VIRTUAL EYE FOR THE VISUALLY IMPAIRED

Aakansha Gupta[1], Harshil Panwar[2,] Dhananjay Sharma[3] and Rahul Katarya[4]

[1] Department of Computer Science and Engineering, Delhi Technological University, Delhi, India, aakanshagupta.74@gmail.com
[2] Department of Computer Science and Engineering, Delhi Technological University, Delhi, India, harshilpanwar_2k18co142@dtu.ac.in
[3] Department of Computer Science and Engineering, Delhi Technological University, Delhi, India, dhanajaysharma_2k18co119@dtu.ac.in
[4] Department of Computer Science and Engineering, Delhi Technological University, Delhi, India, rahulkatarya@dtu.ac.in

Abstract. Visually impaired people in modern society have access to a variety of aids such as volunteer helpers, service dogs, etc., allowing them to go about their daily routine as comfortably and conveniently as possible. Although effective, the above-mentioned solutions introduce an element of dependency in the lives of the visually impaired, thus, eliminating a certain amount of freedom from their lives. Through our project, we aim to create a device, that with the proper equipment, will be able to detect the objects in front of the user and inform them accordingly, thus, acting as a second eye. Successful implementation of this project will prove to be extremely helpful to the impaired by reducing that sense of dependency that they have to live under constantly due to their ailment.

Keywords: YOLOv3, Object Detection, Computer Vision, Text to Speech.

1 Introduction

Technology plays a vital role in every human's life, even when it comes to enabling them to complete many day-to-day tasks effectively and efficiently. In the case of differently-abled people, supportive technology is an extremely relevant example of how technology allows us to live our lives very comfortably. Realizing the importance of this domain of technology, this product's concept gives a comprehensive solution for the visually impaired by providing them an independent and viable alternative to their current human/animal aids that will also prove to be cheaper in the long run.

Deep learning is a modern computational field that attempts to imitate the human brain's learning process in the best possible manner using extensive mathematical equations. A successfully designed and implemented model can enable the machine to recognize patterns and make predictions accurately. Although the field is still picking up momentum, certain subdivisions have already attained accuracy levels surpassing a human expert in the field.

Object recognition, in deep learning, is a subpart of the image processing division. It involves parsing through any input frame (image or video), locating potential objects, and labeling them accordingly based on the list of objects on which the model has been trained.

Most of the detectors employ classifiers to perform detection. To detect an object, classifiers for different objects are tried on the image at various scales. Systems like deformable parts models (DPM) [2] use a sliding window approach where the classifier is run at evenly spaced locations over the entire image.
Object detection using the YOLO algorithm is posed as a regression problem, straight from image pixels to bounding box coordinates and class probabilities. [1]

The YOLO unified model has several benefits over other detection systems:

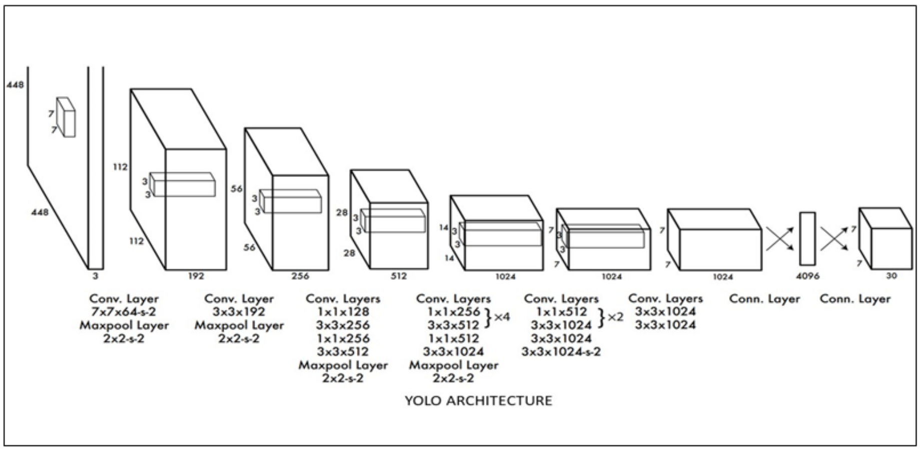


Fig. 1. The architecture of the YOLO Object Detection Network

1. It is extremely fast. Since it avoids the complex pipeline by posing the problem as regression, also, it avoids the sliding windows technique used by detection systems such as DPM, thus making it faster.

2. YOLO sees the entire image during training and test time, so it encodes contextual information about classes as well as their appearance. Fast Region-Based Convolutional Neural Network (R-CNN) [3], a top detection method, mistakes background patches in an image for objects because it cannot see the larger context.

2 Fundamentals

Image Processing is the division of Deep Learning responsible for converting input images into clean, equal-sized, well-formatted, processable images that any neural network can analyze with ease. This is an essential process, as it allows the algorithms to identify features and patterns efficiently and then use them to solve many real-world problems.

Object Detection, as mentioned earlier, is the subdivision of image processing responsible for scanning images and identifying certain objects present in them. The general approach used by more algorithms is to identify the most promising areas, identify the probability of that area being a particular object and confirming it if the probability value is higher than a particular threshold. [5]
Convolutional Neural Networks are neural networks comprising mainly convolutional layers combined with various pooling and dense layers. These networks are the most effective in image feature extraction as they are designed to identify the most relevant pixels accurately, with ease. [4]

The YOLO or the You only look once algorithm, in particular, is one of the most efficient object detection algorithms that converts the complex problem of object detection into a simple regression and bounding box probability problem. It is faster and more reliable than its competitors as it processes the entire image in one go.

3 Related Work

Today, most detectors use two-stage object detection algorithms to classify objects; this technique is slow in real-time and is extensive computation.
The paper that functioned as our guiding light throughout the bulk of this project is known as "YOLOV3: An Incremental Improvement," written by Joseph Redmon and Ali Farhadi.[1] This paper implements the YOLO network with darknet-53 as the feature extractor, further reducing the time taken by the algorithm to detect objects in real time.

Today, most assistive solutions for the visually impaired make use of external hardware for object detection, depth perception and providing feedback [6]. Through our work, we aim to rule out this hardware, thus making the solutions cost-effective.

4 Methodology

4.1 Input

Our first task for the project is to obtain the input on which our model will perform. For this, we will use the OpenCV library in python to tap into the device's cameras and obtain a photo or a video feed. This will serve as our input for object detection.

4.2 Object Detection

For object detection, we will be implementing the YOLO, or the You Only Look Once algorithm. The YOLO algorithm is one of the fastest and simplest (in terms of complexity of the neural network) for performing object detection. Vigorous research over the years has led to a rapid decrease in not only the complexity of the network structure but also the time taken by it to perform object detection.

The way YOLO excels is by converting complex object detection problems into much simpler problems involving regression and class probabilities. First, a frame is divided into a grid of a predetermined size. Now, a sliding window is used to parse through the grid, mini box by mini box, determining the probability of each mini box containing an object.

After a single iteration, all the boxes with a higher probability (say >= 0.5) of containing an object class are taken into consideration and bounding boxes are drawn accordingly. These bounding boxes will contain the detected objects, respectively.

To ensure that the same object is not detected more than once, we introduce the concept of non-maximum suppression or NMS [1]. In NMS, we consider the boxes with the highest confidence value and calculate their IOU (Intersection over Union) with other boxes [1]. The IOU is the ratio of the area common between two bounding boxes and the total area occupied by both boxes, hence the name, Intersection over Union. Now, all the boxes having a high IOU with our chosen box are eliminated as they are most likely to be representing the same object. The same process is repeated for the bounding box with the next highest confidence value until we get our final bounding boxes, representing a unique object.

As an additional contribution to this step, we have further trained our YOLO model to identify certain harmful objects such as guns that were not a part of the multiple classes already present in the COCO Dataset. [7]

4.3 Estimating Position and Depth and Output

Using the boundary box values for both static images and video files position of each object is estimated, along with depth estimation using the safety index. Using the google text to speech API, the output is given in audio format, which is overlaid and synced with the video using FFMPEG.

As part of our additional contribution to this step, a custom function that divides the image into a 3*3 matrix and identifies every object's location relative to the frame of the image was built.

Furthermore, a simple yet efficient method for depth perception was employed rather than the state of the art but computationally heavy techniques such as monocular depth perception. Each frame is assumed to be at 2*LDDV, where LDDV is the least distance for a distinct vision for the Human eye. Now for the safe distance, if the index S (Safety index) >0.6, then the warning symbol is generated for that object.

$$\textit{Safety Index (S)} = I * C \tag{1}$$

C = Confidence Score of Bounding Box
I = Area (Bounding Box) / Area (Frame)

5 Experimentation

Libraries, frameworks used:

TensorFlow. One of the most popular libraries for constructing and implementing neural networks has been used.

Darknet Framework. A 53 Layered, fully convolutional neural network has been used as the feature extractor.

Pydub. A simple, well-designed python modulo for audio manipulation.

GTTS. Google Text To Speech conversion API has been used to obtain the final output.

FFMPEG. One of the best audio/video processing libraries has been used.

OpenCV. Python's most popular library for image processing has been used.

Hardware Requirements:

- 2 Core CPU clocked >= 2.4 GHz
- 8 GB RAM
- 120 GB Disk Space

6 Dataset Description

The YOLO model used to develop our tool was trained on the COCO (common objects in context) dataset. [7] This dataset contains over 80 classes and one of the most popular datasets used for object detection. This forms the base dataset of our tool.

Furthermore, we have utilized two additional datasets containing about 300 images of harmful objects such as guns, [8][9], etc. As these datasets were not labeled, bounding boxes were constructed for every image manually, and then the model was trained on these datasets as well. Therefore, a total of two datasets were utilized.

7 Computation

A Graphics Processing Unit (GPU) is an integrated circuit configured to quickly manipulate and change memory to provide acceleration to create images in a frame buffer. These have various applications in mobile phones, personal computers, etc. They can optimally manipulate images with ease. The parallel structure of the GPUs makes them much more efficient than the CPUs for the purpose of processing large amounts of data simultaneously. Hence in order to increase computational speed and save time when performing computationally heavy tasks such as training networks, GPUs are used.

A CPU usually works on a few cores, averaging between 4-8, while CPUs with 64 and 128 cores are also commonly used in supercomputers, these usually work with one or two threads per core. There are usually a few hundred cores in a GPU, and each of these have tens or hundreds of threads, thus bringing the total to thousands of threads parallelly computing and performing tasks. A GPU works parallelly on tasks that, if performed on a CPU, are done sequentially, using a for loop while in GPU vector addition and vector operations.

CPUs have more powerful cores than GPUs; thus, they can perform better for the computationally complex task if per-core performance is considered GPUs have a more number of weaker cores that can outperform CPU when tasks can be parallelly processed, such as in big data analysis or 3D rendering.

8 Tensor Processing Unit

TPU is also an alternative to the GPU, an ML-specific ASIC, designed to speed up Linear Algebra operations, specifically heavy matrix multiplications. TPU is one of the most advanced DL platforms. It gives up to 30 times better performance than conventional CPUs and GPUs. It provides very high performance with an effective bandwidth of 12.5 Gbps.

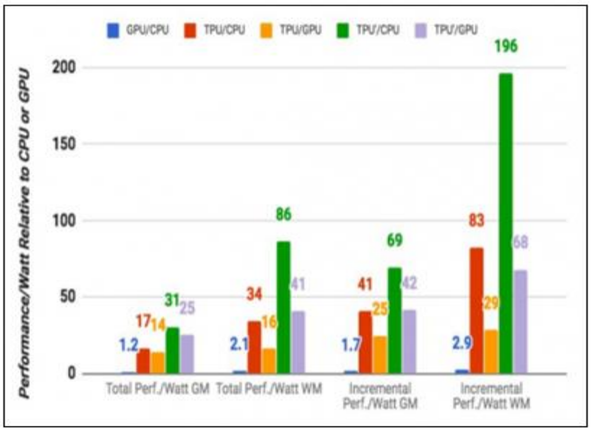


Fig. 2. Relative Performance of Different Processing Units

9 Result

9.1 Training the Model

The model was trained on the famous COCO dataset [7] containing a total of 80 classes, all common objects that one might come across in their everyday lives. Apart from this, two additional databases containing images of harmful objects such as guns, etc.,[8][9] were also labeled and used to train our model. The following images and graphs show the loss function and the how it changed throughout the process of training along with other frames captured during the live testing of the tool.

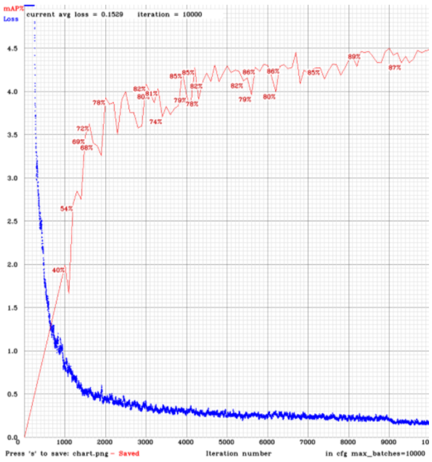


Fig. 3. Loss plot for COCO Dataset

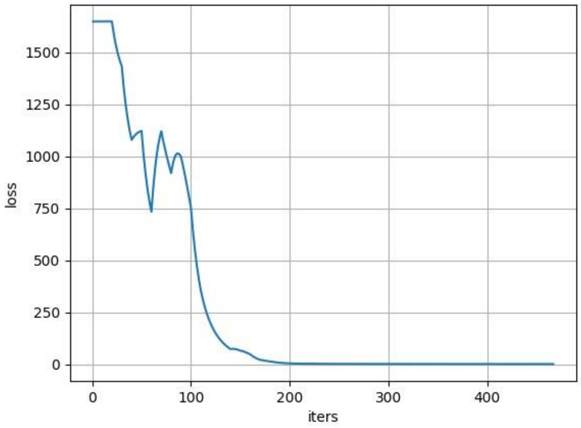
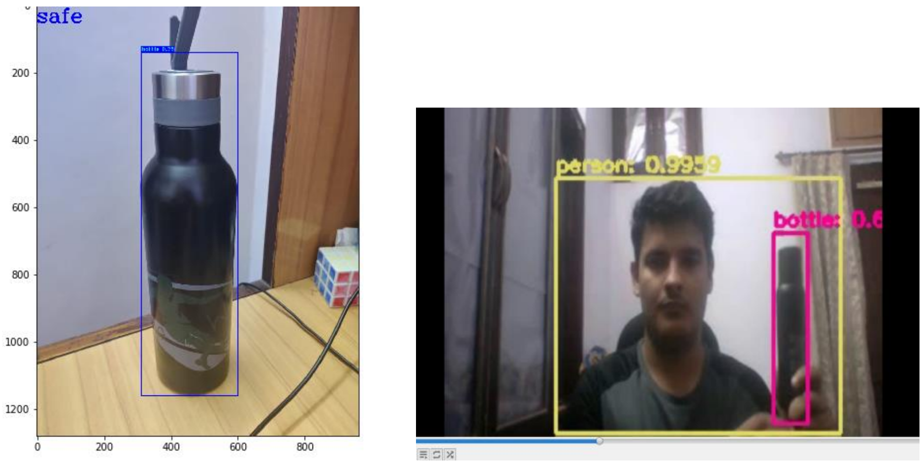


Fig. 4. Loss plot for Custom Classes

For feature extraction Darknet-53 was used. It stood out in terms of fast calculation speed and fewer floating-point operations. Below is the image of a real-time deployment of our tool.



(a) **(b)**

Fig. 2. (a) Output from a static frame (b) Frame capture from real-time

10 Experimental Result and Discussion

In our research work, we witnessed how object detection can be viewed as a regression problem and the advantages of this methodology. We trained the model with new classes, which resulted in a satisfactory mAP. Position determination and depth estimation were determined satisfactorily, leveraging the expensive computation and delayed responses.

11 Future Work

Apart from being of great use in the support and aid domain, our technology also has the potential to form the base of many other potential applications. When combined with OCR technology, the device can be used to identify objects, recognize their brands, and locate them on popular online sellers such as amazon. This technology can even be used to recognize license plates and scan documents and identity cards.

When combined with pose detection algorithms, this technology can be used in many fields such as athletics, sports, yoga, etc., to identify and analyze the various physical activities being performed by athletes and help them train.

As far as our current tool is considered, it may still be improved in terms of its accuracy and performance in real-time, making it even more viable as a probable replacement to old and traditional methods.

References

1. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640
2. Ross Girshick, Forrest Iandola, Trevor Darrell, Jitendra Malik, 2015, Deformable Part Models are Convolutional Neural Networks, arXiv:1409.5403
3. Ross Girshick, 2015, Fast R-CNN, arXiv:1504.08083
4. Krizhevsky, Alex & Sutskever, Ilya &Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.
5. Zhao, Zhong-Qiu & Zheng, Peng & Xu, Shou-Tao & Wu, Xindong. (2019). Object Detection With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems. PP. 1-21.10.1109/TNNLS.2018.2876865.
6. Saidur Rahman, Chandan Debnath, Tahmina Aktar Trisha, "Design and Implementation of a Smart Assistive System For Visually Impaired People Using Arduino", International Journal of Advances in Computer and Electronics Engineering, Vol. 4, No. 11, pp. 1-5,November, 2019.
7. Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham, 2014.
8. Sai Sasank . (2019) . Guns Object Detection, Version 1, Retrieved from https://www.kaggle.com/issaisasank/guns-object-detection

9. Shashank Shekhar . (2020) . Knife Dataset, Version 1, Retrieved from https://www.kaggle.com/shank885/knife-datasetM. Of, “AN EFFECTIVE IMPLEMENTATION OF FACE RECOGNITION USING,” J. SOUTHWEST JIAOTONG Univ. Vol.54, vol. Vol.54, no. No.5, pp. 1–9, 2019, doi: 10.35741/issn.0258-2724.54.5.29.