EYE FOR THE VISUALLY IMPAIRED – YOLOV3 OBJECT DETECTION WITH A VOICE

Koushik Yellisetty
Department of Computer Science
Rutgers, The state university of New Jersey-New Brunswick
NetID: ky278

Kautilya Atul Joshi
Department of Computer Science
Rutgers, The state university of New Jersey-New Brunswick
NetID: kj474

Ralla Jashwanth Yadav
Department of Computer Science
Rutgers, The state university of New Jersey-New Brunswick
NetID: jr1756

Abstract— In today's world of Artificial Intelligence (AI), we see a rapid development of new domains, which make human life simpler. AI brought changes that influenced many domains. Specific to our project, computer vision has transformed multiple sectors such as healthcare, banking, manufacturing technology, Food industry, Autonomous vehicles, and many more. Object detection is a crucial application of computer vision in various domains and has the potential to improve human lives significantly. Our project focuses on developing an object detection application that utilizes recent developments in Deep Learning and it combines them with further software upgrades to build a robust and user-friendly solution.

Our application aims to help people with visual impairments, dementia, and other disabilities to navigate their surroundings more efficiently. We achieve this by integrating the YOLOv3 object detection model, which detects and locates objects accurately and efficiently. The application is designed to be end-to-end, allowing it to be customizable to any dataset and making it more user-friendly. Our application also includes a voice recognition system that allows users to interact with the application using voice commands, making it more accessible for individuals with disabilities such as limited mobility.

By developing a robust and reliable object detection application, we can improve the quality of human interaction with the surrounding environment. Our work contributes to the field of AI by exploring the potential of object detection in enhancing human lives.

Keywords- Computer Vision, Object Detection, YOLOV3, Visual Impairment aid, Convolutional layers, Non- max suppression, Hyperparameters, Anchor Boxes, Feature extractor

1. INTRODUCTION

Dementia and loss of vision are serious health issues affecting a significant portion of the entire human population. These conditions can make daily tasks challenging and negatively impact the quality of life of those who are affected. While there are assistive technologies available, many of them can be expensive to acquire, or difficult to use.

Our project aims to address the challenges faced by individuals with visual impairments, dementia, and other disabilities by leveraging the power of Deep Learning and computer vision. We have developed an end-to-end system that uses the YOLOv3 object detection model to accurately and efficiently detect objects, locate them, and convey the location of the object along with its surrounding to the user in a coherent voice (and text). Along with using and training the YOLOV3 based model, our solution incorporates voice recognition technology, allowing users to provide voice input to the system, which can then identify the intended object and provide its precise location via a voice command. With these features, our application aims to make navigating one's surroundings more accessible and user-friendly for individuals with zero or minimal vision.

Our system has several unique features, including the ability to draw bounding boxes only around the intent object and measure the distance between the intent object and other detected objects. By providing precise location information and voice output, our system helps users to better navigate their surroundings and interact with objects. We believe that our system has the potential to significantly enhance the quality of life of individuals with dementia and ocular issues, by providing them a reliable and accurate tool for improving their independence and overall well-being.

2. RELATED WORKS

In recent years, significant advancements have been made in the field of computer vision and natural language processing, leading to the development of various intelligent personal assistants that can perform a variety of tasks through voice commands. One such technology that has gained popularity is YOLOv3[1] - a real-time object

detection model that is capable of identifying and localizing multiple objects in an image or video stream.

Compared to other voice assistants, such as the "Personal assistant with voice recognition intelligence"[2], YOLOv3 offers the additional capability of object detection, which can help individuals with dementia and vision problems to easily recognize and locate their personal belongings or identify their close ones. However, there are still limitations that need to be addressed, such as language barrier and the lack of a structured user interface in such systems.

The related work discussed in this section provide the basis for the approach we have taken in this project and address the limitations mentioned above.

3. PROBLEM DEFINITION

The problem we aim to address in this paper is the challenge faced by individuals with visual impairments, dementia, and other disabilities in navigating their surroundings and interacting with objects. Existing assistive technologies can be expensive or difficult to use and may not provide accurate or reliable information about object locations. This can make daily tasks challenging and negatively impact the quality of life of those who are affected. Our goal is to develop an end-to-end system that leverages the power of Deep Learning and computer vision to accurately and efficiently detect and locate objects, and then notify the user about the location along with its surrounding as context via a voice command (and text too).

4. BACKGROUND

4.1 YOLOv3

The YOLOv3 architecture is a neural network based on a darknet-53 architecture that uses convolutional layers to extract features from input images. The network predicts bounding boxes, confidence scores, and class probabilities for objects in the image. It does this by performing a series of operations, including convolution, batch normalization, leaky ReLU activation, and max pooling. The generic expected input shape for YOLOv3 is a 416x416 pixel image. The output of the final layer is a tensor representing the predicted bounding boxes, confidence scores, and class probabilities for each grid cell in the input image.

The YOLOv3 model outputs a tensor of shape (N, grid_size, grid_size, num_anchors * (5 + num_classes)), where N is the number of input images, grid_size is the size of the grid used for object detection, num_anchors is the number of anchor boxes per grid cell, 5 corresponds to the bounding box coordinates and confidence score, and num_classes is the number of object classes.

The output tensor is reshaped into a grid with the following dimensions: (N, grid_size, grid_size, num_anchors, 5 + num_classes) after the image has been processed by the YOLOv3 network. Each grid cell predicts the bounding boxes for a predetermined number of anchor boxes and corresponds to a particular area in the image. The projected bounding box coordinates (x, y, width, and height), confidence score, and probability distribution over the object classes are represented by the 5 + num_classes values in each grid cell.

The YOLOv3 model outputs a tensor that contains predictions of bounding boxes, each with a corresponding confidence score and probability distribution over object classes. The confidence score represents the model's confidence in the presence of an object within the predicted bounding box, while the class probabilities indicate the likelihood of the object belonging to a particular class. The non-max suppression algorithm is applied to remove any duplicate detections and provide the final predictions.

4.2 Wit.ai interface and Pyaudio module

Wit.ai is an NLP interface that allows developers to incorporate speech recognition and text understanding capabilities into their apps. It provides a RESTful API that can process text and voice inputs in various formats, including .wav files, JSON, or URL-encoded data.

In our model Wit.ai uses the Pyaudio module to record the user's audio input via a microphone and stores it in .wav format for later use. The Wit.ai interface then takes this .wav file as input and uses its speech recognition API to convert the recorded audio file to text format.

4.3 Spacy and FuzzyWuzzy

In this component, text is tokenized and stored in a list. We remove stop words from the list by using the spacy framework's English language model (en_core_web_sm). If words in the tokenized list match with the object's list (a list containing all possible objects), then that is the required object. In case the text obtained is not precise due to noise or vocabulary mistakes, we use the fuzzy-wuzzy module which calculates the matching percentage between two strings using the Levenshtein distance. We apply the fuzzy-wuzzy module to the object's list and calculate each token's matching percentage by comparing it with every word in the object's list. The word with the maximum matching percentage is taken as the name of the object/item the user wants to locate.

4.4 Pyttsx3 module

Pyttsx3 is a Python library used for converting text into speech. It can be customized according to the user's needs, such as selecting a specific voice or adjusting parameters like pitch, rate, and volume. The

input to the module is text, which is passed as a string. Pyttsx3 generates speech output in real-time, and the output can be either saved as an audio file or played back through a speaker. The module performs several processing steps such as tokenizing, converting text to phonemes, and generating speech output using a speech synthesizer.

5. APPROACH

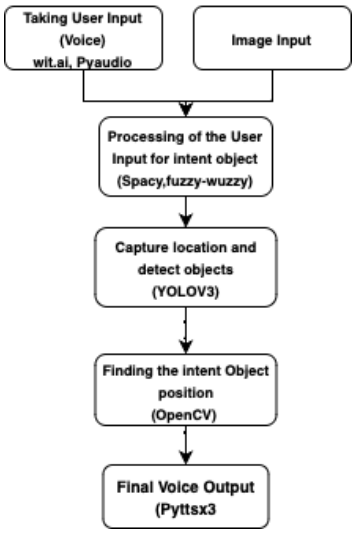

Fig. 1: System Architecture (flow)

Our approach for this project involves a series of steps that aim to provide reliable and efficient object detection and location information to users with vision problems and dementia. The first step involves taking input from the user in the form of voice and image. The user's input is then processed for the intent object using Spacy and fuzzy-wuzzy libraries. Once the intent object is determined, the information is passed to the YOLOv3 object detection model. The YOLOv3 model accurately detects the intended object and nearby objects, and provides the precise location of the intended object, which is then relayed back to the user through voice output using Pyttsx3.

We relied on a modified YOLOv3 object detection model and the NumPy and OpenCV libraries for drawing bounding boxes, displaying image labels, and calculating probabilities. The YOLOv3 model uses the concept of a blob, which is created by normalizing the input image and passing it through the network. To load the YOLOv3 network and labels, we used the OpenCV function readNetFromDarknet() with the required configuration and weights file. We then used the OpenCV function blobFromImage() to generate a 4-dimensional blob of the input image. The YOLOv3 network was implemented with the required layers (yolo 82, yolo 94, and yolo 106) and a forward pass was performed on the blob. Non-maximum suppression was applied to filter out weak predictions, and the resulting bounding boxes were drawn with labels using the method tolist().

In our modification, we obtained the central coordinates of each detected object by calculating the midpoint of the bounding box using the top left corner coordinates and width and height. By comparing the distance between the intent object and detected objects, we were able to determine the position of the object of interest relative to the nearest detected object. This enabled us to provide precise location information to the user.

Overall, our approach provides an effective solution to object detection and location for individuals with vision problems and dementia. By utilizing a modified YOLOv3 object detection model and the Spacy and fuzzy-wuzzy libraries for processing user input, we are able to provide accurate and efficient object detection and location information to the user.

6. RESULTS

We trained and validated the YOLOv3 model on the custom dataset by using Transfer Learning with Fine Tuning. On the other hand, we trained the YOLOv3 model on COCO dataset using transfer learning with no fine tuning. We found that the model accuracy on the custom dataset was worse in comparison to the model accuracy on the COCO dataset. Using the model trained on the COCO dataset, we have mentioned 4 sample test cases below and analyzed the output with respect to the generated output.

Sample Test case 1:

User Input: Where is Book

Output: Model provides the output through voice. Here is the snapshot of the output.

```
You said: where is book
model: the item you're looking for is book
Objects Detection took 0.45715 seconds
book is near to cell phone
model: book is near to cell phone
```

Fig. 2: System console: input vocal command “where is book” and output vocal command as text

Fig. 3: Input Image for identifying and location object (book)

As per fig. 2 and fig. 3, our model correctly located the object and conveyed its location to the user through a voice command. However, please note that the model confused a calculator with a cell phone. In test case 1, our system was able to identify the primary object correctly, but failed to detect the neighboring object correctly.

Sample Test case 2:

User Input: Where is laptop

Output: Model provides the output through voice. Here is the snapshot of the output.

```
You said: where is laptop
model: the item you're looking for is laptop
Objects Detection took 0.40498 seconds
laptop is near to bottle
model: laptop is near to bottle
```

Fig. 4: System console: input vocal command "where is laptop" and output vocal command as text

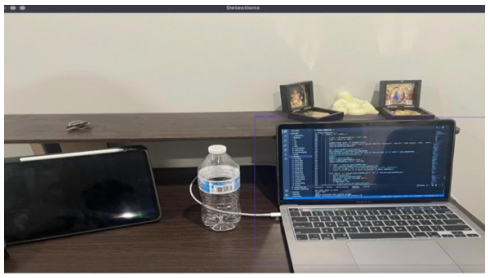

Fig. 5: Input Image for identifying and location object (laptop)

As per Fig. 4 and Fig. 5, the system correctly identified the primary and secondary objects with respect to each other and it gave the correct output.

Sample Test case 3:

User Input: Where is bottle

Output: Model provides the output through voice. Here is the snapshot of the output.

```
You said: where is bottle
model: the item you're looking for is bottle
Objects Detection took 0.41023 seconds
bottle is near to diningtable
model: bottle is near to diningtable
```

Fig. 6: System console: input vocal command "where is bottle" and output vocal command as text

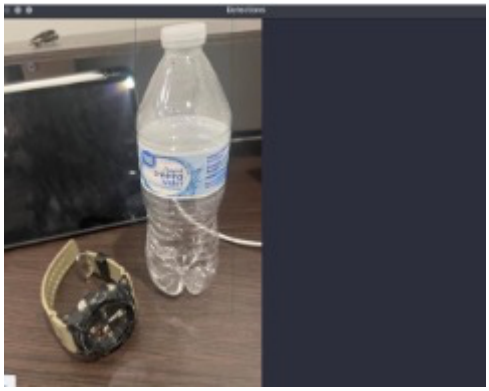

Fig. 7: Input Image for identifying and location object (bottle)

As per Fig. 6 and Fig. 7, the system was correctly able to identify that primary object in the input image. However, it confused the study table with a dining table. However, considering that the input image did not span across the entire table, the output seems reasonable.

Sample Test case 4:

User Input: Where is watch

Output: Model provides the output through voice. Here is the snapshot of the output.

```
You said: where is watch
model: the item you're looking for is watch
Objects Detection took 0.43346 seconds
model: intent object is not detected or probably the model is not trained well
```

Fig. 8: System console: input vocal command "where is watch" and output vocal command as text

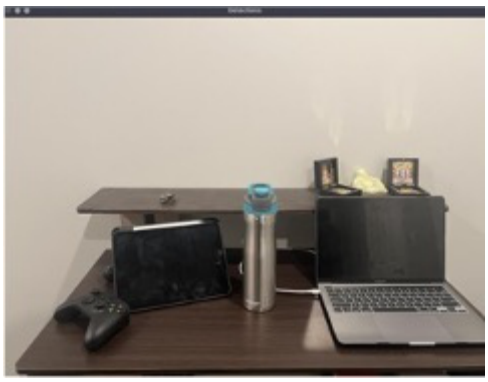

Fig. 9: Input Image for identifying and location object (watch)

As per Fig. 8 and Fig. 9, the system correctly predicts that the primary object that is to be located is not in the picture. Hence, the system passed the negative test case.

7. CONCLUSION

In this paper, we presented our object detection model that conveys the corresponding location of the object to the user via a vocal command. Firstly, we process the user's input voice command using Wit.ai and PyAudio as a speech-to-text module. Then, we use Space and Fuzzy-Wuzzy to determine the intended object that is to be located. After this, the YOLOv3 based model detects and locates the intended and other objects in its vicinity as its output. However, our modification of the YOLOv3 model yields not just the bounding box, but a coherent textual sentence stating the exact location of the intended object with respect to its surrounding. Finally, the Pyttsx3 module acts as the text-to-speech module to give the exact location of the intended object with reference to its neighboring objects as a vocal command. The above model detects, locates, and conveys the location of objects to the user in a manner which is useful for people with zero or minimal vision and thus, it facilitates their daily lives.

8. FUTURE SCOPE

The overall system presented in this paper is a computer application where the user must upload an image and then the application is able to detect and locate the intended object. Instead of just a computer application, we

can create a SoC (System on Chip) device that can be placed at any place, and thus enable our system to run at all times and with zero setup efforts to locate and notify objects as per user requests.

Our current model runs on image inputs. While we did use a video recording to create our custom dataset by processing the video and rendering images from it. However, doing this in real-time would be slow and would take time. Thus, one scope of improvement would be to create a Deep Learning model on video data which can run on real-time video. One way to do this would be to instantly take the current frame of the video as our image input and detect the object in that. However, it may be the case that the video recording apparatus might be blocked by some object or person, so we will have to search in a range of frames as input images and locate the object in those images.

9. REFERENCES

[1] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", arXiv preprint, arXiv:1804.02767 [cs.CV], Apr. 2018.

[2] Kshama V. Kulhalli, Kotrappa Sirbi, Abhijit J. Patankar, "Personal Assistant with Voice Recognition Intelligence, "International Journal of Engineering Research and Technology", ISSN: 0974-3154 on Volume: 10, Nov. 2017

[3] "Wit.ai." Wit.ai, n.d.

[4] J. Rossum and P. L. Drake, "PyAudio: Read and write audio streams in Python," Jun. 2010, [Online]. Available: https://people.csail.mit.edu/hubert/pyaudio/.

[5] J. L. Mehta, "Spacy: Industrial-strength natural language processing in Python," 2021. [Online]. Available: https://spacy.io/.

[6] M. Shubham, "FuzzyWuzzy: Fuzzy string matching in Python," 2017. [Online]. Available: https://github.com/seatgeek/fuzzywuzzy.

[7] Natesh M. Bhat, "Pyttsx3," 2021. [Online]. Available: https://github.com/nateshmbhat/pyttsx3.

[8] OpenCV Library, "OpenCV: Open Source Computer Vision Library," 2021. [Online]. Available: https://opencv.org/.

[9] Ravikumar N R, Prateek C, Sathvik Bhandar, Rahul Kumar, Mayura D Tapkire, "VIRTUAL VOICE ASSISTANT", "International Research Journal of Engineering and Technology (IRJET)", ISSN: 2395-0072 on Volume: 7 Issue. 4, Apr. 2020.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi∗, "You Only Look Once: Unified, Real-Time Object Detection, arXiv preprint, arXiv:1506.02640 [cs.CV], May. 2016.