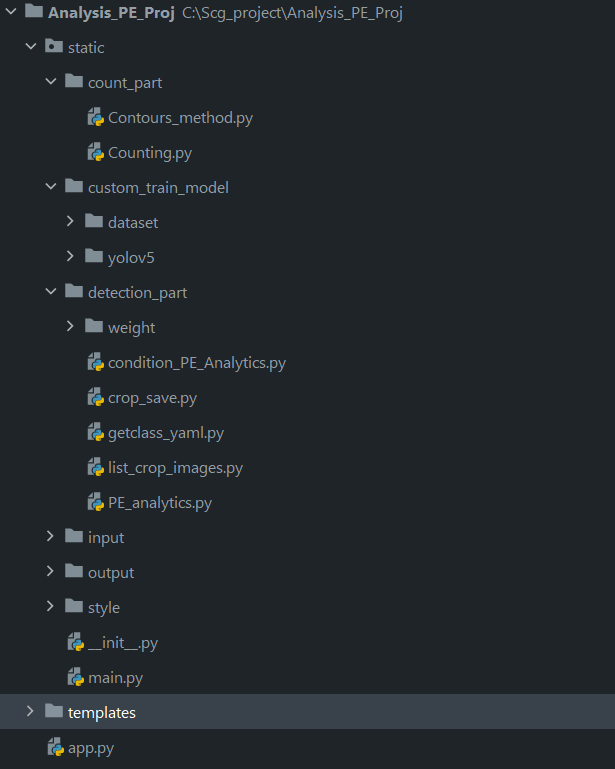
**Project file organize**



API

Back end

**Web app section**

Front end

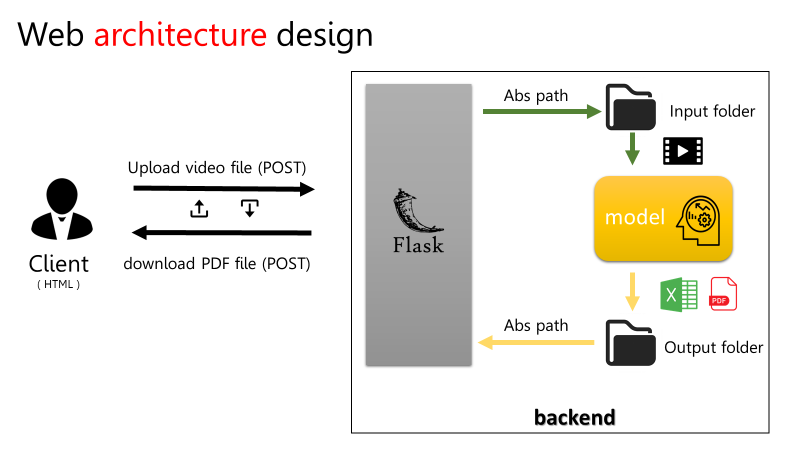
**Flask API section**

**Detection part section**

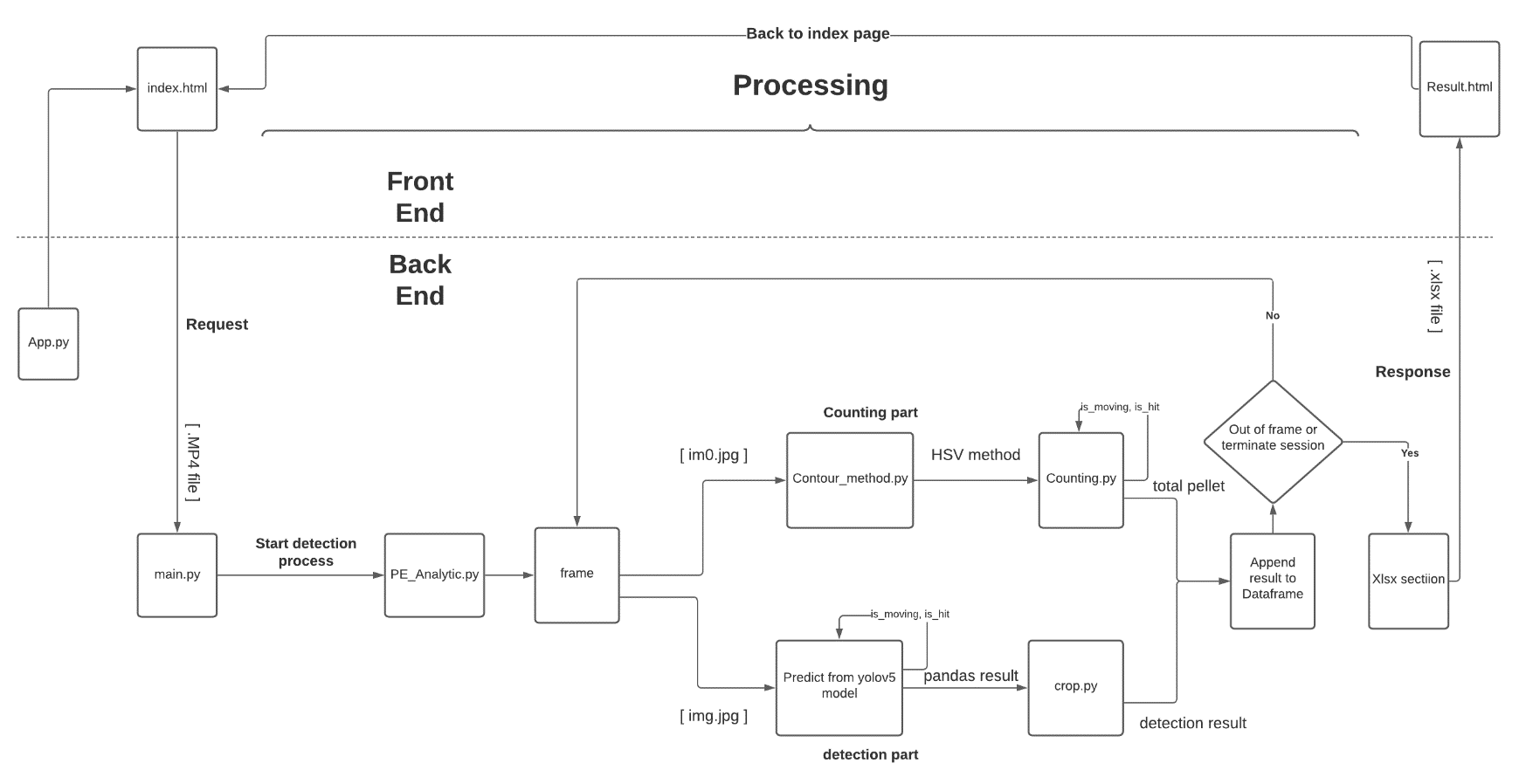
**Local custom training model**

**Counting part section**

**Architecture**

****

**Files relation**



**How to train model (yolov5)**

**Steps**

1. Data Collection
2. Data Labeling
3. Pre-processing images and Data loading
4. model define and parameter configuring
5. train model and evaluate result

\*\*\*Note

* Data Labeling take time and human workload
* Train model need GPU resource and CUDA Libs

1. **Data Collection**

คือ ขั้นตอนในการเก็บรูปเพื่อที่นำไปทำเป็น Dataset โดยวิธีการ

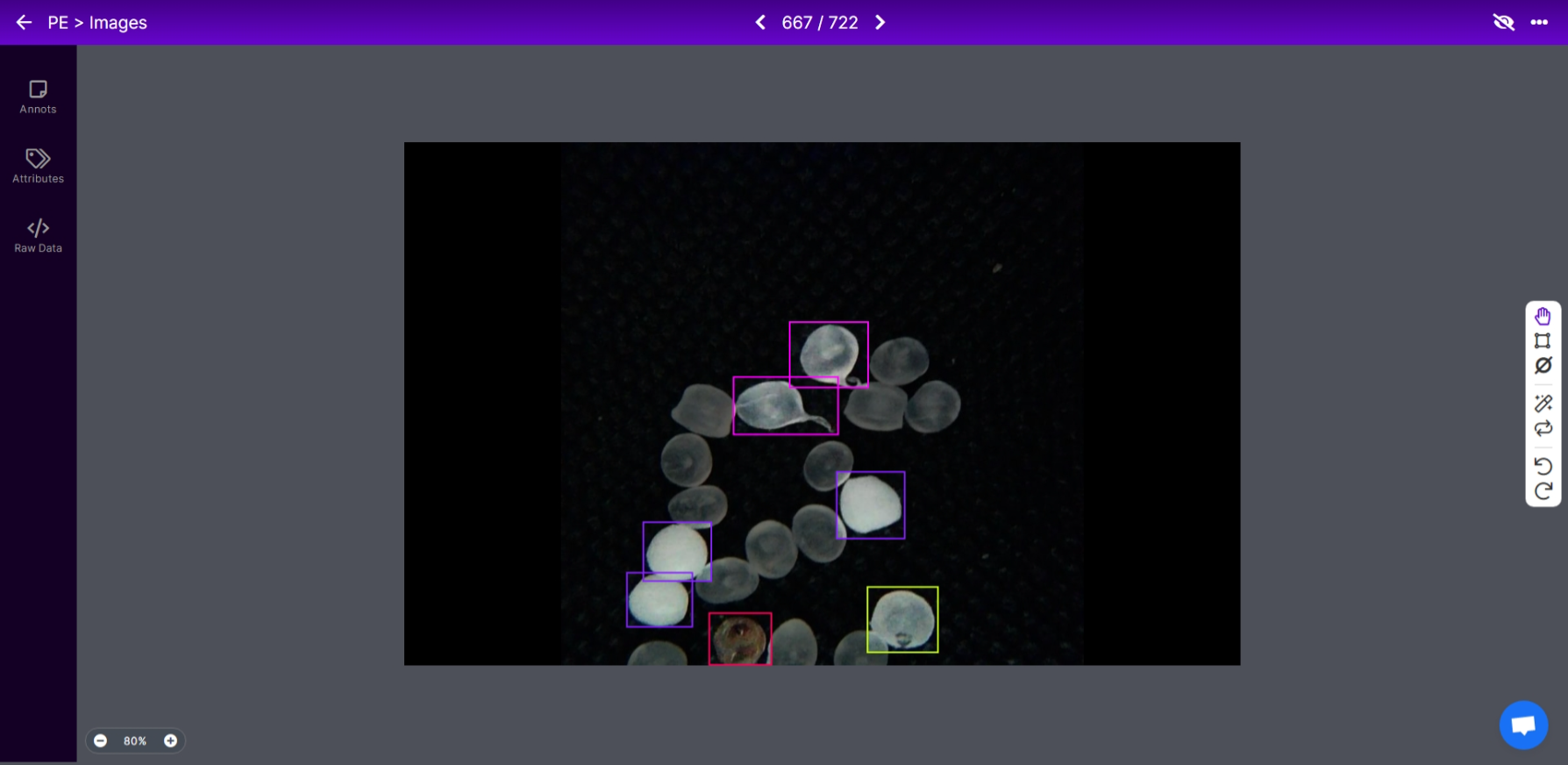
ถ่ายรูปเอง หรือ ไปหาแหล่ง รูป จาก Internet ตัวอย่างเช่น รูปที่ 1.1-1.4

รูปที่ 1.1 รูปที่ 1.2

รูปที่ 1.3 รูปที่ 1.4

1. **Data Labeling**

คือ ขั้นตอนในการระบุตำแหน่งของวัตถุที่เราสนใจภายในภาพทั้งหมด ที่อยู่ใน Dataset ของเรา ซึ่งขั้นตอนนี้ มี tools มากมายให้เราเลือกใช้งาน ตัวผมเลือกใช้งาน Roboflow (รูปที่ 2.1) ซึ่งเป็น web app สำหรับการทำ Dataset โดยเฉพาะ

Ref. [Sign in to Roboflow](https://app.roboflow.com/login)

รูปที่ 2.1

**Result**

* bounding box (location)
* width, height, co-ordinate point
* classes of object

\*\*\*Note

* ในขั้นตอนนี้ค่อนข้างที่จะกินเวลาในการทำงานมาก
* การ label box ให้ object นั้นส่งผลต่อ Accuracy ของ model ดังนั้น จึงควรทำอย่างระมัดระวัง

1. **Pre-processing images and Data loading**

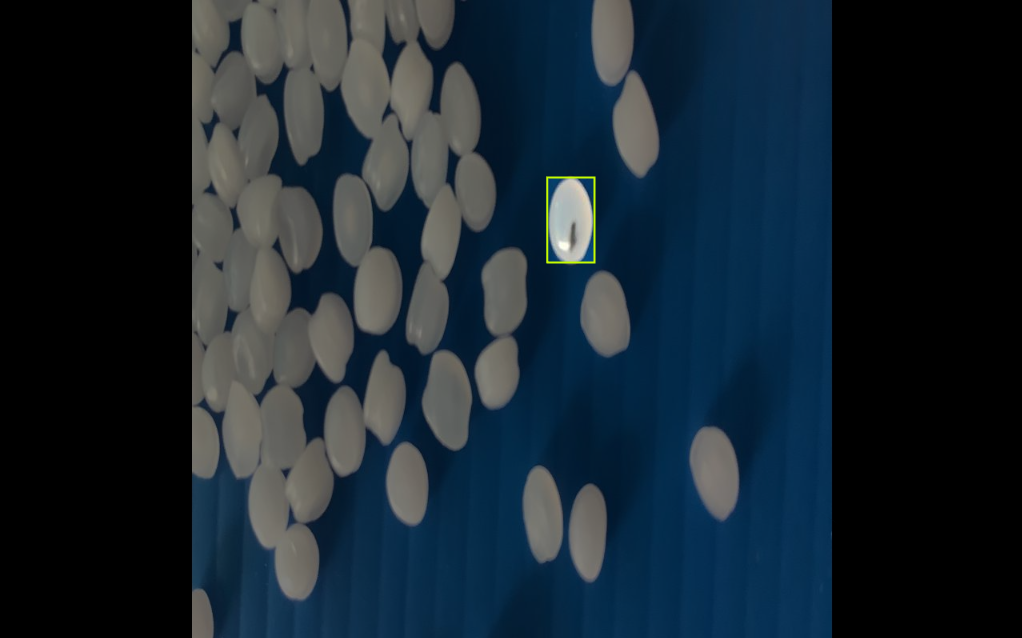
คือ ขั้นตอนในการจัดการเตรียม dataset ให้พร้อมใช้งาน

และ import เข้าไปใน model

* Pre-processing images

การทำอะไรบางอย่างกับ dataset เพื่อที่จะทำให้ model ของเรา train ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น เช่น การ data Augmentation คือ การจำลองรูปภาพจากรูปที่เรามีอยู่ขึ้นมาด้วยการ edit รูป ด้วยวิธีการต่าง ๆ

เช่น Crop ,Gray scale Filter, Flip ดังรูปที่ 3.1 – 3.2



รูปที่ 3.1 รูปที่ 3.2

* Data loading

การนำ Dataset ที่ผ่านการทำ pre-processing และ พร้อมที่จะ

ถูกนำไป train model ของเรามาเตรียมรอไว้ที่ folder

1. **Model define and parameter configuring**

คือ ขั้นตอนในการประกาศ model และ ปรับค่า parameter ต่าง ๆ เพื่อใช้ในการ training model เช่น

* batch size
* epochs
* learning rate
* image sizes
* cache
* IOU threshold
* Source of datasets
* model weight ex. yolov5s6 ,yolov5l6

1. **Train model and Evaluate result**

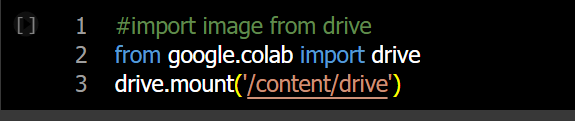
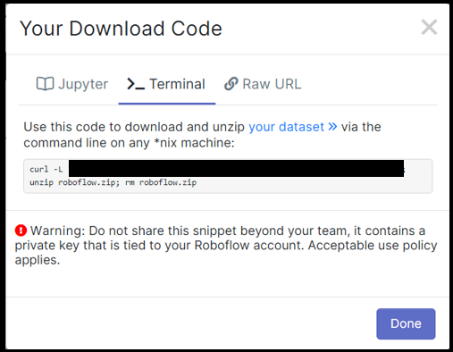
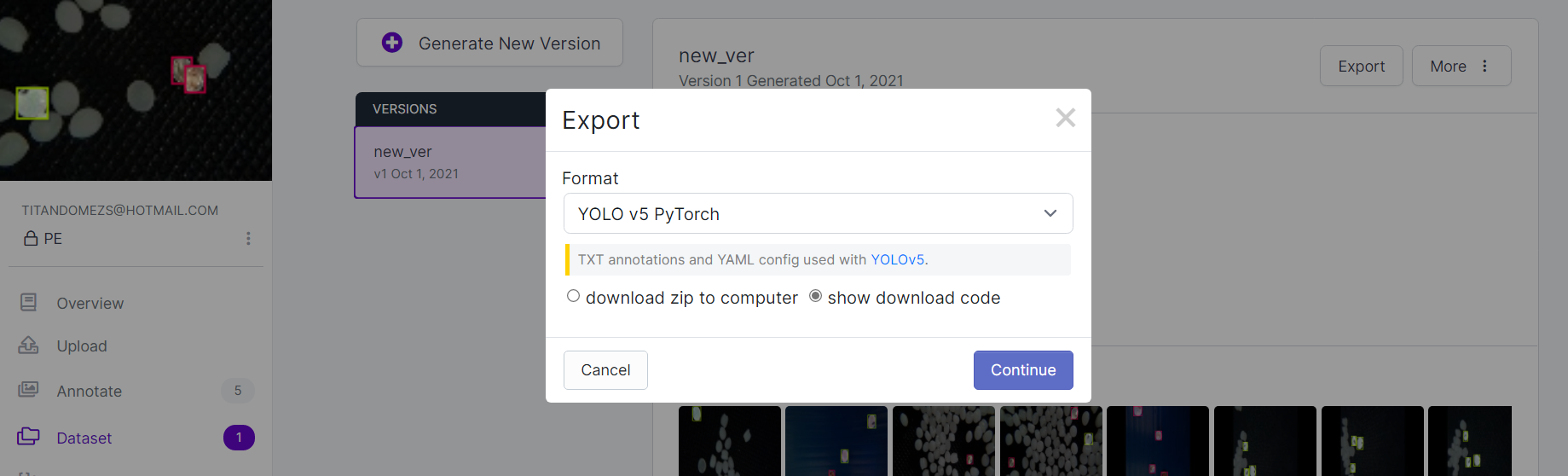
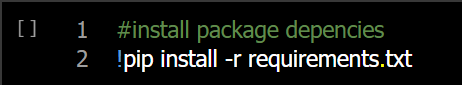
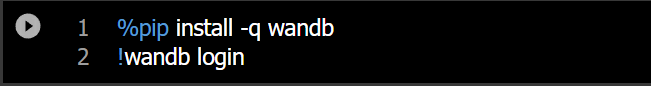
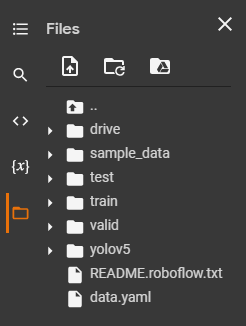
เป็นขั้นตอนในการ train model และ ประเมินผลลัพธ์ หลังจากที่ได้ model มาจากการ train เพื่อที่จะเลือกว่าจะนำ model นี้ไปใช้หรือไม่ โดยประเมินจากค่าต่าง ๆ ดังนี้

* mAP 0.5
* mAP 0.5:0.95
* Recall
* Precision
* Object loss
* box loss
* Class loss
* Confusion matrix

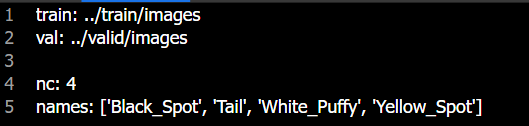
ซึ่งแต่ละค่าก็จะบอกประสิทธิของ model เราได้ในแต่แบบที่ต่างกัน

**Implementation steps**

1. Set up Environment

* mount drive (google colab)
* Export dataset from roboflow and Import to code
* Clone yolov5 model
* install package requirement
* wandb login for model evaluate (optional)
* Your directory will be look like

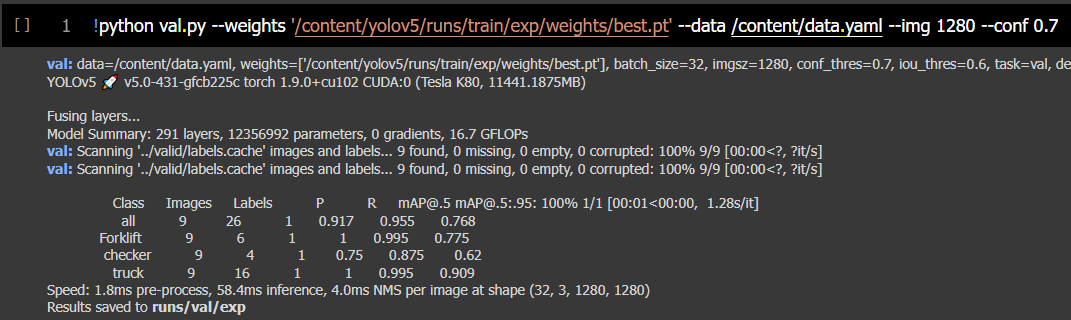
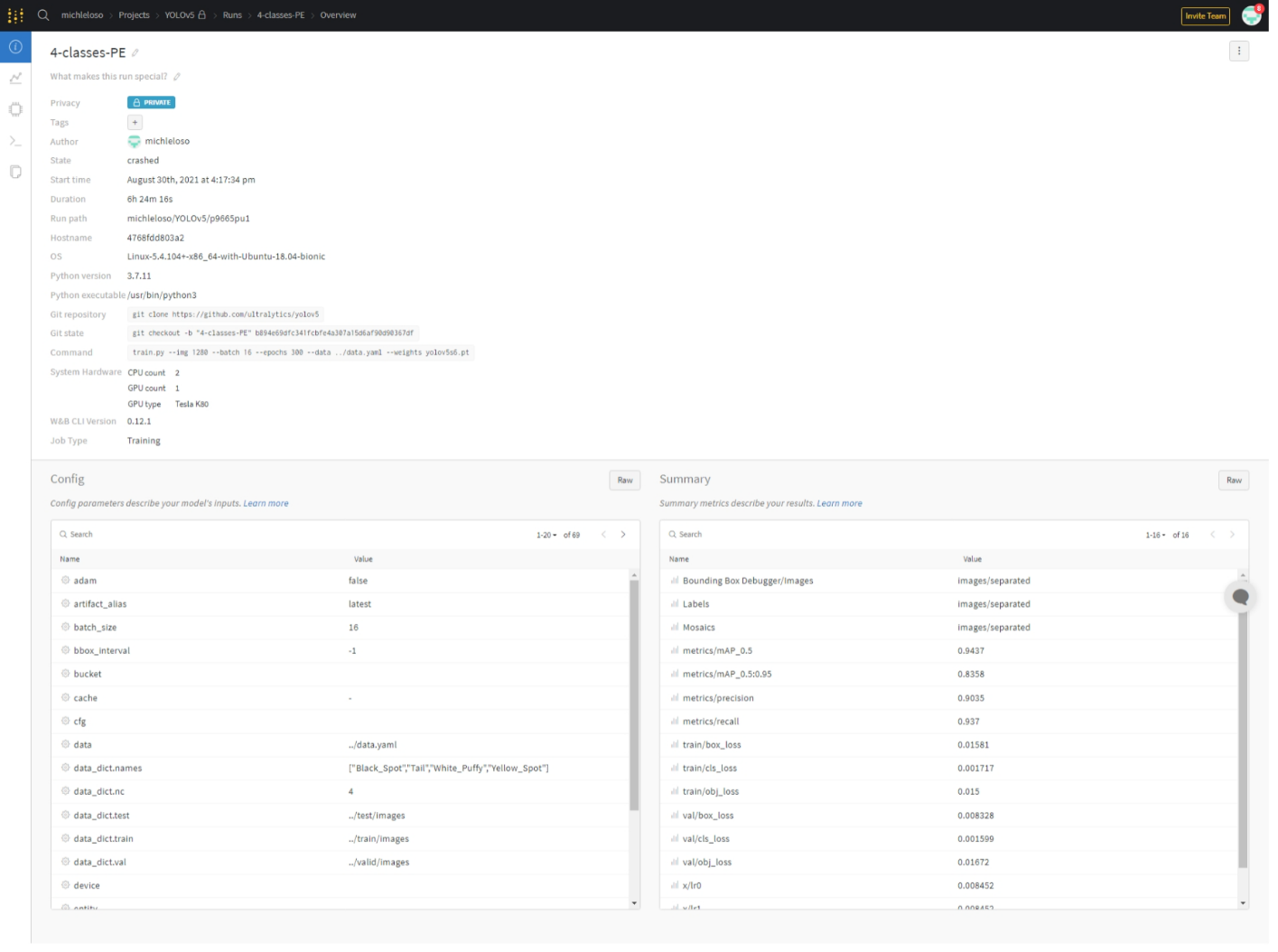
1. Configure dataset path

* go to file data.yaml
* edit path as your dataset เพื่อที่จะ check ว่า path และ class ทั้งหมดที่เราสนใจถูกต้องหรือไม่

1. Training model (take long time)

เราต้องเข้าไปที่ terminal แล้ว cd ไปที่ directory yolov5 จากนั้น run command ด้านล่าง เพื่อที่จะ train model ของเรา

1. validating model (model evaluate)

* train model result on editor
* The link in below lead you to wandb training performance visualization

1. Save weight to use later

เมื่อ train model สำเร็จแล้วโดยปกติ จะมี directory **yolov5/runs/train/exp/weight/best.pt** ขึ้นมา ซึ่งตัว best.pt คือ   
file weight ที่เราจะนำไปใช้งาน

* google colab (path of weight, path to save)