**Smart Assistant for the Visually Impaired**

This project uses computer vision, text recognition, and text-to-speech capabilities to assist visually impaired individuals by detecting objects, reading text, and scanning barcodes using a webcam.

**Features**

* **Object Detection**: Identifies objects in the frame using the YOLOv3 deep learning model.
* **OCR (Optical Character Recognition)**: Reads text from detected objects using Tesseract OCR.
* **Barcode & QR Code Scanning**: Detects and reads barcode/QR codes in real-time.
* **Text-to-Speech (TTS)**: Converts detected objects and text into speech output.
* **Multilingual Support**: Uses Google TTS for speech output in multiple languages.

**Prerequisites**

Ensure you have the following installed:

* Python 3.x
* OpenCV (cv2)
* NumPy
* pyttsx3 (offline text-to-speech)
* gTTS (Google text-to-speech for multilingual support)
* Pytesseract (OCR for text detection)
* pyzbar (Barcode and QR code detection)

**Install Dependencies**

pip install opencv-python numpy pyttsx3 gTTS pytesseract pyzbar langdetect

**Additional Setup**

**Install Tesseract OCR:**

* **Windows**: [Download Tesseract OCR](https://github.com/UB-Mannheim/tesseract/wiki)
* **Linux (Ubuntu/Debian)**:
* sudo apt install tesseract-ocr
* **macOS**:
* brew install tesseract

Set the Tesseract path in your script (only for Windows users):

pytesseract.pytesseract.tesseract\_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

**How to Run the Project**

python smart\_assistant.py

**Project Files**

| **File** | **Description** |
| --- | --- |
| smart\_assistant.py | Main Python script for object detection, OCR, and barcode scanning. |
| yolov3.weights | Pre-trained YOLO model weights. |
| yolov3.cfg | YOLO configuration file. |
| coco.names | File containing class labels for object detection. |

**How it Works**

1. The script initializes the YOLOv3 model for object detection.
2. It opens the webcam and processes frames in real-time.
3. Objects are detected and announced via text-to-speech.
4. Any readable text in the frame is extracted using Tesseract OCR and spoken aloud.
5. Barcodes and QR codes are scanned and read out loud.
6. The user can quit the application by pressing q.

**Contribution**

Feel free to contribute to this project by improving detection accuracy, adding new features, or optimizing performance.