Campy

Table of Contents

Description

The provided Python script defines a class called Campy that connects to security cameras, performs object detection using the YOLOv5 model, and saves frames with detected objects. The class is designed to work with multiple cameras and allows users to specify a list of classes they want to detect. It uses OpenCV for camera access and image processing, Torch for running the YOLOv5 model, and the json5 library for loading camera credentials from a JSON file. The script also demonstrates how to use the Campy class in demo mode, showing how objects are detected and frames are saved for each camera.

Requirements

To run the provided Python script successfully, you will need to have the following requirements installed:

Python:

Make sure you have Python installed on your system. The script is written in Python, so you need to have a compatible version of Python installed (Python 3.x is recommended).

RTSP securty system

a security system that is compatible with RTSP and has RTSP enabled.

Libraries

Install the required Python libraries using `pip` or any other package manager. You can install the required libraries by running the following command in your terminal or command prompt:

```
pip install -r requirements.txt
```

The libraries required are:

- `opencv-python`: OpenCV library for image processing and camera access.
- `numpy`: NumPy library for numerical operations.
- `torch`: PyTorch library for deep learning, required for YOLOv5.
- `torchvision`: TorchVision library, a part of PyTorch, also required for YOLOv5.
- `json5`: JSON5 library for loading credentials from the `secret.json` file.

Note: The `json5` library might not be a standard Python library, so you may need to install it separately.

YOLOv5

The script uses the YOLOv5 model for object detection. The model is loaded using the `torch.hub.load` function. It should be automatically downloaded and loaded when you run the script. However, you can check the official YOLOv5 GitHub repository (https://github.com/ultralytics/yolov5) for more details and updates about the model.

Secret.json

Secret JSON File: The script reads camera credentials from a `secret.json` file. You will need to create this file and provide the necessary information for your cameras. The `secret.json` file should be placed in the same directory as the script. Here's an example of the structure of the `secret.json` file:

```json
{
    "user": "your_camera_username",
    "password": "your_camera_password",
    "ip": "your_camera_ip_address",
    "port": "your_camera_rtsp_port"
}
```

Replace the placeholders with the actual credentials for your cameras.

Once you have met these requirements, you should be able to run the script successfully and observe its functionality in demo mode.

Running the Script

1. meet the requirments above
2. use `python Campy.py` (or something similar depending on your system)
   - note: depending on the last edit you might need to adjust it so it's not in edit more.

for more advanced running techniques read the code

there are different parameters/Args that can be changed when running the code.

Usage

this is used as an addon/extenion to your current securty system.
this can run on a dedicated machine or on your security system machine.

things to do

- [] enable multithreading
- ☑ cleans out old files
- ☑ the current algorithum only looks at how many objects with a specific period of time, but we should be looking both at the objects and their types.

License

See FairSourceLicense.txt