Visual Recognition Based System To Assist Blind Persons

MSc Research Project
Cloud Computing

Ankit Dongre
Student ID: x18147054

School of Computing
National College of Ireland

Supervisor: Manuel Tova-Izquiredo

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ankit Dongre
Student ID:	x18147054
Programme:	Cloud Computing
Year:	2019
Module:	MSc Research Project
Supervisor:	Manuel Tova-Izquiredo
Submission Due Date:	12/12/2019
Project Title:	Visual Recognition Based System To Assist Blind Persons
Word Count:	6500
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	29th January 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Visual Recognition Based System To Assist Blind Persons

Ankit Dongre
x18147054

Abstract

Blind persons are devoid of vision, which plays an important role in day-to-day life. The lack of vision restricts the autonomity of a blind person to a level. Computer Vision and Machine Learning have seen rapid growth in recent years and have been used in the past for developing assistive systems[1][2][3]. In this paper, a visual recognition based assistive system is proposed to compensate for the lack of vision. This project uses a pre-trained model obtained by training COCO dataset on MobileNet network. The YOLO algorithm has been used for object detection. Experiemental results shows the proposed system performs fairly well as compared to existing applications that are available in the Google Play store. The application can handle illumination changes and transformation with negligible performance degrade. This system has its own limitations which are discussed later in the conclusion section.

1 Introduction

1.1 Motivation

Vision plays an important part in our daily life. Humans use vision for navigating through roads, to find objects, etc. However, blind persons are not blessed with Vision which makes their life quite difficult. [4][5]The latest stats from World Health Organisation (WHO) indicates that there are approximately 2.2 Billion people diagnosed with a type of vision impairment. A report by The Journal states that this number is bound to increase threefold of its current value in 2050.

1.2 Evolution In Assistive Technology

In the past few years, many assistive systems have been introduced that promise to help blind persons based on sensors, IoT and computer vision. These systems have their own advantages and limitations. For example, portable assistive systems based on embedded systems such as Raspberry Pi[1][6]. These systems make use of computer vision for the recognition of labels and text on packaged products. These systems are limited in application as they only help the blind persons in shopping in a supermarket. The power backup of these devices is also an area of concern as well as the limited processing capabilities of the device. Another method for assisting blind persons is by using mobile phones as they are more compact and have inbuilt camera.[7] proposed their method of utilising Quick Response (QR) codes for identification of objects. This system is also concerned with shopping in a store and will detect products with a QR code on it. For this system to work, there is a need to enforce labelling of every product with a QR code which is just too much of a task. [8] proposed a mobile application of assisting blind persons by identifying labels on the products. This application is based on OCR algorithm and hence inherits the disadvantages of the algorithm along with the advantages. This application fails to detect the product label in case of changes in scale, and illumination.

1.3 Current Day Scenario

[9]Wearable technology became trending around the years 2013-2014. This technology refers to devices which can be worn on body parts such as wrist or neck. Some of the popular wearable technology devices are fitbit, pebble smartwatch, and Google glasses. [2] propose one such method called as **Finger-Eye**. This system consists of an electronic wearable finger that has an embedded camera on the tips of the finger. The camera will scan the word currently under the finger and read it out for the blind user. However, the finger-eye as not successfully made completed and was tested only using a table mounted webcam.

The concept of *self driving cars* is a hot topic in technology. Studying the way these autonomous cars navigate would help to assist the navigation of a blind person. [10] propose a system for self driving cars using event cameras. Event cameras are the camera that only record events or the change in pixel intensity which makes the output of event cameras different than regular camera. These event cameras works as motion detectors to detect motion of the surrounding objects while removing redundant info.

1.4 Gaps Identified

There are a few gaps that needs to be addressed for object detection. These are:

1. There are conditions that can cause inaccurate results in object identification. One such condtion is called as *camouflage* where an object blends into the background.

2. The assistive systems available today that can be actually helpful to blind persons are made of specialized hardware. Using a specialized hardware can incur additional costs.

1.5 Research Question

This leads us to our research questions that needed to be addressed for this research:

- How to build a system that has high accuracy, speed and is robust to common object identification obstacles at a minimal cost?

- How to make the system accessible to the blind persons ?

In the next section, few popular object identification methods as well as existing best solutions for helping blind persons are surveyed. A solution is then presented based on the review with methodology, design, implementation and finally the results and its implications.

2 Related Work

In this section, we discuss, the relevant work related to assistive systems for helping blind persons and Object identification Algorithms. We discuss the offerings and drawbacks and/or limitations of the algorithms/systems.

2.1 Object Detection Algorithms

2.1.1 SIFT Algorithm

[39]Before Deep Learning took over the SIFT algorithm was one of the most popular algorithms for object detection. [66] [37] [38] [65] [64] [42]present their solutions for detection and tracking of objects in videos. SIFT algorithm is used to extract key features from the frame. Then these key features are grouped using improved k-means clustering algorithm to detect the moving object [37] have used log polar transform to stabilise the video, which makes the video frame robust to scaling and rotation[41].The results obtained from their proposed solution is then compared with SIFT-ME algorithm to compare their transformation affinity.

[65] combines SIFT and SURF algorithm to detect object resulting in a robust algorithm for object tracking. The result from the above experiments shows that SIFT has high accuracy but it is slow. The results show that their method performs well in general but suffers with common object identification obstacles such as scaling, cluttering, and illumination changes.

2.1.2 SURF

SIFT's accuracy was high and it was able to handle scaling. But the downfall was the algorithms speed i.e. it was very slow. In order to increase the speed of object identification a new algorithm **SURF** or Speeded Up Robust Features[40].

[63] [45] [43] [61] [60] [47] [46] [75] [76]performed their studies on the SURF algorithm. These studies includes SURF performance enhancements and comparative studies with respect to algorithms such as SIFT, ORB, BRIEF etc. The enhancements aim to make the SURF algorithm illumination invariant, and to attain higher matching rates. The findings from their experiment are presented as follows:

- Combining SURF with the best bin first algorithm results in faster matching. However, it cannot increase matching accuracy
- More feature points can be detected if geometric algebra is introduced in SURF
- The descriptor dimension can be reduced by dividing the sampling points into 2 groups
- Illumination invariance can be achieved by deriving descriptors from the intensity order pattern of an image
- SURF outperforms SIFT in case of noisy images
- SURF requires less computational power than SIFT

2.1.3 OCR

OCR is another algorithm which has been widely used for object identification. It has been mainly used for detecting text from an image. The following works include the usage of OCR algorithm for object identification:

[8][54][2][55][58]discuss the use of the OCR algorithm for object detection. These studies consists of various experiments siuch as using the OCR algorithm for food label detection, expiry date detection, ID card information retrieval. The results from all these experiments can be summarized as follows:

- OCR performs good in case of reading texts with accuracy as high as 70-90%
- OCR suffers with varying light and rotation
- reflective surface also pose a problem to the OCR algorithm
- OCR performs accurate retrieval of information
- There is a performance degradation in case of face detection

[51] in their paper implement their proposed approach for elevator button detection. This study aims to help service robots in navigating to their desired floor through button detection. The authors have combined OCR and Faster R-CNN into a single neural network called OCR-RCNN. They have generated a custom dataset of elevator panels. The experimental results show good performance of their system even on untrained elevator panel images.

2.1.4 YOLO

YOLO or You Only Look Once algorithm has been popular since its release. YOLO as its name suggests only needs to look once for object detection. Yolo is a single shot detector, which means that it performs the localization and Image classification in only 1 step instead of doing it in two different steps. This speeds up the overall process of object recognition but introduces a minor degradation in accuracy[11]. Over the years there has been 2 major upgradations to YOLO algorithm: Yolo v2 and v3 and while the original algorithm is called as Yolo v1.

[22][24][50][74][12][57] use YOLO for object detection and comparison against Faster-RCNN. These systems use a pre-trained model that has been obtained by training a dataset on a deep neural network. The results from their experiments are as follows:

- The results show that YOLO v3 is faster than its predecessors and is more accurate
- YOLO and Faster-RCNN have comparative accuracy. However, Yolo v3 is better than Faster R-CNN in case of recall rate
- Combining Spatial pyramid with YOLO can increase the mean Average Precision(mAP) of the algorithm
- Increasing the scale of the detection can help with detecting smaller objects
- YOLO3 inherits few problems from the old YOLO v1 such as the localization issues when multiple small objects are near to each other

2.2 R-CNN

[31] in their paper propose MaskLab to solve the problem of Instance Segmentation. The MaskLab model is built on top of Fast-RCNN and feature extraction is performed using ResNet-101[36]. MaskLab has 3 main features: box detection, semantic segmentation, & direction estimation. They have used the COCO instance segmentation benchmark to evaluate the effectiveness of MaskLab. MaskLab shows some promising results in the experimental evaluation.

[32][49][53][30] in their studies use RCNN for object detection. These studies include proposing an improved version to introduce illumination aware, optimize regional proposals etc. [30]built their system using a model called ResNet-101 with a FPN or Feature Pyramid N/W [35][48] The results of the above experiments can be summarized as follows:

- The experiment results show that the algorithm has been able to learn the association between different objects, but it is unclear what information is learned by the relation module[32]
- Optimizing the regional proposal creation proves to introduce efficient iterative refinements and performs better than other RPN algorithms with WIDER, AFW and Pascal Faces
- Adding illumination aware network to Faster R-CNN improves its performance as compated to regular Faster R-CNN for pedestrian detection
- taking large dataset and simple heuristics can improve the performance of the algorithm

2.3 Existing solutions for assisting blind persons

[3] in their paper propose a vision based system to help visually impaired persons navigate through their surroundings. This system consists of a camera, a haptic feedback device to aware blind persons of an object, and an embedded computer. This system is wearable by the blind users thus providing mobility to users. Regional proposals are used to find empty spaces through the navigation path. The authors test their system by making a blind person walk through a maze wearing their system. The experiment results show that this system greatly helps blind user to navigate through a path without collisions. However, navigation using this system is slower as compared to using cane. If this system is combined with cane for blind user, the speed of navigation can be improved.

[2] in their paper propose an assistive system to help blind persons read using the OCR algorithm. This system is in the form of a glove with camera embedded index that can be worn by the blind user. The blind user after wearing the glove will place their index finger over the first sentence from left to right. The camera underneath the finger will process the image of the text and give output in the form of audio. The experiment results show that the finger can successfully read the text under the camera, but the process is slow as the algorithm uses multiple images of the same frame to create a high resolution image. The feasibility of the project is also a matter consideration as the experiments were conducted on a webcam mounted on a table for reading the text. The actual result may vary if finger tip camera is used.

[1] [25] propose an assistive system to help blind users. Their system makes use of Raspberry Pi, which is an embedded computer for processing text present in the images. While [1] uses the OCR algorithm, [25] uses YOLO v1. Text To Speech (TTS) library is used to read out the text to the blind user. The camera is placed on the glasses of the user and is connected to Raspberry Pi. The experiment results show that the OCR algorithm is good in reading text but suffers from poor performance in face detection. YOLO can attain accuracy around 83% with good detection speed.

Algorithm	Authors	Advantages	Limitations
SIFT	[37] [38] [42] [47] [64] [66] [65]	High accuracy, invariance to scaling	Requires high computational power, low speed, performs bad against occlusion, blurring, & change of illumination, patented
SURF	[43] [45] [46] [44] [60] [61] [63] [46]	High speed, robust to transformations, low computation power required	Lower accuracy than SIFT, detects fewer keypoints than other algorithm, patented
OCR	[8] [2] [59] [58] [55] [54] [51]	Faster and accurate in recognising texts	Low accuracy in detecting images, Not resilient to illumination
Regional Proposal Networks	[30] [31] [56] [53] [52] [49] [32]	Fast, ability to handle transformations, good in dealing with illumination changes	Time required to train the model is quite high, It is a 2 stage process making it slower than SSDs
Single Shot Detectors (YOLO)	[12] [22] [24] [25] [57] [29] [50]	**It performs all the computations in 1 step: making it fastest among all the algorithms**, robust to transformations, Performs well in case of occlusion and illumination changes	Lower accuracy than SIFT

3 Methodology

After reviewing various literature, we have decided that implementing the system by using mobile application will provide blind persons mobility and ease of use. In our system, the blind person will hold the mobile phone in their hand while naivgating through the streets while the application is running. The application will identify the objects present in the frame and announce the name of the object detected in the form of audio through headphones worn by the blind user. The audio announcement consists of the name of the object as well the position. The blind user can then use this information to navigate.

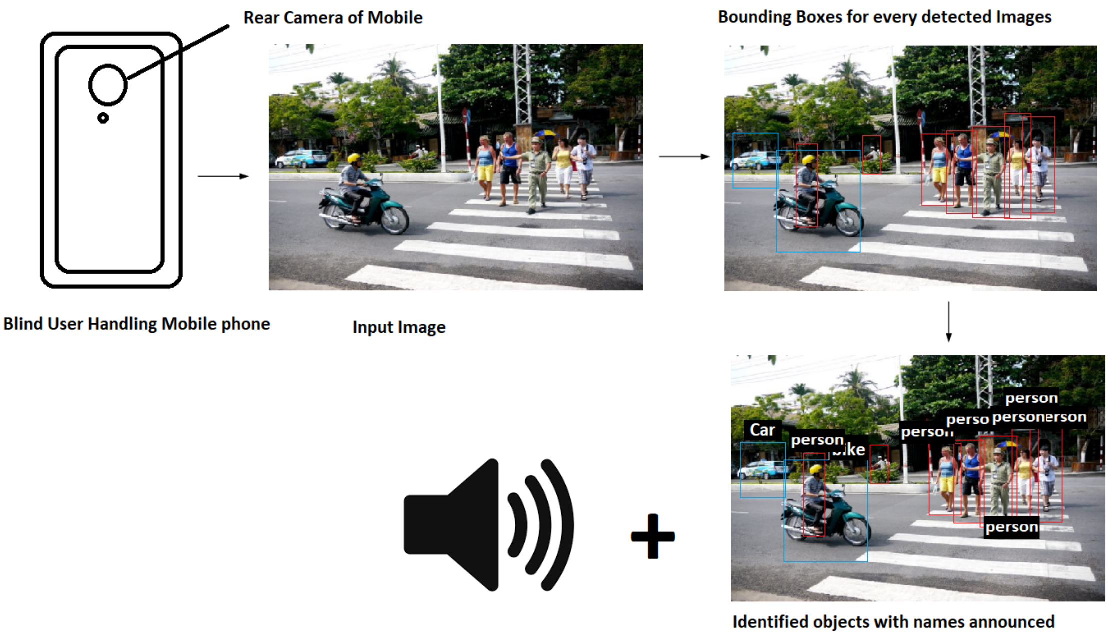

Figure 1: Working of the System

3.1 Data Selection and PreProcessing

COCO dataset has been used for matching which is collected from[70]. COCO dataset has the following features:

- More than 330K images with around 200K labeled images
- Around 80 categories of object
- More than 1 million object instances

The COCO dataset is trained on mobileNet Convolutional Network which is specialised for training computer vision models. The latest version of MobileNet i.e. MobileNet v2 is used. MobileNet v2 performs much better than its predecessor in every aspect. It is about 35% faster than MobileNet v1 with same accuracy.

4 Design Specification

After an intensive review of the latest algorithms and developments in computer vision and machine learning, we have decide to use YOLO algorithm. YOLO is chosen as it is the fastest among all the algorithm and provides robust object detection capabilities [22] [24] [50] [74] [12] [57]. The successive improvements since YOLO v1 has led to the developments of a new better algorithm, called Yolo V3. YOLO V3 overcomes the limitations of YOLO V1 and V2. and provides better results in terms of robustness and finding distinct objects within a given image. MobileNet Network has been used to train the model. COCO dataset is used to store images and pre-trained weights. We are using Tensorflow to implement the YOLO algorithm[77].

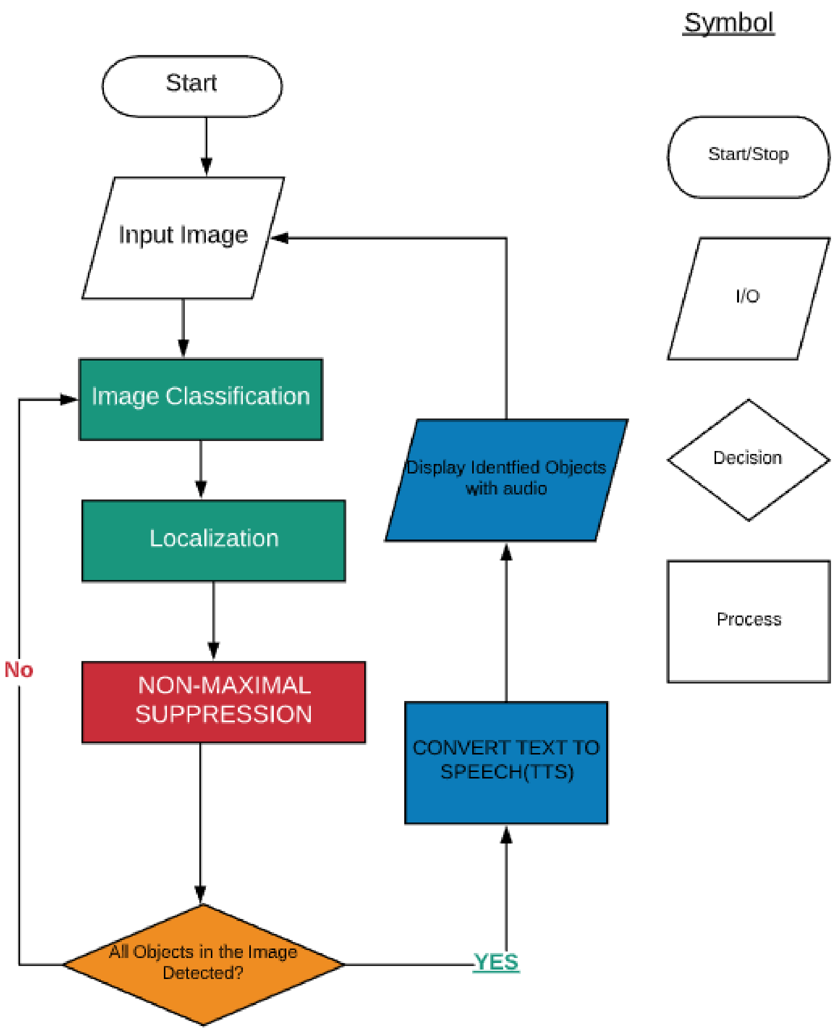

Figure 2: Improvised Algorithm

4.1 Steps in YOLO

After opening the application, the rear camera will capture images in front of the blind person in realtime. Then we apply YOLO algorithm to identify the object and announce it. 2 shows the steps involved in the object identifcation performed by YOLO.

4.1.1 Image Classification

Image classification refers to predicting the object present in an image. For example, consider an image classifier for fruits. We can have several different fruits such as apple,Papaya, guava, Kiwi, Dragon Fruit etc. Performing classification on any given image will give us a value depicting the likelihood of a fruit being present in th given image.

4.1.2 Localization

After we have classifed the image, the next thing for us to do is to find the location where these objects could be found in the image. This process is known as *Localization*.

YOLO segments the given input image into a grid of size NxN. Every grid on the input image plays a part in the detection of the object. These grid are referred to as regions. Afterwards, these regions predicts the number of anchor boxes for an object[68]. An achor box defines the boundaries of any detected objects. Repeating Localization along with image classification will give us all the object present in any image.

4.1.3 Non-Maximal Supression

Any object identification algorithm tends to detect the same object for multiple times. YOLO solves this problem by using a process called as Non-Maximal Suppression. It performs this operation by calculating the confidence value of every bounding box and then removing the ones with the low scores while keeping the ones with highest confidence values. Repeating these steps guarantee that one object is only identified once.

4.2 Class Diagram

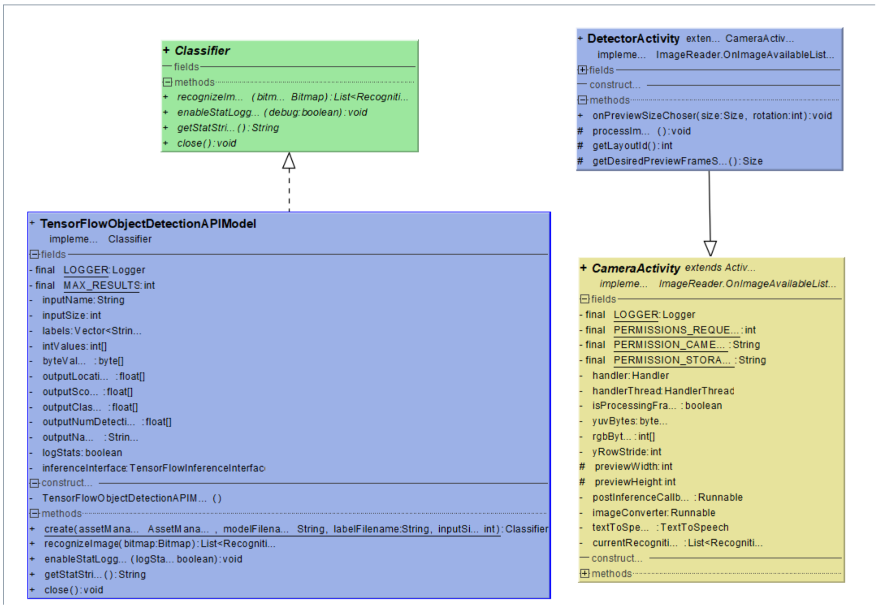

Figure 3: Class Diagram of the application

5 Implementation

5.1 Tools and Languages Used

Android platform has been chosen as it owns around 75% of the global market share of smartphone Operating Systems. The target is Smartphones with version of Android Loliipop or above as they have higher adoption than older versions as well as older versions run at a risk of being dropped out of support. We are using XML and Java for developing the application.

5.2 Models Developed

The app will identify the objects from the image taken via the camera. The output will be all the identified objects in the image with a label presenting the class of object and the confidence as well as an audio notification of their name and position. Fig. 4 shows the outputs of the application. To account for the accessibility, the application will automatically run if headphone jack is inserted which is accustomed to the needs of a blind person for opening the application.

(a) Bottle

(b) Laptop and Mouse

Figure 4: Output(s)

5.2.1 Code Developed

The algorithm takes input then performs image classification using a pre-trained COCO dataset which is trained using MobileNet-v1 Neural Network and localization iteratively to retrieve all the objects present inside the input image. YOLO has 24 convolution layers whose output is fed to 2 Fully connected layers[68].After applying YOLO, a confidence score is calculated. This score indicates the probability of the class of objects present in a given image. Then the class with the highest confidence is chosen and rest others are dumped. The output class is converted to speech which is obtained by using text to speech API. The fig. 5 shows the implementation of text to speech of the objects detected.

6 Evaluation

We define the following performance metrics to evaluate the result:

- Accuracy: Accuracy determines the correctness of the predicted class. It is the percentage of the total correct predicted classes to total trials.
- Confidence: Confidence is the likeliness of the predicted object to the actual object.
- Execution Time: The time taken for the algorithm to detect subsequent objects. The lesser the better.

The evaluation is done by operating the application to detect object under different conditions such as change of lighting, scaling, rotation of the object, camera angles etc. We

```java
private void speak() {

    final double rightStart = previewWidth / 2 - 0.10 * previewWidth;
    final double rightFinish = previewWidth;
    final double letStart = 0;
    final double leftFinish = previewWidth / 2 + 0.10 * previewWidth;
    final double previewArea = previewWidth * previewHeight;

    StringBuilder stringBuilder = new StringBuilder();

    for (int i = 0; i < currentRecognitions.size(); i++) {
        Classifier.Recognition recognition = currentRecognitions.get(i);
        stringBuilder.append(recognition.getTitle());

        float start = recognition.getLocation().top;
        float end = recognition.getLocation().bottom;
        double objArea = recognition.getLocation().width() * recognition.getLocation().height();

        if (objArea > previewArea / 2) {
            stringBuilder.append(" in front of you ");
        } else {


            if (start > letStart && end < leftFinish) {
                stringBuilder.append(" on the left ");
            } else if (start > rightStart && end < rightFinish) {
                stringBuilder.append(" on the right ");
            } else {
                stringBuilder.append(" in front of you ");
            }
        }
```

Figure 5: Text To Speek(TTS) Functionality

have used Logger to log the time of object detectionThese results are compared against the results from the app *Object Detection*[73]. Table column meaning

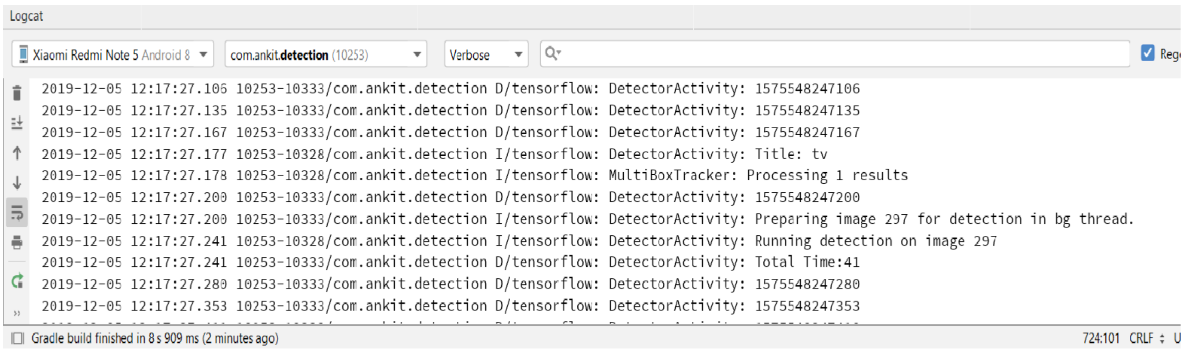

Figure 6: Time in System Logs

Actual: Refers to the actual object present in the image
Detected$_1$: Objected predicted by the proposed application **Detected$_2$**:Objected predicted by the object detection application
CV$_1$: The confidence value calculated by proposed solution **CV$_2$**: The confidence value calculated by object detection application
T$_1$: Time taken by the proposed solution for detection **T$_2$**: Time taken by object detection application

6.1 Experiment 1: Good lighting Conditions

Table 2 represents the observations under good lighting and medium distance between the camera and the object. The images were taken in a room with bright lighting.

Table 1: Output(s) of Object Detection App[73]

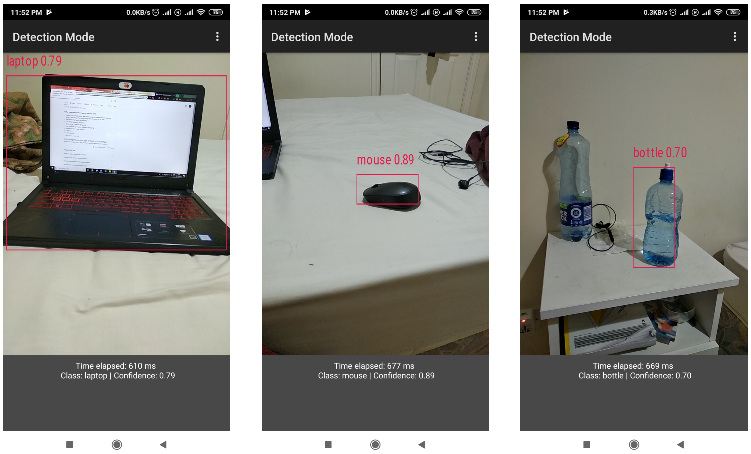

Table 2: Observation 1

Actual	$Detected_1$	CV_1	T_1	$Detected_2$	CV_2	T_2
Mouse	Mouse	0.95	23mS	Mouse	0.82	420mS
Bottle	Bottle	0.89	25mS	Bottle	0.89	120mS
Cellphone	Cellphone	0.72	30mS	Cellphone	0.98	107mS
Chair	Chair	0.94	42mS	Chair	0.65	200mS
Cup	Cup	0.96	20mS	Cup	0.63	356mS

6.2 Experiment 2: Low Light Conditions

Table 3 represents the observations in low light. The images were taken in a room with dim lighting.

Table 3: Observation 2

Actual	$Detected_1$	CV_1	T_1	$Detected_2$	CV_2	T_2
Mouse	Mouse	0.59	25mS	Mouse	0.80	677mS
Bottle	Bottle	0.89	25mS	Bottle	0.56	560mS
Cellphone	Cellphone	0.77	40mS	Cellphone	0.78	190mS
Chair	Chair	0.67	42mS	Chair	0.53	200mS
Cup	Cup	0.67	24mS	Cup	0.51	300mS

6.3 Experiment3: Objects from top

Table 4 represents the observations of object from top.

6.4 Experiment4: Camouflaged Objects

Table 5 represents the observations of object in case where they are kept with other objects of same colour such that they blend in with other objects.

Table 4: Observation 3

Actual	$Detected_1$	CV_1	T_1	$Detected_2$	CV_2	T_2
Mouse	Mouse	0.70	25mS	Mouse	0.87	413mS
Bottle	Bottle	0.65	30mS	Bottle	0.57	780mS
Cellphone	Cellphone	0.61	19mS	Cellphone	0.78	670mS
Chair	Chair	0.57	50mS	Chair	0.69	456mS
Cup	Cup	0.76	30mS	Cup	0.89	120mS

Table 5: Observation 4

Actual	$Detected_1$	CV_1	T_1	$Detected_2$	CV_2	T_2
Mouse	Mouse	0.70	25mS	Mouse	0.87	451mS
Bottle	Bottle	0.65	30mS	Bottle	0.57	490mS
Cellphone	Cellphone	0.51	34mS	Cellphone	0.69	567mS
Chair	??(Unidentified)	NA	50mS	Chair	0.76	700mS
Cup	Cup	0.70	29mS	Cup	0.78	466mS

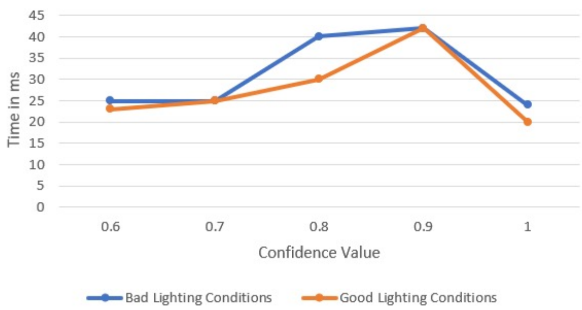

Figure 7: Graph of CV vs time in good and bad lighting conditions

6.5 Discussion

From the observations, we can see that our application performs quite well in every condition. Although, there are minor performance degradations in terms of speed and the confidence level but when compared to Object Detection app, the proposed app is faster and faster in terms of detection and is robust to various object identification obstacles. Fig. 7 shows plot of CV against time in good and bad lighting conditions. Observing the graph, we can say that the time required in case of well-lit condition is less and the object recognition is done with higher confidence as opposed to bad lighting conditions. This indicates that the illumination affects the ability of the algorithm, but the effect is not so severe. Android smartphone has been chosen to deploy the application as most people are using android phones and to use this application they need not buy any other special hardware in as in case of [1][25][3] so as to limit the cost that could incur for running the application. This answers our first research question 1.5. Our system is implemented in such a way that provides accessibility to the user of the application.

To open the application, the blind person has to insert headphone jack into the mobile phone, this will start the application. As soon as the application starts, the camera detects and announces the name of the objects without needing any intervention from the blind person. This answers our second research question 1.5. As for now this application can be used to detect around 100 objects within 20 classes. In the future, the model can be enhanced to detected much more objects. Meanwhile, the Object Detection application can detect around 80 classes of objects. Another concern for the working of the app is related to optical illusions and dimension of the object. For example, if a blind person is walking through a street and at the end of the street there is a wall with photo of a dog. The app is more likely to detect it as a dog. So, it is essential to know if we are identifying the actual object or a picture of that object.

7 Conclusion and Future Work

This paper proposes a computer vision-based system that assists blind persons in navigation as well as tries to answers the questions raised in the Introduction section. The proposed system is tested for different conditions and against similar applications present on the Appstore. It is evident from the observations that the application performs fairly well under different circumstances and is faster and efficient than its alternatives. The application is also fairly accessible to blind users as it allows the blind user to open the application by inserting earphone jack into the mobile phones which is missing in many other applications. In the current state of application, this application can be used for navigation but has a few limitations. There are still many areas that are a concern for improvement.

- As of now this application can detect around 100 objects and about 20 classes of objects, this could be improved to encompass more classes of objects
- The application can be improved by merging it with Internet of Devices technology devices to account for better object detection
- With enhancements in deep learning and YOLO algorithm, the application will see better performance in the future
- The application is limited to android phones and needs redesigning be able to run on cross platform in the future

References

[1] M. Rajesh, B. K. Rajan, A. Roy, K. A. Thomas, A. Thomas, T. B. Tharakan, and C. Dinesh, "Text recognition and face detection aid for visually impaired person using Raspberry PI," in *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*. Kollam, India: IEEE, Apr. 2017, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/8074355/

[2] Z. Liu, Y. Luo, J. Cordero, N. Zhao, and Y. Shen, "Finger-eye: A wearable text reading assistive system for the blind and visually impaired," in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. Angkor Wat, Cambodia: IEEE, Jun. 2016, pp. 123–128. [Online]. Available: http://ieeexplore.ieee.org/document/7784012/

[3] H.-C. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarre, and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, May 2017, pp. 6533–6540. [Online]. Available: http://ieeexplore.ieee.org/document/7989772/

[4] "Vision impairment and blindness." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[5] Afp, "The number of blind people in the world is set to triple by 2050." [Online]. Available: https://www.thejournal.ie/blindness-triple-2050-3527437-Aug2017/

[6] C. Yi, Y. Tian, and A. Arditi, "Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 808–817, Jun. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6517218/

[7] H. S. Al-Khalifa, "Utilizing QR Code and Mobile Phones for Blinds and Visually Impaired People," in *Computers Helping People with Special Needs*, K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 5105, pp. 1065–1069. [Online]. Available: http://link.springer.com/10.1007/978-3-540-70540-6_159

[8] S. Deshpande and R. Shriram, "Real time text detection and recognition on hand held objects to assist blind people," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. Pune, India: IEEE, Sep. 2016, pp. 1020–1024. [Online]. Available: http://ieeexplore.ieee.org/document/7877741/

[9] "Wearable technology," Nov 2019. [Online]. Available: https://en.wikipedia.org/wiki/Wearable_technology

[10] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 5419–5427.

[11] J. Hui, "What do we learn from single shot object detectors (ssd, yolov3), fpn & focal loss (retinanet)?" *Medium [Online]. Available: [Accessed:*, vol. 28, September 2019. [Online]. Available: https://medium.com/@jonathan_hui/what-do-we-learn-from-single-shot-object-detectors-ssd-yolo-fpn-focal-loss-3888677c5f4d

[12] C.-Y. Cao, J.-C. Zheng, Y.-Q. Huang, J. Liu, and C.-F. Yang, "Investigation of a Promoted You Only Look Once Algorithm and Its Application in Traffic Flow Monitoring," *Applied Sciences*, vol. 9, no. 17, p. 3619, 2019.

[13] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," *arXiv:1405.0312 [cs]*, 2014.

[14] A. Beloglazov and R. Buyya, "Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 5, pp. 1310–1333, 2015.

[15] G. Feng and R. Buyya, "Maximum revenue-oriented resource allocation in cloud," *IJGUC*, vol. 7, no. 1, pp. 12–21, 2016.

[16] D. G. Gomes, R. N. Calheiros, and R. Tolosana-Calasanz, "Introduction to the special issue on cloud computing: Recent developments and challenging issues," *Computers & Electrical Engineering*, vol. 42, pp. 31–32, 2015.

[17] R. Kune, P. Konugurthi, A. Agarwal, C. R. Rao, and R. Buyya, "The anatomy of big data computing," *Softw., Pract. Exper.*, vol. 46, no. 1, pp. 79–105, 2016.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv:1506.02640 [cs]*, 2015.

[19] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv:1612.08242 [cs]*, 2016.

[20] A. Wong, M. J. Shafiee, F. Li, and B. Chwyl, "Tiny SSD: A Tiny Single-shot Detection Deep Convolutional Neural Network for Real-time Embedded Object Detection," *arXiv:1802.06488 [cs]*, 2018.

[21] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Freebies for Training Object Detection Neural Networks," *arXiv:1902.04103 [cs]*, 2019.

[22] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3." IEEE, 2019, pp. 1–6.

[23] M. B. Jensen, K. Nasrollahi, and T. B. Moeslund, "Evaluating State-of-the-Art Object Detector on Challenging Traffic Light Data," 2017, pp. 882–888.

[24] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, "Performance Enhancement of YOLOv3 by Adding Prediction Layers with Spatial Pyramid Pooling for Vehicle Detection," 2018, pp. 1–6.

[25] P. Maolanon and K. Sukvichai, "Development of a Wearable Household Objects Finder and Localizer Device Using CNNs on Raspberry Pi 3," 2018, pp. 25–28.

[26] H. Nakahara, H. Yonekawa, T. Fujii, and S. Sato, "A Lightweight YOLOv2: A Binarized CNN with A Parallel Support Vector Regression for an FPGA," 2018, pp. 31–40.

[27] X. Zhang, W. Yang, X. Tang, and J. Liu, "A Fast Learning Method for Accurate and Robust Lane Detection Using Two-Stage Feature Extraction with YOLO v3," *Sensors*, vol. 18, no. 12, p. 4308, 2018.

[28] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, 2019.

[29] R. Widyastuti and C.-K. Yang, "Cat's Nose Recognition Using You Only Look Once (Yolo) and Scale-Invariant Feature Transform (SIFT)," 2018, pp. 55–56.

[30] P. Ammirato and A. C. Berg, "A Mask-RCNN Baseline for Probabilistic Object Detection," *arXiv:1908.03621 [cs]*, 2019.

[31] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "MaskLab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features," 2018, pp. 4013–4022.

[32] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation Networks for Object Detection," p. 10, 2017.

[33] W. Ouyang, X. Wang, X. Zeng, Shi Qiu, P. Luo, Y. Tian, H. Li, Shuo Yang, Zhe Wang, Chen-Change Loy, and X. Tang, "DeepID-Net: Deformable deep convolutional neural networks for object detection," 2015, pp. 2403–2412.

[34] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, "Weakly Supervised Region Proposal Network and Object Detection," 2018, vol. 11215, pp. 370–386.

[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[37] M. Sharif, S. Khan, T. Saba, M. Raza, and A. Rehman, "Improved video stabilization using sift-log polar technique for unmanned aerial vehicles," in *2019 International Conference on Computer and Information Sciences (ICCIS)*, April 2019, pp. 1–7.

[38] P. Deekshitha, B. T. Reddy, S. Badadha, K. Dhruthi, and S. Pandiaraj, "Object motion perception and tracking using sift with k-means clustering," *Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org*, vol. 8, no. 4, 2018.

[39] "Scale-invariant feature transform," Aug 2019. [Online]. Available: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

[40] "Speeded up robust features," Sep 2019. [Online]. Available: https://en.wikipedia.org/wiki/Speeded_up_robust_features

[41] H. Araujo and J. Dias, "An introduction to the log-polar mapping," *Proc. of Second Workshop on Cybernetic Vision*, 01 1996.

[42] L. Zheng, Y. Yang, and Q. Tian, "SIFT Meets CNN: A Decade Survey of Instance Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, May 2018. [Online]. Available: https://ieeexplore.ieee.org/document/7935507/

[43] J. Xingteng, W. Xuan, and D. Zhe, "Image matching method based on improved SURF algorithm," in *2015 IEEE International Conference on Computer and Communications (ICCC)*. Chengdu, China: IEEE, Oct. 2015, pp. 142–145. [Online]. Available: http://ieeexplore.ieee.org/document/7387556/

[44] Y. Kim and H. Jung, "Reconfigurable hardware architecture for faster descriptor extraction in SURF," *Electronics Letters*, vol. 54, no. 4, pp. 210–212, Feb. 2018. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/el.2017.3133

[45] Z. X. Geng and Y. Q. Qiao, "An Improved Illumination Invariant SURF Image Feature Descriptor," in *2017 International Conference on Virtual Reality and Visualization (ICVRV)*. Zhengzhou, China: IEEE, Oct. 2017, pp. 389–390. [Online]. Available: https://ieeexplore.ieee.org/document/8719146/

[46] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. Sukkur: IEEE, Mar. 2018, pp. 1–10. [Online]. Available: https://ieeexplore.ieee.org/document/8346440/

[47] E. Karami, S. Prasad, and M. Shehata, "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images," p. 6, 2015.

[48] C. Shorten, "Introduction to resnets," May 2019. [Online]. Available: https://towardsdatascience.com/introduction-to-resnets-c0a830a288a4

[49] M. Najibi, B. Singh, and L. S. Davis, "Fa-rpn: Floating region proposals for face detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[50] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767 [cs]*, Apr. 2018, arXiv: 1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767

[51] D. Zhu, T. Li, D. Ho, and T. Zhou, "A Novel OCR-RCNN for Elevator Button Recognition," p. 6, 2018.

[52] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain Adaptive Faster R-CNN for Object Detection in the Wild," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 3339–3348. [Online]. Available: https://ieeexplore.ieee.org/document/8578450/

[53] C. Li, D. Song, R. Tong, and M. Tang, "Illumination-aware faster R-CNN for robust multispectral pedestrian detection," *Pattern Recognition*, vol. 85, pp. 161–171, Jan. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0031320318303030

[54] Kento, R. H. Wijaya, T. D. Linh, H. Seya, M. Arai, T. Maekawa, and K. Mizutani, "Recognition of Expiration Dates Written on Food Packages with Open Source OCR," *International Journal of Computer Theory and Engineering*, vol. 10, no. 5, pp. 170–174, 2018. [Online]. Available: http://www.ijcte.org/index.php?m=content&c=index&a=show&catid=98&id=1483

[55] H. D. Liem, N. D. Minh, N. B. Trung, H. T. Duc, P. H. Hiep, D. V. Dung, and D. H. Vu, "FVI: An End-to-end Vietnamese Identification Card Detection and Recognition in Images," in *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*. Ho Chi Minh City: IEEE, Nov. 2018, pp. 338–340. [Online]. Available: https://ieeexplore.ieee.org/document/8606831/

[56] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng, "Learning to Detect Human-Object Interactions," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Lake Tahoe, NV: IEEE, Mar. 2018, pp. 381–389. [Online]. Available: https://ieeexplore.ieee.org/document/8354152/

[57] J. Park, Y.-S. Yun, S. Eun, S. Cha, S.-S. So, and J. Jung, "Deep neural networks based user interface detection for mobile applications using symbol marker," in *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems - RACS '18*. Honolulu, Hawaii: ACM Press, 2018, pp. 66–67. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3264746.3264808

[58] J. E. M. Adriano, K. A. S. Calma, N. T. Lopez, J. A. Parado, L. W. Rabago, and J. M. Cabardo, "Digital conversion model for hand-filled forms using optical character recognition (OCR)," *IOP Conference Series: Materials Science and Engineering*, vol. 482, p. 012049, Mar. 2019. [Online]. Available: http://stacks.iop.org/1757-899X/482/i=1/a=012049?key=crossref.9d0e5bd7b4373783419e54b24b49135b

[59] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large Scale System for Text Detection and Recognition in Images," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*. London, United Kingdom: ACM Press, 2018, pp. 71–79. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3219819.3219861

[60] Z. Bahrami and F. Akhlaghian Tab, "A new robust video watermarking algorithm based on SURF features and block classification," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 327–345, Jan. 2018. [Online]. Available: http://link.springer.com/10.1007/s11042-016-4226-0

[61] P. Ding, F. Wang, D. Gu, H. Zhou, Q. Gao, and X. Xiang, "Research on Optimization of SURF Algorithm Based on Embedded CUDA Platform," in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. Tianjin, China: IEEE, Jul. 2018, pp. 1351–1355. [Online]. Available: https://ieeexplore.ieee.org/document/8688294/

[62] S. J. Chen, S. Z. Zheng, Z. G. Xu, C. C. Guo, and X. L. Ma, "AN IMPROVED IMAGE MATCHING METHOD BASED ON SURF ALGORITHM," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-3, pp. 179–184, Apr. 2018. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-3/179/2018/

[63] R. Wang, Y. Shi, and W. Cao, "GA-SURF: A new Speeded-Up robust feature extraction algorithm for multispectral images based on geometric algebra," *Pattern Recognition Letters*, p. S0167865518308730, Nov. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167865518308730

[64] H. Jabnoun, F. Benzarti, and H. Amiri, "Object detection and identification for blind people in video scene," in *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*. Marrakech: IEEE, Dec. 2015, pp. 363–367. [Online]. Available: http://ieeexplore.ieee.org/document/7489256/

[65] R. Parthiban, D. Saravanan, and S. Usharani, "WORLDWIDE CENTER DISCOVERY USING SURF AND SIFT ALGORITHM," p. 4, 2018.

[66] M. Liu, X. Li, J. Dezert, C. Luo, and E. Dept, "Generic Object Recognition Based on the Fusion of 2d and 3d SIFT Descriptors," p. 8, 2015.

[67] V. K. Jonnalagadda, “Object detection yolo v1 , v2, v3,” Jan 2019. [Online]. Available: https://medium.com/@venkatakrishna.jonnalagadda/object-detection-yolo-v1-v2-v3-c3d5eca2312a

[68] J. Hui, “Real-time object detection with yolo, yolov2 and now yolov3,” Aug 2018. [Online]. Available: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

[69] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, “Microsoft coco: Common objects in context,” 05 2014.

[70] “Common objects in context,” 2014. [Online]. Available: http://cocodataset.org/#home

[71] M. Sandler and A. Howard, “Mobilenetv2: The next generation of on-device computer vision networks,” Apr 2018. [Online]. Available: https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html

[72] J. Hui, “map (mean average precision) for object detection,” Apr 2019. [Online]. Available: https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

[73] “Object detector - apps on google play,” 2019. [Online]. Available: https://play.google.com/store/apps/details?id=com.tecomen.android.objectdetector&hl=en_US

[74] Ju, Luo, Wang, Hui, and Chang, “The Application of Improved YOLO V3 in Multi-Scale Target Detection,” *Applied Sciences*, vol. 9, no. 18, p. 3775, Sep. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/18/3775

[75] J. Farooq, “Object detection and identification using SURF and BoW model,” in *2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*. Quetta, Pakistan: IEEE, Apr. 2016, pp. 318–323. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7495245

[76] A. K. Gupta, M. Sharma, D. Khosla, and V. Singh, “Object detection of colored images using improved point feature matching algorithm,” *CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES*, vol. 1, no. 1, pp. 13–16, 2019.

[77] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

Configuration Manual

MSc Research Project
Cloud Computing

Ankit Dongre
Student ID: x18147054

School of Computing
National College of Ireland

Supervisor: Manuel Tova-Izquiredo

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ankit Dongre
Student ID:	x18147054
Programme:	Cloud Computing
Year:	2019
Module:	MSc Research Project
Supervisor:	Manuel Tova-Izquiredo
Submission Due Date:	12/12/2019
Project Title:	Configuration Manual
Word Count:	1000
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	10th December 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ankit Dongre
x18147054

1 Introduction

1.1 Purpose of this document

This Configuration Manual is done based on the NCI Research project requirements described in the Project Handbook. Main purpose of this document is to describe the required software tools and settings in order to successfully deploy the assitive application.

1.2 Document Structure

Section	**Purpose**
General Information	This module explains the minimum information and prerequisites required for the Android Studio setup
Developement Environment prerequisites	This module explains steps required for successful setup of development environment used for development and update of the solution
Solution Deployment Procedure	This module explains deployment procedure of the Composite model
Validations	This module explains the minimum requirements to validate fruitful deployment of the solution

2 General Information

2.1 Objective

The main objective of Composite model is to assist blind person in navigating. The key is to use the YOLO algorithm on a mobile device to enhance the mobility of the user while restricting the expenses.

2.2 Solution Summary

The Composite model solution consists of six phases which are interconnected and acts as one architecture.
- The input phase takes live images from the camera.
- The second phase is Image classifcation which predicts the object present in an image.
- In third phase, the model finds the location where these objects could be found in the

image.
- In the fourth phase, phase two and three are itererated so as to get all the objects present in the image
- In fifth phase, the name of the objects with their respective positions is announced.
- In sixth phase a new input image is taken and the control goes back to phase 1 until the application is terminated

2.3 Architecture Requirements

This section describes the required tools required for developing the application.

2.3.1 Android Studio

The installation and setting up of Android Studio, step-by-step guide *Install Android Studio : Android Developers* (n.d.).

2.3.2 COCO pre-trained model

Download the COCO pre-trained tensorflow model fromTensorflow (n.d.).

2.3.3 JDK

To install and setup JDK and Java environment follow the steps given in *Installing the JDK Software and Setting JAVA_HOME* (n.d.)

2.4 Android Debugging Bridge

Android Debugging Bridge(ADB) is required for deploying application on an android device. The ADB is specific to the mobile phone vendor.

2.5 Required Skills

In this guide, we assume that user has basic knowledge of using IDEs like Eclipse. The user also needs to have knowledge of Java, XML, Gradle and Maven.

3 Development Environment Requirements

3.1 Code Repository

Refer the zip file submitted in the ICT solution.

3.2 Required Programming Languages

Java Version 1.6+
Android studio lets you create android applications using Java backend with XML frontend. Some of the libraries that are essential for the development of the application are:

- android.hardware.camera2.*

- android.speech.tts.TextToSpeech
- org.tensorflow.*

3.3 Importing the project into android studio

Open Android Studio: File →Open →browse to the root of the directory containing the project →click ok.

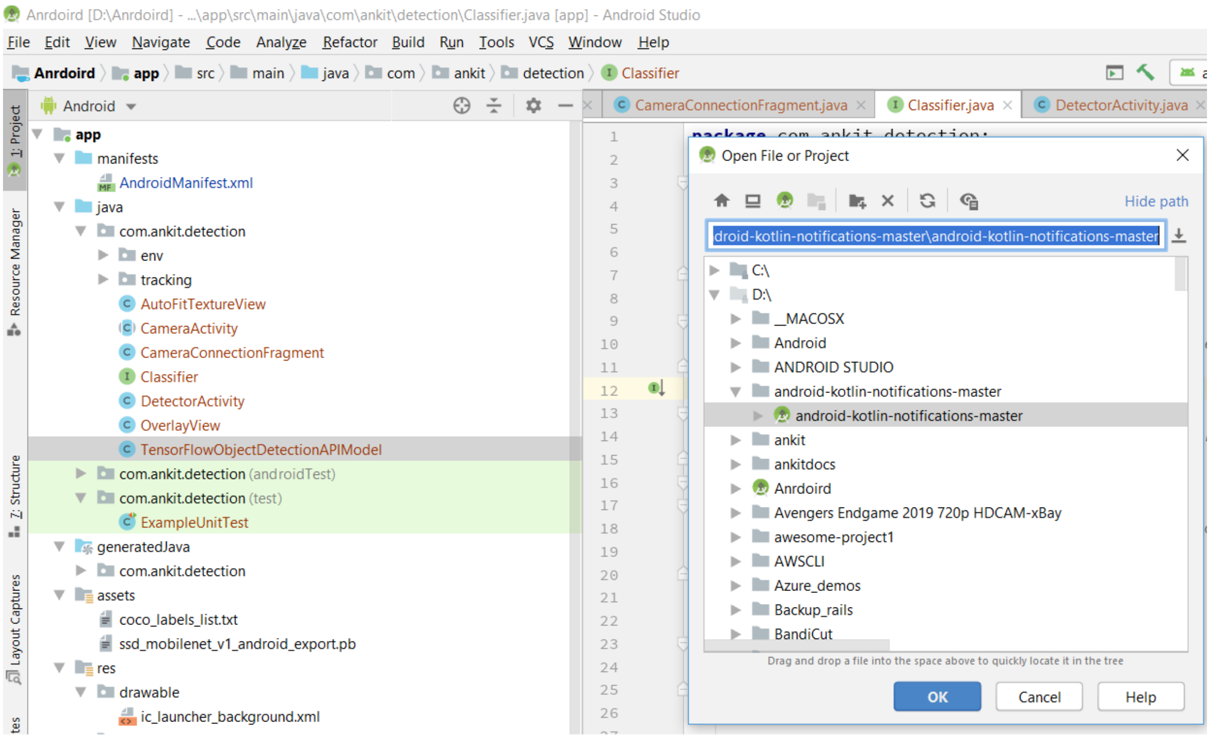

Figure 1: Creating the function

3.4 Gradle Syncing

Android Studio will automatically sync Gradle online. In case of issues refer to *Troubleshoot Android Studio : Android Developers* (n.d.)

3.5 Enabling Developer Mode

To be able to run the application on an android device, the usb debugging option should be enabled. This mode is inside developer options which is hidden. To unhide developer options: In your android phone goto settings →about phone →tap the build number 7 times →the Developer mode is now enabled →return to settings →find developer options →enable usb debugging

4 Validation

Make sure your android phone is connected to your pc via usb. Select the mode of connection as *file transfer*

1. Make sure your android phone is connected to your PC via USB
2. Select the mode of connection as *file transfer*
3. In Android Studio, click the green triangle button to deploy the application on the phone or press shift+F10 keys simultaneously
4. Select the device to deploy. Fig. 2 shows the deployment target selection
5. The app will ask for permission to download. Note: The installation will fail if the option to install applications from SD card is disabled
6. After installation the app will ask for permission if opening for the first time. Grant all the permissions
7. Open the app again
8. For every object successfully identified it will show the the class with confidence score whereas for unsuccessful object identification it will show question '?¿

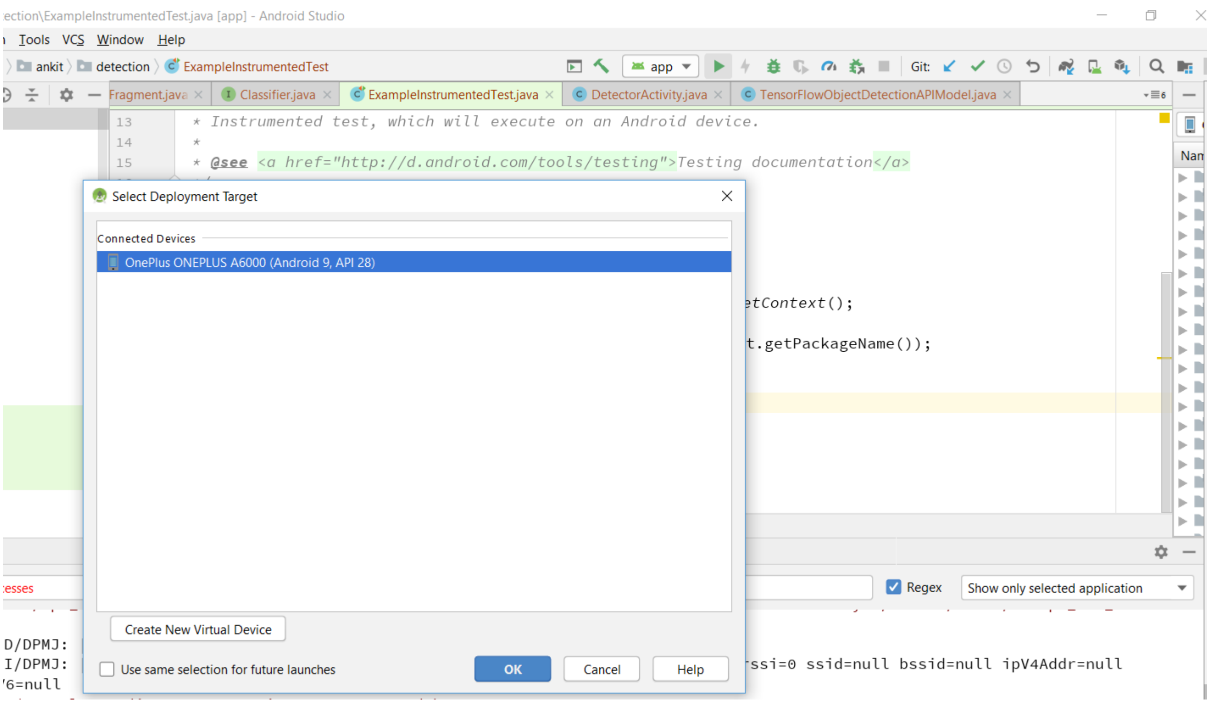

Figure 2: Selecting Deployment Target

Running the Test case:

1. Follow the same procedure as above till the installation of application step
2. Now, the application will

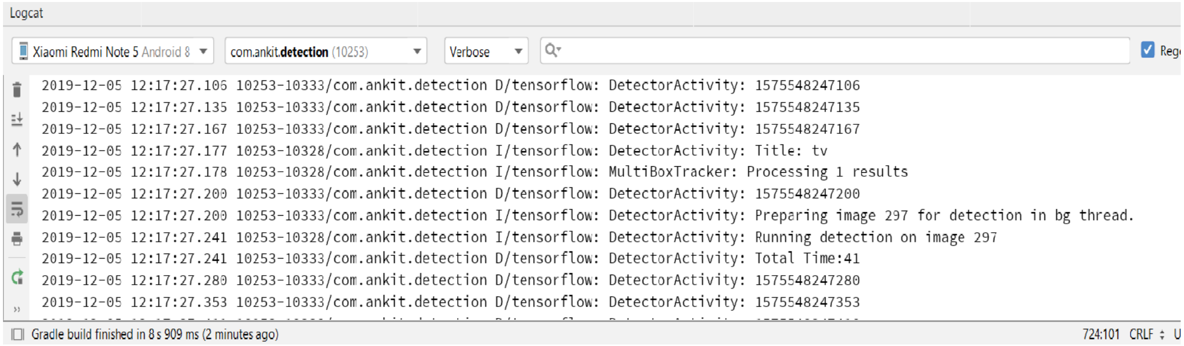

Figure 3: Logs generated for detected objects

References

Install Android Studio : Android Developers (n.d.).
URL: *https://developer.android.com/studio/install*

Installing the JDK Software and Setting JAVA_HOME (n.d.).
URL: *https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/*

Tensorflow (n.d.). tensorflow/models.
URL: *https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection*

Troubleshoot Android Studio : Android Developers (n.d.).
URL: *https://developer.android.com/studio/troubleshoot*