**Real-Time Object Detection with DETR and OpenCV**

This project demonstrates real-time object detection using the **DETR (DEtection TRansformer)** model in combination with **OpenCV** to capture live video input. The model is pre-trained on the COCO dataset, allowing it to recognize and label common objects, such as people, vehicles, animals, and everyday items, directly from your laptop’s camera.

**Project Overview**

This application captures frames from the laptop's camera and applies the **DETR** object detection model to identify objects within each frame. The project showcases how to integrate Hugging Face Transformers with OpenCV for real-time object detection using deep learning models.

**Key Features**

* **Real-Time Object Detection**: Displays live bounding boxes and labels for objects detected with high confidence (>70%).
* **GPU Support**: Automatically utilizes GPU if available, enhancing detection speed.
* **COCO Dataset Labels**: Recognizes and labels objects across 80 common categories defined by the COCO dataset.

**Installation**

1. Clone the repository:

git clone https://github.com/NEVZ-K/Real-Time-Object-Detection.git

cd Real-Time-Object-Detection

1. Install the required Python libraries:

pip install torch torchvision transformers opencv-python

1. Ensure you have a laptop or webcam connected to your device for real-time video feed.

**Usage**

1. Run the script:

python object\_detection.py

1. The application will open a window displaying the live camera feed with bounding boxes and labels for detected objects. Press **'q'** to exit the application.

**Code Explanation**

* **Loading the Model**: The DETR model is loaded using Hugging Face’s transformers library.
* **Device Configuration**: Sets up automatic GPU usage if available.
* **Object Detection Loop**: Captures frames in real-time, preprocesses them, and runs inference using DETR.
* **Bounding Box and Label Visualization**: Draws bounding boxes around objects with a confidence score greater than 0.7.

**Example Output**

The application outputs a live video feed with real-time object detection annotations. For each detected object, it shows:

* **Bounding Box**: A rectangle drawn around the detected object.
* **Label and Confidence Score**: The object's category label from the COCO dataset and its confidence score.

**Project Structure**

* object\_detection.py: The main script for running real-time object detection.
* README.md: This README file.

**Requirements**

* Python 3.8+
* CUDA-enabled GPU (optional but recommended for improved performance)
* Dependencies as listed in requirements.txt

**License**

This project is licensed under the MIT License.