# Vision Assistant for Visually Impaired Individuals

Rithik Seth IIT2018032, Hardik Kumawat IIT2018034,
Aman Joshi IIT2018042, Milind Khatri IIT2018082

*VII Semester B.Tech, Information Technology,*
*Indian Institute of Information Technology, Allahabad, India*

***Abstract*—Artificial Intelligence has been touted as the next big thing that is capable of altering the current landscape of the technological Domain. Through the use of Artificial Intelligence and Machine Learning, pioneering works have been undertaken in the area of Vision and Object Detection. In this paper, we undertake the analysis of a Vision Assistant Application for guiding the visually-impaired individuals. With recent break- through in the computer vision and supervised learning models, the problem at hand has been reduced significantly to the point where new models are easier to build and implement over the already existing models. Now, different Object Detection models exist which provides object tracking and detection with great accuracy. These techniques have been well used in automating the detection task in different areas. Some of the newfound detection approaches such as YOLO(You Only Look Once), SSD(Single Shot Detector) and R-CNNs have proved to be consistent and pretty accurate in Real-Time Object Detection. We are going to have a brief look at these techniques in order to find a good base model for implementing our 'Vision Assistant'.**

***Index Terms*—Convolutional Neural Network(CNN), R-CNN, SSD, YOLO, Object Detection**

## I. Introduction

One of the primary goals of the image-based learning is to understand and differentiate among various scenic description of common objects of interest. This task can be subdivided into a number of subtasks - bounding box creation, object localization, attribute determination and relationship establish- ment. The images of various objects can be broadly classified into Iconic and Scenic Views. The Iconic approach assumes the presence of a single object with clear boundaries and separation edges. But, the iconic view is too simple to accomo- date real-life situations, wherein images are seldom iconic, but involves a large number of intertwined objects in a small space. In order to detect objects of interest, image segmentation and context mining should be applied to filter out points of interest. Most of the existing systems perform well under these iconic views, but achieves lower accuracy in scenic instances. Objects in scenic environments are cluttered, overlapping and without good contrast. Various techniques of segmentation are applied to extract useful information from these scenic views. When building new models, it is of paramount importance to select a learning domain most suitable to the need and implementation. In order to train these models, the dataset employed plays a crucial part in establishing good results. One of the major challenges is to find pertinent training images and samples to accomodate a more modular and robust learning. Various pioneering works have been done in collecting these image samples under one roof into a dataset. Some of these datasets contain millions of samples and training instances, spanning over thousands of objects. Currently, some of the more popular Datasets include - Google's ImageNet, Microsoft COCO Dataset, PASCAL VOC, SUN, etc. We take a look at these datasets in the following sections, aiming to find the most suitable for our Vision Assistant Implementation.

## II. Literature Review

### *A. You Only Look Once : Unified, Real-Time Object Detection (Joseph Redmon, Santosh Divalla, Ross Girshick) [4]*

YOLO is a new approach to object detection with an extremely fast architecture. Base model of YOLO processe images in real time at 45 frames per second, while Fast YOLO, smaller version of the network, processes 155 frames per second. YOLO makes more localization errors but is less likely to predict false positives on image backgrounds. It outperforms other detection methods, including Deformable Parts Model(DPM) and R-CNN. Current Detection Systems repurpose classifiers to perform detection. Models like Deformable Parts Model use a sliding window approach where the classifier is run at evenly-spaced locations throughout the image. On the other hand, recent approaches like R-CNN uses region proposal methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. Then post-processing is done in order to refine the bounding boxes, eliminate duplicate detections and rescore the boxes based on other objects in the images. These models that use complex pipelines are relatively slow and hard to optimize.

With YOLO, you only look once at an image to predict what objects are present and where they are. YOLO is relatively simple and fast, training on full images to make the detection process effective. YOLO is extremely fast because it doesn't use a complex pipeline. It runs a neural network on a new image at test time to predict detections, using which we can process streaming video in real-time. When making predictions, YOLO reasons about the image. During training and test time, it sees the complete image so that it can extract contextual information about classes as well as appearance. This model is implemented as a CNN which extracts features from the images, followed by 2 fully connected layers which predict the output coordinates and their probabilities. For training and inference purposes

it uses the Darknet framework. YOLO faces difficulty with small objects that appear in groups, for example flock of birds and also it struggles to generalize objects that appear in different aspect ratios. Incorrect localizations are the main source of error.

### *B. Incremental Few-Shot Object Detection (Juan-Manuel, Perez-Rua, Xiatian Zhu, Timothy Hospedales, Tao Xiang) [2]*

Despite the success of deep convolutional neural networks (CNNs) in object detection, for almost all the current models, a lengthy process of numerous iterations of batches are used to train them. Here in the current setting, all the target classes with a large number of training samples interpreted with training data are familiar and for training purposes all the training images are used. The potential for these methods to accommodate new classes online and grow is restricted severely by the interpretation cost and training complexity.

To avoid the earlier mentioned limitations, we can study a learning setting known as iFSD (Incremental Few-Shot Detection). The Incremental Few-Shot Detection or iFSD setting is defined as: (1) The set of base classes which have sufficient number of training samples can be used to pre-train the detection model in advance (2) When the training part is completed, the Incremental Few-Shot Detection model must be ready for deployment to a real-world application where the new classes can be added at any time with the help of few annotated examples. The model should work with learning without forgetting principle i.e. it should give a fair result on all the classes registered so far.(3) The learning of novel classes from an unbounded stream of examples should be feasible in terms of memory footprint, storage, and compute cost. The models should be able to be deployed on low-resource devices such as smartphones and robots.

## RELATED WORK

**Object Detection:** Current models in object detection have two categories:(1) One-stage detectors. (2) Two-stage detectors. The two-stage detectors are better than one-stage detectors in case of performance but are less efficient than one-stage detectors due to the need for inference of the object region. Here in both the cases the detectors are needed to train in an offline batch mode and they assume a large amount of training images per class. During the model deployment when the novel classes are needed to add then this restricts the scalability and usability.These can act as the backbone of detection for a few-shot detector although they are non-incremental. The ONCE which we are using is based on the one-stage CenterNet. The CenterNet is chosen because it can be easily broken down in the class-generic and specific parts, competitive detection accuracy and its efficiency.

**Few-shot learning:** FSL (few-shot learning) is studied for efficiently registering new classes in deployment for image recognition. Considering the large number of labelled examples of a set of base classes, few-shot learning try to meta-learn a data-efficient that helps to allow new classes to be learned from very few examples for each class. The few-shot learning is simpler than object detection.

### *C. Microsoft COCO: Common Objects in Context (Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnik, Piotr Dollar) [1]*

This paper has been written as a guide to the novel COCO Dataset created for Object detection and classification. It mainly focuses on the non-iconic or scenic views of im- ages, pointing out the difficulties encountered when detecting scenic views. It outlines image segmentation, bounding-box generation, heatmap, per-pixel color location. The focus is on 2D and 3D image localization and per-pixel semantic segmentation. The paper outlines the need for a large and rich-annotated images with a large number of instances per sample of objects. This collection aids in better learning and accuracy on scenic views of images. The different techniques of image segmentation, classification and detection have been defined with respective limitations. Semantic scene labeling has been defined as pixels of images belonging to each object category. This also helps in detecting objects wherein individual objects are hard to define and establish. Image localization and bounding box has been described as the major step in object detection and face tracking. The task of object classification requires binary image labels and is comparatively easier as we deal with general iconic images. Various statistics have been presented for COCO Dataset in comparison with other contemporary datasets.

### *D. YOLOv3: An Incremental Improvement (Joseph Redmon, Ali Farhadi) [3]*

This paper presents some design changes to YOLO, which makes it a little bigger but more accurate and fast. YOLOv3 is approximately as accurate as an SSD but three times faster. YOLOv3 clusters the dimensions of ground truth labels to generate anchor boxes for predicting the bounding boxes, where each bounding box has 4 coordinates, tx, ty, tw, th. Each box predicts the classes which may be present using multilabel classification. During training it uses the binary cross-entropy loss for making class predictions. YOLOv3 predicts boxes at 3 different scales and then extracts features from them using a network(named as Darknet-53) which is a hybrid between the network used in YOLOv2, Darknet-19 and modern residual networks. This network consists of successive 3x3 and 1x1 convolutional layers with a total of 53 convolutional layers. Darknet-53 performs better than many of the recent classifiers. Darknet-53 is even better than ResNet-101 and ResNet-152 in terms of performance and speed. Because of the better utilisation of GPU, Darknet-53 has the highest measured float- ing point operations per second. On the other hand, ResNets have many layers which makes it very inefficient. YOLOv3 performs extremely well on the old detection metric of mAP at IOU=.5, and is almost as good as RetinaNet and much above SSD variants. Performance of YOLOv3 decreases as the IOU threshold increases which means that it faces difficulty in getting the boxes perfectly aligned with the object. YOLOv3 in comparison

with YOLO struggles with medium and larger size objects. Overall YOLOv3 is a pretty good detector, extremely fast and accurate.

*E. Object Detection in 20 years: A survey (Zhengxia Zou, Zhenwei Shi, Member, IEEE, Yuhong Guo, and Jieping Ye) [5]*

This paper reviews more than 400 papers of object detection spanning from 1990s to 2109, focussing on the technical advancements made in this area. This paper emphasizes on several topics which include several early stage detectors, datasets for detection, metrics, possible speed up techniques which can be used, and the recent state of the art detection methods. This paper also sheds light on some important applications of detection, such as text detection, face detection, pedestrian detection, etc, and makes analysis of the development made and challenges faced in recent time.

Various aspects make this paper different from all the reviews done on object detection. In depth research on the key technologies and state of the art object detection system has been done here, while the previous reviews lacked fundamental analysis to give readers complete understanding of complex techniques. Most of the previous reviews were focussed on a short period of time or on some specific detection task without considering the development history.

Object detection has gone through two historical periods: (i) Traditional object detection period (Before 2014) (ii Deep learning based detection period (After 2014). Traditional object detection algorithms which include Viola Jones Detector, Histogram of Oriented Gradients (HOG) detector and Deformable Part-based Model were built based on handcrafted features and as the performance of handcrafted features became saturated, deep learning based detection methods started evolving. In deep learning era, object detection can be categorised as:"two-stage detection" (which includes RCNN, SPPNet, Fast RCNN, Faster RCNN, Feature Pyramid Networks) and "one-stage detection"(which includes YOLO, SSD, RetinaNet). In object detection several known datasets have been released in recent years, like PASCAL VOC, ImageNet, MS-COCO etc. This paper also reviews AlexNet, VCG, GoogleNet, ResNet as the engine of detectors which affects the accuracy of detectors.

## III. Datasets

Since we will be dealing with assisting blind people, for that we will need training images of several commonly occurring daily objects that the blind people need assistance with. MS-COCO dataset appears to be better suited for our purpose so we are using that. There are several other datasets which are used in object detection some of them are mentioned below:

*A. Microsoft COCO (Common Objects in Context) Dataset*

It is a large, richly-annotated dataset comprising images depicting complex everyday scenes of common objects in their natural context. It addresses 3 major problems in scene understanding i.e. detecting non-iconic views, contextual rea- soning and precise localization of objects. Dataset consists of large set of images containing contextual relationship and non- iconic object views, with 91 common object categories, 25 million labeled instances in 3,28,000 images. COCO has more instances per category as compared with other contemporary datasets.

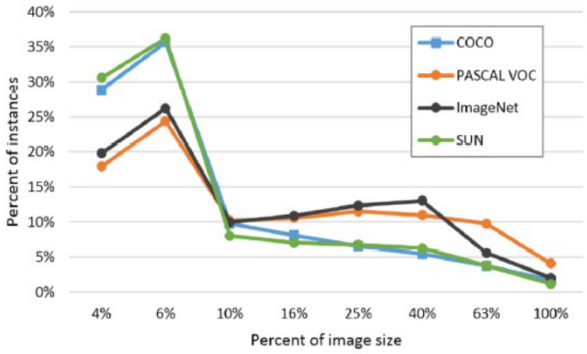


Figure 1. Instances vs Image size comparisons for different datasets

*B. Google ImageNet Dataset*

The ImageNet project is a large visual database designed for use in visual object recognition software research. The database has more than 14 million-annotated images and at for least with one million of the images, bounding boxes are also provided. ImageNet contains more than 20,000 categories. The number of categories is very large, but instances per category are substantially low for rigorous training. The dataset is organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently it has an average of over five hundred images per node.

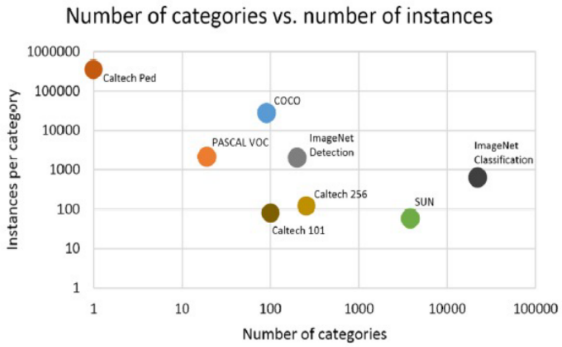


Figure 2. No. of Categories vs Instances comparison for different datasets

*C. SUN Dataset*

The main aim of this dataset is to provide researchers with a comprehensive collection of annotated images covering a large variety of environmental scenes, places and objects within. The samples are built using vocabulary based on scenes and places. The vocabulary is then queried to obtain images from the internet. It has 16, 783 images

of various scenes. The dataset has been optimally divided into training and testing samples.

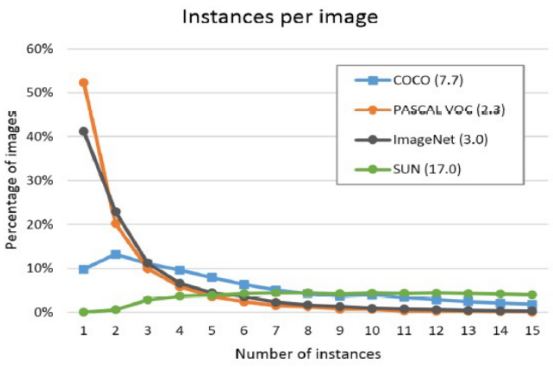


Figure 3. Percentage of Images vs Instances for different datasets

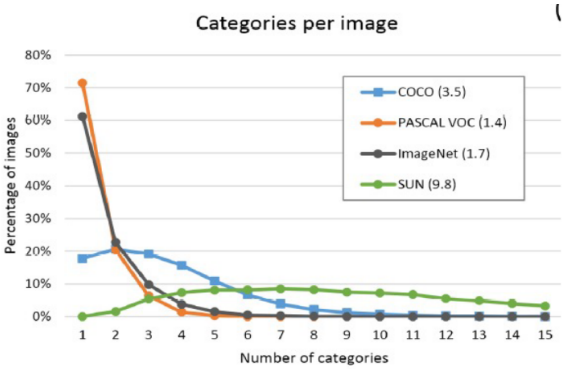


Figure 4. Percentage of Images vs Categories for different datasets

## IV. Proposed Methodology

### A. YOLO - Object Detection model

You only look once (YOLO) is a state-of-the-art, real-time object detection system. It is a fast object detection approach which scans complete image to extract contextual information with high accuracy. Prior detection systems repurpose classi- fiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections. Yolo uses a totally different approach. by applying a single neural network of 24 convolutional layers followed by 2 fully connected layers to the entire image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. Yolo looks at the whole image at test time, so its predictions are informed by global context in the image. It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image. This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN. A newer version of YOLO, YOLOv3 uses a few tricks to improve training and increase performance, including: multi- scale predictions, a better backbone classifier, and more.

### B. Project Architecture

The Project Architecture comprises various dependent component implemented as stand-alone modules. We adopt a client-server architecture, wherein Server is a remote entity running on a local machine. The Client Application is implemented as a mobile application that is connected to the camera device through a wireless network either using Bluetooth, WiFi, or other Wireless transmission protocols. The only requirement is sufficient bandwidth and low latency. The mobile application sends a request to the mirror site which, in turn, forwards it to the local server. The local server, running a YOLOv3 model, detects objects within the input image and creates a list of objects found. This list is finally converted into a string and sent as a response to the mirror site, which, redirects the response to the client application. The client application using text-to-speech functionality converts this string into audio that is fed into the earpiece of the visually-impaired individual. For simplicity, the entire image is divided into 9 different zones, viz., Center, Top Left, Bottom Right. The model also predicts the zone of each object detected using bounding-box location returned by the YOLO model.

The various components of the system are as follows:

- YOLOv3: YOLO model implemented in Python using CV2 and Numpy libraries.
- Local Server: Server implemented in Python using Flask and ngrok libraries.
- Mobile Application: Mobile Application implemented in Dart and Flutter using Dio, tts and camera libraries.
- HTTP Mirror: Ngrok creates a mirror http site that redirects and forwards request and response between client and server. The request is made to this http site.
- Camera: - External camera device installed on the walking cane.

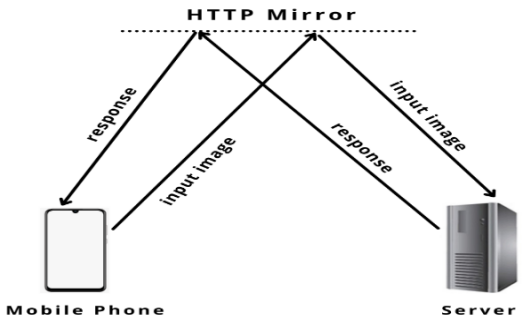


Figure 5. Client-Server Interaction process

### C. Application UI

Since the application is made for visually-impaired individuals, it is obvious that very rudimentary User interface is sufficient. Keeping this in mind, we have developed a simple and lucid Application UI that will be used for demonstration purposes only. The external camera hardware has not been used for demonstration purposes, instead, the mobile device camera is used. A few samples of the Application UI along with the local

server is shown below.

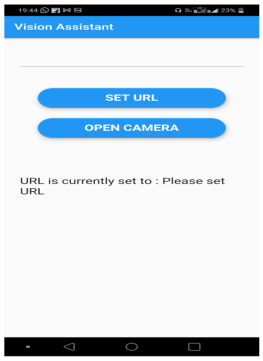


Figure 6. Home Screen on Client Application

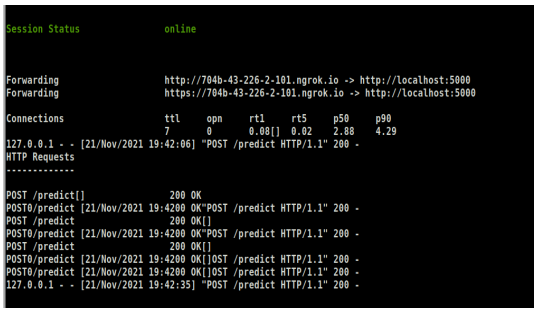


Figure 7. Local Server running on local machine

## V. Results

### A. Statistical Results

- YOLOv3 is on par with SSD variants in terms of COCOs average mean AP metric. At 320x320 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but 3 times faster.
- Looking at the "old" detection metric of mAP at IOU=.5(or AP50), YOLOv3 is almost on par with RetinaNet. It achieves 57.9 AP50 in 51 ms on a Titan X, compared to 57.5 AP50 in 198 ms by RetinaNet, similar performance but 3.8× faster.
- YOLOv3 is not as great on the COCO average AP between .5 and .95 IOU metric, indicating that its performance drops significantly as the IOU threshold increases.
- However, YOLOv3 is a very strong and fast detector, which is very good on the old detection metric of .5 IOU.

### B. Experimental Results

Some sample outputs for demonstration purposes are shown below with verbose feedback which will be audible to the blind person.

Verbose (Fig. 11):

- car1 is in Center Left
- truck1 is in Center
- car2 is in Center
- person1 is in Center

| *One-stage methods* | | | | | | | |
|---|---|---|---|---|---|---|---|
| YOLOv2 [13] | DarkNet-19 [13] | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD513 [9, 2] | ResNet-101-SSD | 31.2 | 50.4 | 33.3 | 10.2 | 34.5 | 49.8 |
| DSSD513 [2] | ResNet-101-DSSD | 33.2 | 53.3 | 35.2 | 13.0 | 35.4 | 51.1 |
| RetinaNet [7] | ResNet-101-FPN | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| RetinaNet [7] | ResNeXt-101-FPN | **40.8** | **61.1** | **44.1** | **24.1** | **44.2** | 51.2 |
| YOLOv3 608 x 608 | Darknet-53 | 33.0 | 57.9 | 34.4 | 18.3 | 35.4 | 41.9 |

Figure 8. Comparison based on Average Precision on MS Coco

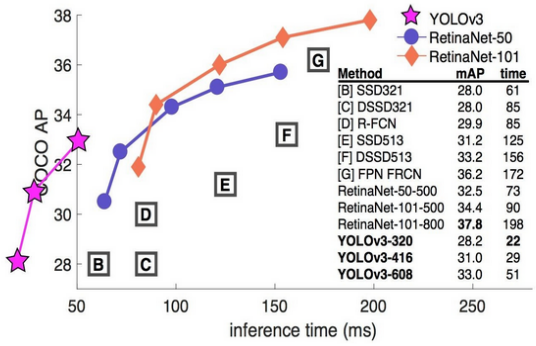


Figure 9. Comparison based on Inference Time

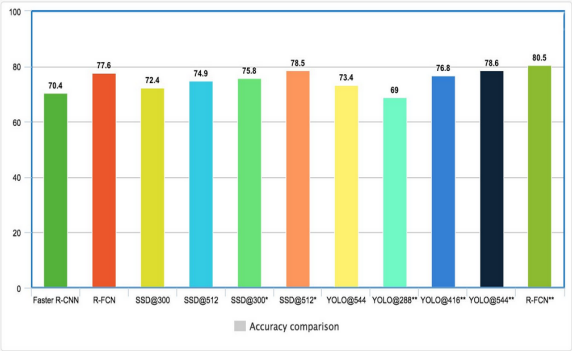


Figure 10. Comparison based on accuracy on MS Coco

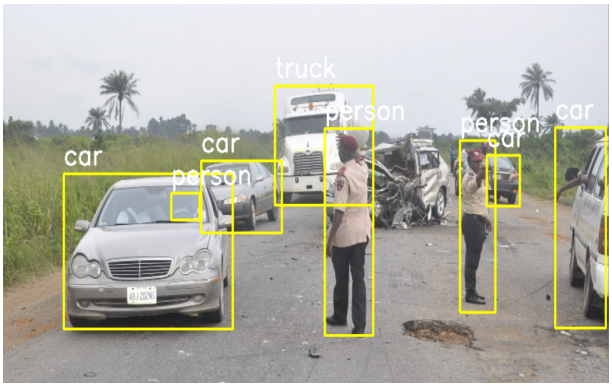


Figure 11. Sample output 1

- person2 is in Center Right
- car3 is in Center Right
- car4 is in Center Right
- person3 is in Center Left

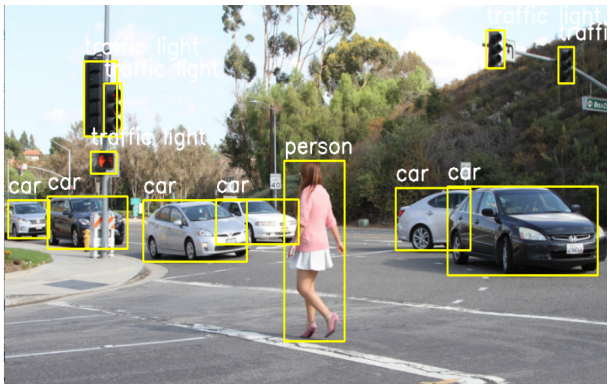


Figure 12. Sample output 2

Verbose (Fig. 12):

- car1 is in Center Right
- traffic light1 is in Top left
- car2 is in Center Left
- car3 is in Center
- car4 is in Center Right
- car5 is in Center Left
- person1 is in Center
- traffic light2 is in Top Right
- traffic light3 is in Top Right
- traffic light4 is in Top left
- traffic light5 is in Center Left
- car6 is in Center Left

## VI. Conclusion

Millions of visually-impaired individuals face a lot of difficulties in running daily errands that require certain visual capacity. Earlier, it was only possible to guide these individuals. But, now with the advent of mobile technology, Internet and Artificial Intelligence, Virtual Guide Applications are developed to provide these individuals with visual aid. We aim to develop a visual assistant for visually-impaired individuals that uses an IoT-based Cane fitted with a camera. The important aspect is to train the model on a large number of samples so that we can obtain a fine-tuned and accurate Object-detection model. Many Object Detection algorithms have been proposed with wide-scale applicability. Choosing one such model that pertinently solves the problem at hand is a major determiner in obtaining good accuracy. We have provided reviews of various object detection techniques that works with scenic view of generic images.

## VII. References

[1] Tsung Yi Lin. "Microsoft coco: Common objects in context. European conference on computer vision. Springer, Cham" (2014). URL: https://arxiv.org/abs/1405.0312.

[2] Juan-Manuel Perez-Rua. "Incremental few-shot object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition" (2020). URL: https://openaccess.thecvf.com/content_CVPR_2020 / html / Perez - Rua_Incremental_Few - Shot_Object_Detection_CVPR_2020_paper.html.

[3] Ali Farhadi Redmon Joseph. "Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767" (2018). URL: https://arxiv.org/abs/1804.02767.

[4] Joseph Redmon. "You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition" (2016). URL: https://arxiv.org/abs/1506.02640.

[5] Zhenwei Zou Zhengxia. "Object detection in 20 years, A survey. arXiv preprint arXiv:1905.05055" (2019). URL: https://arxiv.org/abs/1905.05055.