sense, it is the opposite of ShuffleNet, being a far heavier model. The DeepLabv3 architecture makes use of Atrous Convolutions (also called dilated convolutions). In an Atrous convolution, holes are placed between each unit or pixel of the filter. So, each convolution will cover a wider area, but the filter size will remain the same.

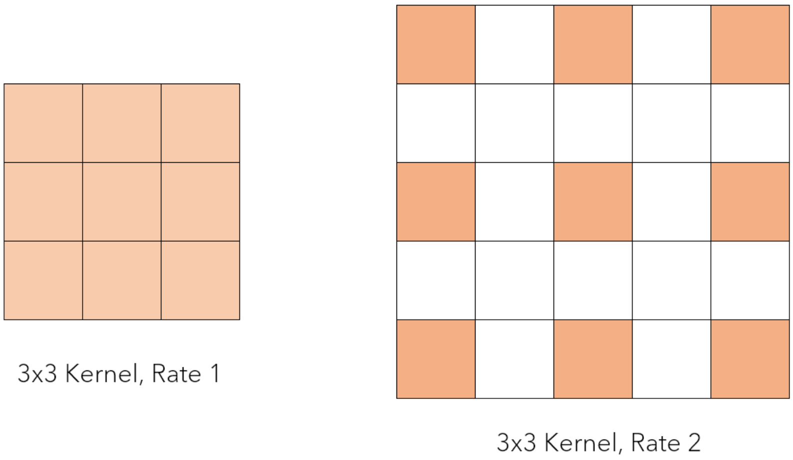


Fig. 6: Atrous Convolutions

So, a 3x3 filter with a Dilation Rate of 2 will cover the same area as a 5x5 filter. The second and fourth positions for the rows and columns will be discarded. In DeepLabv3, a spatial pyramid is used, using multiple Atrous filters with increasing Dilation Rate in succession.

The full architecture of a DeepLabv3 model is given below.

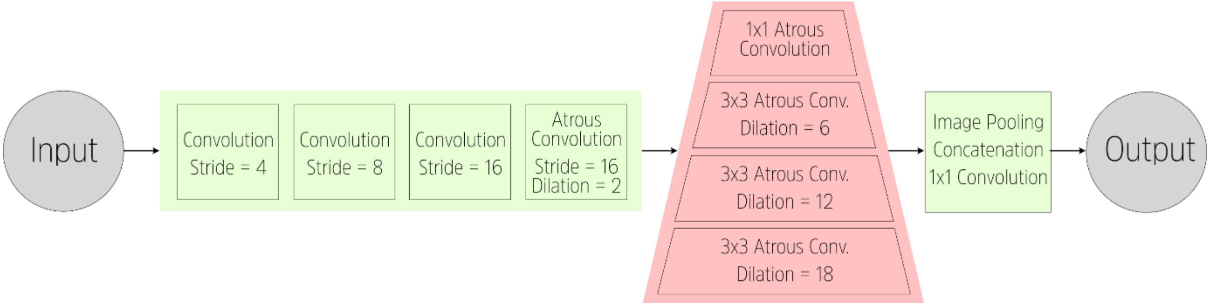

Input
Convolution Stride = 4
Convolution Stride = 8
Convolution Stride = 16
Atrous Convolution Stride = 16 Dilation = 2
1x1 Atrous Convolution
3x3 Atrous Conv. Dilation = 6
3x3 Atrous Conv. Dilation = 12
3x3 Atrous Conv. Dilation = 18
Image Pooling Concatenation 1x1 Convolution
Output

Fig. 7: Deeplabv3 Architecture with Atrous Spatial Pyramid and Pooling

The four initial convolution layers and the Atrous spatial pyramid make up the encoder. The last layer represented in the diagram, with Concatenation and the 1x1 convolution, make up our decoder.

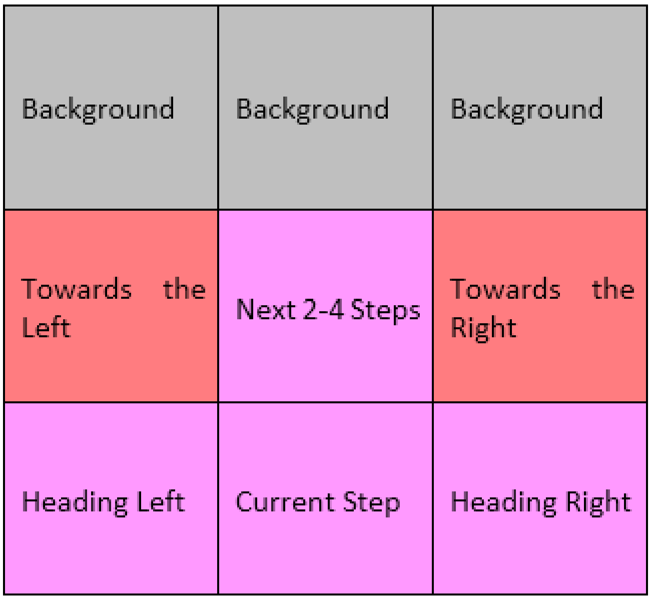
## 4.5. Generating Audio Output

Once a segmented image has been received, the final task is to generate an audio output. While it would be possible to detect the exact location of an obstacle in relation to the user, there is no efficient way to convey this information through audio. Describing the position of multiple objects around the user would take far too long in terms of speech. Additionally, it is mostly unnecessary in the context of navigation.

The output segmented image will be divided into a grid of 9 parts. This grid is visualized in Fig. 8. For guiding a user, the most important part is where their next few steps will be. We can divide the image into a foreground and background section. Since we know the approximate height, angle and field of view of the image, there is no particular need to worry about distances.

The presence of free walking space or clutter is determined by the colour of the segmented image in that particular section of the grid. Our primary architectures, ShuffleNet and DeepLab, return purple and brown coloured overlays for the floor respectively.

If purple is used to denote the floor in the segmented image, then the function will check for the prevalence of purple in the ‘Current Step’ position of the grid. This current step is the most important position in our grid, since it is where the user is expected to step next. As the camera is fixed in position on the user’s body, the user can simply turn around and change direction without having to worry about the camera.



| | | |
|---|---|---|
| Background | Background | Background |
| Towards the Left | Next 2-4 Steps | Towards the Right |
| Heading Left | Current Step | Heading Right |

Fig. 8: Grid Organization for processing the segmented output

If another object appears in the ‘Current Step’ position, then the segmented output will return a different colour. When the output function checks the colour, it will send a call to the text-to-speech function to tell the user “Obstacle in front!” This message will have the highest priority and override any other messages.

Additional warnings can also be built into the app in a similar way. If an obstacle is detected in ‘Heading Left’ or ‘Heading Right’, the system can tell the user “Obstacle to left/right!” If an obstacle is detected in the “Next 2-4 Steps”, the system will tell the user “Obstacle up ahead.” The current system only has the basic “Obstacle in Front” implemented in order to avoid overloading a user with information.

The app remains active in the background, instructing the user even when the user has switched to the home screen or another app.

# Chapter 5

# Experimental Details

## 5.1. Dataset

The primary dataset used in this project is the MIT ADE20k Dataset for Scene Segmentation [31], [32]. This dataset features 20,120 images taken from a wide variety of scenes both outdoors and indoors, followed by an additional 2000 images for validation. This dataset consists purely of RGB images rather than RGB-D, making it suitable for our purposes. It is also extremely richly annotated.

This dataset is an improvement on older datasets like CityScapes and Pascal VOC. The Cityscapes dataset is primarily an outdoors dataset, but it is also capable of predicting indoor environments with a certain level of accuracy. For instance, in interior environments, models trained on the CityScapes dataset recognize the floor as 'road', giving us relatively similar results.

As there is a high incidence of indoor images, this dataset is better suited for the context of this research than other popular image segmentation datasets such as Cityscapes or PASCAL VOC.

An example of several images from the ADE20k dataset is given below. All three images were taken from the official ADE20k Scene Parsing browser.

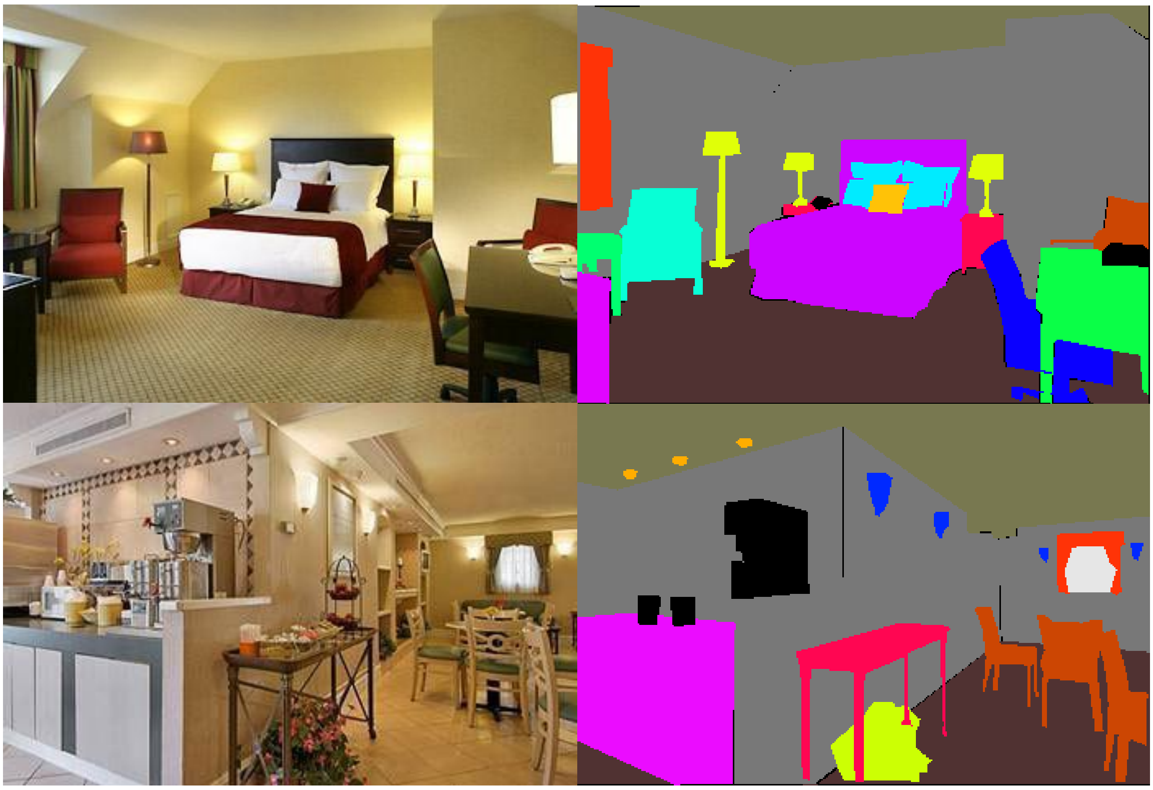


Fig. 9: Examples of ADE20k Training Images

## 5.2. Preprocessing

The ADE20k Dataset features a total of 150 Classes. However, most of these classes are either superfluous, or too finely detailed for the task at hand. We do not need to distinguish, for example, classes such as ‘book’, ‘bag’, ‘ball’, ‘clock’, etc, since this is not an object classification problem but rather navigation and detection of obstacles. Therefore, the class labels were consolidated into the primary classes that will be applicable to the process of interior navigation.

In order to consolidate it, we considered which objects can be classed as the ground or floor – the walkable surface; the wall – the bounds of the room or environment; stairs – the furniture – this represents the range of obstacles that will be an impediment to navigation in indoor environments; stairs – a lot of blind people are extremely insecure about stairs; and finally, everything else, which gets classed as other. If anything under ‘other’ is large enough to impede movement, it will be caught in the same way as furniture.

The consolidated class labels are given below:

**1 (Wall)** ← 9 (window), 15 (door), 33 (fence), 43 (pillar), 44 (sign board), 145 (bulletin board)

**4 (Floor)** ← 7 (road), 14 (ground, 30 (field), 53 (path), 55 (runway))

**8 (Furniture)** ← 8 (bed), 11 (cabinet), 14 (sofa), 16 (table), 19 (curtain), 20 (chair), 25 (shelf), 34 (desk)

**7 (Stairs)** ← 54 (stairs)

**26 (Others)** ← 18 (plant), class numbers larger than 26

Unlike classification problems, there is relatively little need for carrying out image augmentation or other methods. This is because segmentation algorithms rely on extracting contours and pixel-level features. Most image augmentations would cause the file annotation to mismatch, leading to more inaccurate results.

## 5.3. Training

The original plan was to separate the ADE20k dataset into indoor and outdoor spaces, using the list of scene labels provided in the dataset. Additionally, we would collect a number of pictures of various locations in Bangladesh, such as shopping centers, stores, restaurants, apartments, classrooms, etc. and label them ourselves. This would allow us to expand the dataset specifically as well as enable it to recognize Bangladeshi interior environments more accurately. However, due to circumstances, this was not possible. Additionally, we also weren’t able to train a DeepLab v3 model, which is computationally intensive and requires multiple GPUs, from scratch.

The Deeplab v3 Model used was pretrained on ADE20k using a ResNet50 model as the backbone for the encoder. The learning rate of both the encoder and decoder was set at 0.02.

# Chapter 6

# Results

## 6.1. Chapter Focus

This chapter will focus on the results and the final output application, discussing the effectiveness of the implementation and how usable the final app is.

While the app carries out its intended job fairly well, there are some issues regarding the usability of the app. Firstly, it is may not be possible to launch the app from the home screen on all phones, since some older operating systems do not have voice recognition capability. Additionally, the app requires permissions, which must be granted using a physical tap on the screen. Other than these issues, the actual functionality of the app is smooth and nominally effective.

## 6.2. The App's Output

While the bulk of our work focused on attempting to develop a DeepLabv3 model, we implemented a Shufflenetv2 model trained on the Cityscapes dataset as a backup. This model is not as accurate in indoor environments as a DeepLab v3 model. An example of an inference follows:

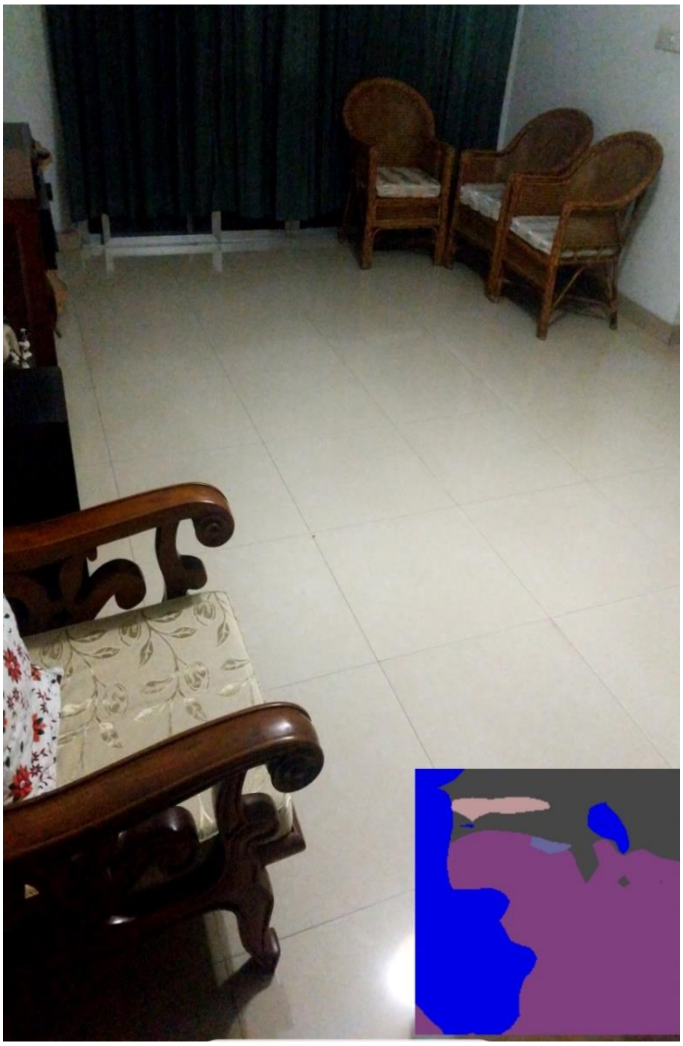


Fig. 10: Inference using ShuffleNet v2

With this model, the furniture and other obstacles are coloured in blue, while the floor is coloured purple.

As apparent from the preview of the segmented image, the segmented output image does not clearly distinguish the various contours of the furniture both in the foreground and the background.

However, the accuracy of the current app is sufficient for most interior environments. As ShuffleNetv2 is an extremely lightweight model, inference is almost instant even on low-end devices.

One issue with the current system is that the input is passed into the segmentation algorithm at a constant rate. This does not stop when the person is moving and the image preview gets blurred. This blurry image is still sent to the segmentation algorithm. When the input image is blurry, the segmentation results are not returned properly and the audio out will tell the user, more often than not, that there is an obstacle ahead. Therefore, the user must either walk in sync with the program or make minute pauses after each step-in order to ensure that the image is not blurred when the program makes an inference.

# Chapter 7

# Software Design

## 7.1. Service Diagram

This section features a service showing the overall process of a user opening and using the app. Most of the details of how the app functions have been covered in previous sections. Following the service diagram, an overview of the front-end UI of the app is also given.

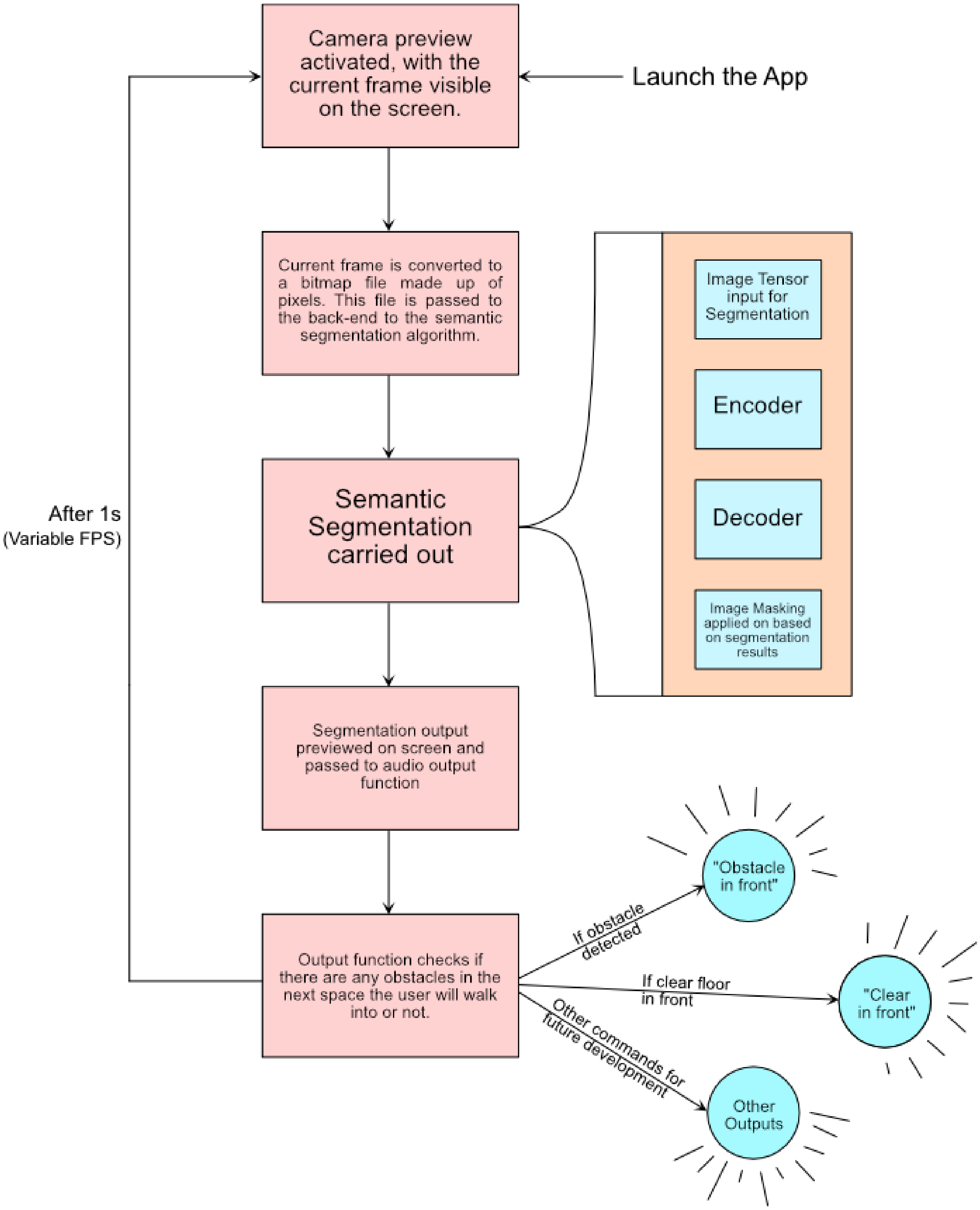
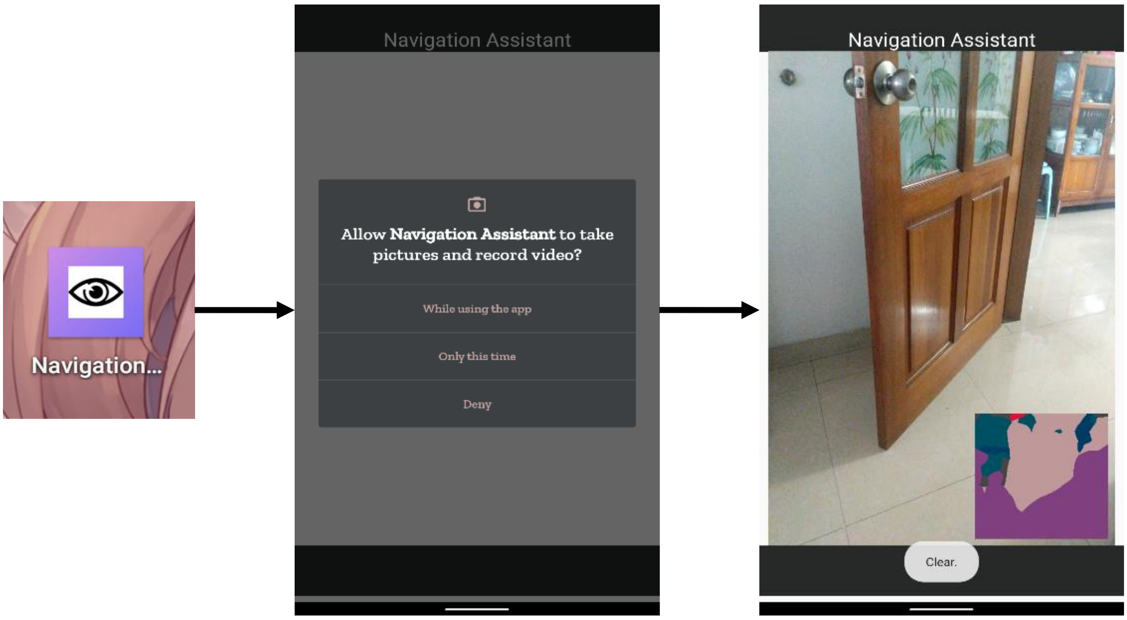

Launch the App
Camera preview activated, with the current frame visible on the screen.
Current frame is converted to a bitmap file made up of pixels. This file is passed to the back-end to the semantic segmentation algorithm.
Semantic Segmentation carried out
Image Tensor input for Segmentation
Encoder
Decoder
Image Masking applied on based on segmentation results
Segmentation output previewed on screen and passed to audio output function
Output function checks if there are any obstacles in the next space the user will walk into or not.
After 1s (Variable FPS)
If obstacle detected
"Obstacle in front"
If clear floor in front
"Clear in front"
Other commands for future development
Other Outputs

Fig. 11: Service Diagram

## 7.2. User Interface

Despite being aimed at the visually impaired, the app was developed with a concrete, if simple, front-end user interface in mind. The goal was to design an interface that would be easy to use for debugging and testing, during development. Once released, the UI could be used by sighted people near the visually impaired user to double-check if the app is giving results correctly, as there is a small chance the output of the inference will give poor results in certain environments.



Allow **Navigation Assistant** to take pictures and record video?

Fig. 12: User Interface

# Chapter 8
# Working Sheets

## 8.1. Chapter Focus

This chapter is focused on a business perspective of the app and how it can be developed further in an entrepreneurial direction.

Compared to most material goods, software products have very different approaches when it comes to commercializing or spreading them out. A software product such as an app has essentially no manufacturing cost, since the files can be copied indefinitely. Additionally, since this app is fully offline and all the required information or data is stored locally, there are no additional server costs.

Most of the costs are for R&D, such as the cost of manpower and of computing resources. In order to cover the costs of further research and development, we propose the following monetary plans.

## 8.2. Development Timeline

From the start to the end, the project was developed over a period of approximately 4 months. Prior to that, the majority of the background research work was carried out.

In addition to the work done so far, two months' proposed 'future work' has been listed. Chapter 10.1 details the technical side of the future scope of development, while the monetary or business side of future development is given here.

The approximate timeline is:

| Month | Developmental Stage |
|---|---|
| Month 1 | App Design, Conceptualization, Finalizing tool selection |
| Month 2 | Experimentation with Segmentation Models |
| Month 3 | Implementation of segmentation model into the app |
| Month 4 | Testing the app, finalizing thesis and documentation |
| **Future Work** | |
| Month 5 | User survey from the visually impaired, further refinement as needed. |
| Month 6 | Product release |

Table 2: Development Timeline

## 8.3. Monetization Plans

There are several approaches we can take to monetizing the app. The first step would be to place it on an App Store – more specifically, the Google Play Store, since it is an android app and does not support iOS devices. However, it must be noted that the app is still in a relatively underdeveloped state and it would not be suitable for distribution until further work has been done.

### 8.3.1. Plan 1: Ad Revenue

If we expect a significant user base, then adding advertisements for the increased revenue is a good option. Based on the current user interface of the app, there is space for advertising at the top and the bottom of the screen. Using the number of blind people in Bangladesh and the smartphone penetration rate, we can estimate a theoretical maximum of approximately 500,000 users of the app in Bangladesh alone. While the actual numbers will only be a fraction of this, even 5% of the blind, mobile-owning population is a major success.

If we account for global users, the number jumps massively. Additionally, with a global audience there is less of a need for making the app suitable for low-end device, allowing more freedom in development.

However, getting significant numbers from ad revenue would require some initial investment into marketing. Additionally, some users are not fond of apps that have ads, which may reduce the userbase a bit.

### 8.3.2. Plan 2: Premium Model

A premium model would be more challenging to implement, especially in Bangladesh. As users need to pay a certain amount of money upfront, it would discourage many users from trying out the app in the first place. However, it would be the fastest way to generate revenue.

An alternate to the premium model would be a freemium model, which would implement advertisements but give users the option to remove them by paying a subscription or one-time fee. Many freemium apps also lock app features behind a paywall, but since this app is designed to help disabled people improve their quality of life, we feel that such an approach would be unethical.

## 8.4. Summary

As this is an app to improve the quality of life for millions all over the world, it should remain free of egregious monetization. If it becomes necessary to monetize the app, we would focus on advertisement revenue as the primary source of revenue. Google monetizes apps based on both downloads and in-app advertisements, both of which have significant potential given the potential userbase of the app.

While there are a number of competing apps out there, we believe our approach has the potential to beat them in accuracy and features once further work has been done as highlighted in section 10.1.

# Chapter 9

# Impact

## 9.1. Social Impact

The social impact of an app designed to help visually impaired people navigate is obvious. In addition to improving the quality of life for them, it would also help them act more independently. Family members would be freed from part of the burden of helping their blind relatives move around, giving them additional time to be productive.

While this app will not remove every single issue that blind people have, it can work as a part of a suite of apps that makes life easier for them.

Generally, we expect to see an increase in the productivity and standard of living in many places where there is a prevalence of blind people.

## 9.2. Environmental Impact

One of our original goals was to design this project by using as minimal hardware as possible. That means no special sensors or secondary boards. The input, processing and output are all handled by a single device that everyone already owns – a mobile phone.

This means that this project has minimal environmental impact. No unused or old hardware needs to be manufactured or dumped. The shoulder mounts for the phone can be made of biodegradable materials such as cardboard or paper.

# Chapter 10
# Project Summary

## 10.1. Future Scope

The project in its current state is quite minimal and was only just able to accomplish its proposed goals. So, there is a significant amount of future scope to this project.

Firstly, we can work on improving the segmentation models. With access to better hardware, we would be able to train a specialized DeepLab v3 model using TensorFlow lite that runs acceptably fast on mobile devices. If the heavier model is not fast enough, lighter alternative models such as ShuffleNet or Faster-RCNN can be used.

Secondly, the system could be modified in the future to provide more details. For instance, a function could be built where the user passes a voice command to the algorithm asking them to describe the room. In that case, the algorithm could use the segmentation output grid to determine the location of various objects or obstacles in relation to the user. It could say "There is a chair in front towards the left." (see Fig. 8).

A long-term priority will be to increase the speed & accuracy of the inference (generating outputs from images) process on our proposed system. The simplest way to accomplish this is to delve into implementing better state-of-the-art neural network models that are capable of delivering more accurate inferences on a wider range of inputs. The usage of these such neural networks though comes with a cost of taking up more space on the user's smartphone. Large multi-layered networks can often take up to 500 Megabytes on the user's smartphone, which is significant on many low-end devices.

As a resolution to this problem, we can set up the smartphone's camera to stream data to a server running our algorithms. This server will include Graphics Processing Units (GPU's) to further boost the time and accuracy of our inferences and increase the frame rate of our system while freeing up CPU cycles on the smartphone for other processes. With the help of ever advancing mobile internet technology, this can be done seamlessly and without taking up much of the resources of the user's smartphone. This would allow us to create an app that functions offline but is more accurate online.

When evaluating technical feasibility and performance, there is a lack of standard computer vision benchmark datasets out in the open. Relatively small datasets can be collected from open-sourced resources, making it impossible to compare the performance between various solutions. It may have been caused by the lack of open computer vision datasets tailored to developing solutions for the visually impaired. Generic benchmark datasets, such as ImageNet (classification) and COCO (object detection) could potentially be used to some extent. However, these datasets lack objects of vital importance to the visually impaired, for example, corridors, stairs, elevators.

A collaborative effort for creating a high-quality benchmark dataset for developing and evaluating computer vision solutions for the visually impaired is required. User-based evaluation presents a common challenge in the academic community. The involvement of visually impaired individuals to research projects is challenging and often suffer from selection bias. Moreover, lack of standardized evaluation methods and potential reporting bias limit the representativity of these experiments. To aid the problem in hand, we will be looking to create a custom dataset of our own with the help of visually impaired people by capturing annotated images of the visually impaired people. With this, we are expecting to acquire a healthy number of samples for training while focusing on objects like stairs, corridors and all other necessary elements that conventional datasets often leave out.  We hope to gain valuable insight into specific requirements for subjects who have been coping with visual dysfunction for a long period of time, that might not be obvious from data collected on normal sighted subjects who have had simulated vision loss for only a brief period.

Lastly, we would like to address the development of tactile or acoustic interfaces to provide better feedback to visually impaired people when it comes to alert them regarding obstacles on their path. The app we have implemented performs localization only, but we hope to transform it into a full-featured wayfinding app with an accessible UI that offers turn-by-turn directions to a desired destination as well as optional announcements of nearby points of interest and also provide haptic feedback as an alert when coming across a hazard or caution.

Some visually impaired travellers might prefer navigation directions presented using spatialized (3D) sound, as implemented in the Microsoft Soundscape App, and we will experiment with this type of interface as a possible alternative (or supplement) to verbal directions. We note that travellers with residual vision may prefer a visual UI (e.g., an augmented reality interface that superimposes high-contrast arrows on the smartphone screen to guide the user) over an audio one. An extensive evaluation/refinement process is to be carried out with the help of blind users, aimed at improving system performance and usefulness of the system.

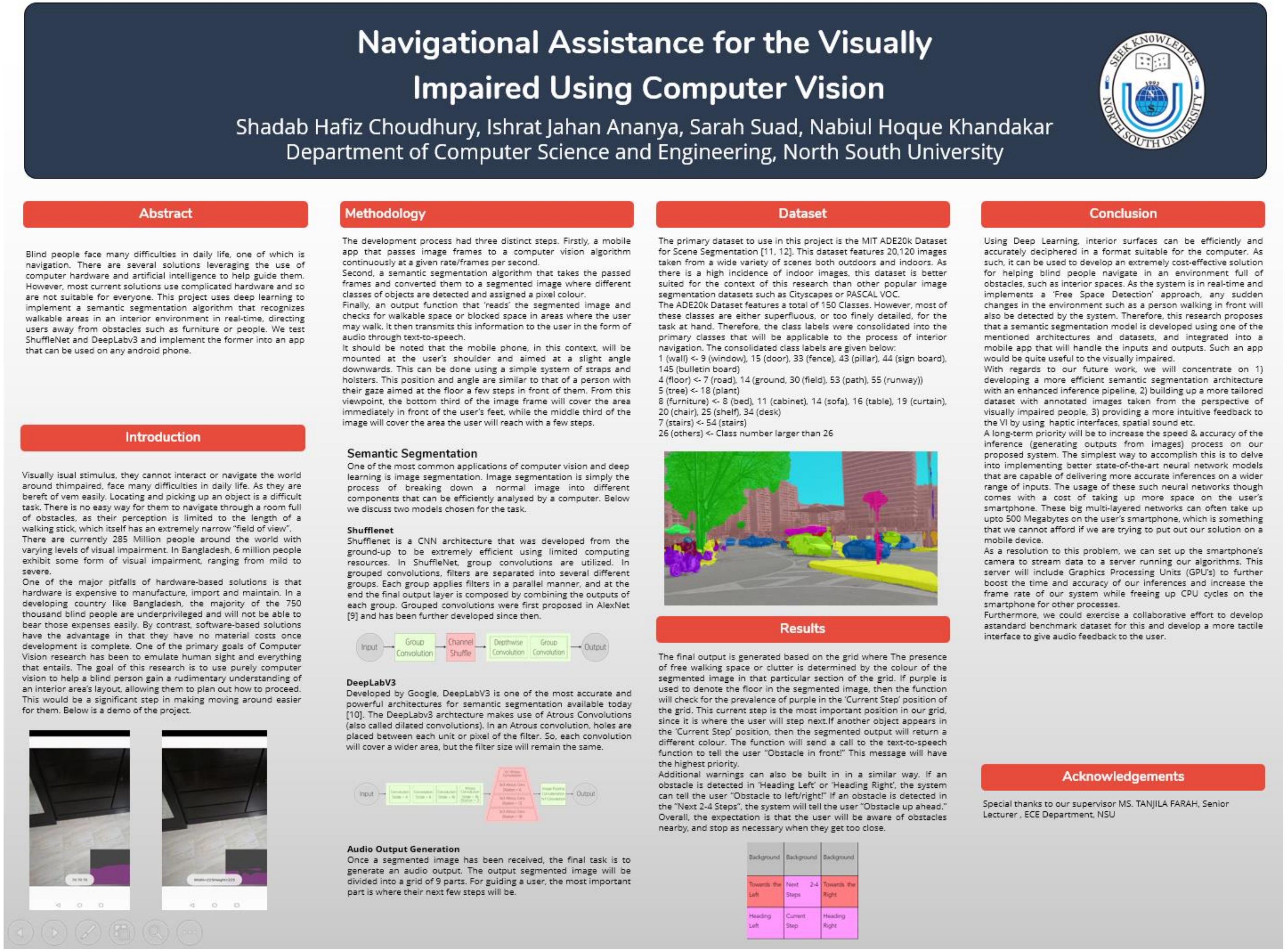
## 10.2. Conclusion

Using Deep Learning and semantic segmentation, interior surfaces can be efficiently and accurately deciphered in a format suitable for the computer. As such, it can be used to develop an extremely cost-effective solution for helping blind people navigate in an environment full of obstacles, such as interior spaces. As the system is in real-time and always checks for free space in front rather than mapping out the room, any sudden changes in the environment such as a person walking in front will also be detected by the system.

Therefore, this research proposes that a semantic segmentation model is developed using one of the mentioned architectures and datasets, and integrated into a mobile app that will handle the inputs and outputs. Such an app would be quite useful to the visually impaired, allowing them to navigate with greater ease even in unfamiliar environments.

## 10.3. Poster

A display of the poster is given here.



# Navigational Assistance for the Visually Impaired Using Computer Vision

Shadab Hafiz Choudhury, Ishrat Jahan Ananya, Sarah Suad, Nabiul Hoque Khandakar
Department of Computer Science and Engineering, North South University

SEEK KNOWLEDGE
NORTH SOUTH UNIVERSITY
1992

## Abstract

Blind people face many difficulties in daily life, one of which is navigation. There are several solutions leveraging the use of computer hardware and artificial intelligence to help guide them. However, most current solutions use complicated hardware and so are not suitable for everyone. This project uses deep learning to implement a semantic segmentation algorithm that recognizes walkable areas in an interior environment in real-time, directing users away from obstacles such as furniture or people. We test ShuffleNet and DeepLabv3 and implement the former into an app that can be used on any android phone.

## Introduction

Visually isual stimulus, they cannot interact or navigate the world around thimpaired, face many difficulties in daily life. As they are bereft of vem easily. Locating and picking up an object is a difficult task. There is no easy way for them to navigate through a room full of obstacles, as their perception is limited to the length of a walking stick, which itself has an extremely narrow "field of view".

There are currently 285 Million people around the world with varying levels of visual impairment. In Bangladesh, 6 million people exhibit some form of visual impairment, ranging from mild to severe.

One of the major pitfalls of hardware-based solutions is that hardware is expensive to manufacture, import and maintain. In a developing country like Bangladesh, the majority of the 750 thousand blind people are underprivileged and will not be able to bear those expenses easily. By contrast, software-based solutions have the advantage in that they have no material costs once development is complete. One of the primary goals of Computer Vision research has been to emulate human sight and everything that entails. The goal of this research is to use purely computer vision to help a blind person gain a rudimentary understanding of an interior area's layout, allowing them to plan out how to proceed. This would be a significant step in making moving around easier for them. Below is a demo of the project.

## Methodology

The development process had three distinct steps. Firstly, a mobile app that passes image frames to a computer vision algorithm continuously at a given rate/frames per second.

Second, a semantic segmentation algorithm that takes the passed frames and converted them to a segmented image where different classes of objects are detected and assigned a pixel colour.

Finally, an output function that 'reads' the segmented image and checks for walkable space or blocked space in areas where the user may walk. It then transmits this information to the user in the form of audio through text-to-speech.

It should be noted that the mobile phone, in this context, will be mounted at the user's shoulder and aimed at a slight angle downwards. This can be done using a simple system of straps and holsters. This position and angle are similar to that of a person with their gaze aimed at the floor a few steps in front of them. From this viewpoint, the bottom third of the image frame will cover the area immediately in front of the user's feet, while the middle third of the image will cover the area the user will reach with a few steps.

### Semantic Segmentation

One of the most common applications of computer vision and deep learning is image segmentation. Image segmentation is simply the process of breaking down a normal image into different components that can be efficiently analysed by a computer. Below we discuss two models chosen for the task.

**Shufflenet**

Shufflenet is a CNN architecture that was developed from the ground-up to be extremely efficient using limited computing resources. In ShuffleNet, group convolutions are utilized. In grouped convolutions, filters are separated into several different groups. Each group applies filters in a parallel manner, and at the end the final output layer is composed by combining the outputs of each group. Grouped convolutions were first proposed in AlexNet [9] and has been further developed since then.

**DeepLabV3**

Developed by Google, DeepLabV3 is one of the most accurate and powerful architectures for semantic segmentation available today [10]. The DeepLabv3 archtecture makes use of Atrous Convolutions (also called dilated convolutions). In an Atrous convolution, holes are placed between each unit or pixel of the filter. So, each convolution will cover a wider area, but the filter size will remain the same.

**Audio Output Generation**

Once a segmented image has been received, the final task is to generate an audio output. The output segmented image will be divided into a grid of 9 parts. For guiding a user, the most important part is where their next few steps will be.

## Dataset

The primary dataset to use in this project is the MIT ADE20k Dataset for Scene Segmentation [11, 12]. This dataset features 20,120 images taken from a wide variety of scenes both outdoors and indoors. As there is a high incidence of indoor images, this dataset is better suited for the context of this research than other popular image segmentation datasets such as Cityscapes or PASCAL VOC.

The ADE20k Dataset features a total of 150 Classes. However, most of these classes are either superfluous, or too finely detailed, for the task at hand. Therefore, the class labels were consolidated into the primary classes that will be applicable to the process of interior navigation. The consolidated class labels are given below:

1 (wall) <- 9 (window), 15 (door), 33 (fence), 43 (pillar), 44 (sign board), 145 (bulletin board)
4 (floor) <- 7 (road), 14 (ground, 30 (field), 53 (path), 55 (runway))
5 (tree) <- 18 (plant)
8 (furniture) <- 8 (bed), 11 (cabinet), 14 (sofa), 16 (table), 19 (curtain), 20 (chair), 25 (shelf), 34 (desk)
7 (stairs) <- 54 (stairs)
26 (others) <- Class number larger than 26

## Results

The final output is generated based on the grid where The presence of free walking space or clutter is determined by the colour of the segmented image in that particular section of the grid. If purple is used to denote the floor in the segmented image, then the function will check for the prevalence of purple in the 'Current Step' position of the grid. This current step is the most important position in our grid, since it is where the user will step next.If another object appears in the 'Current Step' position, then the segmented output will return a different colour. The function will send a call to the text-to-speech function to tell the user "Obstacle in front!" This message will have the highest priority.

Additional warnings can also be built in in a similar way. If an obstacle is detected in 'Heading Left' or 'Heading Right', the system can tell the user "Obstacle to left/right!" If an obstacle is detected in the "Next 2-4 Steps", the system will tell the user "Obstacle up ahead." Overall, the expectation is that the user will be aware of obstacles nearby, and stop as necessary when they get too close.

| | | |
|---|---|---|
| Background | Background | Background |
| Towards the Left | Next 2-4 Steps | Towards the Right |
| Heading Left | Current Step | Heading Right |

## Conclusion

Using Deep Learning, interior surfaces can be efficiently and accurately deciphered in a format suitable for the computer. As such, it can be used to develop an extremely cost-effective solution for helping blind people navigate in an environment full of obstacles, such as interior spaces. As the system is in real-time and implements a 'Free Space Detection' approach, any sudden changes in the environment such as a person walking in front will also be detected by the system. Therefore, this research proposes that a semantic segmentation model is developed using one of the mentioned architectures and datasets, and integrated into a mobile app that will handle the inputs and outputs. Such an app would be quite useful to the visually impaired.

With regards to our future work, we will concentrate on 1) developing a more efficient semantic segmentation architecture with an enhanced inference pipeline, 2) building up a more tailored dataset with annotated images taken from the perspective of visually impaired people, 3) providing a more intuitive feedback to the VI by using haptic interfaces, spatial sound etc.

A long-term priority will be to increase the speed & accuracy of the inference (generating outputs from images) process on our proposed system. The simplest way to accomplish this is to delve into implementing better state-of-the-art neural network models that are capable of delivering more accurate inferences on a wider range of inputs. The usage of these such neural networks though comes with a cost of taking up more space on the user's smartphone. These big multi-layered networks can often take up upto 500 Megabytes on the user's smartphone, which is something that we cannot afford if we are trying to put out our solution on a mobile device.

As a resolution to this problem, we can set up the smartphone's camera to stream data to a server running our algorithms. This server will include Graphics Processing Units (GPU's) to further boost the time and accuracy of our inferences and increase the frame rate of our system while freeing up CPU cycles on the smartphone for other processes.

Furthermore, we could exercise a collaborative effort to develop astandard benchmark dataset for this and develop a more tactile interface to give audio feedback to the user.

## Acknowledgements

Special thanks to our supervisor MS. TANJILA FARAH, Senior Lecturer , ECE Department, NSU

# Chapter 11

# Bibliography

## 11. References

1. Sutradhar, P. Gayen, M. Hasan, R. D. Gupta, T. Roy and M. Sarker, "Eye diseases: the neglected health condition among urban slum population of Dhaka, Bangladesh", *BMC Opthalmol*, vol. 19, no. 1, pp. 38, Jan. 2019.
2. Staff Correspondent, "Six million Bangladeshis visually impaired: Speaker." thedailystar.net, Accessed Oct. 24, 2020. [Online] Available: https://www.thedailystar.net/city/news/six-million-bangladeshis-visually-impaired-speakers-1856473
3. Global System for Mobile Communications, Bangladesh: Mobile industry driving growth and enabling digital inclusion, Global System for Mobile Communications, London, UK, Apr. 9, 2018. Accessed Sept. 21, 2020. [Online]. Available: https://www.gsma.com/mobilefordevelopment/resources/bangladesh-mobile-industry-driving-growth-and-enabling-digital-inclusion/.
4. A.G. Ivakhnenko and V. G. Lapa, *Cybernetics and forecasting techniques, (Modern analytic and computational methods in science and mathematics),* 8th ed. New York, NY, USA: American Elsevier Publishing Co., 1967
5. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
6. U. Neisser. *Cognition and reality: Principles and implications of cognitive psychology*. New York, NY, USA: W H Freeman, 1976.
7. B. Bascle, A. Blake, A. Zisserman. “Motion deblurring and super-resolution from an image sequence,” in *Computer Vision - 4th European Conference on Computer Vision Cambridge*, UK. B. Buxton, R. Cipolla, Eds in *Lecture Notes in Computer Science*, vol 1065. Springer, Berlin, Heidelberg, Apr. 2005, pp 571-582, doi:10.1007/3-540-61123-1_171.
8. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," in *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 1, pp. 44-58, Jan. 2006, doi: 10.1109/TPAMI.2006.21.
9. Ulku and E. Akagunduz, “A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D images”, 2020, arXiv:1912.10230.
10. X. Liu, Z. Deng, and Y. Yang, “Recent progress in semantic image segmentation,” *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1089-1106, Aug. 2019, doi: 10.1007/s10462-018-9641-3.
11. S. A. Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad, and G. Hamarneh, “Deep semantic segmentation of natural and medical images: a review,” *Artificial Intelligence Review*, vol. 54, no. 1, pp. 137-178, June, 2020 doi: 10.1007/s10462-020-09854-1.

12. M. Siam, S. Elkerdawy, M. Jagersand and S. Yogamani, "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, 2017, pp. 1-8, doi: 10.1109/ITSC.2017.8317714.

13. D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld and H. M. Gross, "Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis," 2020, arXiv:2011.06961.

14. R. Ambrus, S. Claici, and A. Wendt, "Automatic Room Segmentation from Unstructured 3-D Data of Indoor Environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 749-756, Apr. 2017, doi: 10.1109/LRA.2017.2651939.

15. E. Brady, M. R. Morris, Y. Zhong, S. White, and J. P. Bigham, "Visual challenges in the everyday lives of blind people," in C*HI '13: Proc. SIGCHI Conf. on Human Factors in Computing Systems*, April 2013, pp 2117-2126, doi: 10.1145/2470654.2481291.

16. C. Liambas and M. Saratzidis, "Autonomous OCR dictating system for blind people," *2016 IEEE Global Humanitarian Technology Conference (GHTC),* Seattle, WA, 2016, pp. 172-179, doi: 10.1109/GHTC.2016.7857276.

17. M. A. Khan Shishir, S. Rashid Fahim, F. M. Habib and T. Farah, "Eye Assistant: Using mobile application to help the visually impaired," *2019 1st International Conf. on Advances in Science, Engineering and Robotics Technology (ICASERT),* Dhaka, Bangladesh, 2019, pp. 1-4, doi: 10.1109/ICASERT.2019.8934448.

18. M. Martinez, K. Yang, A. Constantinescu, and R. Stiefelhagen, "Helping the Blind to Get through COVID-19: Social Distancing Assistant Using Real-Time Semantic Segmentation on RGB-D Video," *Sensors*, vol. 20, no. 18, pp. 5202, Sep. 2020, doi: 10.3390/s20185202.

19. V. Hedau, D. Hoiem and D. Forsyth, "Recovering the spatial layout of cluttered rooms," *2009 IEEE 12th International Conference on Computer Vision, Kyoto,* 2009, pp. 1849-1856, doi: 10.1109/ICCV.2009.5459411.

20. C. Lee, V. Badrinarayanan, T. Malisiewicz and A. Rabinovich, "RoomNet: End-to-End Room Layout Estimation," *2017 IEEE International Conference on Computer Vision (ICCV),* Venice, 2017, pp. 4875-4884, doi: 10.1109/ICCV.2017.521.

21. L. Neumann, B. Vanholme, M. Gressmann, A. Bachmann, L. Kählke and F. Schüle, "Free Space Detection: A Corner Stone of Automated Driving," *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, 2015, pp. 1280-1285, doi: 10.1109/ITSC.2015.210.

22. N. Sahoo, H.-W. Lin, and Y.-H. Chang, "Design and Implementation of a Walking Stick Aid for Visually Challenged People," *Sensors*, vol. 19, no. 1, pp. 130, Jan. 2019, doi: 10.3390/s19010130.

23. M. Bousbia-Salah, M. Bettayeb, and A. Larbi, "A Navigation Aid for Blind People," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 3–4, pp. 387-400, Dec. 2011, doi: 10.1007/s10846-011-9555-7.

24. S. Saleh, H. Saleh, M. A. Nazari, and W. Hardt, “Outdoor Navigation for Visually Impaired based on Deep Learning,” in *Actual Problems of System and Software Engineering (APSSE 2019),* Nov. 2019, pp. 397–406.

25. Edmund Optics Inc., Best Practices for Better Imaging: Understanding Focal Length and Field of View, Edmund Optics Inc., Barrington, NJ, USA, Accessed Sept. 23, 2020. [Online]. Available: https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/

26. R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 580-587, doi: 10.1109/CVPR.2014.81.

27. R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

28. X. Zhang, X. Zhou, M. Lin and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 6848-6856, doi: 10.1109/CVPR.2018.00716.

29. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

30. L. Chen, G. Papandreou, F. Schroff and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation." 2017, arXiv:1706.05587.

31. B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso and A. Torralba, "Scene Parsing through ADE20K Dataset," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Honolulu, HI, 2017, pp. 5122-5130, doi: 10.1109/CVPR.2017.544.

32. B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso and A. Torralba, "Semantic Understanding of Scenes Through the ADE20K Dataset," *Int J Comput Vis*, vol. 127, no. 3, pp. 302-321, doi: 10.1007/s11263-018-1140-0.