

Electronic Orientation Aid for Visually Impaired using Graphics Processing Unit (GPU)

Akshata Parab^{1,1}, Rashmi Nagare², Omkar Kolambekar³, and Parag Patil⁴

¹Department of Electronics Engineering, Ramrao Adik Institute of Technology, Nerul, India

²Department of Electronics Engineering, Ramrao Adik Institute of Technology, Nerul, India

³ Department of Electronics Engineering, Ramrao Adik Institute of Technology, Nerul, India

⁴Department of Electronics Engineering, Ramrao Adik Institute of Technology, Nerul, India

Abstract. Vision is one of the very essential human senses and it plays a major role in human perception about surrounding environment. But for people with visual impairment their definition of vision is different. Visually impaired people are often unaware of dangers in front of them, even in familiar environment. This study proposes a real time guiding system for visually impaired people for solving their navigation problem and to travel without any difficulty. This system will help the visually impaired people by detecting the objects and giving necessary information about that object. This information may include what the object is, its location, its precision, distance from the visually impaired etc. All these information will be conveyed to the person through audio commands so that they can navigate freely anywhere anytime with no or minimal assistance. Object detection is done using You Only Look Once (YOLO) algorithm. As the process of capturing the video/images and sending it to the main module has to be carried at greater speed, Graphics Processing Unit (GPU) is used. This will help in enhancing the overall speed of the system and will help the visually Impaired to get the maximum necessary instructions as quickly as possible. The process starts from capturing the real time video, sending it for analysis and processing and get the calculated results. The results obtained from analysis are conveyed to user by means of hearing aid. As a result by this system the blind or the visually impaired people can visualize the surrounding environment and travel freely from source to destination on their own.

1 Introduction

Vision is the essential part of human biology, since 83% of human knowledge is perceived through eyes [3]. If that is the case, then in today's busy world it is very difficult for the visually impaired or blind people to navigate and travel alone. All regular tasks such as walking, reading, travelling, socializing are become inconvenient for visually impaired which makes their life difficult from the normal people. The traditional and oldest mobility aids for persons with visual impairments is the walking cane (also called white cane or stick) and guide dogs. However, the main pitfalls of these aids are expertise and preparation [5]. With the rapid advances in modern technology, it has given smart navigation capabilities both in the hardware and the software field. In recent years, extensive research work has been conducted to create aids for visually impaired people to interpret, guide and navigate in the surrounding environment. In this study a guidance system called Electronics Orientation Aid (EOA) is built to overcome their navigational challenges. This system allows obstacle detection and intimation of obstacle location which helps the visually impaired person to reach to the desired destination with greater awareness of the environment. The real time images of objects in the surrounding are used, and then from the captured images the objects are detected. Once the object has been identified, the user will be given

some specifics such as what the object is, its location, whether or not it is a hindrance. Audio instructions are also given for the above mentioned information that will be heard by the visually impaired people, ensuring that they could visualize the objects in front and its location. As the capturing of the images and the processing has to be quick and in real time, there is a need to use the Graphics Processing Unit (GPU). Use of GPU speeds up the process of object detection and recognition. Thus using EOA, the user can travel on their own from source to destination with more ease.

This paper is organized as follows: Section I introduces the cause of this work and the brief study of literature is done in Section II is the motivation which led to the idea of this work. Section III defines the hardware organization of the entire project which will focus on the communication between different sub-systems included in our work. Similarly Section IV defines the design implementation of the system. The YOLO V3 algorithm used in EOA is explained in Section V. Section VI shows the detailed working mechanism of our study while the results are showcased in Section VII. Finally Section VIII concludes the work mentioning the future work in Section IX.

¹ Akshata Parab: parabakshata27sp@gmail.com

2 Literature Survey

The visually impaired are the people who can lose vision permanently or temporarily. These people cannot see, read or write like normal people. According to the World Health Organization (WHO), 285 million people worldwide experience visual impairment, 39 million are blind and 246 have low sense of vision. Hence these people require an assistance by which they can visualize the outside world and live a normal life. Various research papers like “Wearable object detection system for the blind” by Alessandro Dionisi, Emilio Sardini, Mauro Serpelloni [11] and “A mobility aid for the blind with discrete distance indicator and hanging object detection” [2] explained the need to design a system which will provide an assistance to the visually impaired people to navigate freely from the source to destination with no or minimal help. Blind people are always dependent on others to cross road or to reach a particular address. Therefore, we build a system called EOA that will help them to carry out their activities like normal people to some extent. Various algorithms and measurements were studied from different papers like “Object Detection Based on YOLO Network” [2], “Real Time Object Detection, Tracking, and Distance and Motion Estimation based on Deep Learning: Application to Smart Mobility” [19] etc. that encouraged us to select the correct algorithm to carry out our work. This led us to complete our project with better understanding and guided us throughout the journey.

3 Hardware Organisation

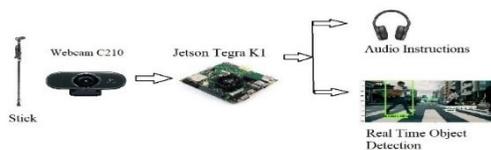


Figure 1. Flowchart depicting the hardware used in the work.

A. NVIDIA Jetson Tegra K1 (TK1): Jetson Tegra K1 is the first embedded graphics processor from NVIDIA. This is similar to modern GPU desktop with advanced features, but still uses less power (less than 6 W). The NVIDIA Jetson TK1 is a full featured package, which includes an integrated TK1 platform. This board gives us the power to utilize 192 CUDA cores for different technical applications such as robotics, computer vision, healthcare, automation, medicine and so on [7]. This board is supported by developer-friendly Ubuntu Linux software environment. The reason for choosing Jetson TK1 for our work includes its low power consumption, compact size and weight, parallel processing capability, etc. B. Logitech Web Camera:

The webcam used in our work is Logitech C210. This camera is used to capture the real time video or image and send the captured images to jetson board for further analysis. The Logitech Webcam C210 features the ability to record 640x480 resolution in a 16:9 wide screen ratio with specification of 1.3 MP. This device offers impressive 720p resolution.

C. Bluetooth Headphones: After detection of the object, recognition of object and its location the final important task in this work is to convey this information to user in form of audio. For this we need bluetooth earphones or headphones that the visually impaired person will wear. Any bluetooth headphones pairable with Jetson TK1 can be used. In our project we have used bluetooth earphones that will be wirelessly connected to Jetson TK1. The purpose of using bluetooth earphones and not the wired is to reduce the hassle and make the design easy to carry.

4 Design Implementation

Computer vision is a field in which images can be captured, stored, analysed and understood. Here, the output is in the form of identification or mathematical measurements. The motive behind doing this is to make the necessary decisions based on the highly dynamic real time data and its analysis.

Computer vision is a field where understanding of human vision can be duplicated. Often referred to as image analysis [16]. The idea is to use computer vision for object detection, that could probably help the blind to visualize the surrounding environment better, using Image-to-Text and Text-to-Voice, without any complex hardware [18].

A. Object detection: Object detection is a computer vision operation that detects the images/videos and creates bounding boxes around them to locate that object. The annotated text on the bounding box can be translated into voice response and the fundamental positions of the objects can be provided from the perspective of the person’s location. By recognizing the front objects and making them aware of the danger, the object sensing module in the system will help the blind to also provide a safe route to reach the destination [3].

B. Various algorithms for Object Detection: There are three main Object detection Algorithms and they are as follows:

Region-CNN (R-CNN): To solve the problem of selecting large number of regions, Ross Girshick et al. proposed a method is used to extract 2000 regions from an image called as region proposals. So, now instead of classifying huge no. of regions RCNN works only with 2000 regions. These 2000 region proposals are

generated using selective search algorithm [1]. 2000 region of proposals are bounded into square and fed into a convolutional neural network that produces a 4096 dimensional feature vector output.

Single Shot Detector (SSD): Object detection and classification tasks are performed in a single forward network pass. Multi Box bounding box regression technique is used in this. This method strikes a great balance between speed and precision. It runs a convolutionary network on an input image only once and computes a feature map. It predicts bounding boxes after multiple convolutionary layers in order to hold the scale. [2].

You Only Look Once (YOLO): Object detection is a task of computer vision involving the localisation of one or more objects within an image and the classification of each object in the image. It is a difficult computer vision function that includes both efficient position of objects to locate and draw a bounding box around each object in an image, and classification of objects to determine the appropriate type of object that has been identified. The solution involves a single deep neural network (originally a GoogleNet version, later revised and renamed DarkNet based on VGG).

4.1.1 You Only Look Once (YOLO):

YOLO is one of the quickest algorithms to detect objects. YOLO trains entire images and increases detection efficiency directly. All the earlier algorithms used to classify the object in the image using regions. Parts of the image which contain the object are highly likely. In YOLO the bounding boxes and class probabilities for these cases are determined by one single convolutionary network. YOLO's very quick. We simply run our CNN on a picture in order to forecast detections [3].

You Only Look Once (YOLO)[10], the computer vision device capable of detecting a range of objects in one image, with an accuracy similiar to RetinaNet, but with higher inferiority than with other existing systems, such as SS[11], R-FCN[12] and FPN FRCN[13]. Its speed makes it one of the most suitable real-time object detection systems used in systems such as robotic operations [4]. The algorithm processes the entire image, not only the region of its inferences, which reflects the overall context of the image, thus making it less likely to detect background content in an object, which enables its inferences to contemplate the overall significance of the image. YOLO has a unique, jointly trained pipeline. In contrast to other systems with different components, such as Faster RCNN [12], which must be training separately. It is widely used in real-time applications due to its tremendous speed. YOLO was one of the most popular algorithms for object detection. Some important things to know are:

YOLO takes an image and divides it into SxS grid, where S is a natural number.

Each pixel in the image causes a finite number (5 in our case) of boundary box estimates. The pixel is responsible for predicting when the centre of the object is found. Of all the boxes found, it is responsible for identifying only one object and the other identifiers are rejected.

It predicts C conditional class probabilities (one per class for the likeliness of the object class).

Total detections to be done per image= $SXS ((B * 5) + C)$
 Where SXS = total number of images yolo divides the Input
 B is the Number of Bounding boxes detected in the image.
 For each bounding box, 5 elements are detected: Detected Objects, Centre coordinates(x, y), Height and Width, Confidence percentage.

C =Conditional probability for Number of Classes.

For example, if the image is divided in a 2x2 grid, and 10 boxes are predicted with 3 classes (Bat, Ball, Gloves), we will have $2 * 2(10 * 5 + 3)$ predictions= 212 predictions.

4.2.1 Different Versions of YOLO:

YOLO requires a Neural Network framework for training and in this case we have used DarkNet.

YOLOv1: Version 1 has total 26 layers, with 24 Convolution Layers followed by two Fully Connected layers. The biggest problem with YOLOv1 is its inability to find the smallest things. Latter, 2 more versions for YOLO released.

YOLOv2: After each Conv layer there are batch normalization layers. Compared to YOLO v1 it has 30 layers. Introducing the anchor boxes. Anchor boxes are pre-defined boxes that a user gives the network an idea of how many things should be received on Darknet. A set of training materials should be used to calculate it. There is no fully connected layer. Of training photos from 320-608, fixed sizes were taken. Many labels can be given to same things, but still the problem with many symbols.YOLOv2 was still not efficient for small objects.

YOLOv3: In YOLOv3 there are 106 layers neural networks. The most important feature of YOLOv3 is that it makes detection at three different scales. It is fully conventional such that the output is obtained by applying 1x1 kernel on feature map.

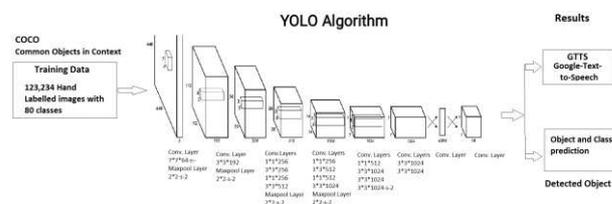


Figure 2. Flowchart depicting the YOLO Algorithm

4.1 YOLO Algorithm for Object Detection

- YOLO algorithm which passes the image through the complexities of a neural network architecture called Darknet [2].
- Darknet is a neural C and CUDA based open source network architecture that is simple and easy to use and supports computation for CPU and GPU systems.
- In this proposed system YOLOv3 model which is more enhanced and complex is used.
- A pre-trained model Common Objects In Context (COCO) dataset is trained on yolov3 [1].
- Also, the python cv2 package has a method to setup Darknet from yolov3.cfg file.
- OpenCV is the computer vision library/ framework that we will be using to support our YOLOv3 algorithm. OpenCV has inbuilt support for Darknet Architecture [14].
- Darknet Architecture is a pre-trained 80 class platform. Our goal is to identify objects using Python language with Darknet (YOLOv3) in OpenCV.
- The input from the camera is used to feed images to this professional model at 3 frames/run.
- Then there is the calculation of various things such as identification of the detected object from the 80 class of the trained model. The measurement of the objects position is determined by its co-ordinates. This helps us to find where the object lies exactly. For example, “top/mid/bottom” or “left/centre/right”.

Further the text is send to the Google-Text-To-Speech API using gTTs package

5 Working Mechanism

The working starts with the web camera. This camera captures the real time images. In other words, live video is read and is converted to individual frames. These frames are further analysed by Jetson TK1 board. From each frame the objects are detected. For the task of object detection the YOLO algorithm is used. If, objects are present in particular frame, the objects are captured and are passed through the layers of neural networks on which the YOLO algorithm works. The time for which the camera should record the video can be varied, and accordingly the frames will be captured. In our work we have kept 3 frames per shot, which collectively takes 28-35 seconds approximately. The 3 frames are read one after the other in continuation. The duration of object detection and recognition for each frame depends on how many objects are detected. If less number of objects are in front, then analysis period of particular frame will be short, if many objects are present then accordingly the period will increase. The captured images are transferred from one layer to another. This filters region of interest from the frame, where the object is present. Once the object is obtained, is compared with the COCO data set which is trained to identify 80 objects. This data set helps to identify the object which the user would be unaware of. Along with

the identification, the position of that object is also calculated with respect to the centre of the frame i.e. whether the object is at top of the frame or mid or bottom. Similarly, if it lies at the right of frame or centre or left [19]. The accuracy of the detection is also mentioned. This shows how correctly the image is identified. This all calculations are done by mathematical formulas and equations. However, the distance of the object from the camera is calculated manually. The next and most important part of the system is the hearing mechanism. Without this the entire system would not be helpful for user. So now, all the detected objects along with their position, precision, distance is narrated to the user via bluetooth headphones. As a result, user will be more aware of the obstructions in front of them.

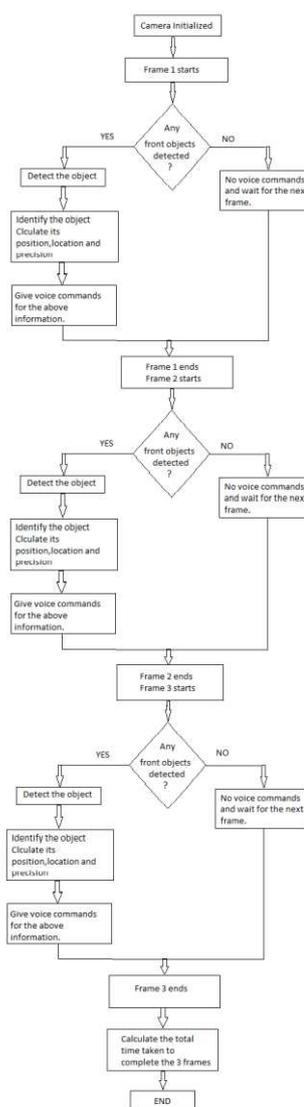


Figure 3. Working Mechanism Flowchart.

After the 3 frames are detected, identified, audio commands are given the total time is calculated. This total time

represents the time from which Frame 1 has started to the end of the Frame 3 including giving of the audio commands. This time factor is of utmost important because this actually proves if this system will be useful to the blind person or not. Considering if this time factor came out to be very long then it would be of no use to the user as by that time another new object would come on the path of the blind person and accidents may happen. So any delay in any process can put the life of the visually impaired in danger. Hence, the time taken should be as small as possible. For this, we are using GPU. With GPU the process of detection becomes fast and subsequently all the process will be rapid giving quick responses and fast audio commands to the visually impaired or the blind person.

6 Results

After taking a number of trials we could capture and detect various objects along with their position and precision. The results are shown below:

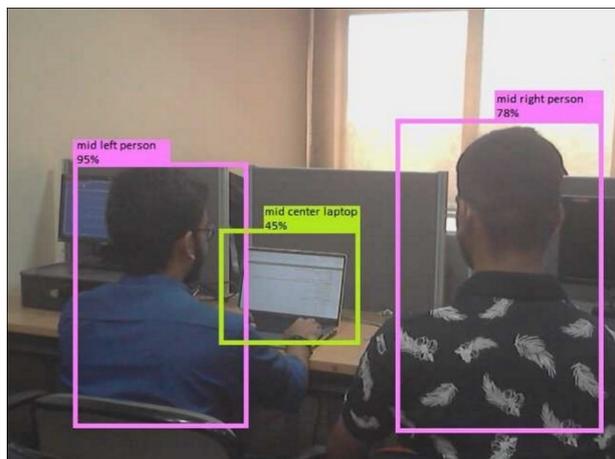


Figure 4. The output of YOLO algorithm for Frame-1.

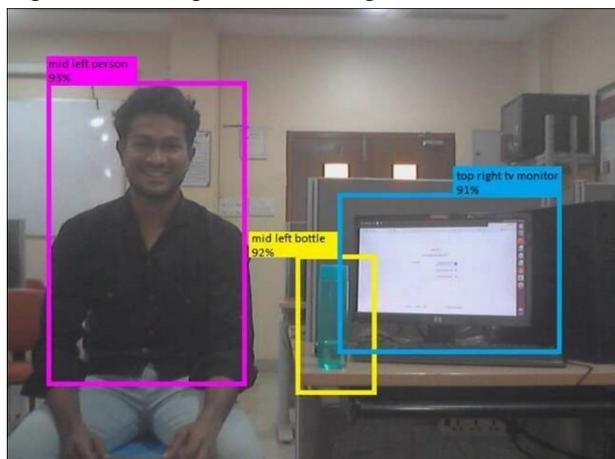


Figure 5. The output of YOLO algorithm for Frame-2.

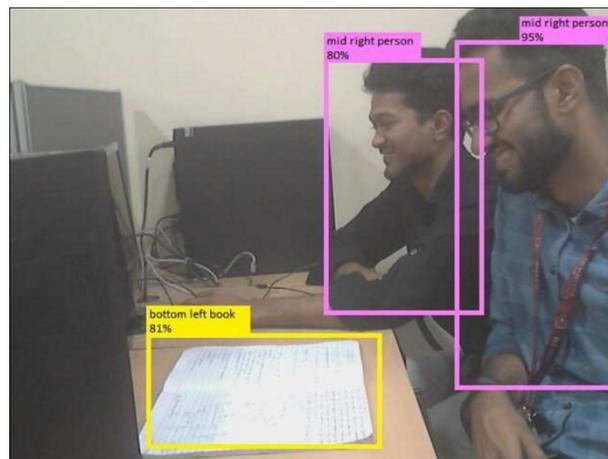


Figure 6. The output of YOLO algorithm for Frame-2.

The table below shows the analysis of the obtained results that gives the clear picture of our working.

Sr. No.	Detected Object	Position	Accuracy	Range
1	Person	Mid Left	90%	3.5 feet
2	Person	Mid Right	78%	3 feet
3	Laptop	Mid Centre	45%	4.1 feet

Table I. The summary of output generated for frame 1.

Sr. No.	Detected Object	Position	Accuracy	Range
1	Person	Mid Left	92%	5.4 Feet
2	Bottle	Mid Left	79%	7.1 Feet
3	TV monitor	Top Right	88%	7.2 Feet

Table II. The Summary of output generated for frame 2.

Sr. No.	Detected Object	Position	Accuracy	Range
1	Person	Mid Right	95%	2.2 Feet
2	Book	Bottom Left	87%	2.4 Feet
3	Person	Mid Right	80%	2.6 Feet

Table III. The summary of output generated for frame 3.

As soon as the camera is initialized, it starts recording. From Figure 4 it is observed that the algorithm has detected 3 objects which are as follows: 1 person, 1 bottle and 1 TV monitor. All 3 are located at different positions. As shown in figure 4, 1 person is located at the bottom of the horizontal axis and at centre of the vertical axis. Secondly, the bottle is at the middle of the horizontal axis and to the left of the vertical axis. Lastly the TV monitor lies at the top of the horizontal axis and to the left of the vertical axis. All these positions are determined considering the distance of the object with respect to the centre co-ordinates. For Frame-2 all these objects along with their positions are detected which is tabulated in the Table II. The output also showcases the precision. This indicates the accuracy with which the algorithm has detected the objects from the image. For the Frame-2, we get the precision for the object person as 92% which tells that the detected object is 96% truly a person. Similarly, object bottle is 79% correctly detected and the precision of the object TV monitor as per detection is 88%. The next important factor of this work is the distance. Distance is very important as it is the prime parameter which will determine the maximum distance from which object can be detected and recognized. This distance is the distance of the object from the camera. The camera will be mounted on the stick and the objects will be in front. Here, for Frame-2 the person is at the distance of 5.4 feet from the camera, the bottle is at the distance of 7.1 feet while the TV monitor is at the distance of 7.2 feet. All these distances are measured manually. Now, all this information is converted into audio. As an output, the user will be able to hear the class/identity of the object along with its position. So for example in case of Frame-2, the user will be able to hear the following audio: “*Bottom Centre Person*”, “*Mid Left Bottle*”, “*Top left TV monitor*”. All this will be heard one after the other in continuation with no breaks in between. After the audio for this frame is heard, again the next frame begins and again the objects are detected, analysed and the audio for that frame is heard.

Thus the overall flow of the project is summarized as follows:

- 1) Camera is initialized.
- 2) Frame-1 is started at $t=0$.
- 3) For Frame-1, objects are detected, the class of the object is identified along with its location and precision.
- 4) The audio of the above mentioned information is heard by the visually impaired.
- 5) Frame-2 starts repeating the same steps as that for Frame-1, giving the corresponding audio output.
- 6) Finally, at the end of the Frame-3 we get the total time taken by the process to complete 3 frames. Here, Frame3 ends at $t=28.29$ seconds.

The time measurement is with the help of Time Module of python. This enables initializing a timer using start and stop commands as per requirement. The total time refers to the time from which the camera is initialized to capture the

image till the time at which the 3rd frame ends. So at $t=0$ the Frame-1 starts and the Frame-3 ends its detection at $t=28.29$ seconds. As the objects are detected it is immediately converted to audio through text-to-speech. So this time is the combination of the object detection, object identification and hearing of the audio.

It can be understood that time factor should be as small as possible. Because “time” factor reflects how long it will take for the user to get the audio. Therefore, it is very important that the entire process right from the object detection, object identification, position identification till conveying audio instructions to user has to be faster. As camera feeds the frames to the algorithm continuously, the fast processing ensures that there are no latencies in the output generated. The faster, continuous and latency free output means hassle free commutation of user. This is achieved by use of parallel processing power of GPU.

Identification, position identification till conveying audio instructions to user has to be faster. As camera feeds the frames to the algorithm continuously, the fast processing ensures that there are no latencies in the output generated. The faster, continuous and latency free output means hassle free commutation of user. This is achieved by use of parallel processing power of GPU.

7 Conclusion

The purpose of EOA is to guide the visually impaired user for the hassle free commutation experience. In this work, we have used OpenCV and tensor flow for capturing the images and Object detection respectively. Through YOLO V3 and Darknet the objects along with their position are identified. The work also focuses on generation of the results in minimum time possible which reduces the output latency. The Graphics Processing Unit (GPU) with its parallel processing capability is a useful tool for this purpose. Therefore, our aim is to create device that will provide artificial eyes to the visually impaired people. As a result this is an aimed combination of several subsystems that collectively make EOA for assistance of user and making their commutation safe [22]. Adding further we can add various features like night vision or safe path for overcoming the obstacles in front of them. Also we can use Raspberry Pi or Jetson Nano. This will not only reduce the overall cost of the system but will also make the system optimize and compact. This will make it easy and handy for the visually impaired people to carry anywhere.

References

1. M. H. Putra, Z. M. Yussof, K. C. Lim, S. I. Salim “Convolutional Neural Network for Person and Car Detection using YOLO Framework”, Journal of Telecommunication, Electronic and Computer Engineering,2014.

2. C. Liu, Y. Tao, J. Liang, K. Li and Y. Chen, "Object Detection Based on YOLO Network", IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2018.
3. Esteban Reyes, Cristopher Gomez, Esteban Norambuena, Javier Ruizdel-Solar, "Near Real-Time Object Recognition for Pepperbased on Deep Neural", Computer Science book series (LNCS, volume 11374) ,2018.
4. Mohammad Javad Shafiee, Brendan Chywl, Francis Li, Alexander Wong, "Fast YOLO: A fast you only look once system for real-time embedded object detection in video", 2017.
5. Anupam Sobti, Chetan Arora, M. Balakrishnan, "Object Detection in Real-Time Systems: Going Beyond Precision", IEEE Winter Conference on Applications of Computer Vision (WACV), 1020-1028, 2018.
6. Michal Irani and P. Anandan, "Unified Approach to Moving Object Detection in 2D and 3D Scenes", IEEEExplore, June 1998.
7. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", Proceedings of the IEEE conference on computer vision and pattern recognition, 779-788, 2016.
8. Martin Simony, Stefan Milzy, Karl Amendey, Horst-Michael Gross, "Complex-YOLO: An euler-region-proposal for real-time 3d object detection on point clouds", The European Conference on Computer Vision (ECCV) Workshops, 2018.
9. Rachel Huang, Jonathan Pedoeem, CuixianChen, "YOLO-LITE: A realtime object detection algorithm optimized for non-GPU computers", IEEEExplore, 2018.
10. Zhihuan Wu, Xiangning Chen, Y Gao, Y Li, "Rapid target detection in high resolution remote sensing images using Yolo model", International Archives of the Photogrammetry (ISPRS), Remote Sensing and Spatial Information Sciences, Volume XLII-3, 2018.
11. Alessandro Dionisi, Emilio Sardini, Mauro Serpelloni, "Wearable object detection system for the blind", IEEE International Instrumentation and Measurement Technology Conference Proceedings, 1255-1258, 2012.
12. Niranjana Debnath, Jaya Bharati Thangiah, Shamala Pararasaingam, Syed Abdul Kader Aljunid, "A mobility aid for the blind with discrete distance indicator and hanging object detection", IEEE Region 10 Conference (TENCON), 664-667, 2004.
13. M Naveen Kumar, K Usha, "Voice based guidance and location indication system for the blind using GSM, GPS and optical device indicator", International Journal of Engineering Trends and Technology (IJETT), 2013.
14. S. Guennouni, A. Ahaitouf and A. Mansouri, "Multiple object detection using OpenCV on an embedded platform", Third IEEE International Colloquium in Information Science and Technology (CIST), Tetouan, 2014.
15. S. Mane and S. Mangale, "Moving Object Detection and Tracking Using Convolutional Neural Networks", Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018.
16. S. Hayat, S. Kun, Z. Tengtao, Y. Yu, T. Tu and Y. Du, "A Deep Learning Framework Using Convolutional Neural Network for MultiClass Object Recognition", IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, 2018.
17. P. Strumillo, "Electronic interfaces aiding the visually impaired in environmental access, mobility and navigation", 3rd International Conference on Human System Interaction, Rzeszow, 2010.
18. Asha G. Hagargund, Sharsha Vanria Thota, Mitadru Bera, Eram Fatima Shaik, "Image to Speech Conversion for Visually Impaired", International Journal of Latest Research in Engineering and Technology (IJLRET) ISSN: 2454-5031.
19. Z. Chen, R. Khemmar, B. Decoux, A. Atahouet and J. Ertaud, "Real Time Object Detection, Tracking, and Distance and Motion Estimation based on Deep Learning: Application to Smart Mobility", Eighth International Conference on Emerging Security Technologies (EST), Colchester, United Kingdom, 2019.
20. P. R. Patil, M. D. Patil and V. A. Vyawahare, "Acceleration of hough transform algorithm using Graphics Processing Unit (GPU)," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, pp. 1584-1588, 2016.
21. Parag Patil, Navin Singhaniya, Chaitanya Jage, Vishwesh A Vyawahare, Mukesh D Patil, PSV Nataraj, "GPU Computing of Special Mathematical Functions used in Fractional Calculus", Frontiers in Fractional Calculus, Vol. 199, 2018.
22. Chikane, Sameer and Patil, Parag and Singhaniya, Navin and Jage, Chaitanya and Patil, Mukesh and Vyawahare, Vishwesh, "HumanLike Interpretation of Lines Using Embedded GPU", Conference on Technologies for Future Cities (CTFC), 2019. M.D. Patil, P.S.V. Nataraj, "GPU Computing of Special Mathematical Functions used in Fractional Calculus" **1**, 200-233 (2017) .