

Object Detection Assistant

An accessible, real-time object detection system that provides spatial awareness through voice feedback. This system uses computer vision and text-to-speech to identify objects in the environment, their distance, position, and movement patterns.

Features

- **Real-time object detection** using YOLOv8 to identify common objects
- **Spatial awareness** with direction and distance information
- **Movement tracking** that estimates object speed and direction
- **Voice feedback** that announces detected objects and their properties
- **Calibration mode** to improve distance calculations for different cameras
- **Performance optimizations** for smooth real-time operation

Requirements

- Python 3.7+
- Webcam
- Internet connection (for initial model download)
- GPU recommended but not required

Installation

1. Clone the repository:

bash Copy

```
git clone https://github.com/yourusername/object-detection-assistant.git  
cd object-detection-assistant
```

2. Create a virtual environment (recommended):

bash Copy

```
python -m venv venv  
source venv/bin/activate # On Windows: venv\Scripts\activate
```

3. Install dependencies:

bash

 Copy

```
pip install -r requirements.txt
```

4. Download the YOLOv8 model (will be done automatically on first run or manually):

bash

 Copy

```
# Automatic: First run downloads model  
# Manual:  
pip install ultralytics  
python -c "from ultralytics import YOLO; YOLO('yolov8n.pt')"
```

Usage

Run the main script:

bash

 Copy

```
python object_detection.py
```

Controls:

- **Q:** Quit the application
- **Space:** Repeat the last announcement
- **C:** Enter calibration mode (follow voice instructions)

How It Works

1. **Detection:** The system captures video frames from your webcam and processes them with YOLOv8 to detect objects.
2. **Distance Calculation:** For each detected object, the system estimates the distance using the apparent width of the object and the focal length of the camera.
3. **Tracking:** Objects are tracked across frames to calculate their movement speed and direction.
4. **Voice Feedback:** The system announces information about detected objects, including:
 - Object type (person, car, etc.)
 - Position (left, right, or ahead)
 - Distance (in centimeters)
 - Movement speed and direction

5. **Visualization:** The system displays the video feed with bounding boxes, labels, distance, and speed information.

Customization

Object Widths

The `OBJECT_WIDTHS` dictionary contains the typical width of various objects in centimeters. You can modify these values for better accuracy:

python

 Copy

```
OBJECT_WIDTHS = {  
    "person": 50,  
    "bicycle": 60,  
    # Add or modify objects as needed  
}
```

Speed Thresholds

Adjust the `speed_threshold` variable to change when movement is reported:

python

 Copy

```
speed_threshold = 5 # cm/s - speeds above this threshold will be announced
```

Voice Settings

Modify the speech rate and volume in the main function:

python

 Copy

```
engine.setProperty('rate', 180) # Speed of speech  
engine.setProperty('volume', 1) # Volume (0-1)
```

Calibration

For more accurate distance measurements, use calibration mode:

1. Press **C** to enter calibration mode
2. Place a known object exactly 1 meter (100 cm) from the camera

3. Press **Space** to calibrate
4. The system will use the largest detected object for calibration

Troubleshooting

- **No video feed:** Check if your webcam is properly connected and not used by another application
- **Slow performance:** Lower the resolution by changing these values:

python

 Copy

```
video.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
video.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
```

- **Inaccurate distances:** Use the calibration feature with a known object at a known distance

Contributing

Contributions are welcome! Please feel free to submit a Pull Request.

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.

Acknowledgments

- [Ultralytics YOLOv8](#) for object detection
- [OpenCV](#) for computer vision capabilities
- [pyttsx3](#) for text-to-speech functionality