IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 3, March 2022

DOI: 10.17148/IJARCCE.2022.11350

VISION – A Tool for Visually Impaired

Mr. Shailendra Singh, Kartikeya Gaur, Muskan Rajput, Vedika Verma

Department of Computer Science & Engineering, Inderprastha Engineering College, Uttar Pradesh, India

Abstract— The visually impaired people in the world face a lot of problems in day-to-day life. They need either a human or a stick to guide them through their different daily life tasks. They often get hit by objects because they're not able to see them coming towards them. They're not able to recognize people and objects without touching or hearing them. Entertainment is also a luxury for them since they can hardly operate any device. This research paper depicts a project that can solve all the above problems and perhaps more in long run.

I. INTRODUCTION

Our project utilizes the YOLO model for the real time object detection with the use of embedded camera and then by using the Pyttsx library and speech recognition python package, the detected objects are notified to the user through the voice message. When implemented using a device having a camera and microphone/ speaker, this is going to guide and indicate the visually challenged people about their surroundings in a very efficient manner. Not only the objects but our project also recognises the faces of the known persons which are appearing in front of the visually challenged person. Our project is also capable of performing face recognition and object detection tasks by using a portable or mobile camera device connection to our model.Therefore, this prototype implementation will be used by the engineers for the welfare of the visually impaired people or those who are unable to see properly due to various vision disorders.

II. VISION MODEL

This section of the paper consists of the different modules of our Vision Model and its working.

A. Object Detection Module

It is the first module of Virtual Drishti to detect the object appearing in front of the impaired person which is based on YOLO Algorithm.

1. YOLO Algorithm

Until now, we saw some very famous and well-performing architecture for Object detection. All these algorithms solved some problems mentioned at the beginning of this article but fail on the most important one i.e. Speed for real-time object detection.

YOLO algorithm gives much better performance on all the parameters along with a high fps for real-time usage. YOLO algorithm is an algorithm based on regression, instead of selecting the interesting part of an image; it predicts classes as well as the bounding boxes for the whole image in one run of the Algorithm.

2. Working of YOLO Algorithm

First, an image is taken and YOLO (You Only Look Once) algorithm is applied. In this example, the image is divided into grids of matrices. We can divide the image into any number of grids, depending on the complexity of the image. Once the image is divided, each one of the grids undergoes classification and localization of the object. The confidence score of each one of the grids is found. If there is no proper object found in the grid, then the confidence score and bounding box value of the grid will be zero or if there found an object in the grid then the objectness score will be 1 and the bounding box value will be its corresponding bounding values of the found object. The bounding box prediction is explained below. Also, Anchor boxes are used to increase object detection accuracy, which is also explained below in detail.

B. Face Recognition Module

Face recognition consists of 3 phases: - Data collection, training and recognition.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 3, March 2022

DOI: 10.17148/IJARCCE.2022.11350

We have implemented this by keeping in mind that a blind person needs to have this so that he can identify who is near around him. This module will help to track who is in front of him and it can track multiple people too.

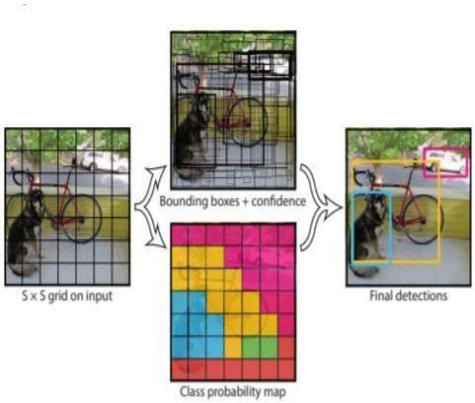
Steps included:-

1. Testing our Camera:-

Once we have OpenCV installed in our RPi let's test to confirm that our camera is working properly. Assuming that the PiCam has already installed. The code will capture the video stream that will be generated by our PiCam, displaying both, in BGR colour and Grey model. To finish the program, we must press the key [ESC] on our keyboard. Click our mouse on the video window, before pressing [ESC].

2. Face Detection:-

The most basic task on Face Recognition is indeed, Face Detecting. Before anything, we just need to capture an image in order to recognize the face. The most used way to detect a face is using the Haar feature-based cascade classifier. Object Detection using Haar cascade classifiers is a great object detection method given by Paul Viola, Realtime Object Detection using a Boosted Cascade of an ordinary feature. It is a machine learning technique where a cascade function is trained by gathering a lot of positive and negative images. It is then used to detect objects and faces in other pictures.



Here we have done a job over face detection. Initially, the LBPH algorithm needs a lot of positive images or images of faces and negative images or images without faces to train the classifier. Then we have to gather the features from it. The beneficial part is that OpenCV comes with a trainer as well as a detector in the inbuilt. If we want to train our own classifier for any object like cars, planes, etc. we can use OpenCV to create it.

We have to call the classifier function, passing it the most important parameters, as scale factor, the number of neighbours and minimum size of the detected face, and even the dimension (co-ordinates). Grey is the input for generating the grayscale image. ScaleFactor is the parameter specifying how much the image size is reduced at each image scale, minNeighbors is a parameter calculating no of neighbours every candidate rectangle will have. A greater number gives fewer false positives, minSize is the minimum requirement for the rectangle size for the consideration of a face.

If faces are found, it returns the positions of detected faces as a rectangle with the corners (X,Y) and having "W" as its Width and "H" as its Height ==> (X,,Y,W,H).

3. Data Gathering:-

It is done by the first phase of our module. 20 images are being taken and are converted into greyscale for training purposes.

4. Recognizer:-

Now, we reached the final phase of our module. Here, we have captured a fresh face on our camera and if the person had his face captured and trained before, our recognizer will make a prediction returning its id and an index no that is showcased during the presentation, index number shows how confident the recognizer is with this match.

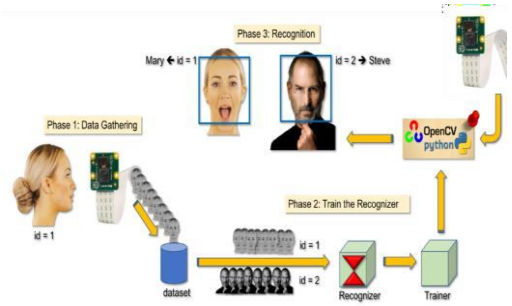
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 3, March 2022

DOI: 10.17148/IJARCCE.2022.11350

The recognizer predict (), will take as a parameter a captured portion of the face to be analyzed and will return its probable name, indicating its id and how much confidence the recognizer is in relation with this prediction. And at last, if the recognizer is able to predict the face, we put a text over the image with the probable name and how much is the probability in % that the match is correctly found ("probability" = 100 — confidence index number). If not found, an "unknown" label is put on the face in real-time.



III. RESULTS AND CONCLUSION

To find the obstacle in front of a person we used the YOLO model which is far descent and is pre-trained. YOLO model also saved our time for collection of various data (objects around us) and made this project work in that short period of time. Its accuracy is around 70-80% which is good.

We are planning on developing an embedded system to make this project more feasible and work perfectly. A blind person can always use this embedded spectacles for his/her day to day chores. To find faces in a real time scenario we used OpenCV and it also did a pretty amazing job. Its accuracy is around 80 % and it can detect faces in real time.
With the combination of these 2 we are able to build a prototype that can be used by a blind person to get a virtual vision. With our efforts, we hope we can do something for the welfare of the visually impaired society.

IV. FUTURE WORK

In the future we are planning to implement our work in a more realistic way. We plan to make a fully working spectacle that has camera, speaker and a microphone inbuilt so that the camera capture the image of obstacle and with the help of speaker the blind person can get information about who is in front of him.

A Bluetooth device or some other connectivity tool will be used to send information to a phone. The phone will process the information with the help of our current model and then it will again send the outcome to the spectacle to narrate who is in front of the person.

V. ACKNOWLEDGMENT

It gives us a great sense of pleasure to present the research paper on Vision during our final year (B.Tech). We would like to thank **Mr. Shailendra Singh**, who is our mentor, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness, and perseverance are continuing sources of inspiration for us. It is only his cognizant efforts that our endeavor has seen the light of the day. We also take the opportunity to acknowledge the contribution of his full support and assistance during the development of the project.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 3, March 2022

DOI: 10.17148/IJARCCE.2022.11350

REFERENCES

1. M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression", Computer Vision—ECCV 2008, pp. 2-15, 2008.
2. J. Yan, Z. Lei, L. Wen and S. Z. Li, "The fastest deformable part model for object detection", Computer Vision and Pattern Recognition (CVPR) 2014 IEEE Conference on, pp. 2497-2504, 2014.
3. P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object detection with discriminatively trained part based models", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, 2010..
4. S. Ren, K. He, R. Girshick and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, 2015, [online] Available: .
5. D. Erhan, C. Szegedy, A. Toshev and D. Anguelov, "Scalable object detection using deep neural networks", Computer Vision and Pattern Recognition (CVPR) 2014 IEEE Conference on, pp. 2155- 2162, 2014.
6. Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2014.
7. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I-I, Dec. 2001..
8. B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, et al., "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a", The IEEEConference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015.
9. Z. Huang, S. Shan, R. Wang, H. Zhang, S. Lao, A. Kuerban, et al., "A benchmark and comparative study of video-based face recognition on cox face database", IEEE Transactions on Image Processing, vol. 24, pp. 5967-5981, Dec. 2015.
10. K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.