Report: Training and Inference of OPG.

* **Training API for OPG dataset:**
* Creating the environment for the API is important locally we can just have it working on a python env but in production deploying on docker instance will require the “Dockerfile” and to run it docker compose file “docker-compose.yaml”.

{

    "yaml\_file\_path": "C:\\Users\\SyedHamzahRayyan\\project\\dataset.yaml",

    "epochs": 10,

    "batch\_size": 4

}

* **Inference API for OPG dataset:**

**Step1:** Creating the environment for the API its important.

**Note:** make sure the python version matches “or” support the packages that are in the flask file. For instance, the current project packages like torch for yolov5 supports python 3.9 instead 3.12 make sure the version are compatible.

**Step2:** All the required packages should be in the same directory where you have the flask api file. In our case we have a “best.pt” file as our trained model which trained on OPG dataset using ultralytics YOLOv5 with 25 epochs.

**Step3:** Import the weights file in working directory.

**Step4:** Creating the python env once created. Active the env.

**Step5:** Run the flask file.

**Step6:** If working on a http browsers make sure it is secure permission are given otherwise the page wont render.

\***To create a env in your** : py -3.9 -m venv myenv39

\***to activate the env** : .\myenv39\Scripts\Activate.ps1

\* **Packages** : pip install torch ultralytics flask pillow

\***TO Run the code** : python app.py