Implementing Visual Assistant Using YOLO and SSD for Visually-Impaired Persons

Submitted: 13th September 2022; accepted: 3rd January 2023

Ratnesh Litoriya, Kailash Chandra Bandhu, Sanket Gupta, Ishika Rajawat, Hany Jagwani, Chirayu Yadav

DOI: 10.14313/JAMRIS/4-2023/33

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 work has been undertaken in the area of Visual and Object Detection. In this paper, we undertake the analysis of a Visual Assistant Application for Guiding Visually-Impaired Individuals. With recent breakthroughs in 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 than the already existing models. Different object detection models exist now that provide object tracking and detection with great accuracy. These techniques have been widely used in automating detection tasks in different areas. A few newly discovered detection approaches, such as the YOLO (You Only Look Once) and SSD (Single Shot Detector) approaches, have proved to be consistent and quite accurate at detecting objects in real-time. This paper attempts to utilize the combination of these state-of-the-art, real-time object detection techniques to develop a good base model. This paper also implements a 'Visual Assistant' for visually impaired people. The results obtained are improved and superior compared to existing algorithms.

Keywords: *YOLO, SSD, Object detection, R-CNN, COCO*

1. Introduction

Visually-impaired or blind people are incapable of seeing, which is crucial for daily life. Visually-impaired people's autonomy is limited to some extent by their inability to see. In the past, assistive systems have been created using computer vision and machine learning, which have recently experienced rapid growth. The way people with cognitive limitations interact with the outside world has changed dramatically as a result of recent advances in assistive technology (AT). Among these disabilities, visual impairment stands out as the most restrictive. Any technology designed to aid a person with a disability is considered AT. With the aid of AT, people with disabilities can participate in civic activities, and the job market, and have a healthy, productive, independent life [1]. The use of AT lessens the need for long-term care, formal health and support services, and caregiver labor. Without AT, people frequently experience exclusion, isolation, and poverty, which worsens the effects of illness and disability on an individual, their family, and society as a whole. Autonomous vehicles are now possible thanks to significant advancements in artificial intelligence. These algorithms can be used effectively in AT to help the blind and visually impaired in the areas of education, navigation, and social interaction. People who are blind or visually impaired can access information through touch or voice. At least one billion people worldwide have near- or distance-vision impairments that could have been avoided or have not yet been addressed, according to the World Health Organization, which estimates that 2.2 billion people worldwide have visual impairments [2]. The expected causes of an increase in the prevalence of vision impairment are population growth and aging. Several studies have already been conducted to investigate the correlation between the prevalence, causes, and social factors of visual impairments and other diseases [3–5]. Mobile applications along with Machine learning, AI, and IoT have been found to be promising and provide lifesaving technologies for assisting humans with various diseases and disabilities [6–9].

One of the primary goals of image-based learning is to understand and differentiate among various scenic descriptions of common objects of interest. This task can be subdivided into several subtasks: bounding box creation, object localization, attribute determination, and relationship establishment. 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. However, the iconic viewpoint is too simplistic to account for real-world situations in which images are rarely iconic but involve a large number of intertwined objects in a small space. 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 achieve 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 given needs and implementation. For training 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 accommodate more modular and robust learning. Various pioneering work has 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 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 Visual Assistant Implementation. To improve a visually-impaired person's perception, a new model is presented that connects an AT device with Smart Objects and their cloud.

The rest of the paper is organized as follows: A detailed and state-of-the-art review of existing literature in the field is presented in Section 2. The details of the dataset used in this study are explained in Section 3. A detailed discussion of the YOLO object detection model is included in Section 4. The architecture of the proposed system is presented in Section 5. Obtained results and detailed discussions are presented in Sections 6. Section 7 concludes the article and sketches future work directions.

2. Literature Survey

Numerous assistive systems have been introduced for object detection in the last few years that rely on sensors, the Internet of Things, and computer vision to help the blind. These systems each have their benefits and drawbacks.

Zou et al. [10] reviews more than 400 papers on object detection spanning from the 1990s to 2109, focusing on the technical advancements made in this area. This paper emphasizes 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 an analysis of the development made and challenges faced in recent times. 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 systems has been done here, while the previous reviews lacked fundamental analysis to give readers a complete understanding of complex techniques. Most of the previous reviews were focused on a short period or some specific detection task without considering the development history.

Ambati and L. Gayer [11] underline how crucial it is to customize the choice of machine learning (ML) techniques based on the particular HAR (Human Activity Recognition) requirements and the features of the associated HAR dataset. Overall, this study aids in comprehending the benefits and drawbacks of ML techniques and directs the applicability of various ML methods to various HAR datasets.

An accessible web interface for visually-impaired individuals is presented by Iyer et al. [12] to maximize ease of use and provide users with a hassle-free experience, the virtual assistant is an operating system that is independent and doesn't rely on keyboard input from the user. Communication with and customization of the system are possible using speech-to-text and text-to-voice interfaces. This presentation provides an overview of the system design and implementation methodology for the three modules currently in use. To answer user queries quickly and accurately, Wikipedia uses a BERT model built from the SQuAD dataset. It was found that 80.88% of the words exactly matched. Anyone with visual impairments can easily access any website using the virtual assistant. With this program, you don't have to memorize complex keyboard commands or use screen readers. As a tool for interacting with the websites, the assistant is not only very convenient but also quite effective. According to the results, the software was successfully run on the three most popular sites: Google, Gmail, and Wikipedia. It was run separately on each of these sites. The software is a stepping stone toward Web 3.0 where all functions can be controlled through voice commands. Visually-impaired people find themselves wandering inside unusual challenging areas. Many smart systems have intended to help blind people in these difficult, often dangerous, situations. However, some of them are not free, hard to find, or simply too expensive. Saffoury et al. [13] presented a low-cost wear system for blind people that was designed to allow them to discover obstacles in their place. The proposed program consists of two main components; hardware components, and a laser pointer ($ 12), as well as an android smartphone, which makes our system cheaper and more accessible. I Conflict avoidance algorithm uses image processing to measure distances to objects in the surrounding area. This is based on laser triangular light. This detection of obstacles is enhanced by the edge discovery within the captured image. An additional feature for a system is to see and alert the user when there are stairs in the camera view area. Obstacles are brought to the user's attention using the acoustic signal. Our system showed that solid, with only 5% of a false alarm level and 90% sensitivity with obstacles 1 cm wide. This system had some limitations: It may not work well in a shiny environment as laser intensity may decrease and the distance between the camera and the laser should not change.

Mohanta et al. [14] proposed an assistant for visually-impaired individuals. In their set up after capturing a photo from a smartphone, the user can easily read menu cards of restaurants, the room number of the hotel, and can also find their belongings. The voice control feedback mechanism is also used in the app through which the user can perform various tasks with the help of the voice assistant. Cloud computing, image processing, and ML are used to develop the application. The central aim of using ML is to allow computers to learn automatically without intervention from humans. Multiple fonts can be detected while reading the text even if the font is unique or not common.

Such unique fonts can be found in greetings cards, business cards, etc. It is also proved to be beneficial in different sectors like banking, education, travel and tourism, etc. Various objects of daily use, vehicles, and food can be detected and recognized with the help of this application.

Sharma et al. [15] created a system that helps a person who is visually impaired navigate by speaking through the earpiece to identify the person. They suggested developing a mobile app that used numerous deep learning models to improve the management of applications. The camera continuously fed images into the system as inputs, the core system processed this information, and the earpiece served as the output device to deliver this output to the user.

An intelligent virtual assistant called Project Nethra [16] offers voice-based communication to users who are blind or visually impaired. It enables a wide range of functionality based on various internet services and social media for the target users to interact with computers and internet-based services. Nethra will perform tasks on the user's behalf rather than just returning search results. Project Nethra will converse with the user conversationally by speaking back to them after hearing what they say and detecting it. The voice recognition module, natural language processing module, conversational agent, and content extraction module are the system's four main parts. Kumaran et al. [17] discuss the development of virtual personal assistants and speech recognition systems. The current system is operated and maintained by a third party and operates online. This application used the local database, speech recognition, and synthesizer while safeguarding user data from outside parties. To recognize the speech, a parser called SURR (Semantic Unification and Reference Resolution) is used. Text is converted to phonemes by a synth.

To design the Next-Generation of VPAs model, Kepuska and Bohouta [18] used multi-modal dialogue systems, which process two or more combined user input modes, such as speech, image, video, touch, manual gestures, gaze, and head and body movement. By utilizing various technologies, including gesture recognition, image/video recognition, speech recognition, a sizable dialogue, a conversational knowledge base, and a general knowledge base, the new model of VPAs will be used to increase interaction between humans and machines.

Iannizzotto et al. [19] developed a virtual assistant architecture for smart home automation systems using some of the most cutting-edge methods in computer vision, deep learning, speech generation, recognition, and artificial intelligence. The developed prototype of the suggested assistant is interactive, resource-efficient, effective, and adaptable, and it runs on a small, inexpensive Raspberry PI 3 device. The system was integrated with an open source home automation environment for testing purposes, and it ran for several days with users being urged to interact with it. It turned out to be precise, dependable, and appealing.

An interesting research, Gnana and Praveen [20], suggested a method for automatically estimating depth from a single image using the local depth hypothesis and its application to help the blind. A camera records the environment in front of the user, and the recorded image is scaled for computational effectiveness. Edge detection and morphological techniques are used to separate the obstacles in the image's foreground. Then, based on the local depth hypothesis, the depth is computed for each barrier. Rahman et al. [21] presents the architectural framework for a smart blind assistant that integrates IoT and deep learning. The suggested approach utilizes a deep learning paradigm, a Raspberry Pi, and a camera module to create an intelligent cap. The suggested concept shows the structural layout of a smart blind stick that makes use of a microprocessor and numerous sensors. For immediate data monitoring, the model uses Bluetooth and the Internet of Things' connectivity. Using an IoT cloud server, the authorized person continues to keep an eye on vision impairment.

2.1. Related Work About Object Detection and Related Technologies

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

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

Object detection has gone through two historical periods: a (i) traditional object detection period (before 2014) and a (ii) deep learning-based detection period (after 2014). Traditional object detection algorithms, which include the Viola-Jones Detector, the Histogram of Oriented Gradients (HOG) detector, and the 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 the deep learning era, object detection can be categorized 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, and ResNet as the engine of detectors that affect the accuracy of detectors.

YOLO (You Only Look Once) is a new approach to object detection with an extremely fast architecture [22]. The base model of YOLO processes images in real-time at 45 frames per second, while Fast YOLO, a smaller version of the network, processes images at 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 the Deformable Parts Model (DPM) and R-CNN. Current detection systems repurpose classifiers for detection. The Deformable Parts Model uses a sliding window approach, where the classifier runs at evenly distributed locations throughout the image. Recent approaches like R-CNN, on the other hand, generate potential bounding boxes first using region proposal methods, then run a classifier on the suggested boxes. In post-processing, the bounding boxes are refined and the detections are eliminated, and the scores are recalculated based on the other objects in the image. 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. Because YOLO does not use a complex pipeline, it is extremely fast. It uses a neural network to predict detections from a new image at test time, which enables it to process streaming video in real-time. YOLO analyzes the image while making predictions. For training and testing, a fully connected layer predicts the output coordinates and their probabilities by taking into account the full image so it can extract contextual information about classes. For training and inference purposes it uses the Darknet framework. YOLO faces difficulty with small objects that appear in groups, for example, a flock of birds, and also it struggles to generalize objects that appear in different aspect ratios. Errors are caused primarily by incorrect localization.

Despite the success of deep convolutional neural networks (CNNs) in object detection, for almost all the current models a lengthy process of numerous iterations in a batch is used to train them. In the current scenario, all of the target classes have a great deal of training data interpreted with training samples, and all of the training images are used for training purposes. Due to their high interpretation cost and complex training requirements, these methods have a limited ability to accommodate online classes and grow.

To avoid the earlier mentioned limitations, we can study a learning setting known as iFSD (Incremental Few-Shot Detection) [23].

The Incremental Few-Shot Detection or iFSD setting is defined as (1) the set of base classes that have a sufficient number of training samples that can be used to pre-train the detection model in advance, and (2) when the training part is completed, the iFSD 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 the principle, i.e., it should give a fair result on all the classes registered so far. (3) Memory footprint, storage, and compute costs should be feasible for the learning of classes from an unbounded flow of examples. The models should be able to be deployed on low-resource devices such as smartphones and robots.

A guide to the novel COCO Dataset created for Object detection and classification is presented by T.Y. Lin et al. [24]. It mainly focuses on the non-iconic or scenic views of images, pointing out the difficulties encountered when detecting scenic views. It outlines image segmentation, bounding-box generation, heatmap, and 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 image 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 have 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 the COCO Dataset in comparison with other contemporary datasets.

Joseph Redmon and Ali Farhadi [25] presented some design changes to YOLO, which makes it a little bigger but more accurate and faster. 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, and 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. Darknet-53, a hybrid network composed of YOLOv2 and Darknet-19, predicts box shapes at 3 different scales and extracts features from them. 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 utilization of GPU, Darknet-53 has the highest measured floating-point operations per second. On the other hand, ResNets have many layers which make them 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.

It is evident from the literature survey that many tools and solutions have been created to aid and direct visually-impaired people around indoor and outdoor pathways. Nevertheless, they haven't entirely satisfied the user needs and technological specifications. Currently, the majority of these unanswered questions are being addressed independently in many research areas, including indoor location, computation offloading, distributed sensing, and the examination of spatially-related perceptual and cognitive processes in visually-impaired individuals. However, mobile phones and other such devices, along with state-of-the-art technologies, are quickly becoming integrated into their daily lives. Old and new solutions have become workable in this setting, and some of them are now on the market as smartphone applications or portable devices.

3. Dataset Used

3.1. Microsoft COCO (Common Objects in Context) Dataset

There are numerous images depicting complex everyday scenes of everyday objects in their natural setting contained within this large, richly annotated dataset. It addresses 3 major problems in scene understanding, i.e., detecting non-iconic views, contextual reasoning, and precise localization of objects. The dataset consists of a large set of images containing contextual relationships 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 compared to other contemporary datasets (Fig. 1).

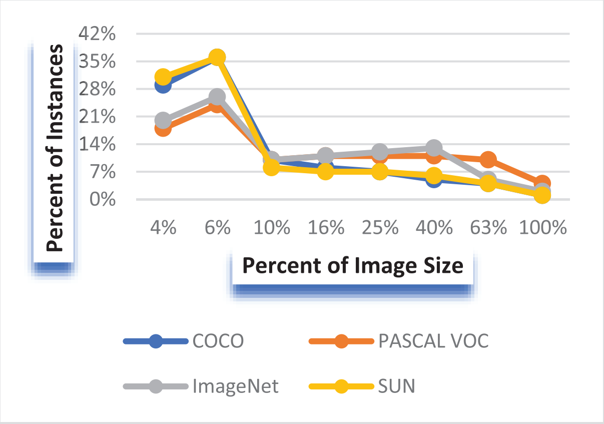


Figure 1. Instances vs image size comparisons for different datasets

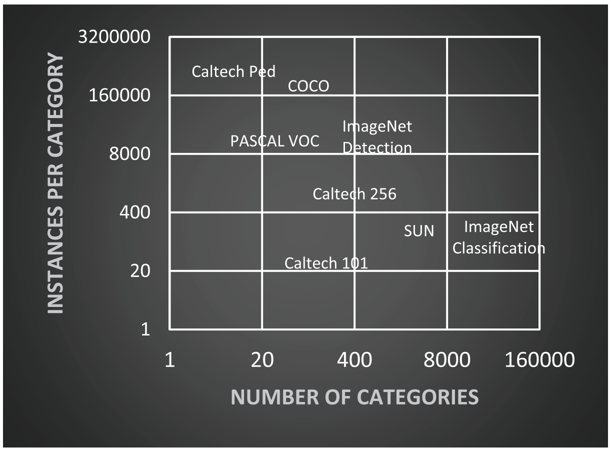


Figure 2. Instances vs no. of categories' comparisons for different datasets

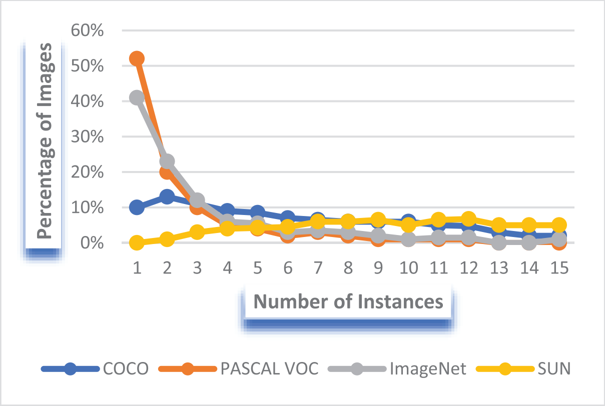


Figure 3. Percentage of images vs instances for different datasets

3.2. Google ImageNet Dataset

Research using ImageNet aims to develop software that can recognize visual objects.

The database has more than 14 million-annotated images and with at least 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 (Fig. 2). The average number of images per node is currently over 500.

3.3. SUN Database

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 (Figs. 3 and 4). 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.

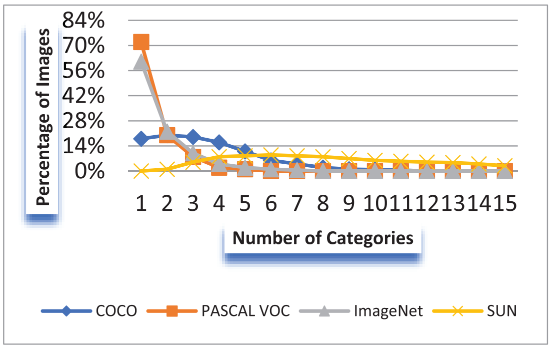


Figure 4. Percentage of images vs categories for different datasets

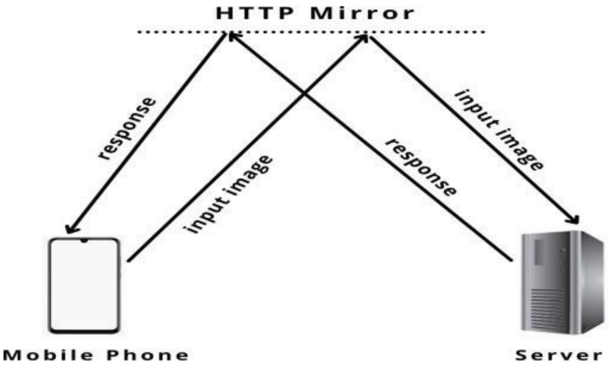


Figure 5. Client-Server interaction

4. YOLO – Object Detection Model

The YOLO system detects objects in real-time using state-of-the-art sensors. It is a fast object detection approach that scans the complete image to extract contextual information with high accuracy. In prior detection systems, classification or localization functions are repurposed to perform detection. The model is applied to an image at multiple scales and locations to accomplish detection. YOLO detects high-scoring regions using a unique method that involves applying 24 convolutional layers followed by two fully connected layers to the entire image. An image is divided into regions by this network and bounding boxes and probabilities are predicted for each region. Predicted probabilities are used to weigh the bounding boxes. The YOLO predicts based on how the image is global at the time of the test, not its components. Unlike systems such as R-CNN, which require thousands of evaluations for a single image, this makes predictions using one network evaluation. Faster than R-CNN and 100x faster than Fast R-CNN, it is more than 1000x faster than R-CNN. A newer version of YOLO, YOLOv3 incorporates several improvements to provide better training and better performance, including multi-scale predictions, a better backbone classifier, and more.

5. Architecture

The Proposed Architecture comprises various dependent components implemented as stand-alone modules as shown in Figure 5. We adopt a client-server architecture, wherein the 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, Wi-Fi, 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 the 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.

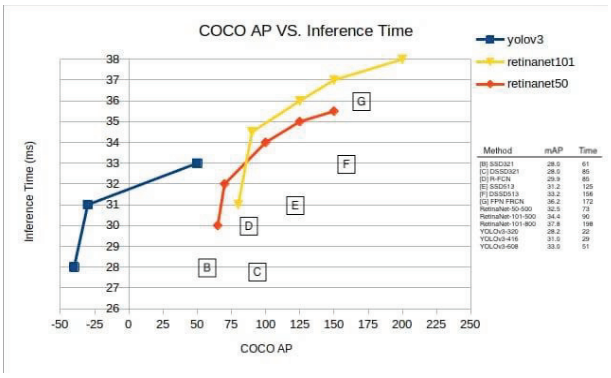
- **Mobile Application:** Mobile Application implemented in **Dart** and **Flutter** using **Dio**, **tts**, and **camera** libraries.

- **HTTP** Mirror: **Ngrok** creates a mirror **HTTP** site that redirects and forwards requests and responses between client and server. The request is made to this HTTP site.

- **Camera:** External camera device installed on the walking cane.

Table 1. Comparison based on average precision on MS coco

One Stage Methods	Backbone	Average Precision IoU			Average Precision Area		
		S	M	L	S	M	L
YOLOv2	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD513	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.5	51.1
RetinaNet	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet	ResNeXt-101-FPN	40.5	61.1	44.1	24.1	44.2	51.2
YOLOv3 608 * 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9



Method	mAP	Time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198
YOLOv3-320	28.2	22
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

Figure 6. Comparison based on inference time

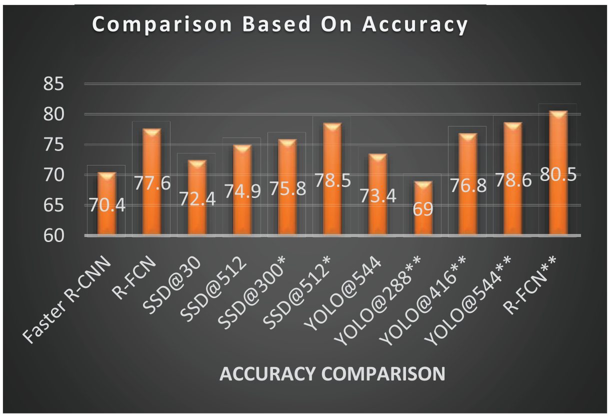


Figure 7. Comparison based on accuracy on MS coco

6. Results and Discussion

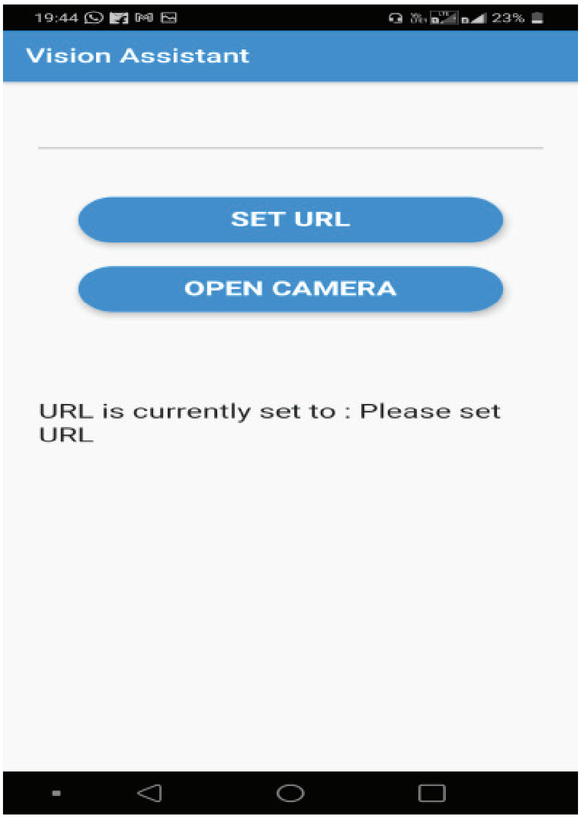
6.1. YOLOv3 Metrics (Table 1)

- Based on COCO's average mean AP metric, YOLOv3 is comparable to SSD variants. YOLOv3's performance at 320x320 is 22 ms at 28.2 mAP, which is 3 times faster than SSD.
- When mAP detection is updated to IOU=.5 (or AP50), YOLOv3 has a performance almost similar to RetinaNet. According to RetinaNet's tests, it achieves similar performance to 57.9 AP50 in 51ms, but it is 3.8* faster than RetinaNet's tests.
- The YOLOv3 performance drops significantly, as indicated by COCO average AP, as the IOU threshold increases, indicating that it is a poor performer.
- However, YOLOv3 is a very strong and fast detector, which is very good on the old detection metric of .5 IOU

6.2. Application UI

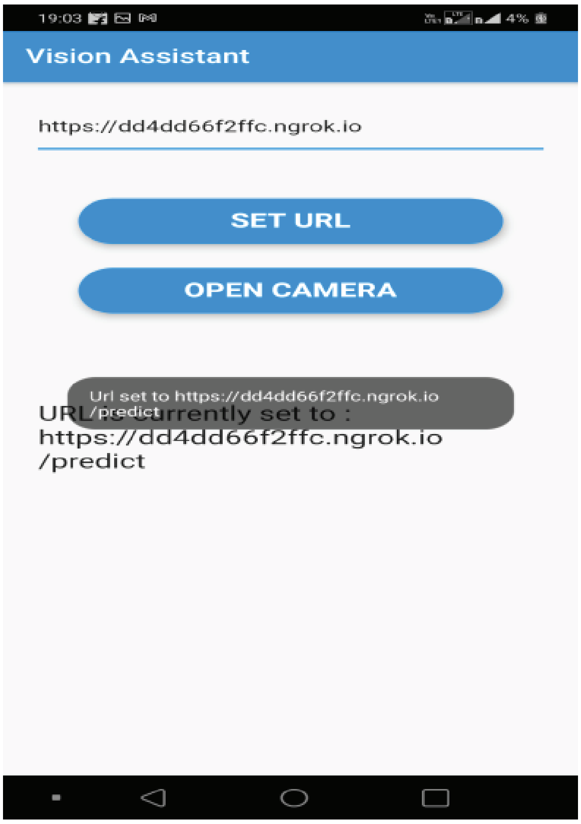
Since the application is made for visually-impaired individuals, it is obvious that a very rudimentary user interface (UI) is sufficient.

Based on this, we created a simple and lucid UI that is being used only for demonstration purposes (Refer to Figs. 8–10). 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 are shown in Figure 11. The mobile application can be found at [25]



URL is currently set to : Please set URL

Figure 8. Home screen on client application



https://dd4dd66f2ffc.ngrok.io

Url set to https://dd4dd66f2ffc.ngrok.io /predict

URL is currently set to : https://dd4dd66f2ffc.ngrok.io /predict

Figure 9. Setting the HTTP mirror address on the client application

and the server. The response time on average was found to be around 5 s. The upper bound response time is also around 5 s.

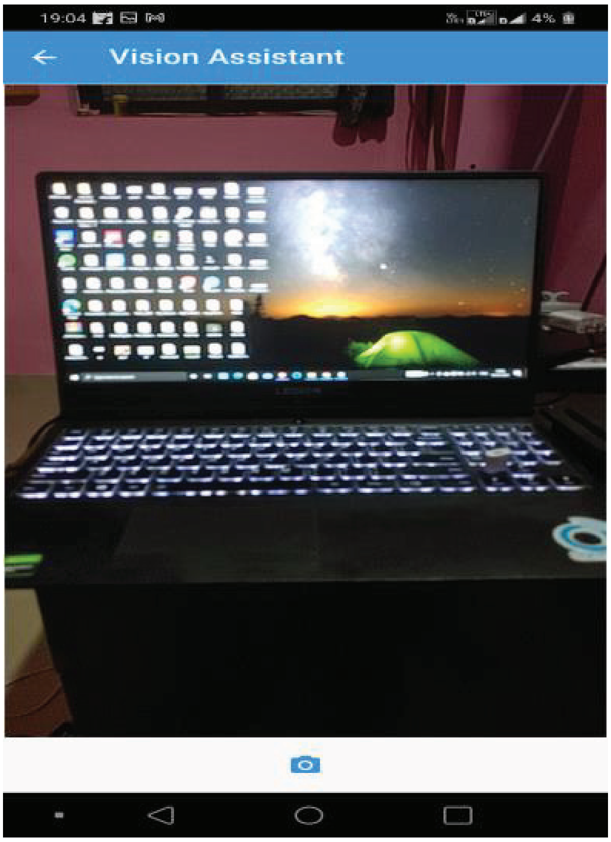
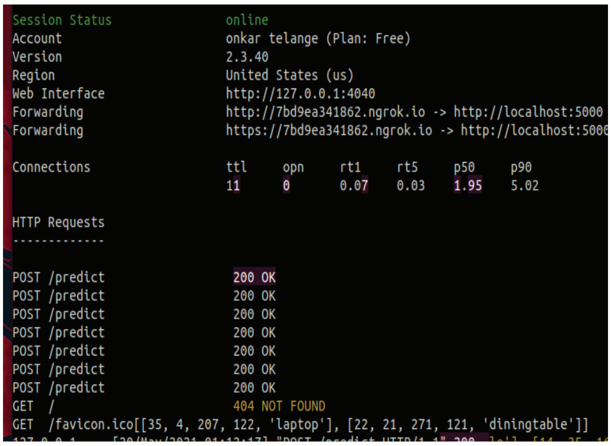


Figure 10. A camera on client application



```
Session Status                online
Account                       onkar telange (Plan: Free)
Version                       2.3.40
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://7bd9ea341862.ngrok.io -> http://localhost:5000
Forwarding                    https://7bd9ea341862.ngrok.io -> http://localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              11      0       0.07    0.03    1.95    5.02

HTTP Requests
-------------

POST /predict                  200 OK
POST /predict                  200 OK
POST /predict                  200 OK
POST /predict                  200 OK
POST /predict                  200 OK
POST /predict                  200 OK
POST /predict                  200 OK
GET  /                         404 NOT FOUND
GET  /favicon.ico[[35, 4, 207, 122, 'laptop'], [22, 21, 271, 121, 'diningtable']]
```

Figure 11. Local server running on a local machine

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 work with the scenic view of generic images. A wide-scale comparison among the various object detectors has encouraged us to use YOLOv3, an incrementally modified form of YOLO, as the object detector. The accuracy and mAP score of YOLO is above par with most contemporary detectors, and for another reason, YOLO is simpler in implementation, allowing the simple and robust construction of an object detector. We have discussed the proposed architecture, i.e., a client-server model along with the various necessary components. The modular approach has enabled us to achieve a great average response time of 5 s.

7. Conclusion and Future Work

People who are visually-impaired struggle to move safely and independently, which prevents them from participating in routine professional and social activities both indoors and outdoors. They also have distressing recognition of basic environmental factors. This paper provides a computer vision-based system that supports blind people in navigation as well as tries to address the problems posed in the introduction section. The suggested method is examined under various circumstances and measured against competing programmes available on the Appstore. The findings show that the programme functions reasonably well under various conditions and is quicker and more effective than its alternatives. As opposed to many other applications, this one lets blind users open it by plugging their mobile phones' earphone jacks in. This makes it somewhat accessible to blind users. This application can be used for navigation in its current condition, although it has several restrictions. There are still a lot of things that need to be fixed. To account for better object detection, the application can be enhanced by combining it with the Internet of Devices' technology devices. The performance of the application will improve in the future thanks to improvements in deep learning and the YOLO algorithm. The app only works on Android phones, so it will need to be redesigned so that it can run on other platforms as well.

AUTHORS

Ratnesh Litoriya* – Medi-Caps University, Indore, India, e-mail: litoriya.ratnesh@gmail.com.

Kailash Chandra Bandhu – Medi-Caps University, Indore, India, e-mail: kailashchandra.bandhu@gmail.com.

Sanket Gupta – Medi-Caps University, Indore, India, e-mail: sanket.jec@gmail.com.

Ishika Rajawat – Medi-Caps University, Indore, India, e-mail: ishika.rajawat30@gmail.com.

Hany Jagwani – Medi-Caps University, Indore, India, e-mail: email2hany@gmail.com.

Chirayu Yadav – Medi-Caps University, Indore, India, e-mail: chirayu725@gmail.com.

*Corresponding author

References

[1] World Health Organization. "Assistive technology," *WHO*, 2018. Assistive technology (accessed Mar. 20, 2022).

[2] World Health Organization. "Blindness and vision impairment," *WHO*, 2021. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed Feb. 20, 2022).

[3] L. S. Ambati, O. F. El-Gayar, and N. Nawar. “Influence of the digital divide and socio-economic factors on prevalence of diabetes,” *Issues Inf. Syst.*, vol. 21, no. 4, 2020, pp. 103–113, 2020. doi: 10.48009/4_iis_2020_103-113.

[4] C. Guo et al. “Prevalence, causes and social factors of visual impairment among Chinese adults: based on a national survey,” *Int. J. Environ. Res. Public Health*, vol. 14, no. 9, 2017, p. 1034. doi: 10.3390/ijerph14091034.

[5] C. Albus. “Psychological and social factors in coronary heart disease,” *Ann. Med.*, vol. 42, no. 7, 2010, pp. 487–494. doi: 10.3109/07853890.2010.515605.

[6] O. F. El-Gayar, L. S. Ambati, and N. Nawar. “Wearables, artificial intelligence, and the future of healthcare,” 2020, pp. 104–129. doi: 10.4018/978-1-5225-9687-5.ch005.

[7] P. Pandey and R. Litoriya. “An activity vigilance system for elderly based on fuzzy probability transformations,” *J. Intell. Fuzzy Syst.*, vol. 36, no. 3, 2019, pp. 2481–2494. doi: 10.3233/JIFS-181146.

[8] P. Pandey and R. Litoriya. “Ensuring elderly well being during COVID-19 by using IoT,” *Disaster Med. Public Health Prep.*, vol. 16, no. 2, 2020, pp. 763–766. doi: 10.1017/dmp.2020.390.

[9] L. S. Ambati, O. F. El-Gayar, and N. Nawar. “Design principles for multiple Sclerosis mobile self-management applications: A patient-centric perspective,” 2021.

[10] Z. Zou et al. “Object detection in 20 years: A survey,” 2019. http://arxiv.org/abs/1905.05055.

[11] L. S. Ambati and O. F. El-Gayar. “Human activity recognition: A comparison of machine learning approaches,” *J. Midwest Assoc. Inf. Syst.*, vol. 2021, no. 1, 2021. doi: 10.17705/3jmwa.000065.

[12] V. Iyer et al. “Virtual assistant for the visually impaired,” *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 1057–1062. doi: 10.1109/ICCES48766.2020.9137874.

[13] R. Saffoury et al. “Blind path obstacle detector using smartphone camera and line laser emitter,” *2016 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, 2016, pp. 1–7. doi: 10.1109/TISHW.2016.7847770.

[14] A. Mohanta et al. “Application for the visually impaired people with voice assistant,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 6, 2020, pp. 495–497. doi: 10.35940/ijitee.F3789.049620.

[15] V. Sharma, V. M. Singh, and S. Thanneeru. “Virtual assistant for visually impaired,” *SSRN Electron. J.*, 2020. doi: 10.2139/ssrn.3580035.

[16] A. M. Weeratunga et al. “Project Nethra - an intelligent assistant for the visually disabled to interact with internet services,” *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, 2015, pp. 55–59. doi: 10.1109/ICIINFS.2015.7398985.

[17] N. Kumaran et al. “Intelligent personal assistant - implementing voice commands enabling speech recognition,” *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2020, pp. 1–5. doi: 10.1109/ICSCAN49426.2020.9262279.

[18] V. Kepuska and G. Bohouta. “Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home),” *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 99–103. doi: 10.1109/CCWC.2018.8301638.

[19] G. Iannizzotto et al. “A vision and speech enabled, customizable, virtual assistant for smart environments,” *2018 11th International Conference on Human System Interaction (HSI)*, 2018, pp. 50–56. doi: 10.1109/HSI.2018.8431232.

[20] R. G. Praveen and R. P. Paily. “Blind navigation assistance for visually impaired based on local depth hypothesis from a single image,” *Procedia Eng.*, vol. 64, 2013, pp. 351–360. doi: 10.1016/j.proeng.2013.09.107.

[21] M. W. Rahman et al. “The architectural design of smart blind assistant using IoT with deep learning paradigm,” *Internet of Things*, vol. 13, 2021, p. 100344. doi: 10.1016/j.iot.2020.100344.

[22] J. Redmon et al. “You only look once: unified, real-time object detection,”*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[23] J.-M. Perez-Rua et al. “Incremental few-shot object detection,” 2020. http://arxiv.org/abs/2003.04668.

[24] T.-Y. Lin et al. “Microsoft COCO: common objects in context,” 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.

[25] J. Redmon and A. Farhadi. “YOLOv3: An incremental improvement,” 2018. doi: arXiv: 1804.02767.