

**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

<b>THÔNG TIN SINH VIÊN .....</b>	<b>3</b>
<b>I. Thực nghiệm với mã nguồn mẫu và bộ dữ liệu được cung cấp .....</b>	<b>4</b>
<b>II. Thực nghiệm với mã nguồn mẫu được cung cấp, thay thế bằng bộ dữ liệu khác .....</b>	<b>9</b>
a) Thực nghiệm với bộ trọng số yolov4.conv.137.pth .....	9
b) Thực nghiệm với bộ trọng số yolov4.pth (tương tự như thực nghiệm ở trên chỉ là thay đổi bộ trọng số) .....	13
<b>III. Thực nghiệm với model khác, mã nguồn khác, chạy cùng trên các bộ dữ liệu.....</b>	<b>20</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>21</b>

## THÔNG TIN SINH VIÊN

MSSV	Họ Tên	Email	Ghi chú
<b>20120201</b>	Phạm Gia Thông	20120201@student.hcmus.edu.vn	

## I. Thực nghiệm với mã nguồn mẫu và bộ dữ liệu được cung cấp

Tải mã nguồn YOLOv4-PyTorch từ github

### Step 03. Download YOLOv4 PyTorch repo

```
[ ] # Commented out IPython magic to ensure Python compatibility.
# %cd /content/gdrive/My\ Drive
# %mkdir colab
%cd /content/gdrive/My\ Drive/colab
#!rm -rf pytorch-YOLOv4
#!git clone https://github.com/EdricJ/pytorch-YOLOv4
%cd /content/gdrive/My\ Drive/colab/pytorch-YOLOv4

/content/gdrive/My Drive/colab
/content/gdrive/My Drive/colab/pytorch-YOLOv4
```

Link: <https://github.com/EdricJ/pytorch-YOLOv4>

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

```
import gdown

%cd /content/gdrive/My\ Drive/colab
!rm -rf weights
%mkdir weights

%cd /content/gdrive/My\ Drive/colab/weights

!gdown https://drive.google.com/file/d/1fcbR0bWzYfIEdLJPz0sn4R5mlvR6IQyA/view?fbclid=IwAR1f2ZdWo5a_P0wc-hjaQzv5dht0B874z2z2hgx3TJTtRiyeZkUlrnDr3okg -O yolov4.conv.137.pth

def Selected():
    flag = False

    while(flag == False):
        print("Number 1: Dataset training 1\n")
        print("Number 2: Dataset training 2\n")
        print("-----\n")
        a = int(input('Enter the number: '))

        if a == 1:
            # a file
            url = "https://drive.google.com/file/d/1fcbR0bWzYfIEdLJPz0sn4R5mlvR6IQyA/view?fbclid=IwAR1f2ZdWo5a_P0wc-hjaQzv5dht0B874z2z2hgx3TJTtRiyeZkUlrnDr3okg"
            output = "yolov4.conv.137.pth"
            gdown.download(url, output, quiet=False)
            flag = True
        elif a == 2:
            # a file
            url = "https://drive.google.com/file/d/1wv_LiFeCRYwtpkqREPeI13-gPELBDwuJ/view?fbclid=IwAR24B1PTJN0HGUmoyU40dtxiMxnbUQSmIpoYwSeazW0J-HPD3PzDBrCkHA"
            output = "yolov4.pth"
            gdown.download(url, output, quiet=False)
            flag = True

    Selected()

# download yolov4.conv.137.pth
!gdown https://drive.google.com/file/d/1fcbR0bWzYfIEdLJPz0sn4R5mlvR6IQyA/view?fbclid=IwAR1f2ZdWo5a_P0wc-hjaQzv5dht0B874z2z2hgx3TJTtRiyeZkUlrnDr3okg -O yolov4.conv.137.pth
# download yolov4.pth
!gdown https://drive.google.com/file/d/1wv_LiFeCRYwtpkqREPeI13-gPELBDwuJ/view?fbclid=IwAR24B1PTJN0HGUmoyU40dtxiMxnbUQSmIpoYwSeazW0J-HPD3PzDBrCkHA
```

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

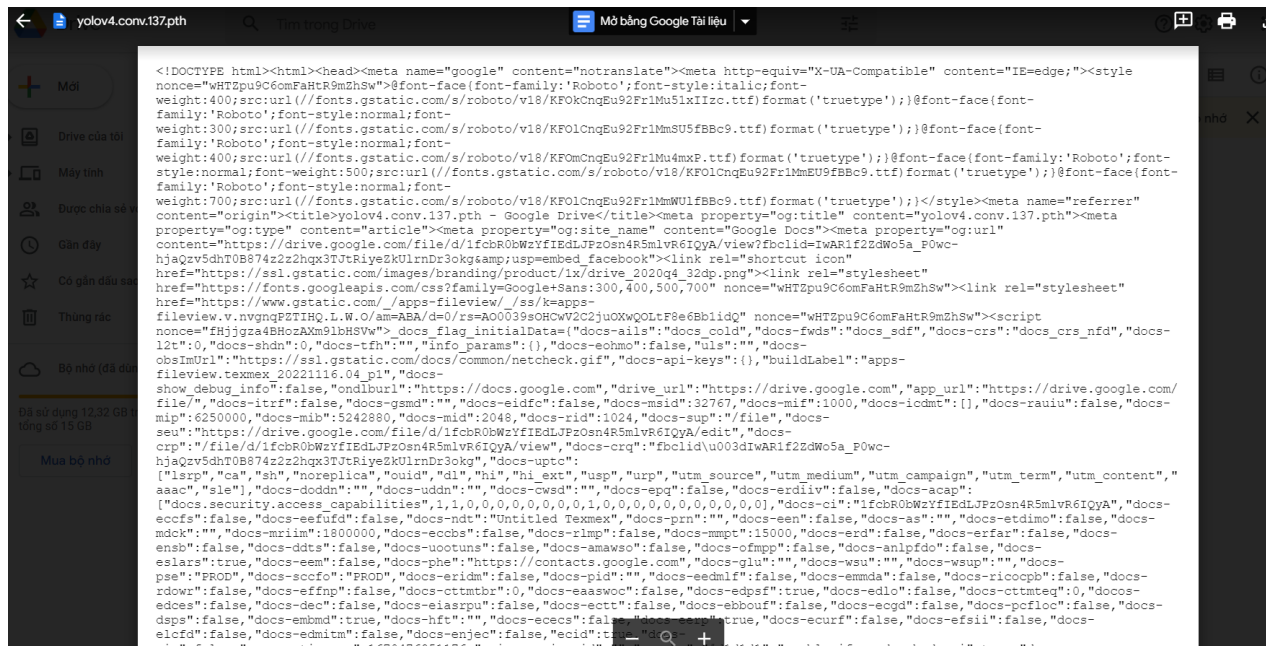
```
# Commented out IPython magic to ensure Python compatibility.

# Commented out IPython magic to ensure Python compatibility.
#start training
# -b batch size (you should keep this low (2-4) for training to work properly)
# -s number of subdivisions in the batch, this was more relevant for the darknet framework
# -l learning rate
# -g direct training to the GPU device
#pretrained invoke the pretrained weights that we downloaded above
# classes - number of classes
#dir - where the training data is
#epoch - how long to train for

%cd /content/gdrive/My Drive/colab/pytorch-YOLOv4
!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./yolov4.conv.137.pth -classes {num_classes} -dir ./tr

# continue training from the checkpoints
#!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./checkpoints/Yolov4_epoch1.pth -classes {num_classes}

/content/gdrive/My Drive/colab/pytorch-YOLOv4
log file path:log/log_2022-12-08_05-08-54.txt
2022-12-08 05:08:54,028 train.py[line:435] INFO: Using device cuda
Traceback (most recent call last):
  File "train.py", line 437, in <module>
    model = YOLOv4(cfg.pretrained,n_classes=cfg.classes)
  File "/content/gdrive/My Drive/colab/pytorch-YOLOv4/models.py", line 398, in __init__
    pretrained_dict = torch.load(yolov4conv137weight)
  File "/usr/local/lib/python3.8/dist-packages/torch/serialization.py", line 529, in load
    return _legacy_load(opened_file, map_location, pickle_module, **pickle_load_args)
  File "/usr/local/lib/python3.8/dist-packages/torch/serialization.py", line 692, in _legacy_load
    magic_number = pickle_module.load(f, **pickle_load_args)
_pickle.UnpicklingError: invalid load key, '<'.
```



Link weights:

google

- yolov4.pth([https://drive.google.com/file/d/1wv\\_LiFeCRYwtpkqREPeI13-gPELBDwuJ/view?fbclid=IwAR3NhtOgnoQZoifRS08LIYvE1Tdh52QHd0hb-VRql24IDahhoS1G8kOC4eo](https://drive.google.com/file/d/1wv_LiFeCRYwtpkqREPeI13-gPELBDwuJ/view?fbclid=IwAR3NhtOgnoQZoifRS08LIYvE1Tdh52QHd0hb-VRql24IDahhoS1G8kOC4eo))
- yolov4.conv.137.pth(<https://drive.google.com/file/d/1fcbR0bWzYfIEdLJPzOsn4R5mlvR6IQyA/view?fbclid=IwAR0tsqk8IMMEqPuOETBHxHvWATSSd-4jWjlcUcUaFpqSdpBYYxUNwwQg0YE>)

baidu

- yolov4.pth([https://pan.baidu.com/s/1ZroDvoGScDgtElja\\_QqJVw](https://pan.baidu.com/s/1ZroDvoGScDgtElja_QqJVw) Extraction code:xrq9)
- yolov4.conv.137.pth(<https://pan.baidu.com/share/init?surl=ovBie4YyVQQoUrC3AY0joA> Extraction code:kcel)

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

```

Step 04.1.1 Unzip dataset / manual in Google Drive

!rm -rf /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip

%cd /content/gdrive/My\ Drive/colab
!rm -rf data
%mkdir data
%cd /content/gdrive/My\ Drive/colab/data
!git clone https://github.com/EdricJ/chess_pieces_data_v1
!unzip /content/gdrive/My\ Drive/colab/data/chess_pieces_data_v1/ChessPieces.v1-416x416auto-orient.yolov4pytorch.zip -d /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip

# 13 classes
!curl -L "https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0" > roboflow.zip; unzip roboflow.zip -d /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip
!curl -L "https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b" > roboflow.zip; unzip roboflow.zip -d /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip

extracting: /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip/train/et10425tuss/et1040/4004430/55304.jpg, 1.5090105728027e7/uet2449801348300.jpg

```

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>

Data\_3: <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

```
#pretrained invoke the pretrained weights that we downloaded above
# classes - number of classes
#dir - where the training data is
#epoch - how long to train for

!cd /content/gdrive/My Drive/colab/pytorch-YOLOv4
!python train.py -b 2 -s 1 -i 0.001 -g 0 -pretrained ./yolov4.conv.137.pth -classes (num_classes) -dir ./train -epochs 50

# continue training from the checkpoints
!python train.py -b 2 -s 1 -i 0.001 -g 0 -pretrained ./checkpoints/Yolov4_epoch0.pth -classes (num_classes) -dir ./train -epochs 50

Epoch 42/50: 100% | 202/202 [00:45:00<00, 4.48s]
Epoch 43/50: 17% | 34/202 [00:09:00<37, 4.52s] 2022-12-07 16:21:16,057 train.py[line:348] INFO: Train step:4200: loss : 13.643383026123847, loss xy : 4.4238691329950855, loss wh : 0.09992081312661171, loss obj : 9.017237632
Epoch 43/50: 37% | 74/202 [00:17:00<25, 4.98s] 2022-12-07 16:21:24,394 train.py[line:348] INFO: Train step:4300: loss : 144.28842602539062, loss xy : 71.5947494586836, loss wh : 2.88601375656128, loss obj : 39.869808897548
Epoch 43/50: 56% | 114/202 [00:25:00<12, 5.01s] 2022-12-07 16:21:32,559 train.py[line:348] INFO: Train step:4380: loss : 115.26133728027344, loss xy : 83.596183115025, loss wh : 0.04196120087865, loss obj : 12.682239628295
Epoch 43/50: 76% | 154/202 [00:34:00<09, 5.06s] 2022-12-07 16:21:40,745 train.py[line:348] INFO: Train step:4320: loss : 80.34551239813672, loss xy : 45.245784759521404, loss wh : 1.079777956080111, loss obj : 17.0345687866
Epoch 43/50: 96% | 194/202 [00:41:00<01, 5.32s] 2022-12-07 16:21:48,460 train.py[line:348] INFO: Train step:4340: loss : 28.5197624206543, loss xy : 17.384037817822266, loss wh : 0.15756547451019287, loss obj : 10.776942253
Epoch 43/50: 100% | 202/202 [00:41:00<00, 4.57s]
Epoch 43/50: 100% | 202/202 [00:41:00<00, 4.57s]
Epoch 44/50: 10% | 32/202 [00:09:00<39, 4.38s] 2022-12-07 16:22:00,228 train.py[line:348] INFO: Train step:4360: loss : 37.928401947021484, loss xy : 21.855426788330078, loss wh : 0.4796936511993408, loss obj : 11.154524509
Epoch 44/50: 30% | 72/202 [00:17:00<26, 4.94s] 2022-12-07 16:22:08,376 train.py[line:348] INFO: Train step:4380: loss : 61.33974838256836, loss xy : 37.23344421386719, loss wh : 1.152767685394287, loss obj : 15.072915077209
Epoch 44/50: 51% | 112/202 [00:25:00<18, 4.86s] 2022-12-07 16:22:16,377 train.py[line:348] INFO: Train step:4400: loss : 94.431804785625, loss xy : 40.66083459472656, loss wh : 1.257193511004336, loss obj : 40.294937137006
Epoch 44/50: 71% | 152/202 [00:31:00<10, 4.94s] 2022-12-07 16:22:24,398 train.py[line:348] INFO: Train step:4420: loss : 46.57600402832031, loss xy : 31.120925903120312, loss wh : 0.8657251596450806, loss obj : 11.3547925940
Epoch 44/50: 91% | 192/202 [00:41:00<01, 5.32s] 2022-12-07 16:22:32,073 train.py[line:348] INFO: Train step:4440: loss : 104.98644256591797, loss xy : 58.4875682722168, loss wh : 12.355515480041504, loss obj : 23.23963928222
Epoch 44/50: 100% | 202/202 [00:41:00<00, 5.29s] 2022-12-07 16:22:34,505 train.py[line:364] INFO: Checkpoint 44 saved !
Epoch 44/50: 100% | 202/202 [00:41:00<00, 4.69s]
Epoch 45/50: 15% | 30/202 [00:08:00<39, 4.48s] 2022-12-07 16:22:43,564 train.py[line:348] INFO: Train step:4460: loss : 14.957091331481934, loss xy : 5.297279357918156, loss wh : 0.28418723984241486, loss obj : 9.2909879680
Epoch 45/50: 35% | 70/202 [00:16:00<27, 4.87s] 2022-12-07 16:22:51,694 train.py[line:348] INFO: Train step:4480: loss : 71.24811553955078, loss xy : 58.00640657128086, loss wh : 0.7125169038772583, loss obj : 12.0158070190
Epoch 45/50: 54% | 110/202 [00:24:00<19, 4.97s] 2022-12-07 16:22:59,809 train.py[line:348] INFO: Train step:4500: loss : 451.30858168583082, loss xy : 93.87407956542909, loss wh : 2.9740609375, loss obj : 45.50868791583066, 1
Epoch 45/50: 74% | 150/202 [00:31:00<10, 5.03s] 2022-12-07 16:23:07,971 train.py[line:348] INFO: Train step:4520: loss : 101.24600375, loss xy : 63.44384002685547, loss wh : 1.956400593322754, loss obj : 21.33523178100586, 1
Epoch 45/50: 94% | 190/202 [00:41:00<02, 5.21s] 2022-12-07 16:23:15,902 train.py[line:348] INFO: Train step:4540: loss : 91.09363555908203, loss xy : 58.136940002441406, loss wh : 5.5900892124938965, loss obj : 20.624900158
Epoch 45/50: 100% | 202/202 [00:41:00<00, 5.35s] 2022-12-07 16:23:18,713 train.py[line:364] INFO: Checkpoint 45 saved !
Epoch 45/50: 100% | 202/202 [00:41:00<00, 4.57s]
Epoch 46/50: 14% | 28/202 [00:08:00<40, 4.25s] 2022-12-07 16:23:27,817 train.py[line:348] INFO: Train step:4560: loss : 34.217919921875, loss xy : 24.283174514770500, loss wh : 0.1720022611427307, loss obj : 7.46017533870
Epoch 46/50: 34% | 68/202 [00:16:00<26, 4.98s] 2022-12-07 16:23:36,003 train.py[line:348] INFO: Train step:4580: loss : 46.30998992919922, loss xy : 32.147274817133884, loss wh : 0.43514040451040085, loss obj : 6.7380599617
Epoch 46/50: 53% | 108/202 [00:24:00<19, 4.86s] 2022-12-07 16:23:44,230 train.py[line:348] INFO: Train step:4600: loss : 134.52752085546075, loss xy : 83.63407920480468, loss wh : 2.5190750164031902, loss obj : 26.5058212280
Epoch 46/50: 73% | 148/202 [00:31:00<11, 4.52s] 2022-12-07 16:23:52,059 train.py[line:348] INFO: Train step:4620: loss : 127.91422271728516, loss xy : 77.58027648925781, loss wh : 4.507273610790715, loss obj : 28.831432342
Epoch 46/50: 93% | 188/202 [00:41:00<02, 5.20s] 2022-12-07 16:24:00,442 train.py[line:348] INFO: Train step:4640: loss : 50.42981719970703, loss xy : 29.930648803710938, loss wh : 0.4625515937805176, loss obj : 8.6134827888
Epoch 46/50: 100% | 202/202 [00:41:00<00, 5.22s] 2022-12-07 16:24:03,072 train.py[line:364] INFO: Checkpoint 46 saved !
Epoch 46/50: 100% | 202/202 [00:41:00<00, 4.49s]
Epoch 47/50: 11% | 26/202 [00:07:00<38, 4.61s] 2022-12-07 16:24:11,525 train.py[line:348] INFO: Train step:4660: loss : 11.363703727722168, loss xy : 4.781116962432861, loss wh : 0.20061878859996796, loss obj : 6.3612823486
Epoch 47/50: 31% | 66/202 [00:15:00<27, 4.96s] 2022-12-07 16:24:19,971 train.py[line:348] INFO: Train step:4680: loss : 36.59770584106445, loss xy : 29.63576808918086, loss wh : 0.34291398525238037, loss obj : 6.50857245095
Epoch 47/50: 51% | 106/202 [00:24:00<19, 4.86s] 2022-12-07 16:24:28,088 train.py[line:348] INFO: Train step:4700: loss : 68.24302033120844, loss xy : 38.60216155080506, loss wh : 0.74223113736316, loss obj : 23.675951804032
Epoch 47/50: 71% | 146/202 [00:31:00<11, 4.79s] 2022-12-07 16:24:36,292 train.py[line:348] INFO: Train step:4720: loss : 47.51852035522461, loss xy : 23.781118392944336, loss wh : 0.9693610872135925, loss obj : 14.5029592314
```

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

```
Step 06. Test the results

# Check the trained weights

!ls checkpoints

# Choose random test image

import os
test_images = [f for f in os.listdir('test') if f.endswith('.jpg')]
import random
img_path = "test/" + random.choice(test_images);

print(img_path)

# Run test for a random image using a chosen checkpoints and visualization the result

!python models.py {num_classes} checkpoints/Yolov4_epoch46.pth {img_path} test/_classes.txt

from IPython.display import Image
Image('predictions.jpg')
```

Yolov4_epoch10.pth	Yolov4_epoch22.pth	Yolov4_epoch34.pth	Yolov4_epoch46.pth
Yolov4_epoch11.pth	Yolov4_epoch23.pth	Yolov4_epoch35.pth	Yolov4_epoch47.pth
Yolov4_epoch12.pth	Yolov4_epoch24.pth	Yolov4_epoch36.pth	Yolov4_epoch48.pth
Yolov4_epoch13.pth	Yolov4_epoch25.pth	Yolov4_epoch37.pth	Yolov4_epoch49.pth
Yolov4_epoch14.pth	Yolov4_epoch26.pth	Yolov4_epoch38.pth	Yolov4_epoch40.pth
Yolov4_epoch15.pth	Yolov4_epoch27.pth	Yolov4_epoch39.pth	Yolov4_epoch50.pth
Yolov4_epoch16.pth	Yolov4_epoch28.pth	Yolov4_epoch30.pth	Yolov4_epoch51.pth
Yolov4_epoch17.pth	Yolov4_epoch29.pth	Yolov4_epoch400.pth	Yolov4_epoch6.pth
Yolov4_epoch18.pth	Yolov4_epoch20.pth	Yolov4_epoch410.pth	Yolov4_epoch7.pth
Yolov4_epoch19.pth	Yolov4_epoch30.pth	Yolov4_epoch420.pth	Yolov4_epoch8.pth
Yolov4_epoch100.pth	Yolov4_epoch310.pth	Yolov4_epoch430.pth	Yolov4_epoch900.pth
Yolov4_epoch200.pth	Yolov4_epoch320.pth	Yolov4_epoch440.pth	
Yolov4_epoch2100.pth	Yolov4_epoch3300.pth	Yolov4_epoch4500.pth	

```
test/d7887071e972684ddf59408eb2702e7_jpg.rf.78288a14dc86417b6d3e798fa7fb5cf3.jpg
black-rook: 0.999990
black-king: 0.999996
black-pawn: 1.000000
black-bishop: 0.999998
white-bishop: 0.999991
```



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



## II. Thực nghiệm với mã nguồn mẫu được cung cấp, thay thế bằng bộ dữ liệu khác

### a) Thực nghiệm với bộ trọng số yolov4.conv.137.pth

Lần này sẽ sử dụng dữ liệu chess Data\_2

```
!rm -rf ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip
...
%cd ./content/gdrive/My Drive/colab
!rm -rf data
%mkdir data
%cd ./content/gdrive/My Drive/colab/data
!git clone https://github.com/EdricJ/chess_pieces_data_v1
!unzip ./content/gdrive/My Drive/colab/data/chess_pieces_data_v1/ChessPieces.v1-416x416auto-orient.yolov4pytorch.zip -d ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip'''

# 13 classes
!curl -L "https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0" > roboflow.zip; unzip roboflow.zip -d ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip
!curl -L "https://public.roboflow.com/ds/HF28POhVUD?key=sbY5k7Pm7b" > roboflow.zip; unzip roboflow.zip -d ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip

% Total      % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed

100 902    100 902    0    0    28    0 0:00:32 0:00:31 0:00:01 199
100 16.2M 100 16.2M    0    0 519k    0 0:00:32 0:00:32 --:--:-- 519k

Archive:  roboflow.zip
  extracting: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/README.dataset.txt
  extracting: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/README.roboflow.txt
    creating: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/
  extracting: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/0b47311f426ff926578cd738d683e76.jpg.rf_40183eae584a653181bbd795ba3c353f.jpg
  extracting: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/1c080a6f868bdc326c5e6388cb6723f.jpg.rf_f02c4668d0e5349bf0e1497992b3657.jpg
  extracting: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/2f6f0803bb80cd401322535c4b2f65.jpg.rf_66c0a46773abdc583fb96c34f41a9a0c.jpg
  extracting: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/410903714e325a1de3e3044fe86edf3a.jpg.rf_657c40ca205ef54da22460130070a075.jpg
  extracting: ./content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/4e3117459d759798537eb52cf5bf534d.jpg.rf_ec961b62d4b0e121fae760ed1f08836b.jpg
```

Link: <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

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#epoch - how long to train for

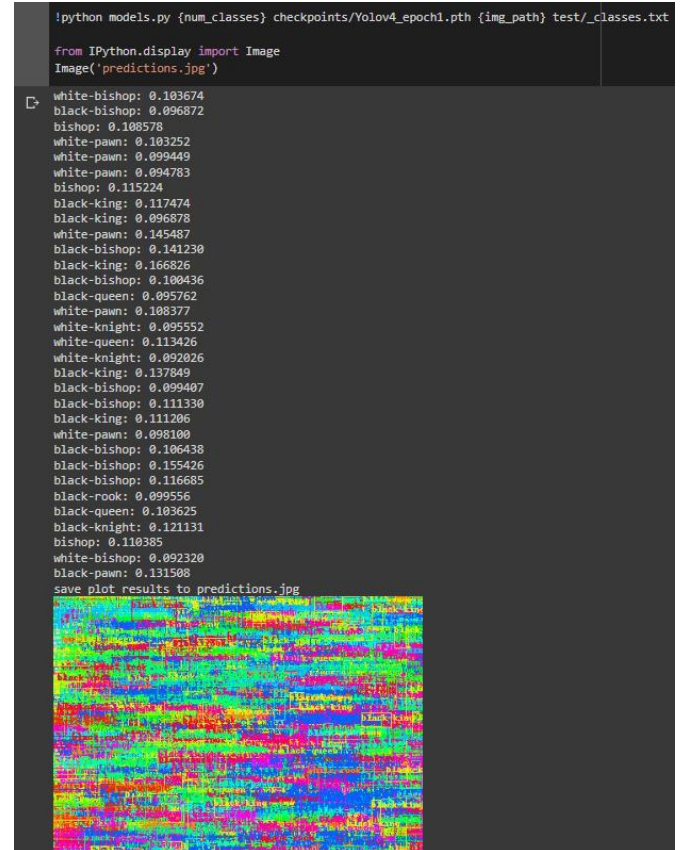
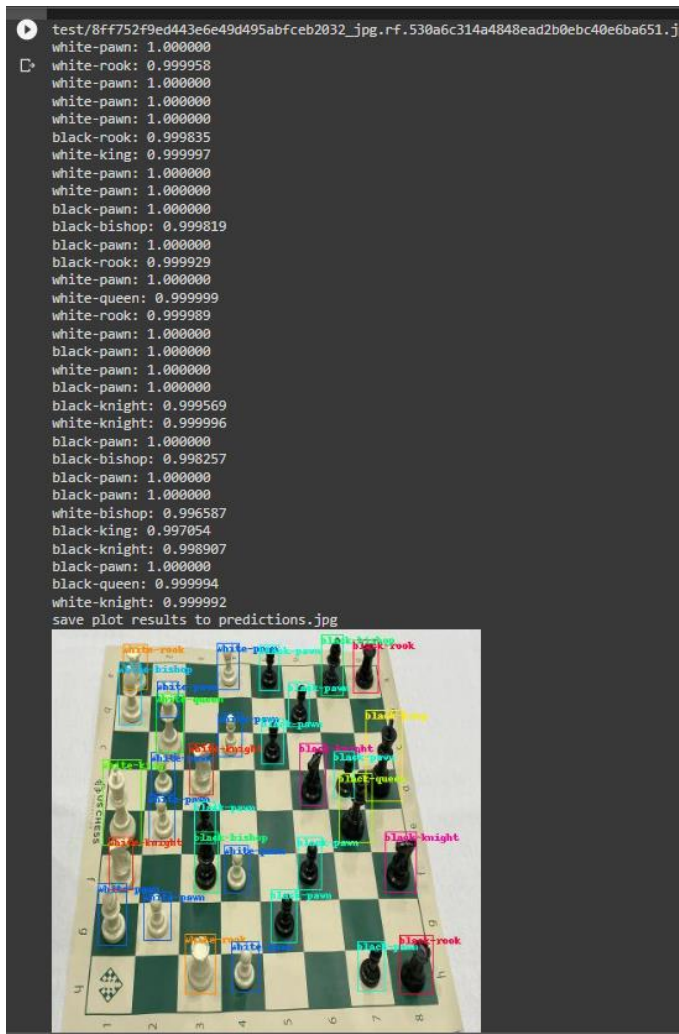
%cd ./content/gdrive/My Drive/colab/pytorch-YOLOv4
!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./yolov4.conv.137.pth -classes {num_classes} -dir ./train -epochs 50

# continue training from the checkpoints
#!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./checkpoints/Yolov4_epoch1.pth -classes {num_classes} -dir ./train -epochs 50

Epoch 45/50: 68% [ 414/606 [01:31:00:40, 4.76im2022-12-08 04:25:55.172 train.py[line:348] INFO: Train step 13540: loss : 104.13667297363281,loss xy : 94.06819915771484,loss wh : 0.9464855194091797,loss obj : 7.2072281837
Epoch 45/50: 75% [ 454/606 [01:40:00:30, 4.90im2022-12-08 04:26:03.403 train.py[line:348] INFO: Train step 13560: loss : 75.677954931646,loss xy : 67.56306404375,loss wh : 1.0066100703613281,loss obj : 4.80253502006792
Epoch 45/50: 82% [ 494/606 [01:48:00:23, 4.71im2022-12-08 04:26:11.754 train.py[line:348] INFO: Train step 13580: loss : 25.2835693359375,loss xy : 24.699745178222656,loss wh : 0.3852452286616333,loss obj : 0.2194252908
Epoch 45/50: 88% [ 534/606 [01:56:00:14, 4.89im2022-12-08 04:26:20.046 train.py[line:348] INFO: Train step 13600: loss : 99.07552337646484,loss xy : 80.4422607421875,loss wh : 0.9999802112579346,loss obj : 10.07178407314
Epoch 45/50: 95% [ 574/606 [02:06:00:08, 3.80im2022-12-08 04:26:29.550 train.py[line:348] INFO: Train step 13620: loss : 69.51655578613281,loss xy : 55.93883953857422,loss wh : 1.4996867179870605,loss obj : 5.385621078086
Epoch 45/50: 100% [ 606/606 [02:12:00:00, 5.07im2022-12-08 04:26:39.120 train.py[line:364] INFO: Checkpoint 45 saved !

Epoch 46/50: 100% [ 606/606 [02:16:00:00, 4.45im
Epoch 46/50: 1% [ 0/606 [00:04:00:16, 1.59im2022-12-08 04:26:43.787 train.py[line:348] INFO: Train step 13640: loss : 159.95802368039844,loss xy : 104.93710327140439,loss wh : 4.847166961401357,loss obj : 33.0120707893
Epoch 46/50: 6% [ 48/606 [00:12:01:53, 4.93im2022-12-08 04:26:52.388 train.py[line:348] INFO: Train step 13660: loss : 73.1171568242188,loss xy : 45.052738189697266,loss wh : 0.7665488719940186,loss obj : 13.5541782375
Epoch 46/50: 15% [ 88/606 [00:22:02:16, 3.80im2022-12-08 04:27:01.928 train.py[line:348] INFO: Train step 13680: loss : 52.14930725097656,loss xy : 34.17978286743164,loss wh : 0.38461978912353516,loss obj : 15.692871093
Epoch 46/50: 21% [ 128/606 [00:31:01:44, 4.58im2022-12-08 04:27:11.359 train.py[line:348] INFO: Train step 13700: loss : 33.347679138183594,loss xy : 26.87806510925293,loss wh : 0.2402300089597702,loss obj : 4.64028549159
Epoch 46/50: 28% [ 168/606 [00:40:01:31, 4.76im2022-12-08 04:27:19.666 train.py[line:348] INFO: Train step 13720: loss : 136.23419189453125,loss xy : 98.71096801757812,loss wh : 1.9078314304351807,loss obj : 27.947807312
Epoch 46/50: 34% [ 208/606 [00:48:01:20, 4.91im2022-12-08 04:27:27.900 train.py[line:348] INFO: Train step 13740: loss : 55.94747543334961,loss xy : 30.44506072998047,loss wh : 1.0707597732543945,loss obj : 14.5868101391
Epoch 46/50: 41% [ 248/606 [00:56:01:11, 4.86im2022-12-08 04:27:36.159 train.py[line:348] INFO: Train step 13760: loss : 165.35243225097656,loss xy : 109.60926818847656,loss wh : 1.9412257671356201,loss obj : 39.19142041
Epoch 46/50: 48% [ 288/606 [01:04:01:04, 4.96im2022-12-08 04:27:44.422 train.py[line:348] INFO: Train step 13780: loss : 106.7471694946289,loss xy : 68.87753295898438,loss wh : 0.8718160390853882,loss obj : 28.0243644714
Epoch 46/50: 54% [ 328/606 [01:15:01:17, 3.59im2022-12-08 04:27:55.072 train.py[line:348] INFO: Train step 13800: loss : 49.550254821777344,loss xy : 46.728397369384766,loss wh : 0.5495093464851379,loss obj : 1.9915106296
Epoch 46/50: 61% [ 368/606 [01:23:00:49, 4.78im2022-12-08 04:28:03.532 train.py[line:348] INFO: Train step 13820: loss : 82.87457275390625,loss xy : 72.7406997680664,loss wh : 0.9879054427146912,loss obj : 8.9427528381347
Epoch 46/50: 67% [ 408/606 [01:32:00:49, 4.98im2022-12-08 04:28:11.832 train.py[line:348] INFO: Train step 13840: loss : 52.270542144477539,loss xy : 46.3321533203125,loss wh : 0.5992865562438965,loss obj : 1.774486064916
Epoch 46/50: 74% [ 448/606 [01:40:00:33, 4.77im2022-12-08 04:28:20.332 train.py[line:348] INFO: Train step 13860: loss : 125.07254638671875,loss xy : 100.91544342041016,loss wh : 1.6284854412078857,loss obj : 13.06396484
Epoch 46/50: 81% [ 488/606 [01:49:00:24, 4.90im2022-12-08 04:28:28.541 train.py[line:348] INFO: Train step 13880: loss : 47.70240557495117,loss xy : 46.540887516035155,loss wh : 0.15029407913816833,loss obj : 0.5753300872
Epoch 46/50: 87% [ 528/606 [01:57:00:20, 3.85im2022-12-08 04:28:37.488 train.py[line:348] INFO: Train step 13900: loss : 23.724440933227539,loss xy : 23.42560588622047,loss wh : 0.1005164384841919,loss obj : 0.1813190877
Epoch 46/50: 94% [ 568/606 [02:06:00:07, 4.79im2022-12-08 04:28:46.306 train.py[line:348] INFO: Train step 13920: loss : 127.51895141601562,loss xy : 80.721923828125,loss wh : 2.4605231285095215,loss obj : 26.96124649047
```

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

```

!rm -rf /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip

# 13 classes
!curl -L "https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b" > roboflow.zip; unzip roboflow.zip -d /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 902    100 902    0     0    444      0  0:00:02  0:00:02 --:--:--  444
100 77.7M 100 77.7M    0     0   9.8M      0  0:00:07  0:00:07 --:--:-- 15.8M
Archive:  roboflow.zip
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/README.dataset.txt
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/README.roboflow.txt
   creating: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/0b47311f426ff926578c9d738d683e76.jpg.rf.0b55f43ac16aa65c889558d8ea757072.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/1c0060ef868bdc326ce5e6389c6732f.jpg.rf.9ce88078ea356949f4ab7ad9cfdfc62d.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/2f6fb003bb89cd481322a535ac642f65.jpg.rf.91ad9df05bd1f86bab67c8368ae5e4ad.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/410993714e325a1de3e394ffe860df3a.jpg.rf.519bf0f0bd5e38cd44cae1cfbc98536.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/4e3117459d759798537eb52cf5bf534d.jpg.rf.5b99421bf416463a8c75cf07f8a68d1.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/5a35ba2ec3e0d0b2b12b1758a8ac29aa.jpg.rf.a907af85301c729635d6ab1c31eb31b2.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/654bb8835258b26c466b1c19893df451.jpg.rf.95aad224dd31ab256cb2bcff02a34dd.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/685b860d412b91f5d4f7f9e643b84452.jpg.rf.5ba8dc0b5d2585d01b2889debd42cd6.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/73a38a5c8f8f1b09f093f304660d5326.jpg.rf.2d2fa2f4b419d9f2a57fb82d38dbbc6b.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/749e9074a77f8d34d86e2218f26cdab4.jpg.rf.8079f8ab9f24ec16e76fcbf18489f46.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/7a34d862023504891b28bcbf3b5572b.jpg.rf.71653deb6fe88ad472dabae1235373d.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/8ff752f9ed443e6e49d495abfceb2032.jpg.rf.c3e91277eea99c26328e39a6f0285189.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0159.JPG.rf.1cf4f243b5072d63e492711720df35f7.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0169.JPG.rf.b1530b71278953ad465d06863135c71e.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0170.JPG.rf.6e336797b63833d78997207d352a44fc.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/_annotations.txt
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/_classes.txt
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/52962d0bc002c21b30e08913b3c56f.jpg.rf.e03123b132d52e02c7d0960e23f1fb.jpg

```

Link: <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

```

#epoch = how long to train for

%cd /content/gdrive/My Drive/colab/pytorch-YOLOv4
!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./yolov4.conv.137.pth -classes {num_classes} -dir ./train -epochs 50

# continue training from the checkpoints
#Yolov4_epoch*.pth / * is the number between 1 - 50

#!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./checkpoints/Yolov4_epoch1.pth -classes {num_classes} -dir ./train -epochs 50

Epoch 21/50: 78%| 158/202 [01:16<00:19, 2.28im2022-12-08 13:06:13,692 train.py[line:348] INFO: Train step_2100: loss : 177.73912048339844,loss xy :
Epoch 21/50: 98%| 198/202 [01:30<00:00, 4.33im2022-12-08 13:06:26,520 train.py[line:348] INFO: Train step_2120: loss : 105.28104400634766,loss xy :
Epoch 21/50: 100%| 202/202 [01:31<00:00, 4.55im2022-12-08 13:06:27,934 train.py[line:364] INFO: Checkpoint 21 saved !
Epoch 21/50: 100%| 202/202 [01:31<00:00, 2.20im
Epoch 22/50: 18%| 36/202 [00:25<01:21, 2.05im2022-12-08 13:06:54,148 train.py[line:348] INFO: Train step_2140: loss : 157.5484619140625,loss xy :
Epoch 22/50: 38%| 76/202 [00:41<00:45, 2.79im2022-12-08 13:07:09,685 train.py[line:348] INFO: Train step_2160: loss : 107.22421264648438,loss xy :
Epoch 22/50: 57%| 116/202 [00:55<00:33, 2.60im2022-12-08 13:07:24,547 train.py[line:348] INFO: Train step_2180: loss : 107.31367492675781,loss xy :
Epoch 22/50: 77%| 156/202 [01:11<00:17, 2.67im2022-12-08 13:07:39,730 train.py[line:348] INFO: Train step_2200: loss : 107.38504791259766,loss xy :
Epoch 22/50: 97%| 196/202 [01:21<00:01, 5.14im2022-12-08 13:07:49,344 train.py[line:348] INFO: Train step_2220: loss : 137.4841766357422,loss xy :
Epoch 22/50: 100%| 202/202 [01:22<00:00, 5.13im2022-12-08 13:07:51,544 train.py[line:364] INFO: Checkpoint 22 saved !
Epoch 22/50: 100%| 202/202 [01:23<00:00, 2.42im
Epoch 23/50: 17%| 34/202 [00:24<01:21, 2.07im2022-12-08 13:08:16,679 train.py[line:348] INFO: Train step_2240: loss : 106.82681274414062,loss xy :
Epoch 23/50: 37%| 74/202 [00:43<01:00, 2.11im2022-12-08 13:08:35,550 train.py[line:348] INFO: Train step_2260: loss : 190.88702392578125,loss xy :
✓ 1h 14m 35s completed at 8:49 PM

```

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

```

!ls checkpoints

# Choose random test image

import os
test_images = [f for f in os.listdir('test') if f.endswith('.jpg')]
import random
img_path = "test/" + random.choice(test_images);

print(img_path)

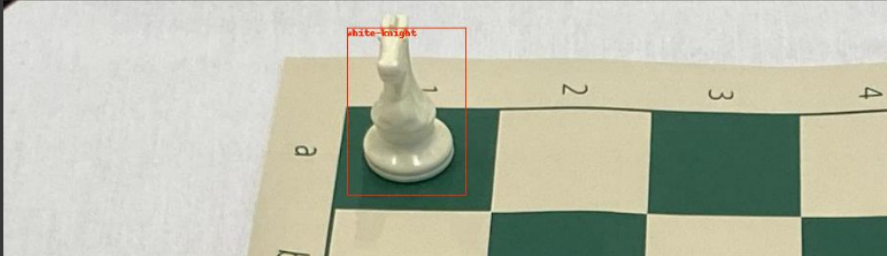
# Run test for a random image using a chosen checkpoints and visualization the result
# Yolov4_epoch*.pth / * is the number between 1 - 50

!python models.py {num_classes} checkpoints/Yolov4_epoch46.pth {img_path} test/_classes.txt

from IPython.display import Image
Image('predictions.jpg')

```

Yolov4\_epoch16.pth Yolov4\_epoch28.pth Yolov4\_epoch3.pth Yolov4\_epoch5.pth  
 Yolov4\_epoch17.pth Yolov4\_epoch29.pth Yolov4\_epoch40.pth Yolov4\_epoch6.pth  
 Yolov4\_epoch18.pth Yolov4\_epoch2.pth Yolov4\_epoch41.pth Yolov4\_epoch7.pth  
 Yolov4\_epoch19.pth Yolov4\_epoch30.pth Yolov4\_epoch42.pth Yolov4\_epoch8.pth  
 Yolov4\_epoch1.pth Yolov4\_epoch31.pth Yolov4\_epoch43.pth Yolov4\_epoch9.pth  
 Yolov4\_epoch20.pth Yolov4\_epoch32.pth Yolov4\_epoch44.pth  
 Yolov4\_epoch21.pth Yolov4\_epoch33.pth Yolov4\_epoch45.pth  
 test/749e9074a77f8d34d86e2218f26cdab4.jpg.rf.8079f8abd9f24ec16e76fcbf18489f46.jpg  
 white-knight: 0.999998  
 save plot results to predictions.jpg



### 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

## b) 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

```
!rm -rf /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip
...
%cd /content/gdrive/My\ Drive/colab
!rm -rf data
%mkdir data
%cd /content/gdrive/My\ Drive/colab/data
!git clone https://github.com/EdricJ/chess_pieces_data v1
!unzip /content/gdrive/My\ Drive/colab/data/chess_pieces_data_v1/ChessPieces.v1-416x416auto-orient-yolov4pytorch.zip -d /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip'''

# 13 classes
!curl -L "https://public.roboflow.com/ds/inVxI0azAW?key=quh4QK7jy0" > roboflow.zip; unzip roboflow.zip -d /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip
!curl -L "https://public.roboflow.com/ds/Hf28P0hvUD?key=sbY5k7Pm7b" > roboflow.zip; unzip roboflow.zip -d /content/gdrive/My\ Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	902	100	902	0	0	28	0 0:00:32 0:00:31 0:00:01 199
100	16.2M	100	16.2M	0	0	519k	0 0:00:32 0:00:32 ---:-- 519k

Archive: roboflow.zip  
 extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/README.dataset.txt  
 extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/README.roboflow.txt  
 creating: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/test/  
 extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/test/0b47311f426ff926578c9d738d683e76.jpg.rf.40183eae584a653181bbd795ba3c353f.jpg  
 extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/test/1c0060ef868bdc326ce5e6389cb6732f.jpg.rf.f02cd668d26a53d9bf0e1497992b3657.jpg  
 extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/test/2f6fb003bb89cd401322a535ac642f65.jpg.rf.66c0a46773a9cd583fb96c3df41a9e0c.jpg  
 extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/test/410993714e325a1de3e394ffe860df3a.jpg.rf.657c49ca295ef54da23469189070a075.jpg  
 extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data\_unzip/test/4e3117459d759798537eb52cf5bf534d.jpg.rf.ec961b62d4b0e131fae760ed1f80836b.jpg

Link: <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

```
#epoch - how long to train for

%cd /content/gdrive/My\ Drive/colab/pytorch-YOLOv4
!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./yolov4.pth -classes {num_classes} -dir ./train -epochs 50

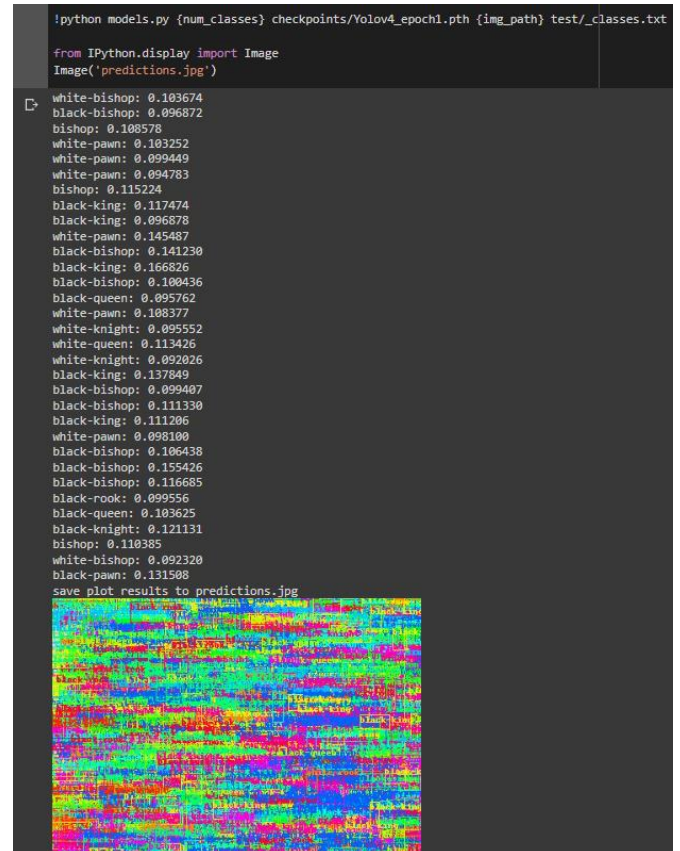
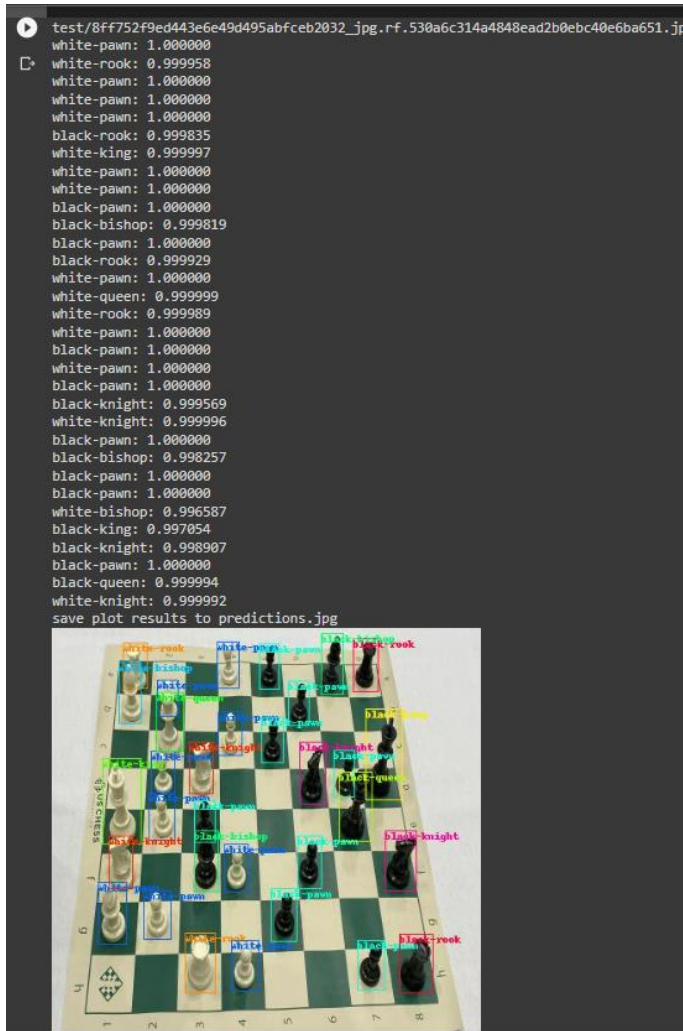
# continue training from the checkpoints
#Yolov4_epoch*.pth / * is the number between 1 - 50

#!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./checkpoints/Yolov4_epoch1.pth -classes {num_classes} -dir ./train -epochs 50
```

/content/gdrive/My Drive/colab/pytorch-YOLOv4



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

```
!rm -rf /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip

# 13 classes
!curl -L "https://public.roboflow.com/ds/Mf28POhvUD?key=sbYSk7Pm7b" > roboflow.zip; unzip roboflow.zip -d /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip; rm roboflow.zip
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	902	100	902	0	444	0	0:00:02
100	77.7M	100	77.7M	0	9.8M	0	0:00:07

```
Archive: roboflow.zip
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/README.dataset.txt
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/README.roboflow.txt
    creating: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/0b47311f426ff926578c9d738d683e76.jpg.rf.0b55f43ac16aa65c889558d8ea757072.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/1c0060ef868bdc326ce5e6389cb6732f.jpg.rf.9ce88078ea356949f4ab7ad9cfd6c62d.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/2f6fb003bb89cd401322a535acba42f65.jpg.rf.91ad9df05bd1f86bab67c8368ae5e4ad.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/410993714e325a1de3e394ffe860df3a.jpg.rf.519bf0f0db5e38cd44caelcfecb98536.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/4e3117459d759798537eb52cf5bf534d.jpg.rf.5b99421bf416463a8c75cfd07f8a68d1.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/5a35ba2ec3e0db2b12b1758a8ac29aa.jpg.rf.a907af85301c729635d6ab1c31eb31b2.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/654bb8835258b26c466b1c19893df451.jpg.rf.95aad224dd31ab256cb2bcbff02a34dd.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/685b860d412b91f5d4f7f9e643b84452.jpg.rf.5ba8dc0b5d2585d01b28089debd42cd6.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/73a38a5c8f8f1b9f093f304660d5326.jpg.rf.2d2fa2f4b419d9f2a57fb82d38d8bc6b.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/749e074a77f8d34d86e2218f26cdab4.jpg.rf.8079f8abd9f24ec16e76fcbf18489f46.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/7a34d862023048917b28bdcfd3b5572b.jpg.rf.71653deb6fe88ad472dabea12353373d.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/8ff752f9ed443e6e49d495abfceb2032.jpg.rf.c3e91277eeaa9c26328e39a6f0285189.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0159.JPG.rf.1cf4f243b5072d63e492711720df35f7.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0169.JPG.rf.b1530b71278953ad465d06863135c71e.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0170.JPG.rf.6e336797b63833d78997207d352a44fc.jpg
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/_annotations.txt
  extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/62873d8be092c11b30e08817b37556f-deg.rf.e431124b320d57e0237dffe469a35c1fb-deg
```

Link: <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

```
%cd /content/gdrive/My Drive/colab/pytorch-YOLOv4
!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./yolov4.pth -classes {num_classes} -dir ./train -epochs 50

# continue training from the checkpoints
# Yolov4_epoch*.pth / * is the number between 1 - 50

#!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./checkpoints/Yolov4_epoch1.pth -classes {num_classes} -dir ./train -epochs 50
```

```
/content/gdrive/My Drive/colab/pytorch-YOLOv4
log file path:log/log_2022-12-08_12-35-05.txt
2022-12-08 12:35:05.421 train.py[line:435] INFO: Using device cuda
2022-12-08 12:35:09.031 train.py[line:268] INFO: Starting training:
  Epochs: 50
  Batch size: 2
  Subdivisions: 1
  Learning rate: 0.001
  Training size: 202
  Validation size: 58
  Checkpoints: True
  Device: cuda
  Images size: 608
  Optimizer: adam
  Dataset classes: 13
  Train label path:train.txt
  Pretrained:

Epoch 1/50: 0% | 0/202 [00:00<?, ?img/s]/usr/local/lib/python3.8/dist-packages/torch/nn/_reduction.py:43: UserWarning: size_average
  warnings.warn(warning.format(ret))
Epoch 1/50: 19% | 38/202 [00:24<01:04, 2.55img/s]/usr/local/lib/python3.8/dist-packages/torch/optim/lr_scheduler.py:224: UserWarning: To get
  warnings.warn("To get the last learning rate computed by the scheduler, "
2022-12-08 12:35:33.899 train.py[line:348] INFO: Train step 20: loss : 31168.21484375,loss xy : 10.912572860717773,loss wh : 1.33473157882696
Epoch 1/50: 39% | 78/202 [00:41<01:02, 1.97img/2022-12-08 12:35:51.641 train.py[line:348] INFO: Train step 40: loss : 31319.87109375,loss
Epoch 1/50: 58% | 118/202 [01:00<00:46, 1.80img/2022-12-08 12:36:10.952 train.py[line:348] INFO: Train step 60: loss : 31713.013671875,loss
```

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

```

!ls checkpoints

# Choose random test image

import os
test_images = [f for f in os.listdir('test') if f.endswith('.jpg')]
import random
img_path = "test/" + random.choice(test_images);

print(img_path)

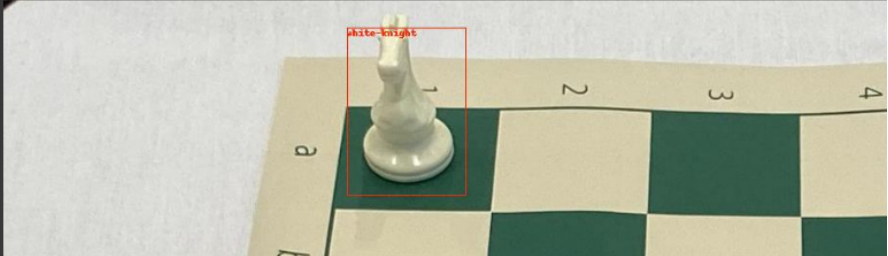
# Run test for a random image using a chosen checkpoints and visualization the result
# Yolov4_epoch*.pth / * is the number between 1 - 50

!python models.py {num_classes} checkpoints/Yolov4_epoch46.pth {img_path} test/_classes.txt

from IPython.display import Image
Image('predictions.jpg')

```

Yolov4\_epoch16.pth Yolov4\_epoch28.pth Yolov4\_epoch3.pth Yolov4\_epoch5.pth  
 Yolov4\_epoch17.pth Yolov4\_epoch29.pth Yolov4\_epoch40.pth Yolov4\_epoch6.pth  
 Yolov4\_epoch18.pth Yolov4\_epoch2.pth Yolov4\_epoch41.pth Yolov4\_epoch7.pth  
 Yolov4\_epoch19.pth Yolov4\_epoch30.pth Yolov4\_epoch42.pth Yolov4\_epoch8.pth  
 Yolov4\_epoch1.pth Yolov4\_epoch31.pth Yolov4\_epoch43.pth Yolov4\_epoch9.pth  
 Yolov4\_epoch20.pth Yolov4\_epoch32.pth Yolov4\_epoch44.pth  
 Yolov4\_epoch21.pth Yolov4\_epoch33.pth Yolov4\_epoch45.pth  
 test/749e9074a77f8d34d86e2218f26cdab4.jpg.rf.8079f8abd9f24ec16e76fcbf18489f46.jpg  
 white-knight: 0.999998  
 save plot results to predictions.jpg



### 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

```

rm -rf /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip

%cd /content/gdrive/My Drive/colab
rm -rf data
mkdir data
%cd /content/gdrive/My Drive/colab/data
!git clone https://github.com/EdricJ/chess_pieces_data_v1
!unzip /content/gdrive/My Drive/colab/data/chess_pieces_data_v1/ChessPieces.v1-416x416auto-orient.yolov4pytorch.zip -d /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip

extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0169.JPG.rf.1de291413bb78e8f8f0eaa8ffac38b06.jpg
extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/IMG_0170.JPG.rf.480e7164cb4727f6654402882f0ce942.jpg
extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/_annotations.txt
extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/_classes.txt
extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/a3863d0be600c21b20ac88817b2c56f.jpg.rf.0413d5178136ace55f588df9556c060a.jpg
extracting: /content/gdrive/My Drive/colab/pytorch-YOLOv4/data_unzip/test/b4ff4132c8c85da97d8bf9a2a4ed3e3d.jpg.rf.ec798769b4818025b7652ca6aab9307e.jpg

```

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

```

%cd /content/gdrive/My Drive/colab/pytorch-YOLOv4
!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./yolov4.pth -classes {num_classes} -dir ./train -epochs 50

# continue training from the checkpoints
# Yolov4_epoch*.pth / * is the number between 1 - 50

!python train.py -b 2 -s 1 -l 0.001 -g 0 -pretrained ./checkpoints/Yolov4_epoch1.pth -classes {num_classes} -dir ./train -epochs 50

```

```

... Epoch 19/50: 21% | 42/202 [00:11:00:36, 4.38img2022-12-08 14:36:09,603 train.py[line:348] INFO: Train step 1840: loss : 161.73110961914062,loss xy : 23.564864825878906,loss wh : 5.172828568884717,loss obj : 118.472702
Epoch 19/50: 41% | 82/202 [00:20:00:25, 4.64img2022-12-08 14:36:18,125 train.py[line:348] INFO: Train step 1860: loss : 126.082061328125,loss xy : 15.224111557006836,loss wh : 0.5662644844055176,loss obj : 100.2450625
Epoch 19/50: 60% | 122/202 [00:28:00:16, 4.93img2022-12-08 14:36:26,637 train.py[line:348] INFO: Train step 1880: loss : 169.5964813222422,loss xy : 33.68889236450195,loss wh : 0.040347170829773,loss obj : 108.134780512
Epoch 19/50: 80% | 162/202 [00:37:00:09, 4.25img2022-12-08 14:36:35,597 train.py[line:348] INFO: Train step 1900: loss : 142.7988477783203,loss xy : 34.33832550048828,loss wh : 0.7728935480117798,loss obj : 97.19023132
Epoch 19/50: 100% | 202/202 [00:47:00:00, 4.41img2022-12-08 14:36:47,928 train.py[line:364] INFO: Checkpoint 19 saved !
Epoch 19/50: 100% | 202/202 [00:50:00:00, 3.98img
Epoch 20/50: 0% | 0/202 [00:00:00, ?img/s]2022-12-08 14:36:50,953 train.py[line:348] INFO: Train step 1920: loss : 186.85708618164062,loss xy : 84.603515625,loss wh : 3.516606330871582,loss obj : 88.13993072509766
Epoch 20/50: 20% | 40/202 [00:11:00:37, 4.29img2022-12-08 14:36:59,819 train.py[line:348] INFO: Train step 1940: loss : 221.7286071773438,loss xy : 93.26774597167969,loss wh : 1.3702222180840942,loss obj : 104.154708
Epoch 20/50: 40% | 80/202 [00:20:00:26, 4.60img2022-12-08 14:37:08,398 train.py[line:348] INFO: Train step 1960: loss : 153.11441040030062,loss xy : 56.67044772330867,loss wh : 1.4797120083129083,loss obj : 83.5087672
Epoch 20/50: 50% | 120/202 [00:28:00:18, 4.51img2022-12-08 14:37:17,199 train.py[line:348] INFO: Train step 1980: loss : 186.00531005859375,loss xy : 62.909988294677734,loss wh : 2.7682135185133057,loss obj : 94.5215606
Epoch 20/50: 70% | 160/202 [00:37:00:08, 4.71img2022-12-08 14:37:25,766 train.py[line:348] INFO: Train step 2000: loss : 243.6894073486328,loss xy : 103.8678970336914,loss wh : 5.061160087585449,loss obj : 115.32882698
Epoch 20/50: 90% | 200/202 [00:45:00:00, 4.95img2022-12-08 14:37:33,960 train.py[line:348] INFO: Train step 2020: loss : 216.96798706054688,loss xy : 57.20146179199219,loss wh : 2.556077718734741,loss obj : 118.3132476
Epoch 20/50: 100% | 202/202 [00:46:00:00, 4.93img2022-12-08 14:37:37,507 train.py[line:364] INFO: Checkpoint 20 saved !
Epoch 20/50: 100% | 202/202 [00:49:00:00, 4.68img
Epoch 21/50: 10% | 20/202 [00:11:00:35, 4.57img2022-12-08 14:37:49,001 train.py[line:348] INFO: Train step 2040: loss : 140.74059093320312,loss xy : 31.69937515250789,loss wh : 2.2826547622680664,loss obj : 94.4570781
Epoch 21/50: 30% | 78/202 [00:19:00:26, 4.74img2022-12-08 14:37:57,560 train.py[line:348] INFO: Train step 2060: loss : 344.434814453125,loss xy : 150.27369689941406,loss wh : 3.9661455154418945,loss obj : 123.9546800
Epoch 21/50: 50% | 118/202 [00:28:00:17, 4.80img2022-12-08 14:38:06,205 train.py[line:348] INFO: Train step 2080: loss : 126.73948669433594,loss xy : 48.12357711791992,loss wh : 1.335530759040527,loss obj : 69.95279693

```

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

```

Step 06. Test the results

# Check the trained weights

ls checkpoints

# Choose random test image

import os
test_images = [f for f in os.listdir('test') if f.endswith('.jpg')]
import random
img_path = "test/" + random.choice(test_images);

print(img_path)

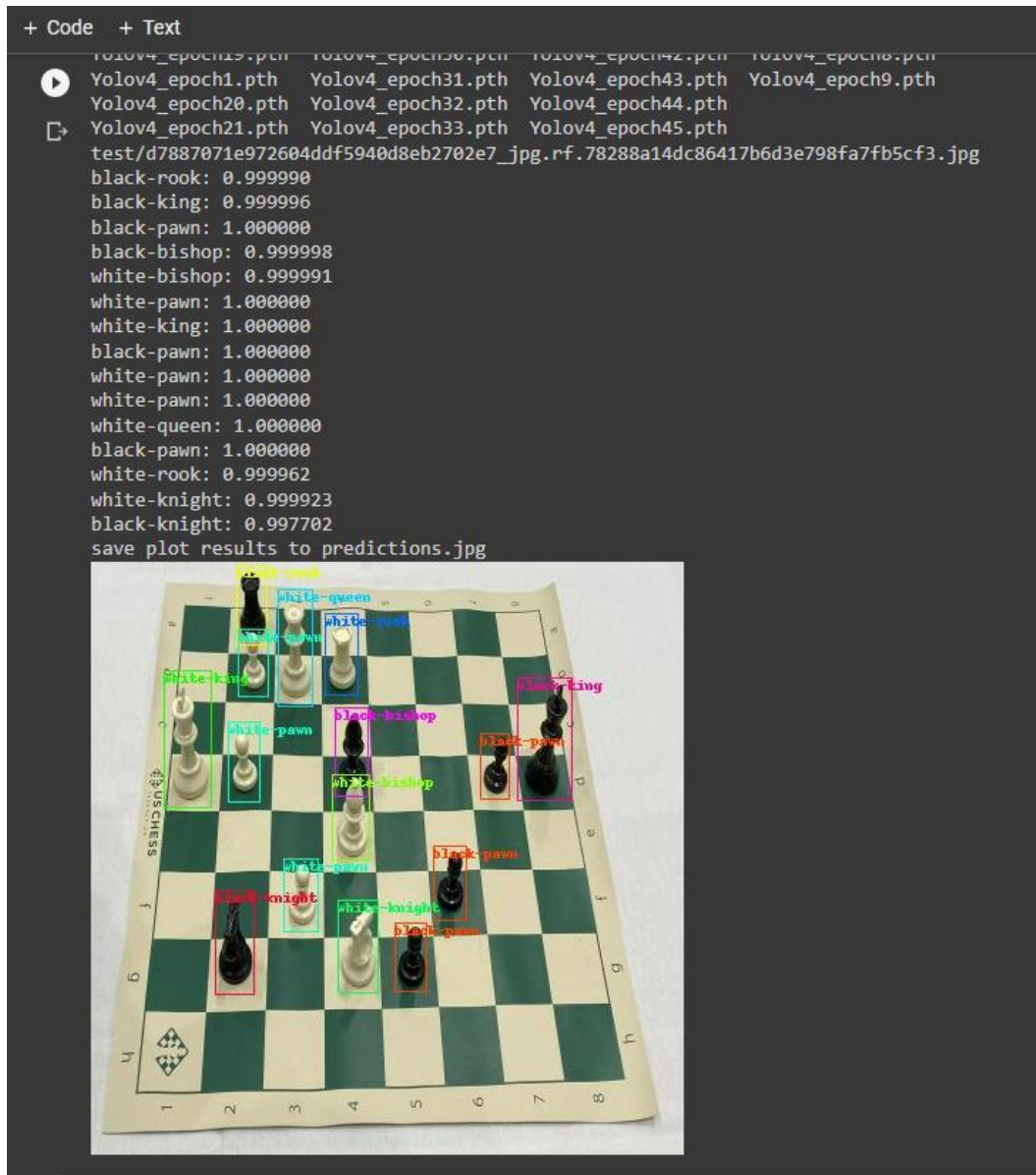
# Run test for a random image using a chosen checkpoints and visualization the result

!python models.py {num_classes} checkpoints/Volov4_epoch46.pth {img_path} test/_classes.txt

from IPython.display import Image
Image('predictions.jpg')

```

Volov4_epoch10.pth	Volov4_epoch22.pth	Volov4_epoch34.pth	Volov4_epoch46.pth
Volov4_epoch11.pth	Volov4_epoch23.pth	Volov4_epoch35.pth	Volov4_epoch47.pth
Volov4_epoch12.pth	Volov4_epoch24.pth	Volov4_epoch36.pth	Volov4_epoch48.pth
Volov4_epoch13.pth	Volov4_epoch25.pth	Volov4_epoch37.pth	Volov4_epoch49.pth
Volov4_epoch14.pth	Volov4_epoch26.pth	Volov4_epoch38.pth	Volov4_epoch40.pth
Volov4_epoch15.pth	Volov4_epoch27.pth	Volov4_epoch39.pth	Volov4_epoch50.pth
Volov4_epoch16.pth	Volov4_epoch28.pth	Volov4_epoch30.pth	Volov4_epoch51.pth
Volov4_epoch17.pth	Volov4_epoch29.pth	Volov4_epoch40.pth	Volov4_epoch60.pth
Volov4_epoch18.pth	Volov4_epoch20.pth	Volov4_epoch41.pth	Volov4_epoch70.pth
Volov4_epoch19.pth	Volov4_epoch30.pth	Volov4_epoch42.pth	Volov4_epoch80.pth
Volov4_epoch10.pth	Volov4_epoch31.pth	Volov4_epoch43.pth	Volov4_epoch90.pth
Volov4_epoch20.pth	Volov4_epoch32.pth	Volov4_epoch44.pth	
Volov4_epoch21.pth	Volov4_epoch33.pth	Volov4_epoch45.pth	
test/d7887071e972684ddf5940d8eb2702e7.jpg.rf.78288a14dc86417b6d3e798fa7fb5cf3.jpg			
black-rook: 0.999990			
black-king: 0.999996			
black-pawn: 1.000000			
black-bishop: 0.999998			
white-bishop: 0.999991			



### 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

### **III. 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/>
- [2] <https://viblo.asia/p/thiet-ke-mang-quantum-neural-network-voi-pytorch-va-qiskit-m2vJPwna4eK>
- [3] <https://phamdinhhkhanh.github.io/2019/08/10/PytorchTutorial1.html>
- [4] <https://github.com/Tianxiaomo/pytorch-YOLOv4/issues/427>