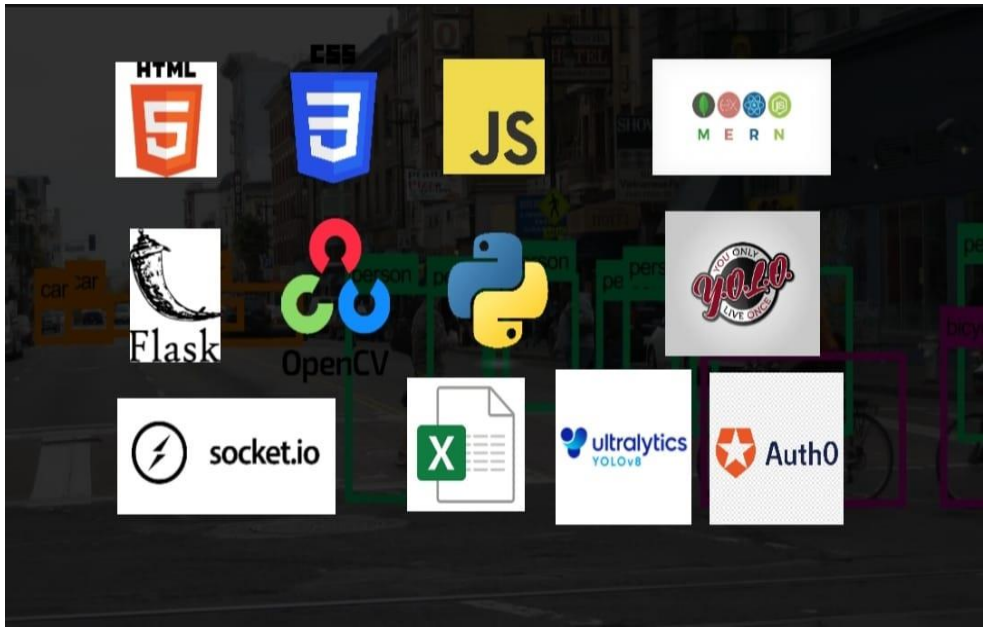
(Progect)- Real-Time Object Detection

OVERVIEW

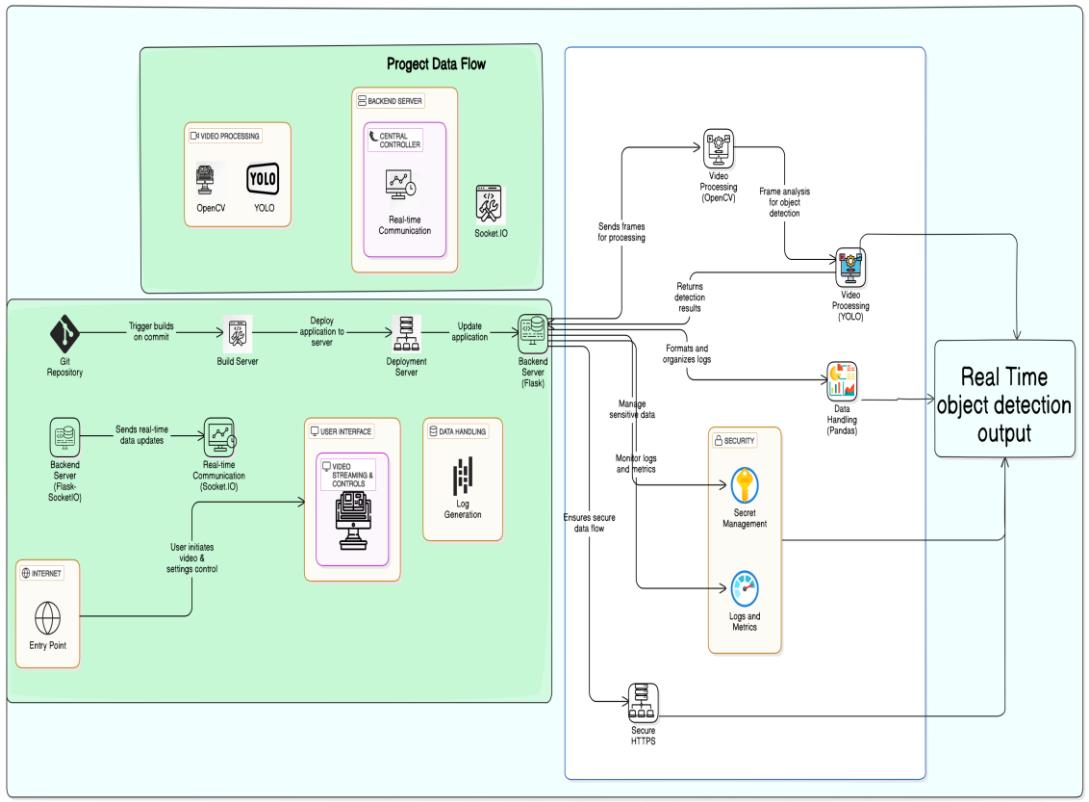
1. Tech Stack
2. Key Features
3. Object Detection Model
4. Future Enhancements
5. Use Cases & Installation

Tech stack



(Progect) Technologies used

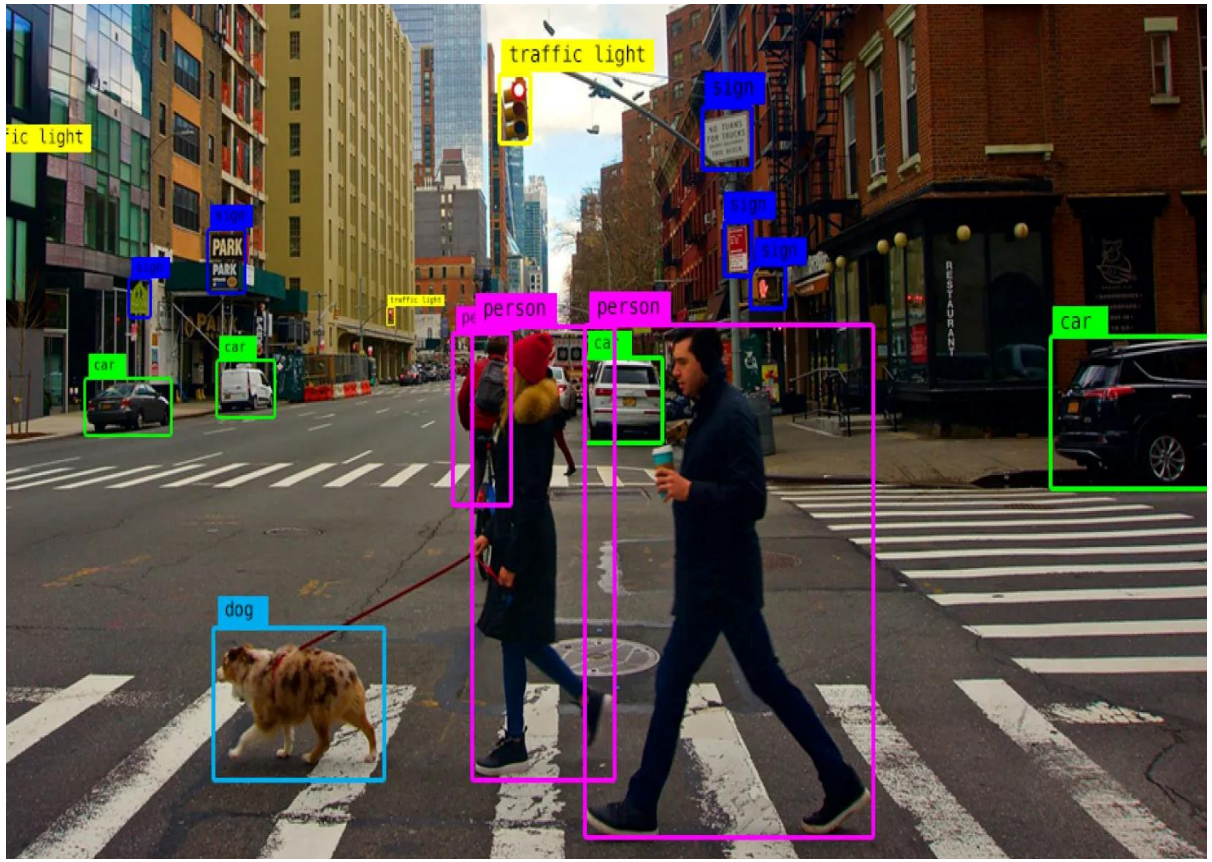
- **HTML**: Structures the layout for video display, object detection details, and user controls.
- **JavaScript**: Adds interactivity, enabling real-time video updates, detection settings adjustments, and log downloads.
- **Python**: Manages backend operations, video processing, and object detection using image processing libraries.
- **Flask**: Serves as the backend framework, handling HTTP requests and data flow.
- **Flask-SocketIO**: Enables real-time, bidirectional communication between frontend and backend for smooth video feed updates.
- **OpenCV**: Used in Python for real-time video frame capture, manipulation, and transformation.
- **YOLO (You Only Look Once)**: Detects objects in real-time on video frames, drawing bounding boxes and labels.
- **Pandas**: Organizes detection data into an Excel format for creating downloadable logs with metadata.
- **Socket.IO**: Ensures seamless data flow between server and client for live video stream updates.



DATA FLOW ARCHITECTURE DIAGRAM (PROGECT)

Key Features

1. **Live Video Streaming**: Stream live video directly within the web browser.
2. **Real-Time Object Detection**: Apply object detection on live video feeds with bounding boxes.
3. **Detection Settings**: Adjust detection parameters, including threshold and frequency.
4. **Downloadable Logs**: Export detection logs in JSON format, capturing details like object type, confidence level, and timestamp.
5. **Dashboard**: View current and historical detections, logs, and statistical insights.
6. **Differentiated Object Detection**: Detect different object types—**persons** and **vehicles**—with unique bounding box colors for better visual distinction.



ODM

yolo object detection

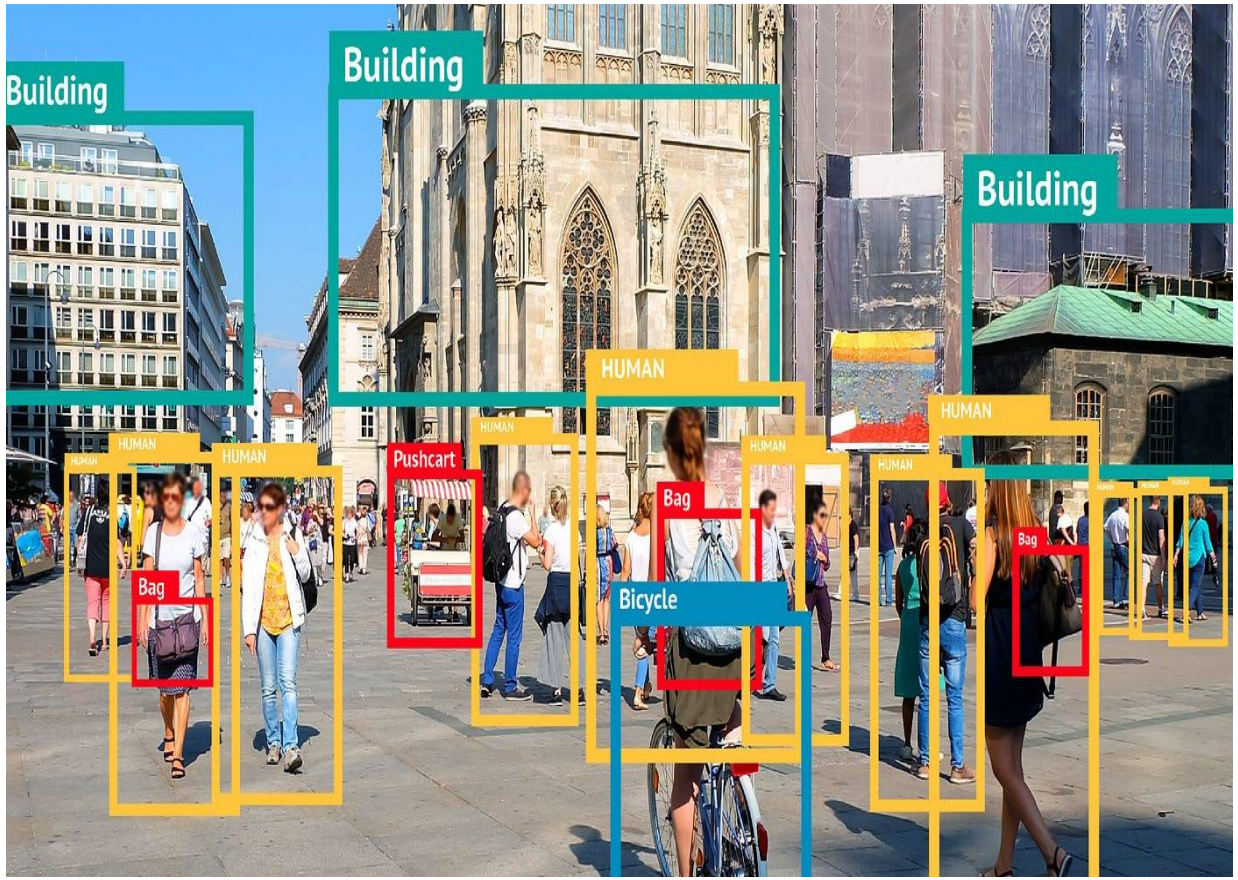
Object Detection Model

•**YOLO-Based Object Detection**
•**Model**: YOLO (You Only Look Once) model used to detect objects in real-time with class-based bounding boxes.

•**Detected Classes**: Over **80** object types including vehicles, animals, household items, and more.

•**Example classes:** Person, car, bicycle, dog, laptop, etc.

video link :
https://drive.google.com/file/d/1kb_AU5jfYz4YmRZbpKbb4_xNtrJBpP2p/view?usp=drive_link



Future Enhancements

Future Enhancements

- **Upcoming Features**:**Custom Model Support**: Allow users to deploy custom detection models.
- **Multi-Camera Integration**: Support multiple video sources.
- **Mobile Compatibility**: Extend capabilities to mobile platforms.
- **Automated Alerts**: Notifications based on specified detection events.
- **Advanced Analytics**: Provide data insights like heatmaps and trend analysis.

How It Works

The `model.pt` file contains a YOLO-based object detection model that identifies the above classes in real-time video feeds. The model works by drawing bounding boxes around each detected object and labeling them with the corresponding class name. This is achieved by processing the video frames through the model, which predicts the class labels and confidence scores for each detected object.

Installation

1. Clone the repository:

```
git clone https://github.com/username/progect.git
cd progect
pip install -r requirements.txt
python app.py
```

Navigate to the project directory:

cd progect

Install the required dependencies:

pip install -r requirements.txt Start the application:

python app.py Open your browser and go to http://127.0.0.1:5000 to view the application.

Use Cases & Installation

Use Cases & Installation

- **Use Cases & Getting Started**

- **Potential Applications**:
- **Security & Surveillance**: Monitor and identify security risks.
- **Retail & Smart Cities**: Track customer and traffic patterns.
- **Healthcare & Agriculture**: Observe patient or crop conditions.

GITHUB LINK : Progect-detector/README.md at main · DurgaPrashad/Progect-detector

THANK YOU