Setup and Installation

Prerequisites Installation:

- Ensure Python is installed on your system. Python 3.6 or newer is recommended.
- Install necessary Python packages including `ultralytics`, `opencv-python`, `Pillow`, `numpy`, `matplotlib`, `pyyaml`, `requests`, `scipy`, `torch`, `torchvision`, `tqdm`, `psutil`, `py-cpuinfo`, `pandas`, and `seaborn`.

Launch the Notebook:

- Open the "Demo_Application.ipynb" in a Jupyter Notebook environment.

Running the Application

1. **Starting the Application:**
    - Execute the notebook cells in order. The initial cells handle the installation and setup of the necessary libraries.
    - The main application is initialized in the cell containing `if __name__ == "__main__":`. This cell launches a Tkinter window with the YOLOv8 real-time object detection setup.
2. **Using the GUI:**
    - The GUI window titled "YOLOv8 Real-Time Detection (Tkinter)" will appear. This interface displays the video feed from your webcam or a specified video source.
    - Objects detected by YOLOv8 in the video feed are highlighted with bounding boxes. Each box is labeled with the object class and the model's confidence level.

Interpreting the Results

Understanding the Output:

- Rectangles with colors painted around items that have been recognized are called bounding boxes. The size and location of each box match the scale and location of the object that was found in the video stream.
- Labels: Above each bounding box is a label that indicates the object's class (for example, "person" or "car") and the model's level of prediction confidence (for example, "person 0.99").
- Metrics of Performance: The frame rate of the detection, which shows how fast the model is processing the video feed, may be shown in the window by the application. Closing the GUI window is the only way to end the application. Any resources used, including the webcam, will be released when the application ends.