

Qualcomm VisionX

Project Name : DetectX

Subtitle : "Round 1 Submission for IIT Bombay Techfest 2024"

Team Details:

Team Name : ETHAN

Team Members :

- ANIRUTH PRABU
- SRIVATSAN K
- RIJUL NAMBIAR P

Team ID : Qual-230345

Email: aniruthprabuk@gmail.com Phone: 8838818224

Introduction

- **Objective:**
 - Automate object detection in images and videos using TensorFlow Object Detection API.
 - Perform instance segmentation to identify specific object areas.
- **Key Features:**
 - Real-time detection
 - Pretrained models for accuracy and speed
 - Applications: Autonomous vehicles, surveillance, retail analytics.





Technology Stack

- Frameworks: TensorFlow Object Detection API, NumPy, PIL, and Matplotlib.
- Tools: Python, Jupyter Notebook, TensorFlow Model Zoo.
- Models Used:
 - SSD MobileNet v1 for object detection.
 - Mask R-CNN for instance segmentation.
- Environment: Python 3.8, TensorFlow 2.x.

Dataset and Pretrained Models

- Dataset: COCO Dataset (Common Objects in Context).
- Model Zoo:
- SSD MobileNet v1: Lightweight model for object detection.
- Mask R-CNN: Advanced model for instance segmentation.



Workflow Overview

- Steps:
 - a. Setup Environment: Install dependencies, clone TensorFlow models, compile protos.
 - b. Load Pretrained Models: Use SSD MobileNet v1 and Mask R-CNN.
 - c. Process Images/Videos: Convert images to tensors for inference.
 - d. Run Inference: Detect objects and visualize results.
 - e. Display Output: Annotated images/videos with bounding boxes and labels.



Code Highlights

- **Environment Setup:**
 - Installation of TensorFlow, pycocotools, and TensorFlow models.
 - Compilation of protobuf files.
- **Model Loading:**
 - `load_model` function for downloading and loading models.
 - Customizable with different models from TensorFlow Model Zoo.
- **Visualization:**
 - Bounding boxes and labels visualized using utility functions.

Detection Process

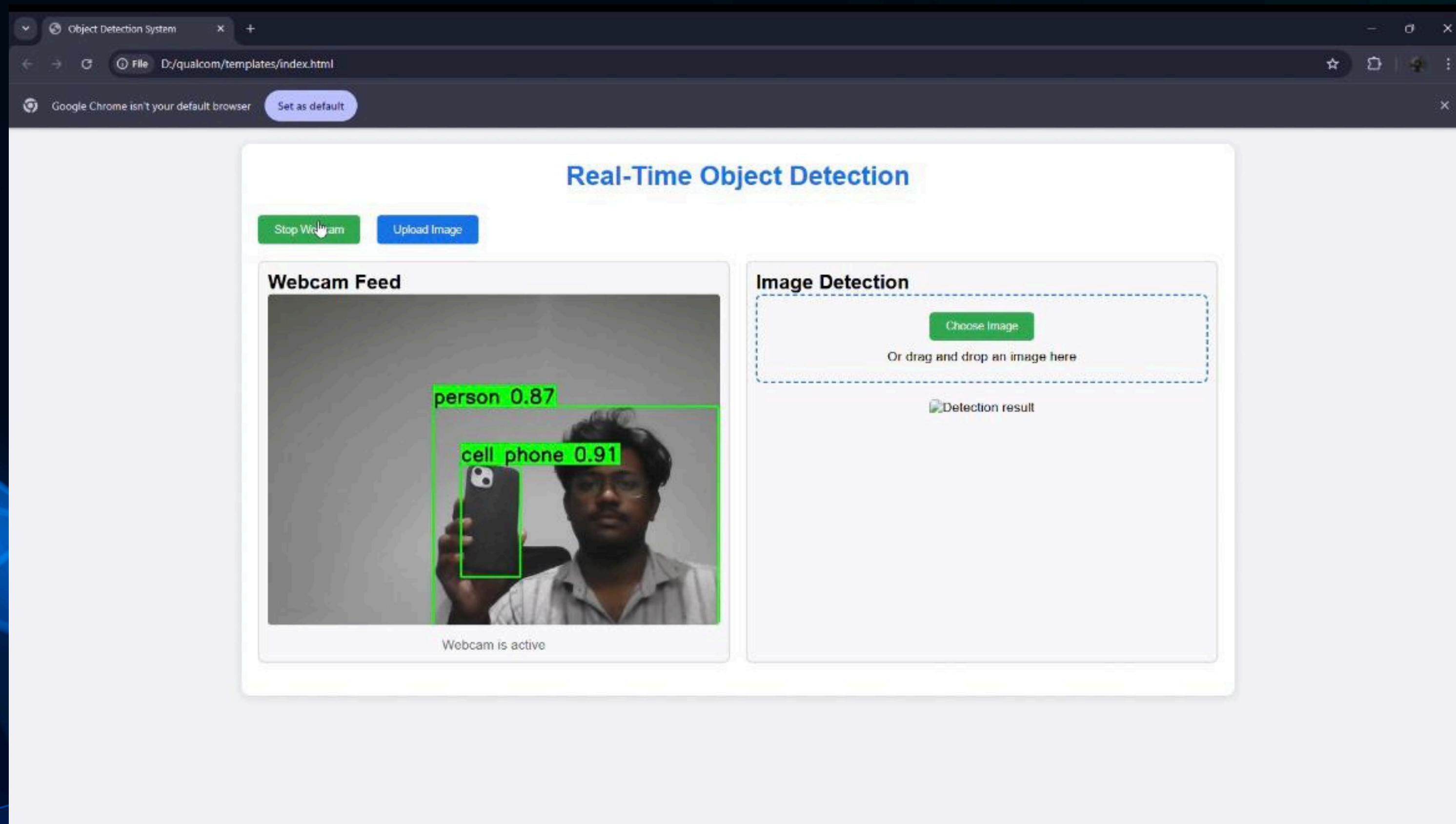
- Inputs: Batch of RGB images (uint8 format).
- Outputs:
- Bounding boxes, detection scores, and class labels.
- Instance segmentation masks for Mask R-CNN.
- Inference Function:
- `run_inference_for_single_image`: Processes single images for detection.
- Outputs cleaned data for easy visualization.



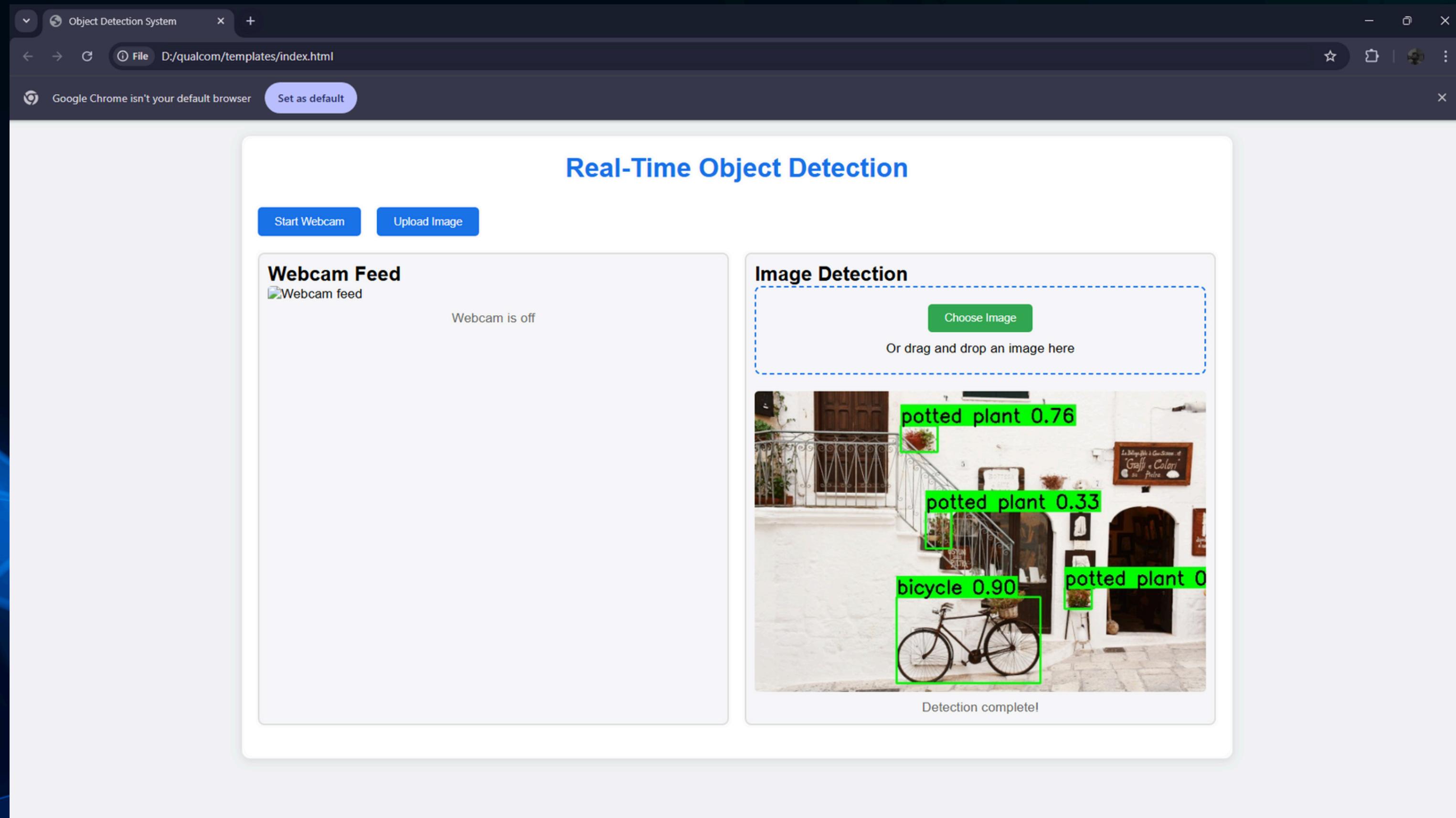


Results and Visualization

- Object Detection Results:
- Annotated images showing detected objects with bounding boxes and labels.
- Instance Segmentation Results:
- Segmented masks highlighting detected object areas.



- Experience Our Prototype in Google Drive: <https://drive.google.com/drive/folders/1R6m--BWw4cMkEwzGqYRBfGMYj-8VlFZ?usp=sharing>



- Experience Our Prototype in Youtube: https://youtu.be/mOKVw-0DdGw?si=SSbWjaBTgG_8qhcY

Conclusion and Future Work

- Conclusion:
 - Successfully implemented object detection and instance segmentation.
 - High accuracy and reliable visualization with pretrained models.
- Future Enhancements:
 - Use custom datasets for specific applications.
 - Deploy models on edge devices for real-time inference.
 - Extend to multi-object tracking in videos.

