Çankaya University
Faculty of Engineering
Computer Engineering Department

Project Report

CENG 407

Innovative System Design and Development I

Advisor: ***Dr. Serdar Arslan***

Prepared by	ID
Batuhan DİLEK	201911023
Görkem KARABAY	201811039
İbrahim Efe ERER	201911403
Mert YILDIRIM	201911071
Zeynep Tulya AYTEKİN	201811008

Content:

Abstract

Machine Learning technology and the object detection algorithms include multiple engineering concepts. In this project, it is aimed to combine these techniques and generate an international app for all of the visually impaired people all around the world. Therefore, real time object detection operation will be handled and will be backed up by the guidance and the warning systems which contains speech-to-text and text-to-speech algorithms in order to have a connection with our users. The main motivation behind this project is to enchant the visually impaired people's life. They have enough struggles as it is and we desire to change that. By using our app, They can stream their surroundings to app and It will guide our users to their desired object. Also, during their daily lives the app will watch and warn them about their surroundings if it calculates it as a threat. In this way, GenEye will improve the living conditions of visually impaired people and make their lives better.

Introduction

In this document we give detailed information about our Project GenEye. Therefore, we mentioned previous Literature Review, Software Requirement Specification (SRS) and Software Design Description (SDD) documents of the project. Finally, we state our work plan for GenEye's development.

Problem Statement

Daily lives of visually impaired people are full of dangers and uncertainties. Normal walks might end with dangerous injuries that are caused by unpredicted environmental obstacles. Searching for an object that they need immediately might become a difficult process without the help of another person. In our application we are replicating the necessary visual capabilities to safely and easily accomplish these activities which are to be used by these people who are in need.

Background or Related Work

Machine Learning is one of the most important topics of Computer Science. Especially in daily life, we use technology in different activities but we can also improve people with disabilities with it. Therefore, companies of different origins try to develop better solutions, such as; Google, SHG Technologies, Baidu etc. developed example solutions of Smart Vision Glasses. These inventions used different kind ways for people with disabilities and without disabilities.

Solution Statement

To solve this issue, we have used an Android mobile phone and effective machine learning algorithms such as YOLO, CNN, RNN, SURF. All techniques and algorithms will be explained in SRS and SDD documents which are stated in this document. By processing real time video and thanks to the effectiveness of the YOLO algorithm , we are able to detect any object or obstacle and warn/notify the user about it in realtime.

Motivation

Various technical principles including artificial intelligence, mobile application development, computer vision and others are used in this project. We simultaneously become better at these concepts as a result of our studies in the project. The main motivation in the development of this application is to support people with visual disability and ease their lives by seeing the surroundings for them.

1. Literature Review

1.1. Introduction

In our daily lives visually impaired people are facing a bunch of difficulties which seriously affects their life. To give a simple example, according to a research conducted by Jones, N., Bartlett, H. E., and Cooke, R. in 2019 [1] more than 65% of visually impaired people either shop with support or do not shop at all. According to the same research participants who shop independently memorize the shopping route they choose and use the same brands all the time and they get confused when these supermarkets change the shop layout. These people deserve to have an easy and secure life as people without visual difficulties. So in the end, we picked this challenge to be solved and helped those people in distress. GenEye is a mobile project that visually impaired people would download into their phones. This app would inform them about their surroundings in order to make their lives safer and help them locate the stuff around them. Also financially, GenEye will help them to be aware of the amount of money which is transferred and used during trades. Based on these solutions and the solutions for challenges may come, we want to achieve the completion of this project for the sake of people who need this.

1.2. Machine Learning

1.2.1. Techniques

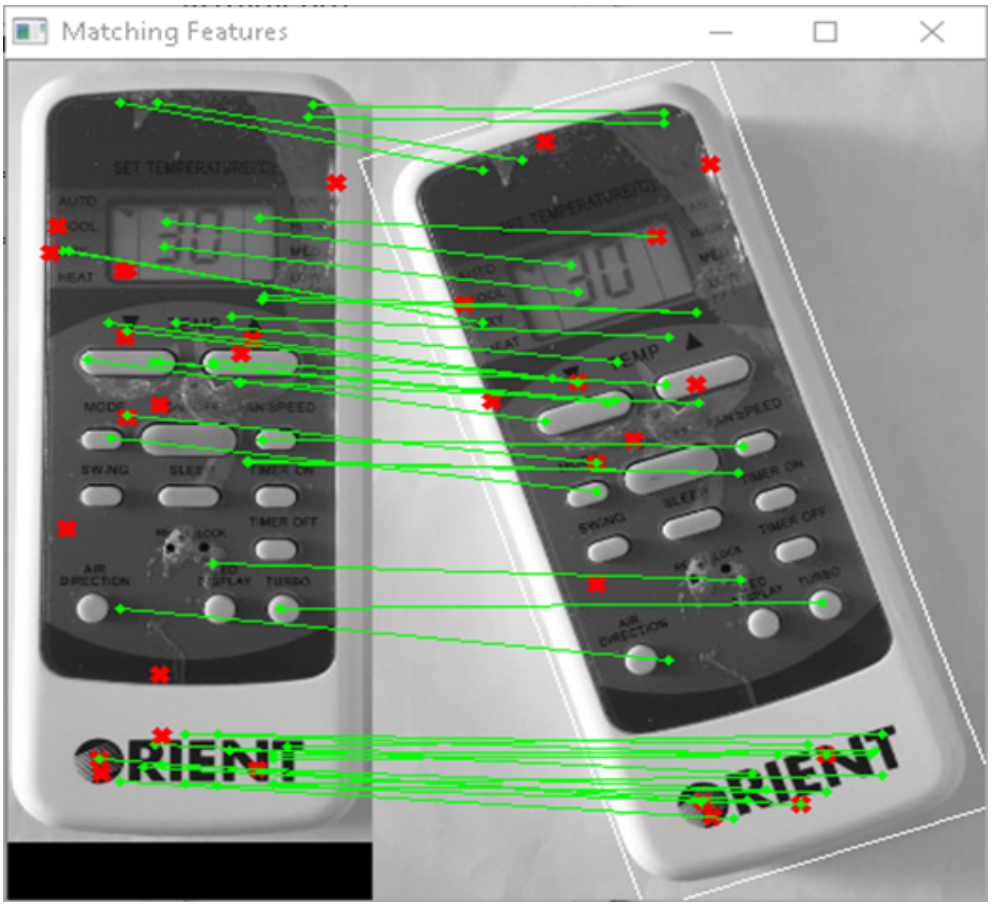
1.2.1.1. YOLO (You Only Look Once) Algorithm

YOLO is used in many object detection projects [2], [3], [4]. In our project we will use YOLO for its pre-trained object detection models and because it is very efficient for real time object detection [5]. Pre-trained models which detect objects like toothbrush, chair, and bed [6] will be used for general object detection modes that will be added to our application. Median Hardiv Nugraha and Dina Chahyati created a mobile application using YOLO which helps tourists to identify or learn about the historical monuments [4]. This application uses phone camera and voice commands of the user which is similar to our project idea, and they obtained high accuracy and high performance on different datasets compared to Retinanet. Some speed comparisons of other object detection algorithms on same data set with YOLO [7]:

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	**73.9**	**85.5**	**82.9**	**76.6**	**57.8**	**62.7**	**79.4**	77.2	86.6	**55.0**	**79.1**	**62.2**	87.0	**83.4**	**84.7**	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	83.3	**81.8**	**48.6**	**73.5**	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	**89.4**	49.4	75.5	57.0	**87.5**	80.9	81.0	74.7	41.8	71.5	68.5	**82.1**	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [27]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	**68.8**	75.9	71.4
NoC [28]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [32]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

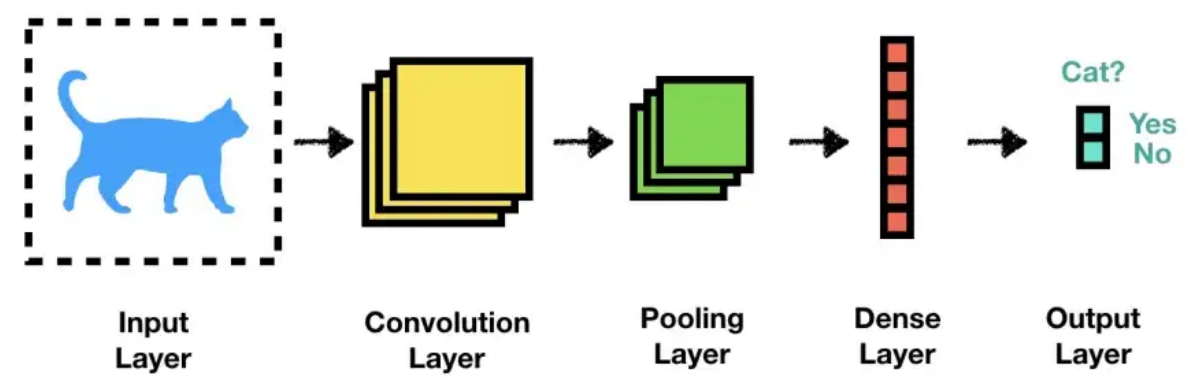
1.2.1.2. SURF

The SURF method (Speeded Up Robust Features) is a fast and robust algorithm for local, similarity invariant representation and comparison of images. The main interest of the SURF approach lies in its fast computation of operators using box filters, thus enabling real-time applications such as tracking and object recognition. SURF is used in similarity matching projects such as: [3] this approach may be useful in our project's money recognition mode. An example of SURF algorithm on action is given below:

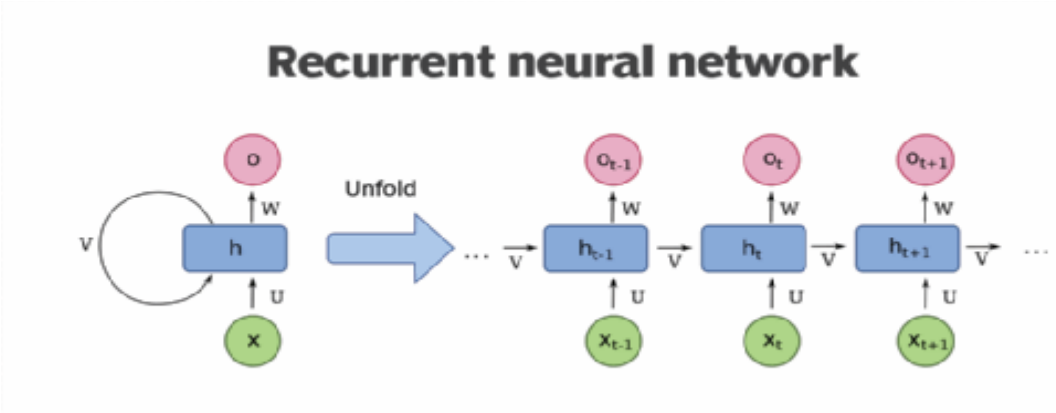


1.2.1.3. CNN and RNN

RNN and CNN [8] are two basic architectures of deep learning [9]. RNN is used for problems expressed as time series. For example, audio processing, speech-to-text, text-to-speech, chatbot, machine translation, and text summarization. In the RNN structure, the information coming to the neural networks is associated with certain weight constants in the layers, resulting in an estimate. If we recall the CNN architecture, the neural network receives data of a certain size. Therefore, CNN is a deep learning algorithm that is generally used in image processing and takes images as input. This algorithm, which captures and classifies features (features) in images with different processes, consists of different layers.



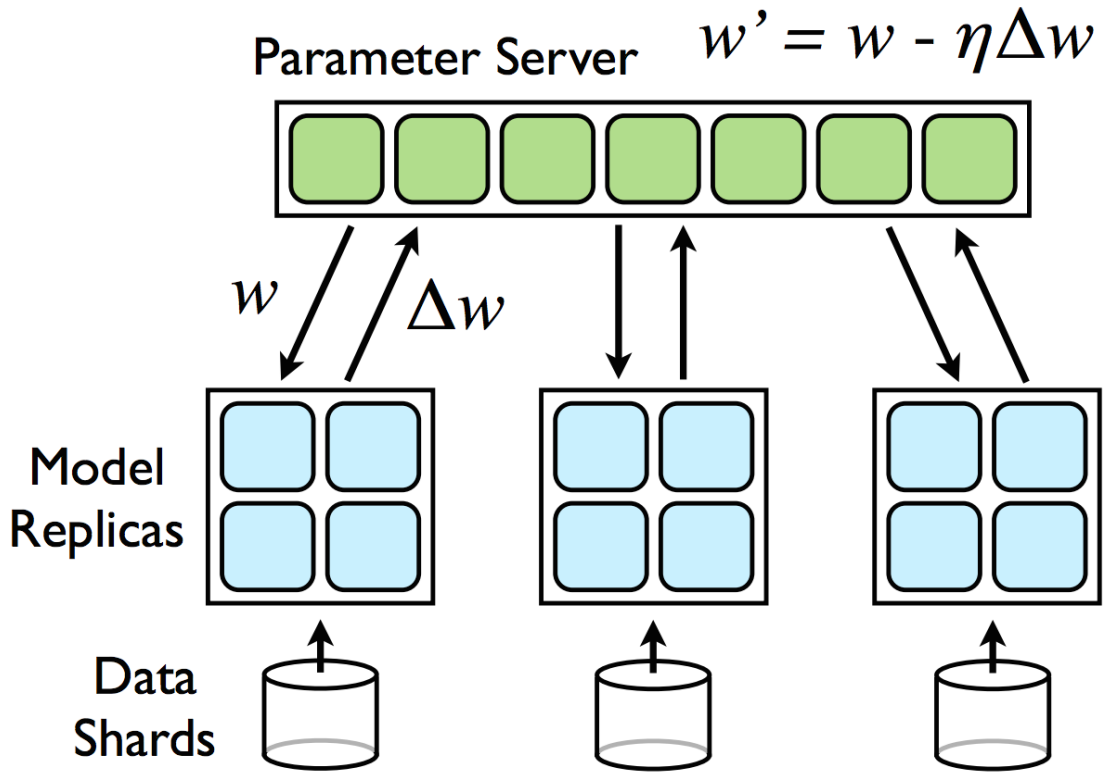
Processing of CNN algorithm



Processing of RNN algorithm

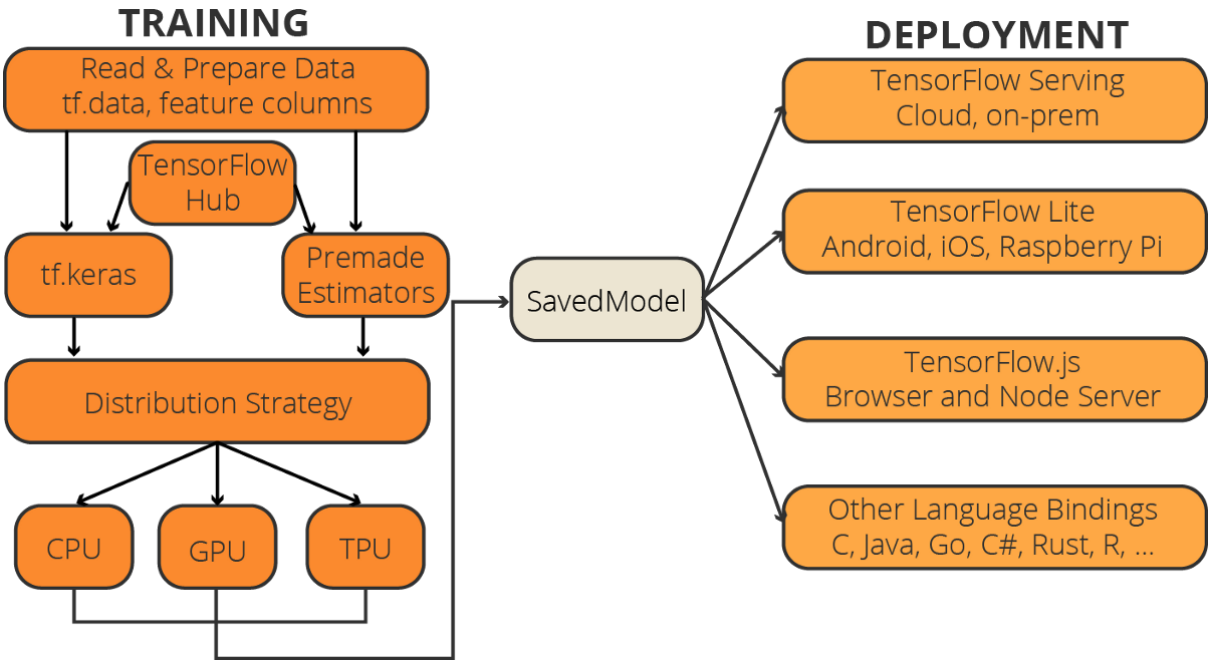
1.2.1.4. DistBelief

From 2011, Google Brain developed DistBelief as a machine learning system based on deep learning in neural networks [10]. Its use in both research and commercial applications in various companies has grown rapidly. TensorFlow is the second generation system from Google Brain and was released in 2017. While the reference application runs on single devices, TensorFlow can run on multiple CPUs and GPUs.



1.2.1.5. TensorFlow

TensorFlow [11] is a free and open source software library for deep machine learning. It can be used for a number of tasks, but with a special focus on the training and the maintenance of deep neural networks. TensorFlow is a symbolic math library based in Phyton and the data acquisition and differentiable programming.



1.3. Computer Vision

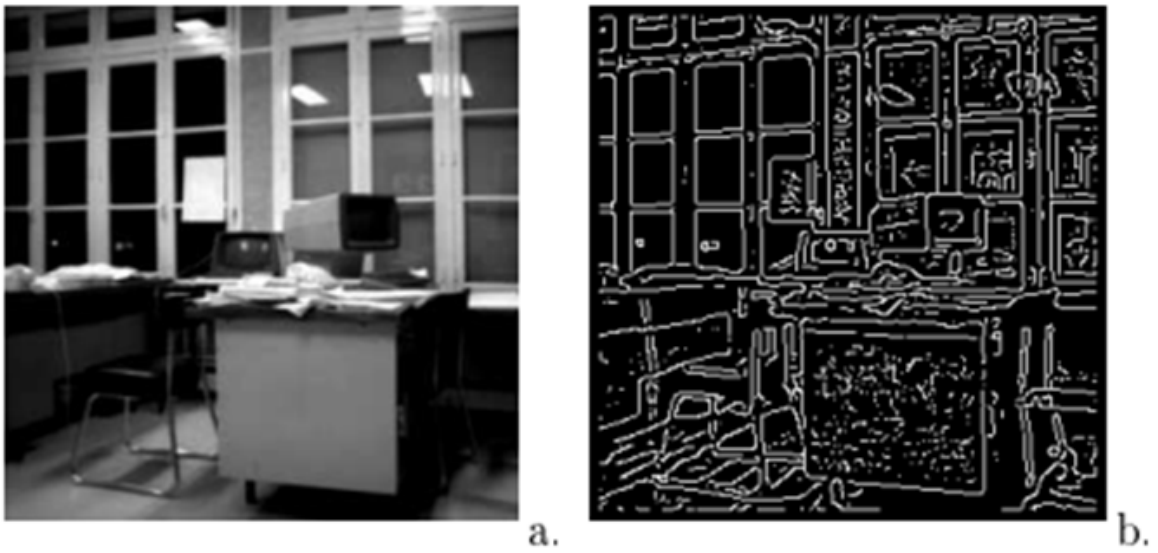
Computer vision is the field of computer science where the aim is to imitate human visual system for computers. We use deep learning algorithms to teach the computer about the objects of interest so that it will be able to recognize these objects in different environments. Using this teaching we can use a computer vision application to help visually impaired people [12].

1.3.1. Techniques

1.3.1.1. Edge-Based Methods

Edge-based method is a technique is segmentation of one or more objects in an image. This section can be detected by the edges or the interiors of them. There are two main steps that can be applied through this method. These are called Edge Detection and Edge Linking.

In Edge Detection, the goal is to recover physical properties of objects in a scene, such as the location of object boundaries and the structure, color, and texture of object surfaces, from the two-dimensional image that is projected onto the eye or camera. The first clues about the physical properties of the scene are provided by the changes of intensity in the image.[13] We can also see different solutions with edge detection with research of Djemel Ziou and Salvatore Tabbone[14].



a) Original image, b) Position information provided by an edge detector

In Edge-Linking, improvement of edge detection process is supervised. It is aimed to form the object by linking adjacent edges. This operation can be applied through two methods called Local Processing and Global Processing. In Local processing, gradient and direction is used in order to link the edges that are neighbors. If the edges which are specified beforehand have a common or similar direction vector, then it means that they can be linked. Whereas in global processing, the main point is using HOG transformation. [15]

Advantages of using this method can be explained in two ways. One of them is that it simulates similar work in the way that the human brain works. Second one is that it can work with images containing powerful contrast, especially between the object and the background.

1.3.1.2. Colour-Based Methods

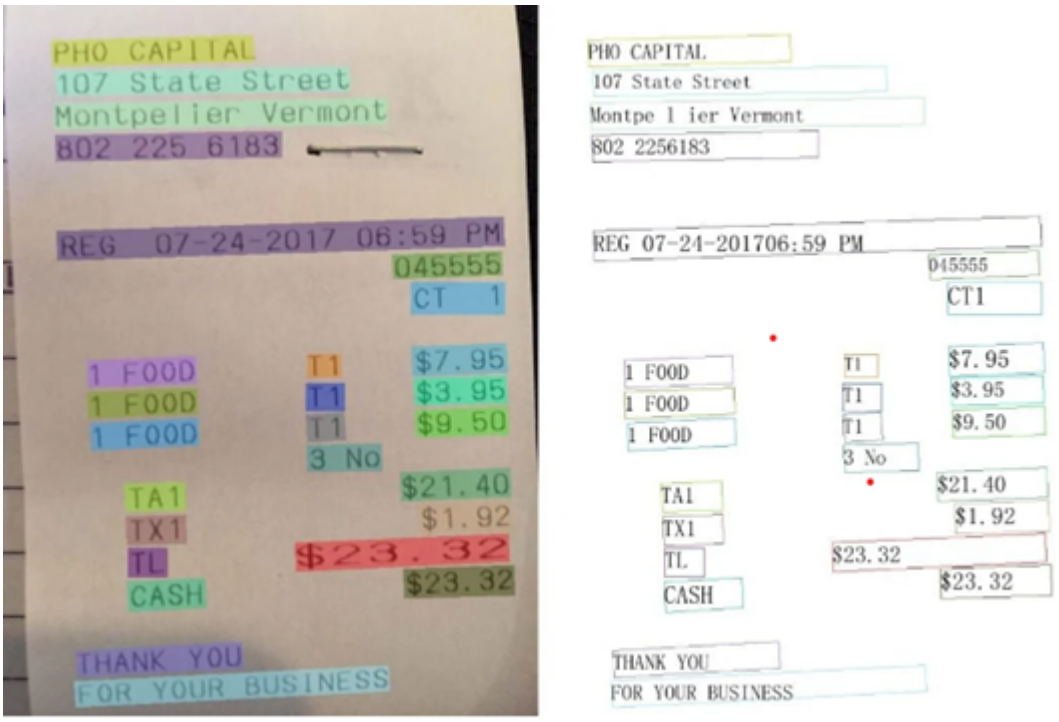
Colour-based methods utilize the fact, that traffic signs are designed using a very limited number of colors, which are usually contrasted to the surroundings. Image areas containing these colors are extracted from the input image and then used as a base for detection

R. Kulkarni, S. Dhavalikar and S. Bangar created a computer vision app which allows self-driving cars to detect traffic lights [16]. They used color segmentation to separate traffic lights from other objects present in the scene. This method can be added to our project for detecting crosswalk lights and whatever the current light means.

1.3.1.3. Character Based Methods

Two of our modes which are: Money Recognition and Public Transport Recognition will be recognizing numbers and characters. For Public Transport ID Recognition purpose, STR can be used. In computer vision, machines can read text in natural scenes by first detecting text regions, cropping those regions, and subsequently recognizing text in those regions. The vision task of recognizing text from the cropped regions is called Scene Text Recognition (STR) [17]. There is a that recognizes text from a two-dimensional perspective using STR. This research may help us identify numbers in our scene [18].

For the Money Recognition and Counting mode, OCR and Edge-based Methods can be used as well. OCR stands for Optical Character Recognition and refers to a software technology that electronically identifies text (written or printed) inside an image file or physical document, such as a scanned document, and converts it into a machine-readable text form to be used for data processing. It is also known as text recognition [17]. There is a project [19] that recognizes Iraqi license plates using OCR.



1.4. GenEye Features

There will be several features in our project mainly considering real time processing. GenEye will process the data that it gathered by observing that specific environment in real time. And this property will be applied mainly using the YOLO algorithm.

YOLO allows the development of object detection models which can work on non GPU devices [20]. As we are developing a mobile application it will be a good idea to use this opportunity.

1.4.1. Surrounding Detection

In our surrounding detection system, when the user activates this mode, it will recognize the environment and inform the user about the dangers and specific things that the user has to pay attention to. This mode will mainly consider easing the way that users walk through in their daily lives.

There is a project developed by W. C. S. S. Simões and V. F. de Lucena [21] which requires additional equipment to use. In our application our aim is to only rely on mobile phones. This user will not have to purchase additional equipment and in situations where a visually impaired person loses a walking stick our application will be available instantly.

1.4.2. Money Recognition and Calculation

In money recognition and calculation mode, the mode will first detect the banknote itself, and how much amount it represents. And it will inform the user about the amount of money they hold in their hand. This mode mainly aims to provide a more comfortable shopping experience to users.

The OCR method can be a good help in the development of this mode since it's pretty good at recognizing text.

Gustavo Adrian Ruiz Sanchez published a paper that shows the implementation of the SURF algorithm and Pixel-Based Adaptive Segmentator(PBAS) to recognize banknotes[3]. In real life scenarios, someone may try to give fake banknotes to a visually impaired. This research may help us overcome this problem. With the help of this research we can detect fake money and inform the user.

1.4.3. Public Transport Recognition

In public transport recognition mode, it will detect the id of the bus by recognizing the number that is written on it. So it will help users to find the right bus that they wanted to take. This mode aims to help users by preventing mistaken road trips that they may experience throughout different routes.

The STR method can be a good help in the development of this mode since it can easily determine the writings in specific sections of an image.

U. Franke and A. Ismail created a project [22] that is for buses to precisely locate the position of bus stops. It gave us the idea of public transport recognition where users are able to learn which bus is near the bus stop by using our app.

1.5. Similar Applications

1.5.1. Smart Vision Glasses by SHG Technologies

SMART VISION GLASSES is a wearable device for the blind/visually impaired using Artificial Intelligence, Machine Vision, and Machine Learning. The device has an electronic brain (artificial intelligence) which transforms their dark world into a more interactive, interesting and independent world. They can read their favorite book, catch up with the news in magazines and newspapers, students can write their exams without scribes and so much more. [23]

- It recognizes objects in its view such as vehicles, obstacles, traffic signals, zebra crossings, furniture
- Identifies people, approximate age, expressions and recognizes people if a name is attached
- Can read books, road signs and newspapers and manuscripts
- Can talk in Indian regional languages such as Hindi, Tamil and Telugu
- Give directions to a destination when asked, using GPS

1.5.2. Dulight by Baidu

Dulight is a mobile application for the blind/visually impaired using Artificial Intelligence, computer vision. Dulight makes it possible for visually impaired people to identify their surroundings and people around them. [24]

- It recognizes the objects and obstacles around the user.
- It can recognize what kind of product the user holds in their hands.
- Can recognize how much money does the user hold in their hands.
- Can recognize faces of people around the user.

1.6. Potential Problems

Our project aims to use Machine Learning, Deep Learning and Computer Vision; a mobile platform may fail to handle all the computation. Due to this fact, there may be some delay problems, which can be a big problem for the user. To overcome this problem we may build a server for the calculation and prediction purposes.

The object detection models may be too general for some specific scenes and this will cause detection rates to decrease.

If we detect any low detection performance for an object it will take some time to increase the number of data for that model and retrain it with this new data.

User's device and device's camera quality will cause different results on different devices. Preprocessing of images might not be an answer to this problem.

1.7. Conclusion

Visually impaired people face many problems in their lives' every day. We can observe different kind of solutions for some of their problems as we mentioned in part 5. In our time, we can see new technologies for helping people, these technologies such as: image process, machine learning and deep learning, etc. We can use these technologies to improve visually impaired people life standards and ease their daily struggles, with image process and machine learning technologies we can teach computers how a human can see and with deep learning computer can understand what it sees. Therefore, GenEye can process all of these new data to return helpful information for users.

1.8. References

[1] Jones, N., Bartlett, H. E., & Cooke, R. (2019). An analysis of the impact of visual impairment on activities of daily living and vision-related quality of life in a visually impaired adult population. British Journal of Visual Impairment, 37(1), 50–63. https://doi.org/10.1177/0264619618814071

[2] Hendry, Rung-Ching Chen, Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning, Image and Vision Computing, Volume 87, 2019, Pages 47-56, ISSN 0262-8856, https://doi.org/10.1016/j.imavis.2019.04.007.

[3] G. A. R. Sanchez, "A Computer Vision-Based Banknote Recognition System for the Blind with an Accuracy of 98% on Smartphone Videos," Journal of the Korea Society of Computer and Information, vol. 24, no. 6, pp. 67–72, Jun. 2019.

[4] M. H. Nugraha and D. Chahyati, "Tourism Object Detection Around Monumen Nasional (Monas) Using YOLO and RetinaNet," 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2020, pp. 317-322, doi: 10.1109/ICACSIS51025.2020.9263240.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. 'You Only Look Once: Unified, Real-Time Object Detection'.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár C. L. (2014, May 1). 'Microsoft Coco: Common Objects in Context'.

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[8] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[9] Li Deng and Dong Yu (2014), "Deep Learning: Methods and Applications", Foundations and Trends® in Signal Processing: Vol. 7: No. 3–4, pp 197-387.

[10] Picton, P. (1994). What is a Neural Network?. In: Introduction to Neural Networks. Palgrave, London. https://doi.org/10.1007/978-1-349-13530-1_1

[11] https://www.tensorflow.org/

[12] Huang, T. S. (1996). Computer Vision: Evolution and Promise. CERN Document Server. Retrieved from https://cds.cern.ch/record/400313

[13] https://dspace.mit.edu/handle/1721.1/6429

[14] https://www.researchgate.net/profile/Djemel-Ziou/publication/312890367_'Edge_detection_techniques_An_overview'/links/59d3c61c0f7e9b4fd7ffbfbb/Edge-detection-techniques-An-overview.pdf

[15] https://www.geeksforgeeks.org/region-and-edge-based-segmentaion/

[16] R. Kulkarni, S. Dhavalikar and S. Bangar, "Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697819.

[17] Visio AI, Optical Character Recognition (OCR)

[18] Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., Yao, C., & Bai, X. (2019). Scene Text Recognition from Two-Dimensional Perspective. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 8714-8721. https://doi.org/10.1609/aaai.v33i01.33018714

[19] S. S. Omran and J. A. Jarallah, "Iraqi car license plate recognition using OCR," 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), 2017, pp. 298-303, doi: 10.1109/NTICT.2017.7976127.

[20] R. Huang, J. Pedoeem and C. Chen, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 2503-2510, doi: 10.1109/BigData.2018.8621865.

[21] W. C. S. S. Simões and V. F. de Lucena, "Blind user wearable audio assistance for indoor navigation based on visual markers and ultrasonic obstacle detection," 2016 IEEE International Conference on Consumer Electronics (ICCE), 2016, pp. 60-63, doi: 10.1109/ICCE.2016.7430522.

[22] U. Franke and A. Ismail, "Recognition of bus stops through computer vision," IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683), 2003, pp. 650-655, doi: 10.1109/IVS.2003.1212989.

[23] SHG Technologies Smart Vision Glasses

[24] Dulight--Eyes for visually impaired

[25] Tsoi, A.C. (1998). Recurrent neural network architectures: An overview. In: Giles, C.L., Gori, M. (eds) Adaptive Processing of Sequences and Data Structures. NN 1997. Lecture Notes in Computer Science, vol 1387. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0053993

2. Software Requirement Specification (SRS)

2.1. Introduction

Gen-Eye is a mobile application that will be used by visually impaired people to help with daily activities. Our app will be able to detect obstacles using various algorithms and device resources in the user's way and warn the user with voice guidance. Device resources include speakers, microphone and camera. Through these resources the user will be able to communicate with the application and the application will be able to warn the user about the environment. Algorithms include YOLO, SURF, CNN and RNN; these algorithms can be used for training object detection models. After training these models we can use these models in our application to detect daily objects through the user's phone camera. This application can be a helpful tool for visually impaired people because it is easy to access to people with a phone with a camera. Main reason this application is required is that people with any kind of visual impairment has dependency on other people for some daily activities and for their safety. Our application will decrease the need for other people to do these activities and increase their safety during daily activities like walking.

2.1.1 Purpose of This Document

In this project, our goal is to develop a system that helps the visually impaired people during their daily life. This mobile app will enhance their way of being aware of their surroundings. If we need to go into the details of its features, It detects the required stuff that the user wants to find, it will inform the user about their surroundings and grant them a more safe and secure life. With these planned features GenEye will provide an easy life to the visually impaired people. While doing the project, we design to provide diverse software requirements such as Machine Learning, Computer Vision, YOLO Algorithm, SURF,(to be added). For the continuation of the project, we are planning to test its Computer Vision with YOLO to recognise the required objects. If the object is recognised, GenEye will inform the user by voice. In short, GenEye will enhance the life of visually impaired people and this SRS document contains the project and software requirements for GenEye.

2.1.2. Intended Audience and Reading Suggestions

This document is Software Requirement Specification report, which is intended for domain experts, software/computer engineers, developers, testers, and project managers. There is another document which we prepared beforehand, Literature View, which is also helpful to understand the concepts, algorithms, techniques covering our project. It also provides an extensive glance about the project. Other than this, this document will provide information about what the software will do and how it will be expected to perform. And It also gives insight into what sort of functionalities the product needs to address for all the stakeholders.

2.1.3. Scope of the Project

In our daily life, Vision is important for us to comprehend the environment. In addition to this, humans' vision has an 87% ratio of understanding the perception [1]. Unfortunately, visually impaired people have major obstacles in their ways. They encounter different kinds of setbacks too. There are some solutions for their problems but those are usually inefficient or insufficient for visually impaired people.

The reason behind GenEye is to create a better solution for visually impaired to use in their smartphones which can be easily accessible and simple to use. Furthermore, GenEye will have Machine Learning algorithms to improve itself and understand the surroundings. GenEye will have a server system easily accessible for image processing systems and data banks.

There are two main features in the system that we offer our users to utilize, these are object detection and obstacle detection. Users can use object detection while they are filming a video footage. Users will say the name of the object they are required to find and the system will inform users with a voice message when it finds the object that the user required. With the program's guidance users should match up the required object by center of the screen which will help them to realize the exact direction of the object. Secondly, the obstacle detection feature also works as real time streaming footage. To be more precise, the program will enhance the visually impaired users during their daily walk routine by notifying them of their surroundings such as when they come to an obstacle, the program will warn the user with voice warning.

Our system will include:

- Object detection for Users' required objects
- Obstacle detection for User's daily routine
- Voice warning and notification system

2.1.4. Glossary

Term	Definition
Actor	An Actor can be an user or another software system that interacts with the main system
User	Users are visually impaired people who seeks help and uses this application
Java	Java is an Object Oriented Programming Language and computing platform first released by Sun Microsystems in 1995.
Python	Python is an interpreted, object-oriented, high-level programming language.
Software Requirement Specification (SRS)	A document that describes what the software will do and how it will be expected to perform.
Android	Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.
Android Studio	Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto.

Object Detection	Object detection is a computer vision technique for locating instances of objects in images or videos.
Computer Vision	Computer Vision is a subfield of Artificial Intelligence that makes systems see and identify objects using digital images, videos and other visual inputs.
Machine Learning	Machine Learning is a branch of Computer Science and Artificial Intelligence which tries to imitate human learning using algorithms.
Chaquopy	Chaquopy makes it possible to include and run Python scripts inside of a mobile application.

2.2. Overall Description

2.2.1 Product Perspective

2.2.1.1 System Interfaces

In our system our software will communicate with mobile phone’s camera and microphone. Firstly, our application will receive the user's environment information from the phone camera then feed this information to the object detection model to find objects of interest. Microphone will be used to receive voice commands from the user to understand and act according to these commands.

2.2.1.2 User Interfaces

Our application will consist of simple and easy to use user interfaces so that user will be able to use our app even with high visual impairment. When the application starts there will be only two options presented to the user, these are “Object Detection Mode” and “Obstacle Detection Mode”. Users will be informed about their choice. If they enter “Object Detection Mode” the user is required to choose the object they want to find, at this stage the user will be allowed to select their decision with voice input but also they will have the option to select it from a list of possible options.

Our system includes multiple software subsystems and multiple computer science concepts like Machine Learning, Deep Learning and Computer Vision. Due to the multiplicity of concepts, we will be using different programming languages such as Python and Java. In addition, we are planning to introduce this system as a mobile application, we will use Android Studio as our primary IDE. For Python scripts other IDEs may be used. In this section, it is summarized what features are included and what those features will do. This project aims to run on Android 11+ mobile phones. Any camera or hardware that is running on Android 11+ will suffice. Our project has 2 modes. Which are Object Detection and Obstacle Detection. For both modes, we will be using Computer Vision, Machine Learning algorithms and Speech to Text / Text to Speech tools to identify objects and inform the user about it.

In the table below, it is stated what modes the project consists of, a brief explanation and which software tools will be used.

Task	Description	Software Tools
Object Detection	System shall detect objects inputted by the user and give a feedback if the object is in sight of the camera	• Python • Java • Android Studio • Chaquopy
Obstacle Detection	System should detect objects which visually impared people should watch out for.	• Python • Java • Android Studio • Chaquopy

2.2.2 Product Functions

2.2.2.1. Object Identification

Our system is mainly dependent on object identification. Program should understand the difference between an object and the background from the footage to work. It does it by finding the color difference between the pixels in the footage, after that program separates these pixels and understands the objects.

2.2.2.2. Environmental Scan

Environmental scan will be used for searching objects in the user's environment. This process requires a mobile phone with a sufficient camera. When the user opens the application two options will be presented. First option is obstacle detection. For this option, the user will direct the phone to the direction that he/she will be walking. Phone camera will get the environment as image inputs and give responses according to the environment whether there is an obstacle in the user's way or not. Second option is object detection. When a user selects this option he/she will have to select the object to be searched for then the environmental scan begins. During the scanning user will sweep through his/her environment to find the object of interest. If the object that is being searched is detected during this sweeping operation then the response will be given as a voice output. Otherwise there will be no response to the user.

2.2.2.3. Guidance System

Guidance System is one of the most significant systems of the GenEye which will help the user to identify the place of the required object with the help of Notification System. To do so, the program will try to guide the user to centralize the required object on the screen with notifications which will grant them the exact direction of the object to know its location.

2.2.2.4. Notification System

Notification system is one of the subsystems of the Object Detection which will help and tell the user to find or know the location of their required object. notification system will inform the user with voice when the required object is detected on the camera. Also, the notification system will aid the user by voice to find their required object during the guidance system.

2.2.2.5. Search by Voice Input

Users should be able to tell the system what they are looking for via speech to text methodology. System should understand the speech input and convert it to a string. Thus, the system knows exactly what to search for.

2.2.2.6. Level of Threat

As a part of the warning system, it is possible to observe that the level of the warning becomes intensified if the intended object/obstacle is coming closer in the sight of the camera.

2.2.3. User Classes and Characteristics

There is only one actor in our system: Visually impaired user.

2.2.3.1. Visually Impaired User

Visually impaired users will not have to login or register to use our application. Only requirement from the user is to download the application and start using it. Tasks are given below:

- Open the application
- Point the camera
- Give voice inputs to the machine

2.2.4. Operating Environment

In this section, it is explained what environment the application needs to be able to run.

- The device must have a camera in order to process frames and detect objects.
- Device should operate within an Android 11+ operating system

2.2.5. Constraints

Our program mainly depends on video functions and user's interactions. If there is any damage or malfunction in a smartphone's camera or user doesn't use the program efficiently enough it will affect the program significantly. There are some scenarios such as:

- Weather conditions can affect camera such as foggy weather or hurricanes. System should warn the user about the camera's condition.
- System should detect objects and obstacles in seconds to work in the proper way. If any malfunction system should warn the user.
- System can misidentify some objects in the environment as intended objects. System should warn the user about these objects.
- System might not be able to see transparent objects such as glass or some plastics.
- User's smartphone should have minimum requirements for the program in hardware and software.

2.2.6. Dependencies and Assumptions

Listing and briefly explaining dependencies and assumptions of our system.

Environment Conditions: Our application doesn’t require any specific environment to work in as long as user and devices can survive.

Operating System: Our system is designed primarily for mobile usage where the system will require usage of the phone's camera. Any android version 4.4.x or higher will be sufficient to run our application so that most phones will be able to run our application and satisfying results can be obtained by users.

Hardware: System requires usage of two hardware components. These are the phone's camera and microphone. Phone should at least have an 8 megapixel camera so that the image quality is not reduced too much. The microphone is sufficient if it can receive voice input that is understandable by the system.

Software: In order to build our application we will require an IDE to write our code which will be Android Studio. Other than writing our code our application will require model training phases and usage of these models in our application. To train object detection models Python programming language will be used with various algorithms like YOLO. We will develop our mobile application using Java programming language. To train out object detection models we will use frameworks like darknet and libraries like Tensorflow. These training phases are long and require strong machines with GPU so we will also use cloud computing to train our models for quality results in relatively shorter times.

2.3. Requirement Specification

2.3.1. Interface Requirements

2.3.1.1. User Interfaces

Our application and system will be in English. It should be easy to use for the visually impaired. Whenever the user touches modes, the program must say what mode is being touched at that moment via text to speech. The buttons should be in different colors since not every visually impaired people are fully disabled. Due to this fact, it shall be a convenience if the buttons are in different vibrant colors.

2.3.1.2. Software Interfaces

The program will be developed to enhance the visually impaired people's daily lives. So we used Android Studio [3] with the Chaquopy Python SDK plugin [2] to get access to Python libraries. With the help of YOLO [4] the required objects will be identified and brought to voice by a Text to Speech tool. There will be text-to-speech to inform our users and speech-to-text to get our users required objects. For speech-to-text we are planning to use Automatic Speech Recognition(ASR) to process the voice. In ASR, an audio fire or speech spoken to a microphone is processed and converted to text.

2.3.2. Detailed Description of Functional Requirements

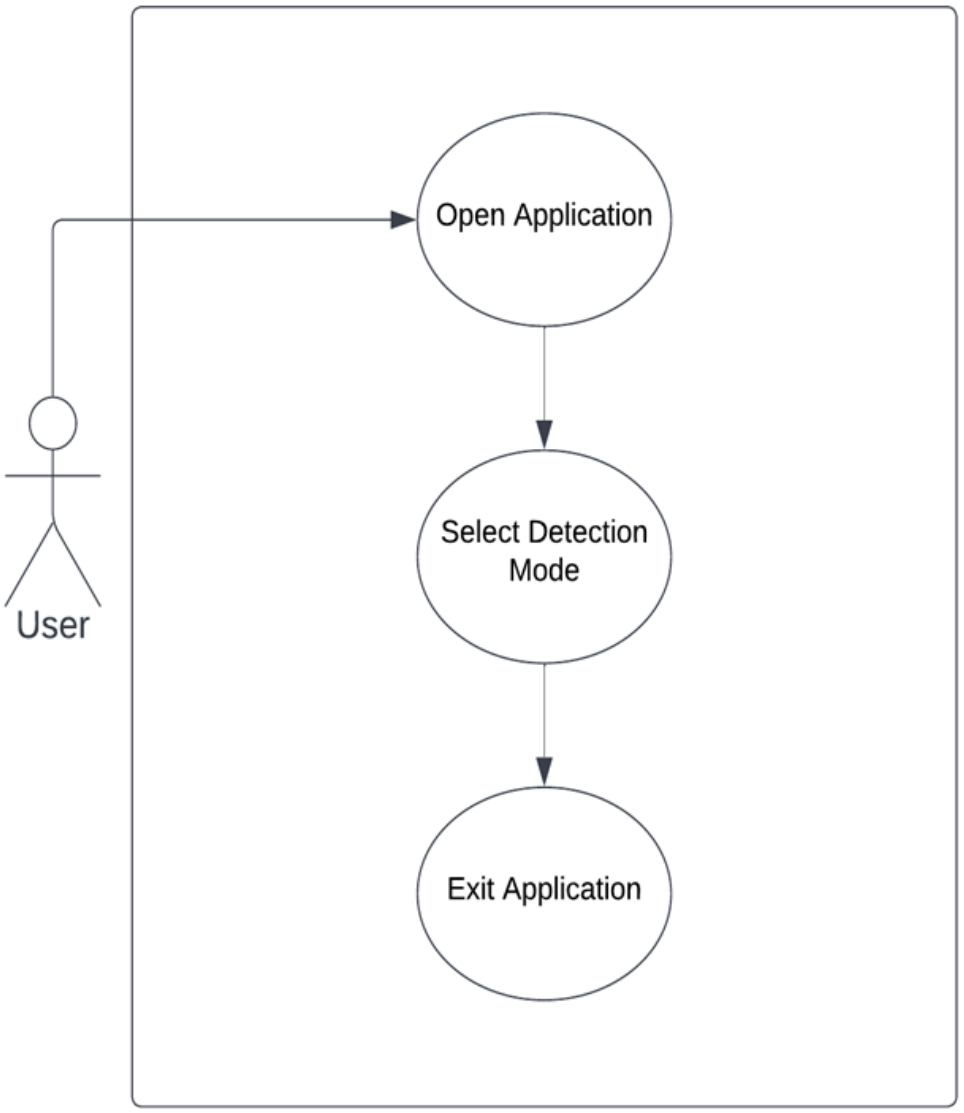
2.3.2.1. Use Cases

All use cases are briefly explained in the below table

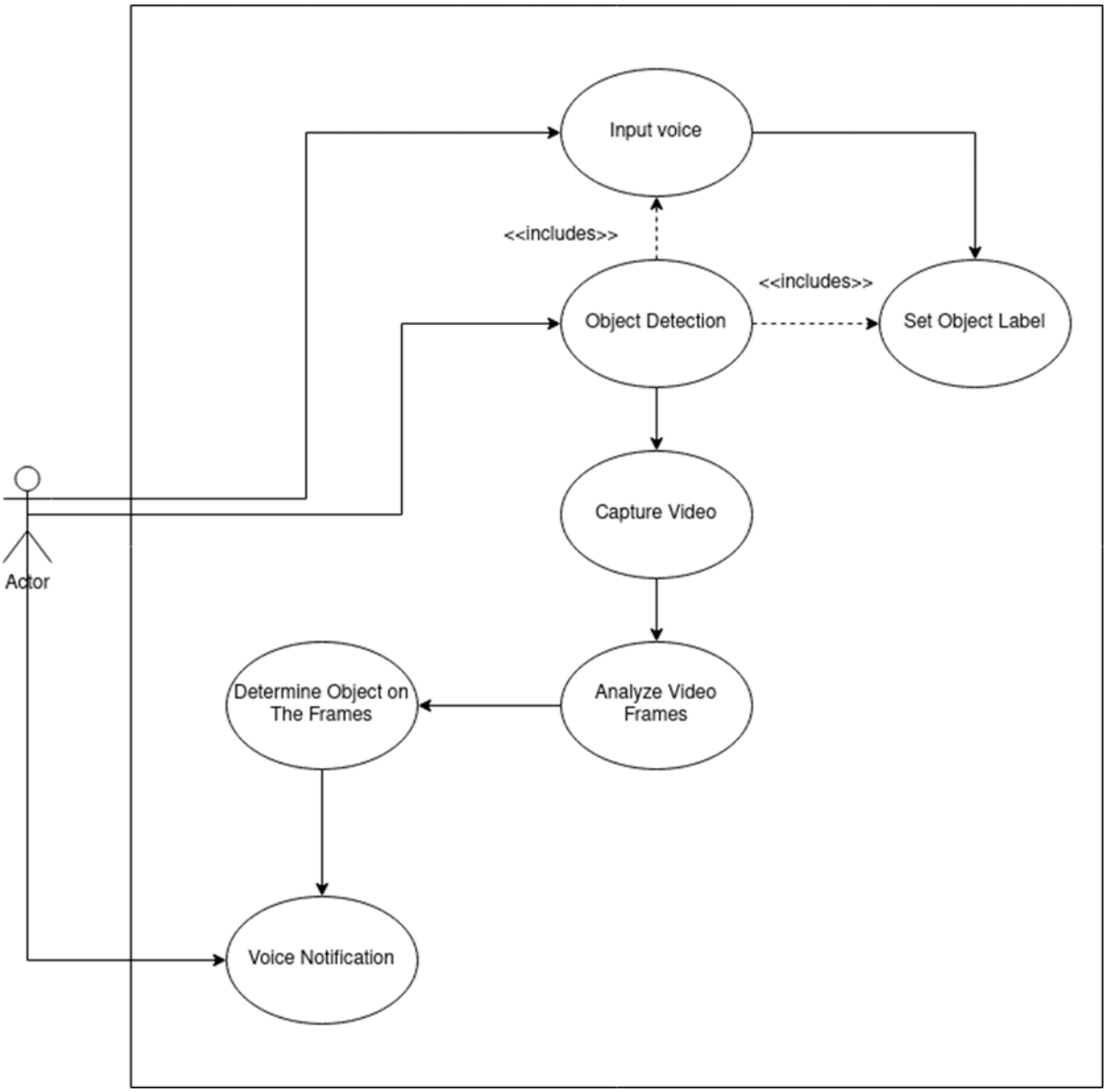
Use Case Title	Description
Open Application	Application starts
Select Detection Mode	System starts the mode that has been selected.
Exit Application	System closes the program due to User’s action.
Input Voice	Application, use User’s smartphone’s microphone to get voice input.
Capture Video	Receive environmental information as pixels
Object Detection	Checking each frame for the searched object.
Detect Obstacle	Checking each frame for an obstacle.
Determine Object on The Frames	GenEye will keep track of the object that is required by the user.
Text to Speech	GenEye guides the user by voice or warns the user from an obstacle.
Set Object Label	System get a voice input for a object label

2.3.2.2. Use Case Diagram

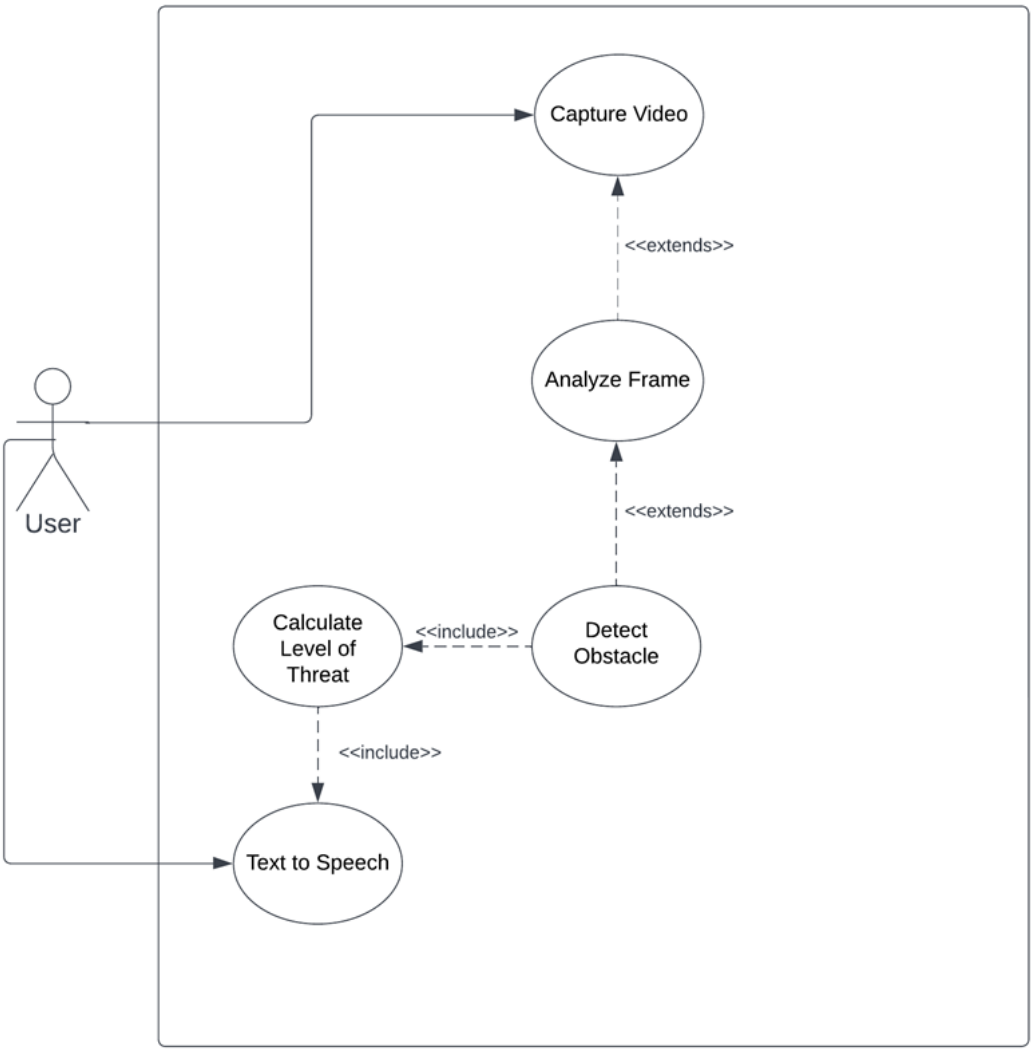
2.3.2.2.1. General Diagram



2.3.2.2.2. Object Detection Diagram



2.3.2.2.3. Obstacle Detection



2.3.2.3 Use Cases

2.3.2.3.1. Open Application

Use Case Number	1
Use Case Name	Open Application
Summary	Application starts
Actor	User
Trigger	Application itself is activated.
Preconditions	User gives the command by voice.
Scenario	● User gives the command by voice to open the application. ● Application starts
Exceptional Situations & Alternative Flows	1. Users may not pronounce the name of the application correctly, this may cause the program not to open. ● System will ask the user to pronounce it one more time.
Postconditions	User reaches the main screen of the application.

2.3.2.3.2. Select Detection Mode

Use Case Number	2
Use Case Name	Select Detection Mode
Summary	System starts the mode that has been selected.
Actor	User
Trigger	User touches the intended mode from the UI.
Preconditions	The application had to be started.
Scenario	● User touches the intended mode from the UI. ● Application gives the information by voice to user which mode is selected ● Intended mode starts to work.
Exceptional Situations & Alternative Flows	1. User may touch the wrong mode. ● There will be a voice command to move to the previous page.
Postconditions	User is able to work with the intended mode.

2. 3.2.3.3. Exit Application

Use Case Number	3
Use Case Name	Exit Application
Summary	System closes the program due to User's action.
Actor	User, Actor
Trigger	User's action
Preconditions	● Program should be active. ● User should do the closing action.
Scenario	● User touches the exit section. ● System starts the closing open modes. ● Before closing the program, the system informs the User about the closing.
Exceptional Situations & Alternative Flows	● User might close the program by manual mode in their smartphone. ● Smartphone might close due to different reasons such as low battery.
Postconditions	Program will stop working.

2.3.2.3.4. Input Voice

Use Case Number	4
Use Case Name	Input Voice
Summary	Application, use User’s smartphone’s microphone to get voice input.
Actor	User
Trigger	Object Detection mod activation.
Preconditions	● User should give permission to program for smartphone microphone. ● User’s smartphone should have a microphone with enough capabilities. ● User should choose the Object Detection mod.
Scenario	● User choose the Object Detection mod. ● Program Open mod. ● Program starts using the Smartphone's microphone. ● Program gets voice input.

Exceptional Situations & Alternative Flows	● User’s voice can interfere with other voices.
	  ● System will ask the user to say it one more time.
	● Other notifications from Smartphone can interfere with User’s voice.
	  ● System will ask the user to say it one more time.

Postconditions	Program process the User’s voice to an action.

2.3.2.3.5. Capture Video

Use Case Number	5
Use Case Name	Capture Video
Summary	Receive environmental information as pixels
Actor	User
Trigger	One of the modes present in our application is activated.
Preconditions	User's phone should have a camera that is allowed for usage for the application.
Scenario	● User activates any of the detection modes present in our application. ● User gives our app the permission to use the phone camera through phone settings.. ● System starts receiving environmental information as pixels. ● Received information stored and required object detection model is run on each frame.
Exceptional Situations & Alternative Flows	● User may not know how to give camera permission to our application. ● User may try to send frames which are not useful for the system. ● Phone's camera might have some problems.
Postconditions	● Environmental information of the user is passed to our application.

2.3.2.3.6. Object Detection

Use Case Number	6
Use Case Name	Object Detection
Summary	Checking each frame for the searched object.
Actor	User
Trigger	Object detection mode is activated.
Preconditions	Label for the searched object should be set beforehand
Scenario	● User activates object detection mode. ● User verbally inputs the object that they want to find in their environment. ● If an object is found in the scene, send a proper response. ● Keep searching until the object is found by the user or the user stops searching for the object.
Exceptional Situations & Alternative Flows	1. System may not have access to the phone's camera. ● Give voice warning to inform the user to allow our application to use the camera and return to the main menu.
Postconditions	● Threat level of the founded obstacle must be calculated

2.3.2.3.7. Detect Obstacle

Use Case Number	7
Use Case Name	Detect Obstacle
Summary	Checking each frame for an obstacle.
Actor	User
Trigger	Obstacle detection mode is activated.
Preconditions	Phone camera which can collect environmental information.User should provide a good angle for the camera to see.
Scenario	● User activates obstacle detection mode. ● User points camera to the environment for active obstacle scan ● If an obstacle which has high level of threat is found in the scene, warn and guide the user
Exceptional Situations & Alternative Flows	● Camera may be blocked by some obstacle. ● Camera quality might decrease because of smudges on the lens. ● Found obstacles may be misidentified.
Postconditions	● User finds the object that he/she was searching for. ● User realizes that the object can’t be found using our application so other methods might be useful.

2.3.2.3.8. Determine Object on Screen

Use Case Number	8
Use Case Name	Determine Object on Screen
Summary	GenEye will keep track of the object that is required by the user.
Actor	User
Trigger	GenEye detects and identifies the required object.
Preconditions	● Object detection mode must be selected. ● GenEye must be used and the phone must be stirred by the user. ● GenEye must have access to the phone's camera. ● There must be enough power on the phone's battery.
Scenario	1. User selects the object detection mode. 2. System detects and identifies the required object. 3. System will guide the user to centralize the required object on the screen. 4. System will keep guiding the user until he/she gets the required object.
Exceptional Situations & Alternative Flows	1. Program may not detect the required object. ● Restart the object finding mod. If it does it again, contact the technical service. 2. Users may lose the required object on screen by moving the phone in the wrong direction. ● System will guide the user to the last side of the object seen on the screen.
Postconditions	User finds their required object.

2.3.2.3.9. Text to Speech

Use Case Number	9
Use Case Name	Text to Speech
Summary	System guides the user by voice or warns the user from an obstacle.
Actor	User
Trigger	Determining the object on Screen, Calculation of the level of thread
Preconditions	• Object detection or Obstacle detection mode must be selected. • System must be used and the phone must be stirred by the user. • System must have access to the phone's camera. • There must be enough power on the phone's battery.
Scenario	1. User selects the object detection mode. 2. System detects and identifies the required object. 3. System will guide the user to centralize the required object on the screen by voice. *or* 1. User selects the obstacle detection mode. 2. System keeps analyzing the frames until it finds an object in front of the user. 3. Calculates the level of thread for the object. 4. If the object contains a dangerous level of thread, System warns the user by voice.

Exceptional Situations & Alternative Flows	1. Program may not understand the required object.
	    • System will inform the user that ASR(Automatic Speech Recognition) couldn't understand what they say, so System will request them to repeat their object.
	2. There may be a corrupt code within the Text-to-speech protocol so the necessary voice output might not happen as expected.
	    • Reinstalling the program.

Postconditions	Users get informed by the System with voice.

2.3.2.3.10. Set Object Label

Use Case Number	10
Use Case Name	Set Object Label
Summary	System get a voice input for a object label
Actor	User
Trigger	Input Voice sends a voice order for the system to act.
Preconditions	● User should give a voice input. ● Input Voice sends a trigger. ● Data bank should have the order.
Scenario	● Input Voice sends a trigger. ● Set Object Label look up for the trigger. ● Do the triggered action.
Exceptional Situations & Alternative Flows	1. Voice input might not be in the order list. ● System should warn the User about it. 2. There might be interference during communication. ● System should ask the user to move to a quieter place.
Postconditions	● System finds the intended object on the data bank.

2. 3.3. Non-Functional Requirements

Performance Requirements	Description
Response Time	Object detection will be done real time. To present satisfying results to the user our system should analyze every frame in at most 10 ms so that the user won't experience high delays. Delayed response in our application may cause real life problems for the user.
Error Handling	If an unpredictable error occurs during the usage of our system our system will log this error and send the error report to the maintainers of the application. Then as soon as possible this error will get solved or made sure that it won't occur again.
Workload	Our system will work on the user's phone, all the object detection models and applications will be installed by the user. Phone resources will be used to run object detection models.
Scalability	As the system only works on the user's phone. In the system's perspective only one user will be using it hence our system will not stop working because of low scalability problems during usage.
Application Requirements	Our application will be installed on mobile phones with object detection models as well hence at least 40mb of free space will be required. And as our application will be running object detection models a minimum of 4 GB ram is required.

2.4. References

[1] Hutmacher F. Why Is There So Much More Research on Vision Than on Any Other Sensory Modality? Front Psychol. 2019 Oct 4;10:2246. doi: 10.3389/fpsyg.2019.02246. PMID: 31636589; PMCID: PMC6787282.

[2] Chaquopy Python SDK for Android. [Online]. Available :

https://chaquo.com/chaquopy/

[3] Meet Android Studio. [Online]. Available :

https://developer.android.com/studio/intro

[4]YOLO: Real-Time Object Detection. [Online]. Available :
https://pjreddie.com/darknet/yolo/

3.SOFTWARE DESIGN DESCRIPTION (SDD)

3.1. Introduction

In this Software Design Description report we tried to explain in detail what GenEye application should include. This document will be an explanation on how the logical work will be done through explaining every layer of the explanation. We will be using Java to program the android application which is basically what the user will interact with and Python will be used to develop the computer vision needs. To develop an application for Android we will use the Android Studio IDE so that we can see the results of our program on a virtual or real android phone. For the computer vision part of our project we will train our models on a cloud based server but the trained model will run on the user's computer. To train a computer vision model we will use the YOLO algorithm because of its high accuracy, high speed during a real time object detection task and also YOLO has pre-trained object detection models which can be used for object detection tasks.

3.1.1. Purpose of This Document

This document will provide technical depth to our project and explain how we will implement features and requirements we talked about in SRS. It will provide an overview of our system. We provide different representations of our project using diagrams like class and usage expectations by providing User Interface design. As we are providing what will be in our project we also try to explain how to do these by providing technical parts as well. The document contains interfaces in our application, diagrams that explain different aspects of the application and user interactions with the system and the responses generated.

3.1.2. Scope of the Project

In this project, it is aimed to design an object and obstacle detection system to make visually impaired people's lives slightly easier. It is aimed to produce a final product that can detect objects and obstacles and notify the user using fast machine learning and image processing algorithms This project needs a mobile phone that has a camera and runs Android as an operating system.

Our project has the following features:

- Object detection
- Deciding threat level for detected object
- Informing the user
- Realtime Processing

In order to make this project successful, software should be tested and tuned if it is not working properly. Final product should be tested in a real environment during and after the development phase

3.1.3. Glossary

Term	Definition
Actor	An Actor can be an user or another software system that interacts with the main system
User	Users are visually impaired people who seeks help and uses this application
Java	Java is an Object Oriented Programming Language and computing platform first released by Sun Microsystems in 1995.
Python	Python is an interpreted, object-oriented, high-level programming language.
Software Requirement Specification (SRS)	A document that describes what the software will do and how it will be expected to perform.
Android	Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.
Android Studio	Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto.

Object Detection	Object detection is a computer vision technique for locating instances of objects in images or videos.
Computer Vision	Computer Vision is a subfield of Artificial Intelligence that makes systems see and identify objects using digital images, videos and other visual inputs.
Machine Learning	Machine Learning is a branch of Computer Science and Artificial Intelligence which tries to imitate human learning using algorithms.
Chaquopy	Chaquopy makes it possible to include and run Python scripts inside of a mobile application.

3.1.4. Overview of This Document

In our Software Design Document, we are focusing on detailed information about our project, GenEye. We divide this document into four parts: Introduction, System Design, Requirement Matrix and References. In Introduction, we provide general information about our project such as scope and motivation of GenEye. 1.1 Purpose of this Document, we gave an explanation of this document's goal. 1.2 Scope of the Project, we stated GenEye' must be achieved goals. 1.3 Glossary, we put a directory for our special terms. 1.5 Motivation, we clarified why we chose this project. In System Design, we focused on more specific details about our software system. 2.1 Architectural Design, we explained what is the context design behind GenEye and elaborated it in detailed diagrams. 2.2 User Interface Design, we explained what User would see and interact. In this part, we will be detailing GenEyes appearance. In the Requirement Matrix, we put a table which explains the connections of GenEye's system which we mentioned in SRS. In addition, we stated which purpose is satisfied with which system. Furthermore, this document will be a general guideline for programmers and technicians and development teams for understanding and future developments of GenEye.

3.1.5. Motivation

Today, technology is advancing very quickly and yet for impaired people their problems remain the same. One of the difficulties for them is having a impaired vision. To improve their vision, we used GenEye. The usage area of GenEye is to enhance their daily day by showing their required objects. Other most important part is used in their daily lives, it watches the stream and warns the visually impaired people about their surroundings. As a group that closely follows technological developments, we determined the future for these people's problems. To overcome their difficulties we develop an app program. We are aware of the fact that we will be competent engineer candidates experienced in this field when we receive the final product, and we continue our work and move forward with the goal of doing the best. The contribution we will make to our people and ourselves is our greatest source of motivation.

3.2. System Design

The system design section gives insights about how the workflow will be provided. It mainly consists of the problem's definition, the technologies that are used, user interface design, and hardware design. Additionally, it includes a number of diagrams, including class diagrams, data flow diagrams, sequence diagrams, and activity diagrams.

3.2.1. Architectural Design

To better explain our architectural design, we presented the challenge, the technologies we are planning to benefit from and many diagrams in this section. Our system will consist of two primary jobs, as we detailed in the previous sections. We will create an architecture that carries through these duties. A basic structure of GenEye is given below.

3.2.1.1. Problem Description

During our most daily activities it is certain that we rely mostly on our visual sense, in other words our eyes. Some people are born without this privilege or due to an accident or health issue they lose it. This causes their quality of life to decrease. They require other people's help for most daily activities. Recent improvements in computer vision technology is promising news for these people. This project's aim is to increase the quality of life for these people using their mobile phone and its camera and reduce the need for other people's help for basic daily activities.

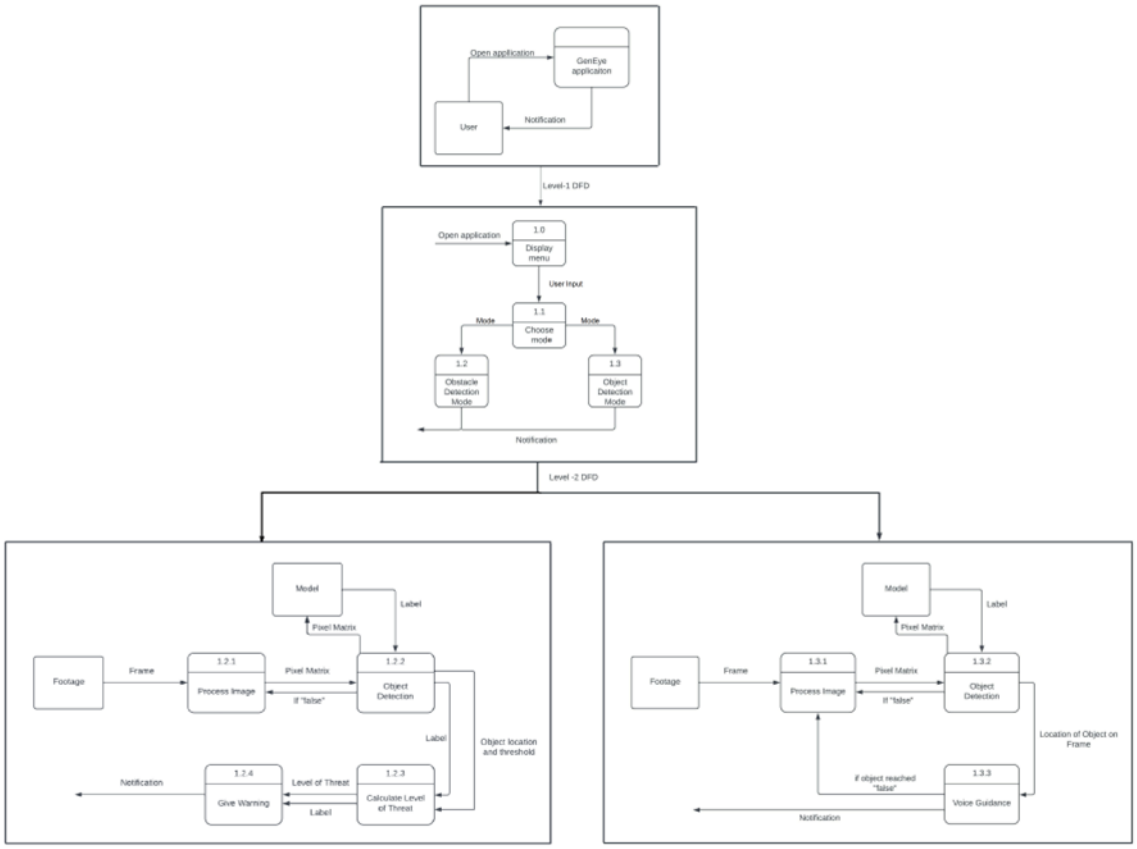
3.2.1.2. Used Technologies

There are all the hardware products we use in our system below:

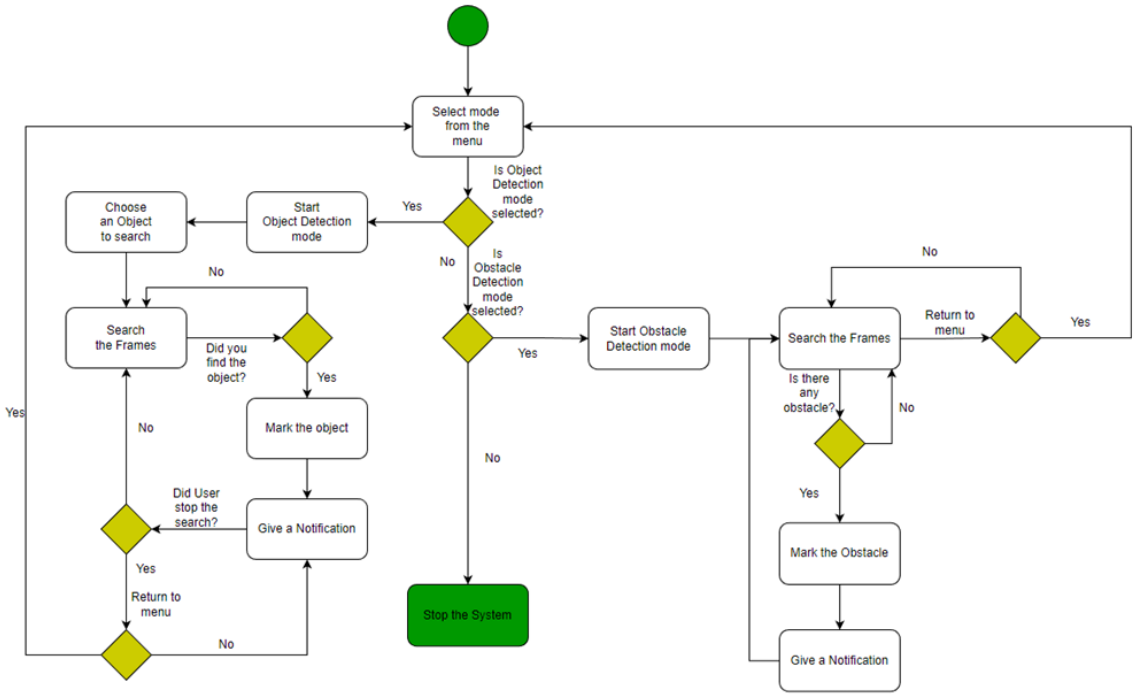
- Android Smartphone
- Android Studio[1]
- Phone Camera
- Java
- Chaquopy[2]
- YOLO Algorithm[3]
- Automatic Speech Recognition(ASR)[4]
- We use the Python programming language for many operations in our system.

We used Chaquopy on Android Studio to enable the Python libraries for YOLO and ASR. The application which will be responsible for guiding the visually impaired user and will be developed by a programming language which can combine above solutions. It can be Java or Python. In the future process of development, it will be decided.

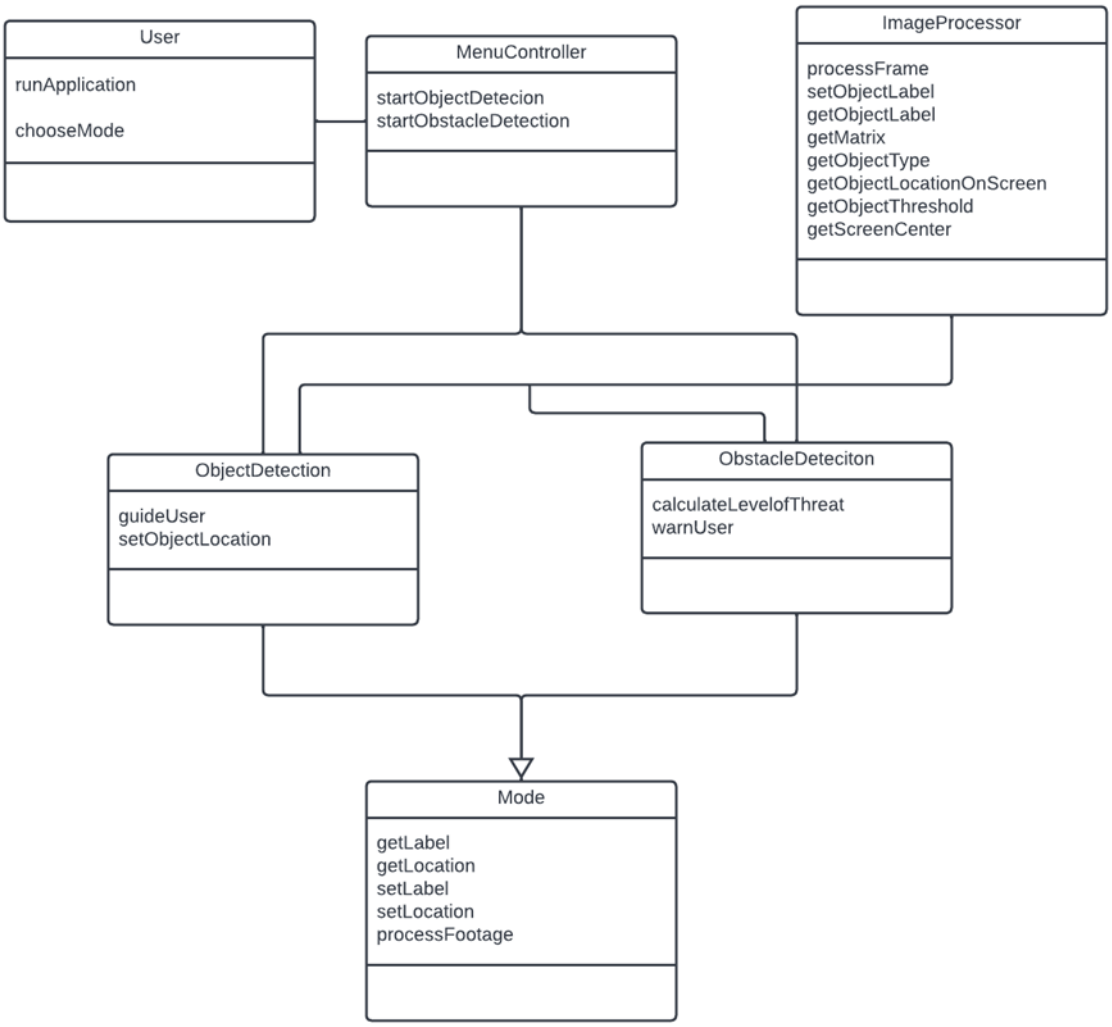
3.2.1.3. Data Flow Diagram



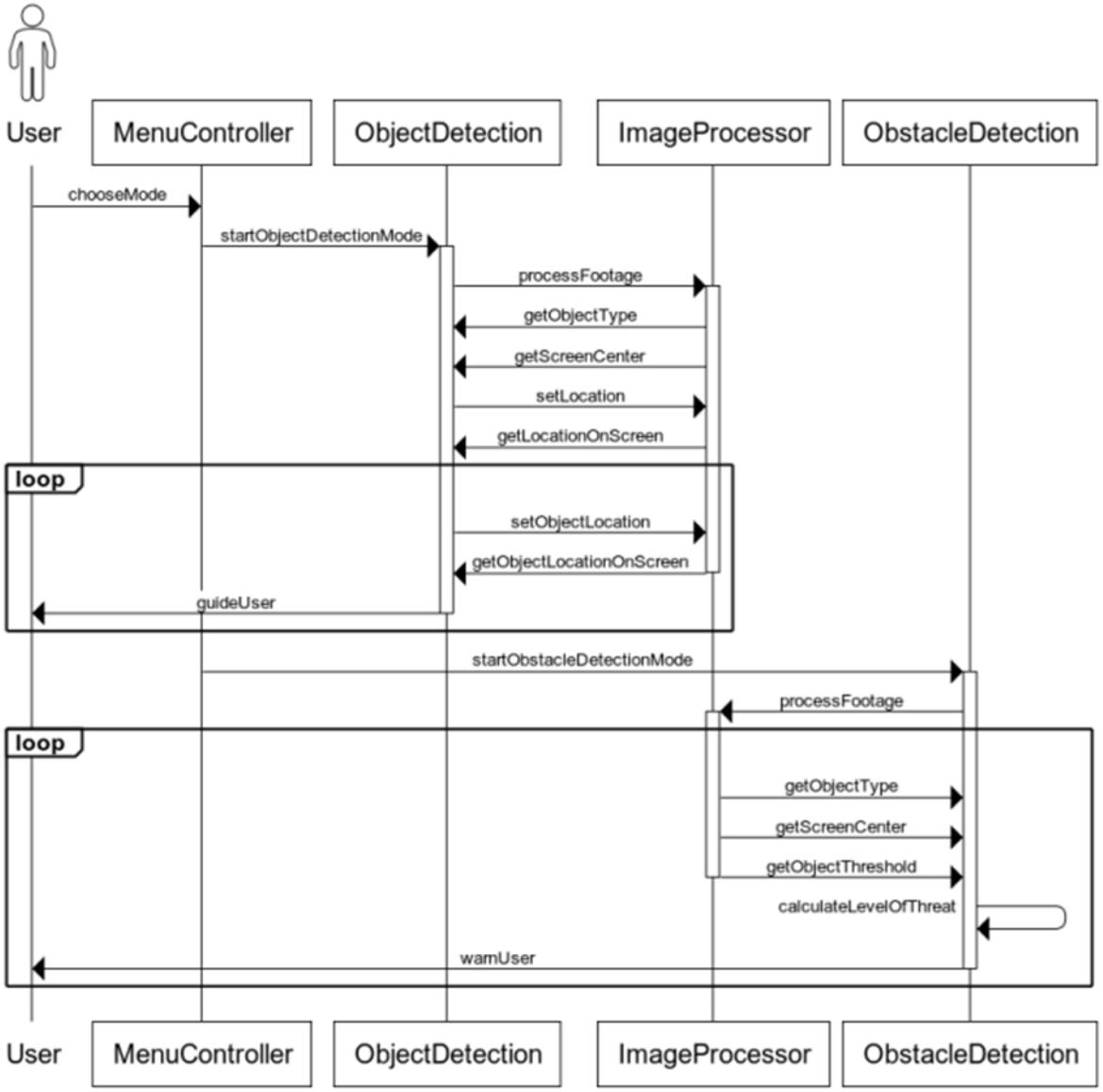
3.2.1.4. Activity Diagram



3.2.1.5. Class Diagram



3.2.1.6. Sequence Diagram



3.2.2. User Interface Design

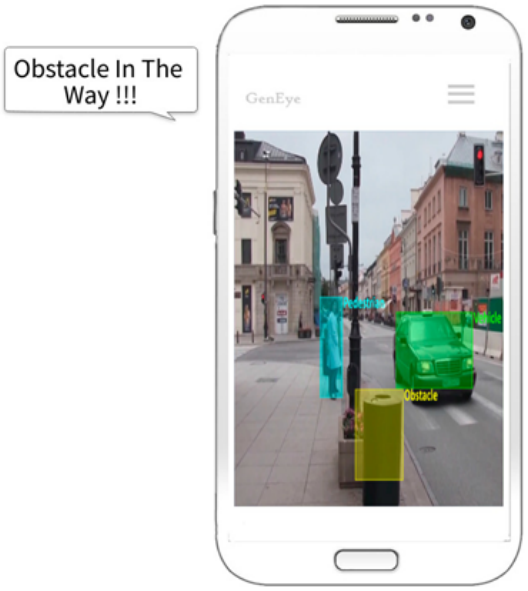
3.2.2.1 Home Page

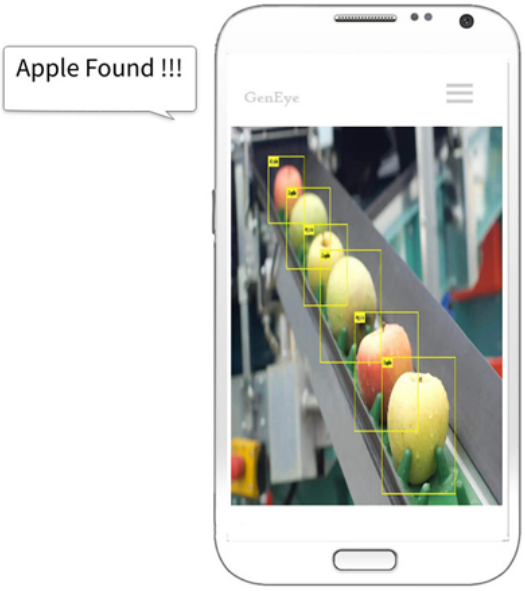
The home page will be visible to all users who have the application in their mobile

phone. As the users are considered to be visually challenged this page also contains voice warnings to ease the user's experience. If the user selects the "Obstacle Detection" mode then the voice warning will inform the user of the choice that's been made.

3.2.2.2 Vision Page

In vision pages every mode will display the environment of the user that is being captured. Also when the object that is being searched for is found or there is an obstacle in the frame a voice warning will be given to the user to inform them about the situation.





3.3. References

[1] Android Studio by Google and JetBrains

https://developer.android.com/studio

[2] Chaquopy by Chaquo Lmt.

https://chaquo.com/chaquopy/

[3] “What is YOLO Algorithm?” by Enes Zvornicanin
https://www.baeldung.com/cs/yolo-algorithm

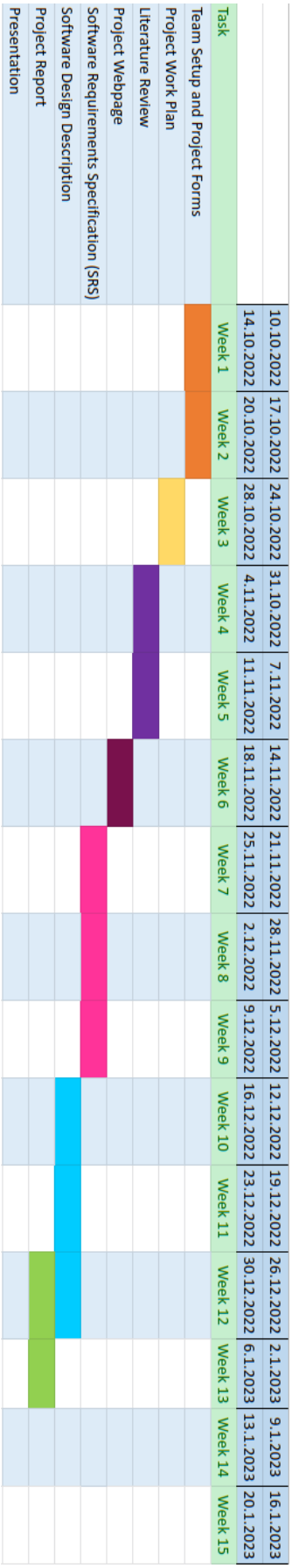
[4] ”Speech Recognition with Python” by Satish Chandra Gupta

https://www.slanglabs.in/blog/automatic-speech-recognition-in-python-programs

Conclusion

Our project will provide daily assistance to visually impaired people. System will help the user through user inputs and voice outputs to guide the user. Daily activities like walking or finding an item will be made easily with our application by the users. Any person with a mobile phone will be able to use our application with ease and as the computer vision models will be running on the user's phone no internet connection is required except for downloading or updating the application.

Work Plan


Task
Team Setup and Project Forms
Project Work Plan
Literature Review
Project Webpage
Software Requirements Specification (SRS)
Software Design Description
Project Report
Presentation
10.10.2022
14.10.2022
Week 1
17.10.2022
20.10.2022
Week 2
24.10.2022
28.10.2022
Week 3
31.10.2022
4.11.2022
Week 4
7.11.2022
11.11.2022
Week 5
14.11.2022
18.11.2022
Week 6
21.11.2022
25.11.2022
Week 7
28.11.2022
2.12.2022
Week 8
5.12.2022
9.12.2022
Week 9
12.12.2022
16.12.2022
Week 10
19.12.2022
23.12.2022
Week 11
26.12.2022
30.12.2022
Week 12
2.1.2023
6.1.2023
Week 13
9.1.2023
13.1.2023
Week 14
16.1.2023
20.1.2023
Week 15