**Specification**

1. **Introduction:**The code provided is an implementation of YOLOv8, an object detection algorithm, using the Ultralytics library. It demonstrates how to perform inference on both images and videos using a pre-trained YOLOv8 model (best09.pt model is used in the code file we submitted.).
2. **Requirements:**The following libraries and resources are required to run the code:

* Ultralytics library
* OpenCV (cv2)
* Pre-trained model file (last.pt)
* Image and video files for testing

1. **Usage:**  
   To identify the treasure pictures taken by the car camera, please follow the following steps:

* Install the Ultralytics library and OpenCV.
* Download the pre-trained YOLOv8 model file (best09.pt) and place it in the same directory as the code.
* Provide the image file for testing. Update the 'img\_path' variable in the code with the file path.
* Run the code.

1. **Image Inference:**The code allows performing YOLOv8 inference on individual images. It uses the 'predict' function of the YOLO model to detect objects in the provided image. The detected objects are then visualized on the image using bounding boxes and class labels. The annotated image is saved with the bounding box overlays.
2. **Configuration:**The code uses the YOLOv8 model loaded from the pre-trained model file (best09.pt). You can modify the code to use a different model or adjust the model's configuration by modifying the 'model' initialization line.
3. **Output:**

An annotated image with bounding box overlays saved in the same directory as the input image.

1. **Conclusion:**The provided code demonstrates how to perform YOLOv8 inference on both images and videos using the Ultralytics library. By following the instructions and modifying the necessary variables, you can use the code to detect objects in images.