IT204: SS Project Report: Smart Cane for the Blind

Your Name - Roll Number
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: youremail.com

Your Name - Roll Number
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: youremail.com

***Abstract*—Should not be more than 200 words**

I. Introduction

Globally there are 285 million people who are visually impaired, and 39 million out of that figure are blind. People over the age of 50 constitute 85 percent of the blind population. In India alone there are 40 million people who are visually impaired which constitutes 20 percent to the world total. The main issue is that many people don't have access to the resources which can cure them. In most cases, the blind don't have enough money to go through the necessary surgeries especially in underprivileged cities. There are a few existing solutions for example "Seeing AI" by Microsoft uses an app which sends pictures to the cloud which then processes them. It basically detects objects and also includes text recognition which helps the user to read books etc. However, it has one major disadvantage that the users can't do anything else while they are on the app. There is also another product called Orcam MyEye2 but the cost is too high and thus is not very attractive to consumers.

We can definitely help to make life simpler for the blind using the aid of artificial intelligence and that is what our project aims to do. Our aim is to provide a "temporary vision" for them until they have enough resources required to treat their impairment. Our smart cane consists of a webcam, an object detection algorithm (YOLO) and a text to speech headset. We are capturing one frame every 9 seconds through the webcam and then detecting the objects and outputting the result in audio format. We also have an additional feature which checks if the image is blurred upon which it reads the frame again. The user can stop the input at any time he/she wants. We have successfully come up with a cost-effective solution which helps the blind to accomplish his/her daily tasks with ease and help them move freely without any guidance.

II. Literature Survey

- Paper 1: Smart Belt for Blind uses a belt embedded with ultrasound sensor which detects the obstacle. The belt also has a buzzer which vibrates when obstacle is detected. The entire system is developed in such a way that the distance calculated is sent as an audio message for the blind person, where in which he also hears the distance calculated using a speaker.
- Paper 2: A camera is used where the captured image is sent to the PC. It also has a controller to intimate the type obstacles and the distance between the obstacles and device is determined through voice. There are a few software and hardware used to design the device which is ultrasonic transmitter and receiver, amplifier frequency to voltage converter, PIC controller, camera, image processing unit, PC, RS232, LCD display, voice board and speaker.
- Paper 3: Pothole detection for visually impaired which uses a camera that captures image 15 frames per second and by the concept of image processing the pothole is detected. Drawback of the system is that a lot of images are captured per second which increases the overhead and storage requirement.
- Paper 4: Smart walking stick - An electronic approach to assist visually disabled persons which is a sensor based circuitry consisting of sensors, ultrasonic Sensor which is used to detect obstacles, a PIC16F690 microcontroller which reads these sensors and drives a buzzer, an LED and a motor with PWM. An audio output is designated by a buzzer alarm.

III. Problem Statement

Design a Smart Blind Cane to help guide the visually impaired by detecting objects and portraying the information to them in the form of speech so that they get a better understanding of their surrounding and don't need any human assistance.

A. *Objectives*

- To successfully carry out object detection and classification using yolo.

Authors	Methodology	Merits	Limitations	Additional Details Based on your project
M.F. Saaid	Smart cane with a distance measurement system which comprises an ultrasonic sensor as input and earphone as the output. Programmed using LabVIEW with FPGA as the intended target.	It detects object and reports it along with the distance of it from the user	It cannot tell which type or how large is the object.	Ultrasonic for distance measurement
P.Bhavishya	System consists of three ultrasonic sensors which help the user to detect the obstacles be directed to direction (front/right/left) which has no obstacles, when the other two directions are blocked by an obstacle.	Users will be directed to go in a direction at which the distance is longer. Mq2 gas sensor if the user is intoxicated.	Might make it long for user to reach the required destination.	Multi-directional object detection and Mq2 gas sensor
Mohammad Hazzaz Mahmud, RanaSaha, and Sayemul Islam	Ultrasonic Sensor used to detect obstacles, A PIC16F690 microcontroller reads these sensors and drives a buzzer, a LED and a motor with PWM	Buzzer alert which doesn't fail to alert in case user has poor hearing ability.	It cannot convey the user about the distance of object from him or her.	Ultrasonic sensor and buzzer alert
Monitoring V. S. M. Madulika S, M. S. Madhan Mohan,, CH.Sridevi, T. V. Janardhana rao	Development of an Electronic Travelling Aid (ETA) kit to help the blind people to find obstacle free path.	It uses ETA which makes it simpler.	Does not convey the user about the distance of the obstacle.	Usage of ETA
Unknown	Pothole detection by capturing image 15 frames per second and by the concept of image processing the pothole is detected.	Has more accuracy of detection.	lot of images are captured per second which increases the overhead and storage requirement	Captures 15 frames per second.

Fig. 1: Summary of litrature survey

- Relay the information to the user by means of sound or speech.
- Making sure that the outputs are properly given and are not too many in number to avoid confusing the user.
- Handling blurred inputs caused by the user's movement and other miscellaneous situations.

IV. Methodology

A. Architecture and Flowchart

This project explains the simulation of the image processing performed by implementing YOLOV3 algorithm in detail. The webcam of the laptop is connected to the Neural Network as an input device and the audio set as the output device. The simulation process is explained in detail with the help of the flowchart below,

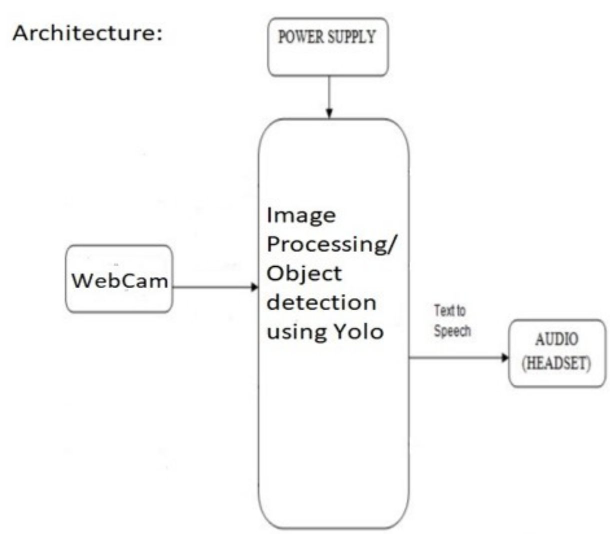


Fig. 2: Architecture of the project.

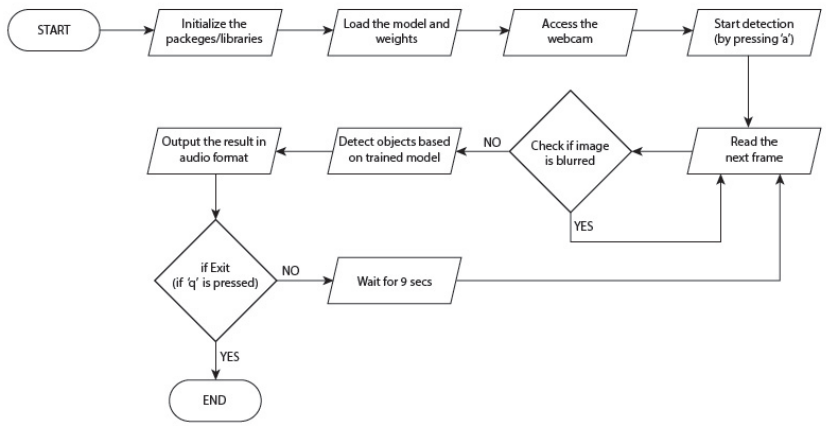


Fig. 3: Flowchart of the project.

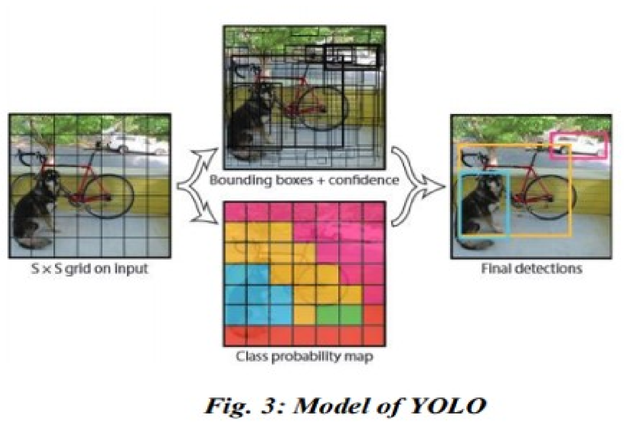


Fig. 4: Architecture of the project.

B. YOLO and TensorFlow

You Only Look Once (YOLO) is a neural network, real time Object detection system. It makes prediction with a single network evaluation unlike the other neural networks such as R-CNN and Fast R-CNN. It applies the model to an image at multiple locations and scales. High scoring regions of the image are considered detections. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

TensorFlow is an open source library created for Python by the Google Brain team. TensorFlow compiles many different algorithms and models together, enabling the user to implement deep neural networks for use in tasks like image recognition/classification and natural language processing. It has an advantage of speed and accurate when compared to other Neural Networks and hence is preferred to be used in this project for the object detection.

C. Detailed Working Code

Initially the dataset is prepared following which the weights are prepared and loaded. The model is trained with the help of YoloV3 and the model contains 80 classes with labels.

The real-time webcam is connected and the image is read and checked for an empty frame.

If the frame is detected as empty, the loop is continued and the next image frame is captured else if the number of empty frames exceeds 3 then the process is brought to a halt.

The verified image is then converted to RGB format. This is because the TensorFlow models have been trained with image arrays in RGB format, but this codebase loads images using OpenCV, which defaults to BGR. Therefore, it is converted to RGB format before the image is sent for detection. The image is checked for the blur level with the help of a Threshold value (value used is 200) and if the computed level value is below the threshold value then the image is considered as blurred and the frame is not processed and the loop is continued.

To check the Blur level of an image,

- Use Laplace filter to find edges in the input image.
- Compute the variance and the maximum over the pixel values of the filtered image.
- High variance suggests clearly distinguished edges, i.e. a sharp image. Low variance suggests a blurred image.

A Laplacian filter is an edge detector used to compute the second derivatives of an image, measuring the rate at which the first derivatives change. This determines if a change in adjacent pixel values is from an edge or continuous progression.

The processed image is sent to Yolo to predict the image and collect details such as boxes, scores, classes and num of the detected object, where

- Boxes are a list of coordinates of the four corners of the rectangle bounding the detected image.
- Scores is a list of percentage matches of all the detected objects with the dataset objects in the trained model.
- Classes are a list of class numbers to which the detected object is predicted to belong.
- Num is the total number of objects detected in that frame.

The output image is designed based on the image details collected in which the bounding boxes are drawn and the class labels are marked.

Based on the output image, the text string is drafted and outputted in the form of audio along with the output image.

The audio is prepared with the help of GTTS API (Google Translate's Text to Speech) which converts the text string to audio speech and stores it in mp3 format. In this project, the Audio input describes the detected object and the position of the object in the frame ie Left, Right or Centre.

When the user presses 'a', the frame is captured automatically every 9 secs and the audio output is given. The process is halted when 'q' is pressed.

D. *Work Done*

If required

V. Results and Analysis

The results from the built system are as follows. The obstacles and objects within the captured image frame passed to the object detection model, are informed to the visually impaired person with reference to its position relative to the person using audio output. This is repeated every few seconds, which allows the user to have a good idea about his surroundings at all times, especially when encountering an unfamiliar environment which can possibility be challenging for the user. The object detection model that is being used, has been trained on 80 different classes which include many common objects encountered by a person.

YOLOv3 has been chosen for the object detection system because of how fast and accurate it is compared to other systems. In prior detection systems, classifiers were repurposed to perform object detection, applying the model to the image at multiple locations and scales, after which the subsequent regions which scored higher were considered as detections. YOLO uses a different approach by applying a single neural network to the entire image, dividing it into regions and predicting bounding boxed and probabilities for each region. The reason YOLO performs much faster than other techniques is that it makes predictions with a single network evaluation unlike other systems which require hundreds or even thousands for a single image.

Blur Handling:
The system checks the Laplacian variance of every image captured from to determine if the image is blurred. The threshold value that has been set after trial and error with different values is 200. Upon detecting a blurred image, the image is discarded and a new image is captured immediately, which repeats until a clear image is captured.

VI. Conclusion

Our project aims to make the world a better living environment for people who are visually impaired by detecting and classifying a wide range of everyday objects. We have come up with a cheap, efficient and easy solution to a problem that's very relevant. In the future, we are aiming to add face detection to help the user recognise familiar faces. We are also planning on including text conversions to help the user read books, signs etc. There is presently a promising growth in fields related to artificial intelligence so in the future we can implement more computations in a smaller scale.

Induvidual contribution

- Working on data set and model training: Aadil
- Connecting to real-time webcam: Gayathri
- Image Processing using yolo: Harini
- Converting Text output to audio output using GTTS: Prasanthi

Implemented/Base Paper

person	bicycle	car	motorbike	aeroplane	bus	train	truck	boat	traffic light
fire hydrant	stop sign	parking meter	bench	bird	cat	dog	horse	sheep	cow
elephant	bear	zebra	giraffe	backpack	umbrella	handbag	tie	suitcase	frisbee
skis	snowboard	sports ball	kite	baseball bat	baseball glove	skateboard	surfboard	tennis racket	bottle
wine glass	cup	fork	knife	spoon	bowl	banana	apple	sandwich	orange
broccoli	carrot	hot dog	pizza	donut	cake	chair	sofa	Potted plant	bed
Dining table	toilet	Tv monitor	laptop	mouse	remote	keyboard	cell phone	microwave	oven
toaster	sink	refrigerator	book	clock	vase	scissors	teddy bear	hair drier	toothbrush

Fig. 5: The labels that the object detector can detect.

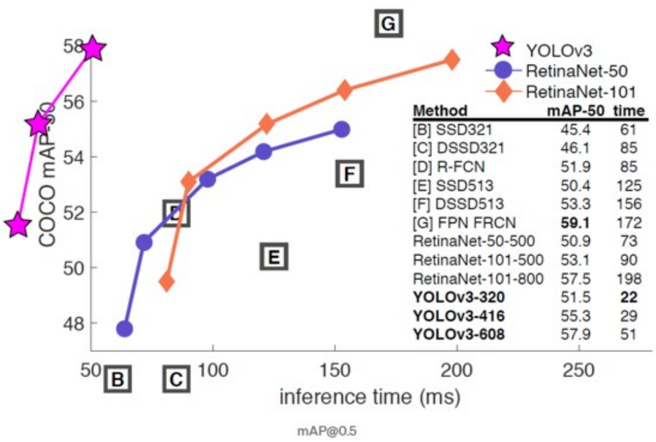


Fig. 6: The labels that the object detector can detect.

Fig. 8: Inuput image vs Processed image
Audio Output: ”There’s a TV monitor on the left. There’s a bed on the right. There’s a chair on the left”

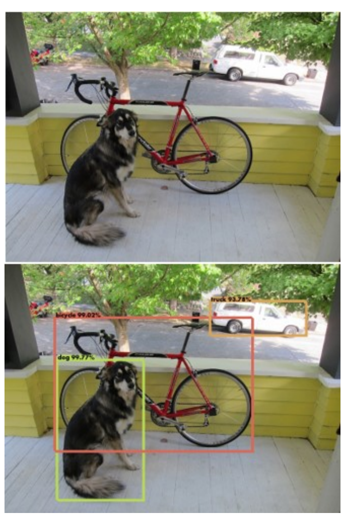


Fig. 7: Inuput image vs Processed image
Audio Output: ”There’s a dog on the left. There’s a bicycle in the center. There’s a truck on the right.”

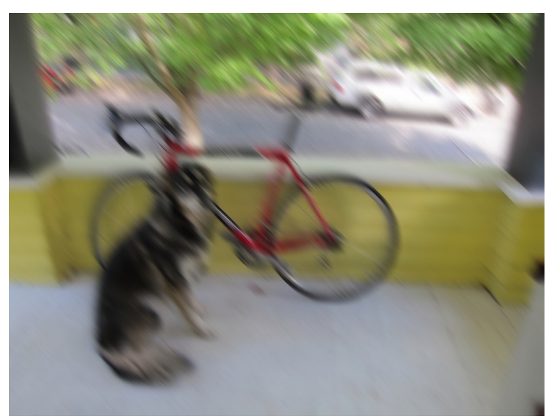


Fig. 9: Output: Blurred image

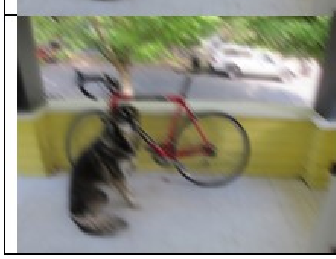
Input Image	Laplacian variance value
	1061.40 (not blurred)
	291.66 (not blurred)
	179.18 (blurred)
	91.40 (blurred)
	47.35 (blurred)

Fig. 10: Output: Blurred image