Grenze International Journal of Engineering and Technology, June Issue

Deep Learning Virtual Assistant for Visually Impaired with Object Detection and Distance Estimation

Kancherla S. D. V. N. Sai Haritha[1], Girija Ravulapalli[2] and Nalluri Sunny[3]
[1-3]VR Siddhartha Engineering College/Computer Science Engineering (CSE), Vijayawada, India.
Email: 198w1a0523@vrsiddhartha.ac.in, 198w1a0550@vrsiddhartha.ac.in , sunny@vrsiddhartha.ac.in

***Abstract*—According to WHO (World Health Organization) 285 million individuals worldwide suffer from vision impairment. These metrics may triple in the coming 30 years. Thus, there is a need to develop a cost-effective guiding system for those people. This system provided a solution using large dataset COCO (Common Objects in Context) dataset which consists of 90 classes of real-time objects to cover all the day-to-day life objects It used the Single Shot Detector algorithm for object detection. It makes the work of blind people easier, and more efficient by transferring wireless voice-based feedback about whether an object is too close to them or at a safer distance. Google's text-to-speech engine is being used to provide audio-based output. It estimates the distance from person to object using depth extraction methodology with focal length as a scaling factor, real object width. This model provides a guiding system to help visually impaired people.**

***Index Terms*— Object detection, Single Shot Detector (SSD), Scaling Factor, Depth Estimation, Google Text to Speech engine(gTTS).**

I. INTRODUCTION

Object detection is a computer vision task that involves detecting and localizing objects of interest within an image or video Object detection seeks to identify and classify the position of objects within an image or video. This is divided into two parts: object localization and object classification. Drawing a bounding box around an object helps in object localization, which is the process of locating it within an image. Object classification entails labelling or categorizing each detected object.Object detection can be accomplished using a number of different methods, including conventional methods of computer vision and deep learning-based approaches. Deep learning-based approaches have advanced significantly in recent years, particularly with the development of convolutional neural networks (CNNs) and the availability of large annotated datasets [1].

SSD (Single Shot Multi Box Detector) is cutting-edge object detection technology [2]. SSD is a deep learning-based object detection framework with a high accuracy and real-time detection speed. The SSD architecture is built around a single convolutional neural network (CNN) that can detect and classify objects at the same time. It detects objects of various sizes and aspect ratios by employing a multi-scale feature map extraction process.

To detect objects, SSD employs a set of default bounding boxes, also known as prior boxes, at various positions and scales in the image. These prior boxes serve as reference points for predicting object location and size. The CNN estimates the offset and confidence score for each prior box, which are then used to refine the predicted object's position and size. The SSD architecture's speed is one of its primary benefits. Unlike other object detection frameworks, SSD detects objects in a single pass through the network, which makes it much faster than

Grenze ID: 01.GIJET.9.2.552_1

other approaches that require multiple network passes.
MobileNetV3[2] is a neural network architecture family designed for high-performance computer vision applications on mobile and embedded devices. It was introduced by Google as a descendant to MobileNetV2. The MobileNetV3 architecture is made up of a series of blocks that use various techniques to improve network efficiency and accuracy, such as squeeze-and-excitation (SE) and inverted residual (MBConv) blocks. The SE block assists the network in adaptively recalibration of the feature maps by learning the importance of each channel, whereas the MBConv block reduces the network's computational cost by utilising a bottleneck structure. The Fig 1. represents the architecture of MobilenetV3.MobileNetV3 has two variants: MobileNetV3-Large and MobileNetV3-Small. MobileNetV3-Large is designed for high-accuracy tasks with a large number of parameters, whereas MobileNetV3-Small is intended for low-latency applications with fewer parameters. MobileNetV3 is an efficient and high-performance neural network architecture designed for mobile and embedded devices
Depth estimation is a computer vision task that entails estimating the distance or depth of objects in a scene based on a two-dimensional image or video. Depth estimation can be accomplished using a variety of techniques, including traditional computer vision methods and deep learning based approaches. Stereo vision and structured light techniques are examples of traditional methods, whereas deep learning based techniques use convolutional neural networks (CNNs) to learn the depth from an image.
gTTS (Google Text-to-Speech) is a python library that converts written text into spoken words by utilising the Google Text-to-Speech API. It can generate audio files in a variety of formats, including MP3 and WAV, by passing a text string as input. With gTTS, you can generate speech from any text string in over 100 languages and save it as an audio file. The language and speed of the speech, as well as the audio format and quality, can all be customized.

II. Related Work or Literature Studies

Hongyu et al. [3] designed a blended deep-learning model to utilize the benefits of deep neural networks, and conventional object detection schema for object detection. The framework is an end-to-end trained PASCAL VOC dataset, COCO datasets are used extensively for experiments. They modularized the deep regionlet framework into two parts. Firstly, a full convolution network extracts features based on Region Selection Network with 3 CNN layers to detect all generic real-time objects nonrectangular region selection is deployed.
Piccialli et al.[4] designed a robotic collaborative top-view surveillance framework. It involves two modules. One for object detection. He trained the top view real-time dataset on two single-stage object detection algorithms YOLO with Darknet architecture, and SSD with VGC for feature extraction, and compared their performance. Though YOLO provides more speed than SSD, SSD wins at the point of accuracy. SSD provides more flexibility over YOLO which divides images into grid cells, by having several predefined bounding boxes that can be used based on our target object size.
Ziying et al.[5] proposed a new model to detect small objects more accurately than all other object detection models. Multi Scale YOLO is used along with the new extraction network CourNet.YOLOV5 is used as the backbone, and CourNet is the addition of two convolution layers. A shallow network is used to detect low-feature small objects. Deep networks for complex large objects. YOLO multi-scale feature also adopted Path Aggregation Network to strongly detect small-scale objects and can save spatial details to mark pixels from the feature maps.
Zhao et al.[6] provided a complete review of generic object detection, challenges of object detection, such as varying object scales, orientations, and occlusions, as well as the need for efficient and accurate detection. They also provided a review of various deep learning-based object detection approaches, including region-based, one-stage like YOLO and SSD, and two-stage detectors like Fast RCNN. They compare and contrast these approaches, discussing their respective strengths and weaknesses as region-based detectors tend to have higher accuracy but can be computationally expensive, while onestage detectors are faster but can struggle with detecting small objects.
Kalake et al.[7] provided an in-depth overview of recent deep learning-based approaches for multi-object tracking. They discussed the challenges of multi-object tracking, including occlusions, object appearance variations, and object interactions. These challenges can be overcome by Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). One-stage trackers, such as Deep SORT and JDE, directly predict object identity and bounding boxes in a single forward pass through the network. Two-stage trackers, such as Faster R-CNN and Mask R-CNN, first generate region proposals and then refine them with a separate network for identity and bounding box regression.

Ashiq et al. [8] proposed a model with Raspberry Pi Digital signal processing (DSP) board with GSM and GPS module, head phones, and a camera for object detection. The DSP detects the live video stream via the camera and sends it to the object detection-CNN model. This CNN model recognises the object and sends its name to the text to speech converter module, which then produces voice output through the headphones. A labelled snapshot along with its location, is stored in the server's database using the JPEG encoder.
Gong et al.[10] created a model for precise identification and validation of moving human targets. For detecting real- time positioning and recognition networks, a single shot detector (SSD) based on the multi-scale feature fusion idea (IMFF) is used. The camera was set up to collect images with a resolution of 19201080 for testing purposes. The final sample is made up of 952 valid samples of 10 people invarious positions and angles. The current model IMFF-SSD has a high detection accuracy when compared to R-CNN,SPP-Net, faster-R-CNN, YOLD, and SSD.
Tian et al.[11] proposed a system for dynamic crosswalk scenes. It detects crosswalks, vehicles, and pedestrians and determines the status of pedestrian traffic lights. The mobile is equipped with an Intel Real sense camera that detects the surroundings of individuals with vision impairments and transmits information via audio signal. They looked at three datasets: the pedestrian traffic light dataset, the key objects on Crossroad dataset, and the crosswalk dataset. This model was used to create the Sensing AI G1 assistive device.
Khan et al.[12] developed a model using camera and sensors to avoid obstacles and detect objects. It proved the functionality of the proposed prototype using a Raspberry Pi 3 Model B+. An ultrasonic sensor can be used to measure the distance between the obstacle and the user. It also allows users to read, acting as a reading assistant by converting images to text and providing audio output. The proposed model can be viewed through a pair of glasses. TensorFlow object detection API, OpenCV, and the Haarcascade classifier were used to detect eyes and faces as well as measure distance. Tesseract (OCR engine) was used to convert images to text, and the Speak module was used to convert text to speech. The obstacle within 40-45 inches is identified and a voice alarm is generated

III. Motivation

Object detection algorithms can be incredibly useful for blind people because they provide a means for them to understand and navigate the world around them. Blind individuals often rely on alternative senses, such as touch and hearing, to interact with their environment, but these senses may not always provide the level of information needed for independent navigation. The motivation for developing object detection algorithms for blind people is to provide them with greater independence, safety, and access to information, allowing them to interact with the world in a more meaningful way.

IV. Problem Domain

This model can estimate distance for only cell phone, person, bottle, laptop. This model can detect all object classes of the COCO dataset. (90 classes). This model considers web camera-based image input, only objects falling within the focal length range of the web camera can be detected.

V. Problem Statement

Daily tasks like reading a book and getting about places might be difficult for blind individuals. There are several instruments at their disposal to deal with the difficulties they experience, but they are insufficient. Vision is the most important thing a person may have, and it is crucial to their quality of life. People who are visually impaired require aid to do their everyday tasks. With this project, the difficulties experienced by blind individuals are solved, and an advanced solution to help them function in daily life is attempted.

VI. Statement

The statement for this model are to identify objects in day-to-day life, provide proximity of objects i.e. near or far, to provide distance to reach the target destination, to communicate to a person by speech.

VII. Problem Formulation or Representation or Design

This paper proposed a system for detecting objects and estimating the distance of the objects from camera for visually impaired. This model makes uses of OpenCV, SSD, MobilenetV3, Depth estimation, gTTS and Python. The Fig. 2 represents the process flow of proposed system.

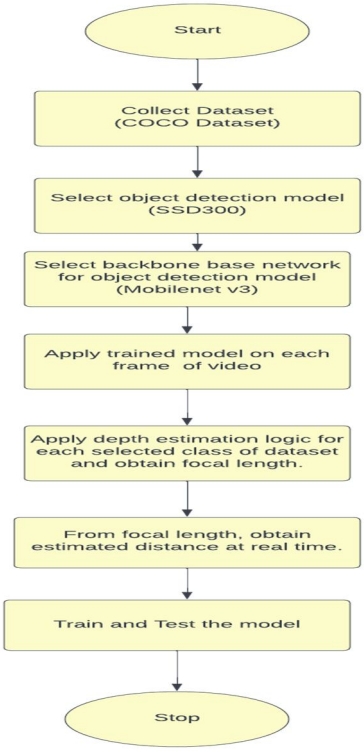

Figure 1. Process Flow Diagram

VIII. Solution Methodologies or Problem Solving

The model was created in 4 stages: Object detection using SSD MobilenetV3, Depth Estimation and Text-To-Speech conversion, Graphical User Interface

A. Object Detection

In this model COCO dataset is being considered to detect the objects[13]. This dataset is considered because, it is a huge dataset consisting of 90 classes and 330K images with annotations. This dataset can be downloaded from its official website. A pre-trained SSD model on the coco dataset is being considered in this model. The Single Shot Detector core algorithm can detect category scores and box offsets for a given set of pre-set bounding boxes by using small convolutional filters implemented to feature maps.[14]. SSD chooses the best default box with the highest IoU for each ground truth box. MobileNetV3 is used as backbone for SSD in this model with image input size as 300x300. The mobileNet's final layers, such as the fully connected layer and SoftMax, are excluded. The final convolutional layer produces a feature map. These feature maps are fed into detection heads in the subsequent layers that detect the objects. The layers of mobileNet are as follows: 1x1 convolutional layer, 3x3 depth wise convolutional layer, ReLU, and batch-normalization. Because depth-wise separable convolutional layers are used in mobileNet, the number of parameters is reduced, making the network lighter in weight when compared to other deep learning models. Depth wise separable convolution entails performing a depth wise convolution followed by a pointwise convolution[15].The first depth wise convolutional layer employs one filter to each input channel, followed by applying point-wise convolution to all depth wise convolution outputs that are inputs for point-wise convolution. In this model, the kernel size, which is the height and width of the convolutional layer, is considered to be three. The distance between two convolutional layers is known as stride, and the value of strides assumed in this model is 2. A batch normalisation with 16 layers is introduced into the model. Theswish layer is introduced in place of the ReLU activation function. Convolutional, batch norm, and relu layers are added to each pointwise and depth wise convolution layer.

B. Depth Estimation

Depth estimation algorithms are used in computer vision to estimate the depth or 3D structure of a scene from a 2D image or video. There are several different approaches to depth estimation, including: Stereo matching: This method involves using two or more cameras to capture the same scene from different angles. By comparing the

images from the different cameras, the algorithm can calculate the depth of each point in the scene[16]. Structured light: In this approach, a pattern of light is projected onto the scene, and the reflections of the pattern are captured by a camera. By analysing the distortions in the way the algorithm can determine the depth of each point in the scene[17].
Time of flight (ToF): ToF cameras emit a pulse of light and measure the time it takes for the light to bounce back from the scene. By measuring the time delay, the algorithm can calculate the depth of each point in the scene[18].
Monocular depth estimation: This method uses a single camera to estimate depth. It typically involves training a neural network on a large dataset of images with known depth values and then using the trained network to predict depth for new images[19].
LiDAR: LiDAR sensors emit laser pulses and measure the time it takes for the light to reflect back from the scene. This method is commonly used in autonomous vehicles for navigation and obstacle detection[20].
Each of these approaches has its own strengths and weaknesses, and the choice of method depends on the specific application and the available hardware. These approaches provide output as a depth map that is relative to other objects' distances. For blind people assistance with a mobile camera depth map output is not suitable
MiDaS (Mixed Depth Estimation of Absolute Scale) belongs to the family of deep learning-based monocular depth estimation networks. It was developed by researchers at the University of Oxford and is designed to estimate the absolute depth of a scene from a single input image. Unlike traditional monocular depth estimation methods that can only estimate relative depth (depth ratios between different points in the image), MiDaS can estimate the absolute depth of a scene in metric units, such as meters or centi meters. To estimate the depth of objects in a scene, MiDaS uses a technique called monocular depth estimation, which infers the 3D structure of a scene from a single 2D image[21]. By estimating the depth of each pixel in the image, MiDaS can provide a dense depth map that captures the 3D structure of the scene. The output using Midas depth estimation is shown in Figure 3. To convert the depth values in the depth map to metric units, you need to know the scaling factor used by MiDaS. The scaling factor isa constant value that is used to convert the depth values from the arbitrary scale used by the neural network to metric units. The scaling factor can be calculated using The ratio of the measured height in the image to the known height in real life can be used to calculate the scaling factor. Absolute depth can be obtained by using below formula:

$$depth\_in\_meters = scaling\_factor / depth\_value$$

Again it isn't easy to obtain the height of every real-world object to derive the scaling factor. MiDaS is a deep neural network that needs a lot of computational power to train and run. A large amount of training data is required to achieve high accuracy. MiDaS is instructed on a specific data set and may not generalize well to new domains. MiDaS is a black box model that can be difficult to interpret or understand. The key limitation, in this case, is that absolute depth values are not interpreted easily. Depth map only gives relative proximity classification of objects based on colours. Distance estimation using focal length:
There is a novel way of estimating the distance to the object. Considering the few classes of COCO dataset such as a person, mobile, etc experimentation is conducted as follows. Initially considered reference objects at reference_distance say 45 inches. Considering reference images for the selected objects being placed at reference_distance and captured those images under reference_images_objectname. Noting down the width of the object in inches as real_width_objectname. Then used the SSD model trained on the COCO dataset to store the reference image width in the frame of each selected object in an array say data_list. Thus calculated the focal length for each object as follows:

$$focal\_length\_objectname = reference\_image\_width\_in\_frame\_objectname * reference\_distance / real\_width\_objectname$$

By capturing and detecting real world objects, having data such real_width_objectname, real_frame_width_objectname and focal_length_objectname from the above experiment. The estimated distance can be calculated as follows:

$$estimated\_distance = real\_frame\_width\_objectname * focal\_length\_objectname / real\_width\_objectname \text{ (in inches).}$$

Thus distance at which the object is located can be estimated.

C. Conversion of Text-To-Speech

Google Text To Speech Conversion (gTTS) is used to convert text output to voice output. The class label of the identified object, distance at which the object is located is provided as input to the gTTS engine. With the help of Digital signal processing, Natural Language processing speech is produced as output. An audio mp3 file is

created first. For each voice output data is written into it and after execution data got erased and used for the next response, the python play sound module is used to play the audio mp3 file. If the object is less than 20 inches distance then the visually impaired is notified by the voice message as the object is coming closer to the camera.

D. Graphical User Interface

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of widgets (UI elements) that can be placed on a canvas to build an interactive application. In Tkinter, one can create windows, buttons, labels, text boxes, menus, and other UI elements, event handling, and call backs to these widgets to create an interactive application.

Tkinter is used in this project because it is easy to learn, cross-platform compatible, large community, customizable, integrated with python

Tkinter library is used in this project for creating the interface. A basic frame with two buttons namely start video, stop video are placed. When the user clicks the start video button camera would turn on and it will start detecting objects and produces voice output to the user. It will stop detecting the objects when the user clicks the stop button. This user interface will facilitate user for using application rather than looking into actual process of working with the code.

IX. Results and Sensitivity Analysis

The results obtained on successful execution of proposed model are shown in this section. The objects in the surroundings are detected and produced the distance of the object as voice output using the proposed system. Figure 3 represents a sample detection of a person and the phone is detected with a voice output as Person is at a distance of 98.89, and the cell phone is at a distance of 83.08(in inches) as shown below. If the object
is at a distance of less than 20 inches then the voice output is generated as the object is coming closer to you and the user will become alerted by this message.

Figure 2 Result image object and distance detection

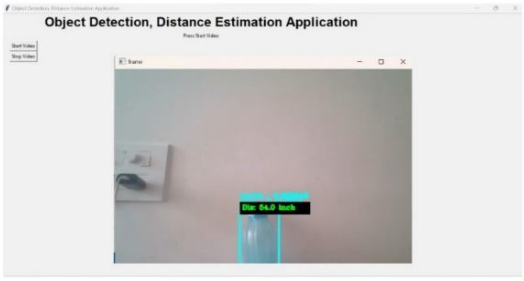

Figure 5.Result image object and distance detection

Figure 5 represents the sample detection of a bottle when the user clicks the start video button on the interface created using the proposed model and provides voice output as Bottle is at a distance of 54(in inches) as shown below.

X. Comparison of Results

This section discusses the effectiveness of the results given by the proposed system. The model performance is evaluated using a confusion matrix [30]. It is also known as an error matrix. A classification model's performance can be seen clearly in a confusion matrix. It aids in evaluating the model's precision [31], recall [32], accuracy, and F1 score [33]. By altering the thresholds and other parameters, a confusion matrix can aid in the optimization of the classification model. In this model, the bottle class of the COCO dataset to test the performance of the model is considered. The proposed model performance is evaluated considering the dataset

of 20 objects collected from the internet source [29] to find the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) using a confusion matrix. The model correctly detected 4 instances of bottles (TP) and missed 6 instances of bottles (FN). It also incorrectly classified 3 instances of bottles (FP) and correctly detected 7 instances of bottles (TP). The confusion matrix of for the above dataset is shown in Figure 6.

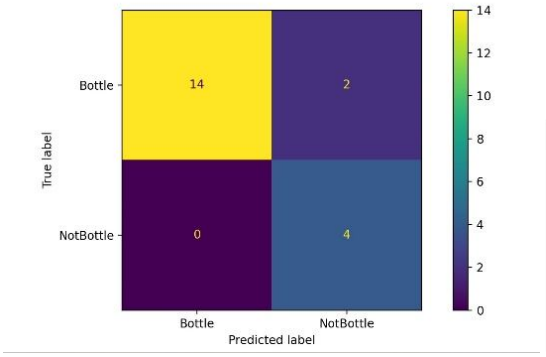

Figure 3.Confusion Matrix for bottle dataset

	precision	recall	f1-score	support
Bottle	1.00	0.88	0.93	16
NotBottle	0.67	1.00	0.80	4
accuracy			0.90	20
macro avg	0.83	0.94	0.87	20
weighted avg	0.93	0.90	0.91	20

Figure 7.Classification Report

The number of cases where the model accurately predicted the positive class is known as true positives (TP).The number of times the model inaccurately predicted the positive class is known as the number of false positives (FP).The number of cases where the model accurately predicted the negative class is known as true negatives (TN).The number of occasions when the model mistakenly predicted the negative class is known as false negatives (FN).

Precision: Precision is estimated as shown in equation 6.1. Precision can range from 0 to 1, with higher values indicating the better performance of the model in predicting the positive class accurately.

Precision = TP / (TP+FP)

Therefore precision of the model considering the bottle class is 1. The recall is estimated as shown in quotation 6.2. Recall can have a value between 0 and 1, with higher values indicating greater model performance in accurately predicting the positive class among all positive cases.

Recall = TP / (TP + FN)

Therefore recall of the model considering the bottle class is 0.88.

A classification report in object detection is a summary of evaluation metrics for each class. It includes details like Precision, Recall, F1Score, and Support. The model evaluated the classification report as shown in Figure 7. In object detection tasks, both recall and precision is important evaluation metrics, as the goal is to accurately detect all objects of interest (high recall) while minimizing false detections (high precision). The number of objects of a specific class in the test set is Support.In some applications such as surveillance, detecting all objects of interest is critical, even if it means some false detections. In this case, the recall may be a more important metric than precision. In applications such as autonomous driving, where false detections can have serious consequences, precision may be a more important metric than recall the mean average precision (mAP) is a commonly used evaluation metric in object detection tasks. mAP considers both recall and precision and provides a measure of the model's overall performance in detecting objects of interest.

F1 Score: The harmonic mean of precision and recall. The F1 Score is calculated as shown above

*F1 score = 2 * (precision * recall) / (precision + recall)*

XI. Conclusion

This model detects objects of the COCO dataset using a Single shot detector object detection algorithm. This model estimates the distance for mobile, cell phone, bottle, and laptop classes. Detected object class labels and estimated distance to the object are provided as audio-based voice output.

Future Work

This model can be extended to estimate distance for all classes of the COCO dataset, to output the voice message in the desired language of the blind person rather than only in English with the study of language translation in the field of Artificial Intelligence.

REFERENCES

[1] Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. Proceedings of the IEEE.
[2] Bradski, G., & Kaehler, A. (2000). OpenCV. Dr. Dobb's journal of software tools, 3(2).
[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multi box detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing
[4] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1314-1324).
[5] Zhao, C., Sun, Q., Zhang, C., Tang, Y., \& Qian, F. (2020). Monocular depth estimation based on deep learning: An overview. Science China Technological Sciences, 63(9), 1612-1627.
[6] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla and R. Chellappa, "Deep Regionlets: Blended Representation and Deep Learning for Generic Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 6, pp. 1914-1927, 1 June 2021, doi: 10.1109/TPAMI.2019.2957780
[7] Ahmed, S. Din, G. Jeon, F. Piccialli and G. Fortino, "Towards Collaborative Robotics in Top View Surveillance: A Framework for Multiple Object Tracking by Detection Using Deep Learning," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 7, pp. 1253- 1270, July 2021, doi: 10.1109/JAS.2020.1003453.
[8] Z. Song, Y. Zhang, Y. Liu, K. Yang and M. Sun, "MSFYOLO: Feature fusion-based detection for small objects," in IEEE Latin America Transactions, vol. 20, no. 5, pp. 823-830, May 2022, doi: 10.1109/TLA.2022.9693567.
[9] Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
[10] L. Kalake, W. Wan and L. Hou, "Analysis Based on Recent Deep Learning Approaches Applied in Real-Time Multi-Object Tracking: A Review," in IEEE Access, vol. 9, pp. 32650-32671, 2021, doi: 10.1109/ACCESS.2021.3060821.
[11] Ashiq, F., Asif, M., Ahmad, M. B., Zafar, S., Masood, K., Mahmood, T., & Lee, I. H. (2022). CNN-based object recognition and tracking system to assist visually impaired people. IEEE Access, 10, 14819-14834.
[12] Khan, S., Nazir, S., & Khan, H. U. (2021). Analysis of navigation assistants for blind and visually impaired people: A systematic review. IEEE Access, 9, 26712- 26734.
[13] X. Wu, D. Hong, Z. Huang and J. Chanussot, "Infrared Small Object Detection Using Deep Interactive U-Net," in IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 6517805, doi: 10.1109/LGRS.2022.3218688.
[14] Tian, S., Zheng, M., Zou, W., Li, X., & Zhang, L. (2021). Dynamic crosswalk scene understanding for the visually impaired. IEEE transactions on neural systems and rehabilitation engineering, 29, 1478-1486.
[15] Khan, M. A., Paul, P., Rashid, M., Hossain, M., & Ahad,
[16] M. A. R. (2020). An AI-based visual aid with integrated reading assistant for the completely blind. IEEE Transactions on Human-Machine Systems, 50(6), 507- 517.
[17] https://cocodataset.org/\#download: last accessed on 04/10/2022
[18] Pressman, Roger. Software Engineering a Practitioner's Approach Seventh Edition. 16 Mar. 2014.
[19] D. Budgen, A. J. Burn, O. P. Brereton, B. A. Kitchenham, and R. Pretorius. Empirical evidence about the UML: a systematic literature review. Software Practice
[20] Gemino, A.,& Parker, D. (2009). Use case diagrams in support of use case modelling: Deriving understanding from the picture. Journal of Database Management (JDM), 20(1), 1-24.
[21] Dumas, M., & Ter Hofstede, A. H. (2001). UML activity diagrams as a workflow specification language. In UML 2001—The Unified Modelling Language. Modelling Languages, Concepts, and Tools: 4th International Conference Toronto, Canada, October 1–5, 2001 Proceedings 4 (pp. 76-90). Springer Berlin Heidelberg.
[22] Bell, D. (2004). UML basics: The sequence diagram. Retrieved July, 17, 2015.
[23] Koonce, B. (2021). MobileNetV3. Convolutional Neural Networks with Swift for TensorFlow: Image Recognition and Dataset Categorization, 125-144.
[24] Adi, K., & Widodo, C. E. (2017). Distance measurement with a stereo camera. Int. J. Innov. Res. Adv. Eng, 4(11), 24-27.
[25] Chen, X., Xi, J., Jin, Y., & Sun, J. (2009). Accurate calibration for a camera–projector measurement system based on structured light projection. Optics and Lasers in Engineering, 47(3-4), 310-319.
[26] Li, L. (2014). Time-of-flight camera—An introduction. Technical white paper, (SLOA190B).
[27] Zhao, C., Sun, Q., Zhang, C., Tang, Y., \& Qian, F. (2020). Monocular depth estimation based on deep learning: An overview. Science China Technological Sciences, 63(9), 1612-1627.
[28] Catapang, A. N., & Ramos, M. (2016, November). Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle. In 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE) (pp. 441-445). IEEE.
[29] Kopf, J., Rong, X., & Huang, J. B. (2021). Robust consistent video depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1611-1621).
[30] Test Images considered for performance evaluation: https://rb.gy/ixxqg
[31] Liang, J. (2022). Confusion Matrix: Machine Learning. POGIL Activity Clearinghouse, 3(4).

[32] Wise, M. N. (Ed.). (1997). The values of precision. Princeton University Press.
[33] Buckland, M., & Gey, F. (1994). The relationship between recall and precision. Journal of the American society for information science, 45(1), 12-19.
[34] Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize F1score. arXiv preprint arXiv:1402.1892