Task Instructions: Detecting tiny objects

Objective:

Your task is to develop a computer vision pipeline to train a deep learning model to detect tiny objects, i.e the building class. You are free to use any technique you want including but not limited to image segmentation, object detection using traditional methods or vision transformers. Extra points will be given for ingenuity! Additionally, you have the option to create a user interface (UI) using Streamlit to showcase the model's detections on uploaded images.

Instructions:

1. **Understanding the data:**
    - As mentioned previously, your task is to detect the building class. The data can be found at `https://drive.google.com/file/d/1OS3qGTbYhzoHOPA6PrnkTt1vbc` `view?usp=sharing`
      Note: This data is present in the YOLO annotation format, with the first class being building, and the rest of the classes being labelled from A to F. They are not essential in your task, and it is up to you how you utilize them.

2. **Creating the model:**
    - As mentioned you are free to use any technique you wish, however the thought process should be clearly documented so that the submission can be assessed fairly.
    - You are free to use ChatGPT, claude, or any other AI tool to assist you in your endeavour, however please disclose it. (No points will be deducted for honesty.)
    - Ensure that the annotations you will use are saved, and submitted alongside the code.

3. **Additional instructions:**
    - Implement your model using a deep learning framework such as PyTorch or Tensorflow

- Monitor the training process and adjust hyperparameters as necessary to achieve satisfactory performance.

4. **Optional: Create a UI with Streamlit:**

    - If you choose to implement the optional UI:
        - Use Streamlit, a Python library for building interactive web applications.
        - Develop a simple UI where users can upload an image.
        - Integrate the trained model to perform inference on the uploaded image.
        - Display the detected objects with bounding boxes and labels on the UI.

Submission Requirements:

- Submit all relevant code files, including scripts for image scraping, generated annotations(the data you collected), model training, and UI development (if applicable).
- Ensure that you report the mAP of each class on the test data. In case mAP is not present, make sure there is another relevant metric.
- Include comprehensive instructions on how to run each component of the pipeline.
- Provide any additional documentation or notes that may assist in understanding your solution.
- Ensure all dependencies and libraries required for running the code are clearly specified.

Submission Deadline:

You have one week from the receipt of this task to submit your solution. Please ensure timely submission to be considered for evaluation.

Note:

- Feel free to reach out for clarification or assistance during the completion of the task.
- Focus on delivering a robust and well-documented solution that showcases your skills in image scraping, annotation generation, model training, and optionally, UI development.