Title

Real-Time Object Detection Using Python

Abstract

This project demonstrates a real-time object detection system using the YOLOv5s model pretrained on the COCO dataset. With the power of PyTorch, OpenCV, and Matplotlib, the system allows users to upload an image and obtain bounding boxes for detected objects with class labels and confidence scores. The system is implemented in Python and runs in a Google Colab environment for ease of use.

Tools, Technologies and Materials

- Programming Language: Python
- Frameworks & Libraries:
  - PyTorch (for model inference)
  - OpenCV (for image processing)
  - PIL (image manipulation)
  - Matplotlib (for result visualization)
- Model Used: YOLOv5s (pretrained on COCO dataset via Ultralytics GitHub)
- Repositories: Ultralytics YOLOv5 GitHub
- Environment: Google Colab / Visual Studio Code

Introduction

Object detection plays a vital role in computer vision by identifying and localizing objects in images or video frames. This project focuses on using YOLOv5s-a highly optimized, real-time model capable of detecting 80 classes from the COCO dataset. The aim is to demonstrate a fast and accurate implementation that can serve as a foundation for more advanced real-time applications.

System Design

1. Setup Phase:

- Install dependencies such as torch, opencv-python, pillow, etc.

- Clone the YOLOv5 repository and install the required packages.

2. Model Initialization:

- Load the pretrained YOLOv5s model using torch.hub.

- Set the model to evaluation mode for inference.

3. Interface Phase:

- Use Google Colab's files.upload() method to upload an image.

- Process the image using OpenCV and PIL to prepare it for detection.

System Flow

1. User uploads an image.

2. Image is converted to BGR format for compatibility with OpenCV.

3. YOLOv5 model processes the image and returns detections.

4. Detected objects are drawn using bounding boxes with labels.

5. Results are displayed using matplotlib.

Implementation Details

A cleaned and commented version of the Python code is used to:

- Load and preprocess an image

- Run inference using YOLOv5

- Display results using OpenCV and Matplotlib

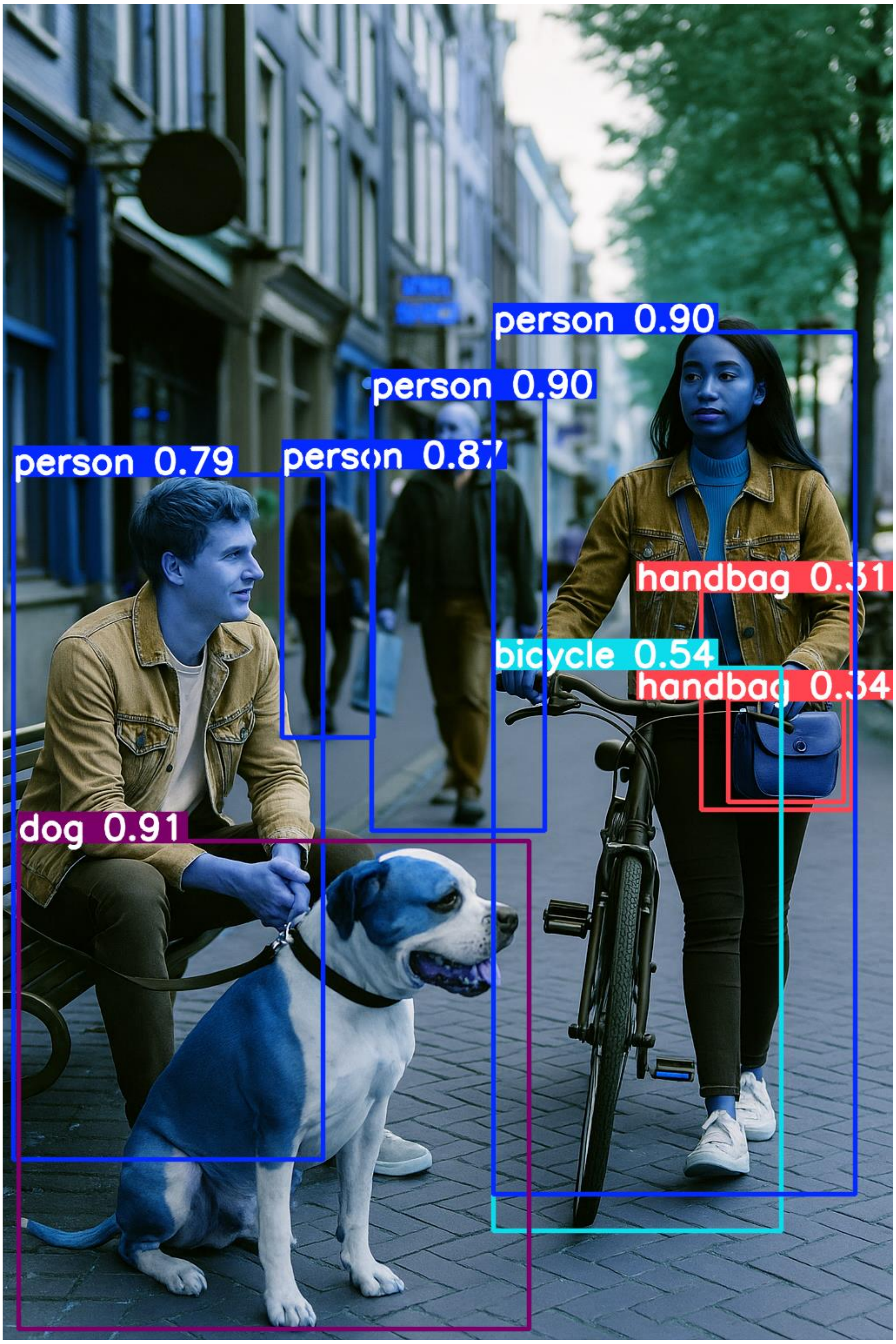
The code is suitable for running inside Google Colab for quick testing.

Result

After uploading an image, the model successfully identifies various objects within the scene, including their bounding boxes and confidence scores. The image is displayed with annotations, and results are saved in the default output directory runs/detect/exp.

Conclusion

This project effectively demonstrates a real-time object detection system using YOLOv5. The use of PyTorch, OpenCV, and Google Colab simplifies the deployment process and offers a great starting point for advanced implementations like video stream detection or integration into web/mobile applications. The model's high accuracy and speed make it suitable for a wide range of real-world applications.


person 0.90
person 0.90
person 0.79
person 0.87
handbag 0.31
bicycle 0.54
handbag 0.34
dog 0.91