**Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM**

**Khoa Công nghệ Thông tin**

**---o0o---**



**Final Project: YOLO Object Detection**

**Môn: Học Thống Kê**

**Giảng viên lý thuyết: - Ngô Minh Nhựt**

**Giảng viên thực hành:**

* **Phan Thị Phương Uyên**
* **Lê Long Quốc**

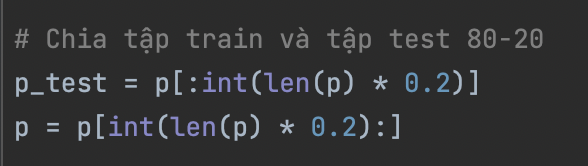
**Sinh Viên:**

* **Nguyễn Trung Hiếu – 19127403**
* **Trần Xuân Phước - 19127516**

1. **Nhiệm vụ.**

|  |  |  |
| --- | --- | --- |
| Task | Người thực hiện | Mức độ hoàn thành |
| 1. Tìm hiểu YOLOv3 | **Nguyễn Trung Hiếu**  **Trần Xuân Phước** | **100%** |
| 2. Tìm hiểu cách train YOLOv3 với custom data. | **Nguyễn Trung Hiếu** | **100%** |
| 3. Tìm hiểu cách dự đoán vật thể dựa trên model có sẵn và viết chức năng. | **Nguyễn Trung Hiếu** | **100%** |
| 4. Tìm dataset đã được thu thập sẵn hoặc tự sử dụng LabelImg để tạo dataset riêng. | **Nguyễn Trung Hiếu** | **100%** |
| 5. Tìm hiểu cách model train và ứng dụng train ra model. | **Nguyễn Trung Hiếu** | **100%** |
| 6. Viết ứng dụng web cơ bản với chức năng dự đoán vật thể trên model có sẵn | **Trần Xuân Phước** | **100%** |
| 7. Viết ứng dụng web cơ bản với chức năng dự đoán vật thể trên model tự train. | **Trần Xuân Phước** | **100%** |
| 8. Viết report. | **Nguyễn Trung Hiếu**  **Trần Xuân Phước** | **100%** |

1. **Model với custom data và hướng dẫn train model.**

Model được train dữ liệu cờ vua thi đấu tại USA với 231 hình ảnh được sử dụng để train và val. Được chia với tỉ lệ 80/20.

**Bước 1: Tải source Darknet về máy**

Gõ lệnh vào cmd để clone : git **clone** https://github.com/pjreddie/darknet

Có thể cài đặt GPU=1 nếu muốn xài GPU, OPENCV=1 nếu muốn xài thư viện openCV trong file MakeFile của darknet.

Sau đó gõ lệnh: Make

**Bước 2: Chuẩn bị dữ liệu train.**

Với custom data cờ vua được lấy từ:

<https://public.roboflow.com/object-detection/chess-full/24>

Với toàn bộ dữ liệu tải xuống tách thành 2 file là file **images** và **labels** và để trong thư mục **darknet/data/.** Nếu trong thư mục đã có sẵn 2 file **images** và **labels** thì ta xóa toàn bộ file trong thư mục đó.

**Bước 3: Chuẩn bị các file cần thiết cho việc train.**

6 file cần làm việc đến:

custom\_data.data

classes.name

train.txt

val.txt

darknet53.conv.74

Việc xử lý từng file thực hiện từng bước như sau:

Với file **darknet53.conv.74** được lấy từ <https://pjreddie.com/media/files/darknet53.conv.74> và trong thư mục **darknet/**

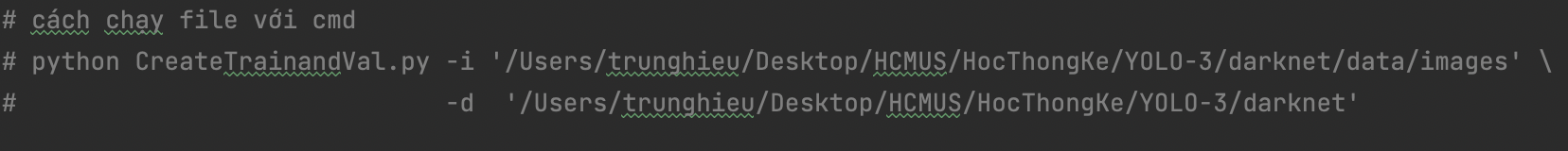
Ta cần tạo một file **classes.txt** trong thư mục **darknet/data/images/** với mỗi dòng là một class ta cần cho model phát hiện.

Với file **train.txt** và **val.txt** ta chạy file **CreateTrainandVal.py** với cửa sổ dòng lệnh cùng với 2 tham số truyền vào là:

--images: địa chỉ tuyệt đối của thư mục images.

--darknet: địa chỉ tuyệt đối của thư mục darknet.

Vd:

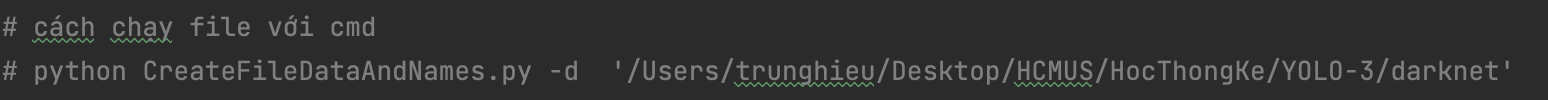


Với file **custom\_data.data** và **classes.names** ta chạy file **CreateFileDataAndNames.py** với cửa sổ dòng lệnh cùng với 2 tham số cần được truyền vào là:

--images : địa chỉ tuyệt đối của thư mục images.

--darknet : địa chỉ tuyệt đối của thư mục darknet.

Vd:



Tiếp theo ta cần chỉnh sửa lại file yolov3.cfg (trong thư mục /darknet/cfg/) sửa các dòng sao cho phù hợp với bộ dữ liệu cần được train.

* Dòng 7, nếu máy bạn nhiều RAM/GPU VRAM thì để nguyên, còn không thì sửa số 8 thành 16 nhé.
* Dòng 603, sửa lại thành filters=18 (số 18 tính bằng các lấy số (class + 5)\*3 nhé. Ví dụ sau bạn train có 2 class thì sẽ là (2+5)\*3 = 21.
* Dòng 610, sửa lại thành classses=1 (hoặc bằng số nào đó khác nếu bạn train nhiều hơn 1 class).
* Dòng 689 và 776, sửa giống dòng 603.
* Dòng 696 và 783, sửa giống dòng 610.

Tùy chọn: Với việc huấn luyện trên google colab ta cần phải set lại giá trị phù hợp lưu lại file weights sau bao nhiêu vòng lặp bằng cách thay đổi file detector trong thư mục darknet/examples/

Ở dòng 138 và sửa lại dòng theo ý muốn:

if(i%2000==0 || (i < 1000 && i%100 == 0)){

Sau đó ta cần nén thư mục **darknet** lại thành file zip. Ta vào drive của google và tạo thư mục có tên tùy ý và đưa file zip lên đó và tạo thêm thư mục backup.

Graphical user interface

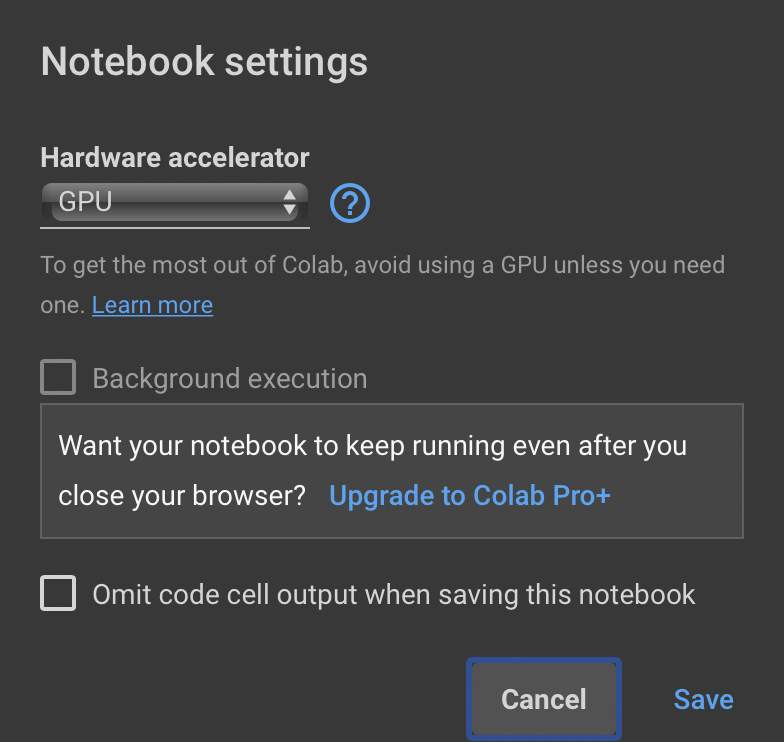
Description automatically generated

**Bước 4: Train mô hình với Google colab.**

Ta đưa file **YOLOv3Train.ipynb google colab** và Change runtime type lại thành GPU

Graphical user interface

Description automatically generated



Sau đó ta chạy từng lệnh trong file để tiến hành train model.

Lưu ý: với dòng lệnh cuối

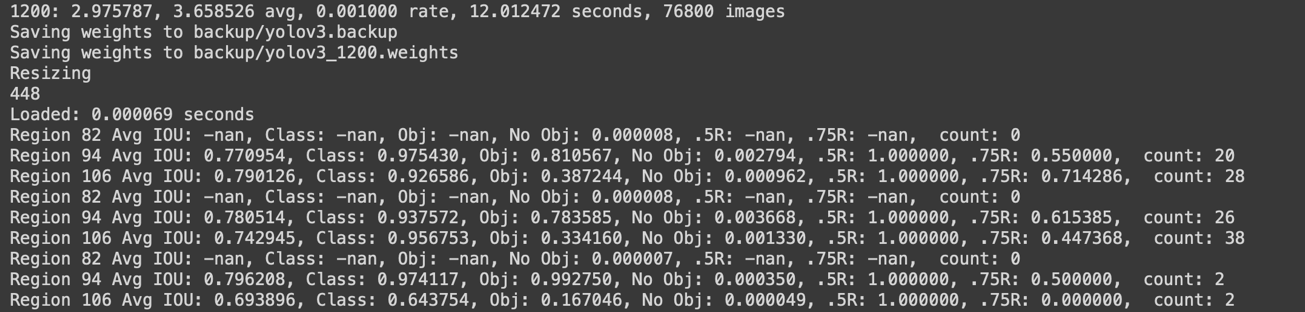
%cd /content/darknet

!./darknet detector train custom\_data.data cfg/yolov3.cfg backup/yolov3\_1200.weights

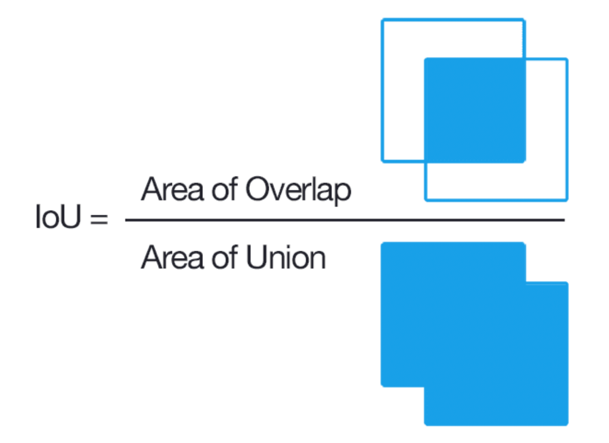
Tham số cuối truyền vào nếu ta mới bắt đầu train mô hình thì ta xài file **darknet53.conv.74.** Sau bao nhiêu vòng lặp ta tự cài đặt trong file detector mô hình sẽ save lại file weights và lưu lại trên thư mục backup trên drive. Lý do ta cần làm điều này là bởi vì google colab reset lại thư mục sau mỗi 10 tiếng và ta không thể lấy file về.

1. **Đánh giá model với custom data.**

Do hiện tại nhóm chỉ có thể mượn GPU của google colab và chạy được 1200 epoch nên ta sẽ đánh giá mô hình trên file weights\_1200

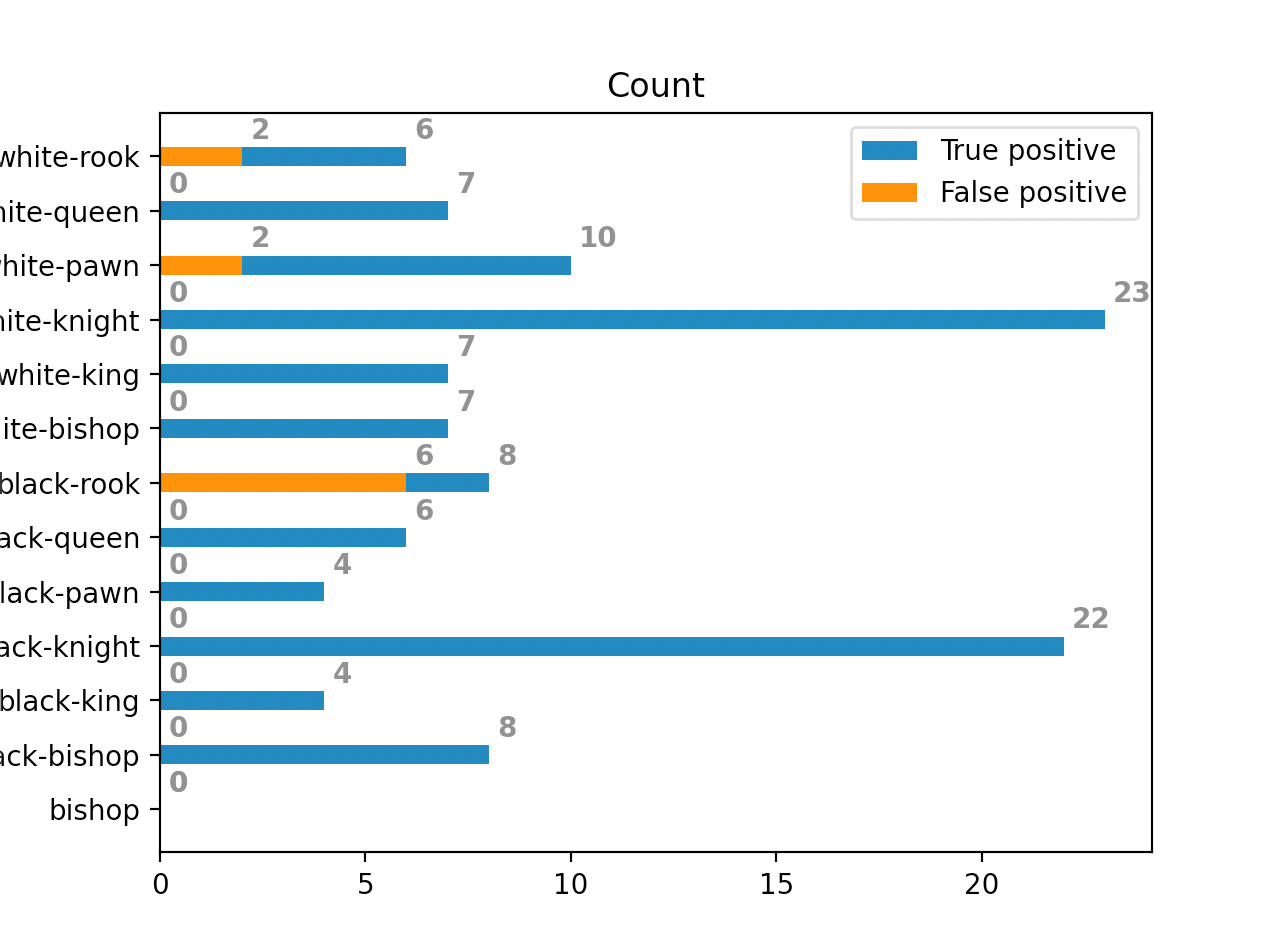
Tại vòng lặp 1200 thì mô hình ta có độ lỗi trung bình được giảm đi từ những vòng lặp trước đáng kể với chỉ 3.65826. Cùng với learning rate 0.01

Tiếp theo là IoU (Intersection over Union)





Được tính từ file **Evaluation.py,** với IoUtrung bình trên mọi hình ảnh từ tập train và tập test đạt đến 0.779 độ khớp. Cùng với các chỉ số True positive và False positive đối với mỗi class.

****

Kết quả cho ra rất khả quan khi mô hình dự đoán khá tốt cùng với IoU khá ăn khớp so với thực tế. Tuy nhiên ta vẫn còn thấy số lượng phát hiện vật thể vẫn còn ít so với 231 tấm ảnh, đó là bởi vì việc đánh giá chỉ dựa trên những tấm ảnh đã được dự đoán đúng với số lượng object có thể được phát hiện trong hình và nhưng tấm hình không phát hiện được số lượng object bằng với thực tế thì bỏ qua. Điều này có thể khắc phục bằng việc cho train tiếp với số lượng epoch lớn hơn. Với việc bỏ qua những tấm ảnh không hoàn hảo sẽ không thực sự ảnh hưởng đến kết quả thực tế bởi vì model vẫn phát hiện các vật thể rất tốt.

1. **Tài liệu tham khảo.**

<https://blog.paperspace.com/mean-average-precision/>

<https://www.miai.vn/2020/05/25/yolo-series-train-yolo-v4-train-tren-colab-chi-tiet-va-day-du-a-z/>

<https://miai.vn/2019/08/15/yolo-series-cach-train-yolo-tren-google-colab/>

<https://miai.vn/2019/08/09/yolo-series-2-cach-train-yolo-de-detect-cac-object-dac-thu/>