ReadMe

Two-Wheeler Detection System

Overview

This research project presents a cutting-edge object detection and classification system by integrating **GroundingDINO**, **Segment Anything (SAM)**, and **DINOv2** models. The system is designed to detect and classify objects based on **template (reference) images**, offering high accuracy and generalization capabilities. This research is focused on detecting two-wheelers such as **bikes, scooters, and bicycles**.

🔑 Key Features

- **Template-based detection**: Detect and classify objects using reference images.
- **Customizable classification**: Easily extend the system to support new object classes.
- **Comprehensive outputs**: JSON metadata and color coded visual outputs.
- **Command-line interface (CLI)**: Simple and flexible configuration using CLI arguments.

🛠️ Installation

Prerequisites

- Set up the environment using instructions from the `ReadMe_conda` file.
- Refer to the `ReadMe_python` file for step-by-step setup and execution guidance.

📁 Repository Structure and File Descriptions

📂 Experiment 1

Objective: Compare the original architecture from this paper with our modified architecture for detection and segmentation of a **single class ("Two-Wheelers")**.

- `nids_net_experiment1_authors.py`

  ➤ Original implementation from the authors.

  ➤ Supports only a single object class.

- `nids_net_experiment1_ours.py`

  ➤ Our improved implementation.

  ➤ Capable of handling multiple classes, but in this experiment, it saves predictions under a unified label: **"Two-Wheelers"**.

🧪 `main_poc.py` (Proof of Concept)

This is our **multi-class detection and segmentation** implementation. It supports CLI execution.

➡️ **Usage instructions** can be found in the `ReadMe_python` file.

📂 Experiment 2

Objective: Compare our model (`main_poc.py`) with **YOLOv8** on the **object detection task**.

📂 Experiment 3

Objective: Compare our model (`main_poc.py`) with **YOLOv8** on the **segmentation task**.