Drishti : A Real-Time Object Recognition for the Visually Impaired

Hitanshu Parekh[1], *Niyati Agarwal*[2], *Pranav Bangera*[3]
Roger D'souza[4], *and Grinal Tuscano*[5]

Abstract

In 2017, the World Health Organization (WHO) reported that nearly 284 million individuals worldwide experienced some degree of visual impairment, with approximately 39 million individuals suffering from total blindness. People with visual impairments often rely on assistance from others or use canes to move around and identify obstacles. Our proposed system aims to aid the visually impaired by identifying and classifying common objects in real-time, as well as recognizing text from various sources such as documents and signs. This system provides voice feedback to enhance understanding and navigation, and utilizes depth estimation algorithms to determine a safe distance between objects and individuals, promoting self-sufficiency and reducing dependence on others. We employ the COCO image dataset, which contains everyday objects and people, and utilize the Mobilenet SSD algorithm for real-time object identification. To enable real-time Optical Character Recognition (OCR) Text-To-Speech functionality, we employ advanced technologies such as OpenCV, Python, and Tesseract for text detection and recognition, and the Pyttsx3 library for converting recognized text into audible speech. Our proposed system is dependable, affordable, realistic, and feasible.

Keywords : **COCO Dataset, Depth Estimation, Machine Learning, Object Detection, Optical Character Recognition (OCR), SSD Mobilenet, TensorFlow Object Detection API, Voice Alerts, Text-to-Speech, Visually impaired people**

I. INTRODUCTION

Amidst the current fast-paced development of Information Technology (IT), extensive research efforts have been dedicated to resolving various commonplace challenges, leading to numerous conveniences for people. However, individuals with visual impairments encounter significant difficulties that impede their daily routines.

The visually impaired encounter significant challenges, particularly with regards to identifying objects and navigating indoors every day. Currently, the majority of visually impaired people in India rely on a stick to detect potential roadblocks. Previous studies used ultrasonic sensors to analyze objects. These approaches, however, make it difficult to determine the precise location of an object, especially in the presence of obstacles. By developing a solution to assist the visually impaired, we hope to achieve our goal.

We aim to provide a solution to assist the visually

Paper Submission Date : January 20, 2023 ; Paper sent back for Revision : February 10, 2023 ; Paper Acceptance Date : February 18, 2023 ; Paper Published Online : April 5, 2023

[1] H. Parekh, *Student* ; Email : hitanshuparekh72@gmail.com ; ORCID iD : https://orcid.org/0009-0009-7441-6168
[2] N. Agarwal *(Corresponding Author), Student* ; Email : niyatiagarwal061101@gmail.com ;
ORCID iD : https://orcid.org/0009-0003-3150-835X
[3] P. Bangera, *Student* ; Email : ranavbangera18@gmail.com ; ORCID iD : https://orcid.org/0009-0002-0613-626X
[4] R. D'souza, *Student* ; Email : rogerds31@gmail.com ; ORCID iD : https://orcid.org/0009-0005-8069-1773
[5] G. Tuscano, *Assistant Professor* ; Email : grinaltuscano@sfit.ac.in ; ORCID iD : https://orcid.org/0009-0004-5529-4592

[1,2,3,4,5] Department of Information Technology, St. Francis Institute of Technology, Sardar Vallabhbhai Patel Road, Mount Poinsur, Borivali West, Mumbai, Maharashtra - 400 103.

DOI : https://doi.org/10.17010/ijcs/2023/v8/i2/172774

impaired by developing a blind assistance system that includes features such as real-time object recognition, voice feedback, proximity warnings, and text recognition from signs and documents. The system is integrated into a single app for a simple and better user-friendly experience.

II. LITERAURE REVIEW

Balakrishnan and Sainarayanan [1] presented a system for processing images for visually impaired individuals that incorporates wearable stereo cameras and stereo headphones affixed to helmets. The system utilizes stereo cameras to capture a frontal view of the visually impaired person and extracts relevant information for scene characterization and navigation. The distance of objects is measured using the stereo cameras. However, the use of stereo cameras has certain disadvantages such as increased complexity and cost in comparison to single-camera systems, a limited depth sensing range due to triangulation, susceptibility to occlusions and environmental factors, and a need for precise calibration that can be affected by variations in lighting conditions.

Mahmud et al. [2] proposed a system that combines an ultrasonic sensor mounted on a cane and a Camshaft Position Sensor (CMP) Compass Sensor (511) to provide blind individuals with information about obstacles and potholes. However, this method has several drawbacks. The cane based system is limited to detecting obstacles upto knee height, and the CMP compass sensor can be affected by the presence of iron objects in the vicinity. Although useful for outdoor navigation, this system has limitations when used indoors. Additionally, the system only detects obstacles without attempting to recognize them, which can be a significant disadvantage.

Jiang, et al. [3] described a real-time multi-module visual recognition system with the output transformed into 3D audio. A GoPro-style portable camera records video, and the server processes the gathered stream for real-time picture recognition using an object detection module. Because of the processing latency, this technology is difficult to employ in real time.

Dai et al. [4] describe a complete convolutional network based on region. R-FCNN is utilized precisely and efficiently to recognize objects. As a result, to recognize the object, this study may simply employ ResNets as fully convolutional image classifier backbones. This paper describes a simple yet effective RFCNN architecture for detection of objects. This method achieves the same accuracy as the quicker R-FCNN method. As a result, implementing state-of-the-art image classification framework became easier.

Choi and Kim [5] present the current techniques for object detection models and standard datasets. The paper covers different types of detectors, including one-stage and two-stage, and examines various object detection methods, both established and new. It also highlights various branches of object detection.

Toro et al. [6] discussed the strategies for creating a vision-based wearable technology that will help visually impaired people navigate a new indoor environment. The proposed approach helps the user with "purposeful navigation." The system recognizes obstacles, walking zones, and other items such as computers, doors, stairwells, and seats.

Khairnar et al. [7] proposed a solution that combines smart gloves with a smartphone app. The smart glove detects and avoids obstacles while also allowing the visually disabled to comprehend one's surroundings. Using mobile-based obstacle and object identification, various items and barriers in the surrounding area are detected.

Kunta et al. [8] described a system that connects the environment and the blind via the Internet of Things (IoT). Sensors are used to detect impediments, moist flooring, and staircases, among other things. The device discussed here is a basic and low-cost smart blind stick. It is equipped with several IoT modules and sensors.

Karmarkar and Hommane [9] proposed a blind item recognition system using deep learning methodology. Voice assistance can also assist visually impaired persons in locating objects. The "You Only Look Once" YOLO method is incorporated into the deep learning model for object recognition. A voice alert is generated using text-to-speech (TTS) to aid the blind in gathering information about products. As a result, the object-detection technology effectively aids the visually handicapped in discovering items in a certain context.

Rajesh et al. [10] used the Tesseract OCR system to extract text from scanned photos. The e-Speak tool for aiding visually impaired persons converts this text data to voice. It helps the visually challenged to identify products by continuously extracting textual data from images and performing voice conversion on the data. Raspberry Pi is utilized for this purpose since it has a good battery backup and is portable.

TABLE I.

MODELS ALONG WITH THEIR INFERENCE TIME AND MAP PERFORMANCE

Model	Inference (ms)	mAP
Tiny Yolo V2	VOC 2007+2012	87.57
SSD MobileNet VI	COCO trainval	91.16
SSD MobileNet V2	COCO trainval	91.90
SSD Inception V2	COCO trainval	96.82
Faster RNN Inception V2	COCO trainval	96.69

The performance and inference time of various TensorFlow mobile object detection models are compared in Table I. The models exhibit a significant variation in mAP performance, with the highly complex Faster RCNN Inception model achieving nearly optimal performance, but taking considerably longer to infer compared to the less complex tiny Yolo model.

Current solutions suffer from several limitations, such as limited functionality and scope, high cost, lack of portability, varying sensor requirements, and the inability to assist visually impaired individuals in real-time both indoors and outdoors. We have worked hard to incorporate all of the outstanding characteristics of the prior systems in order to develop a complete, portable, and cost-effective system capable of handling real-world challenges.

III. WORKING

Our proposed system design (depicted in Fig. 2) revolves around identifying objects present in the surroundings of a person with visual impairment. From frame extraction through output recognition, the proposed object detection technique requires several stages. The query frame and database objects are compared to find objects in each frame. Our system presents a real-time solution for recognizing and locating objects, which triggers an audio file to provide information about the recognized object. This allows for simultaneous detection and identification of objects.

To use the camera to capture real-time images, the user needs to activate the application beforehand. The captured image is then processed by the model, which calculates the object's distance and transforms the output information into an audible signal.

The system employs a Residual Network (ResNet) architecture for feature extraction. ResNet is a type of artificial neural network that enables the model to bypass layers without impacting its performance [11].

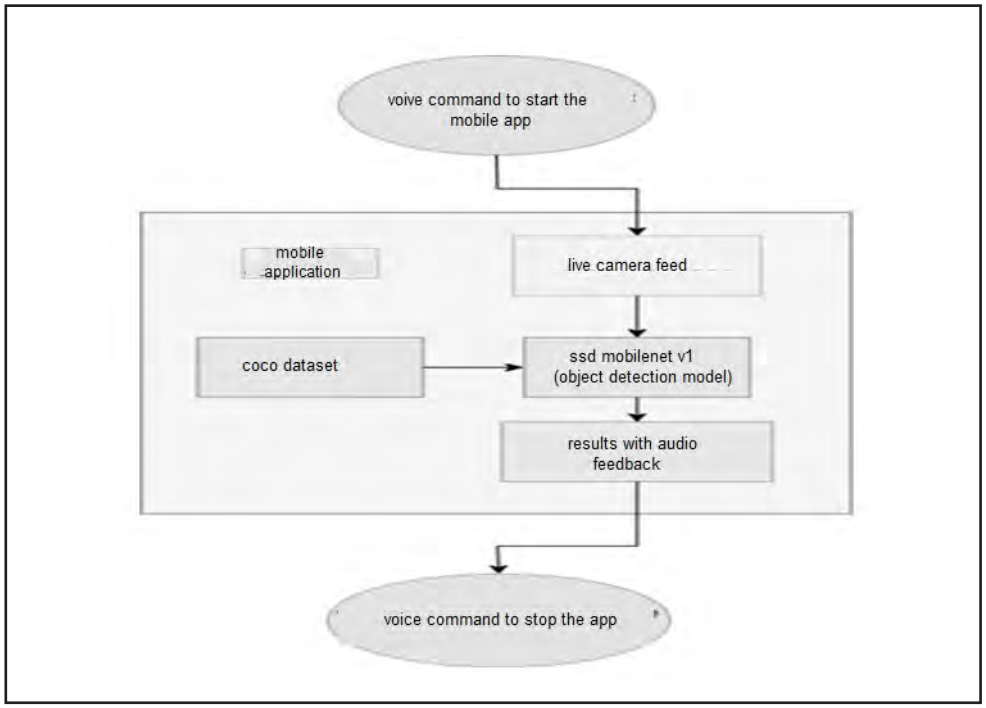

Fig. 1. System Flowchart

A. Methodology

This system consists mainly of :

1) The operational setup of the system involves a mobile application that captures real-time frames and transmits them to a networked server for all computational processes to be carried out.

2) The networked server will utilize a pre-trained detection model called SSD, which was trained on COCO datasets to identify objects. An accuracy metric will be applied to test and classify the identified objects.

3) Upon completion of the speech module testing, the object's category will be transformed into default voice notes and delivered to visually impaired individuals as assistance.

4) In addition to object detection, our system includes an alert mechanism that estimates the distance to detected objects. If a visually impaired person gets too close to an object, the system issues warnings.

5) For recognizing documents and signs, we have integrated a real-time OCR Text-To-Speech technology in this system. This technology performs text detection, recognition, and conversion of text into speech in real time.

B. COCO Dataset

The Common Objects in Context (COCO) dataset is an open-source image database used to train deep learning algorithms in object recognition. It includes hundreds of thousands of images with millions of pre-labeled objects for training purposes [12].

C. TensorFlow Object Detection API

To put it into action, we used the TensorFlow Object Detection API. This is a framework based on TensorFlow, designed to develop models for object detection. It includes a set of detection models that have been pre-trained on the COCO dataset as well as tools for training custom models on your data.

The TensorFlow Object Detection API is compatible with several object detection models, including faster R-CNN, Single-Shot Detection (SSD), and YOLO (You Only Look Once). It also supports several backend technologies, including TensorFlow Lite for mobile devices.

The TensorFlow Object Detection API simplifies the development, training, and implementation of object detection models, making it a potent resource for computer vision tasks.

D. Object Detection

The term "object detection" pertains to recognizing and localizing objects in an image that fall under a predetermined set of categories. Thanks to the abundance of data, high-speed GPUs, and efficient algorithms, computers can now be trained with great precision and accuracy to identify, classify, and detect different types of objects in a picture.

Typically, object detection falls into one of two categories:

1) *Two-stage detector* : The first step of a two-step detection process involves using regional design networks to identify areas of interest that have a high likelihood of being an object. Object detection, which completes the final classification and regression of the bounding box of the objects is the next phase. There are several two-stage detectors, including RCNN, Fast RCNN, and Faster RCNN. Although they analyze information more slowly than one-stage detectors, they are more accurate.

2) *One-stage detector* : Object detection in this scenario can be viewed as a simple regression task where the model learns the probabilities of different classes and the coordinates of bounding boxes from the input. One phase detector includes YOLO, YOLO v2, SSD, RetinaNet, and other technologies. In object detection, which is an advanced type of image classification, a neural network predicts and outlines items in an image using bounding boxes. Although they are renowned for their quick detection times and real-time processing, two-stage detectors are more accurate.

Currently, when comparing one-stage detectors, the Single Shot MultiBox Detector (SSD) is considered superior to You Only Look Once (YOLO) due to its high accuracy in object detection, small object detection, complex scene recognition, and demanding applications such as autonomous vehicles. SSD employs multiple overlapping boxes and a multi-scale feature extraction

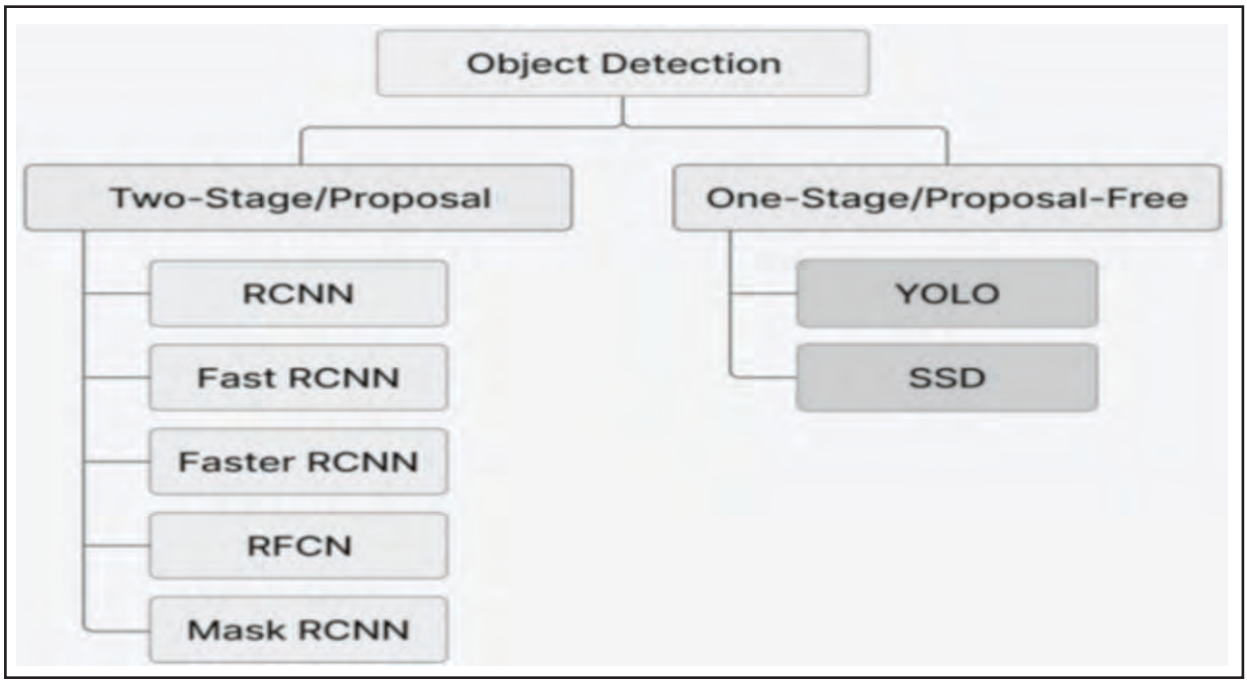

Fig. 2. Object Detection Models

network to forecast objects resulting in greater accuracy than YOLO.

E. SSD Architecture

The SSD consists of a backbone model and an SSD head. The backbone model is a pre-trained image classification network, typically trained on ImageNet, with its last fully

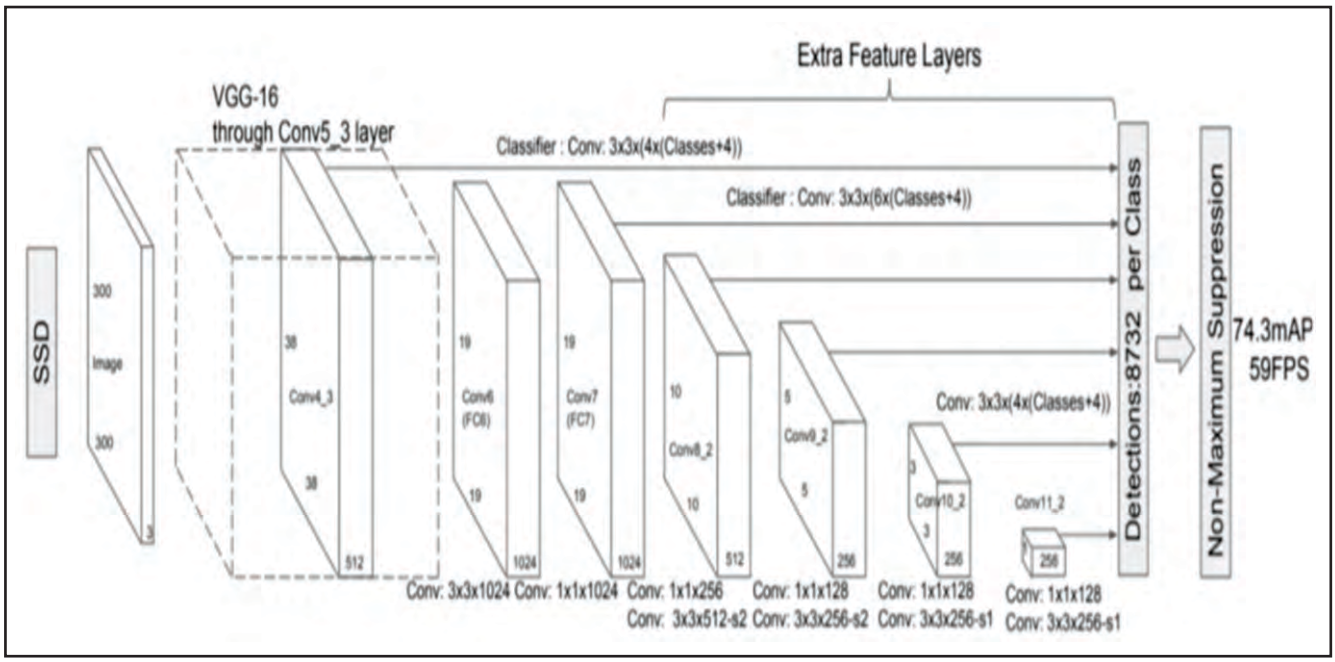

Fig. 3. SSD Model Architecture [13]

connected classification layer removed, such as ResNet. Its purpose is to extract features from an input image. The SSD head is one or more convolutional layers added to the backbone, and its outputs are interpreted as the bounding boxes and classifications of objects in the final layer activations' spatial positions. Therefore, we now have a deep neural network capable of extracting semantic meaning from an input image while preserving its spatial structure, even with low-quality images. In ResNet 34, the backbone generates 256 7x7 feature mappings for an input image. The SSD divides the image into grid cells, and each grid cell is responsible for detecting objects in that specific image area. Object detection requires predicting the type and location of an object in a particular area.

F. Anchor Box

In the Single Shot Detector (SSD) system, numerous predetermined anchor boxes are attached to each grid cell. These boxes are responsible for controlling the size and shape of the grid cells. To ensure a match between anchor boxes and ground truth object bounding boxes during training, the system employs a matching phase. The anchor box with the greatest overlap with the object predicts its class and location. This feature is utilized in training the network and predicting object locations in observed images after training. Each anchor box is assigned a zoom level and an aspect ratio, as not all objects have a square shape. The SSD structure includes predefined aspect ratios for the anchor boxes to address this issue. The ratios parameter for the anchor boxes connected to each grid cell at each zoom or scale level can be adjusted to vary the aspect ratios.

G. Zoom Level

The sizes of the anchor boxes in an SSD do not need to be identical to those of the grid cells. It is possible for the user to seek out objects of varying sizes within a single grid cell. The zoom feature controls how much the anchor boxes are resized relative to the grid cells.

H. MobileNet

MobileNet is a deep neural network architecture designed for real-time object detection and classification, primarily aimed at mobile and embedded devices. Developed by Google, it is optimized for efficiency, utilizing depth-wise separable convolutions that reduce the computational requirements. Depth-wise Separable Convolution is a type of convolution operation utilized in deep learning that divides the traditional convolution operation into two distinct operations: depth-wise convolution and point-wise convolution. The depth-wise convolution operates independently on each channel of the input feature map, while the point-wise convolution combines the output of the depth-wise convolution by performing a 1x1 convolution. The division of operations reduces the computational complexity, making it more efficient and ideal for deployment on resource-constrained devices in real-time.

I. Depth Estimation

The feature of depth estimation or extraction involves using algorithms and techniques to represent the spatial structure of a scene and calculate the distance between objects. In our prototype designed to aid the blind, it is crucial to determine the distance between the individual and obstacles in real-time to provide alerts. Once an object is detected, a rectangular box is formed around it.

Assuming that the object occupies a significant portion of the frame, the system computes an estimate of the object's distance from the person with some limitations. The following code (Fig. 4) used to recognize objects also retrieves data on their location and distance.

```
(boxes, scores, classes, num_detections) = sess.run([boxes,
scores, classes, num_detections],feed_dict={image_tensor:
image_np_expanded})
```

Fig. 4. Code Execution-1

In this context, a Tensorflow session has been

```
for i,b in enumerate(boxes[0]):
boxes[0][i][0] – y axis upper start coordinates
boxes[0][i][1] – x axis left start coordinates
boxes[0][i][2] – y axis down start coordinates
boxes[0][i][3] – x axis right start coordinates
```

Fig. 5. Code Execution-2

established that includes essential detection functions. The boxes, which are an array within an array, are then analyzed by iterating through them and defining the following conditions.

The index of the box-in-box array is denoted as "i" and is used to analyze the score and class of the box. Moreover, the detected object's width is determined by calculating the width of the object in pixels.

```
apx_distance = round((1 - (boxes[0][i][3]boxes[0][i][1]))**4),1)
```

Fig. 6. Code Execution-3

The center point of the identified rectangle is determined by subtracting the start coordinates of the same axis and dividing by two. A dot is then drawn in the center. The default parameter for drawing boxes is a score of 0.5. If scores[0][i] > = 0.5 (i.e. equal to or greater than 50%), then the object is assumed to have been detected.

```
            mid_x = (boxes[0][i][1]+boxes[0][i][3])/2
            mid_y = (boxes[0][i][0]+boxes[0][i][2])/2
            apx_distance = round((1 - (boxes[0][i][3]
boxes[0][i][1]))**4),1)
```

Fig. 7. Code Execution-4

The formula mentioned in Fig. 7 computes the midpoints of the x and y axes as mid_x and mid_y, respectively. By checking if apx_distance < 0.5 and mid_x is between 0.3 and 0.7, it can be inferred that the distance between the person and the object is too short. This code can be applied to determine the relative distance between the person and the object. Once an object is detected, the system employs the code to compute the object's distance relative to the individual. In the event that the object is too near, the speech generation module provides a warning or signal to the person.

J. Real-time OCR Text-To-Speech

Real-time OCR text-to-speech is a technical solution that utilizes OCR and TTS technologies to instantly identify and transcribe text. It allows for the automatic conversion of scanned or digital text into speech, allowing for immediate audio output and potentially enabling hands-free access to written information for visually impaired or otherwise disabled users. This technology typically involves capturing video frames, using OCR to extract text from the images, and then using TTS technology to convert the text into speech in real time.

K. Text detection and recognition procedures implemented through OpenCV, Python, and Tesseract

OpenCV, which stands for Open Source Computer Vision, is a freely available library designed for use in computer vision, image processing, and Machine Learning applications.

OCR or Optical Character Recognition is a process of converting scanned images of text, handwriting, or symbols into text that computers can read. To achieve accurate OCR results, multiple sub-processes are usually involved. The subsequent steps are:

- Preprocessing of the image
- Text localization
- Character segmentation
- Character recognition
- Post Processing

Naturally, the sub-processes from this list may vary, but they are the general stages that are required to get close to automatic character recognition.

Tesseract is a freely available Optical Character Recognition (OCR) engine that is open-source and was developed by Google. It is used to recognize text in images and convert it into machine-readable text. Tesseract uses Machine Learning algorithms to recognize text in different languages and scripts, including handwritten text. It is widely used for text recognition in various applications, such as document scanning, mobile applications, and automatic data entry systems.

To detect text, we employ OpenCV's EAST deep learning model. The OpenCV EAST text detector is a highly precise deep-learning model for detecting text in natural scene images. By employing this model, we can accurately detect and determine the bounding box coordinates of text present in an image. Then, for each of these text-containing sections, we used OpenCV and Tesseract to recognize and OCR the text.

Process of text recognition using OpenCV OCR and Tesseract is shown in Fig. 8.

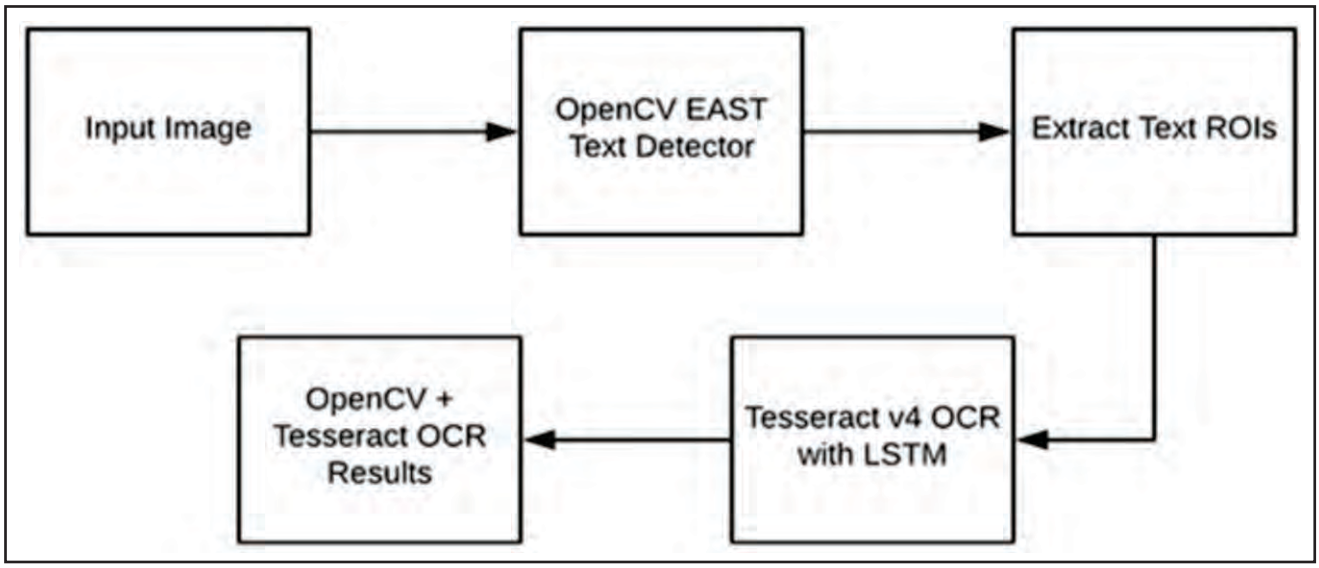

Fig. 8. The OpenCV OCR pipeline

1) To initiate the text recognition process, we utilize OpenCV's EAST text detector to scan an image for text. The detector returns the coordinates of the region of interest (ROIs) that encompass the text in the form of a bounding box (x, y).

2) Each of these ROIs will be isolated and forwarded to Tesseract v4's LSTM deep learning text recognition algorithm for text recognition.

3) Our actual OCR results will be provided via the LSTM's output.

4) The OpenCV OCR results will then be drawn on our output image.

L. Voice Generation

After detecting an object, it is crucial to notify the individual of its presence on his route. Pyttsx3 is a Python-based library that enables the conversion of text into speech through text-to-speech conversion. It uses the synthesis engine of the Microsoft Speech API (SAPI) to produce spoken output from the input text, which can be used in a variety of applications, including assisting the visually impaired. The library is highly versatile and can be easily installed and integrated with other software, making it a convenient option for real-time text-to-speech conversion. Unlike other libraries, it can function offline and is compatible with both Python 2 and 3. The output generated is in the form of audio commands. In situations where the detected object is too close, the warning "Warning: The object (class of object) is too close to you, be careful!" is conveyed.

IV. OUTPUT

Fig. 9. Object detection of a person and bottle

Fig. 10. Object detection of a plant

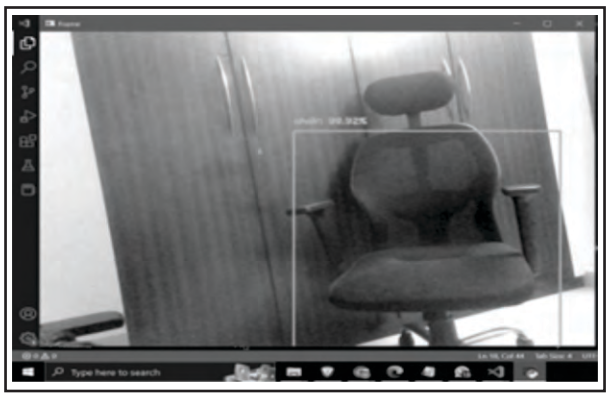

Fig. 11. Object detection of a chair

Fig. 12. Object detection of a television(TV)

V. CONCLUSION

The proposed system has demonstrated its ability to accurately identify and label 90 objects. The system is designed to provide assistance to visually impaired individuals by calculating the distance between the object and the camera and providing auditory feedback as they approach the object. The performance of the system was evaluated using the SSD Mobilenet V1 model, resulting in improved object detection speed and reduced latency. Additionally, the system is capable of recognizing text from different sources, such as documents, signs, and natural scenes by utilizing OpenCV, Python, and Tesseract. To convert the text to speech, the system incorporates the Python Pyttsx3 library. Overall, the system has the potential to significantly enhance the independence and quality of life for visually impaired individuals by assisting them in their daily activities.

VI. FUTURE SCOPE

Future uses of the technology might include enhancing the accuracy of the present system, language translation, currency verification, a chatbot for communication and engagement, smart shopping, email reading, real-time location sharing etc.

AUTHORS' CONTRIBUTION

Niyati Agarwal contributed to the development of the user interface and user experience design of the Object Detection System. She ensured that the system was intuitive and easy to use for visually impaired users. Pranav Bangera focused on the cyber security aspect of the system, ensuring that it was protected from potential cyber attacks and that users' data was secure. He also worked on integrating text-to-speech technology into the system. Hitanshu Parekh contributed to the development of the Artificial Intelligence and Machine Learning algorithms that power the object detection system. He worked on training the system to recognize various objects and to provide accurate descriptions to visually impaired users. Roger D'souza played a key role in testing and debugging the system to ensure its reliability and functionality. He also worked on integrating voice commands into the system to make it even more accessible to visually impaired users. Grinal Tuscano provided resources and assistance with finding and citing sources. She helped the students in preparing their presentations and defending their work during the final examination, encouraging and motivating them to stay focused and on track as they completed their final projects. She provided insight and expertise in the relevant field of study and helped the students in achieving their academic goals by acting as a mentor and role model for them offering advice on career paths and networking opportunities.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in the manuscript.

FUNDING ACKNOWLEDGEMENT

The authors have not received any financial support for the research, authorship, and/or for the publication of the article.

REFERENCES

[1] G. Balakrishnan and G. Sainarayanan, “Stereo Image Processing Procedure for Vision Rehabilitation,” *Appl. Artif. Intell.,* vol. 22, no.6, pp. 501–522, Jul. 2008, doi: 10.1080/08839510802226777.

[2] N. Mahmud, R. K. Saha, R. B. Zafar, M. B. H. Bhuian, and S. S. Sarwar, "Vibration and voice operated navigation system for visually impaired person," in *2014 Int. Conf. Inform., Electronics Vision,* 2014, pp. 1–5, doi: 10.1109/ICIEV.2014.6850740.

[3] R. Jiang, L, Qian, anad Q. Shuhui,, “Let blind people see: Real-time visual recognition with results converted to 3D audio,” pp. 1–7, 2016. [Online]: Available: http://cs231n.stanford.edu/reports/2016/pdfs/218_Report.pdf

[4] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region based fully convolutional networks,” in *Conf. Neural Inform. Process. Syst.*, Barcelona, Spain, Dec. 4–6, 2016, pp. 379–387, doi: 10.48550/arXiv.1605.06409.

[5] D. Choi and M. Kim, “Trends on object detection techniques based on Deep Learning,” *Electronics Telecommun. Trends*, vol. 33, no. 4, pp. 23–32, 2018, doi: 10.22648/ETRI.2018.J.330403.

[6] A. A. D. Toro, S. E. Campaña Bastidas, and E. F. Caicedo Bravo,"Methodology to build a wearable system for assisting blind people in purposeful navigation," in *2020 3rd Int. Conf. Inf. Comp. Tech.*, San Jose, CA, USA, 2020, pp. 205–212, doi: 10.1109/ICICT50521.2020.00039.

[7] D. P. Khairnar, R. B. Karad, A. Kapse, G. Kale, and P. Jadhav, "PARTHA: A visually impaired assistance system," in *2020 3rd Int. Conf. Communication Syst.*, *Comput. IT Appl.*, 2020, pp. 32–37, doi: 10.1109/CSCITA47329.2020.9137791.

[8] V. Kunta, C. Tuniki, and U. Sairam, "Multi-functional blind stick for visually impaired people," in *2020 5th Int. Conf. Commun. Electronics Sys.*, Coimbatore, India, 2020, pp. 895–899, doi: 10.1109/ICCES48766.2020.9137870.

[9] R. R. Karmarkar and V. N. Hommane, “Object detection system for the blind with voice guidance,” *Int. J. Eng. Appl. Sciences Technol.*, vol. 6, no. 2, 2021, pp. 67–70. [Online]. Available: https://ijeast.com/papers/67-70,Tesma602,IJEAST.pdf

[10] M. Rajesh, K. R. Bindhu, K. A. Roy, A. Thomas, A. Thomas, T. B. Tharakan, and C. Dinesh, "Text recognition and face detection aid for visually impaired person using Raspberry PI," in 2017 *Int. Conf. Circuit, Power Comput. Technol.*, pp. 1–5. IEEE, 2017, doi: 10.1109/ICCPCT.2017.8074355.

[11] Z. Feng, “ResNet architecture and its variants: An overview.” Builtin.com. [Online]. Available: https://builtin.com/artificial-intelligence/resnet-architecture

[12] “What is the COCO Dataset? What you need to know in 2023.” viso.ai. [Online]. Available: https://viso.ai/computer-vision/coco-dataset/#:~:text=The%20large%20dataset%20comprises%20annotated,popular%20technique%20in%20computer%20vision.

[13] SSD Model Architecture. Packtpub.com. [Online]. Available: https://subscription.packtpub.com/book/programming/9781838821654/11/ch11lvl1sec74/5-ssd-model-architecture

About the Authors

Hitanshu Parekh is an IT engineering student with a passion for Artificial Intelligence and Machine Learning. He has worked on several projects related to developing intelligent algorithms that can improve efficiency and accuracy in various applications. He is highly innovative and is constantly exploring new ways to integrate AI and ML into software development.

Niyati Agarwal is a final-year IT engineering student who has a keen interest in web development and user interface design. She has worked on several projects related to developing websites and apps that are both user-friendly and visually appealing. Her attention to detail and creativity make her a valuable asset to any software development team.

Pranav Bangera is an IT engineering student who specializes in cyber security and data protection. He has extensive knowledge of network security protocols and has worked on various projects related to securing sensitive data. He is highly analytical and is able to identify potential vulnerabilities and implement effective security measures.

Roger D'souza is an IT engineering student who excels in database management and software testing. He has experience in designing and implementing complex database structures and ensuring the accuracy and functionality of software applications. He is highly organized and has a strong attention to detail, making him an essential member of any software development team.

Grinal Tuscano is a seasoned guidance mentor with over 10 years of experience in helping students achieve their academic and personal goals. She has a passion for empowering students to reach their full potential and has a proven track record of success in mentoring and coaching. She has a deep understanding of the academic and emotional challenges faced by students and is dedicated to providing personalized support and guidance to help them overcome these challenges and succeed.