

# BÁO CÁO YOLO

## PHẦN 1

---

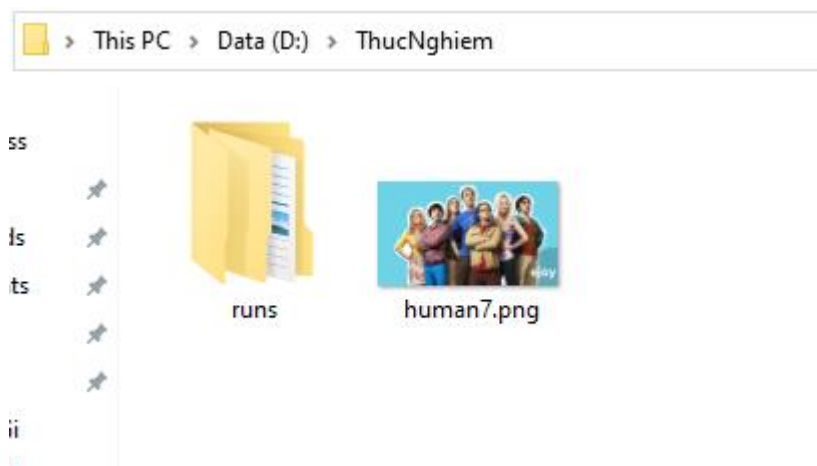
- Bước 1: Truy cập vào: <https://github.com/ultralytics/yolov5>
- Bước 2: Làm theo hướng dẫn để git clone về máy tính:

▼ Install

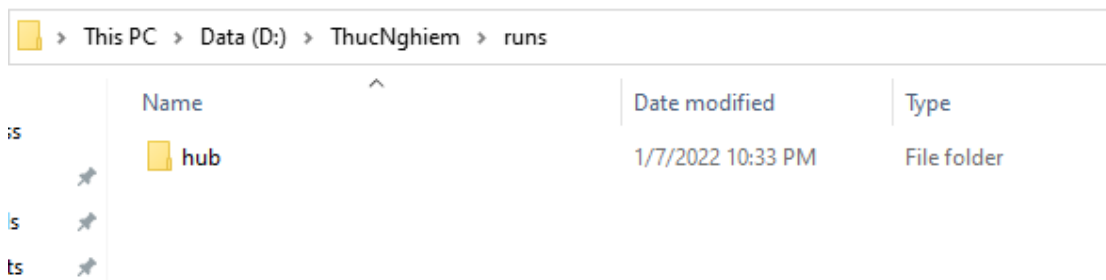
Clone repo and install [requirements.txt](#) in a **Python>=3.6.0** environment, including **PyTorch>=1.7**.

```
git clone https://github.com/ultralytics/yolov5 # clone
cd yolov5
pip install -r requirements.txt # install
```

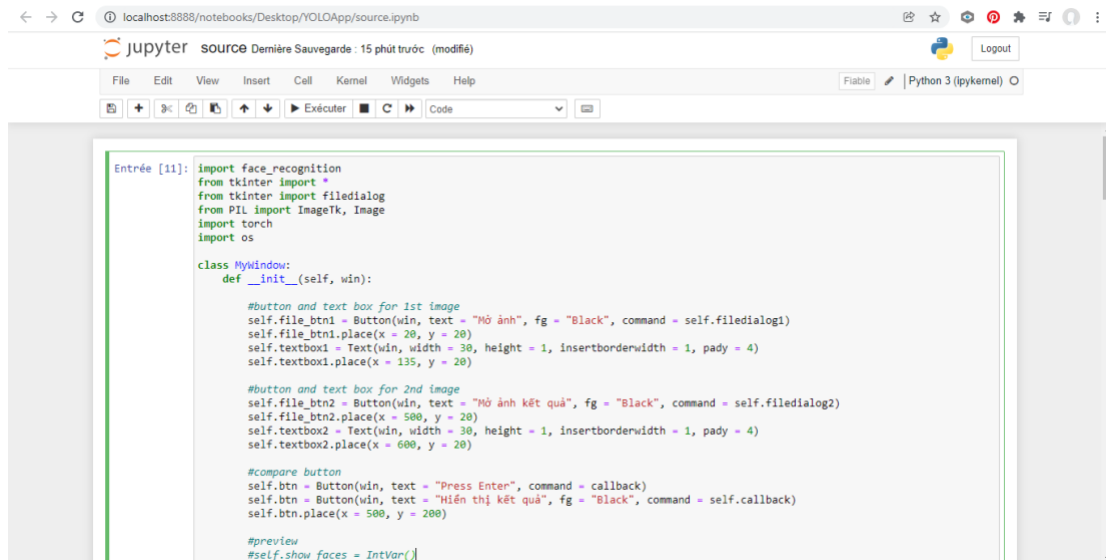
- Bước 3: Vào ổ đĩa D của PC. Tạo thư mục có tên “ThucNghiem”
- Bước 4: Trong thư mục “ThucNghiem”, ta tạo một ảnh bìa cho app có tên là “human7.png”.



- Bước 5: Trong thư mục “ThucNghiem”, ta tạo một thư mục con có tên “runs”.
- Bước 6: Trong thư mục “runs”, ta tạo một thư mục con có tên “hub”. Đây sẽ là nơi lưu trữ file ảnh đã được nhận dạng vật thể.



- Bước 7: Mở file source.ipynb trên Jupyter Notebook, ta sẽ thấy được:



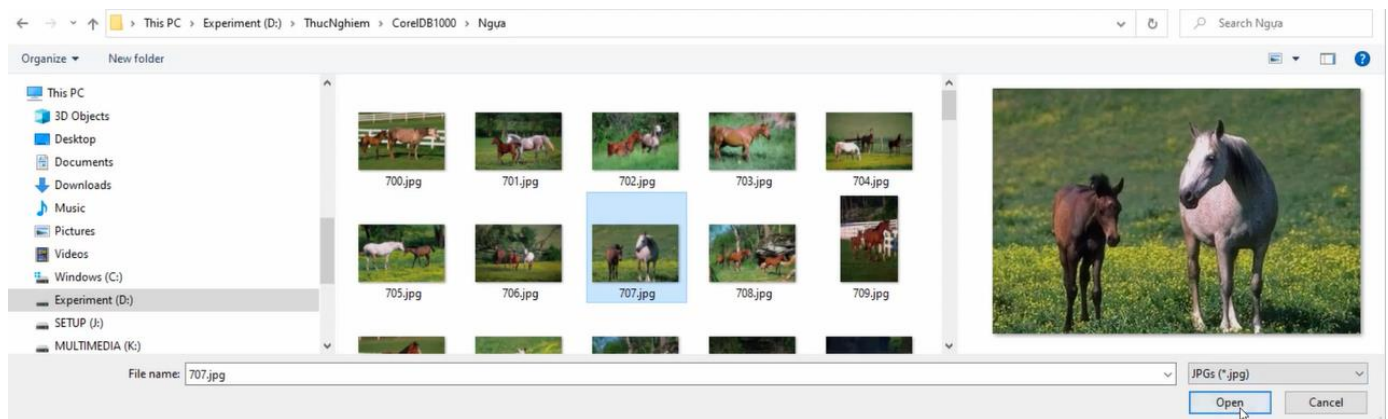
- Bước 8: Bấm vào nút “Executer”/”Run” để chạy file code, ta sẽ thấy một cửa sổ bật lên:



- Bước 9: Click nút “Mở ảnh”



- Bước 9: Chọn ảnh cần xử lý, rồi click nút “Open”



- Bước 10: Click nút “Hiển thị kết quả”

Phát hiện nội dung trong ảnh sử dụng thư viện YOLO

Mở ảnh

D: /ThucNghiem/CorelDB1000/Ngựa

Mở ảnh kết quả

Ảnh:



Ảnh kết quả:

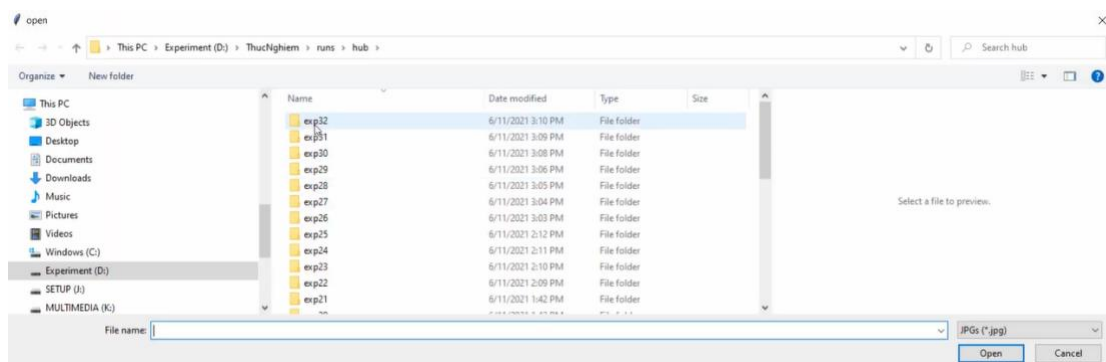
Hiển thị kết quả

Thông tin:

- Bước 11: Click vào nút “Mở ảnh kết quả”

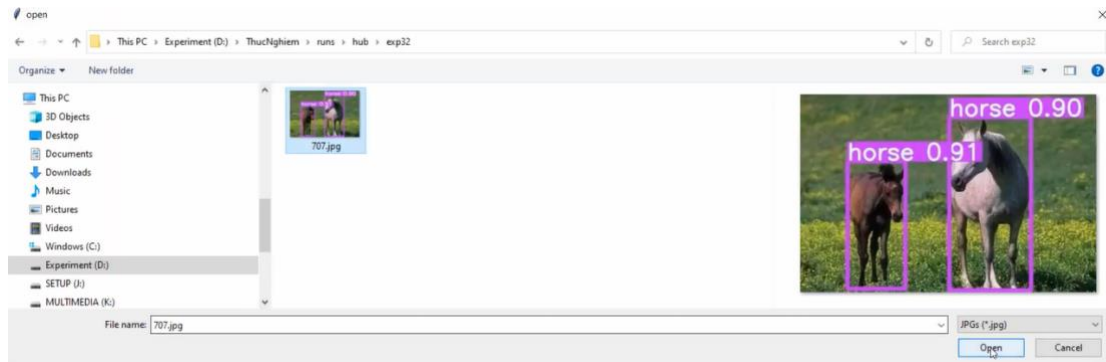


- Bước 12: Chọn thư mục vừa được tạo trong thư mục “hub”

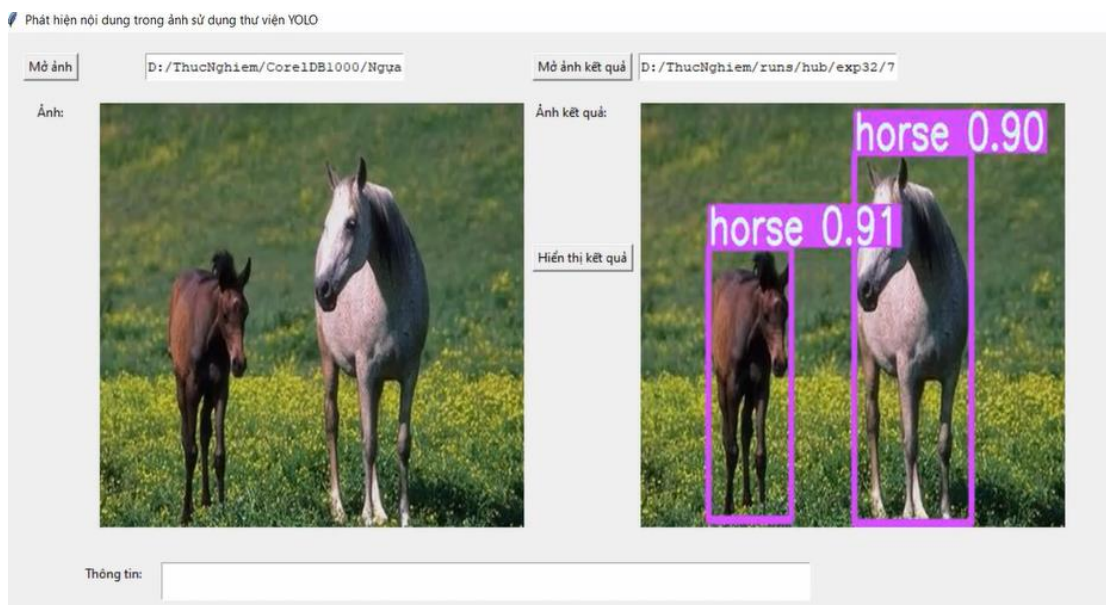




- Bước 13: Chọn ảnh kết quả trong thư mục đó, rồi click nút “Open”



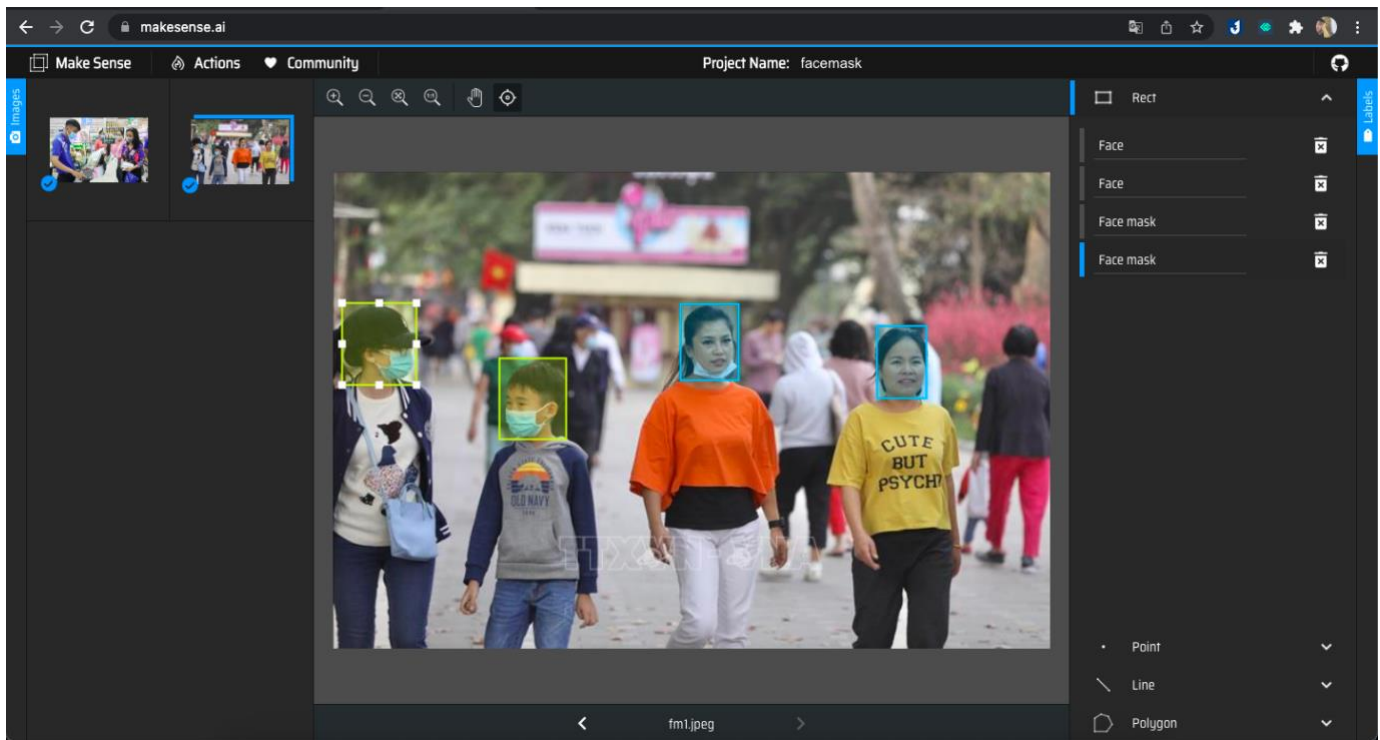
- Bước 14: Ta thấy được kết quả cuối cùng của chương trình.



## PHẦN 2

### Bước 1: Chuẩn bị data để training

- Tìm kiếm thật nhiều hình ảnh có liên quan đến đối tượng mình cần train.
- Truy cập website [makesense.ai](https://makesense.ai) để gắn nhãn cho các đối tượng có trong hình.
- Tạo nhãn (label) theo số lượng muốn phân loại, sau đó gắn nhãn lên từng đối tượng sao cho phù hợp.



- Sau khi gắn nhãn xong các hình ảnh, chọn Actions -> Export Annotations -> Tại khung hình chữ nhật, tick vào “A .zip package containing files in YOLO format.” -> Export là xong.

## Bước 2: Chuẩn bị file yolov4-custom.cfg

- Tải file [yolov4-custom.cfg](#) về máy.
- Xác định số class cần train. Vì ở đây train ‘Face’ và ‘Face mask’ nên chỉ cần 2 class. Thay thế các dòng có “classes=X” từ X thành số class cần train.
- Lưu file lại.

## Bước 3: Chuẩn bị Makefile

- Tải [Makefile](#) về máy.
- Mở lên, sửa dòng 1 thành GPU=1, sửa dòng 2 thành CUDNN=1, sửa dòng 4 thành OPENCV=1.
- Lưu lại.

## Bước 4: Tạo Notebook mới trên Google Colab

- Truy cập [Google Colab](#) và tạo Notebook mới.
- Tại Runtime, chọn Change runtime type -> GPU để chạy nhanh hơn.

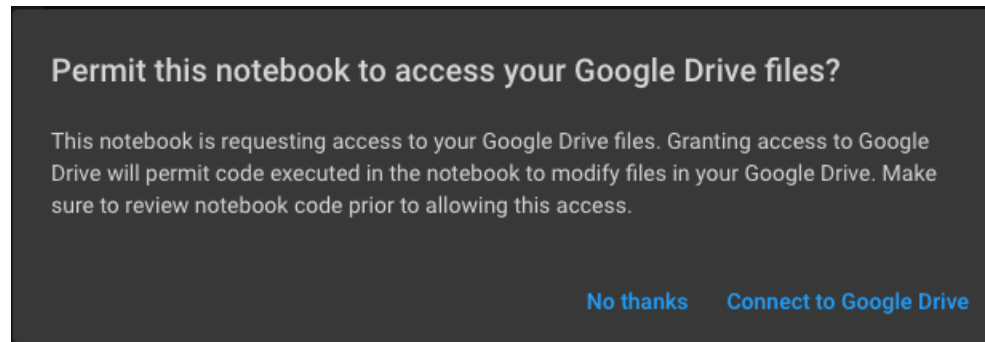
## Bước 5: Train YOLOv4 trên Google Colab

### 5.1. Connect với Google Drive

- Tạo code block mới và nhập đoạn mã sau:

```
from google.colab import drive
drive.mount('/content/gdrive')
```

- Sau khi nhập xong, nhấn vào nút chạy ở bên đầu code block, sau đó sẽ Colab sẽ hiện ra mục như sau:



- Nhấn vào Connect to Google Drive và sẽ hiện ra một tab mới, đăng nhập tài khoản Google và nhấn Allow là xong.

## 5.2. Tải mã nguồn của YOLO về Google Drive

---

- Tạo code block mới và nhập lệnh:

```
!rm -rf darknet
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO
!git clone https://github.com/AlexeyAB/darknet
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO
!rm -rf data
!mkdir data
```

- Chạy code.

## 5.3. Upload các file liên quan lên Google Drive

---

Truy cập vào thư mục darknet trên Google Drive, upload các file ở Bước 1, Bước 2 và Bước 3 lên theo thứ tự như sau:

- *data.zip* vào thư mục data.
- *yolov4-custom.cfg* vào thư mục cfg.
- *Makefile* vào thư mục darknet.

## 5.4. Giải nén lấy data

---

- Tạo code block mới và nhập lệnh:

```
%cd /content/gdrive/MyDrive/ColabNotebooks/
19127251_19127251_YOLO/darknet/data
!unzip data.zip
```

## 5.5. Tạo file yolo.names

---

- Tạo code block mới và nhập lệnh:

```
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO/darknet
!echo "Face" > yolo.names
!echo "Face mask" >> yolo.names
```

- File này chứa tên các class. Nếu có 2 class thì tạo 2 dòng.

## 5.6. Tạo file train.txt và val.txt để chứa danh sách các file ảnh.

---



- File *train.txt* chứa danh sách file dùng để train, *val.txt* chứa danh sách các file để val.
- Tạo code block mới và nhập:

```
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO/darknet

import glob2
import math
import os
import numpy as np

files = []
for ext in ["*.png", "*.jpeg", "*.jpg"]:
    image_files = glob2.glob(os.path.join("data/data/", ext))
    files += image_files

nb_val = math.floor(len(files)*0.2)
rand_idx = np.random.randint(0, len(files), nb_val)

# Tạo file train.txt
with open("train.txt", "w") as f:
    for idx in np.arange(len(files)):
        if (os.path.exists(files[idx][: -3] + "txt")):
            f.write(files[idx]+'\\n')

# Tạo file vali.txt
with open("val.txt", "w") as f:
    for idx in np.arange(len(files)):
        if (idx in rand_idx) and (os.path.exists(files[idx][: -3] + "txt")):
            f.write(files[idx]+'\\n')
```

- Chạy thử, nếu trong thư mục darknet xuất hiện 2 file *train.txt* và *val.txt* thì đi qua bước tiếp theo.

## 5.7. Tạo file yolo.data

- File này chứa tham số train để YOLO đọc.
- Tạo code block mới và nhập:

```
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO/darknet
!mkdir backup
!echo classes=2 > yolo.data
!echo train=train.txt >> yolo.data
!echo valid=val.txt >> yolo.data
!echo names=yolo.names >> yolo.data
!echo backup=backup >> yolo.data
```

- Chạy thử, kiểm tra nếu xuất hiện yolo.data trong thư mục darknet thì tiếp tục.

## 5.8. Biên dịch mã nguồn darknet

- Tạo code block mới và nhập lệnh:

```
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO/darknet
```

```
!rm darknet
!make
```

## 5.9. Tải pretrain weights và train

- Tạo code block mới và nhập lệnh:

```
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO/darknet
!wget
https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
%cd /content/gdrive/MyDrive/ColabNotebooks/19127251_19127251_YOLO/darknet
!./darknet detector train yolo.data cfg/yolov4-custom.cfg yolov4.conv.137 -
dont_show
```

- Chạy và bắt đầu train.

**\* Lưu ý:** Đây chỉ là bước train lúc ban đầu, nếu train lần kế tiếp thì bỏ qua các bước này (từ 5.1 đến 5.9), mà phải bắt đầu từ bước 5.10.

## 5.10. Tải pretrain weights và train

- Tạo code block mới và nhập lệnh:

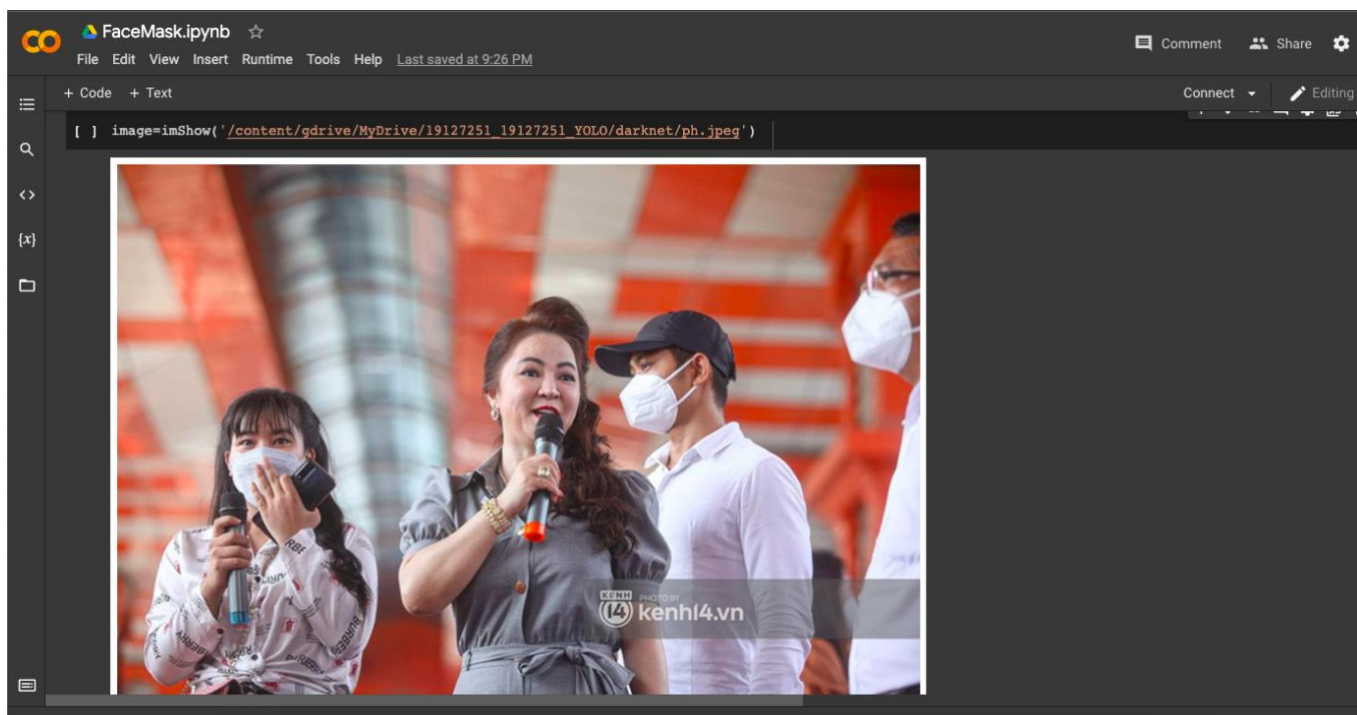
```
!chmod +x ./darknet
%cd /content/gdrive/MyDrive/19127251_19127251_YOLO/darknet
!./darknet detector train yolo.data cfg/yolov4-custom.cfg backup/yolov4-
custom_last.weights -dont_show
%cd /content/gdrive/MyDrive/19127251_19127251_YOLO/darknet
!./darknet detector map yolo.data cfg/yolov4-custom.cfg backup/yolov4-
custom_last.weights -iou_thresh 0.5
%cd /content/gdrive/MyDrive/19127251_19127251_YOLO/darknet/cfg
!sed -i 's/batch=64/batch=1/' yolov4-custom.cfg
!sed -i 's/subdivisions=16/subdivisions=1/' yolov4-custom.cfg
%cd ..
```

- Cứ sau mỗi lần chạy, ta thu được file weight mới nhất là *yolov4-custom\_last.weights*.

## 5.11. Viết hàm upload và show ảnh

- Tạo code block mới và viết hàm show ảnh.
- Sau đó upload thử một ảnh liên quan lên Colab (lưu ý rằng ảnh này sẽ bị xóa bỏ sau khi kết thúc phiên làm việc với Colab).
- Show ảnh gốc ra màn hình:

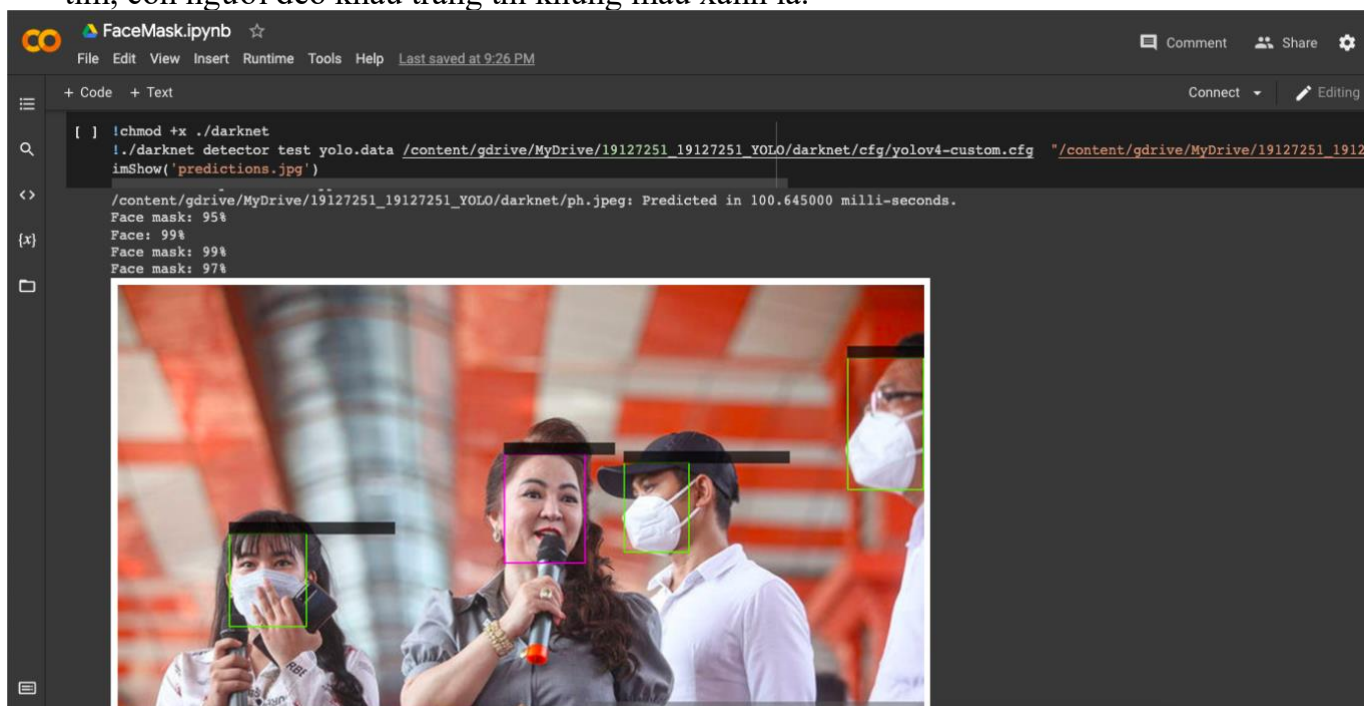
```
image=imshow('<path_của_ảnh>')
```



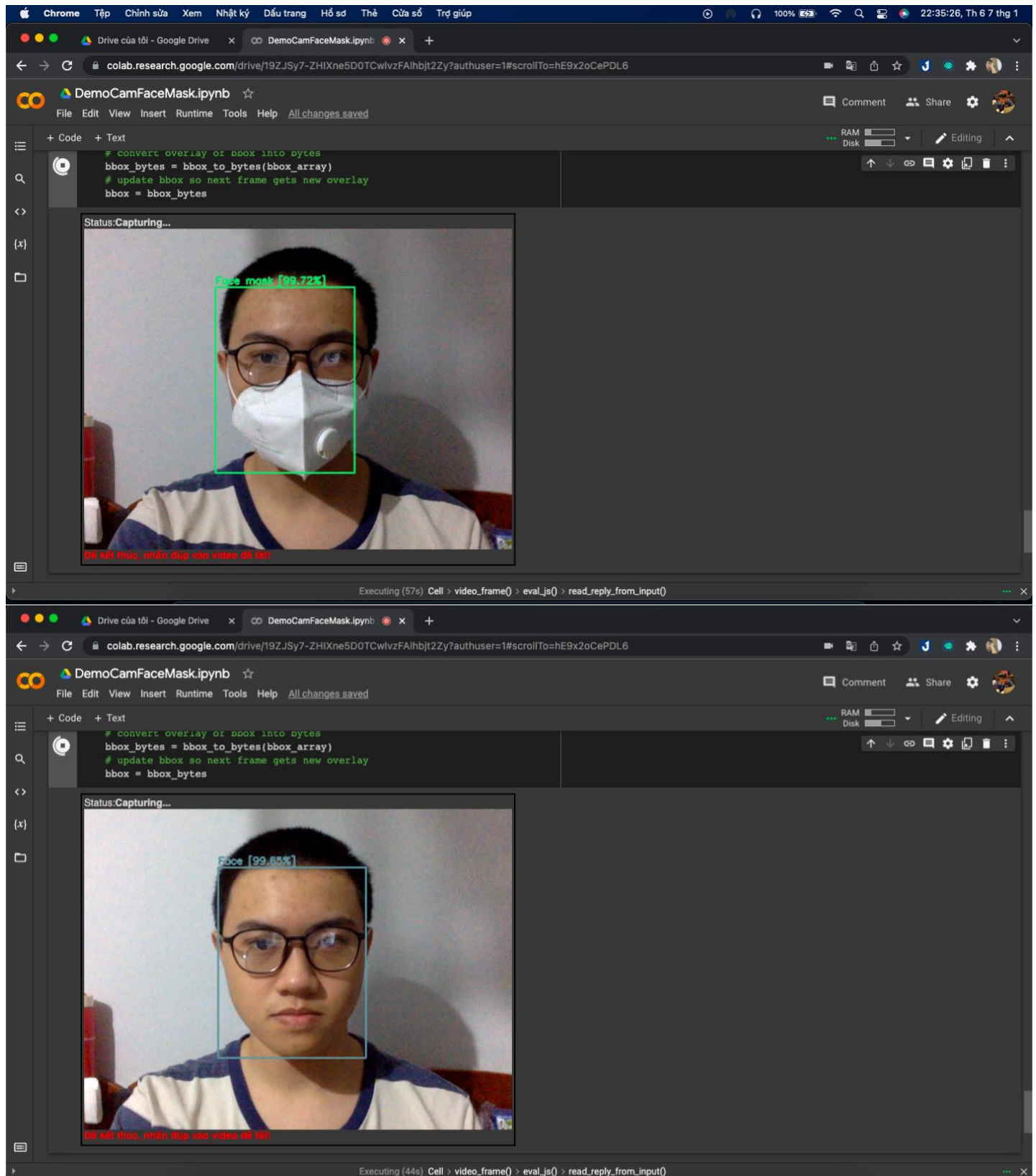
- Nhập dòng lệnh sau:

```
!chmod +x ./darknet
!./darknet detector test yolo.data
/content/gdrive/MyDrive/19127251_19127251_YOLO/darknet/cfg/yolov4-custom.cfg
"/content/gdrive/MyDrive/19127251_19127251_YOLO/darknet/backup/yolov4-
custom_final.weights" <path_của_ảnh> -thresh 0.3 -dont_show
imshow('predictions.jpg')
```

- Nhận được kết quả bằng text và ảnh. Hình này được lưu dưới tên *predictions.jpg* trong thư mục darknet. Ta thấy được người không đeo khẩu trang sẽ được nhận diện bằng khung màu tím, còn người đeo khẩu trang thì khung màu xanh lá.



## \* TEST LIVE WEBCAM (File DemoCamFaceMask.ipynb)



Nguồn tham khảo:

<https://github.com/ultralytics/yolov5>

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

[https://github.com/thangnch/MIAI\\_Colab\\_Realtime/blob/main/WebcamColab.ipynb](https://github.com/thangnch/MIAI_Colab_Realtime/blob/main/WebcamColab.ipynb)