A Project Report

On

PERSONAL VOICE ASSISTANT

BY

GADADHAR TEEGALA 18XJ1A0519

MANISH MADDIMSETTY 18XJ1A0527

KOUSHIK YELLISETTY 18XJ1A0532

Under the supervision of

DR. SANATAN SUKHIJA

DR. YAYATI GUPTA

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

PR 301 PROJECT TYPE COURSE

ECOLE CENTRALE SCHOOL OF ENGINEERING

HYDERABAD

(MAY 2021)

ACKNOWLEDGMENTS

We are sincerely thankful to, Mahindra University for providing us the opportunity to write a Term paper in the form of a thesis on the topic "PERSONAL VOICE ASSISTANT"

We are thankful to Dr. Sanatan Sukhija, Dr.Yayati Gupta for supporting us in every aspect of this term paper. We would like to express our deep concern and gratitude to our family, friends for standing by our side in all situations

Through this Term paper, we have gained a lot of knowledge in the domains such as Computer vision and Natural language processing

Ecole Centrale School of Engineering

Hyderabad

Certificate

This is to certify that the project report entitled "**Personalised voice assistant**" submitted by **Mr.Koushik Yellisetty (HT No. 18XJ1A0532), Mr. Manish Maddimsetty (HT No. 18XJ10527), Mr. Gadadhar Teegala (HT No. 18XJ1A0519)** in partial fulfillment of the requirements of the course PR301, Project Course, embodies the work done by him/her under my supervision and guidance

(Dr. Sanatan Sukhija & Signature)

Mahindra Ecole Centrale,

Hyderabad.

Date: 30 MAY 2021

(Dr. Yayati Gupta & Signature)

Mahindra Ecole Centrale,

Hyderabad

Date: 30 MAY 2021

CONTENTS

ABSTRACT

In Today's world of Artificial intelligence, we see many new emerging domains which made human life more simple. AI brought changes that influenced many domains such as healthcare, banking, manufacturing technology, the Food industry, Autonomous vehicles, and many more. We also see many voice assistants such as Siri, Alexa, Cortona, these days they are integrated into many applications such as mobiles, vehicles, assisting patients in hospitals, augmented reality, virtual reality and these help people in communicating with others, who don't have any prior experience of speaking in that specific language, directly showing an impact on human-human interaction. Together, we will explore recent developments in the field of Machine Learning (spanning over the last decade) and combine them to build a robust application that is useful for many (not just for the seniors, also for the differently-abled).
Our main aim is to develop a personal voice assistant which helps elder people to locate their necessary belongings in a confined area and they can interact with this voice assistant to open a website, to know time and weather. Primarily this voice assistant is focused on image detection, for this, we are using the Yolov3 model for detecting objects and it specifies the position of the intended object which is meant to be located by the user. Initially, the voice input is taken from the user, and after processing the image and the audio is given as output, specifying the position of the object of interest

1) INTRODUCTION

As we all know that there had been many advancements in the domain of machine learning and Artificial intelligence, which also showed an impact on many Voice Assistants. In the early stages of developing voice assistants, the precision rate was very low, but nowadays due to advancements in computations, many libraries increased the precision rate tremendously. Voice assistants Perform various activities by turning on appliances such as tv, fans, and lights, making living simpler. Voice assistants are more helpful for People with visual impairment and having problems such as dementia in helping them find the location of the object required. They also assist us in getting to know about the weather conditions, current time, and help us in many more daily activities. In general, we limit our level of understanding of voice assistants to Siri, Alexa, or google assistant. But there is a lot more to learn about them.

Some of the recent works include," Personal assistant with voice recognition intelligence [8]"[1].Our application cannot work without the internet whereas their application doesn't require any internet connection. Their application is just based on Speech recognition but our model is integrated with both speech recognition and object detection, so rather than just limiting to the basic commands (greetings, finding whether time) it will help in finding objects in a confined area. Our model has a language barrier that is it can only understand English whereas their application doesn't have any language barrier. Our application doesn't have any structured UI so it can't be deployed, whereas their application can be used in android.
Some of the research papers included in this paper [3], [4]

[1] https://www.ripublication.com/irph/ijert_spl17/ijertv10n1spl_80.pdf

The second term paper [2]is about “Jarvis”, a personal voice assistant which is capable of performing commands like a Search engine with voice interaction, It helps in medical diagnosis with medical aid, It has an inbuilt reminder, to-do application. It also has a vocabulary app to show meanings and correct spelling errors. This application even contains a chat system to interact with the user. Compared to our application, the Jarvis model contains more features that we intend to add in our future work but it lacks in the area of computer vision where we will be able to find the location of an object with a simple voice command. You can find their application in the below Github link[3]

The third work similar to our project is the “International Research Journal of Engineering and Technology (IRJET)” voice recognition system [4], it will process the user's voice identifies and produces a proper computer-readable format, whereas in our application we also do a similar process in addition that we also produce a voice output where user can hear the final output. In our application, we process the image and specifies the location of the user’s object of interest whereas in the IRJET voice recognition system we won’t get to this feature. IRJET is similar to many voice assistants like Siri, Alexa. Some of the research papers included in this paper [3], [4],[5], [6], [7]

We developed a voice assistant ETE(Eye For The Elderly) which is synonymous with elderly people with vision impairment. This assistant will help them to locate their object of interest. ETE is built using various Python libraries and frameworks. The implementation process is divided into three phases, the first phase is to record the users audio via a microphone and store that in “WAV”(wave file) which is done using python library Pyaudio, the audio in the WAV format is converted into text using wit.ai and after preprocessing steps the intent found is given to the yolo model to identify the object required, after finding the object location the Pyttsx3 module will output the location in the form of audio.

1.1 PROBLEM STATEMENT

Our problem statement is to Develop a Voice Assistant for object recognition, that can understand user commands to find a lost/misplaced items and responds with the object(s) location

2) BACKGROUND AND RELATED WORK

There are many voice assistants for elder people like LifePod, Ask my Buddy which works with amazon’s Alexa, but none of them cannot find the object position within user surroundings that the user requests. The reason to add this feature is to assist people who suffer from dementia and elderly people who have memory issues. ETE helps them to find the object of interest. This project aims to develop an AI-based robust application where we integrate a voice assistant with a model that performs object detection and gives the location of the objects in a confined area. To achieve this, we integrated the existing NLP interface and frameworks -Wit.ai, Pyaudio, Fuzzywuzzy, Esrgan model (for image super-resolution), and Yolov3 (for object detection)

---

[2] https://files.gitter.im/COSS-Jarvis/community/nUVs/JARVIS-_-Report-_2_.pdf

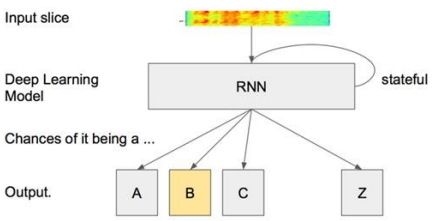
[3] https://github.com/Harkishen-Singh/Jarvis-personal-assistant

[4] https://www.irjet.net/archives/V7/i4/IRJET-V7I4657.pdf

2.1) SPEECH RECOGNITION:

(Fig 2.1.0) Image source [5]

The speech recognition model contains a recurrent neural network (RNN), where the current state influences the next state i.e for language tasks each letter influences the likelihood of the next letter from the audio input. RNN needs to be trained on millions of voice samples to get precise and accurate results, so it is better to go with a pre-trained model.
To dive deep into speech recognition refer to the link [6]



Demonstrates the working of a speech recognition model in nutshell (Fig 2.1.1) Image source [7]

There are many pre-trained models available, Wit.ai is one such interface that provides a speech recognition model. Its speech recognition model gives pretty accurate results compared to other models. It is easy to implement and unlike other speech recognition APIs, it is completely free and provides unlimited API calls per day. To convert the user's speech to text, speech recognition API requires an audio file of the user's speech. To record the audio via a microphone, the Pyaudio module is used where one can access and record the audio via a microphone, then save it in an audio file format. This file is given as an argument to the wit.ai speech recognition API to covert the speech to the text. To Know more about wit.ai refer to the link [8]

---

[5] https://nordicapis.com/5-best-speech-to-text-apis/

[6] https://realpython.com/python-speech-recognition/

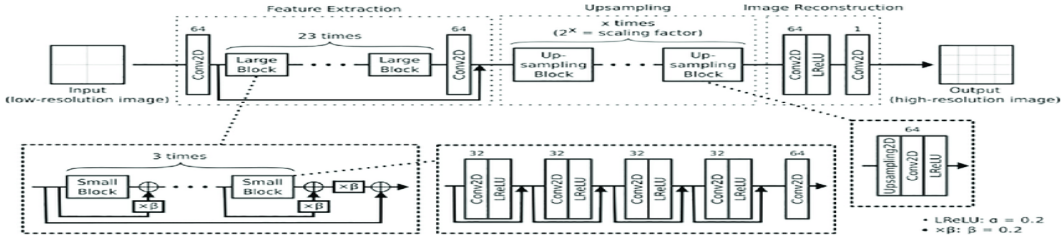
[7] https://www.liip.ch/en/blog/speech-recognition-with-wit-ai

[8] https://blog.codingblocks.com/2017/speech-recognition-using-wit-ai/

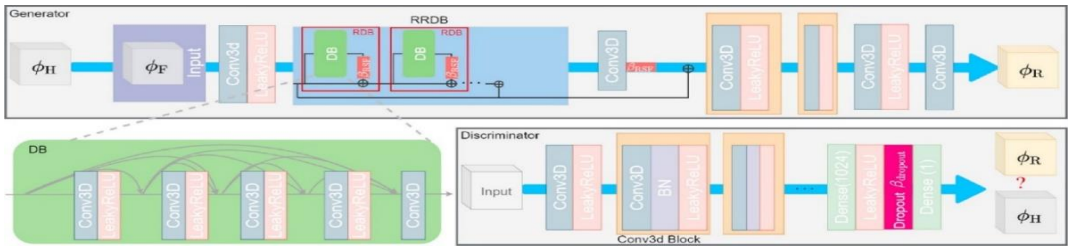
2.2) IMAGE SUPER-RESOLUTION PACKAGE

The main reason to use this package is to increase the resolution of the input image which increases the number of features in that image for accurate detections in the Yolo model. This package contains 3 default models a basic model of residual dense network(RDN) with small and large psnr and an advanced gan-based model, we used the 3rd model (Gan-based model) with RRDN being the general building block for enhanced super-resolution generative adversarial network (ESRGAN) and the weights are trained on the DIV2k dataset. This package is cloned from the Github link[9], refer to that link for more details ESRGAN uses relativistic discriminator and Adam Optimizer. During the training process, a high-resolution image is down-sampled, and a low-resolution image is given as input to the generator of ESRGAN. The generator up-samples the low-resolution image to a super-resolution image and the discriminator classifies the high-resolution image (real images from a database or any source) and super-resolution image (generated by the generator). The computed adversarial loss is back-propagated to train both the generator and discriminator when the. The basic architecture of GAN is shown below fig 2.2.1.
To dive deep into ESRGAN [1] refer to the research paper link [10]

basic Gan architecture showing working of both Discriminator and generator simultaneous (Fig 2.2.1) Image Source:[11]



Numbers on each Conv layer indicate the number of filters in it, Large Block is RRDB. small block is RDB (Fig 2.2.2)Image source:[12]



The Architecture of SRGAN consisting of both Generator and relativistic discriminator (Fig 2.2.3) Image Source [13]

---

[9] https://github.com/idealo/image-super-resolution#additional-

[10] https://openaccess.thecvf.com/content_ECCVW_2018/papers/11133/Wang_ESRGAN_Enhanced_Super-Resolution_Generative_Adversarial_Networks_ECCVW_2018_paper.pdf

[11] https://www.geeksforgeeks.org/super-resolution-gan-srgan/

[12] https://www.researchgate.net/figure/The-architecture-of-the-ESRGAN-generator-8-ie-the-deep-neural-network-used-in-the_fig4_342616974

[13] https://www.sciencedirect.com/science/article/pii/S1540748920300481

2.2.1) Network Architecture:

Generator

The generator shown in fig 2.2.3 uses convolution 3D layers in a combination with the Leaky-ReLU activation function. The convolution layers will extract the complex features. Leaky-ReLU activation removes issues such as vanishing and exploding gradients that occur during backpropagation. RRDBs are used in the initial layers of the generator. The residual in the residual dense block shown in Fig 2.2.3 (RRDB) uses residual dense blocks (RDB'S) shown in Fig 2.2.3 as its fundamental blocks with skip functions containing deep connections inside them. There are 23 such RRDBs in this generator and each RRDB is made of 3 RDBs. Each RDB(DB Dense Block) contains both convolutional and leaky Relu layers.RRDB being a deep network with many residual connections is capable of learning very complex features and information. The output of each RRDB is given to the Up-Sampling block repeating x times where 2^x is the scaling factor shown in Fig 2.2.2, After this process the image is reconstructed

Relativistic Discriminator

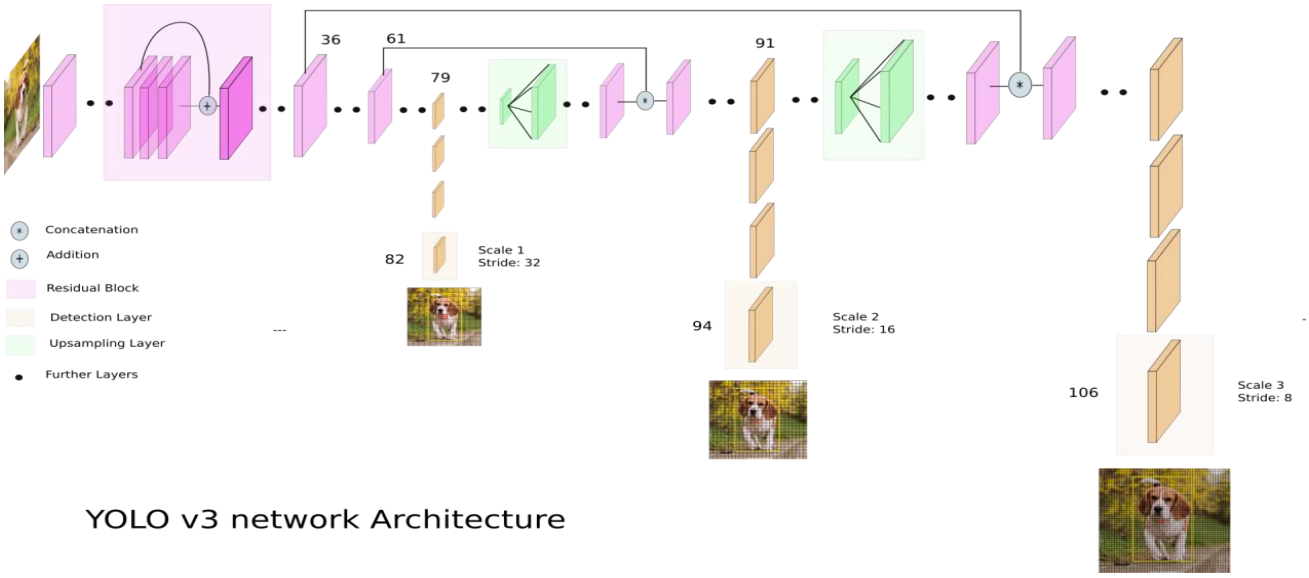
The discriminator in Fig 2.2.3 contains the basic Convolutional Neural Network Architecture. It contains a convolutional 3D layer one without batch normalization and the other seven layers with batch normalization. The layers which are very close to the input given learn fewer complex features whereas the layers far away from the input and near to the output learn high complex features. The no of feature maps increases as the network grows. The last block contains 4 layers dense 1024, leaky Relu, drop-out layer, and dense. The high dimensional outputs from previous layers are projected onto the dense 1024 flat layer, and the output of this layer is passed to the dropout layer, which reduces overfitting. **β** probability is used as the drop-out probability shown in Fig 2.2.3. Importantly ESRGAN uses a relativistic discriminator which is more enhanced than the actual discriminator in gan.

2.3) YOLOv3

YOLO stands for "You Only Look Once" and YOLO-V3 uses convolutional neural networks for object detection. Yolo v3 is one of the best object detection algorithms and detects small objects better than its previous versions. For an image, Yolo applies a single neural network, and converts it into grids and generates probabilities and bounding boxes for each grid, covering the entire image, and picks out the best one which has maximum probability. In the yolov3 paper, the author presents a new architecture called Darknet-53, this architecture consists of 53 convolutional layers and each is followed by Residual Blocks, Skip connections, Up-sampling, and Leaky-Relu activation. Below is the architecture of Yolo v3 Fig 2.3.1

To dive deeper into the Yolov3 [2] model refer to the research paper [14]

[14] YOLOv3.pdf (pjreddie.com)



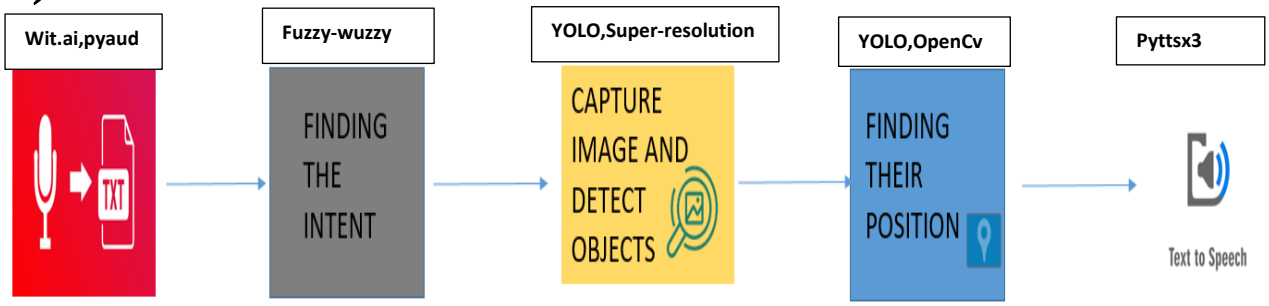
Yolo v3 Architecture (Fig 2.3.1) Image Source [15]

2.4) PYTTSX3

Python's pyttsx3 module is a text-to-speech library. This model is very easy to implement and provides different output voice variations and even works offline. The limitation of this module is, it only supports English.

To know more about Pyttsx3 please refer to the link [16]

3) IMPLEMENTATION



Flow chart of the different modules present in the overall model (Fig 3.1)

In this section, we describe our implementation strategy in detail. The overall model can be divided into 5 modules – speech-to-text, processing of text, single-image-super-resolution, Object detection, text-to-speech. The 3.1 figure depicts the entire process of ETE

---

[15] https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[16] https://pyttsx3.readthedocs.io/en/latest/

Demo of the Application can be found in the below GitHub link:

https://github.com/gad69/YOLOV3-ASSISTANt

Please Download the weights from the below

Link:https://drive.google.com/file/d/1MGc40mKZzZlMImaznn_o_1XPhgB25Cf6/view?usp=sharing

3.1.1) SPEECH - TO - TEXT :

The first task of ETE is to convert the user's speech to text format and process it. The user's speech is converted to text using the Wit.ai interface and Pyaudio module. To use wit.ai we created a new app on their website which provides a server access token, later this token is used as an API key for speech recognition. Python's Pyaudio module is used to record the user's audio via a microphone and store that in ".wav"(wave file) format for later use. Then we use wit.ai's speech recognition API to convert the recorded audio file to text format.

3.1.2) PROCESSING OF TEXT:

NLU (Natural language understanding): Text is tokenized and stored in a list, by using the spacy framework's English language model (en_core_web_sm) we remove stop words from the list. By using conditional statements, if words in the tokenized list match with the object's list (a list containing all the possible objects), then that is the required object. If the text obtained is not precise because of the noise or words containing vocabulary mistakes, then to find the most accurate object, we used the fuzzy-wuzzy module which calculates the matching percentage between two strings using the Levenshtein distance. We applied a fuzzy-wuzzy module to the object's list and we calculated each token's matching percentage by comparing it with every word in the object's list, then the word with a maximum matching percentage is taken. This is the name of the object/item the user wants to locate

Command to detect the location of the Object

```
Listening...
Finished recording.

You said: where is my bottle
['bottle']
the item you're looking for is: bottle
```

Greetings command

```
Listening...
Finished recording.

You said: hello
['hello']
greetings
12
Hello Sir. Good afternoon
Hello Sir. Good afternoon
```

Command to find the current Weather in a city or country

```
Listening...
Finished recording.

You said: current weather in hyderabad
['current', 'weather', 'hyderabad']
It is 28.98celsius and Clouds in hyderabad
```

Command to find the current Time

```
Listening...
Finished recording.

You said: what is the time
['time']
Current time is 23 hours 5 minutes
Current time is 23 hours 5 minutes
```

Command to Open Website

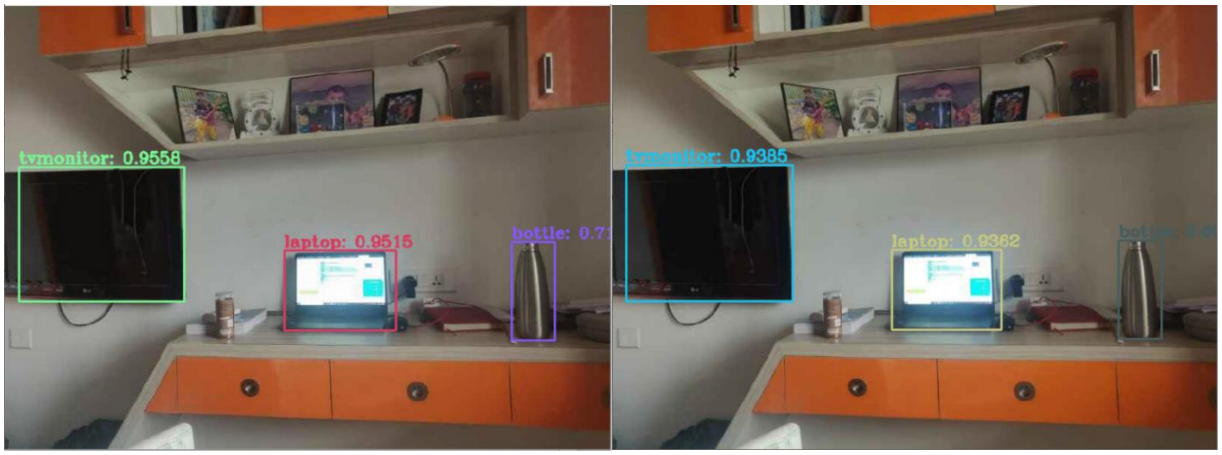
```
Listening...
Finished recording.

You said: open netflix
['open', 'netflix']
netflix
The website you have requested has been opened for you Sir.
```

3.2) SINGLE IMAGE SUPER-RESOLUTION:

The image is given as the input to the ESRGAN model and this model outputs a super-resolution image and which is sent as input to the Yolo model for detecting the objects. The network architecture and the working of ESRGAN have been described in the background work section 2.2. Below are the results with and without super-resolution. Below are the probabilities mentioned with and without super-resolution. Super-resolution acts as a catalyst to our application. It helps to increase the no of features of each cell in an image which helps the Yolo model in detecting the objects in a better way. Below are some live examples that are taken from our custom data set, the images in the left column indicate the detections of Yolo after applying super-resolution to the input image, and the right column indicates the detections without applying super-resolution.

With super-resolution

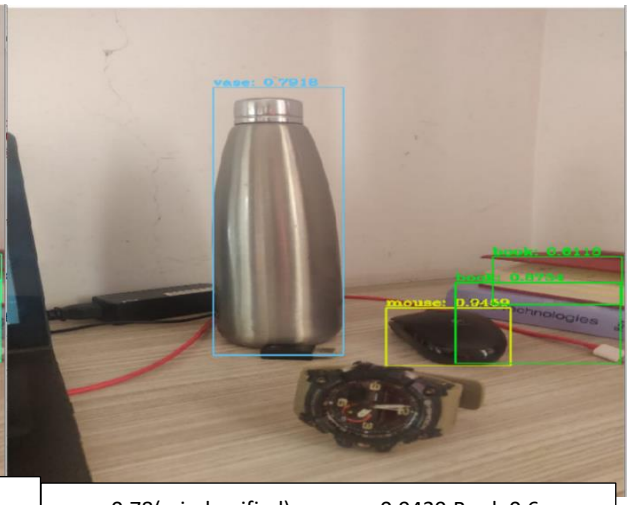
Without super-resolution



Tv monitor:0.9558 ,Laptop: 0.9515,Bottle:0.71

Tv monitor:0.9385 ,Laptop: 0.936,Bottle:0.6

Bottle:0.78 ,mouse: 0.9435,Book:0.59



vase:0.78(misclassified) ,mouse: 0.9439,Book:0.6

Chair:0.98 Bottle:0.9921,0.77

Chair:0.97 Bottle:0.9914,0.62

In the majority of the cases, the probability of detecting the objects is slightly more with super-resolution.

3.3) OBJECT DETECTION:

In this model Yolov3 object detector network is imported with the help of 'dnn' library from Open Cv. It can detect multiple objects in one image and it also predicts the labels of the classes with which it is trained and provides the location of the detected objects. Yolo predicts bounding boxes and confidence for every object and the probability of included object in it. Through non-maximum suppression, it excludes the bounding boxes which are having less confidence than the specified limit. YOLO-V3 has 53 convolution layers with each layer accompanied by Leaky ReLU activation and batch normalization.

The super-resolution image generated by ESRGAN is given as input to the object detector.The object detector is described below

3.3.1) MODEL :

The object detection model we used is a YOLOv3 model. We used NumPy(source: [17]) and OpenCV(source: [18]) libraries for drawing bounding boxes, displaying an image with text labels, and calculating probabilities. The YOLOv3 model uses the concept of blob .is created and a forward pass is implemented with the blob. We used the OpenCV function blobFromimage() that generates a 4-dimensional blob of the input image is normalized. .After blob is generated, labels and yolo v3 network is loaded by OpenCV deep learning library by using function readNetFromDarknet() with the required configuration and weights file. Through getUnconnectedOutLayers() we get the required layers(yolo 82, yolo 94, and yolo 106) and these layers are used for forward pass. After that, the image is processed by a non-maximum suppression technique through which it filters the weak predictions. The YOLO data format consists of top left corner coordinates of the bounding box and its current width and height and scaling of bounding box coordinates to draw bounding boxes with labels around detected objects using the method tolist(). By Comparing the distance between the intent object and detected object(s), it specifies the position of the object of interest concerning the nearest detected object. This model will detect any object present in the coco dataset

3.3.2)DATASET PROBLEMS AND ETE LIMITATIONS

For now, we are limiting to objects in the coco dataset, which consists of only 80 classes(labels). This Data set doesn't have all the everyday objects and furniture items so, we are planning to expand our dataset by adding classes related to them. For example, Watch is not present in the coco data set so it can't detect this item. Our model requires an active internet connection. Only works for images, not for videos. COCO dataset contains 91 common object categories with 82 of them having more than 5,000 labeled instances. In total the dataset has 2,500,000 labeled instances in 328,000 images. You can find more details regarding the Coco data set [9]. Below is a list of Object of interest

Furniture Items	Everyday Items
TV unit	Bottle
Chair, sofa	Mouse, Keyboard
Dining table	Laptop, Book

3.4) TEXT-TO-SPEECH

This is the final step of a voice assistant, this is performed using the Pyttsx3 module. Create a voice engine that is the object of the pyttsx3 class, then need to set the type of voice that needs to be the voice assistant's output voice. Then pass the text as an argument to the engine. say(), which gives the voice output.

---

[17] https://numpy.org/doc/stable/

[18] https://docs.opencv.org/master/index.html

4) RESULTS (p = Probability of that object)

We have tested our Yolo model on our custom data set. Below are some of the real-life scenario detections where a bounding box is created around the object of interest. We considered two categories Furniture items(chair, dining table, sofa, Tv) and everyday items(bottle, mouse, laptop, book, keyboard) from the coco data set. You can find our custom data set in the below Github link[19].

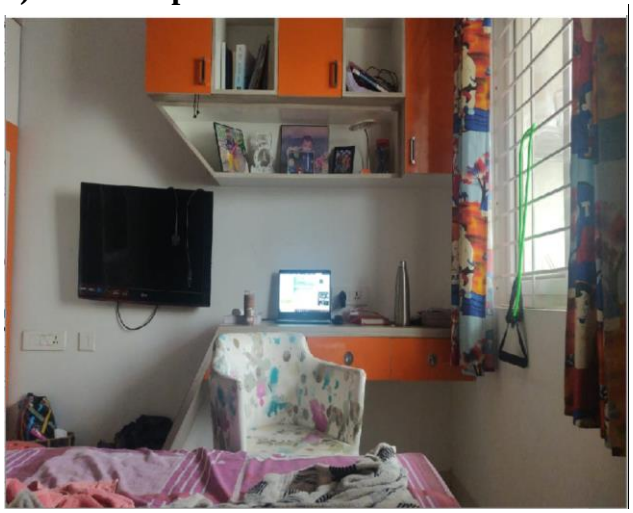
Sample Voice input: Where is my bottle

1)Final output:

2)Final output:

Voice output: `bottle is near to sofa` **(p=0.587)** Voice output: `bottle is near to bed` **(p=0.65)**

3) Final Output: Detection Failed Case



4) Final Output

(Due to the low resolution of the image bottle is not detected)

Voice output `no object detected` **(p=0)** Voice output: `bottle is near to mouse` **(p=0.80)**

---

[19] https://github.com/gad69/YOLOV3-ASSISTANt

5) CONCLUSION AND FUTURE IMPROVEMENTS

ETE is designed to help elderly people and people suffering from diseases like dementia which works on their voice commands. As of now, it works only with an active internet connection. So it takes our voice command as the input and actively responds to users it is very helpful for the blind. In this model, we are planning to use the most appropriate dataset to detect more no of household items. Further, we are planning to use live video through a camera for instant object detection and more accurate predictions. We are also planning to change the final part of the model. The current model we use is just a distance-based system for specifying the position of the object of interest. The next approach we want to try out involves two models. The first model is the same model to detect the objects needed. However, the second model has a different purpose. It will be trained to detect furniture objects like beds, tables, and such, i.e.; places where you would generally find your things like keys, bottles, etc. We also wanted to develop a user interface (UI) for this model which could make the model more user-friendly.

6) REFERENCES

[1] Wang, Xintao, et al. "Esrgan: Enhanced super-resolution generative adversarial networks." *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018.

[2] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[3] Accurate and compact large vocabulary speech recognition on mobile devices," in INTERSPEECH. 2013, pp. 662–665, ISCA.

[4] Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in ICML, 2006, pp. 369–376.

[5] M. Fishbein and I. Ajzen, Belief, Attitude, Intention and Behavior: An Introduction to Theory and Research, Addison- Wesley Publishing Company, Inc.: Reading, 1975.

[6] DOUGLAS O'SHAUGHNESSY, SENIOR MEMBER, IEEE, "Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis" proceedings of THE IEEE, VOL. 91, NO. 9, SEPTEMBER 2003

[7] . Arriany A. A., Musbah M. S. Applying voice recognition technology for smart home networks Engineering & MIS(ICEMIS), International Conference on. – IEEE, 2016. – C. 1-6.

[8] Kulhalli, Kshama V., Kotrappa Sirbi, and Mr Abhijit J. Patankar. "Personal assistant with voice recognition intelligence." *International Journal of Engineering Research and Technology (IJERT)* (2017).

[9] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." *European conference on computer vision*. Springer, Cham, 2014.