**Nhóm U: Đinh Trọng Đạt, Nguyễn Trung Nguyên, Mai Thanh Lâm**

**Real-Time Face Mask Detector với môi trường Python, thư viện OpenCV và Keras.**

* Nhóm mình sẽ xây dựng một hệ thống thời gian thực để phát hiện xem người trên webcam có đeo khẩu trang hay không. Chúng mình sẽ training mô hình nhận diện khuôn mặt bằng thư viện Keras và OpnCV.
* **Dowload source code:** https://github.com/Dat0309/Face-Mask-Detection-PPNCKH.git

**Mô hình bài toán:**

output

Max-Pool

Convolution

Input

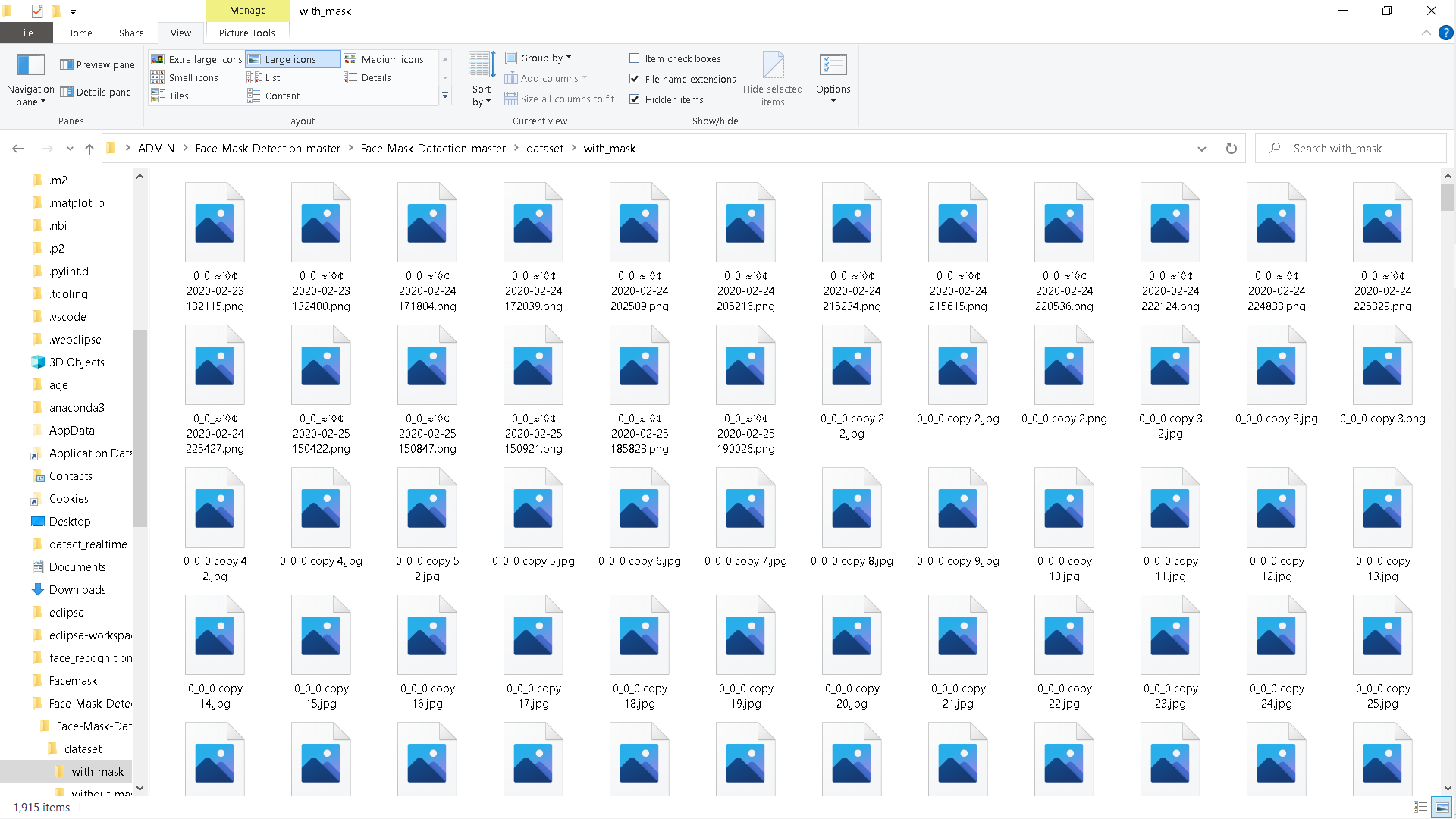
Fully

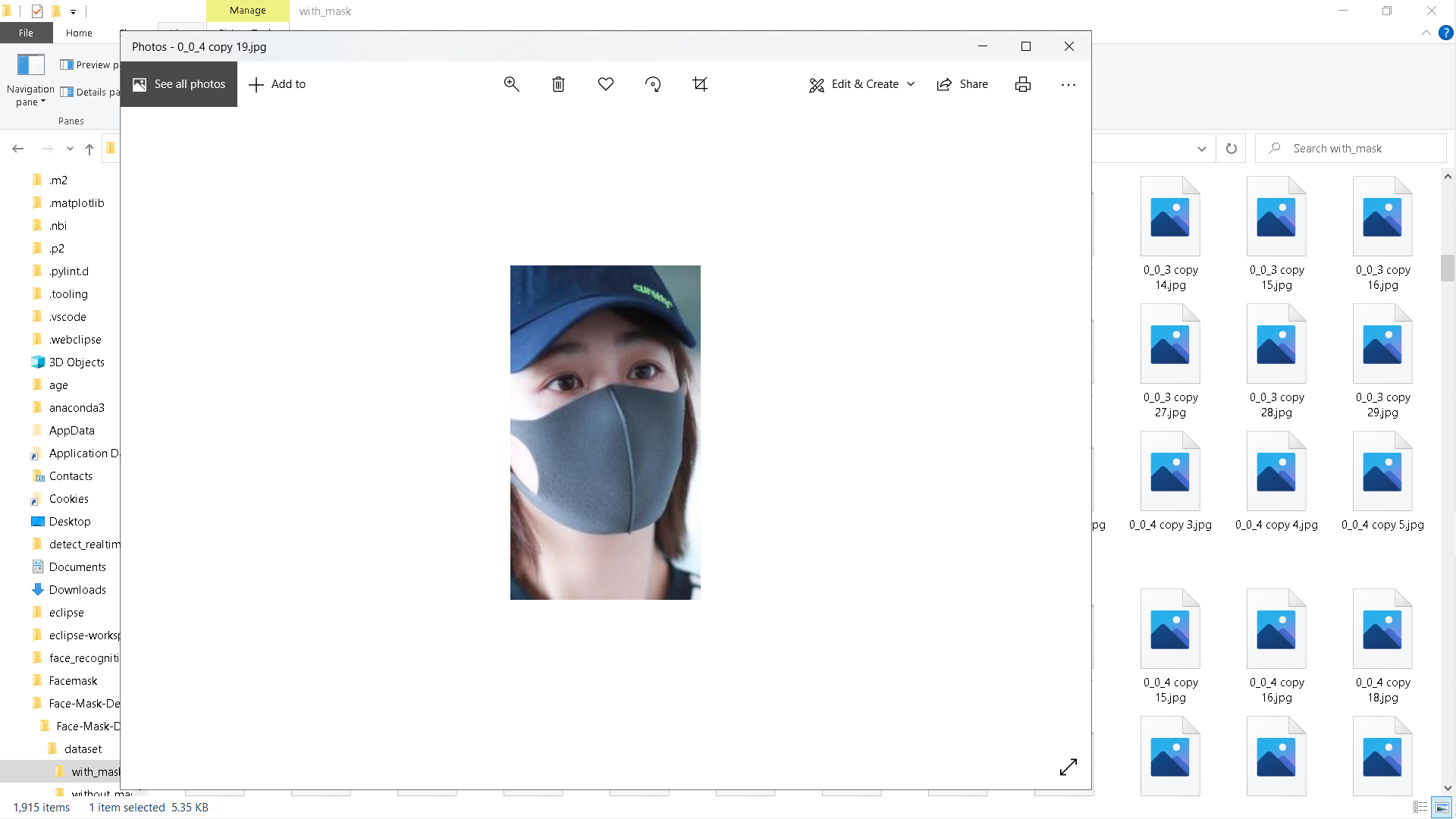
Sau đây là mô tả cách cài đặt một trường. Gồm 3 bước:

**Bước 1: Tải xuống tập dữ liệu.**

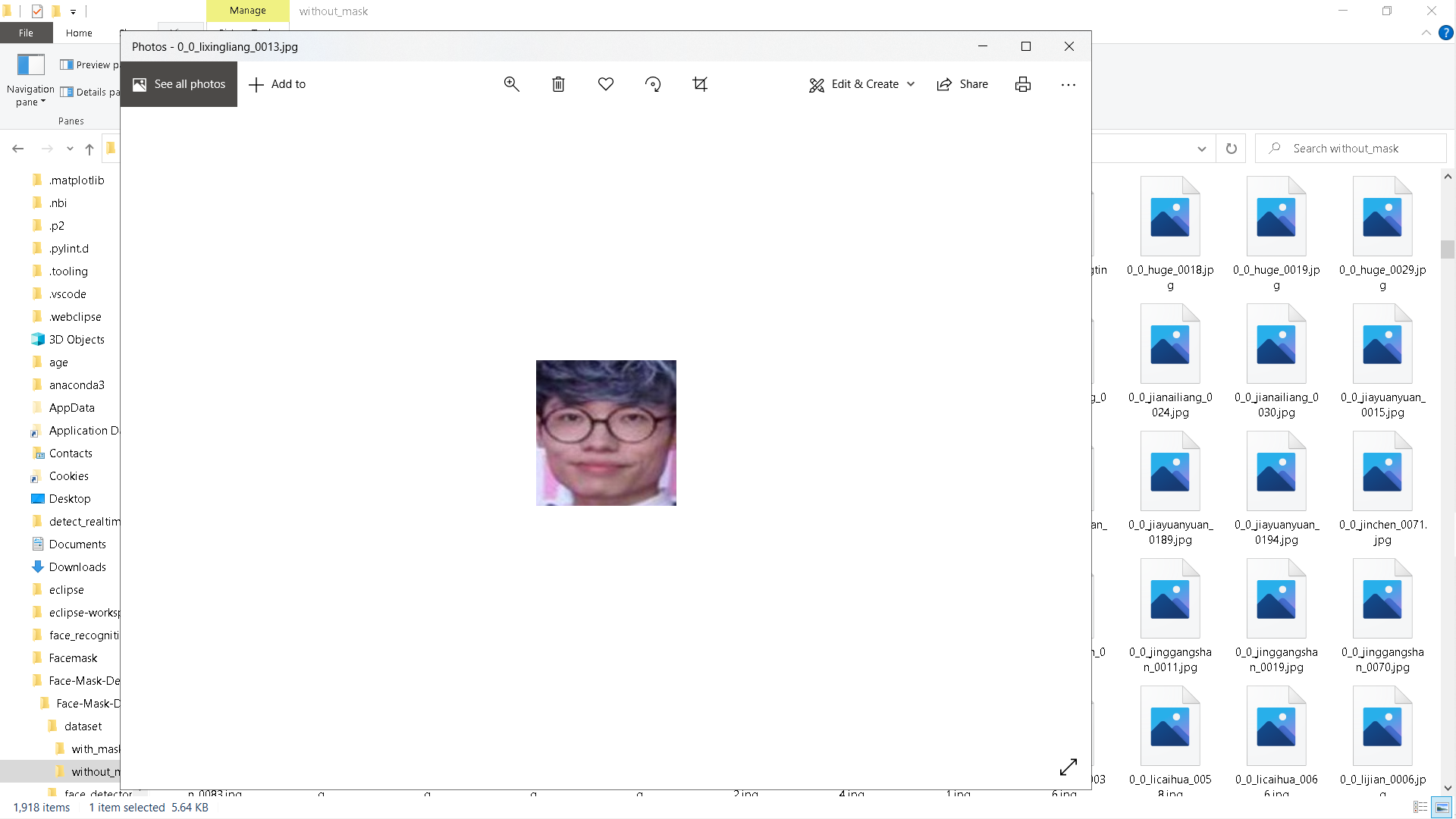
* Để training cho máy nhận diện được khuôn mặt, trước hết ta cần có tập hợp những hình ảnh có đeo khẩu trang và không đeo khẩu trang:

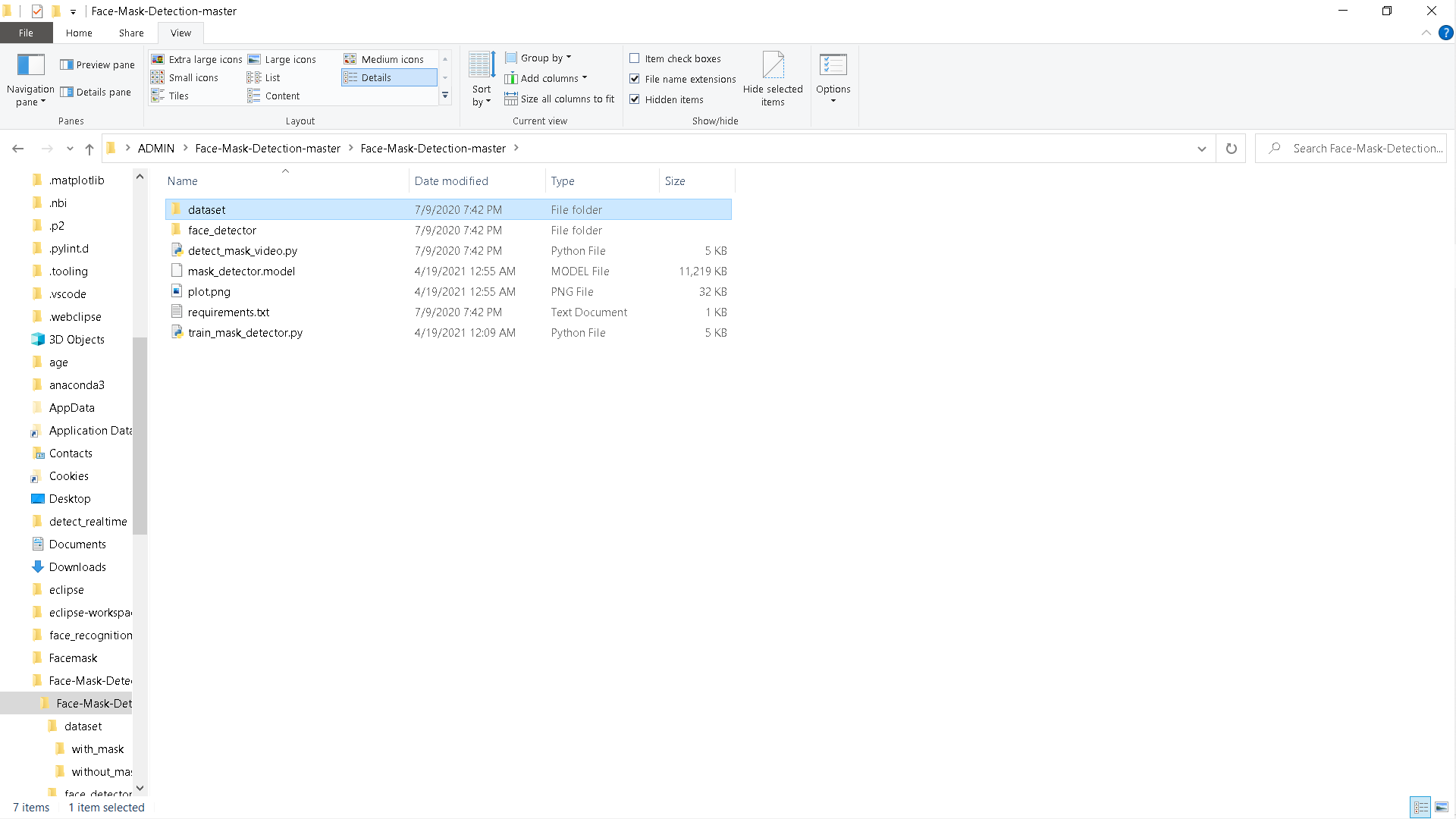
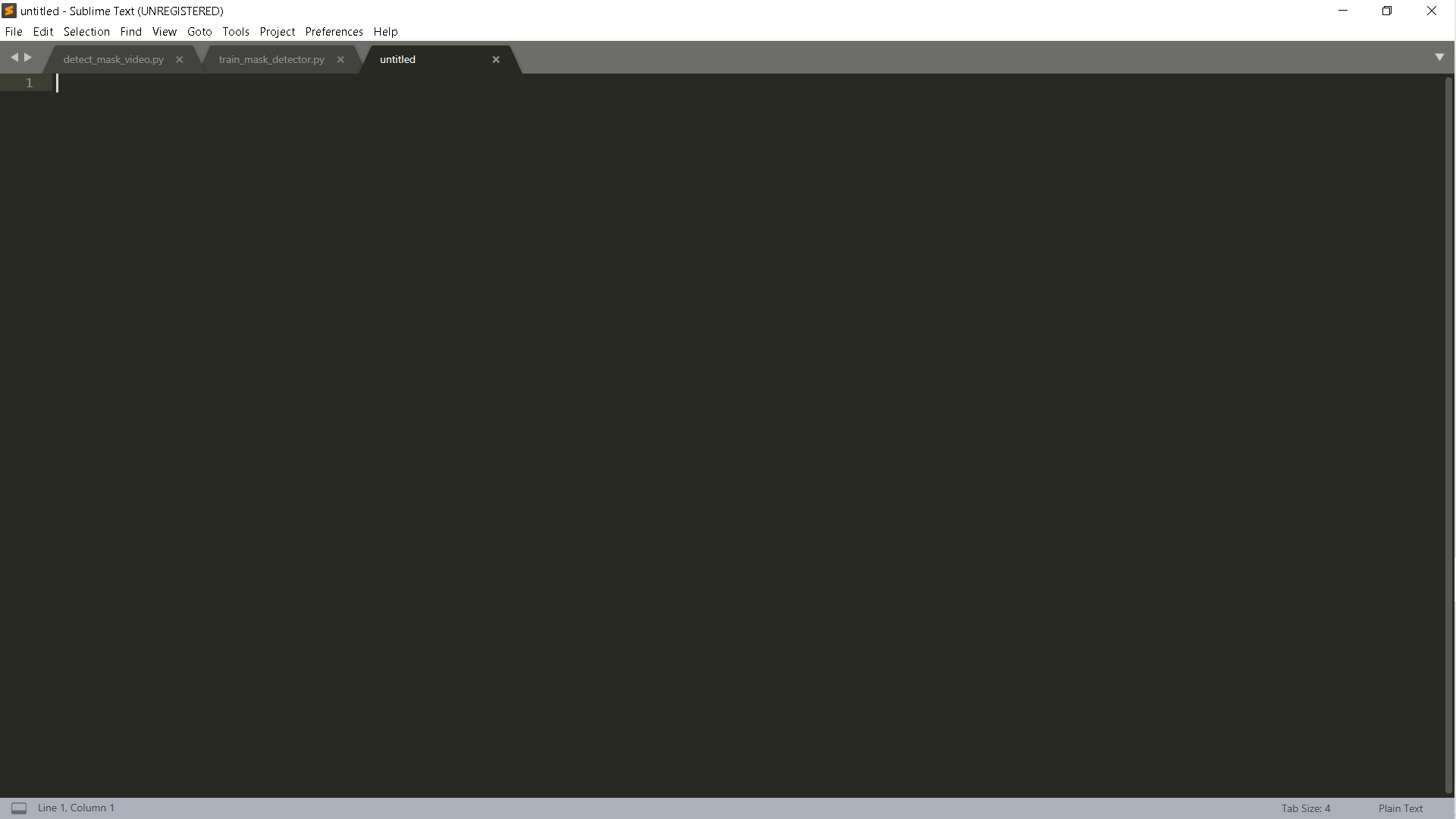
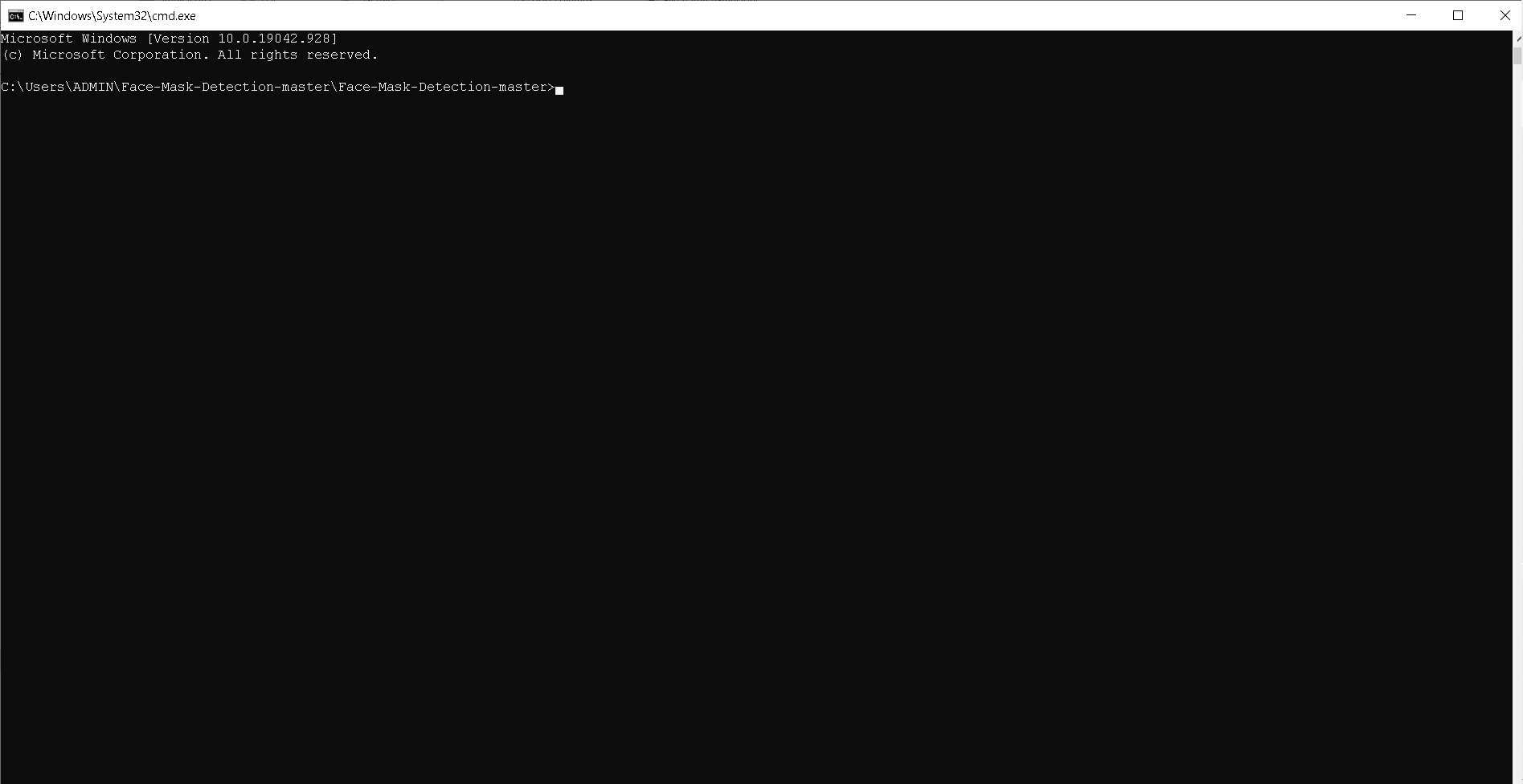
+ Ở file with\_mask:





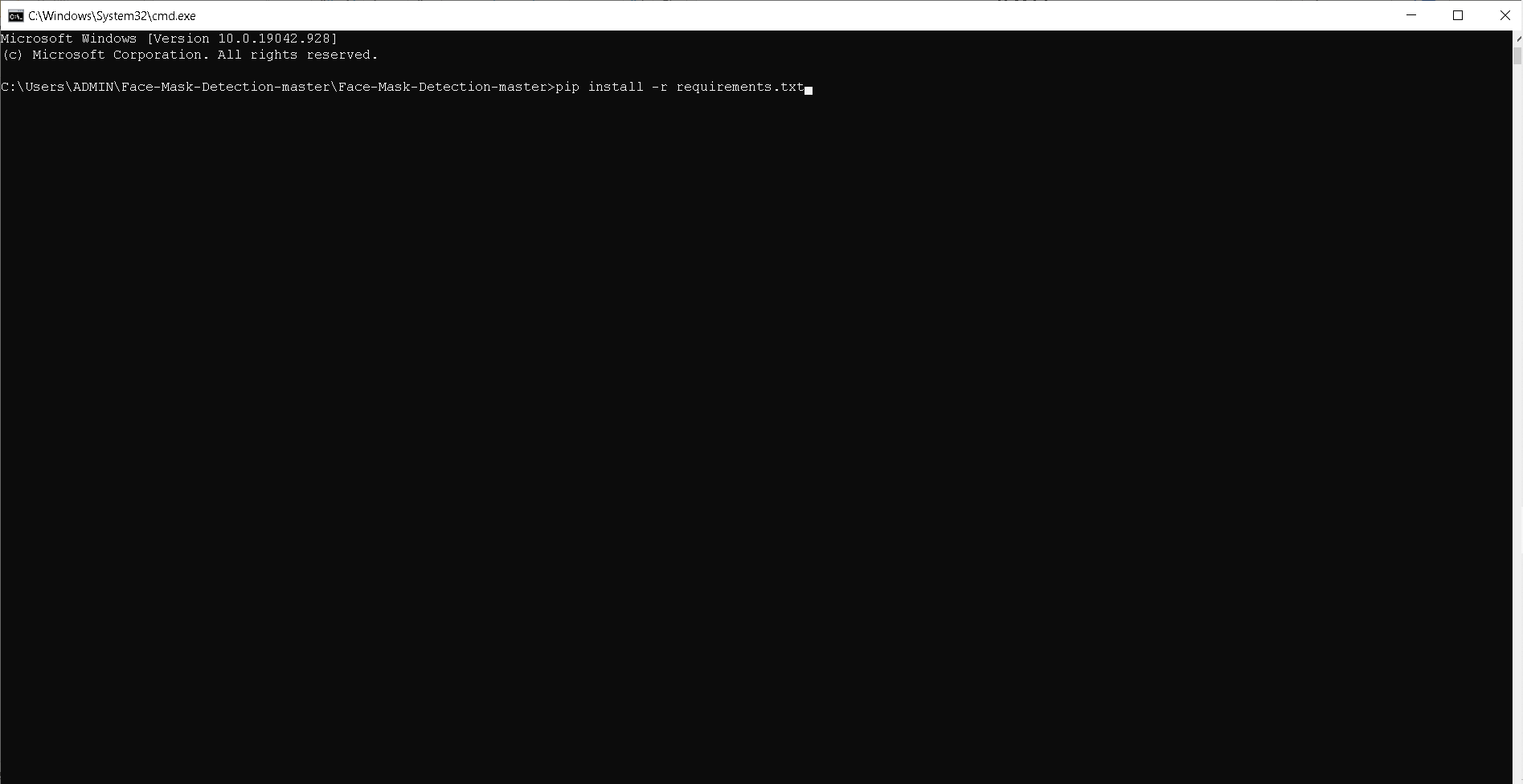
+ Ở file without\_mask:

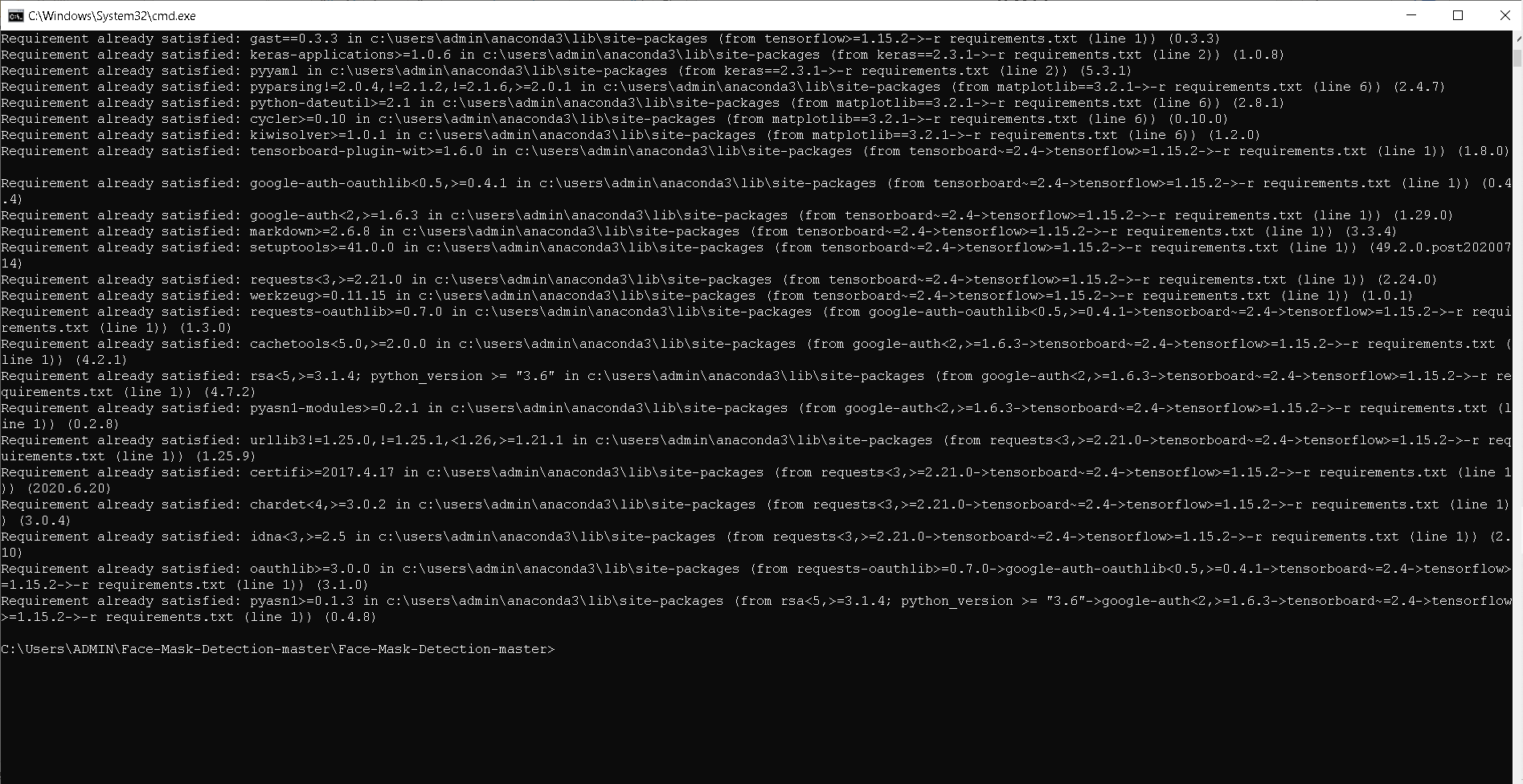


* Tất cả những hình ảnh này, chúng ta sẽ gói vào thư mục dataset và để chung thư mục với file python mà chúng ta sẽ cài đặt.
* 
* Tiếp theo, chúng ta cài đặt những thư viện cần thiết để chạy được đoạn code python. Chúng ta cần cài đặt môi trường Python, Subline text:
* 
* Sau đó ta mở Command line
* 
* Tạo một file requirements.txt để lưu các lệnh để install thư viện bằng command line:



* Tiến hành cài đặt các thư viện:





* Bước tạo môi trường chạy code đã xong, chúng ta hãy đi sâu vào phần code của dự án.

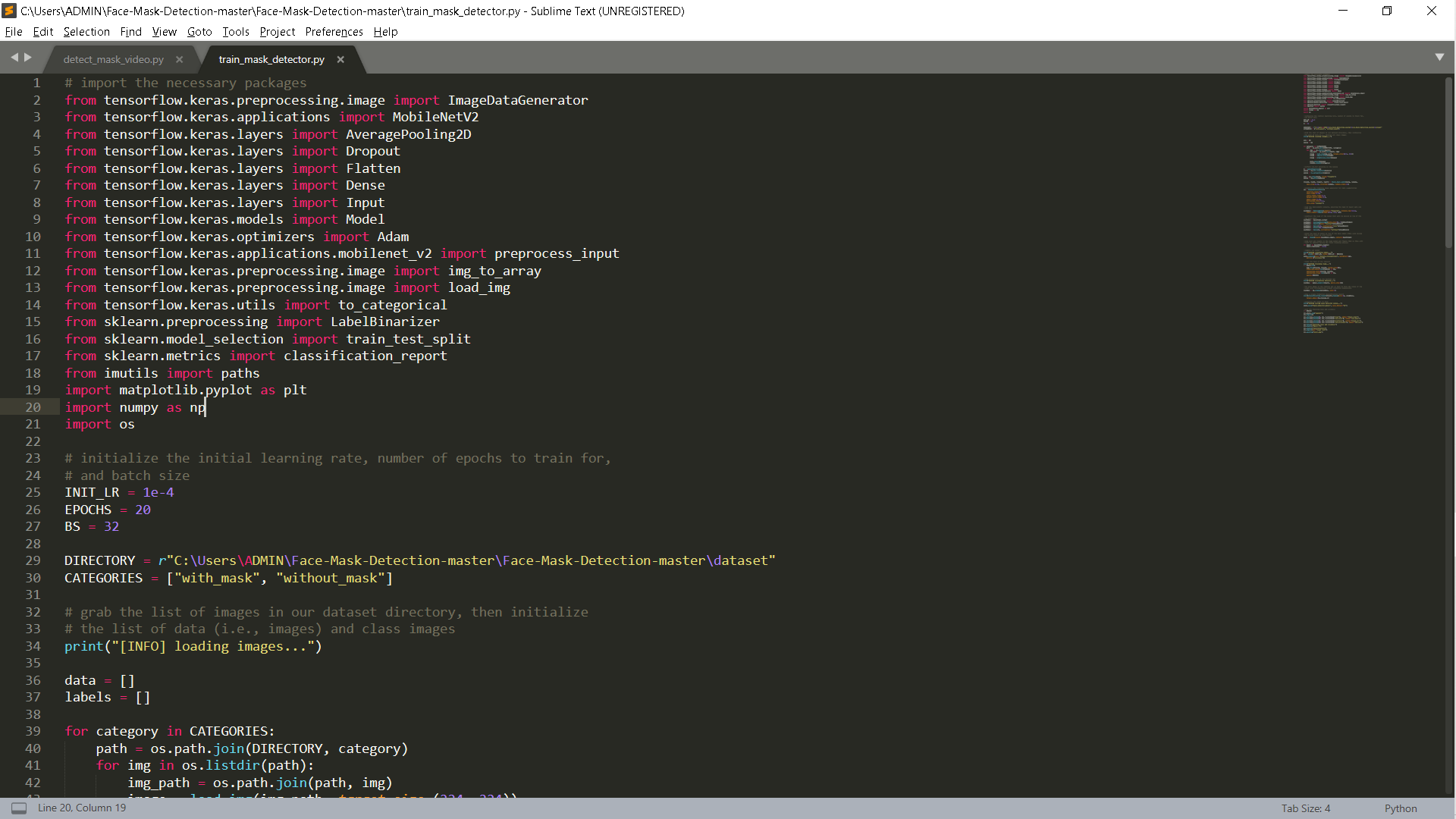
**Bước 2: Code**

Chúng tôi sẽ xây dựng dự án này trong hai phần. Trong phần đầu tiên, chúng ta sẽ viết một đoạn mã python và sử dụng thư viện Keras để đào tạo mô hình real-time facemask detector. Trong phần thứ hai, chúng tôi kiểm tra kết quả bằng webcam với thư viện OpenCV.

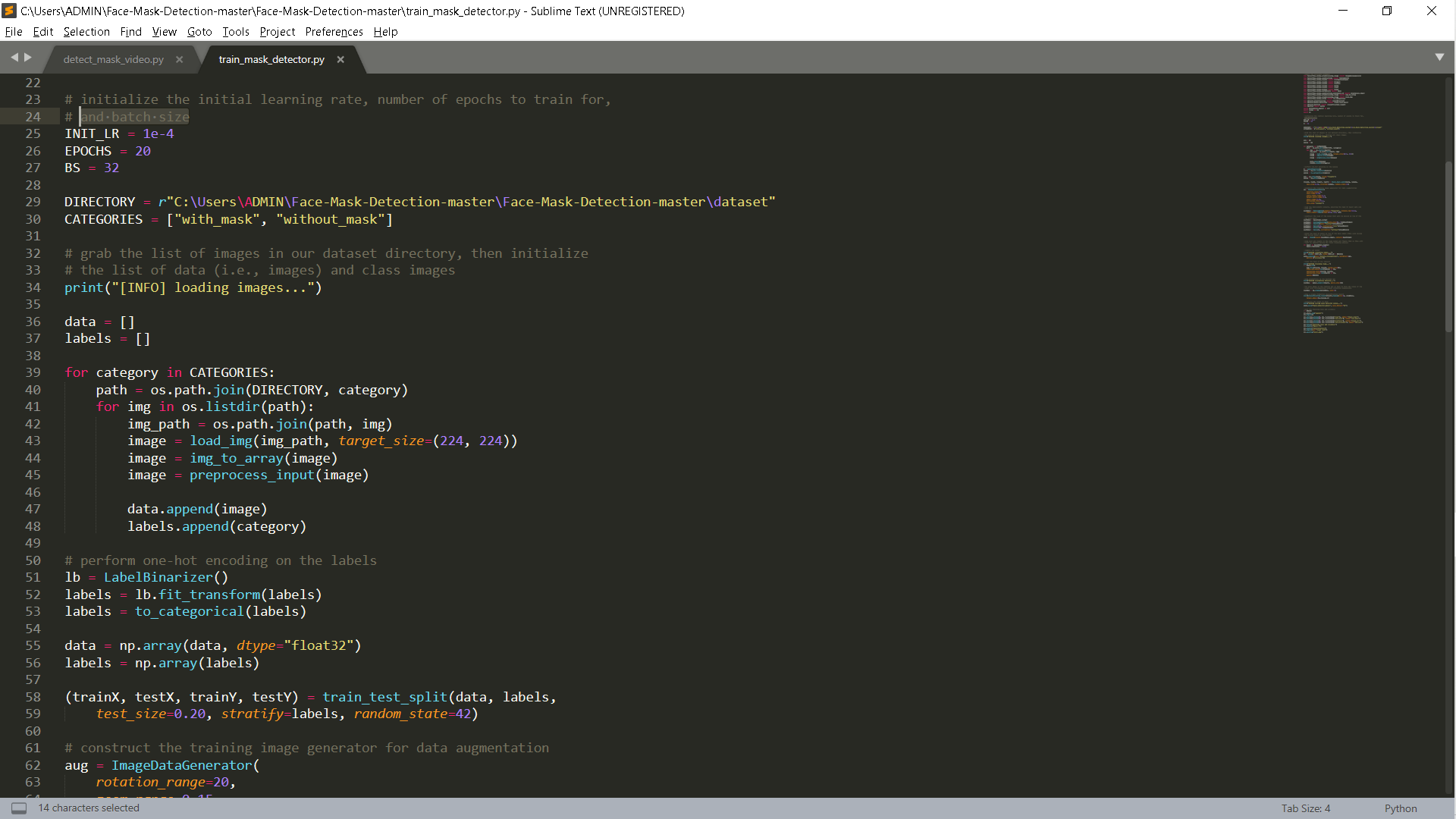
Tạo một python file tên là train\_mask\_detector.py để viết code training mạng thần kinh trên tập dữ liệu dataset theo các bước:

1.Import

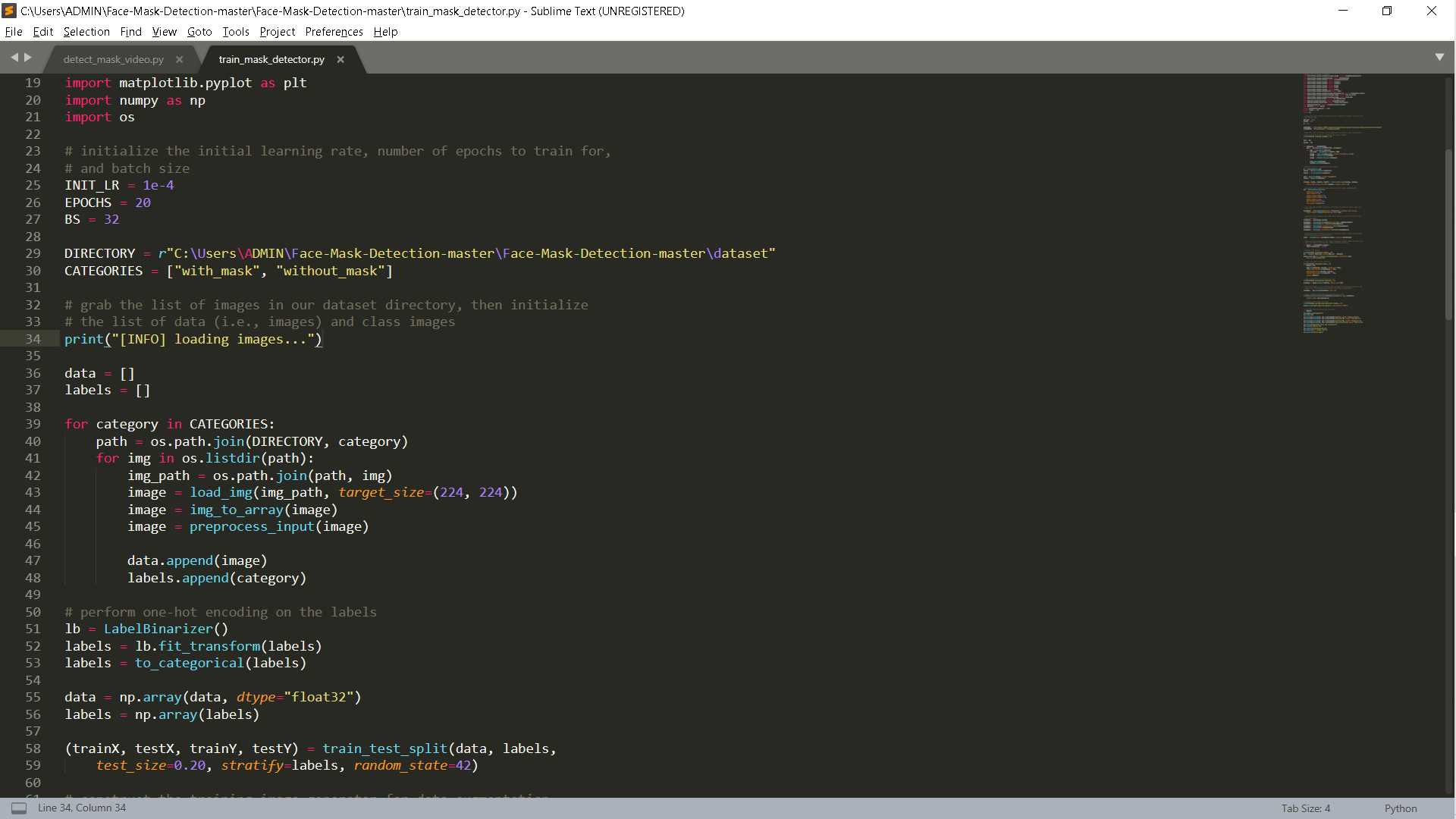
Import tất cả các thư viện và modules cần thiết.



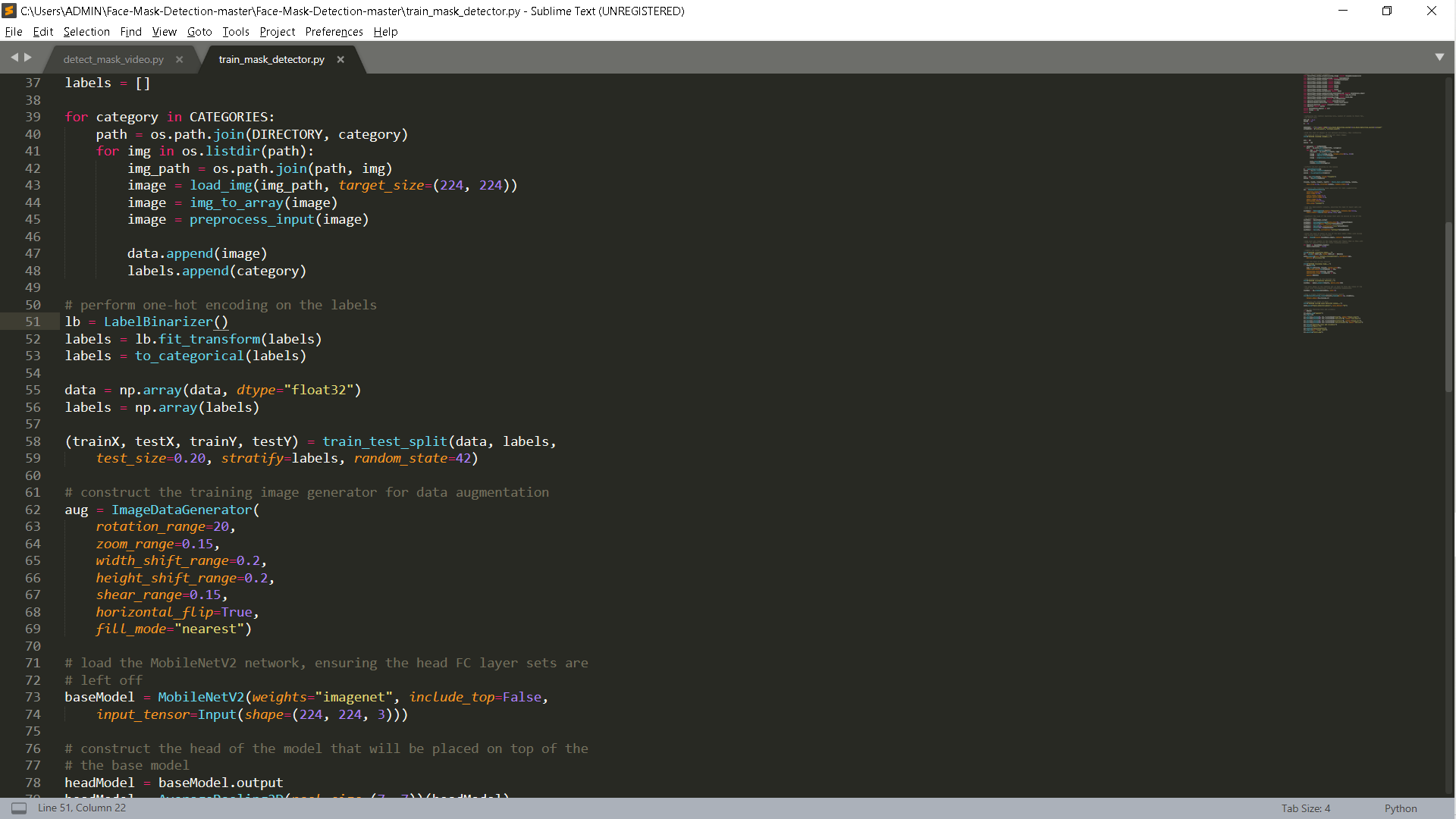
2. Khởi tạo tốc độ học ban đầu, số thời gian để đào tạo và kích thước Batch size:



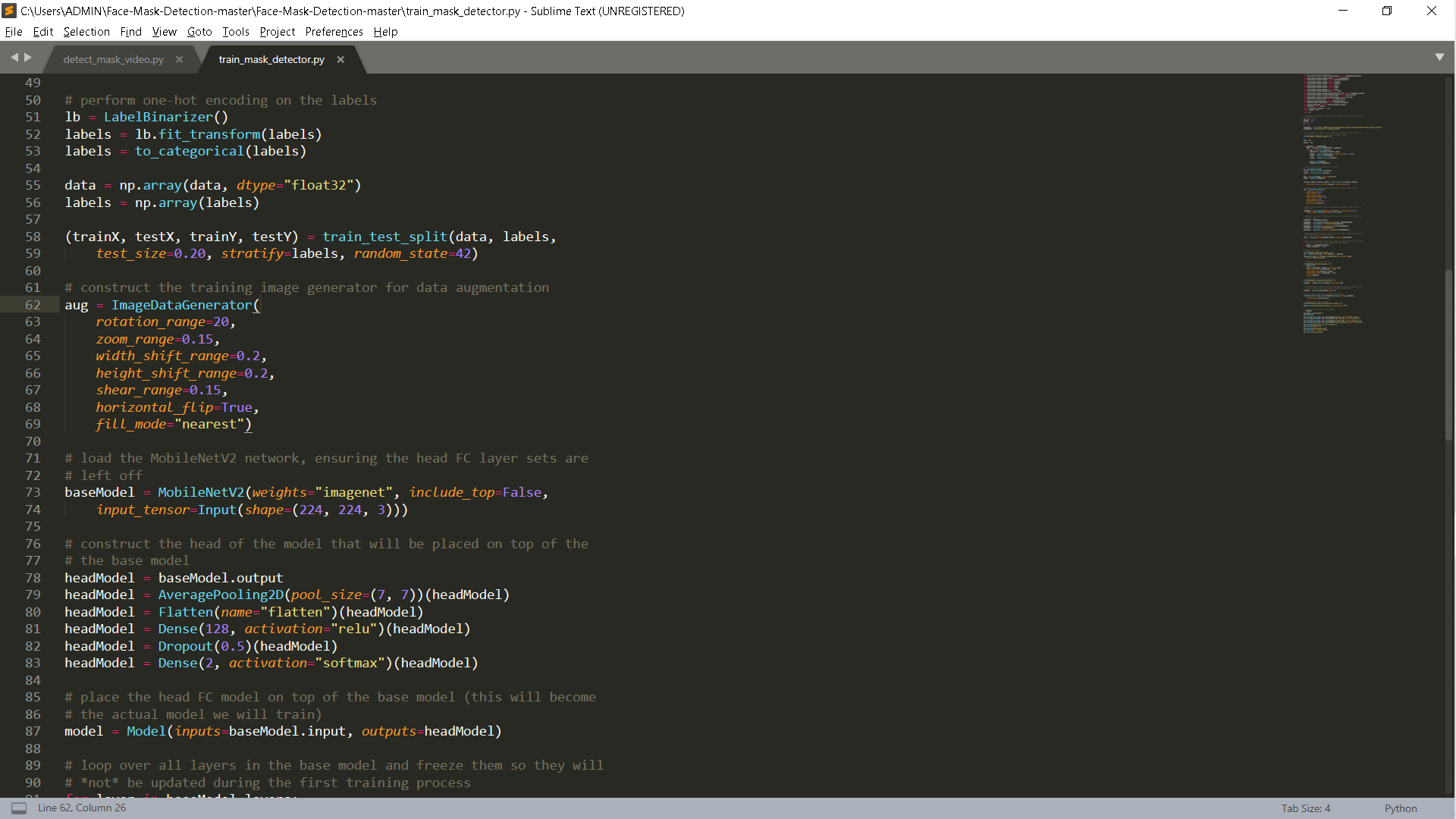
Lấy danh sách hình ảnh trong thư mục tập dữ liệu dataset, sau đó khởi tạo một mảng danh sách dữ liệu (hình ảnh,…)



Thực hiện mã hóa một lần trên các label:

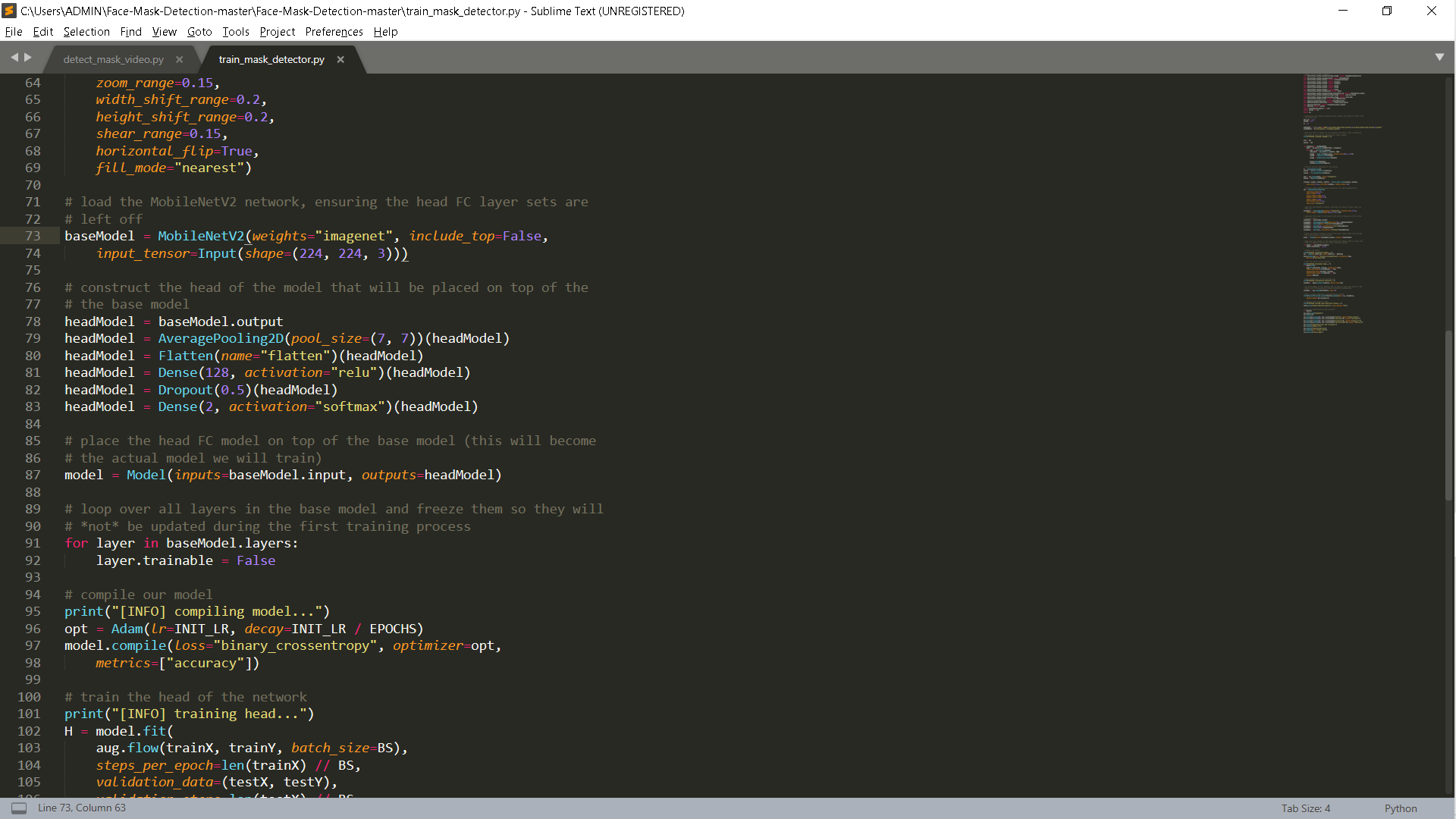


Xây dựng trình tạo hình ảnh đào tạo để tăng cường dữ liệu:

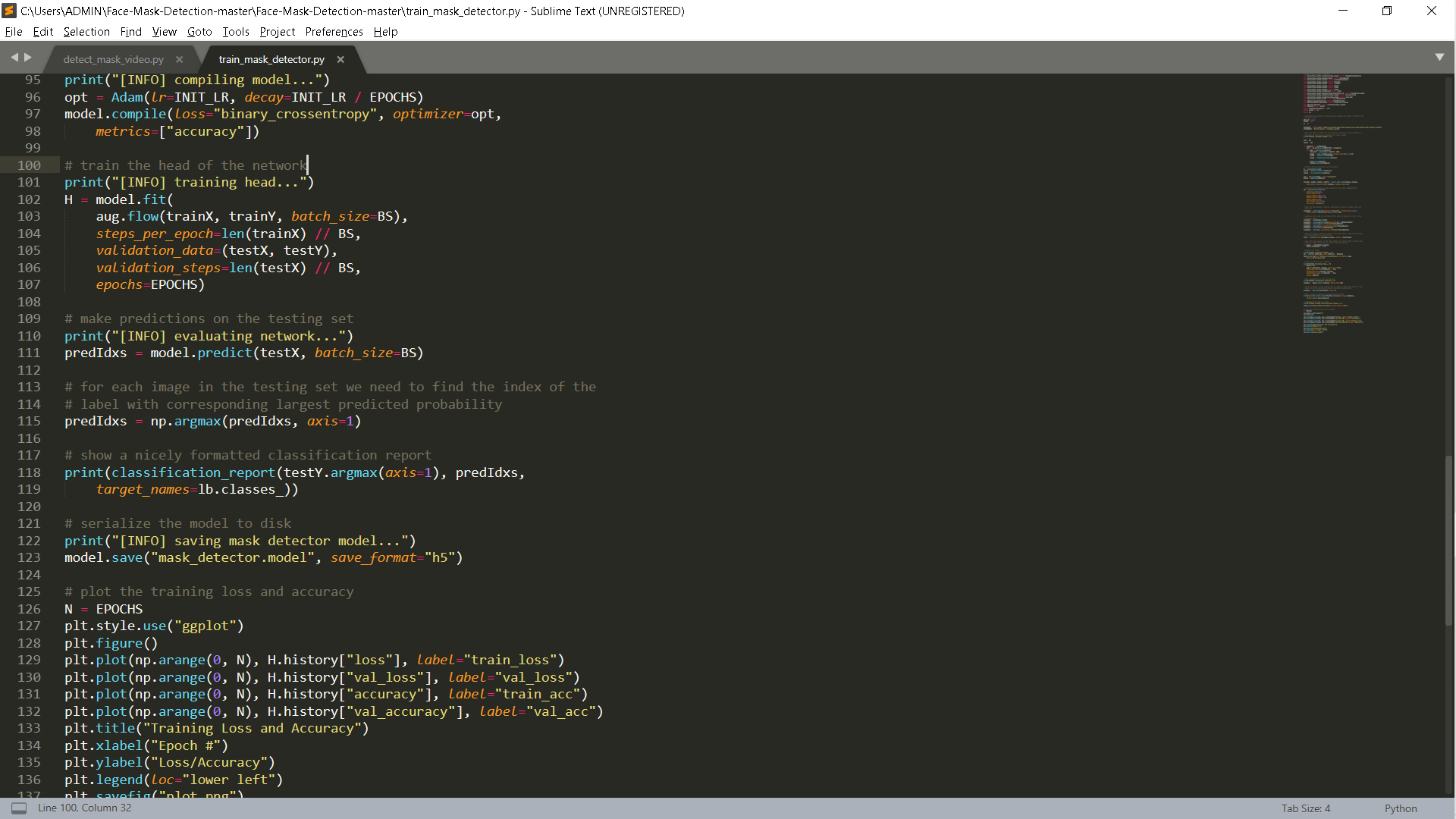


Tải mạng MobileNetV2, đảm bảo các bộ lớp FC đầu được tắt:

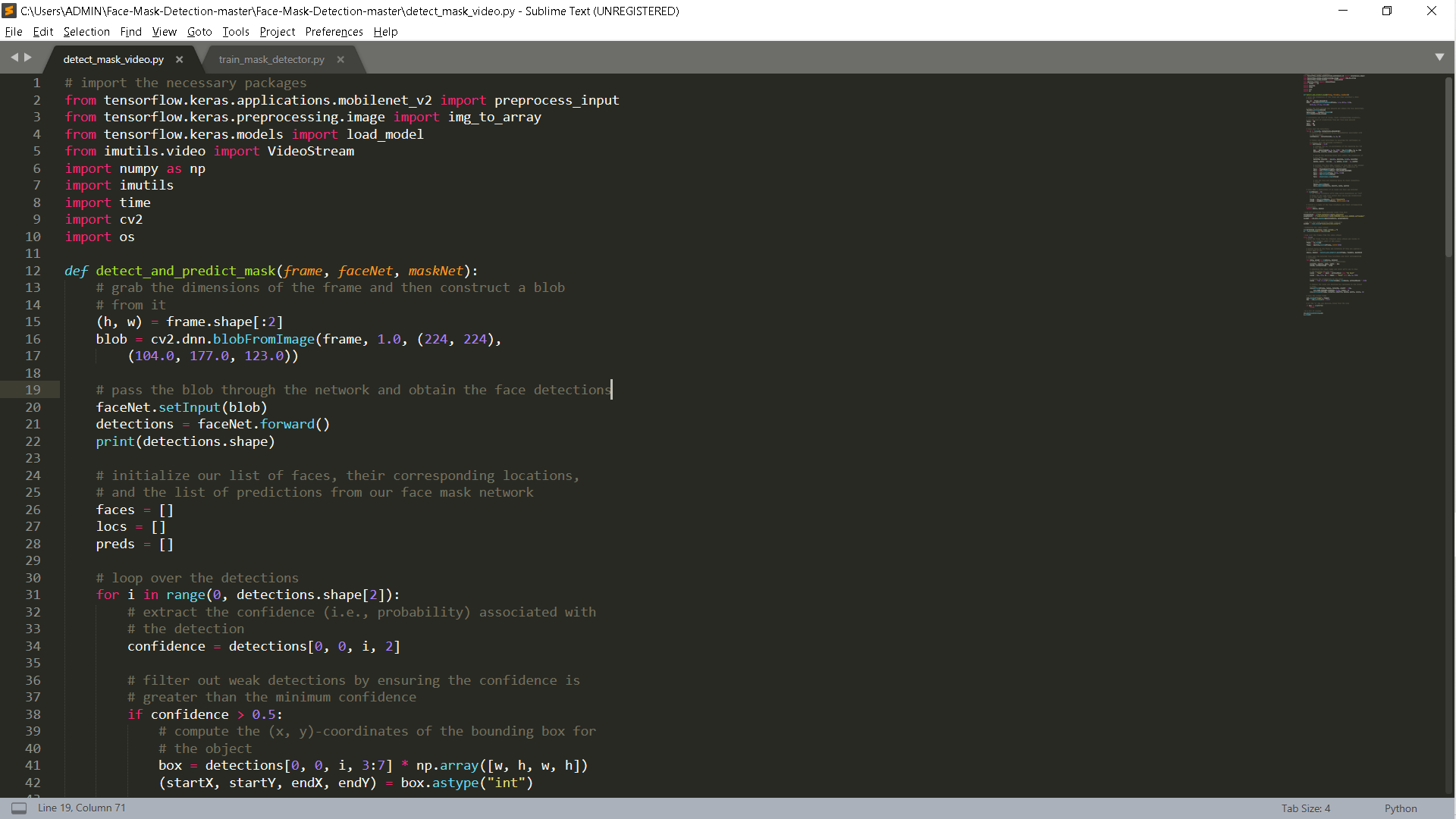
Xây dựng phần đầu của mô hình sẽ được đặt trên đầu của mô hình cơ sở:

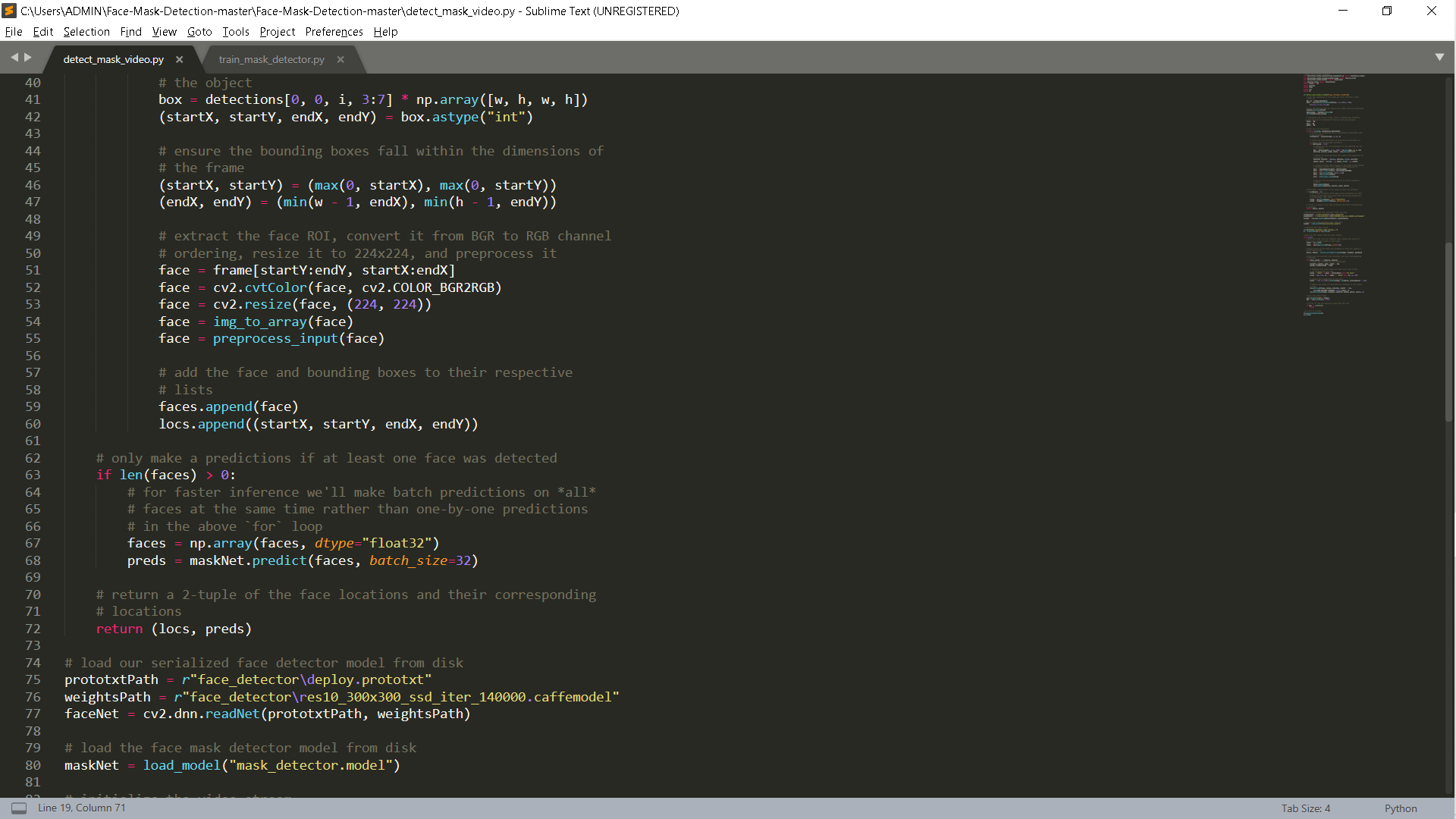


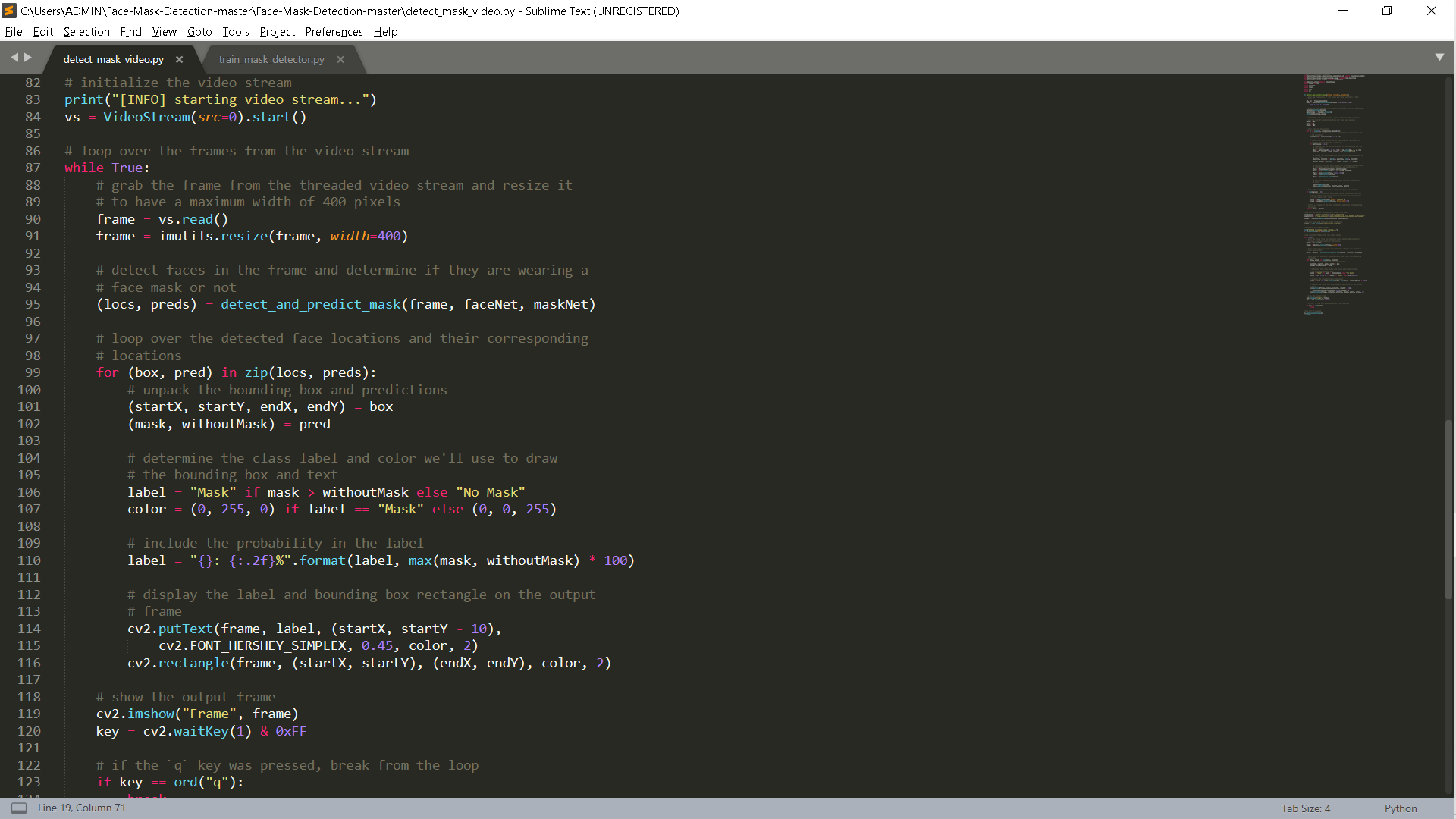
Biên dịch mô hình, đào tạo phần đầu của mạng lưới:

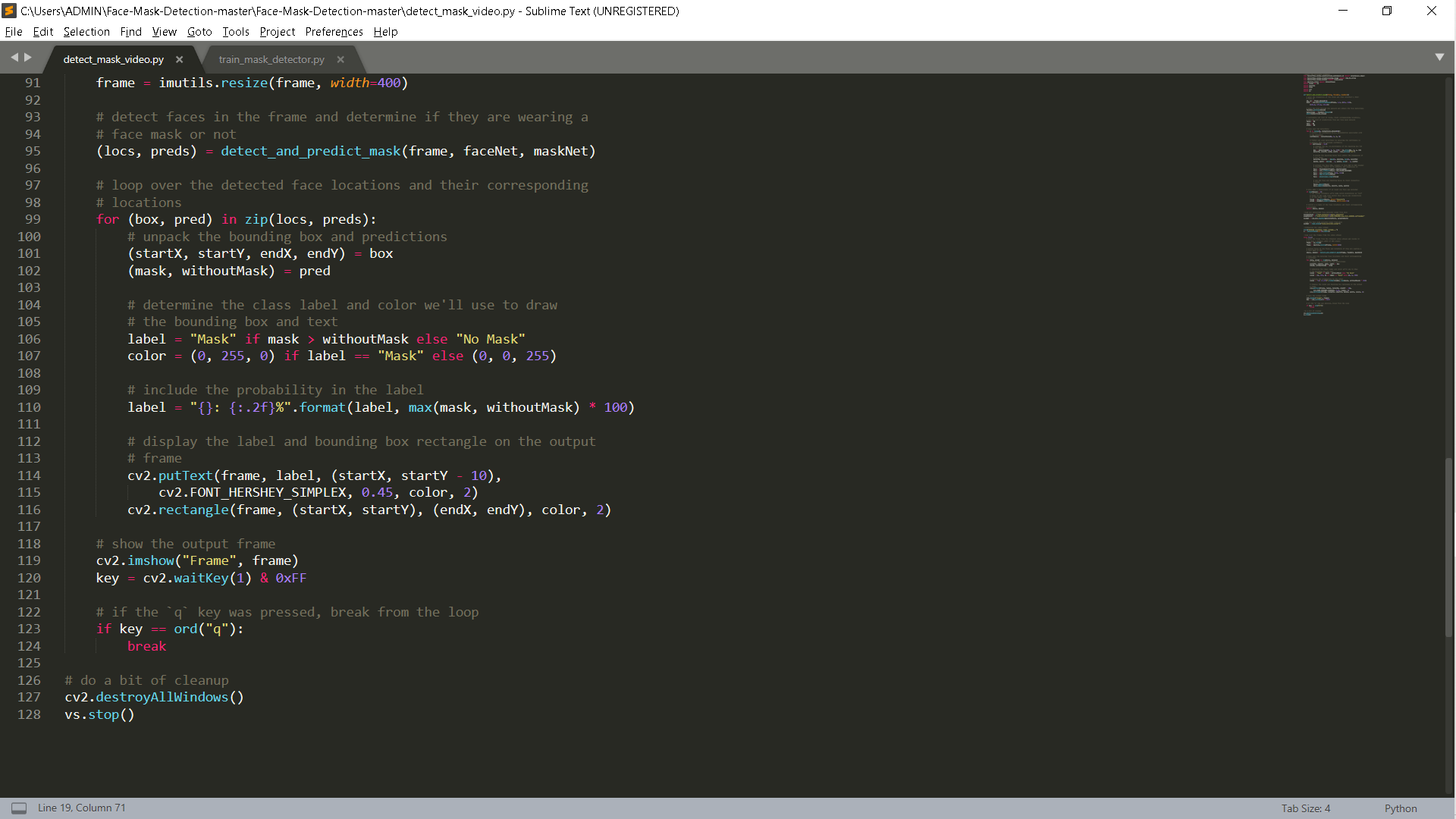


Tạo file python detect\_mask\_video.py, chúng ta sẽ chạy file này.

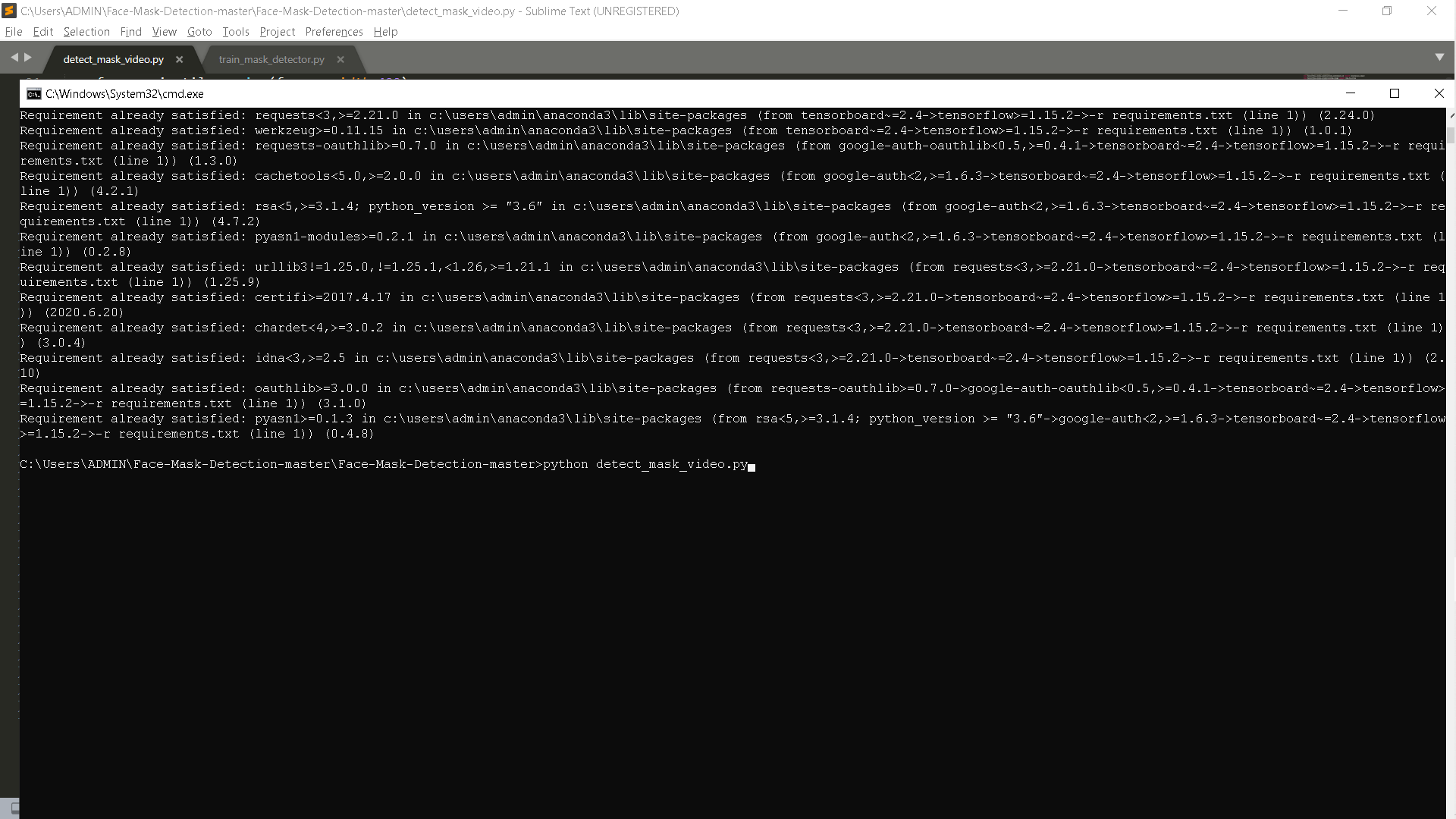




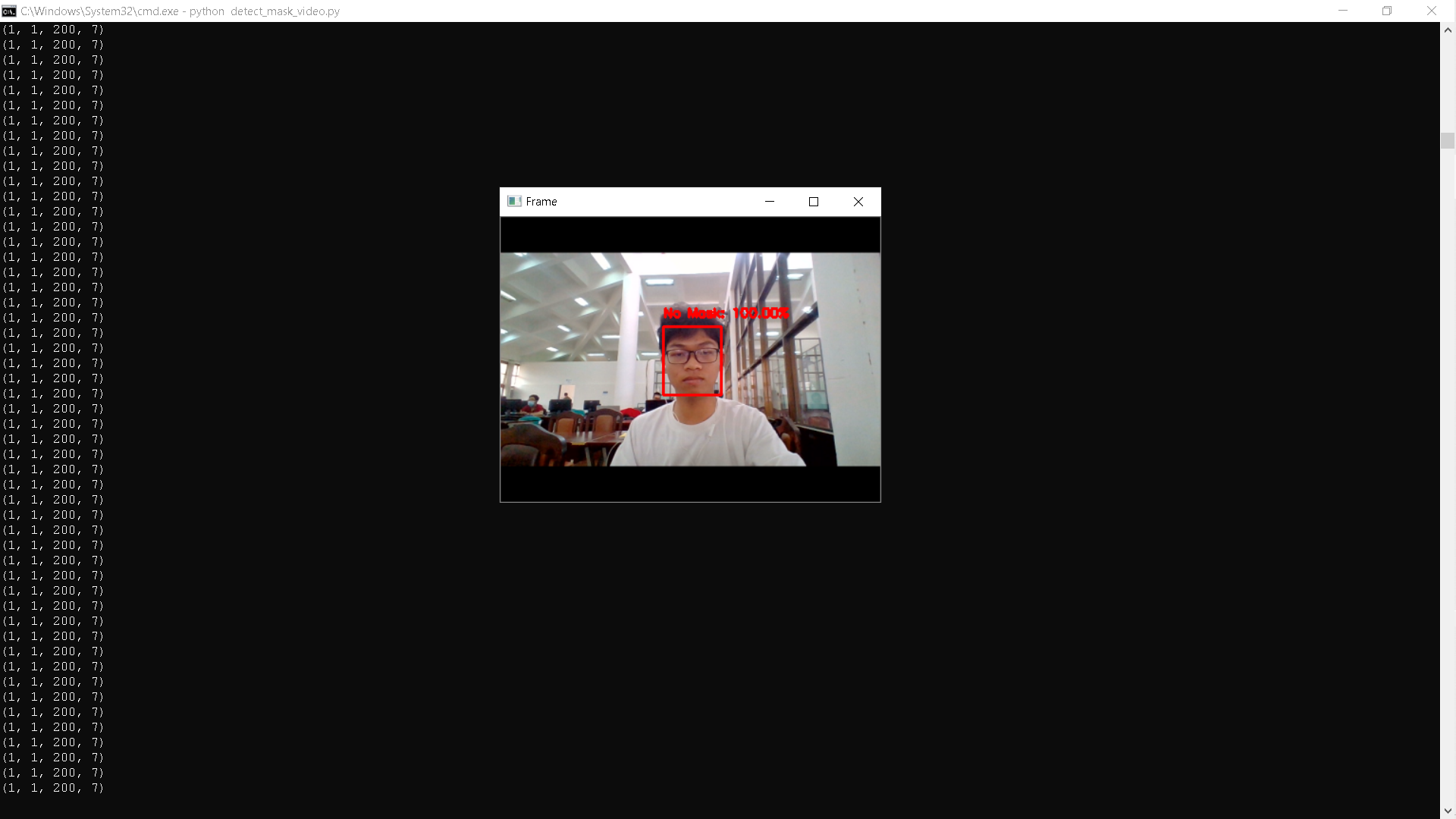


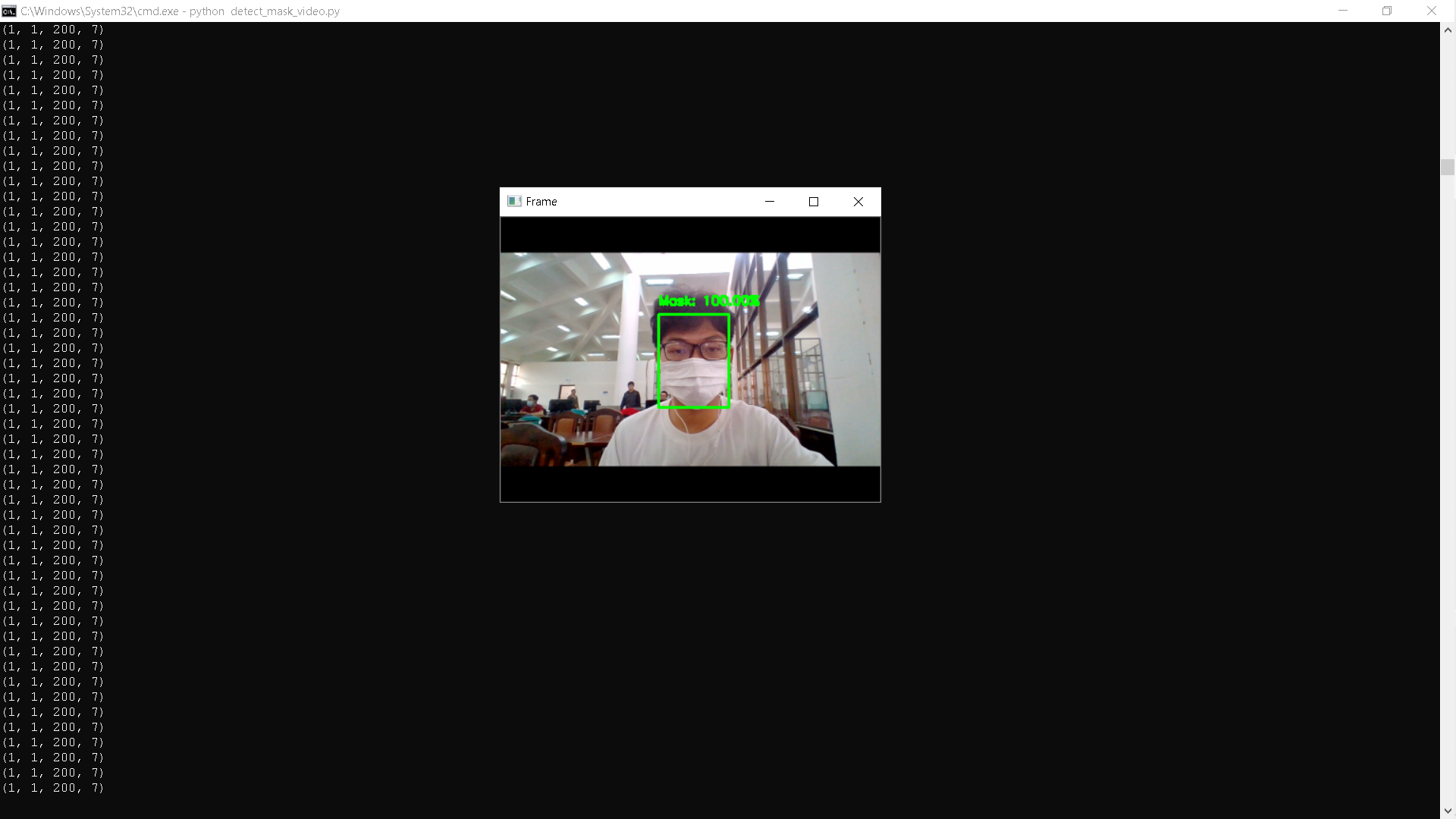


**Bước 3: Chạy và test chương trình.**



**Kết quả chạy thử chương trình:**





**Đánh giá mô hình:**

Mô hình đạt độ chính xác tới 97%

**Hạn chế:** ở những nơi có ánh sáng yếu sẽ khó để nhận dạng.