****

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**BÁO CÁO**

**Lab 04**

Xử lý ảnh số và video số 20\_23

*Giảng viên hướng dẫn – Nguyễn Mạnh Hùng*

Thành phố Hồ Chí Minh - 2022

**MỤC LỤC**

[**THÔNG TIN SINH VIÊN** 3](#_Toc121428342)

[**I.** **Thực nghiệm với mã nguồn mẫu và bộ dữ liệu được cung cấp** 4](#_Toc121428343)

[**II.** **Thực nghiệm với mã nguồn mẫu được cung cấp, thay thế bằng bộ dữ liệu khác** 9](#_Toc121428344)

[a) Thực nghiệm với bộ trọng số yolov4.conv.137.pth 9](#_Toc121428345)

[b) Thực nghiệm với bộ trọng số yolov4.pth (tương tự như thực nghiệm ở trên chỉ là thay đổi bộ trọng số) 13](#_Toc121428346)

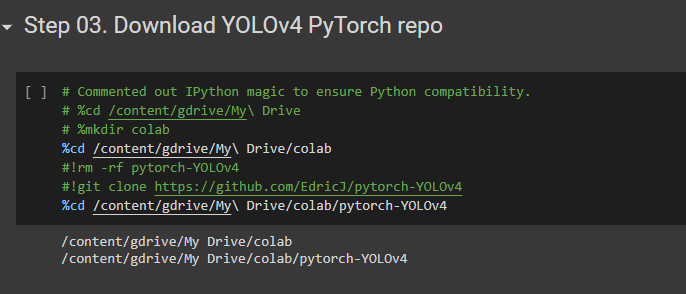
[**III.** **Thực nghiệm với model khác, mã nguồn khác, chạy cùng trên các bộ dữ liệu** 20](#_Toc121428347)

[**TÀI LIỆU THAM KHẢO** 21](#_Toc121428348)

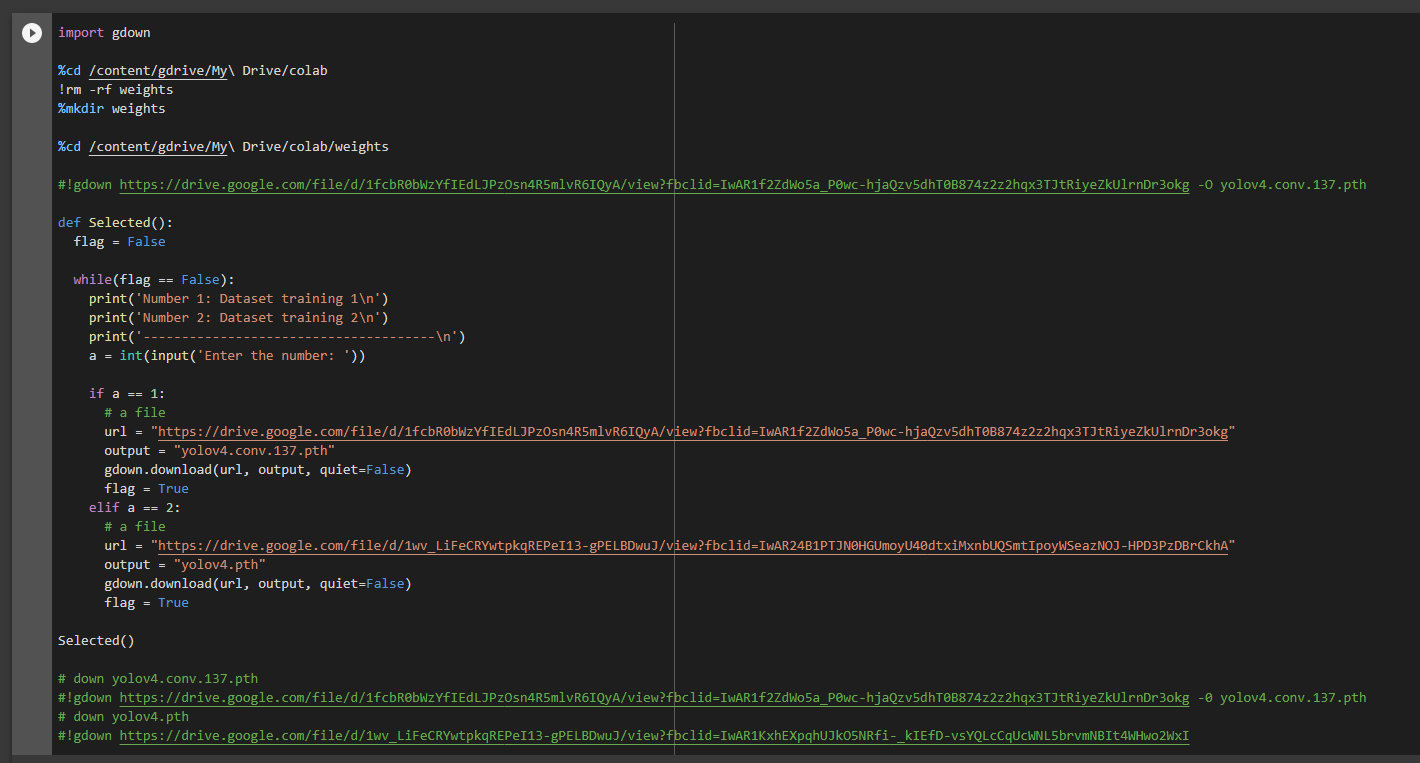
# **THÔNG TIN SINH VIÊN**

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ Tên | Email | Ghi chú |
| 20120201 | Phạm Gia Thông | 20120201@student.hcmus.edu.vn |  |

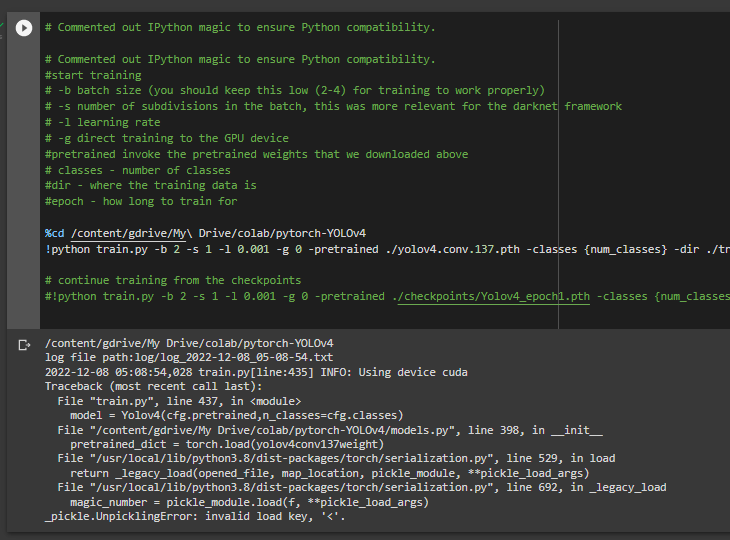
1. **Thực nghiệm với mã nguồn mẫu và bộ dữ liệu được cung cấp**

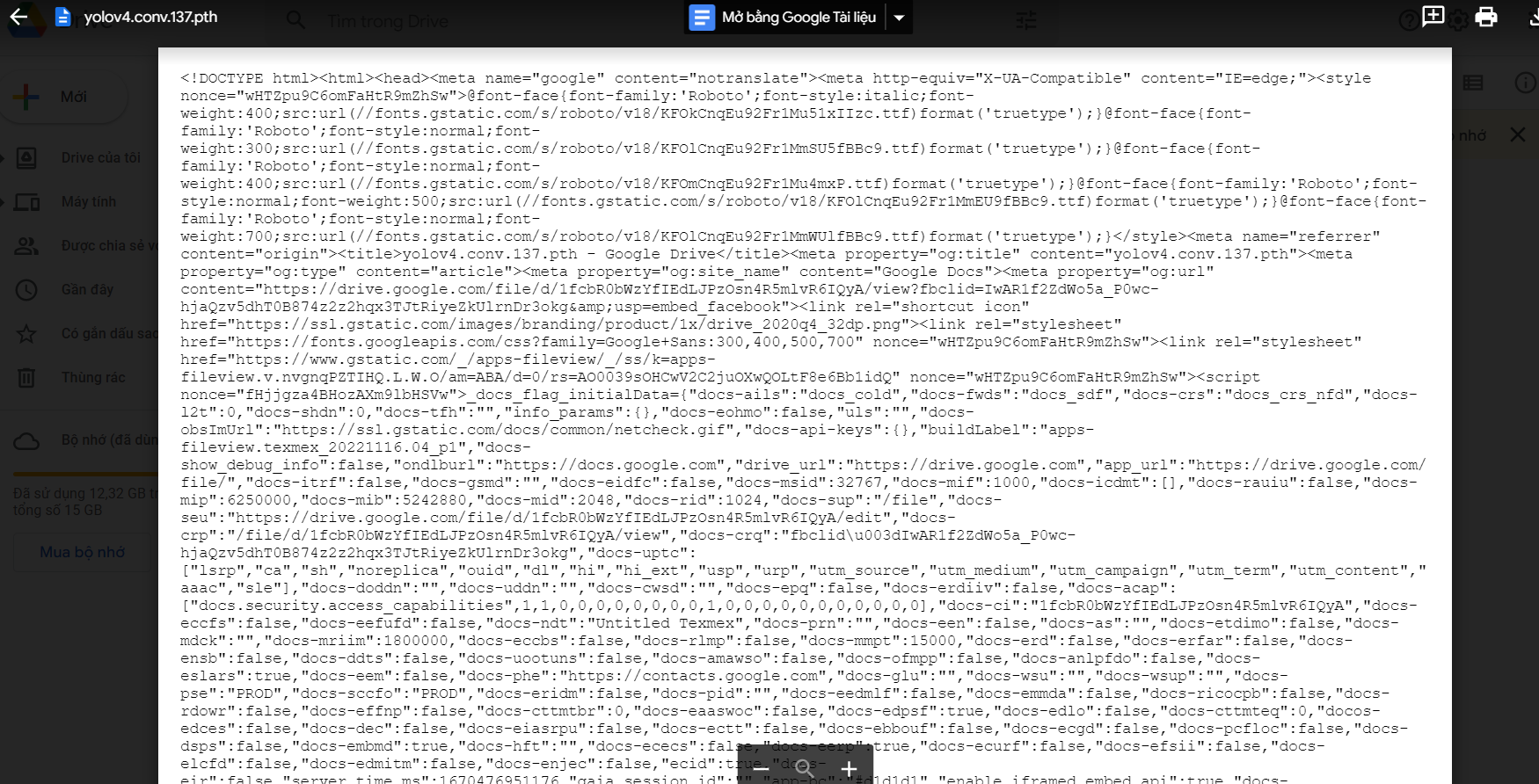
****Tải mã nguồn YOLOv4-PyTorch từ github

Link: [*https://github.com/EdricJ/pytorch-YOLOv4*](https://github.com/EdricJ/pytorch-YOLOv4)

****Tải 1 trong 2 tập tin bộ trọng số đã được huấn luyện (pre-trained weights)

Tuy nhiên việc gdown dữ liệu từ drive về sẽ làm type of file bị thay đổi dẫn đến việc training sẽ bị lỗi, có thể tải thủ công về và upload vào không gian làm việc mà google colab đang tương tác





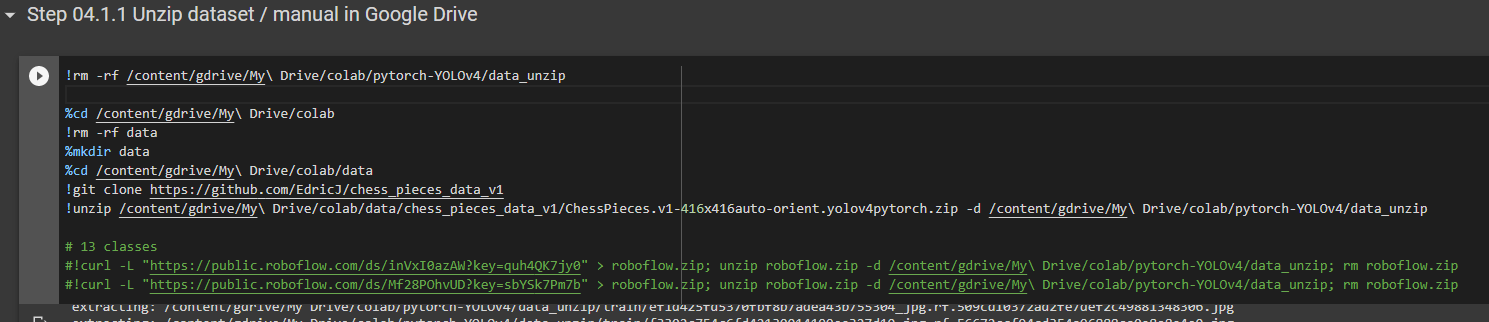
Link weights:

google

* yolov4.pth([*https://drive.google.com/file/d/1wv\_LiFeCRYwtpkqREPeI13-gPELBDwuJ/view?fbclid=IwAR3NhtOgnoQZoifRS08LlYvE1Tdh52QHd0hb-VRql24IDahhoS1G8kOC4eo*](https://drive.google.com/file/d/1wv_LiFeCRYwtpkqREPeI13-gPELBDwuJ/view?fbclid=IwAR3NhtOgnoQZoifRS08LlYvE1Tdh52QHd0hb-VRql24IDahhoS1G8kOC4eo))
* yolov4.conv.137.pth([*https://drive.google.com/file/d/1fcbR0bWzYfIEdLJPzOsn4R5mlvR6IQyA/view?fbclid=IwAR0tsqk8lMMEqPuOETBHxHvWATSSd-4jWjlcUcUaFpqSdpBYYxUNwwQg0YE*](https://drive.google.com/file/d/1fcbR0bWzYfIEdLJPzOsn4R5mlvR6IQyA/view?fbclid=IwAR0tsqk8lMMEqPuOETBHxHvWATSSd-4jWjlcUcUaFpqSdpBYYxUNwwQg0YE))

baidu

* yolov4.pth([*https://pan.baidu.com/s/1ZroDvoGScDgtE1ja\_QqJVw*](https://pan.baidu.com/s/1ZroDvoGScDgtE1ja_QqJVw) Extraction code:xrq9)
* yolov4.conv.137.pth([*https://pan.baidu.com/share/init?surl=ovBie4YyVQQoUrC3AY0joA*](https://pan.baidu.com/share/init?surl=ovBie4YyVQQoUrC3AY0joA) Extraction code:kcel)

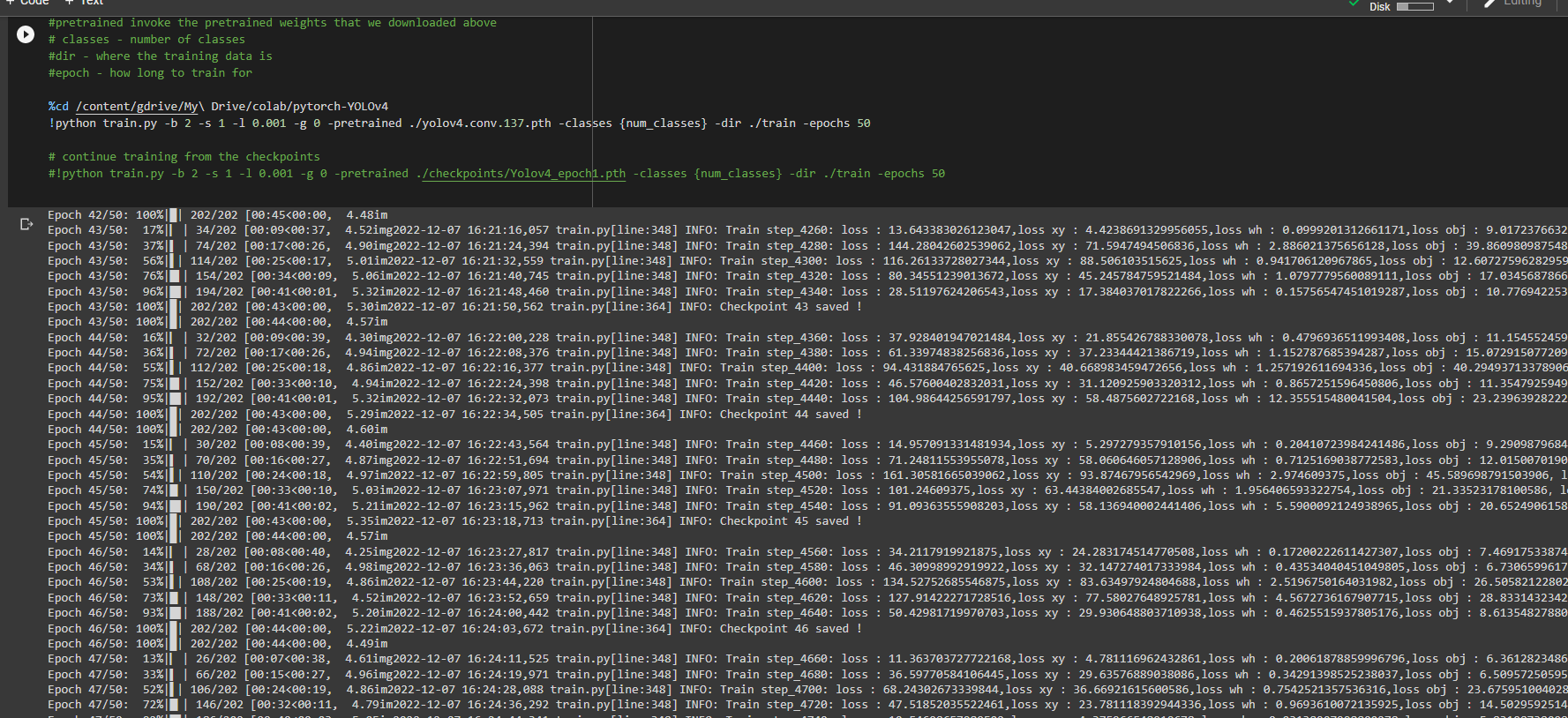
Tải 1 trong 3 bộ dữ liệu dataset (chess object detection)

Link:

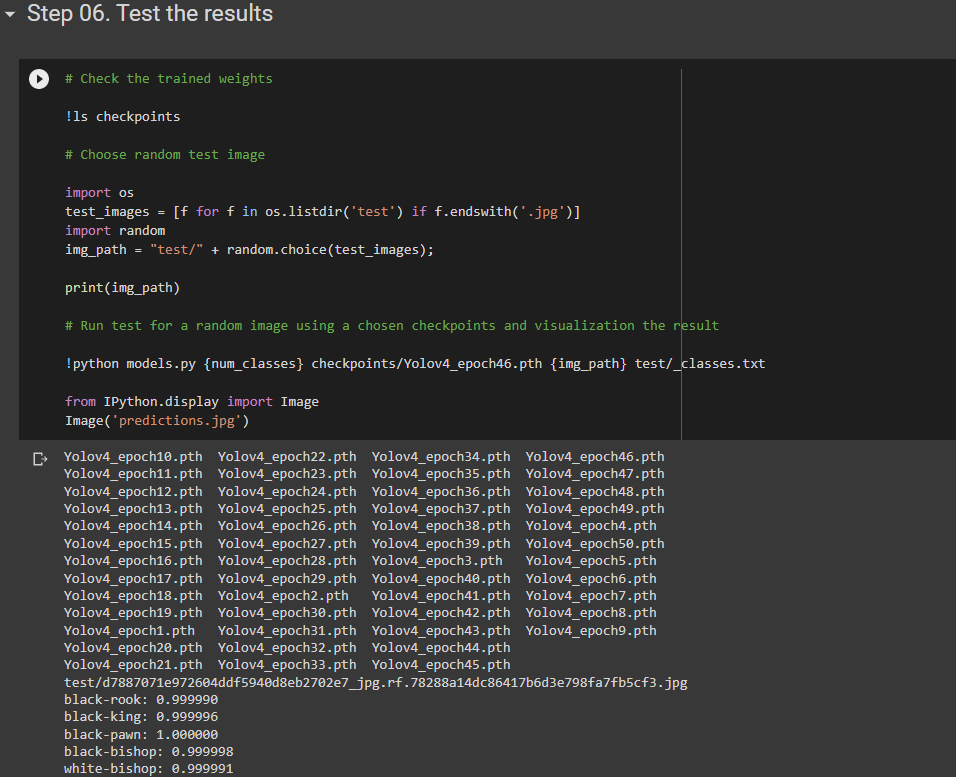
Data\_1: [*https://github.com/EdricJ/chess\_pieces\_data\_v1*](https://github.com/EdricJ/chess_pieces_data_v1)

Data\_2: [*https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0*](https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0)

Data\_3: [*https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b*](https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b)

Bắt đầu quá trình training với epoch là 50 (~35 – 40mins), ở đây chỉ training một lần, không training thêm từ file checkpoints để tiết kiệm thời gian chạy ở những bộ dữ liệu khác

Kiểm tra kết quả từ file checkpoints ở lớp epoch thứ 46, có thể kiểm tra ở các lớp epoch khác

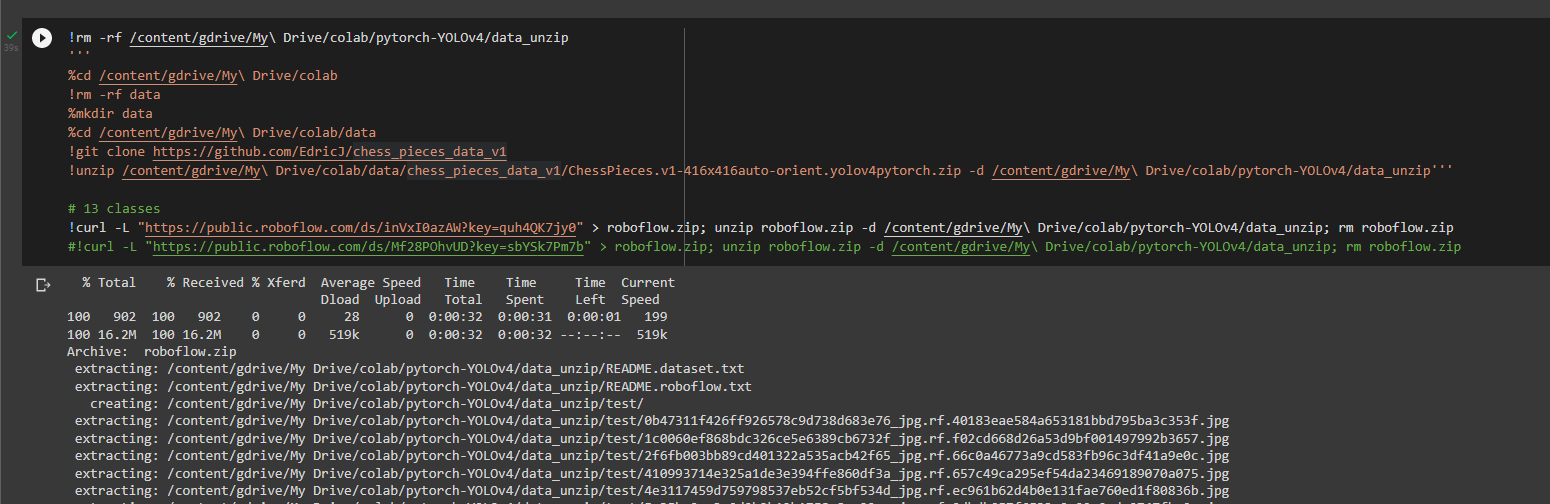


****

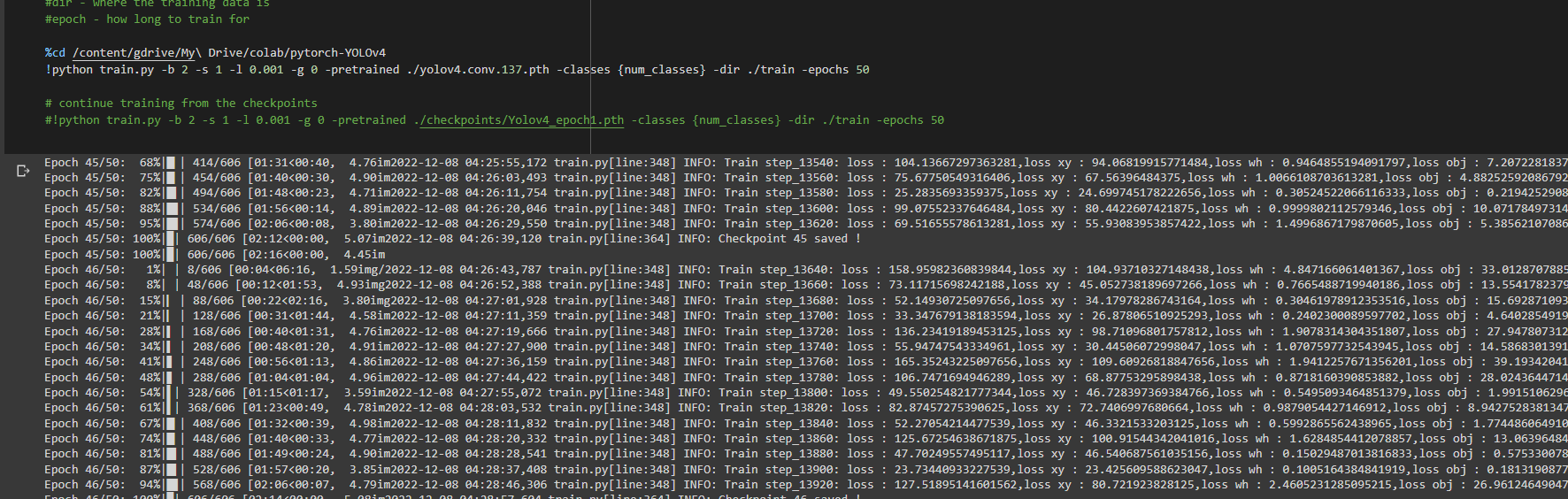
Nhận xét:

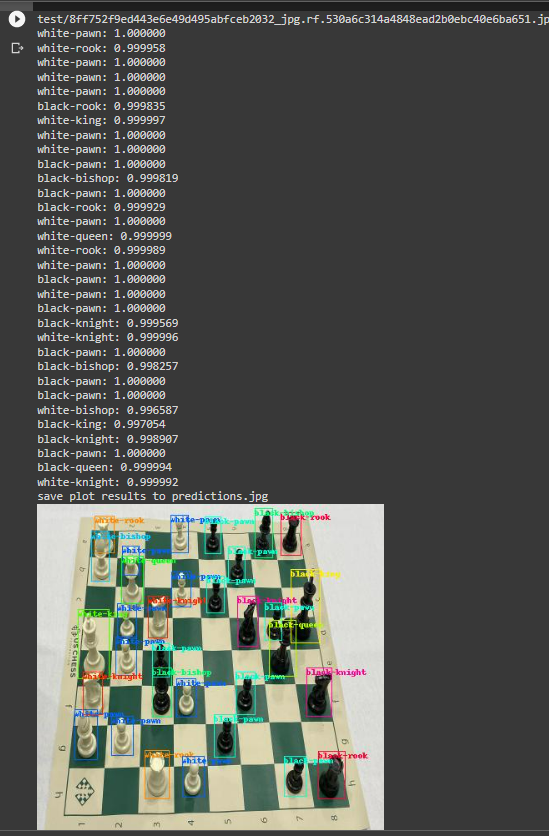
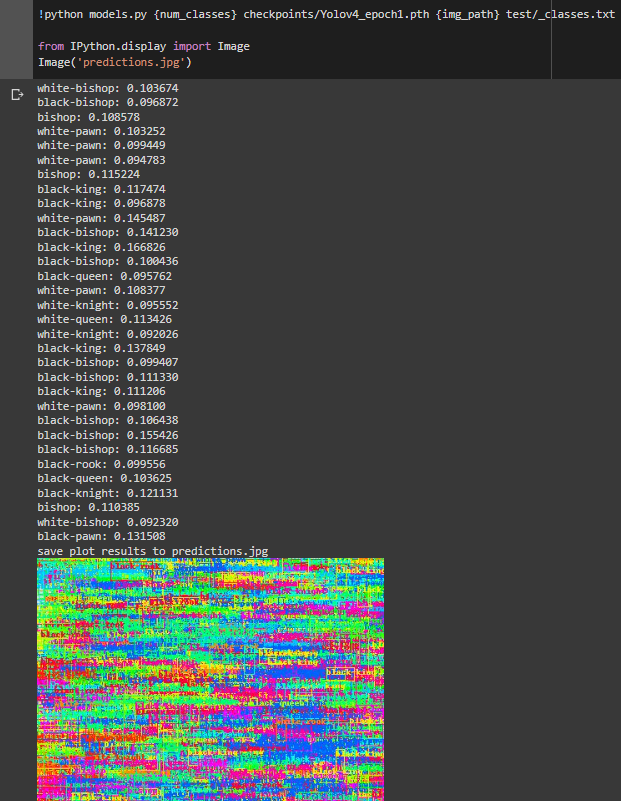
* Với bộ dữ liệu chess Data\_1 thì tốc độ training nhanh hơn ở hai bộ dữ liệu còn lại là Data\_2 và Data\_3 cùng số lần training epoch là 50, cùng bộ trọng số weights yolov4.conv.137.pth

1. **Thực nghiệm với mã nguồn mẫu được cung cấp, thay thế bằng bộ dữ liệu khác**
2. **Thực nghiệm với bộ trọng số yolov4.conv.137.pth**

****Lần này sẽ sử dụng dữ liệu chess Data\_2

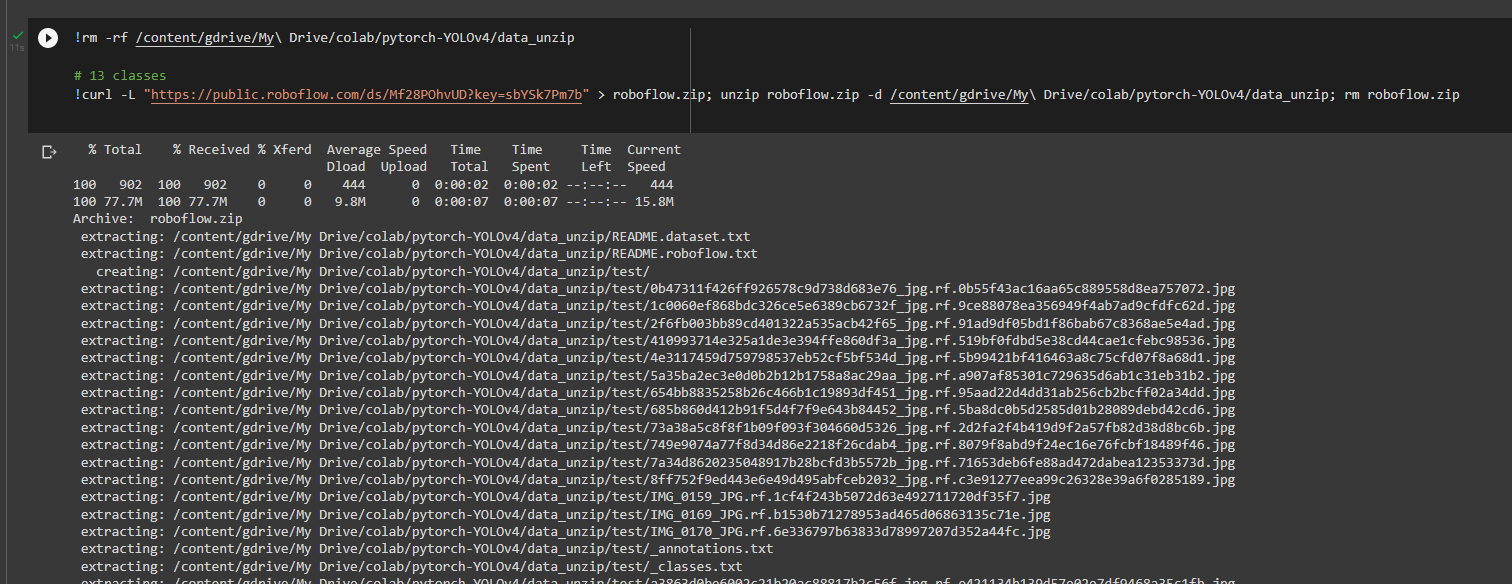
Link: [*https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0*](https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0)

Bắt đầu quá trình training với epoch là 50 (~1h54mins – 2h), ở đây chỉ training một lần, không training thêm từ file checkpoints để tiết kiệm thời gian chạy ở những bộ dữ liệu khác

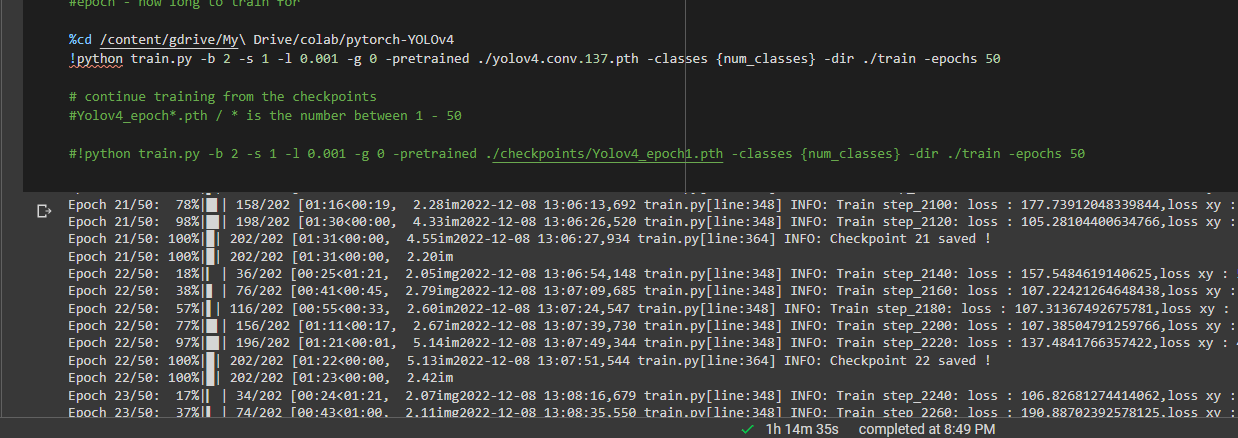
 Kiểm tra kết quả từ file checkpoints ở lớp epoch thứ 46, có thể kiểm tra ở các lớp epoch khác (hình phải epoch1)

Nhận xét:

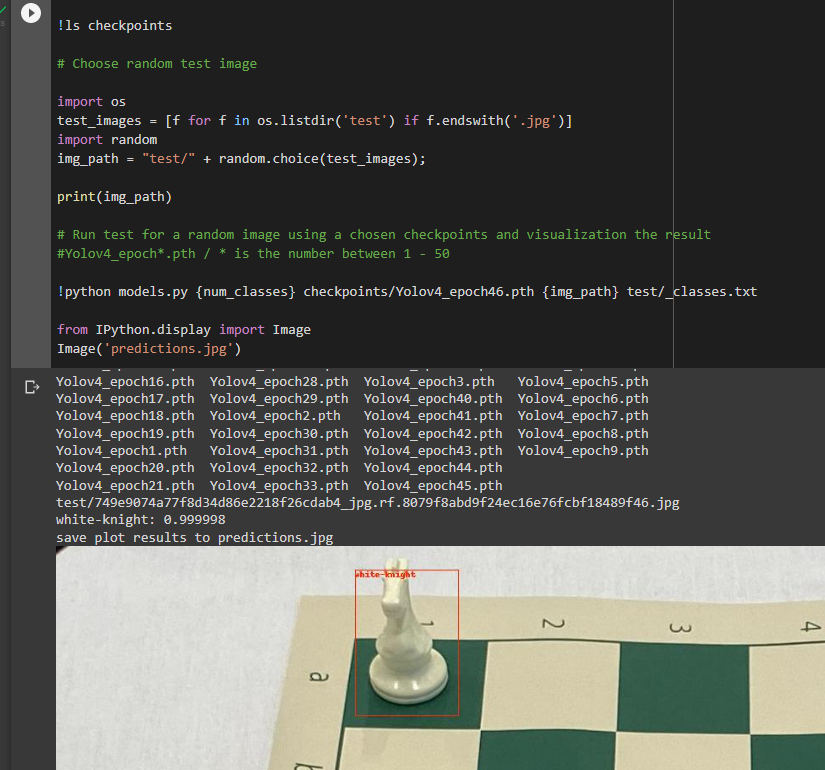
* Với bộ dữ liệu chess Data\_2 thì tốc độ training vô cùng tốn thời gian so với bộ dữ liệu ở Data\_1 và Data\_3 cùng số lần training epoch là 50, cùng bộ trọng số weights yolov4.conv.137.pth

Lần này sẽ sử dụng dữ liệu chess Data\_3

Link: [*https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b*](https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b)

Bắt đầu quá trình training với epoch là 50 (~1h14m35s – 2h), ở đây chỉ training một lần, không training thêm từ file checkpoints để tiết kiệm thời gian chạy ở những bộ dữ liệu khác

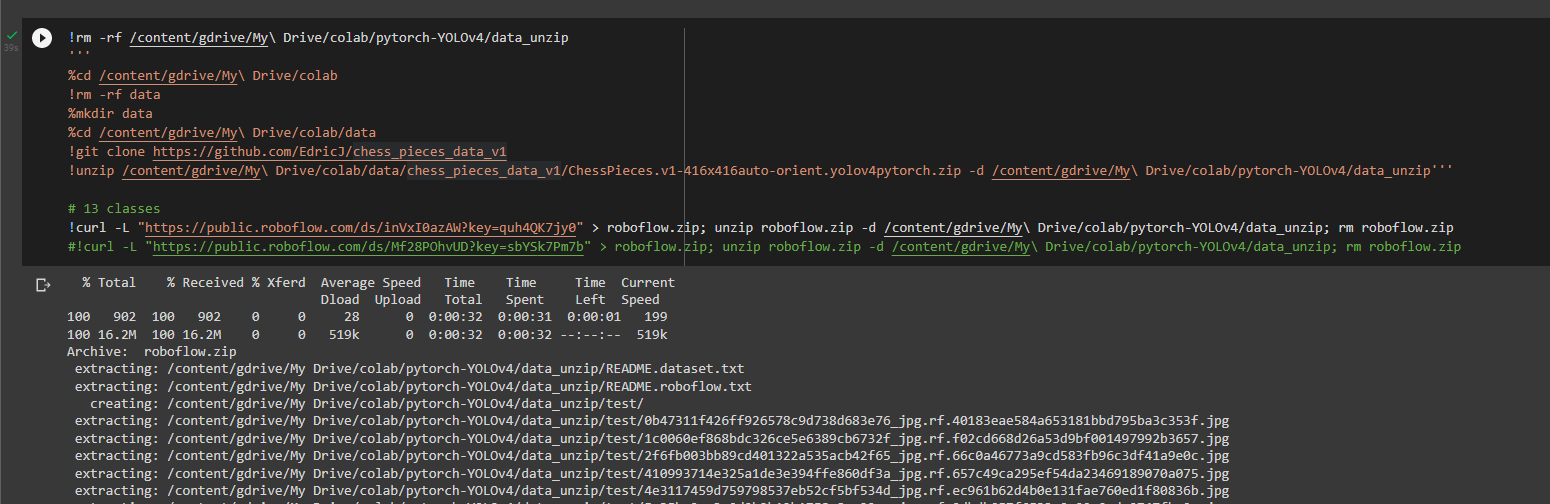
Kiểm tra kết quả từ file checkpoints ở lớp epoch thứ 46, có thể kiểm tra ở các lớp epoch khác



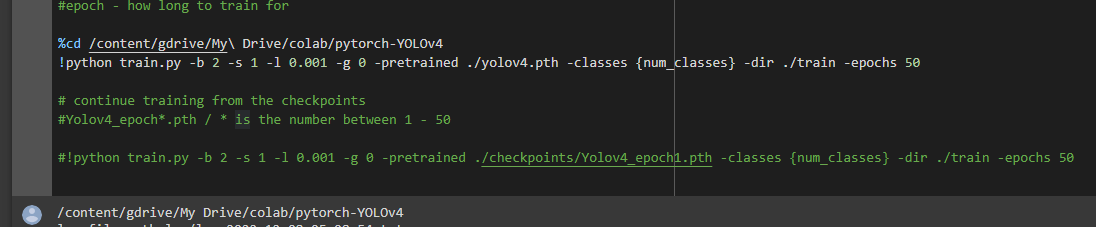
Nhận xét:

* Với bộ dữ liệu chess Data\_3 thì tốc độ training cũng tốn thời gian so với bộ dữ liệu ở Data\_1 nhưng nhanh hơn ở Data\_2 cùng số lần training epoch là 50, cùng bộ trọng số weights yolov4.conv.137.pth

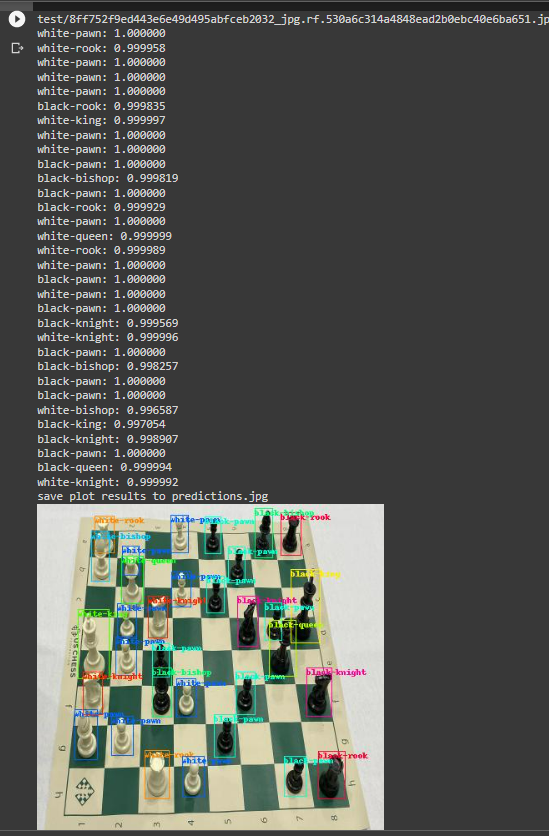
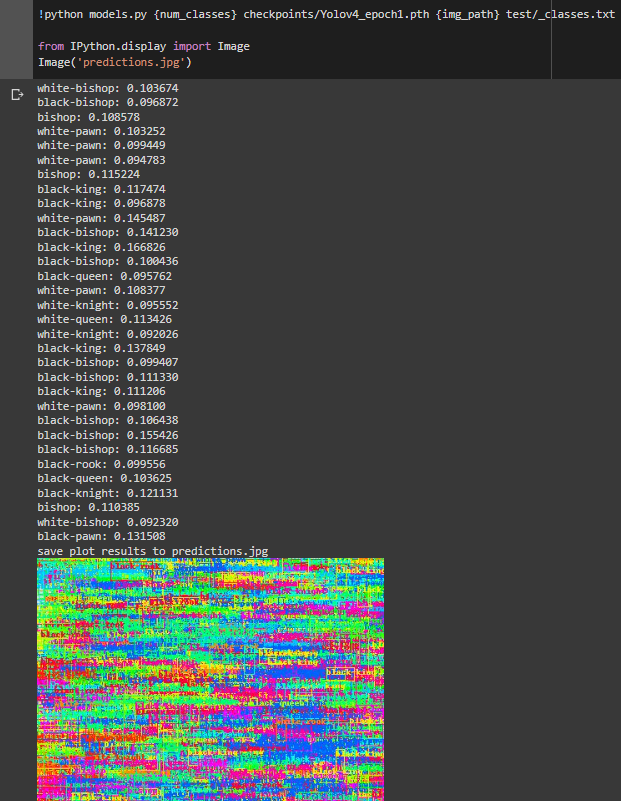
1. **Thực nghiệm với bộ trọng số yolov4.pth (tương tự như thực nghiệm ở trên chỉ là thay đổi bộ trọng số)**

****Lần này sẽ sử dụng dữ liệu chess Data\_2

Link: [*https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0*](https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0)

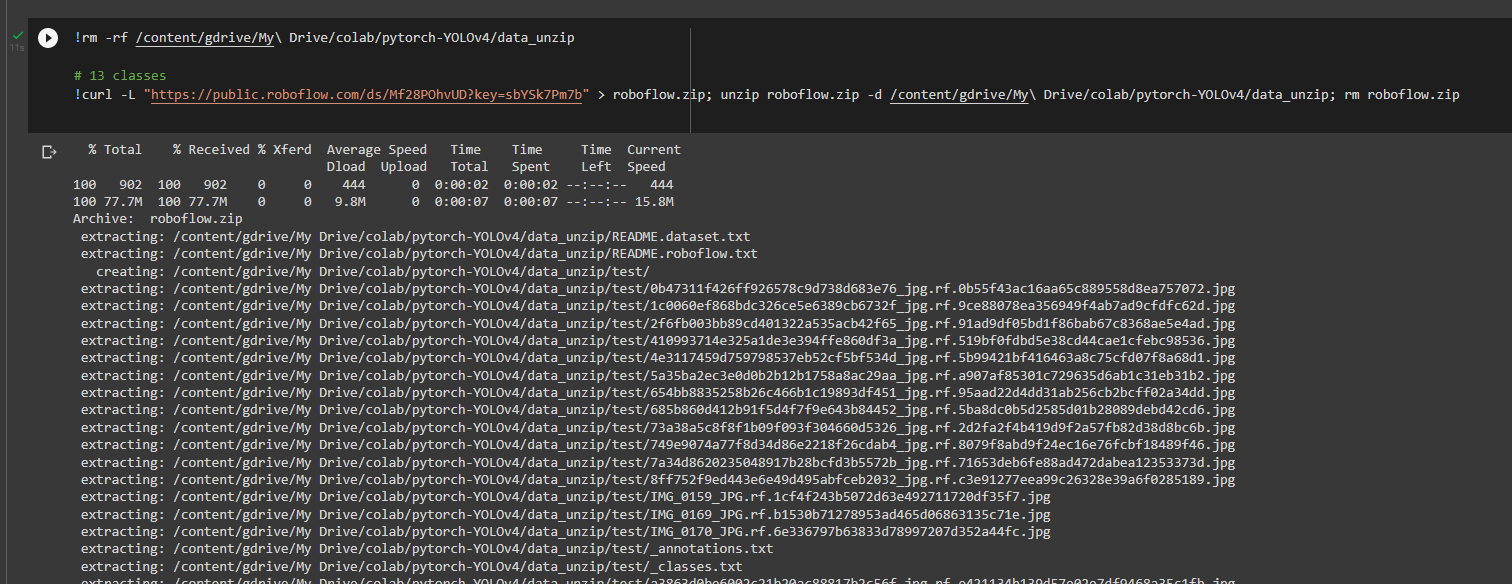
Bắt đầu quá trình training với epoch là 50 (~1h54mins – 2h), ở đây chỉ training một lần, không training thêm từ file checkpoints để tiết kiệm thời gian chạy ở những bộ dữ liệu khác

Kiểm tra kết quả từ file checkpoints ở lớp epoch thứ 46, có thể kiểm tra ở các lớp epoch khác (hình phải epoch1)

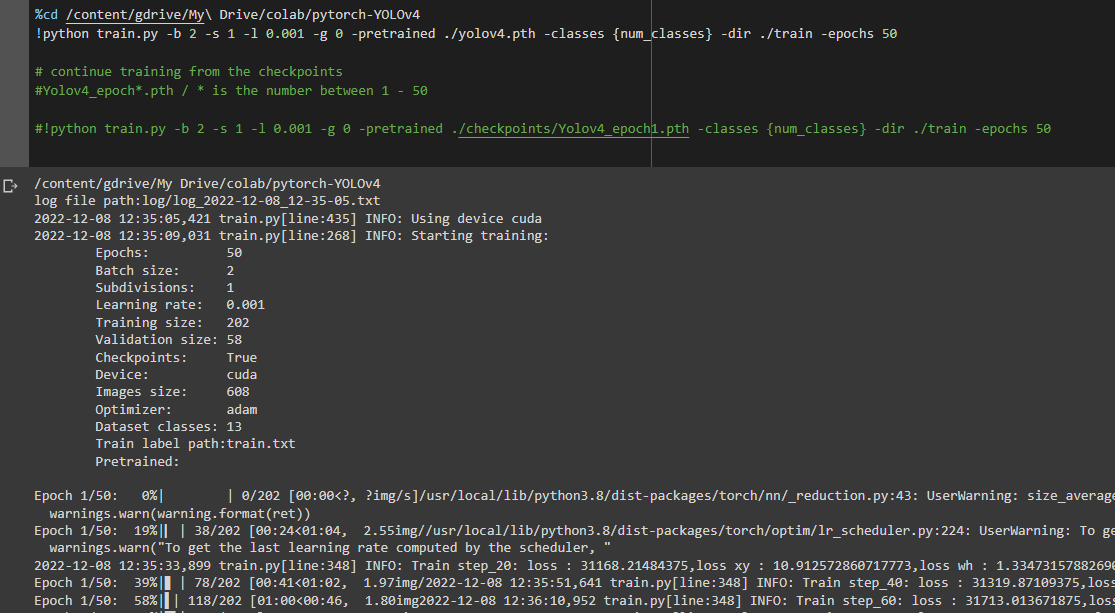


Nhận xét:

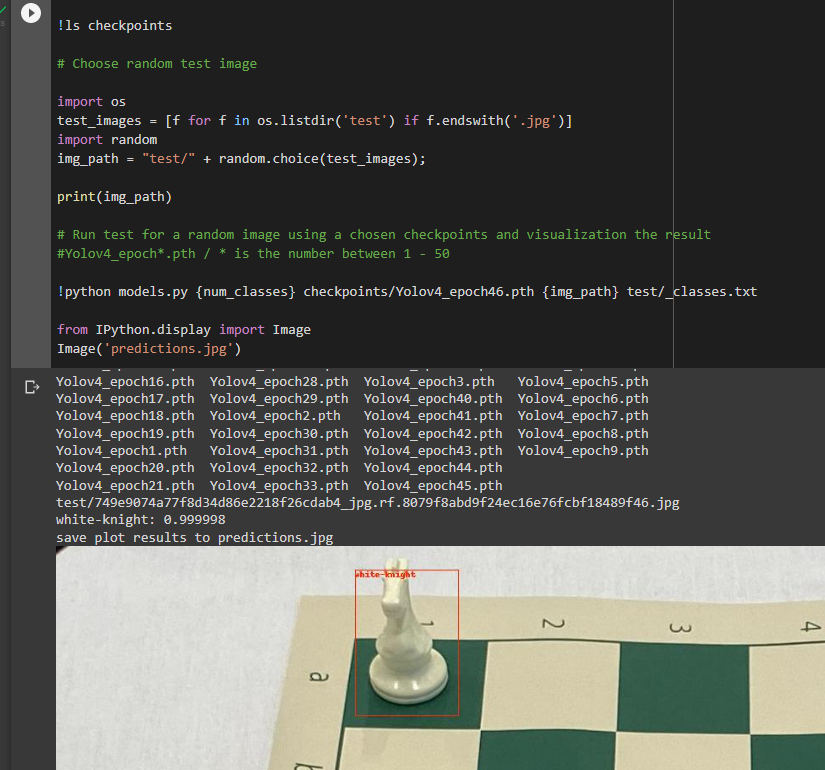
* Với bộ dữ liệu chess Data\_2 thì tốc độ training vô cùng tốn thời gian so với bộ dữ liệu ở Data\_3 cùng số lần training epoch là 50, cùng bộ trọng số weights yolov4.pth

Lần này sẽ sử dụng dữ liệu chess Data\_3

Link: [*https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b*](https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b)

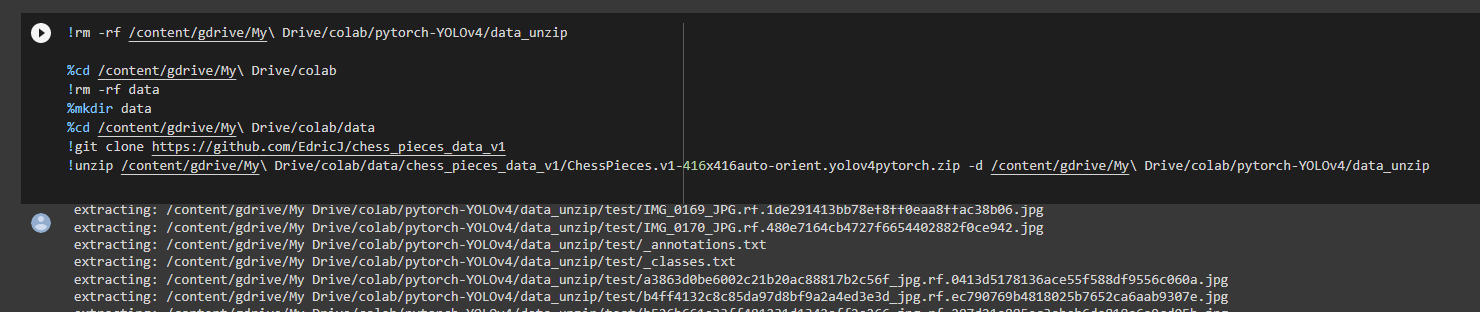
Bắt đầu quá trình training với epoch là 50 (~1h14m35s – 2h), ở đây chỉ training một lần, không training thêm từ file checkpoints để tiết kiệm thời gian chạy ở những bộ dữ liệu khác

Kiểm tra kết quả từ file checkpoints ở lớp epoch thứ 46, có thể kiểm tra ở các lớp epoch khác

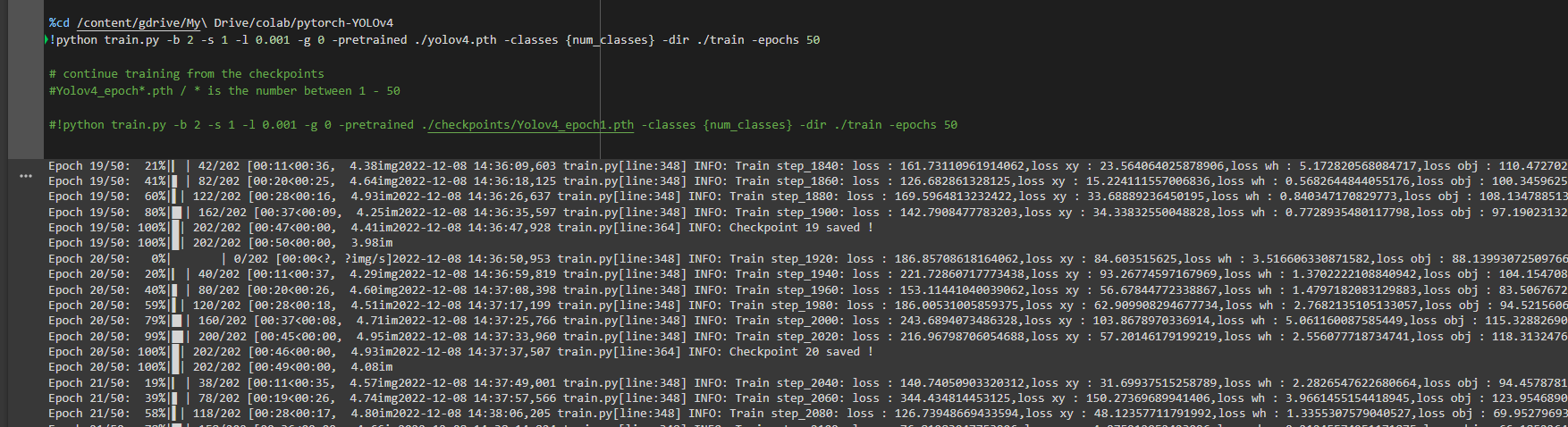


Nhận xét:

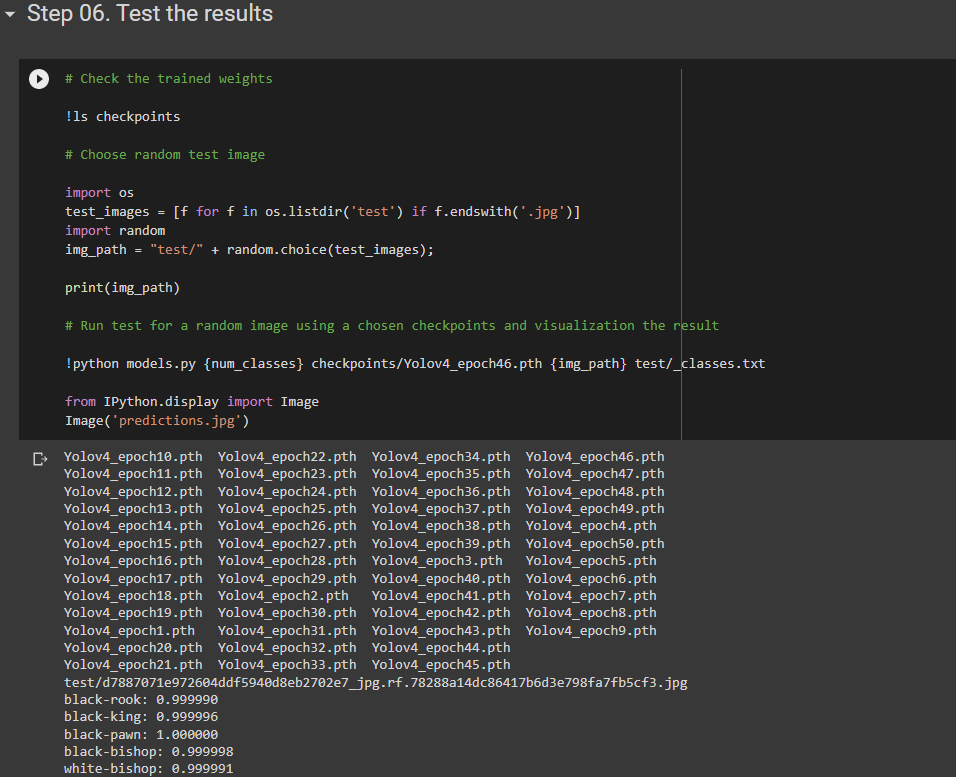
* Với bộ dữ liệu chess Data\_3 thì tốc độ training cũng tốn thời gian so bộ dữ liệu Data\_1 nhưng nhanh hơn ở bộ dữ liệu Data\_2 cùng số lần training epoch là 50, cùng bộ trọng số weights yolov4.pth

 Lần này sẽ sử dụng dữ liệu chess Data\_1

Link: [*https://github.com/EdricJ/chess\_pieces\_data\_v1*](https://github.com/EdricJ/chess_pieces_data_v1)

Bắt đầu quá trình training với epoch là 50 (~35 – 40mins), ở đây chỉ training một lần, không training thêm từ file checkpoints để tiết kiệm thời gian chạy ở những bộ dữ liệu khác

Kiểm tra kết quả từ file checkpoints ở lớp epoch thứ 46, có thể kiểm tra ở các lớp epoch khác



****

Nhận xét:

* Với bộ dữ liệu chess Data\_1 thì tốc độ training nhanh hơn ở hai bộ dữ liệu còn lại là Data\_2 và Data\_3 cùng số lần training epoch là 50, cùng bộ trọng số weights yolov4.pth

1. **Thực nghiệm với model khác, mã nguồn khác, chạy cùng trên các bộ dữ liệu**

Chưa đủ thời gian để thực nghiệm được

# **TÀI LIỆU THAM KHẢO**

**Danh mục tài liệu tham khảo:**

[1] [*https://pytorch.org/*](https://pytorch.org/)

[2][*https://viblo.asia/p/thiet-ke-mang-quantum-neural-network-voi-pytorch-va-qiskit-m2vJPwna4eK*](https://viblo.asia/p/thiet-ke-mang-quantum-neural-network-voi-pytorch-va-qiskit-m2vJPwna4eK)

[3] [*https://phamdinhkhanh.github.io/2019/08/10/PytorchTurtorial1.html*](https://phamdinhkhanh.github.io/2019/08/10/PytorchTurtorial1.html)

[4] [*https://github.com/Tianxiaomo/pytorch-YOLOv4/issues/427*](https://github.com/Tianxiaomo/pytorch-YOLOv4/issues/427)