Lylian Cazale 31/12/2024

Internship Candidate Assignment

Probe Detection Using Deep Learning

1. **Chosen Model**

For this task, I've chosen the model YOLO (You Only Look Once). It's a famous model for object detection and classification, but here only detection is interesting for us. I've chosen this model because it's a fast model with great accuracy. It's also easy to use when you know how to make the architecture. It also can easily be fine tuned and propose all the graphs and analysis that you need to understand and adapt more on your objective. It also can be used in real time applications.

2. **Data manipulation**

I started by setting up the architecture for using YOLO. Then, I merged the two provided datasets to have all the necessary information in one place. It was necessary to create .txt files with the bounding box positions and the object class, even though this information is not relevant in our case. The YOLO model also requires that the bounding box information includes the center of the box and that all values are normalized relative to the image size

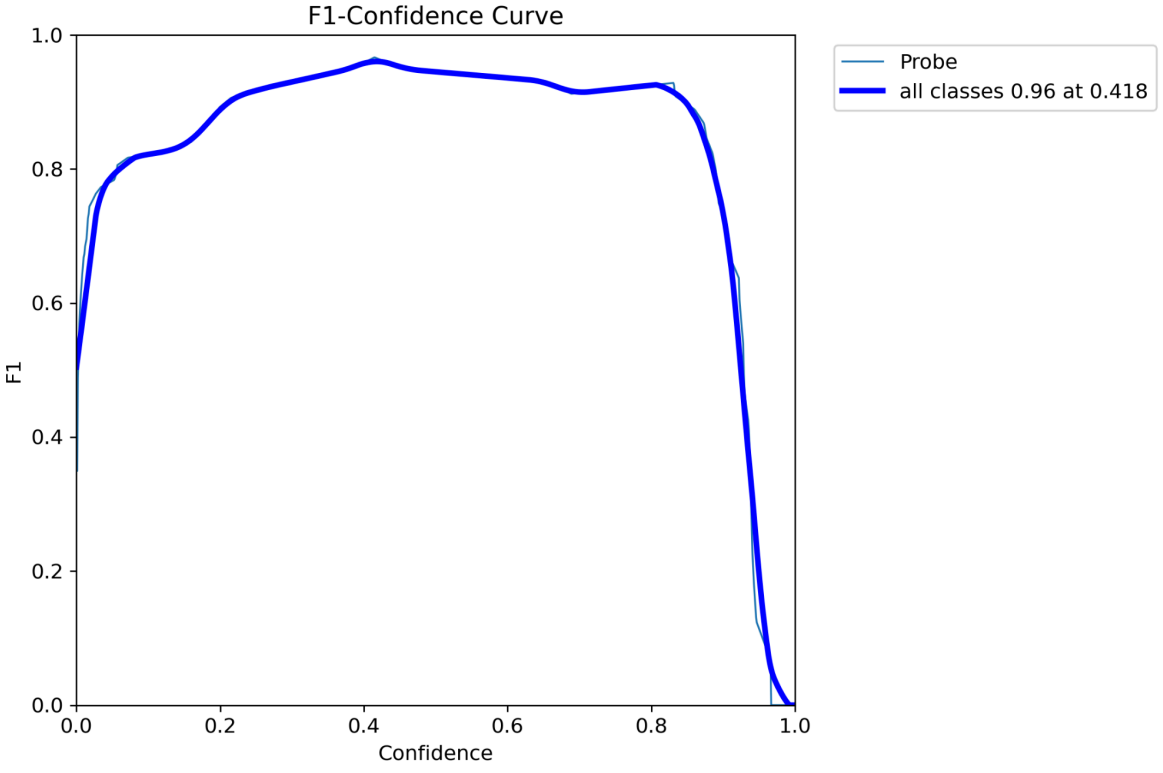
3. **Evaluation**

Each confidence score is displayed around the detected bounding box. All graph information for precision, recall, F1-score, IoU, and loss is available in run/detect/train.

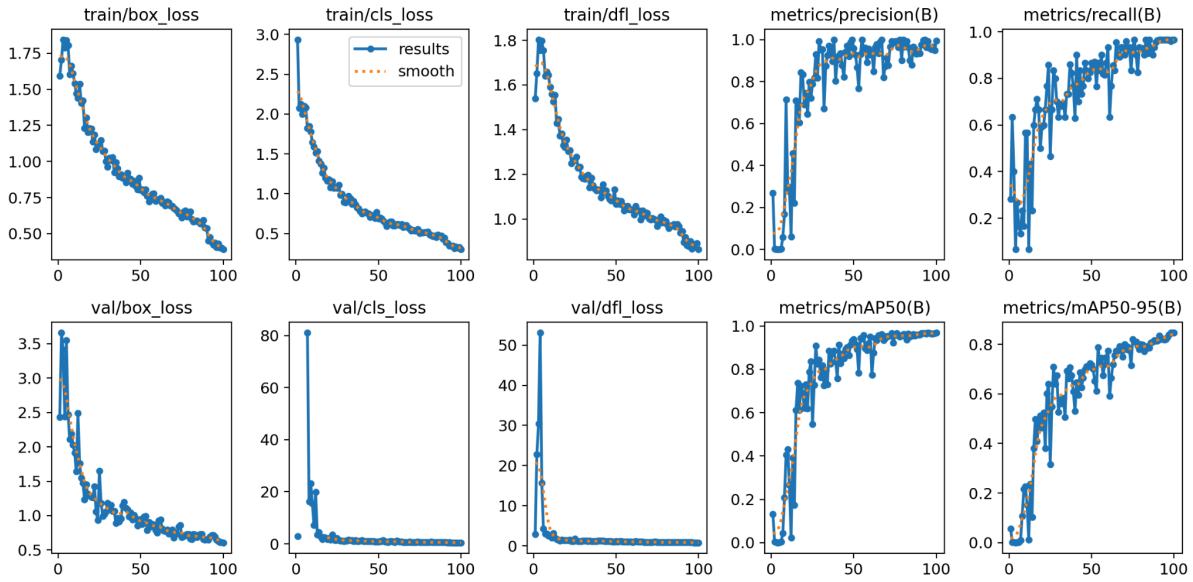
The learning rate is variable, starting at 0.01 and decreasing to 0.001.

The chosen optimizer is AdamW, which enhances convergence and overall performance..

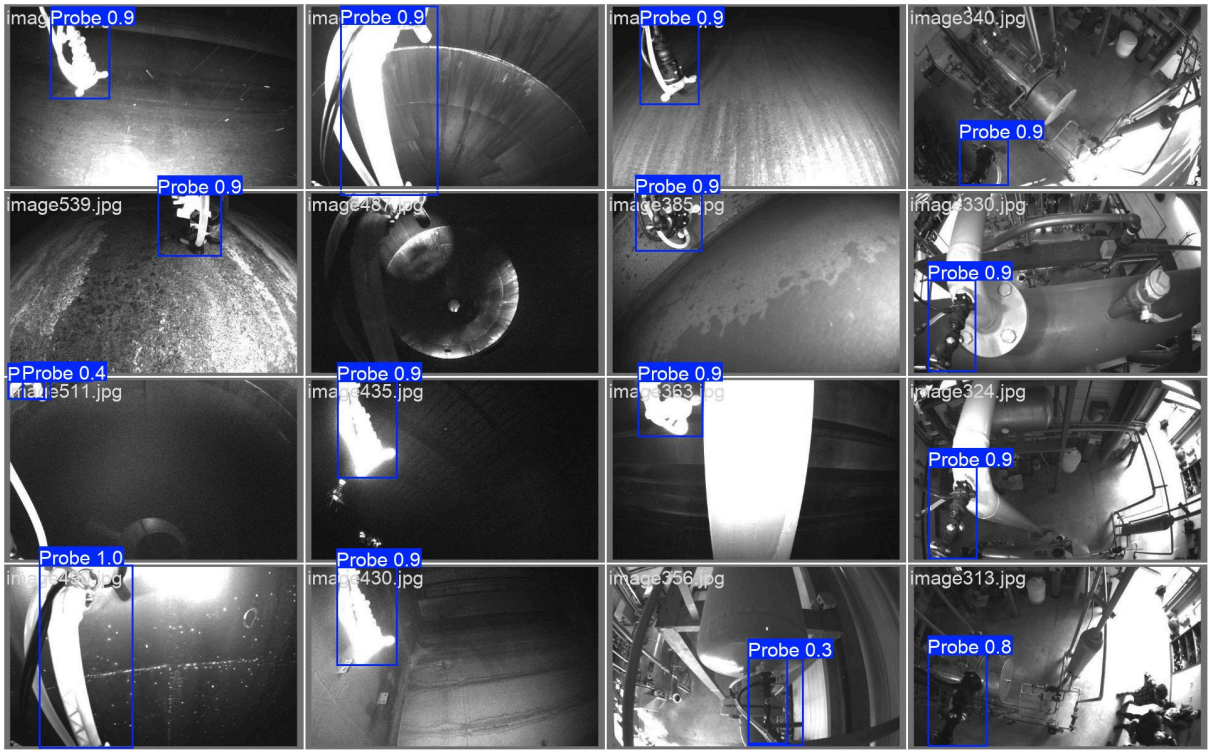
For the YoloV8 Nano model , F1 score during training :



And recall, la precision and loss during epoch



And sur predictions on images :

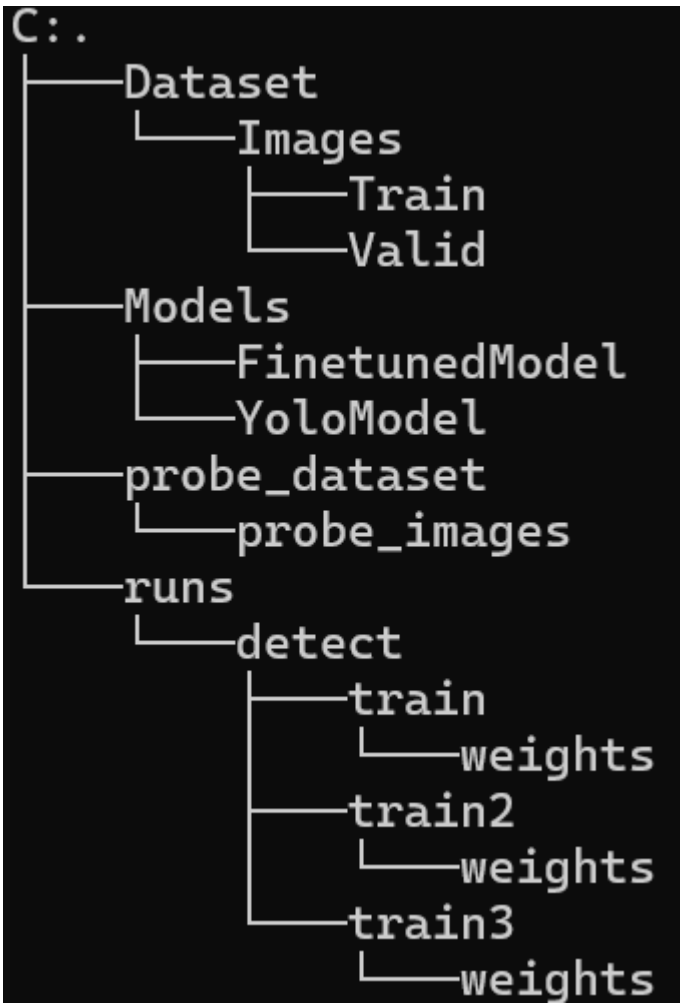


4. **Future Improvements**

Possible improvements could include searching for the best hyperparameters for larger models, which would be more adaptable if we aim to scale up the number of tasks. I attempted to run a larger model, but due to my limited resources, it would have taken too much time. Another option is to preprocess the images to try to eliminate overexposure, which could improve accuracy by reducing noise.

5. **To test my model**

Here is the architecture of the project :



In Dataset folder, all given images splitted in Train/Valid folder (90/10)

In Models folder, the originals YOLOv8 models used and all the fine tuned models

In runs folder, all the information about the model, the weights, the graphs, ….

- train = yolov8 nano 100 epochs
- train2 = yolov8 nano 30 epochs
- train3 = yolov8 medium 50 epochs

Lylian Cazale 31/12/2024

To execute the model :

- open ModelTest.py
- Change the path to the image you want to test
- (Optional) Change the model path to the one wanted
- Execute it
    - You may need to install the libairies needed : ultralytics
        - pip install ultralytics
        - pip uninstall ultralytics
- The result is now saved in the image created