

YOLO GUI Detection System

Professional Technical Documentation

Version 1.1 | February 16, 2026

Index

Index	1
YOLO GUI Detection System	2
Table of Contents	2
1. Purpose	2
2. System Overview	2
3. Architecture	2
4. Supported YOLO Models	3
5. Installation and Environment Setup	3
6. Application Execution	3
7. User Workflow	3
8. Person Alert Behavior	4
9. Troubleshooting	4
10. Configuration and Customization	4
11. Performance Guidance	5
12. File Reference	5

YOLO GUI Detection System

Document Version: 1.1

Date: February 16, 2026

Table of Contents

1. Purpose
2. System Overview
3. Architecture
4. Supported YOLO Models
5. Installation and Environment Setup
6. Application Execution
7. User Workflow
8. Person Alert Behavior
9. Troubleshooting
10. Configuration and Customization
11. Performance Guidance
12. File Reference

1. Purpose

This document defines setup, operation, and maintenance guidance for the GUI-based YOLO webcam detection system.

2. System Overview

The application provides real-time object detection from webcam input with a desktop GUI.

Core functions:

- Model selection from a dropdown
- Live object detection and bounding boxes
- Person detection alert with audible beep
- Manual stop from GUI or keyboard

3. Architecture

The solution is implemented in Python with:

- Tkinter for GUI
- OpenCV for webcam capture and video display
- YOLOv5 models loaded by `torch.hub`
- YOLOv8/YOLO11 models loaded by `ultralytics.YOLO`

Execution flow:

1. User selects a model in GUI.
2. App loads selected model backend.
3. Webcam stream is read frame-by-frame.
4. Inference runs and detections are rendered.
5. If person class is present, beep alert is triggered.

4. Supported YOLO Models

Current dropdown model options:

- YOLOv5: `yolov5n`, `yolov5s`, `yolov5m`, `yolov5l`, `yolov5x`
- YOLOv8: `yolov8n`, `yolov8s`, `yolov8m`, `yolov8l`, `yolov8x`
- YOLO11: `yolo11n`, `yolo11s`, `yolo11m`, `yolo11l`, `yolo11x`

Notes:

- First model load may download weights from internet.
- Smaller variants (`n`, `s`) are faster on CPU.
- Larger variants (`l`, `x`) provide stronger accuracy at lower FPS.

5. Installation and Environment Setup

Windows PowerShell commands:

```
cd c:\Users\ELCOT\Downloads\codex
python -m venv .venv
.\.venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
pip install -r requirements.txt
```

6. Application Execution

Run command:

```
.\.venv\Scripts\Activate.ps1
python app.py
```

7. User Workflow

1. Launch application.
2. Select required model from dropdown.
3. Click `Start Detection`.
4. Observe live detections in webcam window.
5. Stop by clicking `Stop Detection` or pressing `Q`.

8. Person Alert Behavior

- Person class identifier is COCO class `0`.
- App overlays `PERSON DETECTED` on frame.
- Beep uses Windows `winsound.Beep(1200, 250)`.
- Alert cooldown is 1 second to reduce repeated sound spam.

9. Troubleshooting

9.1 OpenCV putText error

Symptom:

- `Expected Ptr<cv::UMat> for argument 'img'`

Cause:

- Drawing performed on non-writable image memory.

Resolution implemented:

- Frames are converted to writable format using:
- `np.ascontiguousarray(...).copy()`

9.2 Webcam not opening

- Close other apps using webcam.
- Check Windows privacy camera permissions.
- Confirm device index in `cv2.VideoCapture(0)`.

9.3 Slow first startup

- Initial delay is expected during first model weight download.

9.4 No beep sound

- Confirm Windows volume/output device.
- Verify person object is detected in frame.

10. Configuration and Customization

Model entries are defined in `MODEL_OPTIONS` in `app.py`.

Example:

```
"yolov8n": {"backend": "ultralytics", "weights": "yolov8n.pt"}
```

You can:

- Add new model keys to dropdown

- Change weight files
- Map model to `v5` or `ultralytics` backend

11. Performance Guidance

- Prefer `yolov5n`/`yolov8n` for low-resource systems.
- Use higher models only when hardware can sustain target FPS.
- Close unnecessary background applications.

12. File Reference

- `app.py`: GUI, model loading, inference loop, beep logic
- `requirements.txt`: dependency definitions
- `README.md`: quick start commands
- `DOCUMENTATION.md`: this technical document
- `DOCUMENTATION.pdf`: professional PDF export