

Documentation: Real-Time Object Recognition System Using Mediapipe and OpenCV

1. Introduction

I build a real-time object recognition system using MediaPipe and OpenCV. It detects and classifies a small set of common objects (e.g., a cup, a phone) in a live video stream from a camera. The system utilizes MediaPipe's object detection and EfficientDet Lite0 pre-trained mediapipe model for classification. OpenCV is used for video capture and displaying the video with bounding boxes and object labels.

2. Objectives

The goal of the project is to:

- Detect and recognize objects in real-time using a webcam feed.
- Display the detected objects with bounding boxes and labels on the screen.
- Use a pre-trained model to classify the detected objects with appropriate labels.

3. System Overview

The system processes real-time video input from the webcam, applies object detection using the pretrained mediapipe model, and show the video with bounding boxes and labels. The detection is visualized in real-time using OpenCV.

4. Technologies Used

- **Python:** The core development language.
- **Mediapipe:** A cross-platform framework for building pipelines for object detection and classification.
- **OpenCV:** A computer vision library used for video capture and displaying results.
- **Pretrained Model:** I chose MediaPipe's Object Detector EfficientDet Lite0 model for object detection due to its balance between accuracy and computational efficiency. EfficientDet Lite0 is a lightweight version optimized for real-time applications on resource-constrained devices, such as mobile phones and edge devices. It provides fast inference speeds and satisfactory accuracy for detecting small objects like cups and phones. Additionally, since it uses TensorFlow Lite, the model is optimized for efficient on-device inference, eliminating the need for external computation resources.
- The model pretrained on cocodataset (<https://cocodataset.org/#home>). It cover only 80 common objects.

5. Key Components

1. Object Detection with Mediapipe:

- **MediaPipe's Object Detector** is used to detect a small set of common objects in the video stream. The detector is configured using a TensorFlow Lite model (efficientdet_lite0.tflite), which offers a balance of speed and accuracy for real-time applications.

2. Preprocessing and Visualization:

- I pre-processed the video frames from the camera via noise reduction (Gaussian blur) before being fed into the object detection model.
- After detection, bounding boxes are drawn around the detected objects, and labels with their respective confidence scores are displayed.

3. Real-Time Processing:

- The system continuously captures frames from the webcam using OpenCV, processes them using the Mediapipe object detection model, and visualizes the results (bounding boxes and labels) in real-time.

4. User Interface:

- The annotated video feed is displayed in an OpenCV window.
- The user can exit the system by pressing the 'Esc' key.

5. Result:

- I recorded the video of live object detection and store the result in Results folder. You can clearly view result there.
- It contains 2 pictures, and 2 video recordings.

Conclusion

This project demonstrates how to integrate Mediapipe and OpenCV for real-time object recognition and classification using a pre-trained model. The system is efficient, accurate, and displays object detection results with bounding boxes and labels in real-time.

Future Improvements

- Integration of more advanced models for higher accuracy (e.g., SSD or YOLO).
- The current pre-trained model cannot be used in specific context. It's just for detection of common objects. If your company goal is to use this for specific context, then Adaptation to domain and fine-tuning will be required.

HOW TO RUN THIS PROJECT:

- Install the required library via (pip install mediapipe opencv-python numpy tensorflow)
- Run the code
- Monitor the model detection
- Press Esc to terminate