

Introduction:

We have prepared a code on how to perform real-time object detection using YOLOv8 in a virtual environment. YOLO (You Only Look Once) is a popular object detection algorithm known for its speed and accuracy.

Install Required Packages:

- Open a terminal or command prompt.
- Execute the following command to install the required packages
- `pip install ultralytics roboflow opencv-python chardet charset-normalizer`

This command will install the necessary Python packages (ultralytics, roboflow, opencv-python, chardet, and charset-normalizer) for running the project.

Import Ultralytics and Check Installation:

- Open a Python environment (e.g., Jupyter Notebook, Python script).
- Import the `ultralytics` module.
- Run the `ultralytics.checks()` function to verify that the Ultralytics package is installed correctly.

Dataset Creation:

- Use Roboflow to input all the images that was captured by using `import cv2` algorithm.
- After feeding all the images, Roboflow will allocate train, test and validation images based on our requirement.
- It annotates and creates Labels for each folder (train, test and val) and creates a custom dataset for us.
- We can use the custom dataset using the unique API key.

Download Dataset:

- Use the Roboflow API to download the dataset required for the project.
- Initialize a `Roboflow` object with your API key.
- Specify the workspace, project, and version from which to download the dataset.
- Use the `download()` method to download the dataset.

Run Training:

- Execute the YOLO training task.
- Use the `yolo` command with the parameters:

- `task=detect` to specify the detection task.
- `mode=train` to indicate training mode.
- `model=yolov8m.pt` to specify the YOLOv8 model file to use.
- `data=Hat-Detection-1/data.yaml` to specify the data configuration file.
- `epochs=20` to set the number of training epochs.
- `imgsz=640` to specify the input image size.

Run Validation:

- Execute the YOLO validation task.
- Use the `yolo` command with parameters similar to the training task, but with `mode=val` to indicate validation mode.

Run Prediction:

- Execute the YOLO prediction task.
- Use the `yolo` command with parameters similar to the training task, but with `mode=predict` to indicate prediction mode.
- Additionally, set the confidence threshold for predictions using the `conf` parameter.

Visualize Results:

- Import the `YOLO` class from the `ultralytics` module.
- Create an instance of the `YOLO` class with the path to the trained model weights.
- Use the `predict()` method to perform object detection on the specified source (e.g., image, video).
- Specify additional parameters such as `show=True` to display the results and `conf` for the confidence threshold.
- Finally, print or visualize the detection results.

This code demonstrates how YOLOv8 can be used for real-time object detection. By leveraging the power of deep learning and computer vision, we can efficiently detect and classify objects in live video streams, opening numerous possibilities for applications in diverse fields.

GitHub Link :- <https://github.com/Gayathri1295/Hat-Detection>