E22CSEU1284

Training and Testing Pipeline for Mask Wearing Dataset

1. Introduction

The goal is to train and test object detection models—YOLOv5 and Faster R-CNN—on the **Mask Wearing Dataset**. This dataset contains labeled images of people with and without masks, enabling the models to detect whether individuals are wearing masks.

Why these models?

- **YOLOv5** is known for its speed and efficiency, making it suitable for real-time applications.
- **Faster R-CNN** is more accurate but computationally heavier, often preferred in scenarios where accuracy is more critical than inference speed.

2. Dataset Preparation

1. **Download Dataset**
   - The dataset is available on Roboflow.
   - It contains annotated images in formats compatible with YOLO and COCO (for Faster R-CNN).
2. **Preprocessing**
   - Convert images and annotations into the required format for each model.
   - Resize images to match the model's input size (e.g., 640x640 for YOLOv5).
   - Normalize pixel values.
   - Split into training, validation, and test sets (e.g., 80% training, 10% validation, 10% testing).

3. Training the Models

YOLOv5 Training Steps:

- Use a pre-trained YOLOv5 model (e.g., YOLOv5s, YOLOv5m).
- Modify the configuration file to match the dataset classes (e.g., "with_mask", "without_mask").
- Train the model using transfer learning on a GPU.
- Save the trained model weights.

Faster R-CNN Training Steps:

- Use a pre-trained Faster R-CNN model (e.g., based on ResNet-50).
- Fine-tune the model on the Mask Wearing dataset.
- Train for multiple epochs, monitoring loss and performance.
- Save the trained model.

4. Testing and Evaluation

- Run inference on test images.
- Measure performance using:
  - **Mean Average Precision (mAP)**
  - **Precision and Recall**
  - **Confusion Matrix**
  - **Inference speed (YOLO vs. Faster R-CNN)**
- Visualize results with bounding boxes overlaid on test images.

Feature Extraction from Stanford Car Dataset

1. Objective

We aim to extract visual features from images of cars using different pre-trained CNN models (ResNet-50, VGG-16, VGG-19, and InceptionNetV3). The extracted features will help in tasks like classification, clustering, or retrieval.

2. Dataset Overview

- The **Stanford Cars Dataset** contains images of various car models.
- Each image has a label corresponding to the car's make and model.

3. Feature Extraction Process

Step 1: Load Pretrained CNN Models

- Use TensorFlow Keras models like ResNet-50, VGG-16, VGG-19, and InceptionNetV3.
- Load the weights trained on ImageNet.

Step 2: Select Three Layers for Feature Extraction

- Extract features from different layers (e.g., an early convolutional layer, an intermediate layer, and a deep layer).
- Early layers capture basic textures, edges, and shapes.
- Deeper layers capture complex object-level features.

Step 3: Extract and Visualize Features

- Select 5 random images from the dataset.
- Pass each image through the network and extract features.
- Convert extracted features into a visual representation using a technique like **Feature Maps** or **Grad-CAM**.
- Display the extracted features as images.

Final Report Structure

1. Introduction

- Overview of both tasks (Object Detection & Feature Extraction).
- Importance of object detection in mask-wearing compliance.
- Role of feature extraction in car classification.

2. Training and Testing YOLOv5 & Faster R-CNN

- Dataset description.
- Training methodology for YOLOv5 and Faster R-CNN.
- Evaluation metrics and results.
- Comparison of model performance.

3. Feature Extraction on Stanford Car Dataset

- CNN models used for feature extraction.
- Layers selected and their significance.
- Visualization of extracted features.

4. Conclusion

- Summary of findings.
- Observations on YOLOv5 vs. Faster R-CNN.
- Insights from feature extraction.
- Future work recommendations.