[1] **Mohammed Fayiz Ferosh** *

[2] **Pournami Sanalkumar**

[3] **Prathibha P R**

[4] **Sreelakshmi Manoj**

[5] **Hari C V**

Assistive Technology for Navigation of Visually Impaired People

JES Journal of Electrical Systems

Abstract: - In this paper, an assistive navigational technology was proposed for those who are blind. People who are visually impaired are more likely to be physically inactive, move more slowly, and dread falling. As a result, the majority of them lack the confidence necessary to travel on their own in strange places. Despite the fact that some technology advancements can help them, not all of them are affordable to the common man. Through our efforts, we hope to help visually impaired persons feel secure and confident while navigating new surroundings. Assistive tech- nology for navigation of visually impaired people is an affordable and efficient computer vision based blind guidance system that helps the user detect the distance to obstacles well in advance and avoid them. The suggested system also includes an emergency distress alert feature that enables users to communicate their locations in an emergency.

Keywords: Kinect, OpenCV, Raspberry Pi, YOLO, depth image.

I. INTRODUCTION

Visual impairment affects a substantial global population, with 36 million individuals facing complete blindness out of 253 million with visual impairments [1][2]. Navigating unfa- miliar environments poses a formidable challenge for those with visual impairments, often leading to increased physical inactivity, slower mobility, and heightened fear of falling. In addressing these challenges, our initiative aims to significantly mitigate the risks associated with independent navigation for individuals with visual impairments, fostering a safer and more comfortable experience in unknown surroundings.

The distinctive contribution of our endeavor lies in the development of a cost-effective and highly efficient alternative to current navigation solutions reliant on computer vision and image processing. Primarily tailored for individuals with partial or total visual impairments struggling with obstacle de- tection and avoidance, our technology harnesses the power of computer vision to recognize obstacles and precisely measure their proximity to the user. Real-time feedback is seamlessly delivered to the user through connected headphones, enhanc- ing their spatial awareness. Moreover, the system incorporates a robust distress alert mechanism to swiftly respond to emer- gency situations. Through these innovations, our objective is to revolutionize the landscape of navigation assistance, empower- ing individuals with visual impairments to navigate confidently and independently in diverse and unfamiliar environments.

II. LITERATURE REVIEW

The You Only Look Once (YOLO) network [3] is one of the representatives of the single-stage object detection network. It directly treats the detection task as a regression for fast detection. By comparison with the region-based convolutional neural networks [4], it has a faster detection velocity and can implement end-to-end prediction. YOLOv4-tiny [5] is the lightweight version of YOLOv4 [6] and has a simpler construction and good performance in embedded devices for object detection. This algorithm [7] has proven to be useful in situations where images are blurry, small objects are present, or objects are present under different climate conditions where they are clouded or foggy and cannot be easily detected.

According to Chetan Kumar B et al. [8], YOLOv4 accu- rately detects several objects with an efficiency of about 99% based on an input video dataset. Additionally, as evidenced by the experimentation in [8], the model's entire output is reliant on the quality of the training dataset utilised, which in turn affects how efficiently the model

[1, 2, 3, 4, 5] Department of Applied Electronics & Instrumentation Engineering, Rajagiri School of Engineering & Technology (Autonomous), Kochi, Kerala, India

* Corresponding Author Email: 159fayiz@gmail.com

performs under various environmental situations. Because of this, in our work, we employ the YOLOv4-tiny model for object detection, which was previously trained using the MS-COCO dataset [9].

A depth camera, the Microsoft XBox Kinect Sensor, of- fers accurate depth information about the surroundings in its sensing area. According to Riyad A. El-Iaithy et al. in their paper [10], the Kinect sensor essentially employs an RGB and infrared camera to calculate the true depth. As stated in [10], it is feasible to acquire 3D video data in any lighting condition thanks to the presence of both RGB and infrared cameras. In our work, we detect obstacles in the user's path using this depth data from the Kinect Sensor, and we calculate the distance to each obstacle separately. As stated in [10], in order to connect the Kinect with Raspian OS (a Linux- based operating system) on the Raspberry Pi, we also use the OpenKinect open-source software that was made available for Linux PCs.

Another study by Jungong Han et al. [11] examined how Kinect was being used by researchers to improve computer vision algorithms. In essence, this paper analyses the computer vision issues—such as object tracking or indoor 3-D mapping, etc.—that Kinect can solve or improve. As the authors note in the publication, this paper gave us insight into all of the tasks we can accomplish with the Kinect; therefore, we also believe that it serves as a reference for computer vision researchers

using the Kinect.

The YOLO algorithm is used to detect objects, which is one of the main components of our work. Convolutional neural networks, which predict multiple bounding boxes and class probabilities for those boxes simultaneously, form the neural network that underlies the YOLO algorithm. The work by Eli- ganti Ramalakshmi et al. [12], which included object detection and distance calculation towards humans on the Raspberry Pi for the visually impaired, was another work we identified with some objectives comparable to ours. In contrast to the object detection algorithm employed by Eliganti Ramalakshmi et al. in their work, which is YOLOv1, we use YOLOv4-tiny because it is much faster and optimised for embedded devices. The main distinction between our work and that of [12] is that, while our work uses the true depth information provided by the depth camera to calculate the distance towards obstacles detected, [12] uses the Haar cascade classifier to distinguish humans from obstacles detected and calculates the distance towards them using a theoretical approach based on the width and height of the bounding box of detection, which may or may not be accurate. Furthermore, unlike [12], our approach will be able to determine the distance to any detected obstacle, not just humans. In our work, the user is guided down an obstacle-free path, which is not done in [12]. All obstacles in front of the user are detected in [12]. However, our work just focuses on identifying obstacles in the user's path. For emergency scenarios, our work also includes a distress alert system.

III. Methodology

The proposed system makes use of the YOLOv4-tiny al- gorithm. YOLOv4-tiny is a compressed version of YOLOv4 intended for use on machines with limited computing power. The system uses a Raspberry Pi 4B (4 GB) as the host controller. A Kinect sensor captures images around the user, and using the YOLOv4-tiny algorithm, it detects obstacles. The user is then alerted using an audio feedback message that gives details of both the obstacle ahead and the distance towards it. The system also contains a button that, when pressed, sends the exact location of the user to a specific mobile number.

The steps in the functioning of this system are as follows:

- STEP 1: Read the RGB image from the Kinect sensor v1 and also, in parallel, read depth data from the Kinect sensor.
- STEP 2: The RGB image is then pre-processed and the YOLOv4-tiny object detection algorithm is applied.
- STEP 3: If an object is detected in the user's path, it is identified using YOLO, and the distance toward it is calculated from the depth data. Both of these are obtained as feedback, which is converted from text to speech for the user to hear.
- STEP 4: The depth data from the Kinect sensor is con- verted into a depth image, and a colour mask is applied to isolate objects within near detection.
- STEP 5: The contour area is measured in the colour- masked image in the user path and on both sides, de- pending on which user guidance command is obtained.
- STEP 6: Through text-to-speech conversion, this com- mand is given to the user as audio feedback to move left, right, forward, or if all ways are blocked in front of the user.

- STEP 7: A button is incorporated in the system, which, when pressed, gets the latitude, longitude, and time data of the user from the GPS module and sends it to a specific mobile number using the GSM module.

Fig. 1 represents the wiring diagram of the system. For the software part, all the coding on the Raspberry Pi is done in the Python programming language. The YOLOv4- tiny object detection algorithm pre-trained on the MS-COCO dataset is used. The GPS data is obtained in NMEA format, which is converted to the required location and time data. We create a Google Maps link using the latitude and longitude information, which we then send as an SMS along with the UTC time. The user guidance part was realised using two functions, namely obstacle detection and obstacle-free path guidance based on contour area measurement, where the obstacle detection function also calculated the distance towards the obstacle.

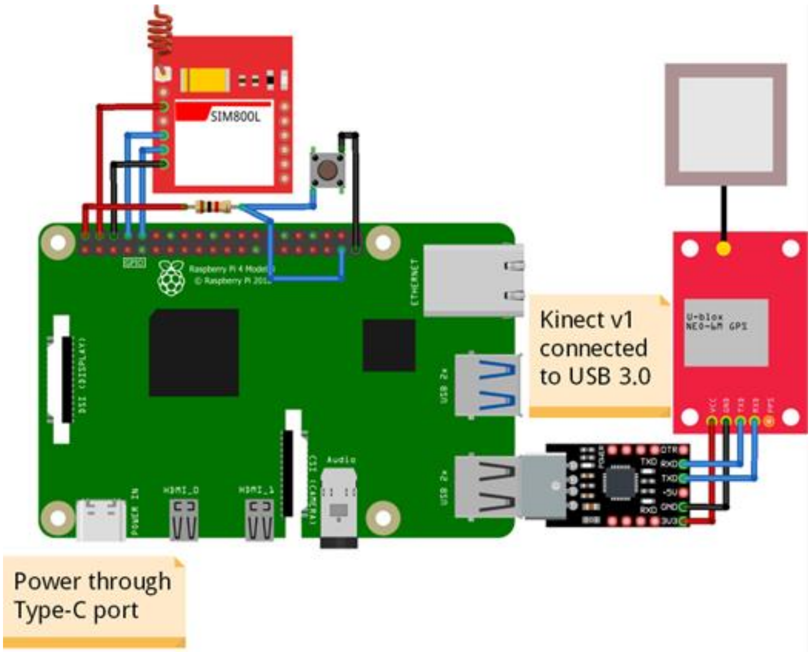

Fig. 1. Wiring Diagram

IV. Experimental Setup

We performed studies both indoors and outdoors to evaluate the prototype's functionality and efficacy. Even though the arrangement is still not entirely portable, testing was made pos- sible by the Kinect camera's long cord. The performance of the prototype was assessed based on whether the person reached the endpoint by dodging all obstacles in the experimental setup, which consisted of a single person wearing a blindfold and the Kinect camera mounted on his chest in the middle with straps as shown in Fig 2. For the outside scenario, a college open hallway was used and the obstacles were unpredictable. Indoor locations comprised lab and classrooms with tables, chairs, and other similar impediments as obstacles.

Fig. 2. Illustration of indoor experiment

V. Results

The prototype of assistive technology for navigation of visually impaired people was successfully accomplished with the help of Python programming on the Raspberry Pi, where the Raspberry Pi acts as the host controller. It is done with the help of the YOLOv4-tiny pre-trained object detection algorithm and a Kinect sensor for the user guidance part, and GPS and GSM modules and a button for the emergency distress alert part. Initially, we successfully obtained an RGB image and raw depth data as a grayscale image from the Kinect sensor, as shown in Fig. 3.

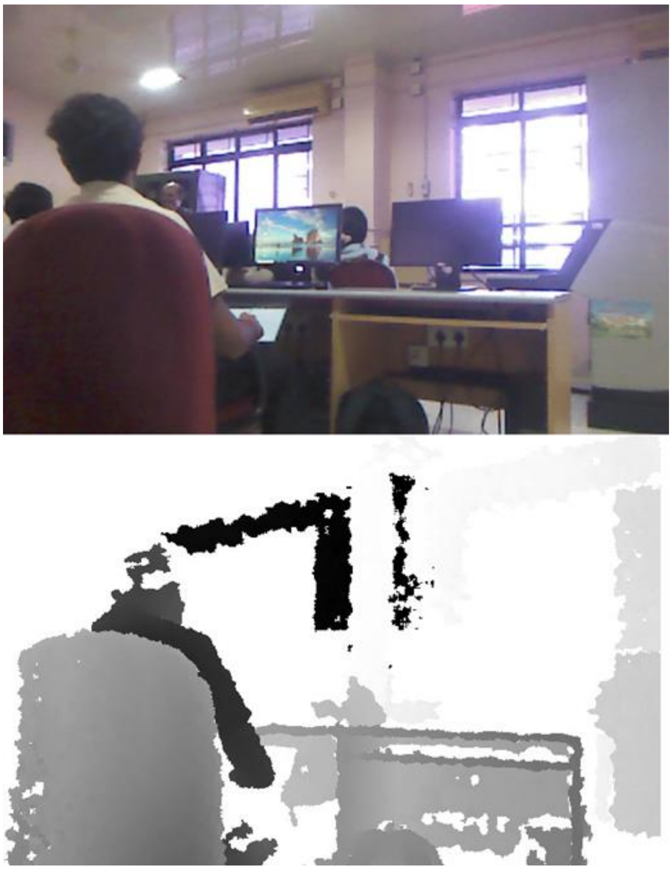

Fig. 3. RGB image and raw depth data as grayscale image from Kinect

A. Obstacle detection

To implement the obstacle detection function, two lines were drawn on the RGB image to segment the user path using OpenCV, and the RGB image obtained is fed to the YOLOv4- tiny network, which is run using OpenCV DNN, and in the output obtained, we filter the obstacles in the user path based on the two lines drawn, giving us the obstacles in the user path shown in Fig. 4. Now, from filtered detections, we calculated the centroid of the bounding box of these detections, converted the raw depth data of the centroid to actual distance, and also took the class of the detection and used it to give a voice alert to the user using the Python text to speech library. For example, the voice command output for the detection in Fig. 4 is ”person at 2.3 metre distance”.

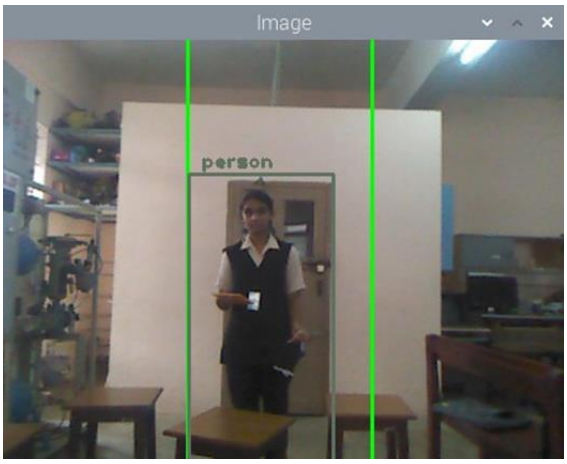

Fig. 4. Obstacle detection in user path

B. Obstacle free path guidance

For implementing the obstacle-free path guidance function, initially we created a depth image from the raw depth data where each colour denoted a specific distance, as shown in Fig. 5 where reddish colours denoted the nearest points and violet or blueish colours denoted the farthest points. The depth image was created using the PyGame library, and the points out of range of the Kinect sensor or where the distance couldn't be properly determined by the Kinect sensor are shown by black pixels in the depth image. Once we obtained the depth image, we applied a colour mask to it to isolate near detection, where we actually isolated the reddish colour, and on the resultant image, we drew two lines as mentioned before in the obstacle detection function in order to segment the user path as shown in Fig. 6. Once this was done, we basically just measured the contour area in all 3 segments and derived the user guidance commands based on this. The process behind the user path determination starts with the contour area measurement of the user path segment, and unless the area threshold is met, the voice command is "move foreward". And, in the case where the area threshold is met, which literally means there is an obstacle in front of the user, then the contour areas of both the left and right sides are measured, and the side with the minimum contour area is chosen as the ideal path with the voice command "move right" or "move left" unless the area threshold of the side segment is met, in which case it means all the ways are blocked for the user, where we used the voice command "all ways are blocked".

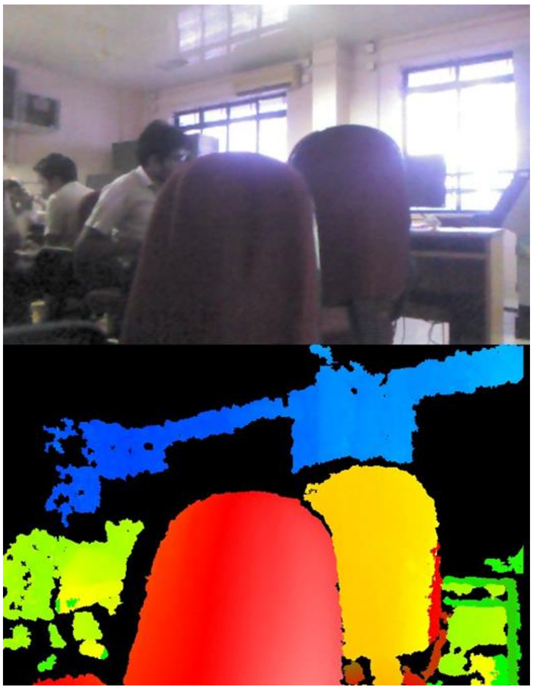

Fig. 5. Depth image created from RGB image

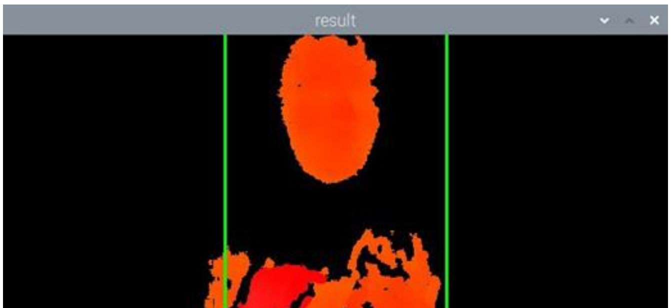

Fig. 6. User path segmented near detection depth image

C. Emergency distress alert

This was implemented by collecting location and time data from the GPS module when the button was pressed and sending an SMS to a desired mobile number with this data using the GSM module. The SMS received by the desired mobile number owner is shown in Fig. 7.

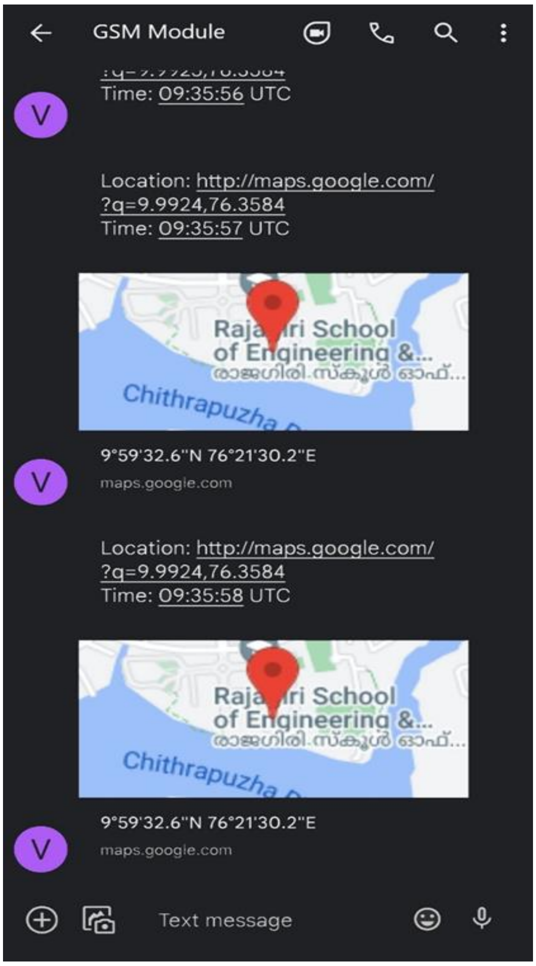

Fig. 7. SMS recieved by desired mobile number owner

Fig. 8 shows the prototype named assistive technology for navigation of visually impaired people that was developed for obstacle-free path guidance for visually impaired people as a major concern. The system is cost-effective compared to existing solutions.

D. Performance Analysis

TABLE I illustrates the results of the tests performed in the experimental setting. As you can see, the majority of the data in the table is self-explanatory, with the exception of" Accuracy (%)," which is simply an examination of how well the prototype worked and how effectively the user followed

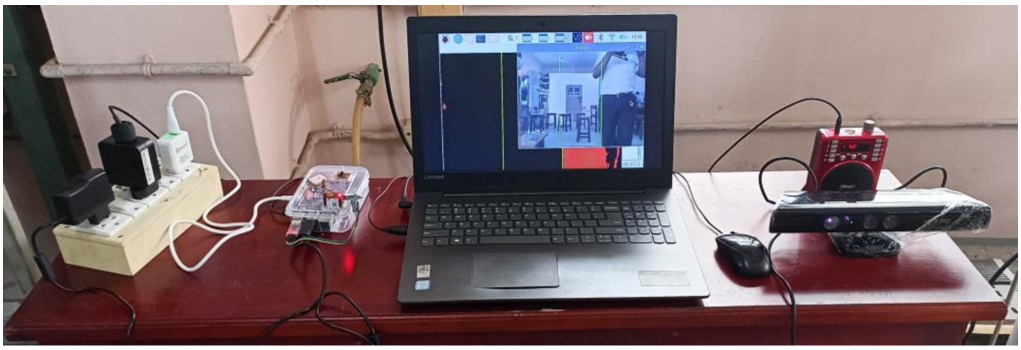

Fig. 8. Assistive Technology for Navigation of Visually Impaired People

Table I: Experimental Setup Test Results

Experimental Region	Time Taken (min)	Distance (m)	Approx. Total No. of Obstacles	Static Obstacles	Moving Obstacles	Max Speed of Moving Obstacle (m/s)	User Reached Destination?	Accuracy (%)
College Classroom - Run 1	3.0	15	3	2	1	0.4	Yes	98
College Classroom - Run 2	3.2	15	7	4	3	0.5	Yes	96
College Classroom - Run 3	3.4	15	5	3	2	2.2	Yes	89
College Lab - Run 1	3.7	20	5	3	2	0.6	Yes	99
College Lab - Run 2	3.8	20	4	3	1	2.3	Yes	87
College Lab - Run 3	3.5	20	6	4	2	0.7	Yes	97
College Hallway - Run 1	8.5	30	14	6	8	1.5	Yes	65
College Hallway - Run 2	13.0	30	15	5	10	2.7	No	29
Outside Scenario - Run 1	10.0	30	28	11	17	2.5	No	35
Outside Scenario - Run 2	11.5	30	23	9	14	2.2	No	42

the commands provided by the prototype to reach the destina- tion. Assume that the maximum speed of a student strolling inside a classroom or lab is around 0.7 m/s, whereas the maximum speed of a student dashing across the classroom is approximately 2.3 m/s. With these parameters in mind, we can confidently state that the prototype operated successfully in an indoor situation where the majority of impediments were static, allowing the user to repeatedly reach the endpoint in both a lab and a classroom context. However, if a very fast moving obstacle was present, the prototype struggled to recognise it properly, though this drawback depended on the speed of the obstacle. For instance, when a student raced across in front of the user, it was not adequately recognised and no input was given to avoid collision. In contrast, a person strolling to a nearby desk was properly detected and the appropriate feedback to avoid contact was given by the prototype. In the outside setting and the college hallway, the user was only able to reach the destination once without colliding with anything. The college hallway is also regarded as an outdoor setting because it is an open space that is similar to an outdoor environment. The setup wasn't functioning effectively because there were too many quick and movable impediments in the outdoor environment. This led to either no detection at all or incorrect detections, which caused the prototype to give the incorrect feedback, which in turn caused repeated collisions in the outdoor scenario. The time taken in the college hallway and outdoor scenarios in which the user did not reach the destination was calculated as until the time the user determined the test run to be futile due to incorrect or no commands from the prototype resulting in multiple collisions preventing him from reaching the destination.

VI. Conclusion And Future Scope

Assistive technology for navigation of visually impaired people helps guide the user (who is visually impaired) through an obstacle-free path and prevent collisions. It also helps the user send a distress alert in emergency situations. The solution is effective in virtually all indoor settings with mainly static impediments. Since this is just a prototype, further work using the most recent technologies can be done to make the system detect fast-moving obstacles and completely portable by running the entire system off a battery. We are confident that in the future, the visually impaired will find this system to be very useful for navigating unfamiliar indoor and even outdoor environments. The system can be further made lightweight by using the latest technology and miniaturised components.

VII. Conflict Of Interest

The authors have no conflict of interest to declare.

VIII. Funding

The authors received no specific funding for this work.

References

[1] The Lancet Global Health, "Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the Global Burden of Disease Study", GBD 2019 Blindness and Vision Impairment Collaborators, vol. 9, no. 2, pp. 130-143, Feb. 2020, doi: 10.1016/S2214-109X(20)30425-3.

[2] The Lancet Global Health, ”Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: the Right to Sight: an analysis for the Global Burden of Disease Study”, GBD 2019 Blindness and Vision Impairment Collaborators, vol. 9, no. 2, pp. 144-160, Dec. 2020, doi: 10.1016/S2214-109X(20)30489-7.

[3] Joseph Redmon et al., “You Only Look Once: Uni- fied, Real-Time Object Detection”, presented at IEEE Conference on Computer Vision and Pattern Recogni- tion (CVPR), Las Vegas, USA, Jun. 27-30, 2016, doi: 10.1109/CVPR.2016.91.

[4] Jeong-ah Kim et al., “Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recog- nition”, presented at IEEE International Conference on Consumer Electronics - Asia (ICCE), Seoul, Korea, Nov. 1-3, 2020, doi: 10.1109/ICCE-Asia49877.2020.9277040.

[5] Zicong Jiang et al., ”Real-time object detection method based on improved YOLOv4-tiny”, arXiv, Nov. 2020, doi: 10.48550/arXiv.2011.04244.

[6] Alexey Bochkovskiy et al., ”YOLOv4: Optimal Speed and Accuracy of Object Detection”, arXiv, Apr. 2020, doi: 10.48550/arXiv.2004.10934.

[7] Donglin Zhu et al., “Object Detection in Complex Road Scenarios: Improved YOLOv4-Tiny Algorithm”, pre- sented at 2nd Information Communication Technologies Conference (ICTC), Nanjing, China, May. 7-9, 2021, doi: 10.1109/ICTC51749.2021.9441643.

[8] B Chethan Kumar, Punitha and R Mohana, “YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Ap- plications.”, presented at Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, Aug. 20-22, 2020, doi: 10.1109/IC- SSIT48917.2020.9214094.

[9] Tsung-Yi Lin et al., ”Microsoft COCO: Common Objects in Context”, Computer Vision – ECCV 2014, vol. 8693, pp. 740–755, May. 2014, doi: 10.1007/978-3-319-10602-1 48.

[10] Riyad A. El-Iaithy, Jidong Huang and Michael Yeh, “Study on the use of Microsoft Kinect for robotics appli- cations”, in Proc. of the IEEE/ION Position, Location and Navigation Symposium, Myrtle Beach, SC, USA, April. 23-26, 2012, doi: 10.1109/PLANS.2012.6236985.

[11] Jungong Han et al., ”Enhanced Computer Vision With Microsoft Kinect Sensor: A Review.”, IEEE Trans. on Cybernetics, vol. 43, no. 5, pp. 1318–1334, Oct. 2013, doi: 10.1109/TCYB.2013.2265378.

[12] Eliganti Ramalakshmi, Dixitha Kasturi and Gouthami V, ”Object Detector for Visually Impaired with Dis- tance Calculation for Humans”, International Jour- nal of Engineering and Advanced Technology (IJEAT), vol. 9, no. 4, pp. 834-838, April. 2020, doi: 10.35940/ijeat.D7868.049420