7

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI, KARNATAKA

mini-project Report on

“BLIND ASSISTANCE SYSTEM””

Submitted in partial fulfilment for the award of degree of

Bachelor of Engineering

In

INFORMATION SCIENCE AND ENGINEERING

By

MONISHA KUMAR(4NN21IS057)
AMNA MALAIKA MACCI (4NN21IS059)
ADARSH M R(4NN22IS400)
PRANAVKRISHNA S N(4NN22IS408)

Under the Guidance of

Dr. Prassana Kumar G

Asst. Professor, Dept. of Information Science and Engineering

NIEIT, Mysuru-18

ESTD-2008

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

NIE INSTITUTE OF TECHNOLOGY

MYSURU-570018 2023-2024

NIE INSTITUTE OF TECHNOLOGY

#50(part), Hootagalli Industrial area, KoorgalliVillage Mysuru-18

ESTD-2008

DEPARTMENT OF
INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the mini-project work entitled "**BLIND ASSISTANCE SYSTEM"** carried out by**,** **Miss. MONISHA KUMAR USN: 4NN21IS057**, **Miss. AMNA MALAIKA MACCI USN:4NN21IS059, Mr. ADARSH MR USN:4NN22IS400, Mr. PRANAVKRISHNA S N, USN:4NN22IS408** is a bonafide student of 6th Sem in partial fulfilment for the award of **Bachelor of Engineering in** of the Visvesvaraya Technological University, Belagavi during the year 2023-2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini-project report has been approved as it satisfies the academic requirements in respect of work prescribed as a part of the B.E degree.

Signature of the Guide	Signature of the Co-Ordinator	Signature of the HOD
DR. PRASANNA KUMAR G	**MR. SUDEEP J**	**DR. NANDINI M S**
Assistant Professor	Assistant Professor	Associate Professor& HOD
NIE-IT, Mysuru	NIE-IT, Mysuru	NIE-IT, Mysuru

External Viva

1……………………………………..   1……………………………………..

2……………………………………..   2……………………………………..

Acknowledgement

We Sincerely owe our gratitude to all people who helped and guided us in completing this miniproject work.

We are thankful to Dr. ROHINI NAGAPADMA, Principal, NIEIT, Mysore, for having supported us in academic endeavours.

We are thankful to Dr. NANDINI M.S, head of Department of Information Science and Engineering, NIEIT for providing us timely suggestion, encouragement and support to complete this mini-project.

We would like to sincerely thank our mini-project guide DR. PRASANNA KUMAR G for providing relevant information, valuable guidance and encouragement to complete this mini-project.

We would also like to thank all our teaching and non-teaching staff members of the Department. We are grateful to the college for keeping labs open whenever required and providing us Systems and required software.

We are always thankful to our parents for their valuable support and guidance in every step. We express our deepest gratitude and indebted thanks to NIEIT which has provided us an opportunity in fulfilling our most cherished desire of reaching our goal.

Your's Sincerely,

MONISHA KUMAR(4NN21IS057)
AMNA MALAIKA MACCI(4NN21IS059)
ADARSH M R(4NN22IS400)
PRANAVKRISHNA S N(4NN22IS408)

ABSTRACT

It’s a known fact that estimated number of visually impaired person in the **world** is about 285 million, approximately equal to the 20% of the Indian Population. They suffer regular and constant challenges in **Navigation** especially when they are on their own. They are mostly dependent on someone for even accessing their basic day-to-day needs. So, it’s a quite challenging task and the technological solution for them is of utmost importance and much needed.

One such try from our side is that we came up with an **Integrated Machine Learning System** which allows the Blind Victims to **identify** and classify **Real Time** Based Common day-to-day Objects and generate **voice feedbacks**. Calculates **distance** which produces warnings whether he/she is very close or far away from the object. The same system can be used for Obstacle Detection Mechanism.

Computer Vision The core of the Blind Assistance System is its computer vision capabilities, which allow it to interpret and understand the visual environment. By using a camera module, the system captures real-time video feed, which is then processed to identify objects, read text, and detect obstacles.

Machine Learning: Machine learning algorithms are employed to enhance the accuracy and reliability of object and text recognition. These algorithms are trained on diverse datasets to ensure robust performance across various scenarios and environments.

Sensor Integration: The system incorporates ultrasonic sensors to detect obstacles in the user's path. These sensors provide real-time feedback, enabling the system to alert the user to potential hazards and help them navigate safely.

Audio Feedback: One of the critical aspects of the Blind Assistance System is its audio feedback mechanism. The system uses text-to-speech (TTS) technology to convert recognized text and object information into audible speech, which is then communicated to the user through headphones or a speaker. Portability and Ergonomics: The device is designed to be lightweight and portable, ensuring that it can be easily carried and used by the visually impaired. The ergonomic design includes intuitive controls, allowing users to operate the device with minimal training.

CONTENTS

LIST OF FIGURES

CHAPTER 1

1.1 INTRODUCTION

Blindness is one of the most frequent and debilitating of the various disabilitiesThe proposed system is designed to aid visually impaired persons with real-time obstacle detection, avoidance, indoors and out navigation, and actual position tracking.

The gadget proposed is a camera-visual detection hybrid that performs well in low light as part of the recommended technique, this method is utilized to detect and avoid impediments, as well as to help blind persons in identifying the environment around them.

A simple and effective method for people with visual impairments to identify things in their environment and convert them into speech for improved comprehension and navigation.

Along with these, the depth estimation, which calculates the safe distance between the object and the person, allowing them to be more self-sufficient and less reliant on others. It was able to achieve this model with the help of Tensor Flow and pre-trained models. The approach we suggest is dependable, inexpensive, practical, and reliable.

The main purpose of this project is to build a model for the blind people to deduct the object in front of them and to navigate the way with the help of voice feedbacks.

The primary motivation behind the development of the Blind Assistance System is to empower visually impaired individuals by giving them the tools they need to navigate the world more safely and independently.

Traditional methods of assistance, such as guide dogs or canes, have their limitations and cannot always provide the comprehensive support required. By integrating advanced technologies like computer vision, machine learning, and sensor fusion, the Blind Assistance System aims to offer a more robust and versatile solution.

CHAPTER 2

2.1 LITERATURE SURVEY

A literature survey on blind assistance systems provides an overview of the research and developments in the field of assistive technologies for visually impaired individuals. This survey encompasses various approaches, technologies, and methodologies that have been explored to enhance the mobility, independence, and quality of life for the visually impaired.

Historical Context and Early Developments:
The quest to aid visually impaired individuals dates back centuries, with early methods including the use of canes and guide dogs. These traditional tools provided foundational support but were limited in scope and effectiveness. The advent of technology in the 20th century introduced electronic travel aids (ETAs), which used ultrasonic and infrared sensors to detect obstacles. These early ETAs, however, often faced challenges such as high cost, bulkiness, and limited functionality.

Modern Technologies and Approaches:
Recent advancements in technology have led to significant improvements in blind assistance systems. Key areas of focus include:

1. Computer Vision: Modern blind assistance systems extensively use computer vision to interpret and understand the visual environment. Techniques such as image processing, object detection, and scene understanding are leveraged to provide real-time information to the user. Research has shown that computer vision can effectively recognize objects, read text, and detect obstacles, significantly enhancing the user's situational awareness.

2. Machine Learning and Artificial Intelligence: Machine learning algorithms play a crucial role in improving the accuracy and reliability of blind assistance systems. Deep learning models, trained on vast datasets, have demonstrated remarkable success in recognizing objects and text in diverse environments. Studies highlight the potential of AI to continuously learn and adapt, further refining the system's performance over time.

3. Sensor Integration:The integration of various sensors, such as ultrasonic, infrared, and LIDAR, has been a focal point in the development of blind assistance systems. These sensors complement computer vision by providing additional spatial information, enabling the detection of obstacles and hazards that may not be visible in the camera's field of view. Research indicates that multi-sensor fusion can enhance the robustness and accuracy of obstacle detection.

4. Wearable Devices: Wearable technology has emerged as a promising avenue for blind assistance systems. Devices such as smart glasses, belts, and shoes equipped with cameras and sensors offer a hands-free and unobtrusive solution. Studies have explored the ergonomics and user acceptance of these devices, emphasizing the importance of comfort and ease of use.

5. Audio Feedback and Haptics:Effective communication of information to visually impaired users is critical. Audio feedback, particularly through text-to-speech (TTS) technology, has been extensively researched and implemented. Additionally, haptic feedback, which uses vibrations to convey information, has been explored as an alternative or complementary method. Literature suggests that combining audio and haptic feedback can enhance user experience and situational awareness.
technologies.

Project Name with software	Description	Drawback
Detection using Raspberry Pi	• This study advocates the creation of an intelligent system to assist blind people with daily tasks.. • Those who are profoundly deaf require constant assistance in all situations, particularly during day-to-day activities in many circumstances.	• Its slower processor speed hence lag time and working is very time consuming and slow . • Its better to use up rock64 media board being more powerfull.
Detection using Pi Cam	• Ultra-Sonic Sensor and Open CV Visually impaired people confront a wide range of issues in their everyday lives. • . The suggested invention intends to supply voice-controlled wearable visual help for visually impaired persons. It can, among other things, find goods and signboards.	• Complications of errors is being defined here . • Truncation problems are noticed .

Fig 2.1 LITREATURE SURVEY 1

Project Name with software	Description	Drawback
Detection using Arduino Uno	• Obstacle detection, obstacle avoidance, interior and outside routing, and true position sharing are among the capabilities planned for the VI (Visually Impaired) Incorporating Advanced. • A combo of smart gloves and phone software that performs well in low light is the suggested device . • Presented as a method for detecting and avoiding obstacles while also helping visually impaired individuals in recognizing their environment.	• With this detetction comes easy overriding firmware and lack of power protection. • Limitated in handling LED pins and photo sensors .
Detection using Ultrasonic Sensor and Buzzer	• Research was written by Ali Khan and Aftab Khan with the purpose of designing something specifically for the blind. This research culminated in the invention of an ultrasonic obstacle detecting system. • A sensor module, as well as a vibration device and a buzzer, are included in the wearable garment. Sensors check the user's surroundings and vibrate and emit a buzzer to alert them to any impediments.	• It comes with measurement inaccuracies. • Sensor and buzzer would misjudge the object..

Fig 2.2 LITREATURE SURVEY 2

CHAPTER 3

3.1 PROBLEM STATEMENT

This system aims to make visually impaired individuals more independent by enabling them to explore and interact with their environment more effectively. By providing real-time feedback about nearby objects or obstacles, the system enhances their ability to navigate and engage in daily activities autonomously.

3.2 OBJECTIVES

1. Image Capture: Using a webcam or camera to capture real-time images of the surroundings.
2. Networked Server: Sending these images to a central server (typically over a network connection), where the heavy computations such as image processing and object recognition are performed.
3. Object Recognition: The server processes these images to identify objects, people, or other relevant features in the environment.
4. Voice Feedback: After identifying objects, the system converts this information into voice feedback that is sent back to the user in real-time.

Making visually impaired people self-sufficient in their day-to-day activities by recognizing nearby things for self-contained exploration. Traditional method includes just detecting an item and sounding an alert to find it. The system is set up so that the webcam (if you use a laptop) takes real-time images and sends them to a computer Networked Server, which does all of the computations like identifying the image and giving real time voice feedbacks.

3.3 SCOPE OF THE PROPOSED WORK

1. **Problem Definition and Requirements Gathering**

- **Identify User Needs**: Conduct surveys and interviews with visually impaired individuals to understand their challenges and requirements.
- **Define Objectives**: Specify the primary goals of the system, such as obstacle detection, object recognition, navigation assistance, and user feedback.

.

2. System Design and Architecture

Hardware Selection:

Sensors: Choose appropriate sensors for environmental perception, such as cameras, LiDAR, ultrasonic sensors, and GPS.
Processing Units: Select processing units capable of handling computational tasks (e.g., Raspberry Pi, NVIDIA Jetson).
User Interface Devices: Determine output devices for user feedback, such as speakers, headphones, or haptic feedback units

Software Architecture:

Algorithm Development: Design algorithms for real-time image processing, object detection, and distance measurement.
Integration: Ensure seamless integration between hardware and software components.

3. Algorithm Development and Implementation

Computer Vision:

Object Detection: Implement models (e.g., YOLO, SSD, Faster R-CNN) for detecting and identifying objects in the user's path.
Scene Understanding: Develop methods for scene segmentation and context awareness.

Navigation and Guidance:
Path Planning: Design algorithms for route planning and obstacle avoidance.

The scope of work for a blind assistance system is extensive and involves multidisciplinary efforts in hardware design, software development, user experience, and accessibility. The ultimate goal is to create a reliable, efficient, and user-friendly system that significantly improves the quality of life for visually impaired individuals.

.

Fig 6. Image Recognition of person and cell phone

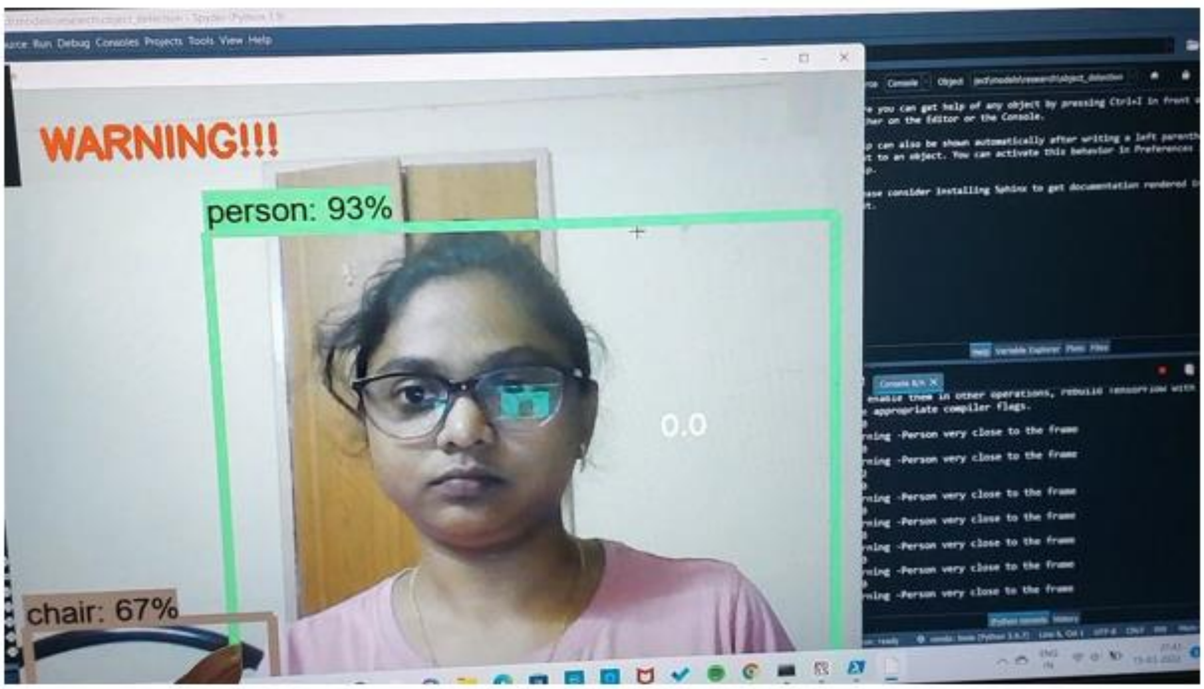


Fig 7. Image Recognition of person and chair

Fig 3.1 image recognition of person and chair

CHAPTER 4

4.1 FUNCTIONAL REQUIREMENTS

Navigation Assistance: Real-time guidance for indoor and outdoor navigation, possibly integrating GPS and indoor positioning systems.
Object Recognition: Identifying and describing objects, obstacles, and text in the environment.
Voice Commands: Enabling interaction through voice commands for hands-free operation.
Feedback Mechanism: Providing audio or tactile feedback to convey information and instructions.
Integration: Compatibility with existing technologies and platforms (e.g., smartphones, smartwatches).

Technical Specifications:
Hardware: Sensors, cameras, microphones, and wearables used in the system.
Software: Algorithms for object recognition, voice processing, and navigation.
Data Handling: Methods for processing and storing user data securely.

User Interface:
Accessibility: Ensuring ease of use for visually impaired users.
Customization: Allowing users to adjust settings according to their preferences.

Testing and Validation:
Usability Testing: Evaluating the system with real users to ensure effectiveness and user-friendliness.
Performance Metrics: Measuring accuracy, response time, and reliability.

Deployment and Support:
Implementation: Strategies for rolling out the system to users.
Training and Support: Providing resources for users to learn how to use the system and offer technical support.

Future Enhancements:
Scalability: Plans for future upgrades and expanding functionalities.
User Feedback: Incorporating feedback for continuous improvement.
This scope provides a comprehensive framework for developing a blind assistance system, addressing both technical and user-centric aspects.

4.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements for a blind assisstiance system system include:

Performance: The system should a produce vocal output with minimal latency.

Reliability: The system should have high availability and low failure rates, ensuring consistent performance.

Scalability: The system should handle an increasing number of users or more complex gestures without degrading performance.

Usability: The system should be easy to learn and use, providing a user-friendly interface and clear instructions.

Security: The system should protect user data from unauthorized access and ensure secure data transmission.

Maintainability: The system should be easy to update and maintain, with clear documentation and support for troubleshooting.

Portability: The system should be adaptable to various hardware and software environments, including different devices and operating systems.

Efficiency: The system should optimize resource usage, including CPU, memory, and battery power, to operate effectively on portable devices.

Compatibility: The system should integrate smoothly with other software and hardware, supporting standard protocols and interfaces.

Accessibility: The system should be designed to accommodate users with diverse abilities, including those with disabilities.

Localization: The system should support multiple languages and cultural contexts, allowing for use in various regions.

4.3 RISK ANALYSIS

Risk analysis for a blind assisstiance system involves identifying potential risks, assessing their impact and likelihood, and determining mitigation strategies. Here are the key risks associated with developing and deploying a blind assisstiance system :

4.3.1. Technical Risks

Object detection system: The system might fail to accurately recognize gestures, especially in complex or ambiguous scenarios.

Mitigation: Implement robust machine learning algorithms and continuously train the system with diverse gesture datasets.

Latency Issues: Delays in processing gestures and generating vocal output can frustrate users.

Mitigation: Optimize the processing algorithms and ensure efficient hardware utilization.

4.3.2. User Experience Risks

Usability: The system may be difficult for users to learn or use effectively.

Mitigation: Conduct thorough user testing and incorporate feedback to improve the user interface and experience.

Accessibility: The system may not be accessible to all users, particularly those with severe disabilities.

Mitigation: Design the system with inclusive practices and ensure it meets accessibility standards

4.4 SYSTEM REQUIREMENTS

4.4.1 HARDWARE REQUIREMENTS

PROCESSOR: Intel core processor (Core i5 processor)

RAM: 4 GB and above

HARD-DISC: 194 MB and above

4.4.2 SOFTWARE REQUIRMENTS

YOLO V8 VERSION

PYCHARM

COCO DATASETS

PYTHON

CHAPTER 5

5.1 PROPOSED METHODODLOGY

Creating a blind assistance system using the COCO (Common Objects in Context) dataset involves a detailed methodology designed to effectively utilize this dataset for training and evaluating object detection models. The primary objective is to develop a model capable of recognizing objects and contextual information in real-time, thereby assisting visually impaired users. The system focuses on detecting common objects relevant to daily navigation and interaction.

The methodology begins with the preparation of the COCO dataset, which includes over 80,000 images with more than 500,000 labeled objects across 80 categories. To tailor the dataset to the needs of blind assistance, the project selects object categories most relevant to users, such as people, vehicles, furniture, and other common objects. Data augmentation techniques, such as rotation, scaling, and flipping, are applied to enhance the model's robustness.

For model selection and architecture, the focus is on choosing an object detection model that balances accuracy and speed. Options include YOLO (You Only Look Once) for its real-time performance, SSD (Single Shot Multibox Detector) for mobile and embedded systems, and Faster R-CNN for high accuracy scenarios. The architecture is designed with computational constraints in mind, especially for mobile or edge device deployment, and may involve lightweight models or model compression techniques.

Training and optimization involve training the selected model on the COCO dataset using frameworks like TensorFlow or PyTorch. Transfer learning with pre-trained models expedites training and improves performance. Hyperparameter tuning, including optimizing learning rate, batch size, and number of epochs, is conducted to achieve optimal results. The methodology employs appropriate loss functions, such as cross-entropy for classification and smooth L1 loss for bounding box regression, along with optimizers like Adam or SGD.

Evaluation and testing use metrics like mean Average Precision (mAP), precision, recall, and F1-score to assess model performance on the COCO validation set. Real-world testing ensures robustness and reliability, with feedback from visually impaired users providing insights into usability and effectiveness. Integration into a blind assistance system involves implementing real-time object detection with low latency and high accuracy. The model is optimized for deployment on target hardware using tools like TensorRT or ONNX for accelerated inference.

User feedback mechanisms, such as audio descriptions or haptic feedback, are developed to provide intuitive information conveyance.
Continuous improvement is facilitated by collecting real-world usage data to refine and enhance the model over time. Ethical and privacy considerations include ensuring data privacy during real-world testing and designing the system to be accessible and inclusive, meeting relevant accessibility standards. This comprehensive methodology leverages the COCO dataset to develop a robust and effective object detection model for a blind assistance system, balancing performance and usability while focusing on real-world conditions and meaningful feedback for users.

The proposed methodology for developing a YOLOv8-based object detection system for visually impaired individuals is focused on creating a robust, real-time solution. The primary objective is to develop a system that can identify and describe nearby objects to assist users in everyday navigation. The project begins with dataset preparation, where the COCO dataset is employed, emphasizing objects pertinent to daily activities. To enhance the model's robustness and generalization, data augmentation techniques such as rotation, scaling, and flipping are applied.

YOLOv8 is selected for its exceptional speed and accuracy in real-time object detection. The training process leverages transfer learning, utilizing pre-trained weights to expedite the training process and improve accuracy. The model undergoes fine-tuning to concentrate on specific object classes relevant to the blind assistance context.

Hyperparameter tuning is also performed to optimize parameters such as learning rate and batch size, further enhancing the model's performance.
Evaluation and testing of the system are conducted using metrics such as mean Average Precision (mAP), precision, recall, and F1-score to assess performance comprehensively. Real-world testing in diverse environments is carried out, and user feedback is collected for further refinement.

For integration and deployment, the model is implemented for fast and efficient real-time inference on edge devices, ensuring accessibility and ease of use for visually impaired individuals. Audio or haptic feedback mechanisms are developed to convey detected object information to users effectively.

Finally, a continuous improvement strategy is adopted, incorporating a feedback loop that utilizes real-world data and user interactions to update and enhance the model consistently. This comprehensive approach aims to deliver an effective and user-friendly blind assistance system, prioritizing real-time performance, usability, and continuous enhancement.

5.2 BLOCK DIAGRAM

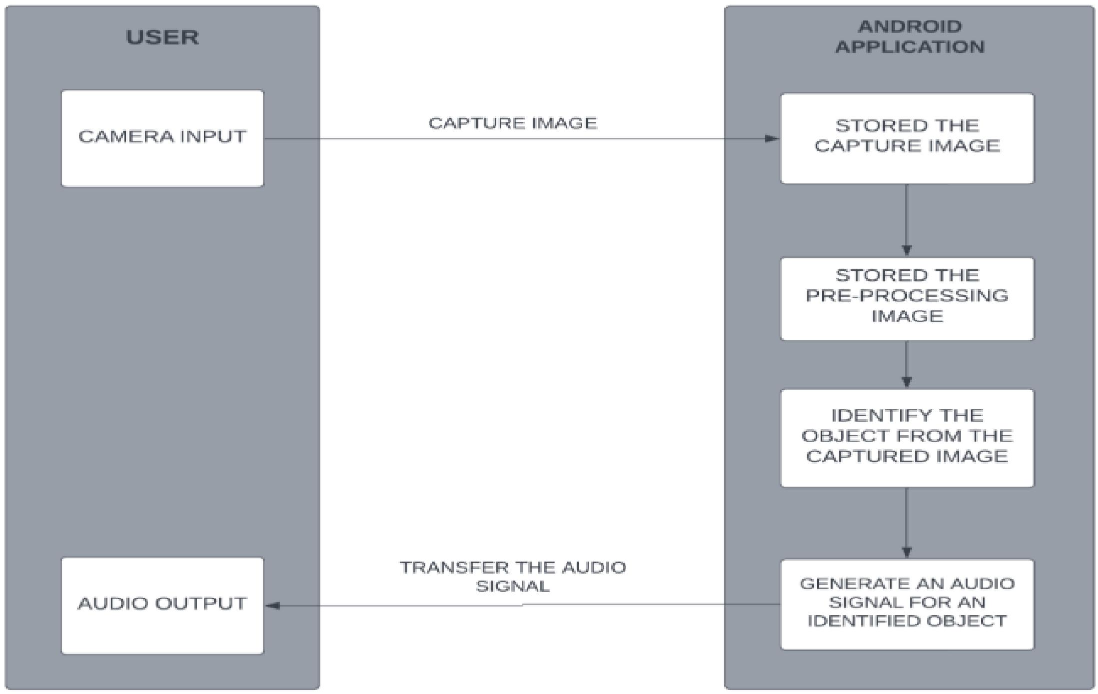


Fig 5.2 Blind assisstiance system block Diagram.

5.3 WORKING PRINCIPLE

The design aims to replace existing technologies of navigation systems based on detectors and buzzers with a simpler yet effective way of creating a backing system based on machine literacy, where we can describe an object while receiving real-time voice feedback and distance calculation with the required details. The proposed system is more efficient and reliable. The system is set up to record real-time frames and execute all calculations

refers to the camera input shown in this diagram will collect the image and transfer it to storage, where it will be stored and pre-processed before trying to find the image. Once the image has been found, it will generate a speech signal that the user can hear. Once the image has been acquired, voice feedback will be supplied as well.

The basic method used was object avoidance and object detection. It also includes outdoor location sharing which is quite a tedious task. So, we only have a buzzer to detect the object and alarms would be sent accordingly to the blind peoples.

Along with that, they can also get depth estimation features which will help them in their safe traversal. So once the object is found in the vicinity of the visually challenged person, with the help of the algorithm that we have developed that is Single Shot Detection (SSD)algorithm and the pre-trained datasets, the object found will be compared with the dataset and an output will be sent after detection.

Once the object gets identified, voice feedbacks are sent which is the name of the obstacle or object along with the distance calculation. It is also supported by warnings to notify if the person is at a safe distance or not. This kind of mechanism will help the blind person in navigating safely throughout making them more independent.

CHAPTER 6

6.1 SOFTWARE TOOL

Software tools are essential for building and developing the project. The following software tools were used:

Developing a blind assistance system using the COCO dataset with YOLOv8n and Python in PyCharm requires several specific software components. The core programming language is Python, with a recommended version of 3.6 or higher to ensure compatibility with the latest libraries and tools.

PyCharm, an integrated development environment (IDE) tailored for Python development, provides the necessary features for efficient coding, debugging, and project management.

The YOLOv8n (You Only Look Once, version 8 - Nano) model, known for its real-time object detection capabilities, is essential for recognizing and identifying objects in the user's environment. This model can be easily installed via the Ultralytics YOLO package.

Additionally, the COCO (Common Objects in Context) dataset, a large-scale dataset designed for object detection, segmentation, and captioning, serves as the primary training and testing dataset for the system. This dataset can be downloaded from the official COCO website.

Together, these software components create a robust framework for developing a real-time blind assistance system that can detect and identify objects, providing critical information to visually impaired users through auditory or haptic feedback.

6.2 PROGRAM

```
from ultralytics import YOLO
import cv2
import cvzone
import math
import pyttsx3
import pytesseract
from cvzone.FaceMeshModule import FaceMeshDetector
import numpy as np
import face_recognition
import os

# creating a list of known persons

path = '../YOLO/knownPersons'
images = []
classNamesKnown = []
myList = os.listdir(path)
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNamesKnown.append(os.path.splitext(cl)[0])

# mandatory/formality encoding for module to work

def findEncoding(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
```

```
        encodeList.append(encode)
    return encodeList


encodeListKnown = findEncoding(images)


# capturing frame in real time


cap = cv2.VideoCapture(1)
cap.set(3, 1280)
cap.set(4, 720)


# setting max numbers of face required to detect(more face leads to slow
processing)


detector = FaceMeshDetector(maxFaces=1)


# cap = cv2.VideoCapture("../videos/cars.mp4")


model = YOLO('../yolo-weights/yolov8n.pt')


# coco data set configration class


classNames = [
            "person", "bicycle", "car", "motorbike", "aeroplane", "bus", "train",
"truck", "boat", "traffic light",
            "fire hydrant", "stop sign", "parking meter", "bench", "bird", "cat",
"dog", "horse", "sheep", "cow",
            "elephant", "bear", "zebra", "giraffe", "backpack", "umbrella", "hand
bag", "tie", "suitcase", "frisbee",
            "skis", "snowboard", "surfboard", "sports ball", "kite", "baseball bat",
"baseball glove", "skateboard",
            "tennis racket", "bottle", "wine glass", "cup", "fork", "knife", "spoon",
"bowl", "banana", "apple",
            "sandwich", "orange", "broccoli", "carrot", "hot dog", "pizza", "donut",
```

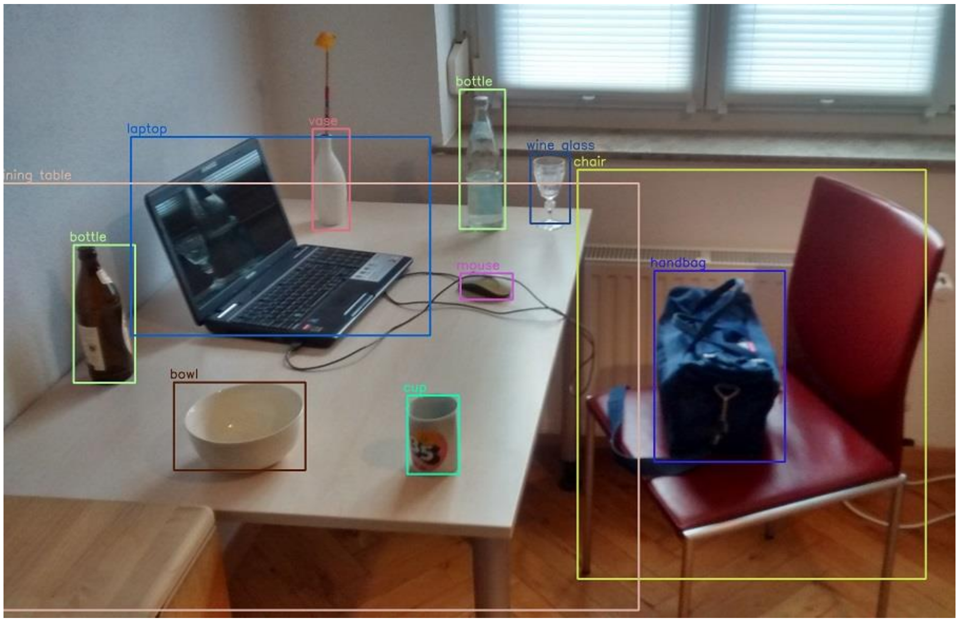
```
"cake", "chair", "sofa",
            "potted plant", "bed", "dining table", "toilet", "tv monitor", "laptop",
"mouse", "remote", "keyboard",
            "cellphone", "microwave", "oven", "toster", "sink", "refrigerator",
"book", "clock", "vase", "scissors",
            "teddy bear", "hair drier", "tooth brush"
            ]


while True:
    success, img = cap.read()
    img, faces = detector.findFaceMesh(img, draw=False)
    results = model(img, stream=True)

    # identification of known persons

    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)
    name = ''

    for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
        # print(faceDis)
        matchIndex = np.argmin(faceDis)

        if matches[matchIndex]:
            name = classNamesKnown[matchIndex].upper()
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
```

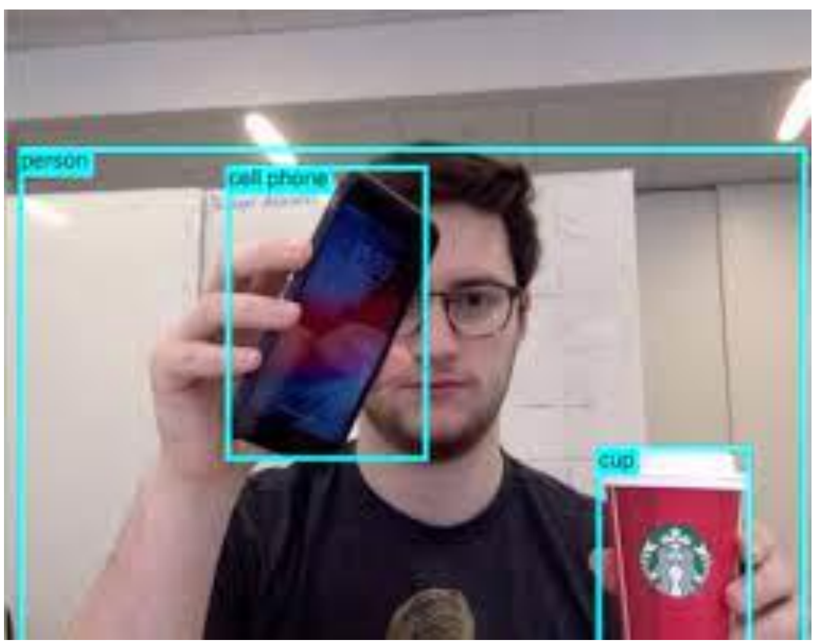
```
                cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
                cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_PLAIN,
        1, (255, 255, 255), 2)

    # putting box around detected object

    for r in results:
        boxes = r.boxes
        for box in boxes:
            x1, y1, x2, y2 = box.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)

            w, h = x2-x1, y2-y1
            cvzone.cornerRect(img, (x1, y1, w, h))

            conf = math.ceil((box.conf[0]*100))/100

            cls = int(box.cls[0])
            # displaying the detected object's text on image

            cvzone.putTextRect(img, f'{classNames[cls]} {conf}', (max(0, x1),
        max(35, y1)), scale=1, thickness=1)

            # finding the distance of the person
            if faces:
                face = faces[0]
                PointLeft = face[145]
                PointRight = face[374]
                # cv2.line(img, PointLeft, PointRight, (0, 200, 0), 2)
                # cv2.circle(img, PointLeft, 5, (255, 0, 255), cv2.FILLED)
                # cv2.circle(img, PointRight, 5, (255, 0, 255), cv2.FILLED)
                w, _ = detector.findDistance(PointLeft, PointRight)
                W = 6.3
```

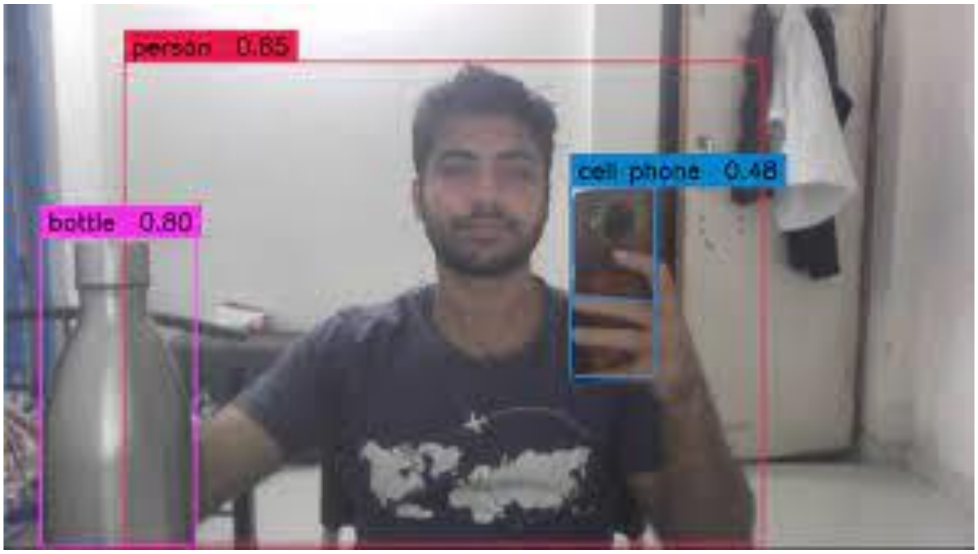
```
                f = 840
                d = ((W * f) / w)
                # print(d)
                cvzone.putTextRect(img, f'distance: {int(d)}centimeters', (face[10][0]
- 125, face[10][1] - 45), scale=2)
            else:
                d = "0"

            # voice feedback of all the stuff detected

            text = f'{classNames[cls]} {conf}, person distance: {int(d)}centimeters',
f'{name}'

            engine = pyttsx3.init()
            engine.setProperty('rate', 150)
            pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-
OCR\tesseract.exe'
            engine.say(text)
            engine.runAndWait()

    cv2.imshow("Image", img)
    cv2.waitKey(1)
```

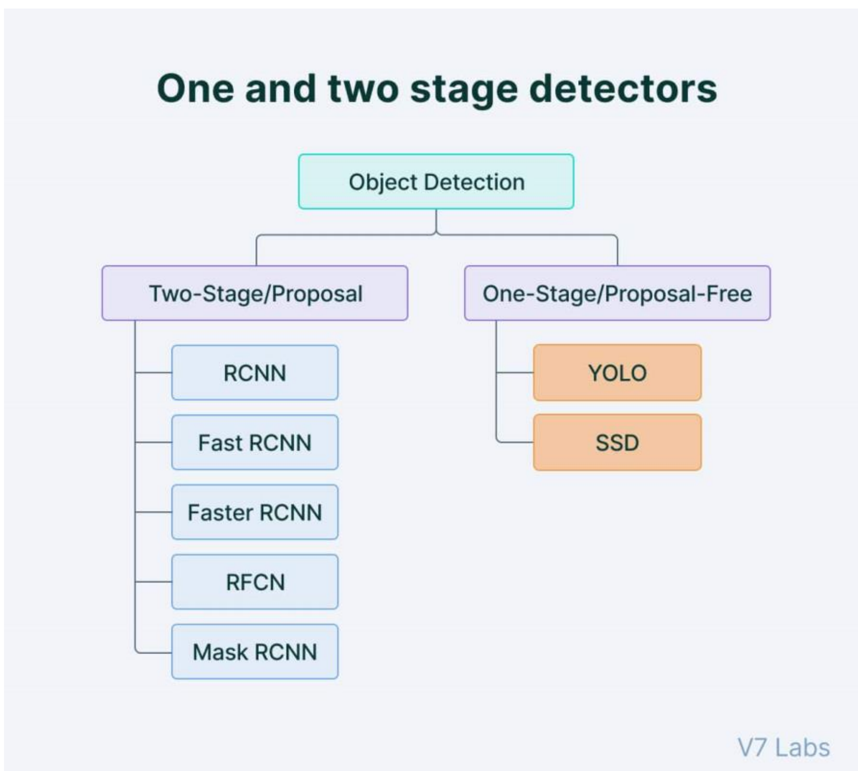
CHAPTER 7

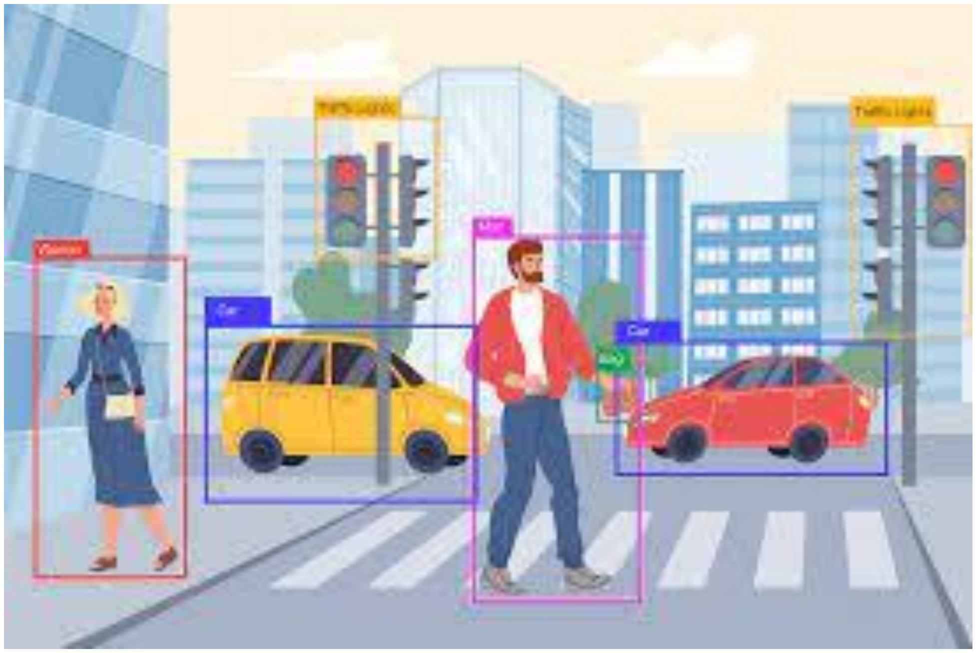
7.1 RESULTS WITH SNAPSHOTS

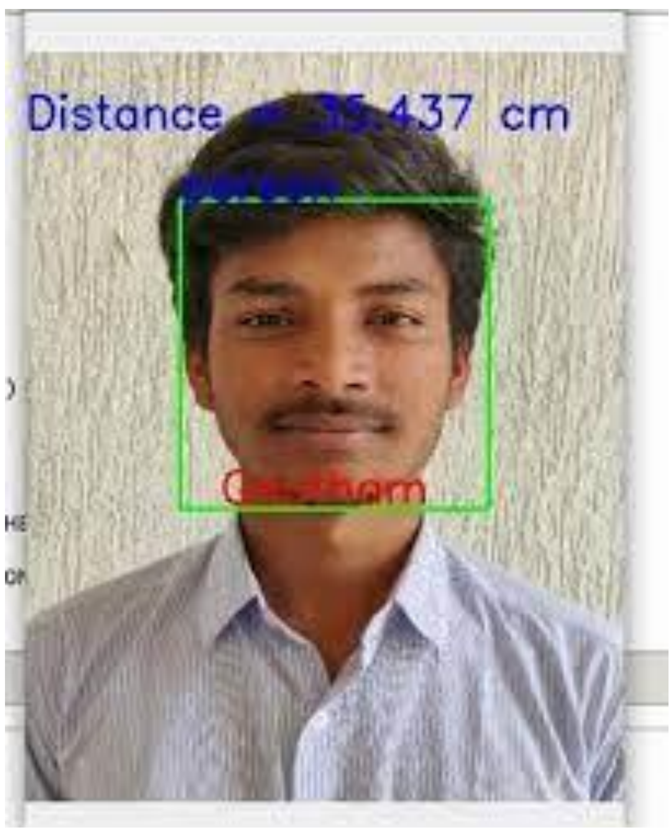





person 0.85
bottle 0.80
cell phone 0.48


One and two stage detectors
Object Detection
Two-Stage/Proposal
One-Stage/Proposal-Free
RCNN
Fast RCNN
Faster RCNN
RFCN
Mask RCNN
YOLO
SSD
V7 Labs




Distance

7.2 CONCLUSION

Using machine learning and pre-trained models, we designed a Blind Assistance System that assists in detecting objects in the environment. For this project, we utilized Python, YOLOv8n, and the COCO Dataset. The system uses object detection to identify items and provides real-time audio feedback to the user.

This proposed system has a variety of applications, making it easier for blind individuals to acquire, analyze, and translate information about their surroundings. The primary goal is to enable visually impaired persons to navigate freely, allowing them to move quickly and safely. The device provides distance and object detection, giving users speech-based awareness of their environment.

In conclusion, the proposed blind assistance system effectively integrates cutting-edge computer vision and machine learning technologies to provide real-time object detection and environmental awareness for visually impaired individuals. By utilizing an Android application to capture frames and a laptop-based server to perform object detection using a pre-trained SSD model on the COCO dataset,

the system offers accurate identification and classification of objects. The audio feedback mechanism ensures that users receive immediate, understandable information about their surroundings, enhancing their independence and safety.

Moreover, the incorporation of an alert system to calculate distances between the user and detected objects provides an additional layer of safety by notifying users when they are too close to obstacles. This system, leveraging the efficient and fast YOLOv3 model, ensures that the solution remains lightweight and practical for real-world application on devices with limited computational power.

Overall, this comprehensive approach addresses both the detection and communication needs of blind users, offering a significant step forward in assistive technology for visually impaired individuals. The continuous improvement and user feedback loops integrated into the system will further refine its performance and usability, ensuring that it remains an invaluable tool for enhancing the quality of life for those it serves.

7.3 REFERENCE

1. https://hindustanuniv.ac.in/assets/naac/CA/1_3_4/2666_NAVEEN_KUMAR.pdf
2. https://www.geeksforgeeks.org/python-opencv-cv2-imshow-method/#:~:text=OpenCV%2DPython%20is%20a%20library,an%20image%20in%20a%20window.
3. https://www.quora.com/Why-should-I-pick-the-TensorFlow-Object-Detection-API-over-YOLOv3
4. https://www.reddit.com/r/tensorflow/comments/uoqkwm/newbie_question_what_is_the_difference_between/
5. https://www.scaler.com/topics/object-detection-using-opencv-python/
6. https://typeset.io/questions/what-are-the-limitations-of-the-arduino-uno-uq9ueh6f7l#:~:text=Arduino%20Uno%20has%20several%20limitations,corruption%20and%20unexpected%20behavior%201.
7. https://digitalcommons.odu.edu/cgi/viewcontent.cgi?article=1077&context=covacci-undergraduateresearch#:~:text=Using%20the%20Raspberry%20Pi%2C%20a,attacks%20(Sepio%2C%202023).
8. https://www.ijraset.com/research-paper/blind-assistance-system-using-image-processing
9. https://www.journal-dogorangsang.in/no_2_Book_21/60.pdf
10. https://ijcrt.org/papers/IJCRT2401367.pdf