# Wearable Belt For A Blind Women Person

**Purtee Jethi Kohli[1*], Deepak Kumar[2]**

[1*]Research Scholar, Banasthali Vidyapith, Device supporting Visually Impaired People Using Obstacle Detection, purtijk@gmail.com
[2]Assistant Professor, Banasthali Vidyapith, Device supporting Visually Impaired People Using Obstacle Detection, deepakkumar@banasthali.in

[*]**Corresponding Author:** Purtee Jethi Kohli
[*]Research Scholar, Banasthali Vidyapith, Device supporting Visually Impaired People Using Obstacle Detection, purtijk@gmail.com

**Abstract**
Loss of vision is a very common problem which leads to vision loss permanent or temporary this accident or genetics can have a serve impact on individuals life direction; with such advance in technology our researchers have given some respite to such people. By the Google glass if a human can see technology helps them which is easier for developed couriers like Japan United Kingdom, and USA to name some so in India like developing country we work on indigenous tools and knowledge

**Keywords:** Yolo, Smart glass, Microphone, Obstacles detection

**My work**
The problem of a blind person has been worked by many of addressed this problem of seeing my blind person , things like goggle glass ,India an intectually rich country but sill falls in developing countries basket. So we have thought of a wearable belt on which gadgets are mounted


Fig1: wearable belt with fitted device

**Introduction**
Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results .In this work, we have created software in which user can detect object in three different modes that is image based, video based and real time based. The raspberry pi is the brain of the device, front facing camera the eyes to see and detect the objects As compared to other devices like spectacles helmet can add to strain on head or eyes this wearable belt wins over.We all have someday or other seen blind people struggling due to the objects that comes in their way and that is why we decided to make Object Detection System which will help the blind people by detecting any object that is there in the surrounding of that person. This project will help the blind people in their day-to-day life and also it will make their life easier. In this project we are detecting objects in three different modes that are image based, video based and real time based.

**Real world problem**

Here we are finding a solution with areal word problem With this project we are targeting the blind people as the blind people holds for about 1 billion people with moderate or severe impairment/blindness and therefore we can say that this is a very significant problem that we need to target.The problem is not novel as there have been several researches and inventions in this field but there is not any invention which has high accuracy and is also pocket friendly, that

Some solution team thought

Working on a wearable belt where we have camera a raspberry pi for computation a ear phone or speaker for hearing .For object detection we are asking the user either to give us an image we

Steps performed:

capture the pic, for Object Detection

Image Classification

Image Captioning

Image Reconstruction

After the user selects any of these three, we are using the YOLO (You Only Look Once) algorithm for object detection. Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images. According to this algorithm it will divide the whole image in the grids and for every cell it will store a vector which store information like whether the object is present or not, what is the class of the object and it's coordinates, after this it will create bounding boxes around the object, there are chances where we have several bounding boxes for a single object, so in those cases we use Intersection Over Union Method which states that for every bounding box we will calculate the confidence score and the bounding box that has the highest confidence will be selected and all other will be neglected and finally we will get the output image which will show the bounding boxes around the objects along with the prediction of the type of the object and how much confidently we are prediction that score is also there.

COMPARISON WITH THE EXISTING APPROACHES FOR OBJECT DETECTION

There are several algorithms used for object detection like R-CNN, Fast R-CNN but YOLO algorithm is better than all of them in many ways like :

R-CNN :

The R-CNN detector first generates region proposals (i.e that region which might contain the object) using an algorithm such as Edge Boxes. The proposal regions are cropped out of the image and resized. Then, the CNN classifies the cropped and resized regions. Finally, the region proposal bounding boxes are refined by a support vector machine (SVM) that is trained using CNN features.

Disadvantages:

It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image.

It cannot be implemented real time as it takes around 47 seconds for each test image.

The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.

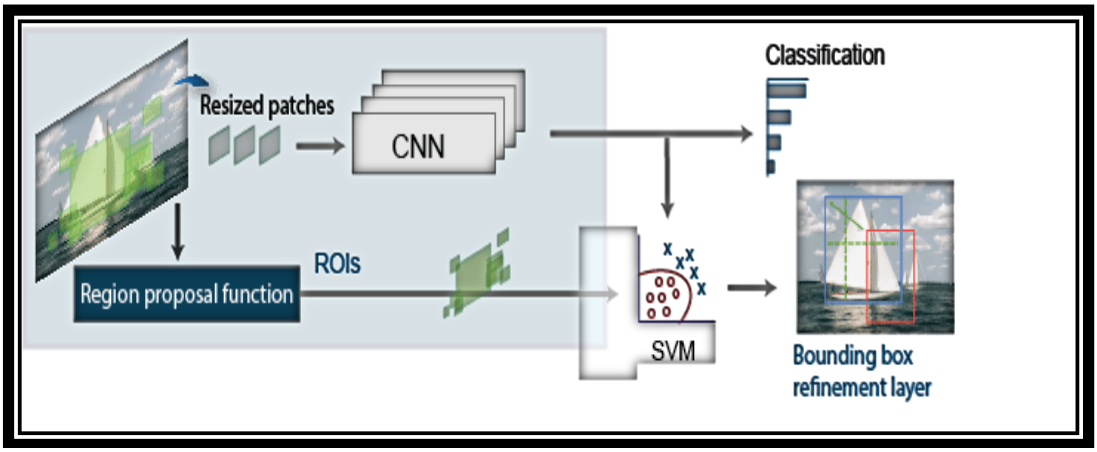



Figure 1: Artistic View of CNN and bounding box in refinement layer

Fast R-CNN:

As in the R-CNN detector, the Fast R-CNN detector also uses an algorithm like Edge Boxes to generate region proposals. Unlike the R-CNN detector, which crops and resizes region proposals, the Fast R-CNN detector processes the entire image. Whereas an R-CNN detector must classify each region, Fast R-CNN pools CNN features corresponding to each region proposal. Fast R-CNN is more efficient than R-CNN, because in the Fast R-CNN detector, the computations for overlapping regions are shared.

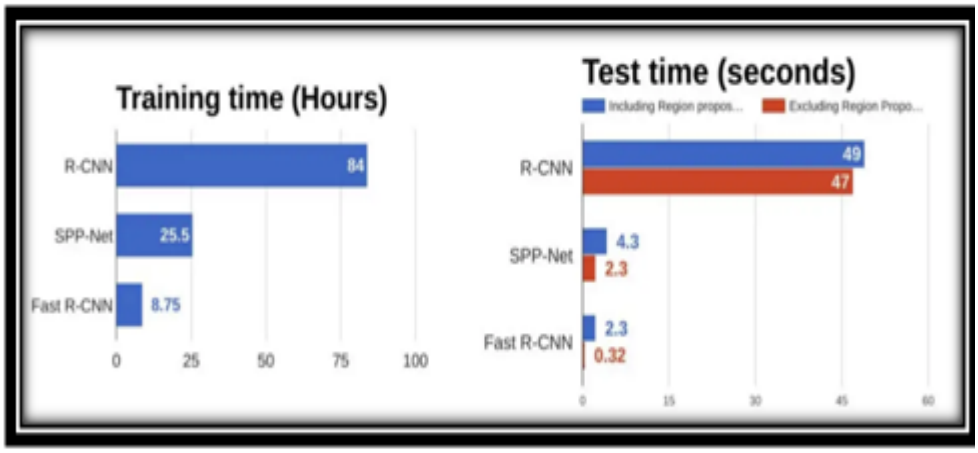



Figure 2: the graph shows comparison in testing and training time

Disadvantages:
From the above graphs, you can infer that Fast R-CNN is significantly faster in training and testing sessions over R-CNN. When you look at the performance of Fast R-CNN during testing time, including region proposals slows down the algorithm significantly when compared to not using region proposals. Therefore, region proposals become bottlenecks in Fast R-CNN algorithm affecting its performance.

**YOLO:**
All of the previous object detection algorithms use regions to localize the object within the image. The network does not look at the complete image. Instead, parts of the image which has high probabilities of containing the object. YOLO or You Only Look Once is an object detection algorithm much different from the region based algorithms seen above. In YOLO a single convolution network predicts the bounding boxes and the class probabilities for these boxes.

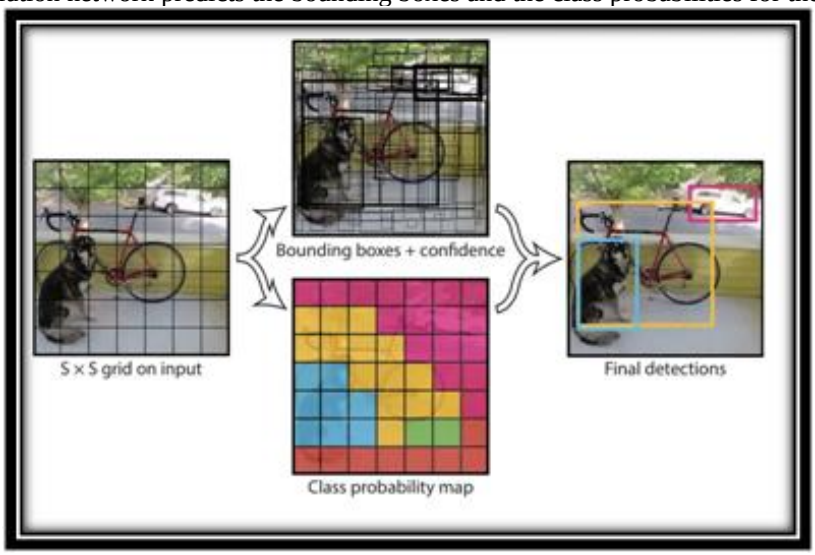



Fig3: box bound of images

**Disadvantages:**
The limitation of the YOLO algorithm is that it struggles with small objects within the image, for example it might have difficulties in detecting a flock of birds. This is due to the spatial constraints of the algorithm.
MY METHOD:
Overall description of the work
Whenever user runs this software they will be prompted with 3 options to choose from, i.e

1. Detect Object From An Image
2. Detect Objects From A Video
3. Detect Object In Real Time

Whenever the user chooses the first option, he will be asked to select an image from the device and after selecting the image, that image will be shown on the software under the selected image label and the user will be asked to click on the Detect Objects Button, after clicking on the Detect Object the YOLO algorithm will run on this image and output will be shown to the user and also this output image will be saved as well.

Whenever the user chooses the second option, he will be asked to select a video from the device and after selecting the video, that video will be shown on the software under the selected video label and the user will be asked to click on the Detect Objects Button, after clicking on the Detect Object the YOLO algorithm will run on the video that too frame by frame and detect objects in those frames and we will get the output where you can see the bounding boxes and name of the object.
Whenever the user chooses the third option, the software opens the web camera of the users laptop and starts detecting objects in the real time. It captures the real time environment frame by frame and detects objects in it and displays them.
list of technologies are required

- **Python :**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

- **OpenCV :**

OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source Apache 2 License.

- **TkInter :**

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and macOS installs of Python. The name Tkinter comes from Tk interface.

- **Python Imaging Library (PIL) :**

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux.

- **Numpy :**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

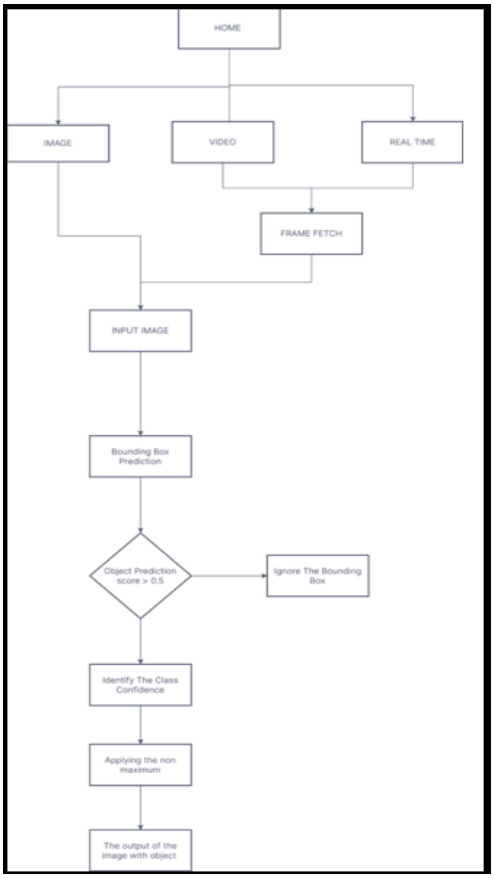
**Steps in method used**

Used have created the interface of the software using the tkinterlibrary :

- We create the object of the tkinter library.
- This library provides us with many widgets like labels, buttons, frames, panels.
- Now first of all we create a frame. Frame is a widget of tkinter library which enables us to arrange / wrap all the widgets inside it in an ordered fashion.
- Now inside the first frame, we will be creating our landing page , in this frame we will be having three buttons , first one for Image Based Detection, second one for Video Based Detection and third one for the Real Time Based Detection.
- Now when the user clicks on the Image Based Detection button, func1() function is called and in this function we destroy the main frame and create another frame only for Image Based Detection and in this frame we will be having the button that will be used for Selecting the Image.
- When the User clicks on this Select Image Button, getImage function is called and in this function we destroy the frame for the Image Detection and create a new frame for this particular page only and in this page first of all a File Dialog will appear to select the file , we are implementing this using the filedialog module of the tkinter library and after the user selects the image, we will be displaying that image on the page and also a button will be there to Detect Objects.
- When the User clicks on the Detect Object button the actual YOLO algorithm code will be called that is written for the image based object detection.
  - In this part we will be reading the image using the OpenCV.
  - Now we will divide the image in the grids on the network by using the layers.
  - Now in each grid cell we have details about whether any object is there or not , x, y, width, height, and values of all the object that our model can detect.
  - Now we will check these details for every grid cell and see which object is present and what is the accuracy score.
  - Now we will find the center of this object and the width and height so that we can make a box around this object in the output.
  - Now we have all the bounding boxes and the confidences related with them and now at last we will perform non maximum suppression to remove the bounding boxes that are colliding with the other bounding boxes for the same object.
  - Now we will display the name of the object along with the bounding box and the accuracy score in the image and save it.
- Now this output image with the object detection is displayed to the user in the software.
- Now if the user clicks on the Video Based Detection, he will be asked to select the video.

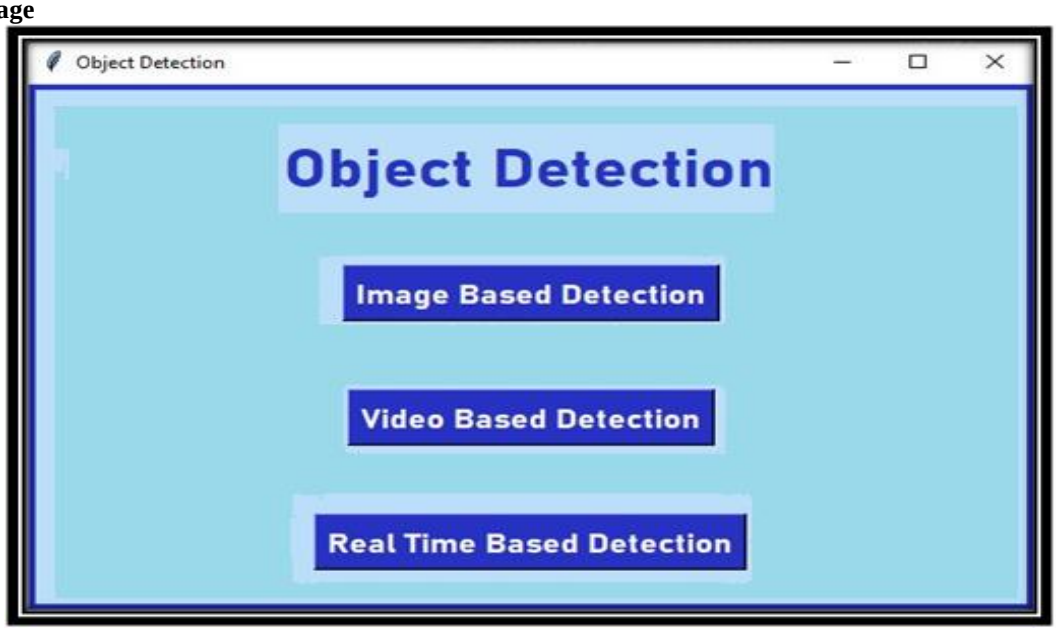
- After selecting the video we will check for the extension if it's MP4 then only we will move forward else we will show a message box displaying the error.
- Now we will be displaying the video selected by the user on the software with the help of tkVideoPlayer library.
- Now if the user clicks on the Detect Object Button, the outVideo function is called and inside this function the code for the video based detection is called.
  - First of all we will load the video using the openCV.
  - Now we will loop around the video frame by frame and each frame is just like an image.
  - Now for every frame we will divide the image in the grids on the network by using the layers.
  - Now in each grid cell we have details about whether any object is there or not , x, y, width, height, and values of all the objects that our model can detect.
  - Now we will check these details for every grid cell and see which object is present and what is the accuracy score.
  - Now we will find the center of this object and the width and height so that we can make a box around this object in the output.
  - Now we have all the bounding boxes and the confidences related with them and now at last we will perform non maximum suppression to remove the bounding boxes that are colliding with the other bounding boxes for the same object.
- Now this output for all the frames will be shown as a video to the user using the openCV and to return to the main menu the user needs to press 'q' from their keyboard.
- Whenever the user clicks on the Real Time Based Detection the function func3() will be called which straight away calls the codes for the real time based detection.
  - First of all we will load the live camera feed using the openCVvideoCapture method by passing the argument for the web camera as 0.
  - Now we will loop around the live feed, frame by frame and each frame is just like an image.
  - Now for every frame we will divide the image in the grids on the network by using the layers.
  - Now in each grid cell we have details about whether any object is there or not , x, y, width, height, and values of all the objects that our model can detect.
  - Now we will check these details for every grid cell and see which object is present and what is the accuracy score.
  - Now we will find the center of this object and the width and height so that we can make a box around this object in the output.
  - Now we have all the bounding boxes and the confidences related with them and now at last we will perform non maximum suppression to remove the bounding boxes that are colliding with the other bounding boxes for the same object.

**Flow chart, modeling and implementation details**

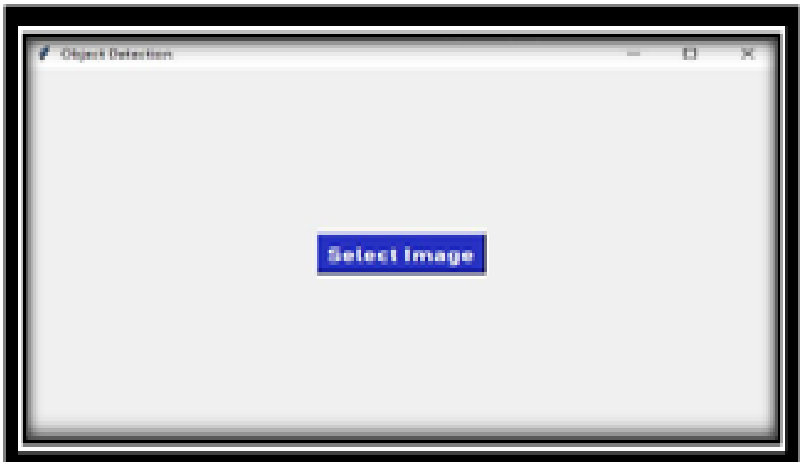


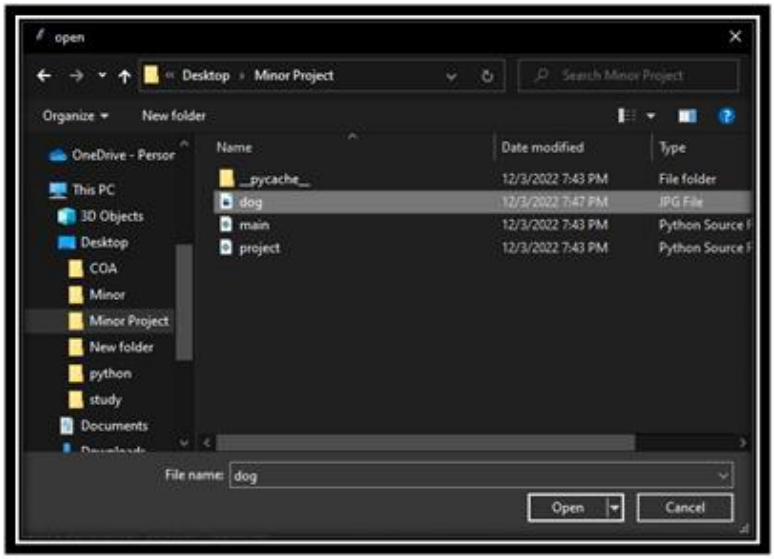
**DIAGRAMS:**

- **Home Page**



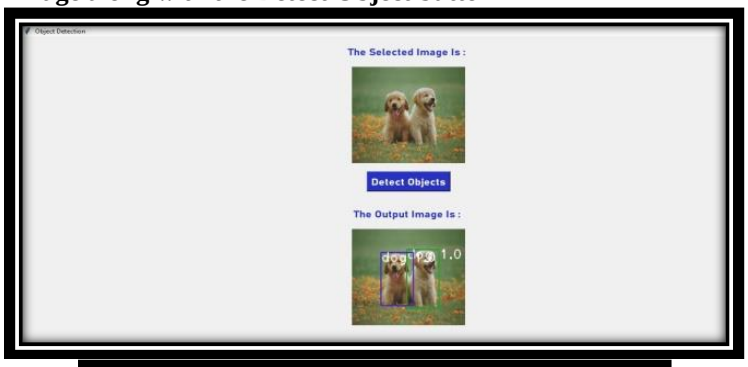

- **Image Based Detection**
  - **Selecting the Image from the device**

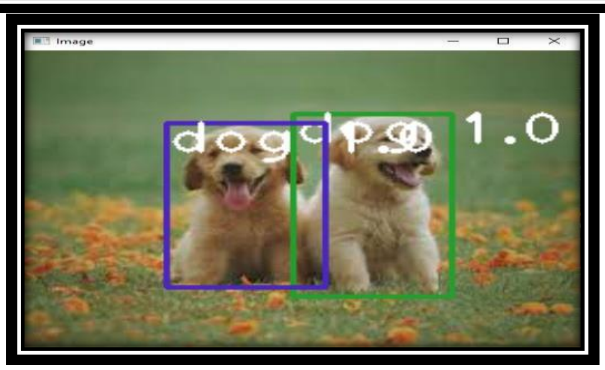


**Choosing the file using the file dialog box**

- **Displaying the selected image along with the Detect Object button**

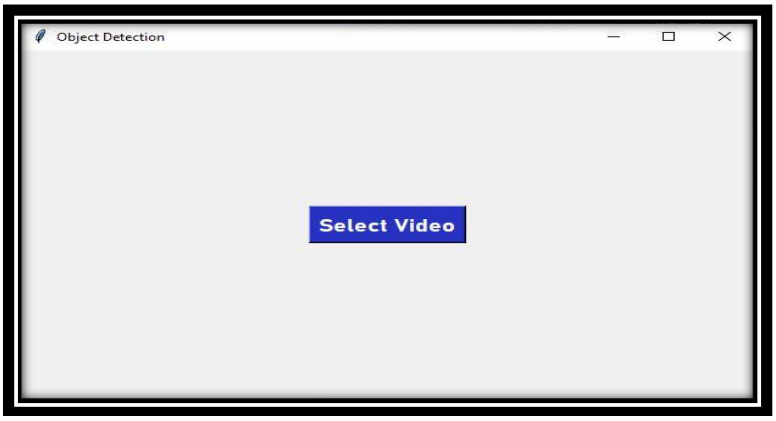



- **After clicking the Detect Objects Button**

- **Video Based Detection**

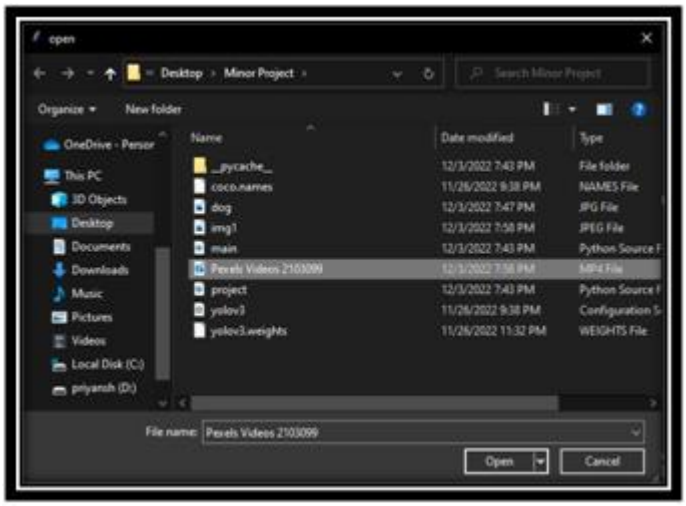
- **Selecting the video from the device**




- **Choosing the file from the device**
- **Displaying the selected video along with the Detect Objects Button**
- **After clicking on the Detect Object Button**

- **Real Time Based Detection at own roam**
  - **After clicking on Real Time Based Detection it straight away opens the web camera and starts detecting the object.**


Images captured at own room 1,2,3,4,5
Patent figA*




swimming n drowning situation shouting help help help out of order hand pattern , no regular hands (still sank drown)

**TESTING**

In the testing phase of the project we used several images to test the object detection system for the image based detection system, for the video based detection system we have used some videos of the traffic to check whether our software is working fine or not in detecting the objects that are of various types in a single frame and for the real time object detection we have used several objects like fork, spoon, cell phone, notebooks, bed, chair, etc.

## LIMITATIONS

- **Portability:** As of now, we are detecting the objects in real time using the web camera of the laptop.
- **Output Format**: As of now, we are displaying the output on the screen of the laptop, but it's of no use for the blind people.
- **Accuracy:** As of now, our software is accurate to great extent but for the real implementation we need to increase its accuracy.

## CONCLUSION AND FUTURE WORK

### CONCLUSION

We have created the software for detecting the objects that are present in your environment and this software is perfectly working and that too with high amount of accuracy. Anyone can detect the object from any kind of media i.e image, video and live camera feed. This software can be used in several fields like for blind people, in automatic vehicles and many more fields.

### FUTURE WORK

There is still a great scope for improvements and updates for our projects which shall be implemented in due course of time.

The points that we have written in the limitations section of the project are going to be our first target to work at.

- For the Portability Issue, we need to make hardware using the Raspberry Pi and then put that hardware in some kind of headgear like hat, caps, etc which the blind people can wear on a day to day basis.
- For the Output Format Issue, we need to integrate audio in the hardware so that we can inform the blind that there is some kind of object present in their surroundings.

### References

1. Baranski, P., Polanczyk, M., Strumillo, P.: A remote guidance system for the blind. In: The 12th IEEE International Conference on e-Health Networking, Applications and Services, pp. 386–390 (2010) Google Scholar
2. Dhod, R., Singh, G., Singh, G., Kaur, M.: Low cost GPS and GSM based navigational aid for visually impaired people. Wireless Pers. Commun. 92(4), 1575–1589 (2017) CrossRef Google Scholar
3. Al-Fahoum, A.S., Al-Hmoud, H.B., Al-Fraihat, A.A.: A smart infrared microcontroller-based blind guidance system. Active and Passive Electronic Components 2013, 1–7 (2013). https://doi.org/10.1155/2013/726480
4. Saquib, Z., Murari, V., Bhargav, S.N.: Blindar: an invisible eye for the blind people making life easy for the blind with internet of things (IoT). In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), pp. 71–75 (2017) Google Scholar
5. Chandana, K., Hemantha, G.: Navigation for the blind using GPS along with portable camera based real time monitoring. SSRG Int. J. Electron. Commun. Eng 1, 46–50 (2014) Google Scholar
6. Tayyaba, S., Ashraf, M.W., Alquthami, T., Ahmad, Z., Manzoor, S.: Fuzzy-based approach using IoT devices for smart home to assist blind people for navigation. Sensors 20(13), 3674 (2020). https://doi.org/10.3390/s20133674 CrossRef Google Scholar
7. Roy, S., Gharge, S., Jasoriya, R., Agrawal, P., Jounjalkar, I.: ORCap: Object recognition cap (a navigation system for the blind). In: 2020 IEEE International Conference for Innovation in Technology (INOCON). IEEE, November 2020. https://doi.org/10.1109/inocon50539.2020.9298310
8. Cardillo, E., Li, C., Caddemi, A.: Millimeter-wave radar cane: a blind people aid with moving human recognition capabilities. In: IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology pp. 1–8 (2021). https://doi.org/10.1109/jerm.2021.3117129
9. Sheth, R., Rajandekar, S., Laddha, S., Chaudhari, R.: Smart white cane-an elegant and economic walking aid. Am. J. Eng. Res. 3(10), 84–89 (2014)

- Appendix website resources
- https://pjreddie.com/darknet/yolo/
- https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-
- https://github.com/pjreddie/darknet/blob/master/cfg/yolov3.cfg
- https://www.mygreatlearning.com/blog/yolo-object-detection-using-opencv/