

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ  
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP  
NGÀNH KHOA HỌC MÁY TÍNH**

**Đề tài**

**PHÂN LỚP ẢNH X-QUANG PHỔI  
BẰNG PHƯƠNG PHÁP HỌC SÂU**

**Sinh viên thực hiện : Trần Anh Khoa  
Mã số : B1913240  
Khóa : 45**

**Cần Thơ, 05/2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ  
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP  
NGÀNH KHOA HỌC MÁY TÍNH**

**Đề tài**

**PHÂN LỚP ẢNH X-QUANG PHỔI  
BẰNG PHƯƠNG PHÁP HỌC SÂU**

**Giáo viên hướng dẫn:  
Th.S. Võ Trí Thức**

**Sinh viên thực hiện: Trần Anh Khoa  
Mã số: B1913240  
Khoá: 45**

**Cần Thơ, 05/2023**

## This image shows a full page of white paper with horizontal dashed lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the paper.

## LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn thầy Võ Trí Thức, thầy đã tận tình hướng dẫn, cung cấp những kiến thức quý báu cho em trong suốt quá trình thực hiện đề tài luận văn tốt nghiệp. Cảm ơn thầy đã tạo điều kiện và cho em cơ hội để thực hiện đề tài này. Cảm ơn sự tâm huyết của thầy trong quá trình giảng dạy, ngọn lửa nhiệt huyết của thầy truyền lại sẽ là một hành trang cho em vững bước trên con đường mưu cầu tri thức.

Mặc dù đã cố gắng và nỗ lực rất nhiều trong quá trình thực hiện đề tài này. Tuy nhiên, do kiến thức và kỹ năng còn hạn chế nên khó tránh những sai sót. Em rất mong được sự thông cảm, góp ý của thầy để đề tài luận văn tốt nghiệp của em được hoàn thiện hơn.

Cần Thơ, ngày 02 tháng 05 năm 2023

Người viết

Trần Anh Khoa

---

## MỤC LỤC

<b>LỜI CẢM ƠN.....</b>	<b>4</b>
<b>DANH MỤC HÌNH .....</b>	<b>3</b>
<b>DANH MỤC TỪ VIẾT TẮT .....</b>	<b>7</b>
<b>PHẦN GIỚI THIỆU.....</b>	<b>8</b>
1. Đặt vấn đề.....	8
2. Lịch sử giải quyết vấn đề.....	8
3. Mục tiêu đề tài .....	9
4. Đối tượng và phạm vi nghiên cứu .....	9
5. Phương pháp nghiên cứu .....	9
6. Kết quả đạt được.....	10
7. Bố cục luận văn.....	11
<b>PHẦN NỘI DUNG.....</b>	<b>12</b>
<b>CHƯƠNG 1 .....</b>	<b>12</b>
<b>MÔ TẢ BÀI TOÁN .....</b>	<b>12</b>
1. Mô tả chi tiết bài toán .....	12
2. Vấn đề và giải pháp liên quan đến bài toán .....	12
2.1. Thu thập tập dữ liệu .....	12
2.2. Xây dựng các mô hình máy học .....	13
3. Xây dựng giao diện cho hệ thống dự đoán .....	20
<b>CHƯƠNG 2 .....</b>	<b>22</b>
<b>THIẾT KẾ VÀ CÀI ĐẶT .....</b>	<b>22</b>
1. Sơ đồ hệ thống sử dụng 5 mô hình truyền thống để dự đoán nhãn .....	22
1.1. Sơ đồ quá trình huấn luyện mô hình.....	22
1.2. Sơ đồ quá trình dự đoán nhãn .....	23
2. Sơ đồ sử dụng mô hình MoCo dùng để huấn luyện và dự đoán nhãn .....	24
2.1 Sơ đồ quá trình huấn luyện của mô hình Moco .....	24
<b>2.2. Sơ đồ quá trình điều chỉnh trọng số của mô hình MoCo.</b> .....	<b>25</b>
3. Cài đặt hệ thống .....	26
3.1. Tiền xử lý dữ liệu.....	26
3.2 Xây dựng các mô hình máy học truyền thống và huấn luyện.....	29
3.3. Tải mô hình đã được huấn và xây dựng giao diện website để dự đoán kết quả ...	33

---

2.5. Tiến hành huấn luyện mô hình MoCo-CXR.....	34
<b>CHƯƠNG 3 .....</b>	<b>36</b>
<b>KẾT QUẢ THỰC NGHIỆM .....</b>	<b>36</b>
1. Giao diện sản phẩm.....	36
2. Giới thiệu các tính năng.....	36
3. Kết quả kiểm thử và đánh giá .....	39
3.1. Các chỉ số đánh giá .....	39
3.2. Kết quả thực nghiệm.....	41
<b>PHẦN KẾT LUẬN .....</b>	<b>50</b>
1. Kết quả đạt được .....	50
2. Hạn chế .....	50
3. Hướng phát triển .....	50
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>51</b>
<b>ĐƯỜNG DẪN THAM KHẢO .....</b>	<b>53</b>

## DANH MỤC HÌNH

Hình 1. Kiến trúc của mô hình DenseNet121 .....	14
Hình 2. Kiến trúc mô hình MobileNetV2 .....	15
Hình 3. Kiến trúc của mô hình ResNet50V2 .....	16
Hình 4. Kiến trúc mô hình VGG16.....	17
Hình 5. Kiến trúc mô hình VGG19.....	18
Hình 6. Ảnh minh họa trang chủ gradio .....	20
Hình 7. Xây dựng giao diện dễ dàng với demo gradio .....	21
Hình 8. Sơ đồ tổng quát quá trình huấn luyện .....	22
Hình 9. Sơ đồ tổng quát quá trình tải mô hình và dự đoán nhãn của 5 mô hình truyền thống .....	23
Hình 10. Sơ đồ quá trình huấn luyện mô hình Moco với 120000 ảnh không nhãn.....	24
Hình 11. Sơ đồ quá trình điều chỉnh trọng số của mô hình Moco .....	25
Hình 12. Cấu trúc tập dữ liệu sau khi được lấy từ Kaggle .....	26
Hình 13. Ảnh sau khi đã resize về kích thước 256x256 .....	26
Hình 14. Cấu trúc tập dữ liệu sau khi chia thành 3 phần để tiến hành huấn luyện mô hình .....	27
Hình 15. Nội dung file train.csv .....	28
Hình 16. Thông tin chi tiết về các tập dữ liệu.....	28
Hình 17. Các ảnh X-quang sau khi được tiền xử lý.....	29
Hình 18. Sơ đồ chi tiết quá trình huấn luyện mô hình truyền thống.....	32
Hình 19. Sơ đồ quá trình tải mô hình và dự đoán kết quả trên website bằng thư viện gradio .....	33
Hình 20. Sơ đồ chi tiết quá trình huấn luyện mô hình Moco sử dụng tập dữ liệu 120000 ảnh .....	34
Hình 21. Sơ đồ mô tả chi tiết quá trình tinh chỉnh trọng số của mô hình MoCo sử dụng 2 trọng số khác nhau .....	35
Hình 22. Giao diện website dùng để dự đoán hình ảnh X-quang phổi.....	36
Hình 23. Vị trí tải ảnh .....	37
Hình 24. Giao diện tải ảnh .....	37
Hình 25. Hệ thống đang tiến hành dự đoán nhãn của tám ảnh đã tải lên .....	38
Hình 27. Kết quả dự đoán của website .....	39
Hình 28. ResNet50V2 model loss.....	41
Hình 29. ResNet50V2 model accuracy.....	41
Hình 30. VGG16 model loss.....	42
Hình 31. VGG16 model accuracy.....	42
Hình 32. VGG19 model loss.....	43
Hình 33. ResNet50V2 model accuracy.....	43
Hình 34. DenseNet121 model loss .....	44
Hình 35. DenseNet121 model accuracy .....	44
Hình 36. MobileNetV2 model loss .....	45
Hình 37. MobileNetV2 model accuracy .....	45
Hình 38. Biểu đồ biểu thị giá trị Accuracy Score trên tập kiểm tra của 5 mô hình truyền thống .....	46

---

Hình 39. Biểu đồ biểu thị giá trị F1-score trên tập kiểm tra của từng lớp của 5 mô hình truyền thống .....	47
Hình 40. Biểu đồ Accuracy Score của mô hình Moco sử dụng tập dữ liệu 4800 ảnh để huấn luyện.....	47
Hình 41. Biểu đồ F1 Score của mô hình Moco sử dụng tập dữ liệu 4800 ảnh để huấn luyện .....	48
Hình 42. Biểu đồ Accuracy Score của mô hình Moco sử dụng tập dữ liệu 300 ảnh để huấn luyện.....	48
Hình 43. Biểu đồ F1 Score của mô hình MoCo sử dụng tập dữ liệu 300 ảnh để huấn luyện .....	49



---

## **ABSTRACT**

Diagnosis of lung disease through chest X-ray is a revolutionary concept that has the potential to revolutionize the way we diagnose and treat respiratory conditions. With the help of advanced AI algorithms, radiologists can quickly and accurately detect abnormalities in chest X-rays, which can then be used to diagnose a wide range of diseases. such as pneumonia, tuberculosis and lung cancer. Not only does this make it easier for doctors to identify these conditions, but it also helps them devise a more accurate treatment plan for their patients. Stemming from the above factual problem, this thesis implements the topic "Classification of chest X-ray images by deep learning method". The system is built on advanced machine learning models used to classify lung diseases such as Covid-19, pneumonia and disease-free imaging. The training model uses data taken from Kaggle, which is a website specializing in providing fairly large datasets with high reliability, trusted by the Machine Learning research community around the world. The first step, implementing the test system, showed that the model achieved results with an F1 score of 0.92. Through the experimental results, it can be seen that the application of the topic in practice is completely feasible.

---

## **TÓM TẮT**

Chẩn đoán bệnh phổi thông qua chụp X-quang phổi là một khái niệm mang tính cách mạng có khả năng cách mạng hóa cách chúng ta chẩn đoán và điều trị các tình trạng hô hấp. Với sự trợ giúp của các thuật toán AI tiên tiến, các bác sĩ X-quang có thể phát hiện nhanh chóng và chính xác những bất thường trong ảnh chụp X-quang phổi, sau đó có thể được sử dụng để chẩn đoán một loạt bệnh như viêm phổi, lao và ung thư phổi. Điều này không chỉ giúp các bác sĩ dễ dàng xác định các tình trạng này mà còn giúp họ đưa ra phác đồ điều trị chính xác hơn cho bệnh nhân của mình. Xuất phát từ vấn đề thực tế trên, luận văn này thực hiện đề tài “Phân lớp ảnh X-quang phổi bằng phương pháp học sâu”. Hệ thống được xây dựng dựa trên các mô hình máy học tiên tiến dùng để phân loại các bệnh về phổi như Covid-19, viêm phổi và ảnh chụp không có bệnh. Mô hình huấn luyện sử dụng dữ liệu được lấy từ Kaggle, đây là một website chuyên cung cấp các bộ dữ liệu khá lớn với độ tin cậy cao, được cộng đồng nghiên cứu về ML trên khắp Thế Giới tin dùng. Bước đầu tiên, triển khai hệ thống thử nghiệm cho thấy mô hình đạt kết quả với `accuracy_score` là 0.95. Thông qua kết quả thực nghiệm, có thể thấy rằng việc áp dụng đề tài vào thực tế hoàn toàn khả thi.

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Diễn giải
AI	Artificial intelligence	Trí tuệ nhân tạo
CNN	Convolution Neural Network	Mạng nơ ron tích chập
ML	Machine Learning	Máy học
DS	Data Science	Khoa học dữ liệu

---

## **PHẦN GIỚI THIỆU**

### **1. Đặt vấn đề**

Máy học là một chuyên ngành đã phát triển khá lâu và có nhiều ứng dụng rộng rãi trong đời sống con người, giúp cho máy tính hiểu được dữ liệu. Ứng dụng của máy học ngày càng được đánh giá cao như hệ thống gợi ý, nhận dạng vật thể, phát hiện sai phạm trong giao thông, theo dõi chuyển động, nhận dạng cử chỉ,...

Những năm gần đây, y khoa là một lĩnh vực được áp dụng máy học nhằm giúp hạn chế những sai lầm của bác sĩ khi chẩn đoán bệnh của bệnh nhân vì những sai lầm này có thể ảnh hưởng đến kết quả điều trị bệnh và đưa ra phương pháp điều trị không phù hợp. Việc sai lầm trong chẩn đoán thường do một số nguyên nhân sau: Môi trường xung quanh, các công cụ được sử dụng để chẩn đoán, sự thiếu kinh nghiệm của bác sĩ có thể dẫn đến sai sót trong chẩn đoán. Tất cả những yếu tố được đề cập này đều có thể góp phần gây ra ảnh hưởng đến sức khỏe của bệnh nhân cả thể chất lẫn tinh thần, có thể gây đến tử vong, bên cạnh đó, còn làm tăng chi phí điều trị.

Việc chẩn đoán hình ảnh X-quang phổi là rất quan trọng để phát hiện các bệnh về phổi, bao gồm cả bệnh lao và ung thư phổi, ảnh hưởng đến hàng triệu người trên thế giới mỗi năm. Công việc tốn thời gian này thường đòi hỏi bác sĩ X-quang chuyên nghiệp phải đọc các hình ảnh. Gần đây, các phương pháp học sâu đã có thể đạt được hiệu suất ở cấp độ chuyên gia trong các nhiệm vụ phân tích hình ảnh y khoa nhờ vào sự hỗ trợ bởi các kiến trúc mạng và sự xuất hiện của các tập dữ liệu lớn. Mục đích của đề tài này là áp dụng các thuật toán học sâu trong việc phát hiện các bệnh lý bằng ảnh X-quang phổi gồm bệnh Covid-19 và viêm phổi thông thường. Sau đó, so sánh kết quả thực nghiệm của các phương pháp học sâu.

### **2. Lịch sử giải quyết vấn đề**

Việc sử dụng hình ảnh X-quang phổi để phát hiện Covid-19 là một chủ đề mới trong lĩnh vực y học. Tuy nhiên, các nghiên cứu và phát triển trong lĩnh vực này đã diễn ra rất nhanh trong suốt đại dịch Covid-19.

Ở Trung Quốc, các nhà nghiên cứu đã sử dụng hình ảnh X-quang phổi để phát hiện Covid-19 và đưa ra các phương pháp tiên tiến để xử lý dữ liệu hình ảnh. Các nhà nghiên cứu đã tiến hành một nghiên cứu về sự phát hiện Covid-19 bằng cách sử dụng hình ảnh X-quang phổi, kết quả cho thấy phương pháp này có thể phát hiện các trường hợp Covid-19 với độ chính xác cao [14].

Ngoài ra, các công ty công nghệ như Amazon và Alibaba đã phát triển các ứng dụng dựa trên trí tuệ nhân tạo để phát hiện Covid-19 từ hình ảnh X-quang phổi. Ví dụ: Amazon đã phát triển một mô hình phát hiện Covid-19 sử dụng hình ảnh X-quang phổi và bộ dữ liệu COVIDx, trong khi đó Alibaba đã phát triển một hệ thống AI dựa

trên hình ảnh X-quang phổi để giúp nhân viên y tế chẩn đoán nhanh các trường hợp Covid-19. Các công ty công nghệ khác như Google, Microsoft và Facebook cũng đã tham gia nghiên cứu và phát triển các công nghệ tương tự.

Trong thời gian gần đây, các mô hình học sâu và trí tuệ nhân tạo như ResNet [3], Inception [15][16], EfficientNet [17][18] và các kiến trúc mạng neural convolutional (CNN) khác đã được sử dụng để xây dựng các mô hình phân loại X-quang phổi để phát hiện Covid-19. Những nỗ lực này đã giúp nâng cao độ chính xác và độ nhạy của việc phát hiện Covid-19 từ hình ảnh X-quang phổi.

### **3. Mục tiêu đề tài**

- Thu thập tập dữ liệu hình ảnh X-quang phổi.
- Tìm hiểu các kiến trúc mạng học sâu thường dùng để phân lớp ảnh X-quang phổi
- Tìm hiểu các kiến trúc học tự giám sát (Self-Supervised Learning) để nâng cao khả năng dự đoán nhãn.
- Xây dựng website hiển thị kết quả.

### **4. Đối tượng và phạm vi nghiên cứu**

- ✓ Đối tượng nghiên cứu trên tập dữ liệu 6000 ảnh X-quang phổi bao gồm 3 nhãn: Covid-19 (viêm đường hô hấp), Pneumonia (viêm phổi thông thường), Normal (không có bệnh).
- ✓ Phạm vi nghiên cứu trên các mô hình học sâu bao gồm: ResNet50, MobileNetV2, DenseNet121, VGG16, VGG19 và Self-Supervised Learning (MoCo CXR).

### **5. Phương pháp nghiên cứu**

- Tổng hợp và nghiên cứu các tài liệu liên quan đến những mô hình xử lý hình ảnh X-quang phổi.
- Thu thập dữ liệu hình ảnh X-quang phổi
- Tiền xử lý dữ liệu
- Xây dựng các mô hình học máy.
- Triển khai mô hình trên website.
- Kiểm thử và đánh giá kết quả.

---

## **6. Kết quả đạt được**

- Xây dựng thành công website dùng để dự đoán nhân của hình ảnh X-quang phổi được tải lên từ thiết bị cá nhân như máy tính.

---

## **7. Bố cục luận văn**

### **Phần giới thiệu**

Giới thiệu tổng quát về đề tài.

### **Phần nội dung**

**Chương 1** : Mô tả bài toán.

**Chương 2** : Thiết kế, cài đặt giải thuật.

**Chương 3** : Kiểm thử hệ thống và đánh giá độ chính xác, tốc độ của hệ thống.

### **Phần kết luận**

Trình bày kết quả đạt được và hướng phát triển hệ thống.

---

# PHẦN NỘI DUNG

## CHƯƠNG 1

### MÔ TẢ BÀI TOÁN

#### 1. Mô tả chi tiết bài toán

Hệ thống phân loại hình ảnh X-quang phổi dùng để hỗ trợ các bác sĩ trong quá trình chẩn đoán bệnh thông qua ảnh chụp X-quang phổi. Từ đó, góp phần tiết kiệm thời gian cũng như hạn chế rủi ro nhầm lẫn trong quá trình chẩn đoán.

Hệ thống có khả năng chẩn đoán bệnh Covid-19, viêm phổi thông thường và người bình thường không có bệnh. Các nhãn được dùng trong hệ thống phân loại bệnh về phổi bao gồm:

- ❖ Bình thường (Normal): Đây là trường hợp khi ảnh chụp X-quang của phổi không có bất kì dấu hiệu về bệnh lý.
- ❖ Viêm phổi (Pneumonia): Viêm phổi là bệnh lý liên quan đến phổi, do vi khuẩn hoặc virus gây ra. Viêm phổi có thể gây ra các triệu chứng như ho, khó thở, sốt và đau ngực.
- ❖ Viêm đường hô hấp (Covid-19): Căn bệnh này gây ra bởi một chủng virus corona dẫn đến tình trạng viêm đường hô hấp cấp tính ở người. Ngoài ra, nó còn dẫn đến các triệu chứng từ nhẹ đến nặng như: Đau đầu, khó chịu, sốt cao trên 38 độ, chảy nước mũi, ho hoặc đau họng, cảm thấy khó thở, đau cơ, mệt mỏi.

Người dùng có thể tải lên một tấm ảnh X-quang phổi và hệ thống sẽ sử dụng các mô hình đã được huấn luyện từ trước để dự đoán kết quả.

#### 2. Vấn đề và giải pháp liên quan đến bài toán

##### 2.1. Thu thập tập dữ liệu

Để xây dựng một mô hình dự đoán, việc đầu tiên chúng ta cần phải làm đó chính là thu thập tập dữ liệu để huấn luyện mô hình. Trong phạm vi nghiên cứu đề tài này, chúng ta sử dụng tập dữ liệu 6000 ảnh X-quang phổi được lấy từ Kaggle. Trong đó 5345 tấm ảnh bao gồm: 2000 ảnh Covid-19, 2000 ảnh Normal, 1345 ảnh Pneumonia được lấy từ tập dữ liệu có tên là “COVID-QU-Ex Dataset” do một nhóm các nhà nghiên cứu từ Đại học Qatar, Doha, Qatar và Đại học Dhaka, Bangladesh cùng với các cộng tác viên của họ từ Pakistan và Malaysia hợp tác với các bác sĩ y khoa tạo ra [\[7\]\[8\]](#). Và 655 có nhãn là Pneumonia được lấy từ [\[19\]](#).



## **2.2. Xây dựng các mô hình máy học**

Các mô hình phân loại ảnh chụp X-quang phổi đã được phát triển trong thời gian gần đây và đã đạt được kết quả tốt trong các bài toán dự đoán. Trong phạm vi nghiên cứu đề tài này, chúng ta sẽ sử dụng các mô hình như: DenseNet121, MobileNetV2, Resnet50V2, VGG16 và VGG19 để tiến hành huấn luyện cũng như đánh giá kết quả dự đoán của các mô hình trên. Bên cạnh đó, chúng ta còn nghiên cứu thêm về một loại mô hình học tập tự giám sát, cụ thể là mô hình Moco-CXR.

### 2.2.1. DenseNet121

DenseNet121 là một mô hình học sâu được sử dụng trong bài toán nhận dạng ảnh. Nó là một phiên bản của mô hình DenseNet, được đề xuất bởi Gao Huang, Zhuang Liu, và các cộng sự vào năm 2017 [6]. DenseNet là viết tắt của Dense Convolutional Network, nó được thiết kế để giảm thiểu hiện tượng mất thông tin khi mạng nơ ron trở nên sâu hơn.

Mô hình DenseNet121 bao gồm 121 lớp tích chập (convolution layer) và lớp tổng hợp (pooling layer). Nó có cấu trúc rất dày đặc, trong đó các lớp tích chập được kết nối trực tiếp với nhau để tạo ra một mô hình rất sâu. Sự kết nối trực tiếp này giúp mô hình tránh được vấn đề giảm gradient và tăng tính ổn định của quá trình huấn luyện. Mô hình DenseNet cũng sử dụng các khối (block) được gọi là Dense Block, trong đó mỗi khối chứa nhiều lớp tích chập được kết nối trực tiếp với nhau, cùng với một lớp tổng hợp và một lớp chuẩn hoá (normalization layer).

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	$112 \times 112$		$7 \times 7$ conv, stride 2		
Pooling	$56 \times 56$		$3 \times 3$ max pool, stride 2		
Dense Block (1)	$56 \times 56$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	$56 \times 56$		$1 \times 1$ conv		
	$28 \times 28$		$2 \times 2$ average pool, stride 2		
Dense Block (2)	$28 \times 28$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	$28 \times 28$		$1 \times 1$ conv		
	$14 \times 14$		$2 \times 2$ average pool, stride 2		
Dense Block (3)	$14 \times 14$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	$14 \times 14$		$1 \times 1$ conv		
	$7 \times 7$		$2 \times 2$ average pool, stride 2		
Dense Block (4)	$7 \times 7$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	$1 \times 1$		$7 \times 7$ global average pool		
			1000D fully-connected, softmax		

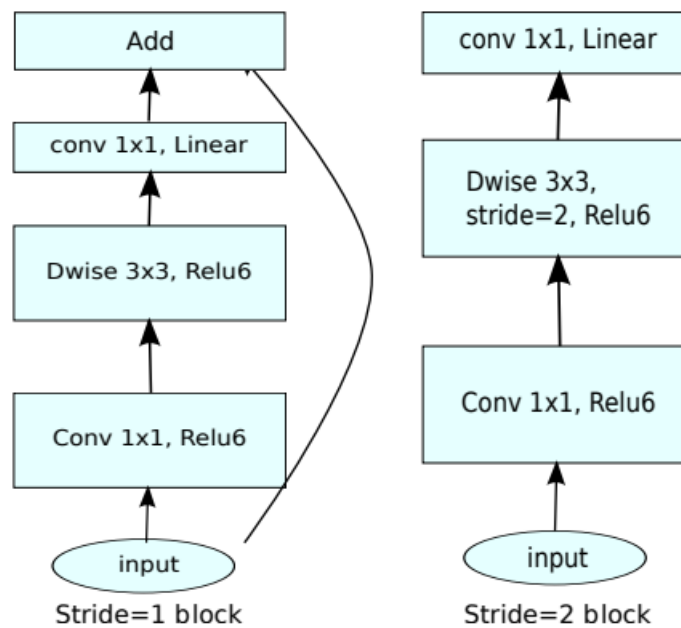
Hình 1. Kiến trúc của mô hình DenseNet121

Với cấu trúc dày đặc và kết nối trực tiếp giữa các lớp, mô hình DenseNet121 được đánh giá là hiệu quả và có khả năng học tốt trên các bộ dữ liệu ảnh lớn và phức tạp, chẳng hạn như bộ dữ liệu ImageNet.

### 2.2.2. MobileNetV2

MobileNetV2 là một kiến trúc mô hình CNN được thiết kế để chạy trên các thiết bị di động với tốc độ xử lý nhanh và tốn ít bộ nhớ. Nó được phát triển bởi nhóm nghiên cứu Google AI và được giới thiệu lần đầu tiên vào năm 2018 [5].

Mô hình MobileNetV2 sử dụng một số kỹ thuật để giảm số lượng tham số trong mô hình, nhưng vẫn giữ được hiệu quả của mô hình. Một trong những kỹ thuật đó là sử dụng depthwise separable convolution, trong đó lớp tích chập được chia thành hai lớp tích chập nhỏ hơn, một là depthwise convolution và một là pointwise convolution. Sử dụng kỹ thuật này giúp giảm số lượng tham số và tính toán, giảm kích thước của mô hình và cải thiện tốc độ xử lý.



**Hình 2. Kiến trúc mô hình MobileNetV2**

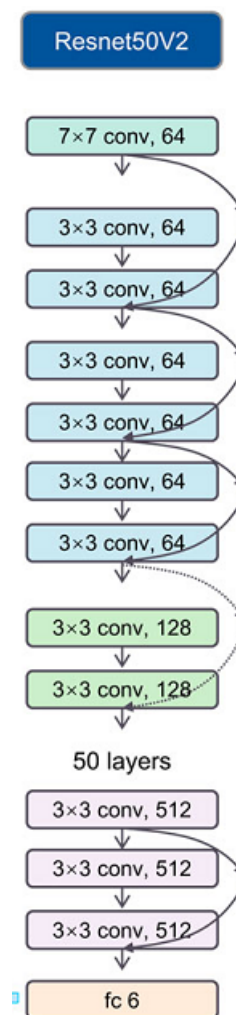
Mô hình MobileNetV2 đã được huấn luyện trên bộ dữ liệu ImageNet và có thể được sử dụng cho các bài toán phân loại ảnh, nhận dạng đối tượng và xử lý ảnh trên thiết bị di động. Nó cũng đã được sử dụng trong nhiều ứng dụng thực tế, như xử lý ảnh trong ứng dụng camera trên điện thoại di động và trích xuất đặc trưng trong các ứng dụng thị giác máy tính.

### 2.2.3. ResNet50V2

ResNet50V2 là một kiến trúc mô hình mạng nơ ron sử dụng trong bài toán phân loại hình ảnh. được đề xuất bởi nhóm nghiên cứu của Microsoft Research do Kaiming He làm trưởng nhóm vào năm 2016 [3] [9], đây là một phiên bản cải tiến của mô hình ResNet50 (Residual Network với 50 lớp), một trong những mô hình đạt được hiệu quả tốt nhất trong các cuộc thi phân loại hình ảnh.

ResNet50V2 sử dụng một kiến trúc mạng sâu với nhiều lớp tích chập và các lớp residual block. Residual block được thiết kế để giải quyết vấn đề đối diện với hiện tượng mất thông tin khi càng đi sâu vào các lớp của mạng, dẫn đến hiện tượng độ chính xác giảm.

Ngoài ra, ResNet50V2 sử dụng một số kỹ thuật tối ưu hóa như bottleneck architecture và SE-Net (Squeeze-and-Excitation Network) để tăng hiệu quả của mô hình và giảm số lượng tham số.



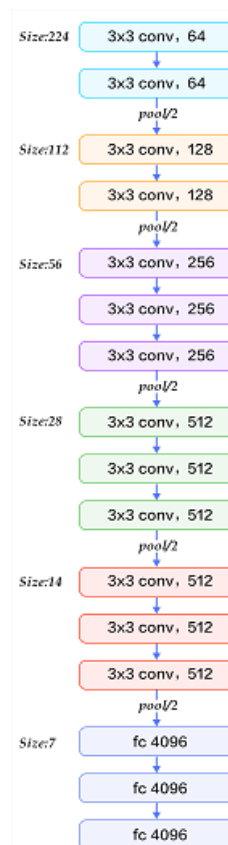
**Hình 3. Kiến trúc của mô hình ResNet50V2**

Mô hình ResNet50V2 đã được huấn luyện trên bộ dữ liệu ImageNet với hơn 1,2 triệu hình ảnh và 1.000 lớp. Nó có thể được sử dụng trong nhiều bài toán phân loại hình ảnh, bao gồm cả trong lĩnh vực y học, như phân loại ảnh X-quang phổi, phát hiện ung thư và phân loại bệnh lý.

#### 2.2.4. VGG16

VGG16 là một kiến trúc mô hình mạng nơ được sử dụng trong bài toán phân loại hình ảnh. Nó được giới thiệu bởi nhóm nghiên cứu tại Đại học Oxford vào năm 2014 trong bài báo "Very Deep Convolutional Networks for Large-Scale Image Recognition" [4].

VGG16 sử dụng một kiến trúc mạng sâu với 16 lớp, trong đó có 13 lớp tích chập và 3 lớp kết nối đầy đủ (fully connected layer). Các lớp tích chập đều có kích thước kernel là 3x3 và số lượng bộ lọc (filter) tăng dần từ 64 tới 512. Các lớp giảm kích thước (max pooling layer) được sử dụng để giảm kích thước của feature map.



Hình 4. Kiến trúc mô hình VGG16

Điểm đặc biệt của VGG16 là kiến trúc của nó được thiết kế đơn giản và đồng nhất, giúp việc triển khai trở nên dễ dàng. Nó cũng được huấn luyện trên bộ dữ liệu ImageNet với hơn 1,2 triệu hình ảnh và 1.000 lớp.

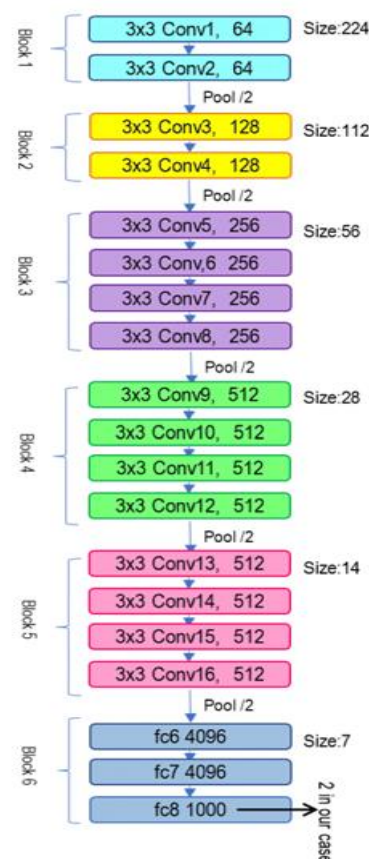
VGG16 đã được sử dụng trong nhiều bài toán phân loại hình ảnh, bao gồm phân loại đối tượng, phân loại chủ đề, phân loại khuôn mặt và phân loại ảnh y học.

### 2.2.5. VGG19

VGG19 là một kiến trúc mô hình mạng nơ ron được phát triển bởi nhóm nghiên cứu tại Đại học Oxford vào năm 2014 [4], như một phiên bản cải tiến của mô hình VGG16.

Tương tự như VGG16, VGG19 sử dụng một kiến trúc mạng sâu với các lớp tích chập và các lớp kết nối đầy đủ xen kẽ nhau. Tuy nhiên, khác với VGG16, VGG19 có thêm 3 lớp tích chập và 1 lớp kết nối đầy đủ, nâng tổng số lớp lên 19. Điều này giúp cho VGG19 có thể học được những feature phức tạp hơn và đạt hiệu suất tốt hơn trong các tác vụ phân loại hình ảnh.

Mỗi lớp tích chập của VGG19 đều sử dụng kernel có kích thước là 3x3 và số lượng filter tăng dần từ 64 đến 512. Các lớp giảm kích thước được sử dụng để giảm kích thước của feature map. Cuối cùng, các lớp kết nối đầy đủ được sử dụng để đưa ra dự đoán cho các lớp đầu ra.



**Hình 5. Kiến trúc mô hình VGG19**

VGG19 đã được huấn luyện trên bộ dữ liệu ImageNet với hơn 1,2 triệu hình ảnh và 1.000 lớp. Nó đã được sử dụng trong nhiều bài toán phân loại hình ảnh,

bao gồm cả trong lĩnh vực y học, như phân loại ảnh X-quang phổi, phát hiện ung thư và phân loại bệnh lý.

#### 2.2.6. Moco-CXR

MoCo được viết tắt là Momentum Contrast [1] đây là mô hình có thể tận dụng dữ liệu chưa được gán nhãn để tạo ra các mô hình được đào tạo trước để tinh chỉnh dữ liệu được gán nhãn sau này. Học tập tương phản về các biểu diễn trực quan đã nổi lên như một ứng cử viên hàng đầu cho việc tự giám sát và đã thể hiện hiệu suất vượt trội đối với các nhiệm vụ phân loại hình ảnh. Ngoài MoCo, chúng bao gồm các framework như SimCLR [2] và PIRL [10]. Tất cả các framework học tập tương phản đều liên quan đến việc tối đa hóa sự tương đồng giữa các cặp hình ảnh giống nhau so với các hình ảnh khác nhau thông qua hàm lỗi tương phản, mô hình được huấn luyện trước này buộc mô hình phải học các đặc trưng giống nhau của các hình ảnh. Các cách tiếp cận này thường khác nhau về cách chúng tạo ra các cặp hình ảnh giống nhau và khác nhau từ dữ liệu chưa được gán nhãn và cách dữ liệu được lấy mẫu trong quá trình đào tạo trước. Mặc dù MoCo và các phương pháp học tương phản khác đã cho thấy kết quả đầy hứa hẹn đối với các nhiệm vụ phân loại hình ảnh tự nhiên, nhưng ứng dụng của chúng đối với các cài đặt hình ảnh y học vẫn còn hạn chế. Ngoài ra moco còn sử dụng trọng số của mô hình imageNet để tinh chỉnh trọng số của mô hình trong quá trình huấn luyện.

Học tập tự giám sát (self-supervised learning) là một phương pháp học máy mà không yêu cầu sự can thiệp của con người để gán nhãn dữ liệu huấn luyện. Thay vào đó, phương pháp này dựa trên khả năng tự tạo ra các nhãn giả lập (fake labels) từ dữ liệu không gán nhãn và sử dụng chúng để huấn luyện mô hình. Với học tập tự giám sát, mô hình có thể học được những đặc trưng của dữ liệu mà không cần đến việc gán nhãn từ con người. Ví dụ, trong bài toán phân loại ảnh, chúng ta có thể sử dụng các phương pháp như cắt ra các vùng ảnh khác nhau (Image Segmentation) và đưa vào mô hình hoặc tạo ra các hình ảnh giả ngẫu nhiên từ các bức ảnh có sẵn (Data Augmentation). Các phương pháp học tập tự giám sát đang trở thành một chủ đề nghiên cứu rất hot trong lĩnh vực học máy, và đã cho thấy tiềm năng lớn trong việc giải quyết các bài toán huấn luyện mô hình với lượng dữ liệu không đủ hoặc không có nhãn.

Tác giả SOWRIRAJAN nghiên cứu về Moco đã chỉ ra rằng việc tăng cường dữ liệu bằng các phương pháp như: cắt xén một phần ngẫu nhiên của hình ảnh hay làm mờ đi bằng Gaussian sẽ dẫn đến việc làm thay đổi nhãn trong quá trình dự đoán. Hơn nữa, việc làm biến đổi ngẫu nhiên về màu sắc của hình ảnh hay đưa ảnh về màu xám cũng không có ý nghĩa đối với tập dữ liệu ảnh X-quang. Thay vào đó, tác giả sử dụng các phương pháp Data Augmentation như: xoay ảnh ngẫu nhiên (10 độ) hoặc lật ảnh ngang [12][13]. Kết quả của nghiên cứu cho thấy rằng việc mô hình sử dụng trọng số được huấn luyện trước của mô



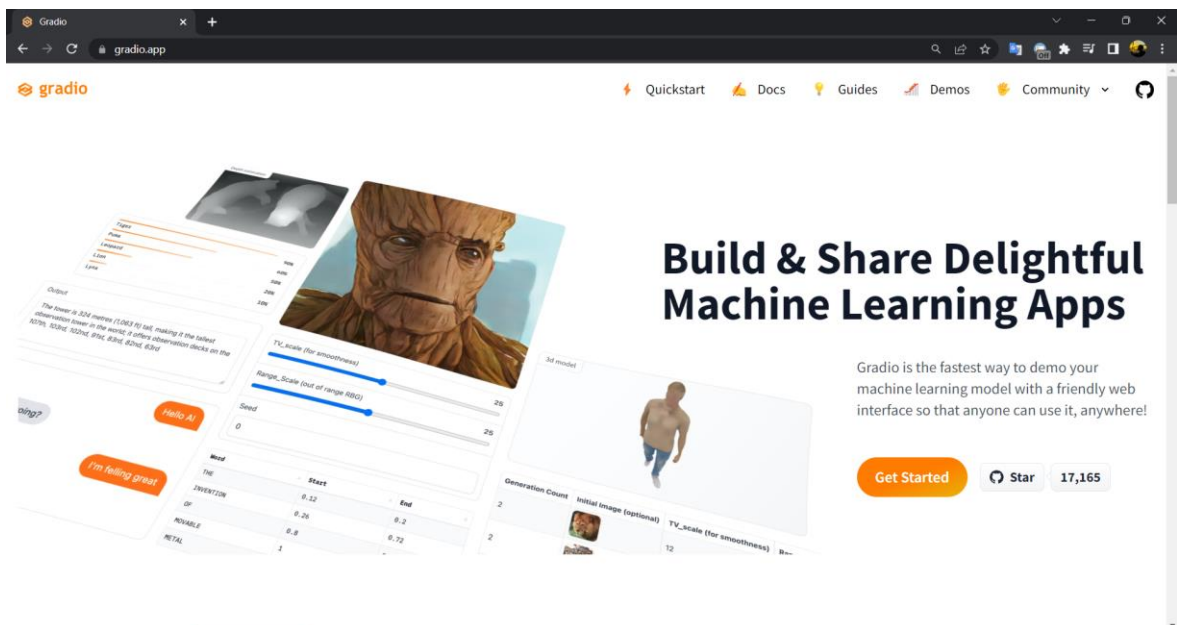
hình MoCo CXR luôn đạt độ chính xác cao hơn so với trọng số của mô hình imageNet. Từ đó, cho thấy rằng mô hình đào tạo trước MoCo-CXR tạo ra các biểu diễn chất lượng cao hơn so với đào tạo trước ImageNet.

### 3. Xây dựng giao diện cho hệ thống dự đoán

Gradio là một thư viện mã nguồn mở cho phép bạn tạo các giao diện đồ họa tương tác để thử nghiệm các mô hình ML. Gradio cung cấp cho người dùng một cách dễ dàng để triển khai các mô hình ML mà không cần kiến thức về lập trình hoặc thiết kế giao diện.

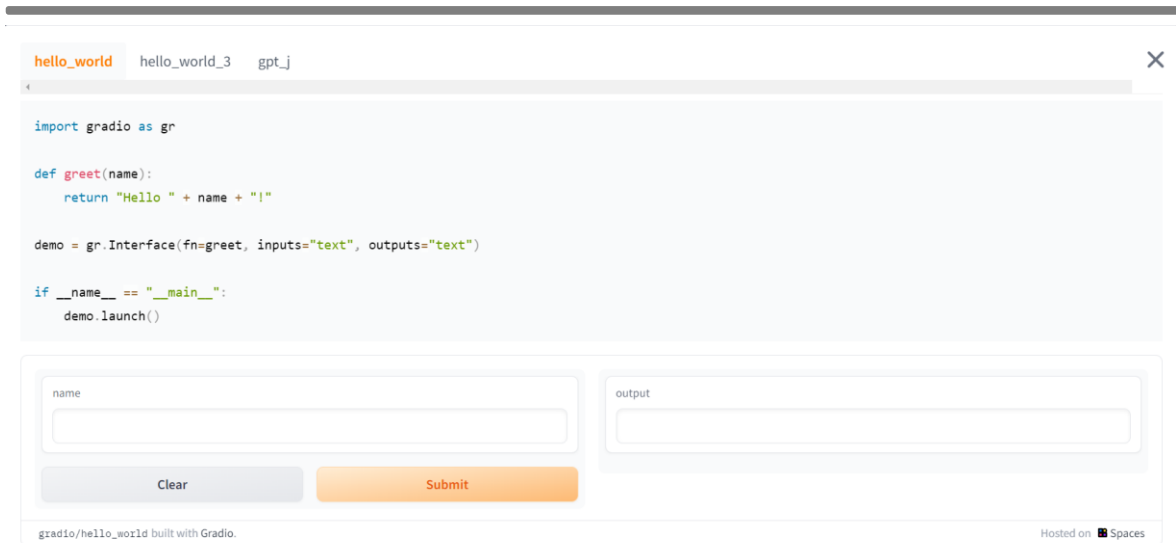
Với Gradio, bạn có thể tạo các giao diện đơn giản hoặc phức tạp để thử nghiệm mô hình của mình bằng cách trực tiếp tương tác với nó thông qua giao diện web. Gradio cho phép bạn tùy chỉnh giao diện người dùng, tải lên dữ liệu và hiển thị kết quả dự đoán của mô hình. Bạn có thể sử dụng Gradio để chia sẻ mô hình của mình với người dùng khác hoặc để kiểm tra và sửa lỗi mô hình của mình.

Gradio cung cấp cho người dùng một cách nhanh chóng và dễ dàng để triển khai các mô hình ML và xem kết quả của chúng, làm cho việc thử nghiệm và phát triển mô hình trở nên dễ dàng hơn.



**Hình 6. Ảnh minh họa trang chủ gradio**





**Hình 7. Xây dựng giao diện dễ dàng với demo gradio**

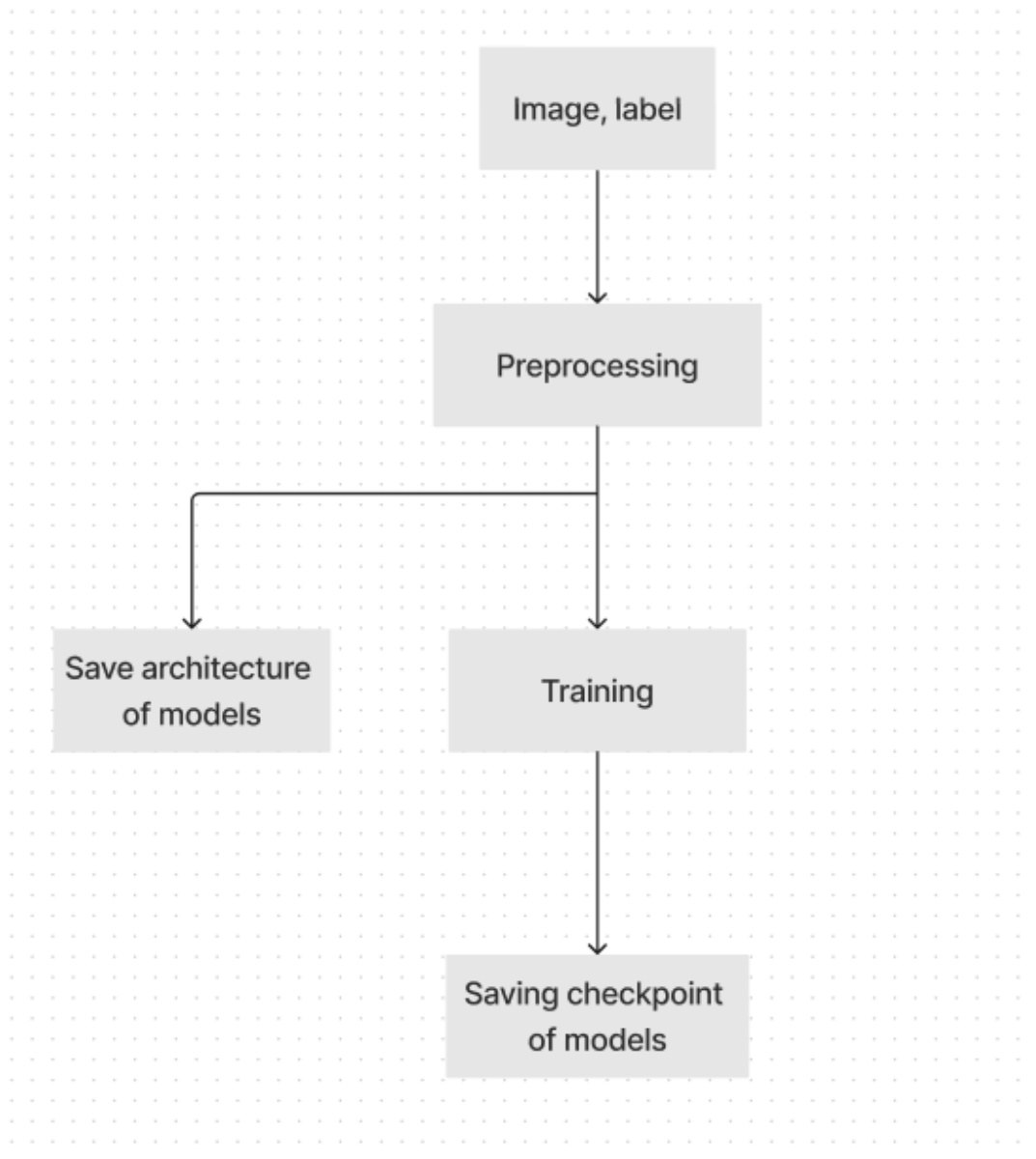
---

## CHƯƠNG 2

### THIẾT KẾ VÀ CÀI ĐẶT

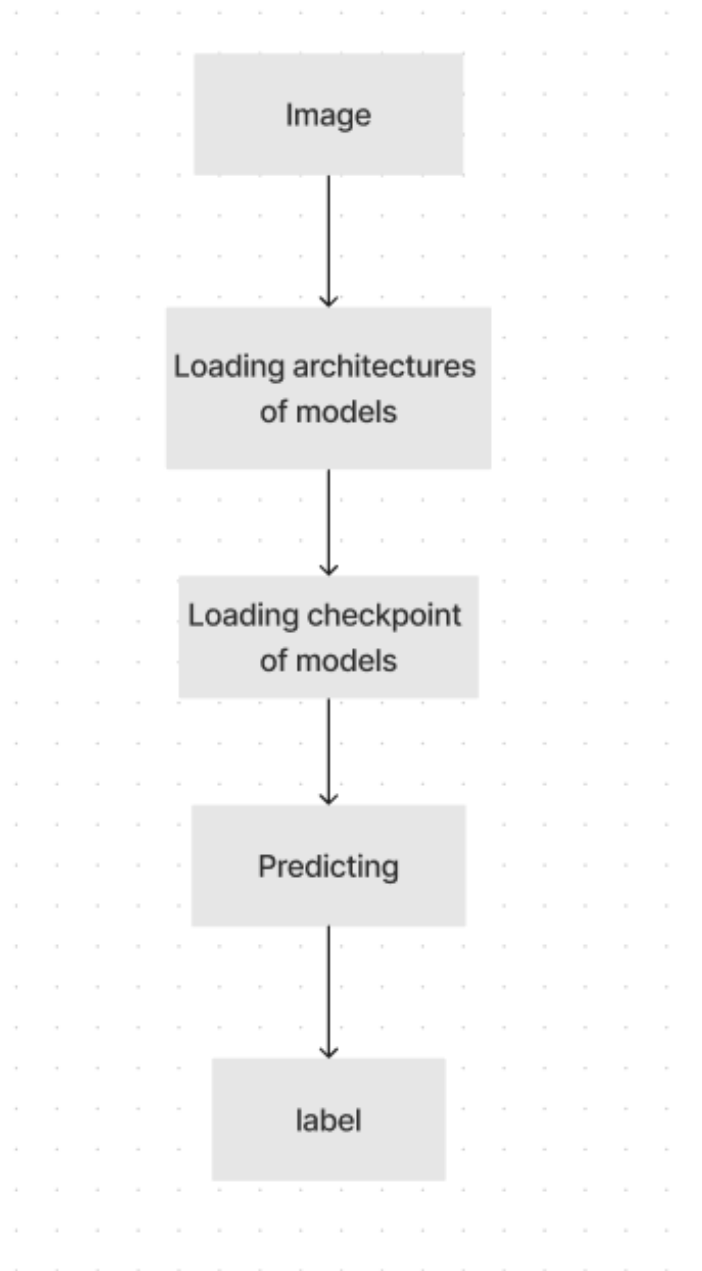
#### 1. Sơ đồ hệ thống sử dụng 5 mô hình truyền thống để dự đoán nhân

##### 1.1. Sơ đồ quá trình huấn luyện mô hình



**Hình 8. Sơ đồ tổng quát quá trình huấn luyện**

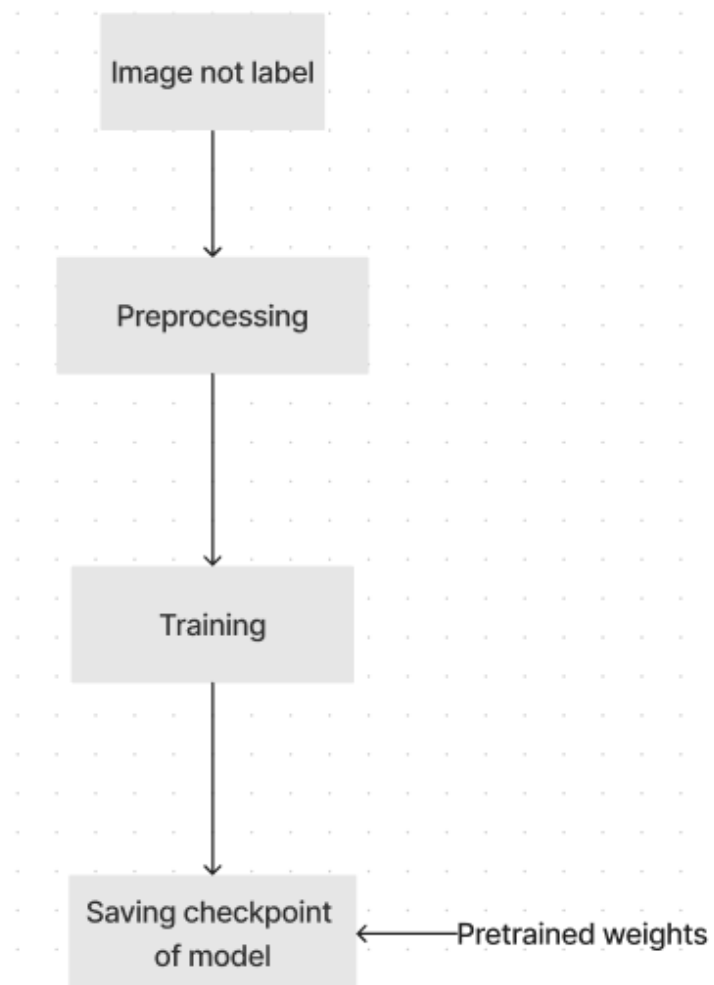
## 1.2. Sơ đồ quá trình dự đoán nhãn.



Hình 9. Sơ đồ tổng quát quá trình tải mô hình và dự đoán nhãn của 5 mô hình truyền thống

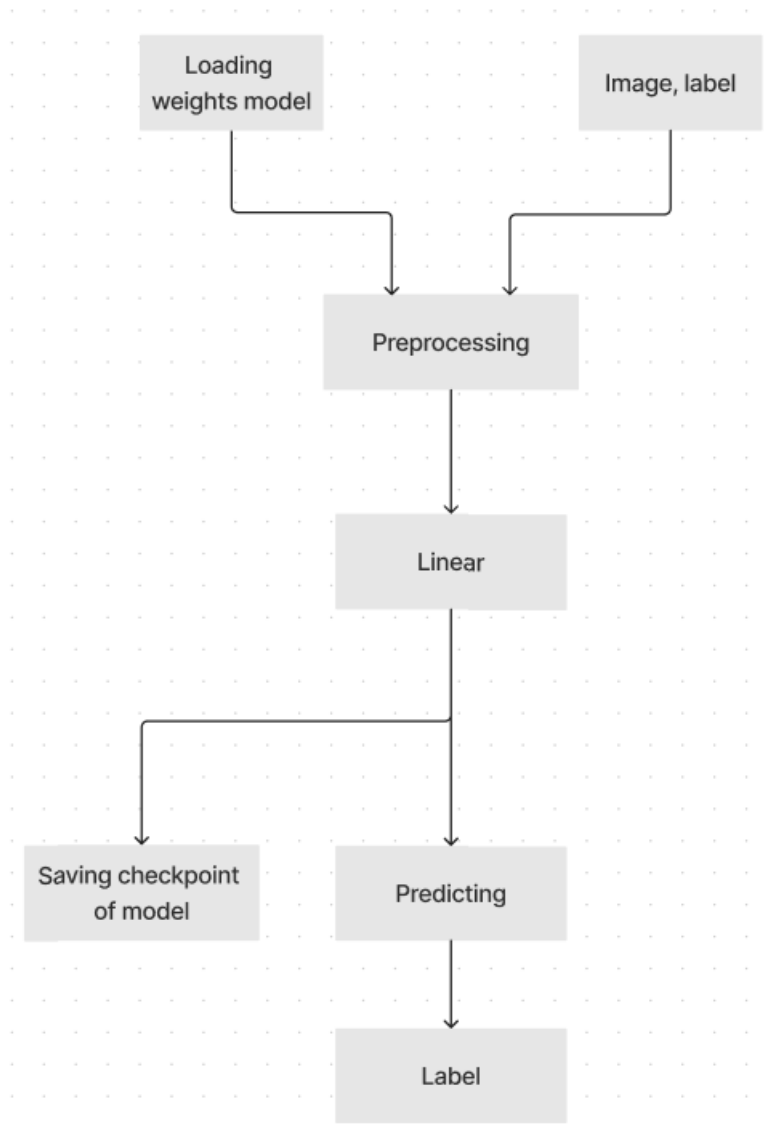
## 2. Sơ đồ sử dụng mô hình MoCo dùng để huấn luyện và dự đoán nhãn

### 2.1 Sơ đồ quá trình huấn luyện của mô hình Moco



Hình 10. Sơ đồ quá trình huấn luyện mô hình Moco với 120000 ảnh không nhãn

## 2.2. Sơ đồ quá trình điều chỉnh trọng số của mô hình MoCo.

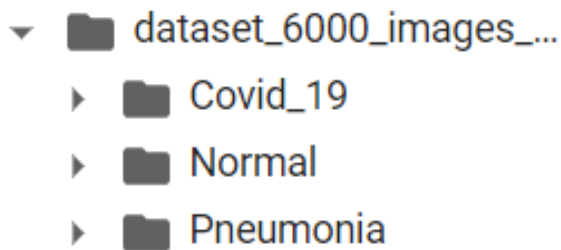


Hình 11. Sơ đồ quá trình điều chỉnh trọng số của mô hình Moco

### 3. Cài đặt hệ thống

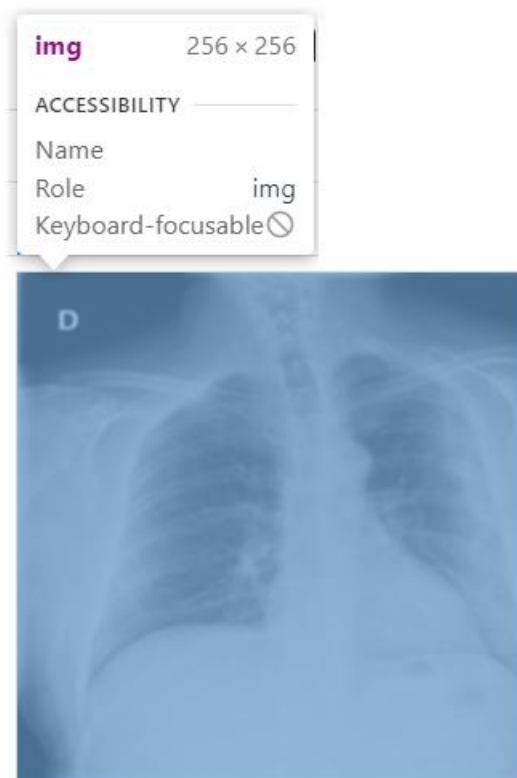
#### 3.1. Tiền xử lý dữ liệu

Sau khi lấy dữ liệu từ Kaggle, cấu trúc tập dữ liệu chúng ta có dạng như sau:  
Trong mỗi thứ mục Covid\_19, Normal, Pneumonia chứa 2000 ảnh.



Hình 12. Cấu trúc tập dữ liệu sau khi được lấy từ Kaggle

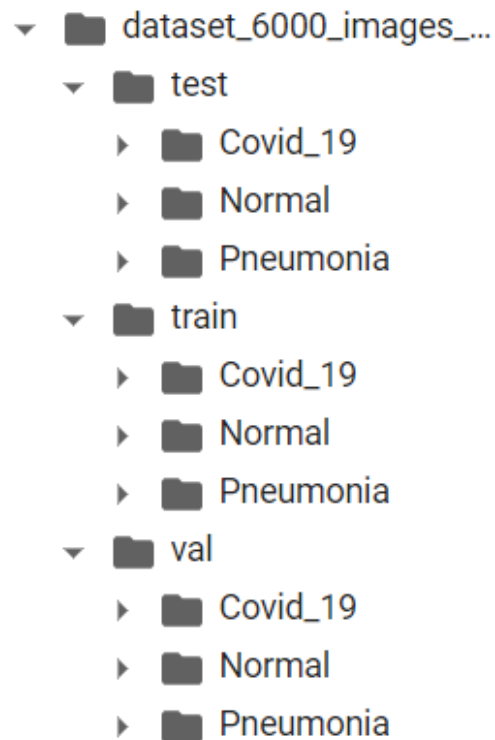
Do các ảnh trong tập dữ liệu có kích thước: 915x560 và 299x299 nên chúng ta tiến hành resize tập dữ liệu về kích thước chung là 256x256.



Hình 13. Ảnh sau khi đã resize về kích thước 256x256

---

Tập dữ liệu sau khi resize sẽ được tiến hành phân chia lần lượt thành 3 phần: train, test, val tương ứng với tỉ lệ 80%, 10% ,10%.



**Hình 14. Cấu trúc tập dữ liệu sau khi chia thành 3 phần để tiến hành huấn luyện mô hình**

Sau khi đã chia tập dữ liệu hoàn tất, chúng ta tạo lần lượt 3 file: train.csv, val.csv, test.csv. Các tệp này sẽ chứa các thông tin như: path (đường dẫn đến file hình ảnh), name(tên nhãn), label(nhãn sau khi được mã hoá về dạng số). Các nhãn được mã hoá tương ứng: Normal(0), Pneumonia(1), Covid\_19(2).

	path	name	label
1657	/content/drive/MyDrive	Normal	0
2644	/content/drive/MyDrive	Normal	0
3614	/content/drive/MyDrive	Pneumonia	1
3560	/content/drive/MyDrive	Pneumonia	1
2095	/content/drive/MyDrive	Normal	0
1873	/content/drive/MyDrive	Normal	0
3905	/content/drive/MyDrive	Pneumonia	1
544	/content/drive/MyDrive	Covid_19	2
1141	/content/drive/MyDrive	Covid_19	2
2683	/content/drive/MyDrive	Normal	0
1922	/content/drive/MyDrive	Normal	0
2342	/content/drive/MyDrive	Normal	0

**Hình 15. Nội dung file train.csv**

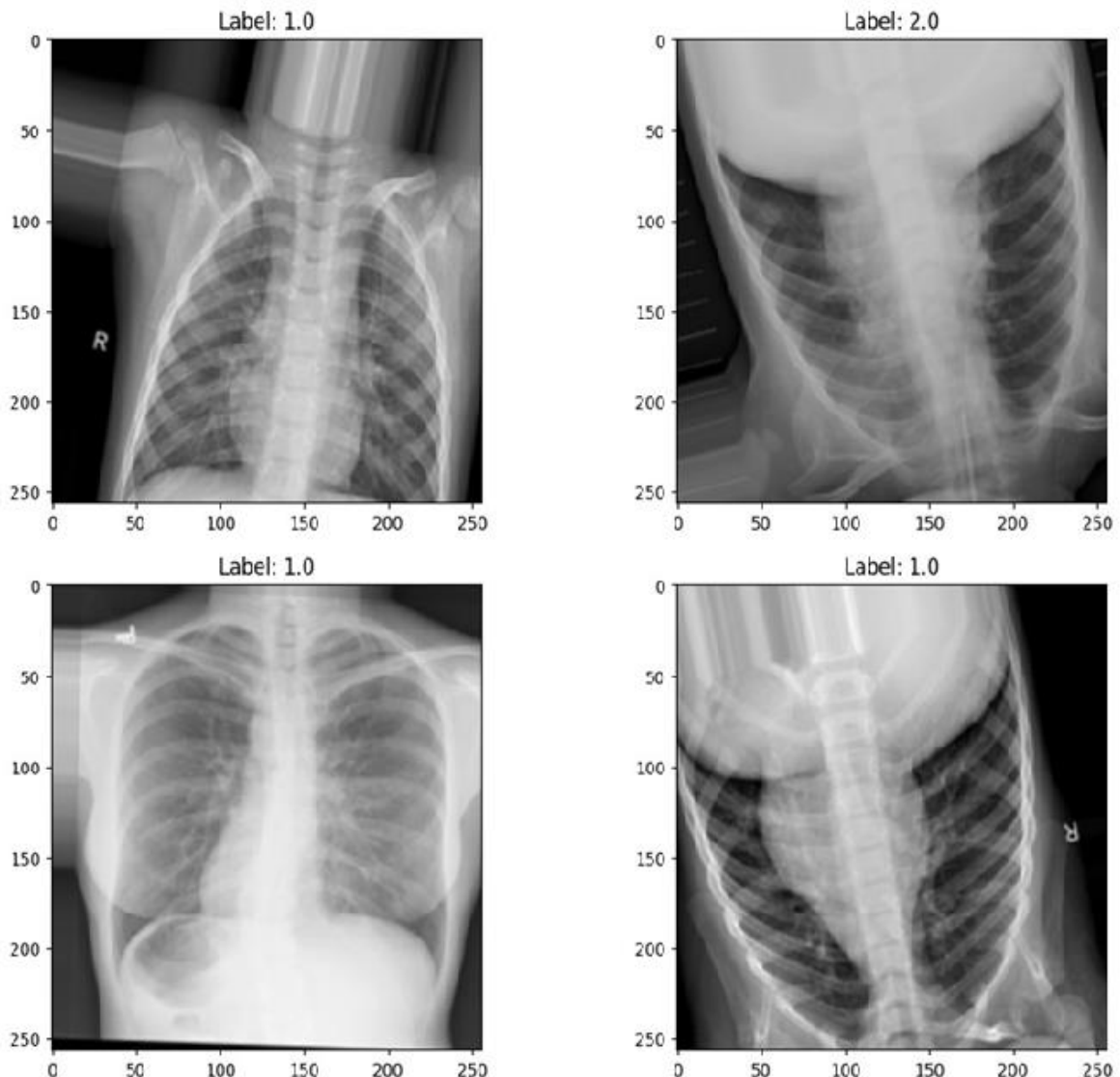
Khi đã có được các tập tin lưu trữ dữ liệu như mong muốn, chúng ta tiến hành tải dữ liệu từ các tập tin và tiền xử lý hình ảnh đầu vào trước khi đưa chúng vào mô hình để huấn luyện.

```
=====TRAIN=====
Covid_19      1600
Normal        1600
Pneumonia     1600
Name: name, dtype: int64
=====VAL=====
Covid_19       200
Pneumonia      200
Normal         200
Name: name, dtype: int64
=====TEST=====
Normal         200
Pneumonia      200
Covid_19       200
Name: name, dtype: int64
Training 4800 Validation 600 Test: 600
```

**Hình 16. Thông tin chi tiết về các tập dữ liệu**



Mỗi hình ảnh sẽ được thay đổi giá trị từng pixel ảnh về trong khoảng  $[0,1]$ , được xoay ảnh theo 1 góc  $(-20, 20)$ , lật theo chiều ngang, lật theo chiều dọc một cách ngẫu nhiên nhờ vào phương thức **ImageDataGenerator** của thư viện **Keras**. Ngoài ra các ảnh còn được tịnh tiến theo cả chiều ngang và chiều dọc ngẫu nhiên 20% ảnh gốc.



**Hình 17. Các ảnh X-quang sau khi được tiền xử lý**

### **3.2 Xây dựng các mô hình máy học truyền thống và huấn luyện**

Sau khi đã hoàn tất quá trình tiền xử lý dữ liệu, chúng ta tiến hành cài đặt các mô hình máy học để luyện mô hình. Các mô hình máy học được xây dựng bởi **Keras** bao gồm: DenseNet121, MobileNetV2, ResNet50V2, VGG16 và VGG19. Sử dụng trọng số **ImageNet** để huấn luyện lại toàn bộ mô hình.

**Keras** là một thư viện mã nguồn mở và miễn phí được sử dụng rộng rãi trong lĩnh vực học . Keras được phát triển bởi François Chollet và được viết bằng ngôn

ngữ Python. Keras cho phép người dùng dễ dàng xây dựng và huấn luyện các mô hình học sâu, như mạng nơ ron truyền thẳng, mạng nơ ron tái lập, và mạng nơ ron tích chập, v.v. Keras cung cấp cho người dùng một API đơn giản và dễ sử dụng để xây dựng các mô hình học sâu. Keras cũng có khả năng tích hợp với các thư viện học sâu khác như TensorFlow và Theano để tận dụng tối đa sức mạnh của các thư viện này. Keras cũng là một trong những thư viện học sâu phổ biến nhất hiện nay và được sử dụng rộng rãi trong cộng đồng học sâu. Keras cung cấp cho người dùng nhiều tính năng hữu ích như tùy chọn mô hình tự động, kiểm tra đa dạng và tiện lợi, và khả năng mở rộng và tùy chỉnh linh hoạt.

**ImageNet** là một tập dữ liệu lớn gồm khoảng 14 triệu hình ảnh được thu thập từ internet và được chia thành hơn 20.000 danh mục khác nhau. Tập dữ liệu này được sử dụng phổ biến trong các bài toán nhận dạng hình ảnh và phát triển mô hình học sâu. ImageNet được sử dụng trong các cuộc thi thường niên về nhận dạng hình ảnh, như cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) đã được tổ chức từ năm 2010 đến năm 2017. Các đội tham gia cuộc thi phải xây dựng các mô hình học sâu để phân loại hình ảnh trong tập dữ liệu ImageNet, và các kết quả của cuộc thi đã đóng góp đáng kể cho sự phát triển của học sâu và computer vision. Các mô hình học sâu phát triển trên ImageNet đã đạt được kết quả tốt trên nhiều bài toán nhận dạng hình ảnh khác, như phân loại chủ đề trong các bài viết trên mạng, phân loại ảnh y học, phát hiện đối tượng, v.v.

Do đây là bài toán phân loại ảnh có 3 lớp nên chúng ta sẽ xây dựng tầng Dense cuối cùng có 3 nơ ron dùng để dự đoán nhãn của ảnh đầu vào. Mô hình sử dụng thuật toán tối ưu hoá “Adam”, hàm lỗi “sparse\_categorical\_crossentropy” do nhãn đầu vào được mã hoá về số nguyên và hàm kích hoạt softmax.

**SparseCategoricalCrossentropy**: được sử dụng khi label y được biểu diễn dưới dạng các số nguyên thay vì one-hot encoding. Điều này có nghĩa là, label y sẽ là một vector các số nguyên tương ứng với các nhãn. Ví dụ, nếu bài toán phân loại 3 nhãn (cat, dog, bird), nếu một ảnh thuộc nhãn "dog", thì label của ảnh đó sẽ được biểu diễn là số nguyên 1. khi sử dụng SparseCategoricalCrossentropy, mô hình sẽ yêu cầu nhãn đầu vào phải có dạng các số nguyên.

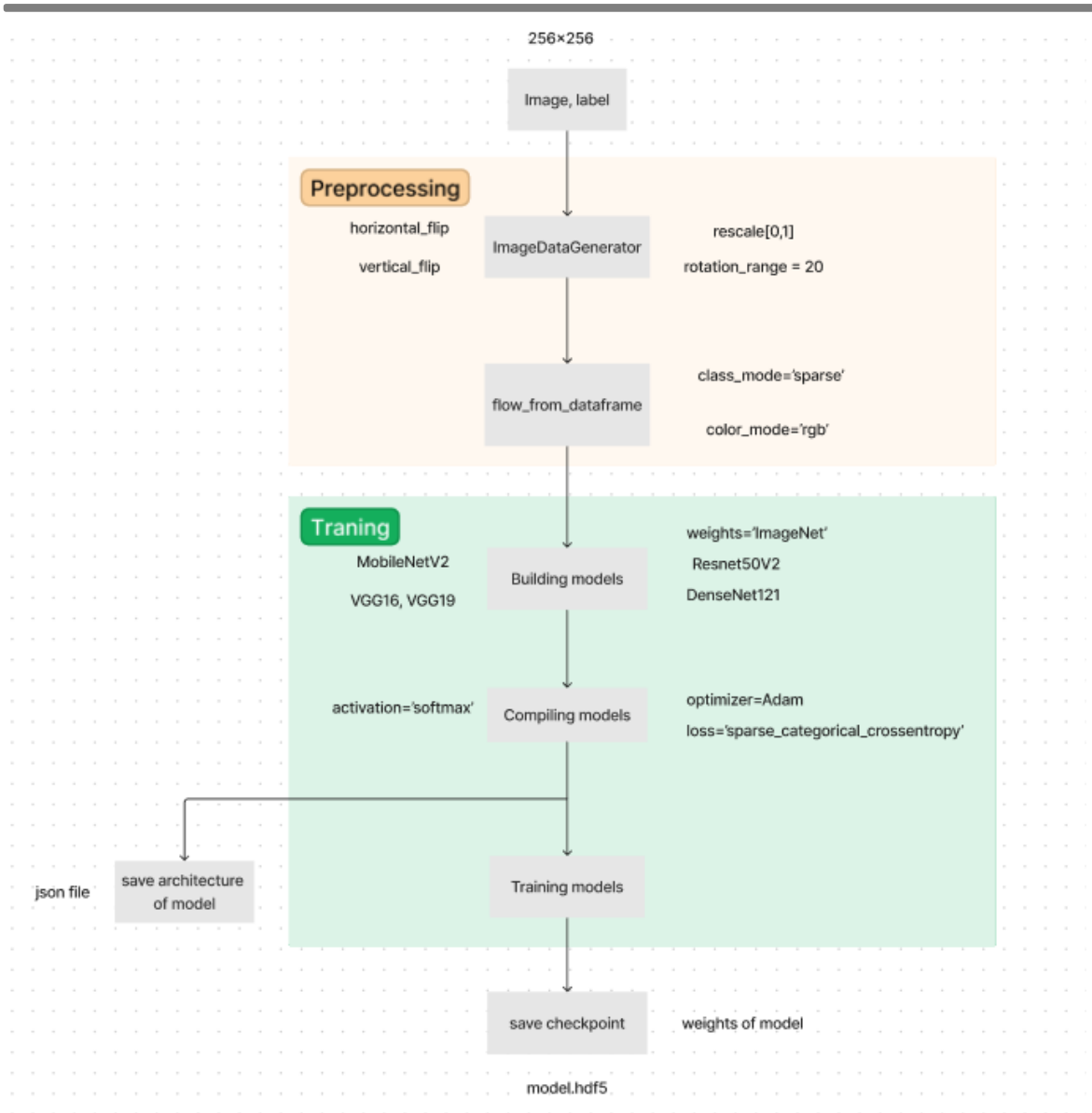
**Softmax** là một hàm số được sử dụng trong mô hình học sâu để chuyển đổi các giá trị đầu ra thành xác suất. Nó thường được sử dụng trong các bài toán phân loại, nơi mục tiêu là dự đoán lớp của một điểm dữ liệu đầu vào. Hàm Softmax chuyển đổi một vector đầu ra bất kỳ thành một vector xác suất, sao cho tổng của tất cả các phần tử trong vector đó bằng 1. Nó được định nghĩa như sau:

---

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$$

Trong đó,  $x$  là vector đầu ra đầu vào,  $K$  là số lượng lớp và  $i$  là chỉ số lớp được dự đoán. Hàm Softmax giúp chúng ta tính toán xác suất của mỗi lớp dựa trên giá trị đầu ra của mô hình. Với xác suất này, chúng ta có thể lựa chọn lớp có xác suất cao nhất là kết quả của bài toán phân loại.

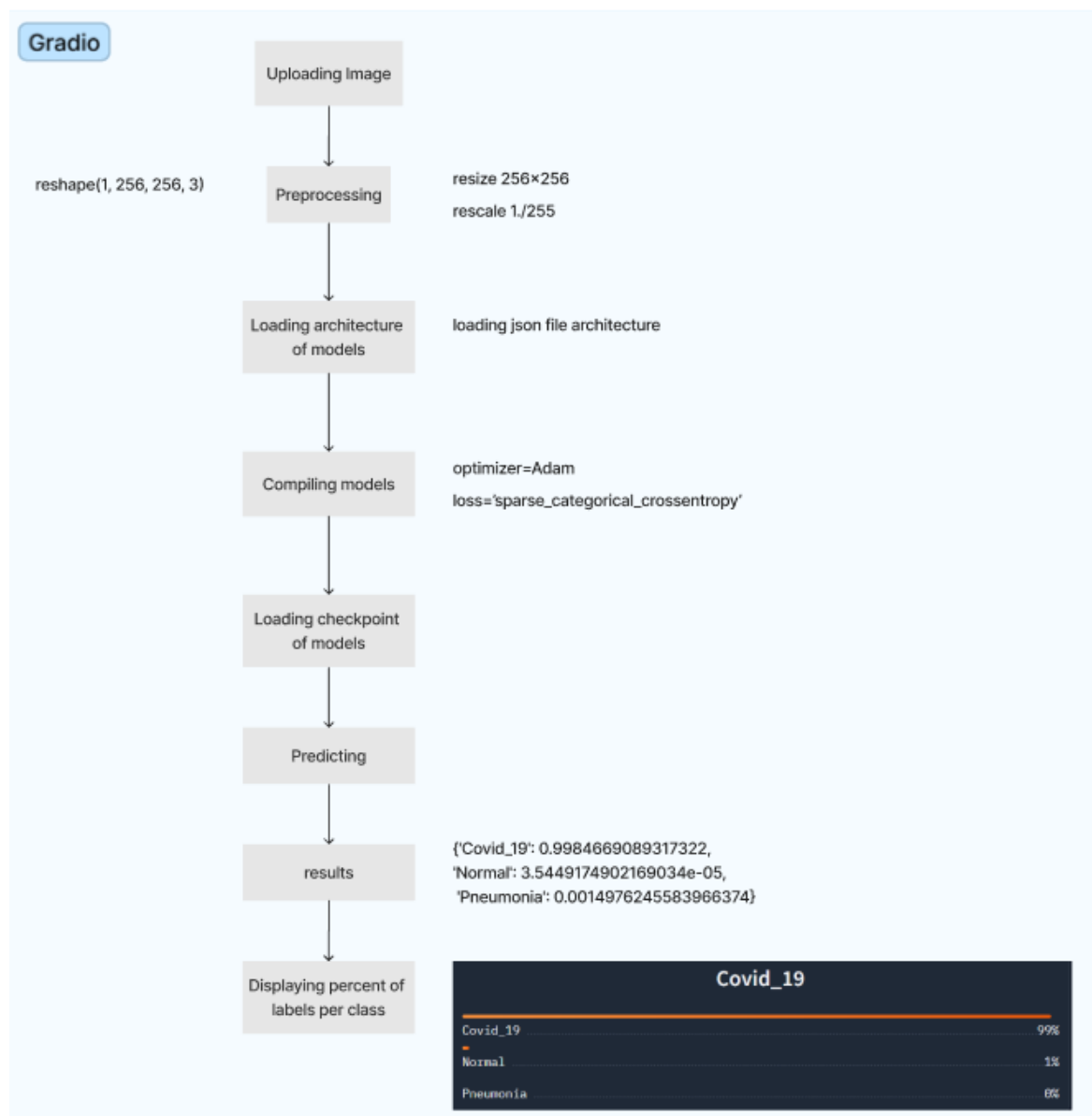
Trong suốt quá trình huấn luyện, chúng ta sẽ đánh giá giá trị hàm lỗi trên tập validation. Nếu giá trị này không giảm trong suốt 30 epochs chúng ta sẽ dừng quá trình huấn luyện. Khi đã hoàn tất quá trình huấn luyện, chúng ta sẽ thu được các tập tin có đuôi .json dùng để lưu các kiến trúc của mô hình và .h5 để lưu các trọng số của mô hình trong suốt quá trình huấn luyện.



Hình 18. Sơ đồ chi tiết quá trình huấn luyện mô hình truyền thống

### 3.3. Tải mô hình đã được huấn luyện và xây dựng giao diện website để dự đoán kết quả

Tương tự quá trình huấn luyện, để có thể sử dụng mô hình dùng để dự đoán ảnh, chúng ta cần phải khởi tạo mô hình bằng cách tải lên các kiến trúc của mô hình và trọng số của nó để có thể tiến hành dự đoán. Kế tiếp chúng ta sẽ sử dụng thư viện gradio để hỗ trợ xây dựng giao diện website cho các mô hình máy học. Cuối cùng là tải dữ liệu hình ảnh lên bằng giao diện trang web và hiển thị kết quả dự đoán của mô hình.



Hình 19. Sơ đồ quá trình tải mô hình và dự đoán kết quả trên website bằng thư viện gradio

#### 4. Tiến hành huấn luyện mô hình MoCo-CXR

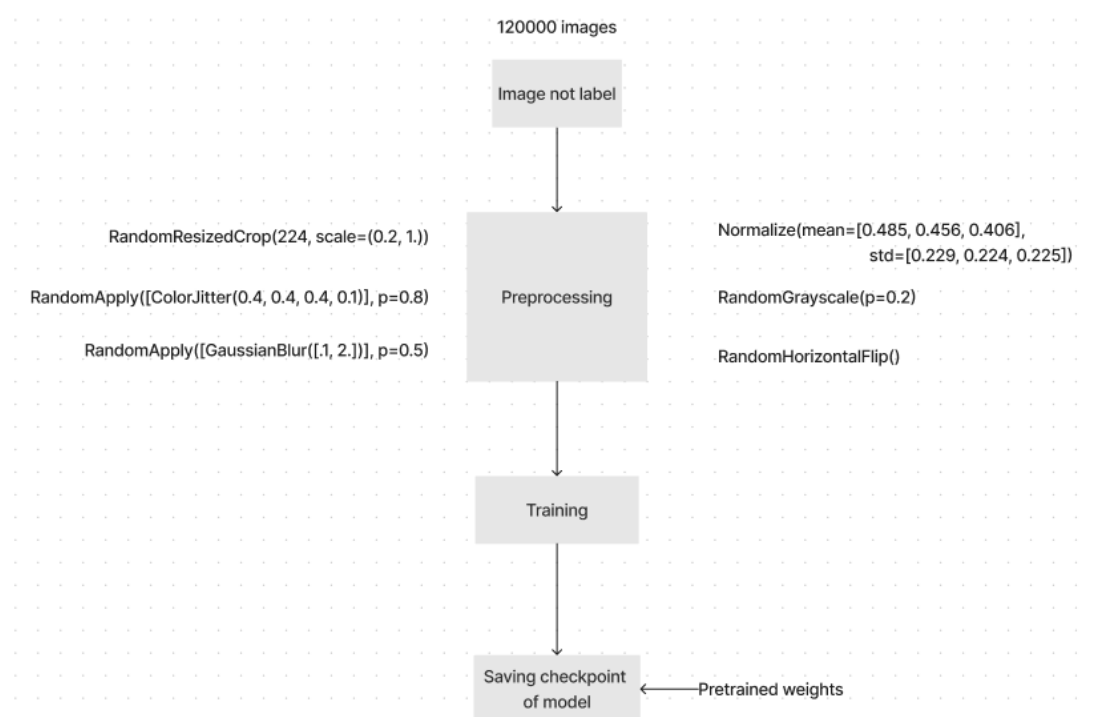
Trong phạm vi nghiên cứu của đề tài này chúng ta sẽ tiến hành huấn luyện và đánh giá kết quả từ mô hình MoCo sử dụng trọng số Pretrained của mô hình đã được huấn luyện trên tập dữ liệu 120000 ảnh X-quang không nhãn được lấy ngẫu nhiên từ bộ dữ liệu CheXpert [11], đây là một bộ dữ liệu lớn chứa 224 nghìn hình ảnh X-quang ngực của 65 nghìn bệnh nhân được dán nhãn, kích thước của mỗi ảnh là 320x320, sử dụng kiến trúc của mô hình Moco để trích xuất đặc trưng và tinh chỉnh lại trọng số của mô hình. Chúng ta sẽ tiến hành huấn luyện để tinh chỉnh lại trọng số của mô hình huấn luyện trên 2 tập dữ liệu:

- Tập dữ liệu 6000 ảnh X-quang mà chúng ta đang nghiên cứu (3 lớp).
- Tập dữ liệu 300 ảnh X-quang được lấy từ tập dữ liệu 6000 ảnh ở bên trên.

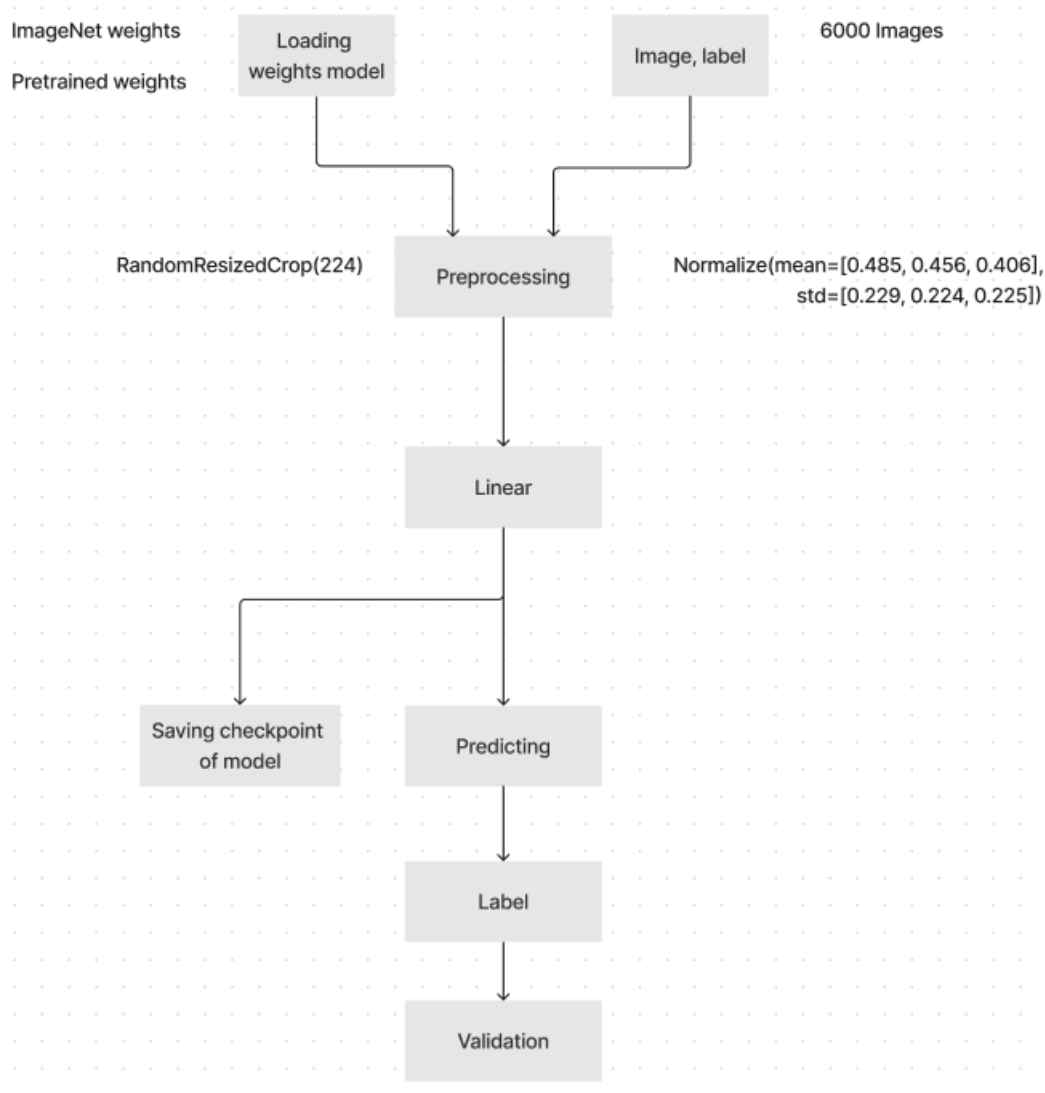
Tương ứng 100 ảnh đầu tiên của mỗi lớp.

Chúng ta sẽ tiến hành tinh chỉnh lại trọng số của mô hình bằng cách truyền vào trọng số của mô hình ImageNet và Pretrained để đánh giá kết quả, đối với các mô hình truyền thống ở bên trên chúng ta sẽ cập nhật lại toàn bộ trọng số của mô hình huấn luyện, riêng ở phần này chúng ta sẽ chỉ cập nhật lại trọng số của tầng cuối cùng của mô hình và đóng băng tất cả các tầng còn lại.

##### 4.1. Sơ đồ chi tiết quá trình huấn luyện mô hình Moco và đánh giá kết quả



Hình 20. Sơ đồ chi tiết quá trình huấn luyện mô hình Moco sử dụng tập dữ liệu 120000 ảnh



Hình 21. Sơ đồ mô tả chi tiết quá trình tinh chỉnh trọng số của mô hình MoCo sử dụng 2 trọng số khác nhau

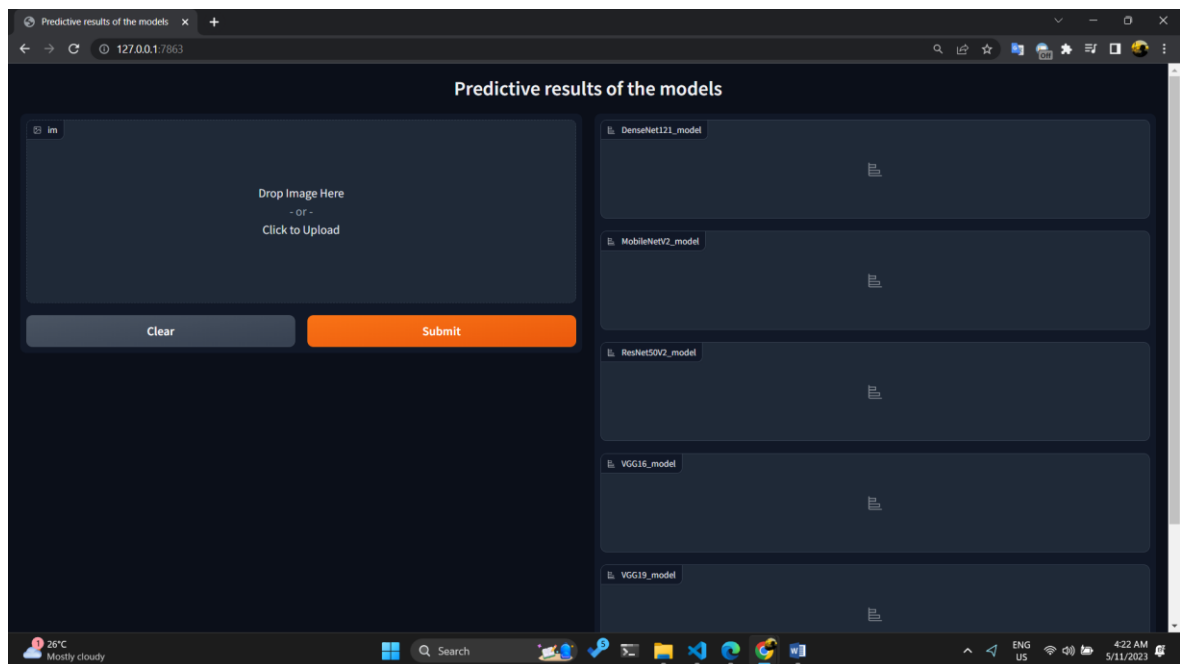
---

## CHƯƠNG 3

### KẾT QUẢ THỰC NGHIỆM

#### 1. Giao diện sản phẩm

Xây dựng thành công giao diện trang web dùng để dự đoán nhãn của hình ảnh X-quang phổi.



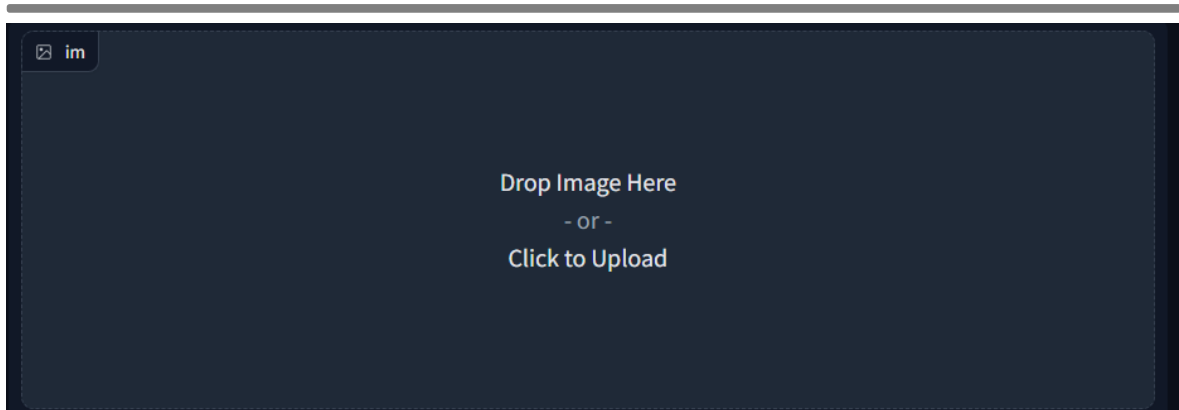
Hình 22. Giao diện website dùng để dự đoán hình ảnh X-quang phổi

#### 2. Giới thiệu các tính năng

Trang web có tính năng tải ảnh X-quang phổi từ máy tính cá nhân, tiến hành dự đoán và hiển thị đồng thời cùng lúc các kết quả từ 5 mô hình được huấn luyện.

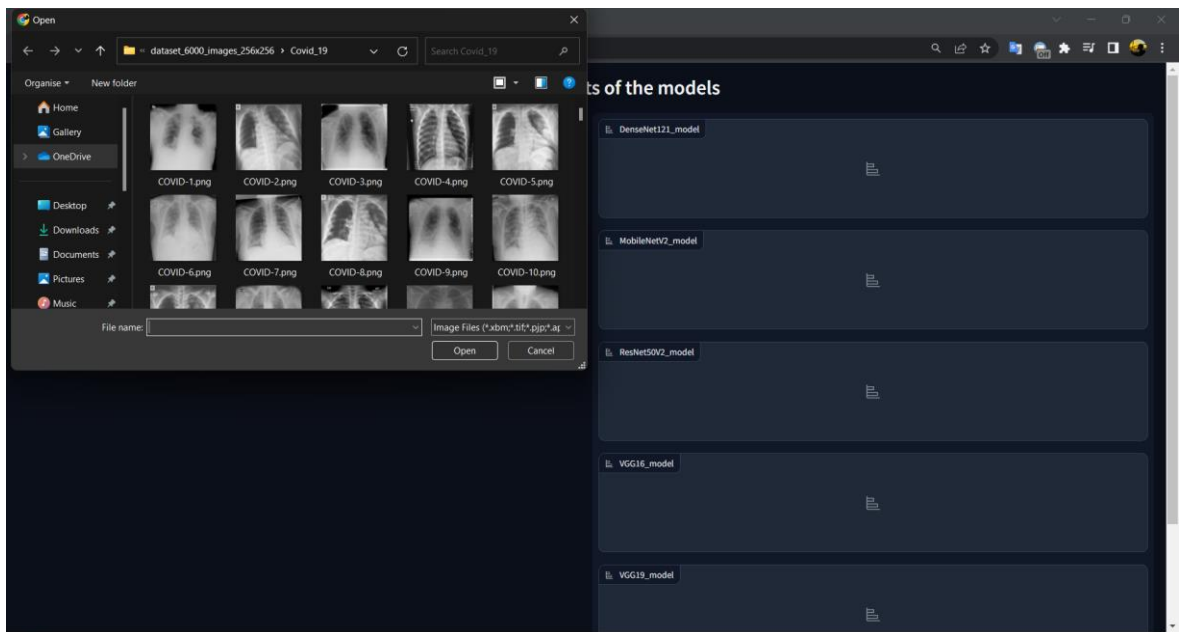
- Chức năng tải ảnh từ máy tính cá nhân: Để tải hình ảnh ta click vào phạm vi này để chọn ảnh tải lên từ máy tính.





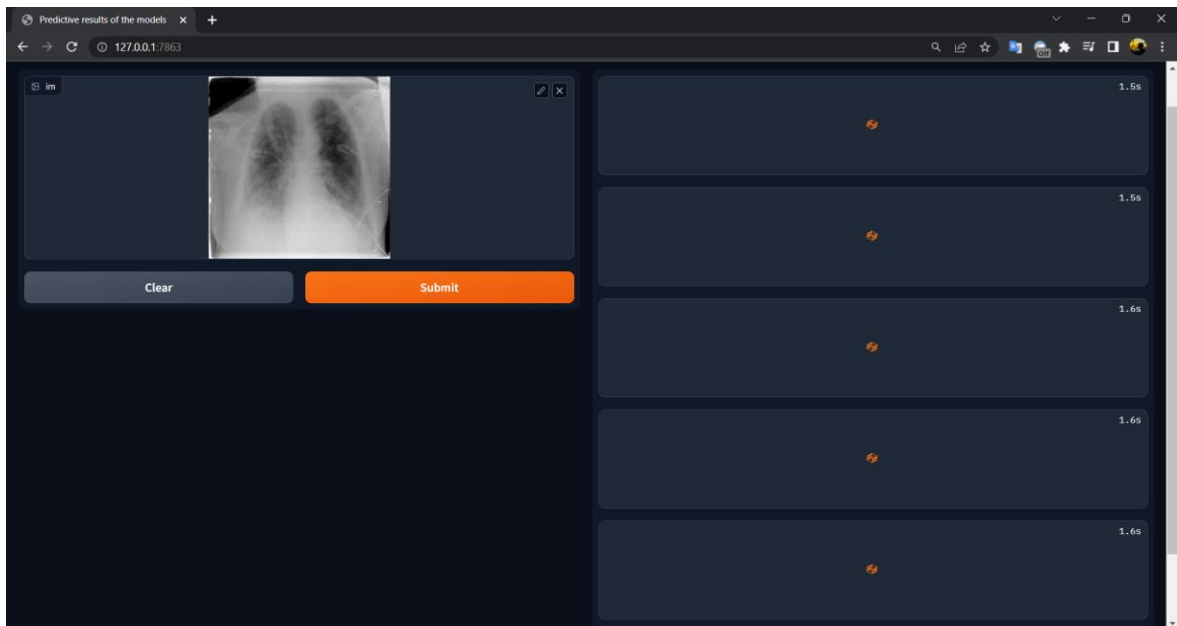
**Hình 23. Vị trí tải ảnh**

Sau khi click vào trang web sẽ hiển thị các thư mục trên máy tính để người dùng có thể tải lên và dự đoán kết quả.



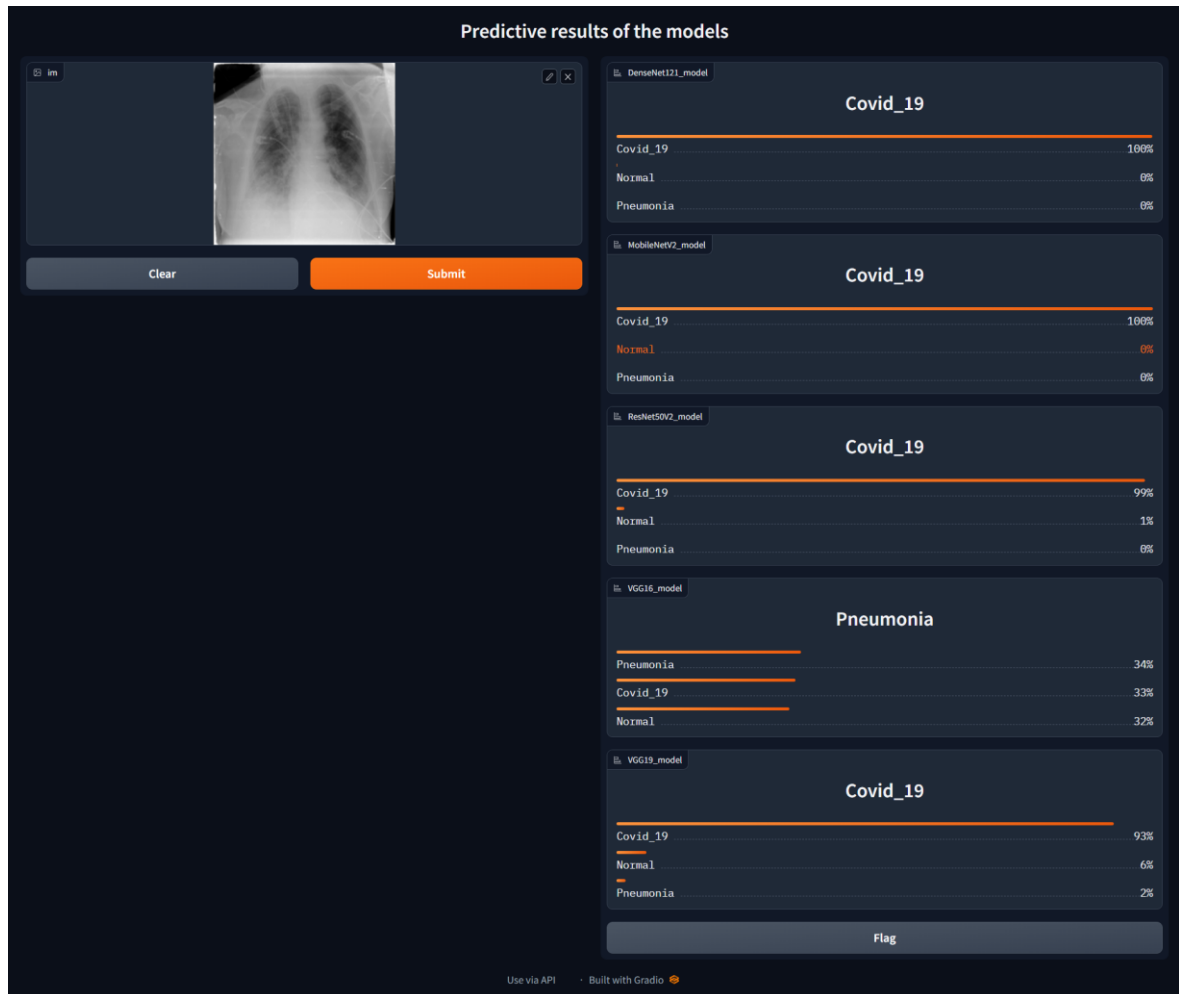
**Hình 24. Giao diện tải ảnh**

- Sau khi chọn ảnh đã tải lên, chúng ta sẽ click vào ô submit để hệ thống bắt đầu dự đoán.



**Hình 25. Hệ thống đang tiến hành dự đoán nhãn của tám ảnh đã tải lên**

- Kết quả sau khi hệ thống dự đoán



Hình 26. Kết quả dự đoán của website

### 3. Kết quả kiểm thử và đánh giá

Giao diện website được xây dựng bằng Gradio đây là một thư viện mã nguồn mở cho phép bạn tạo các giao diện đồ họa tương tác để thử nghiệm các mô hình ML, dễ dàng để triển khai các mô hình ML mà không cần kiến thức về lập trình hoặc thiết kế giao diện. Hệ thống đã thành công sử dụng các mô hình máy học để dự đoán nhãn của hình ảnh tải lên.

#### 3.1. Các chỉ số đánh giá

Dưới đây là một số phương pháp đánh giá các mô hình:

- **Accuracy:** cách đánh giá này đơn giản tính tỉ lệ các lớp đã phân loại đúng / tổng số dự đoán.
- **Confusion matrix:** là một ma trận tổng quát thể hiện kết quả phân loại chính xác và quả phân loại sai được tạo ra bởi một mô hình phân loại. Đây là một ma trận vuông với kích thước các chiều bằng số lượng lớp dữ liệu. Giá trị tại hàng

thứ  $i$ , cột thứ  $j$  là số lượng điểm lẽ ra thuộc vào class  $i$  nhưng lại được dự đoán là thuộc vào class  $j$ .

- **Precision / Recall:**

- **Precision** là tỉ lệ số dự đoán (ý định hay thực thể) mà mô hình dự đoán đúng (TP) trên tổng số dự đoán số mô hình dự đoán là Positive (TP+FP), theo công thức:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$0 < \text{Precision} \leq 1$ , Precision càng lớn có nghĩa là độ chính xác của các dự đoán ý định/thực thể được càng cao. Precision = 1, tức là tất cả số dự đoán của mô hình dự đoán là Positive đều đúng, hay không có dự đoán nào có nhãn là Negative mà mô hình dự đoán nhầm là Positive.

- **Recall** là tỉ lệ số dự đoán (ý định hay thực thể) mà mô hình dự đoán thực sự đúng (TP) trên tổng số dự đoán đúng (hay tổng số dự đoán được gán nhãn là Positive ban đầu TP+FN), chỉ số này được tính theo công thức:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall càng cao, tức là số dự đoán là positive bị bỏ sót càng ít. Recall = 1, tức là tất cả số dự đoán có nhãn là Positive đều được mô hình nhận ra.

- **F1 – score:** chỉ sử dụng Precision hay chỉ có Recall thì không đánh giá được chất lượng mô hình.

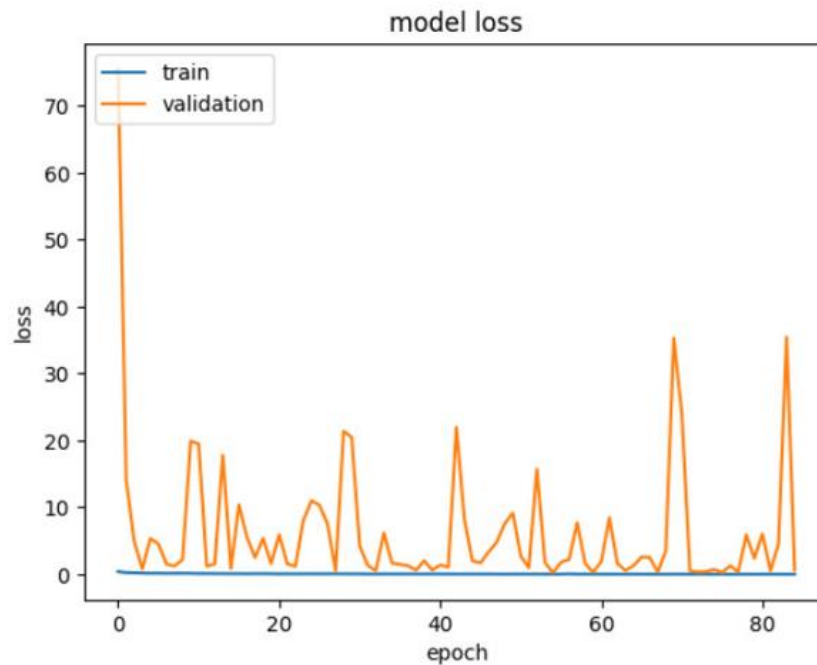
Khi đó F1-score được sử dụng. F1-score là trung bình điều hòa của precision và recall (giả sử hai đại lượng này khác 0). F1-score được tính theo công thức:

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

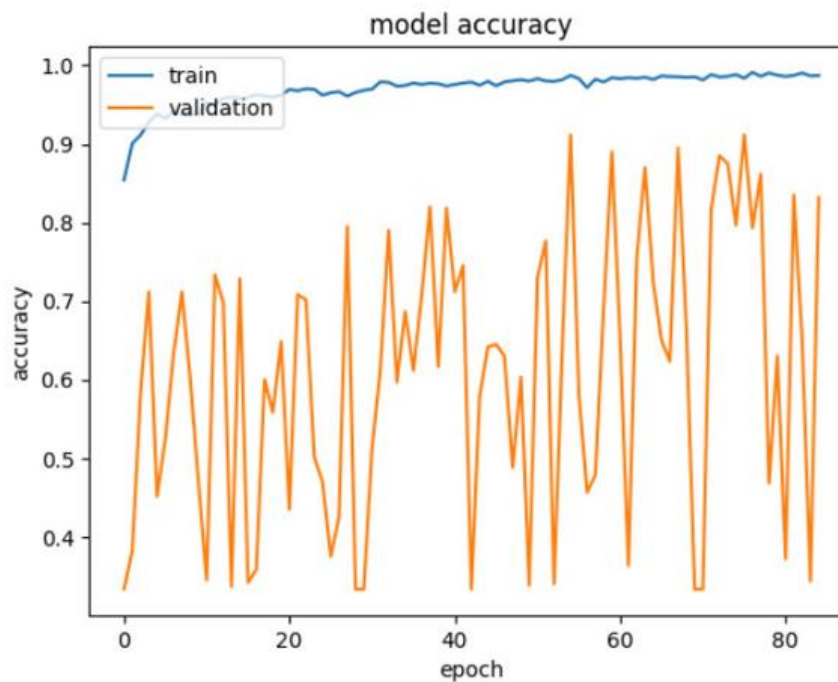
F1-score có giá trị nằm trong khoảng (0, 1), F1-score càng cao thì mô hình phân loại càng tốt.

### 3.2. Kết quả thực nghiệm

- Quá trình huấn luyện 5 mô hình truyền thống
- + ResNet50V2:

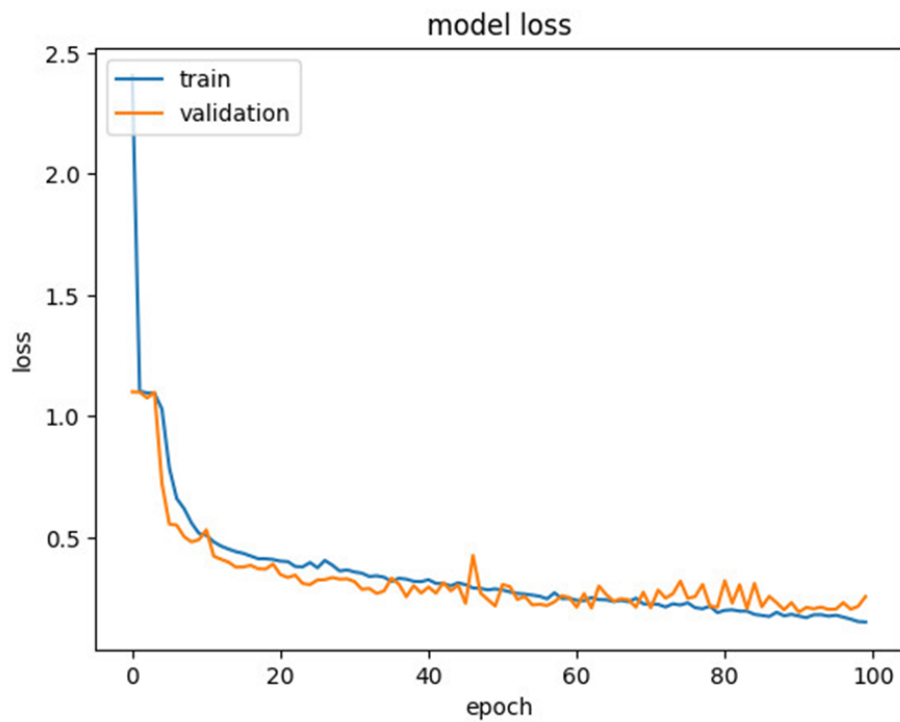


Hình 27. ResNet50V2 model loss

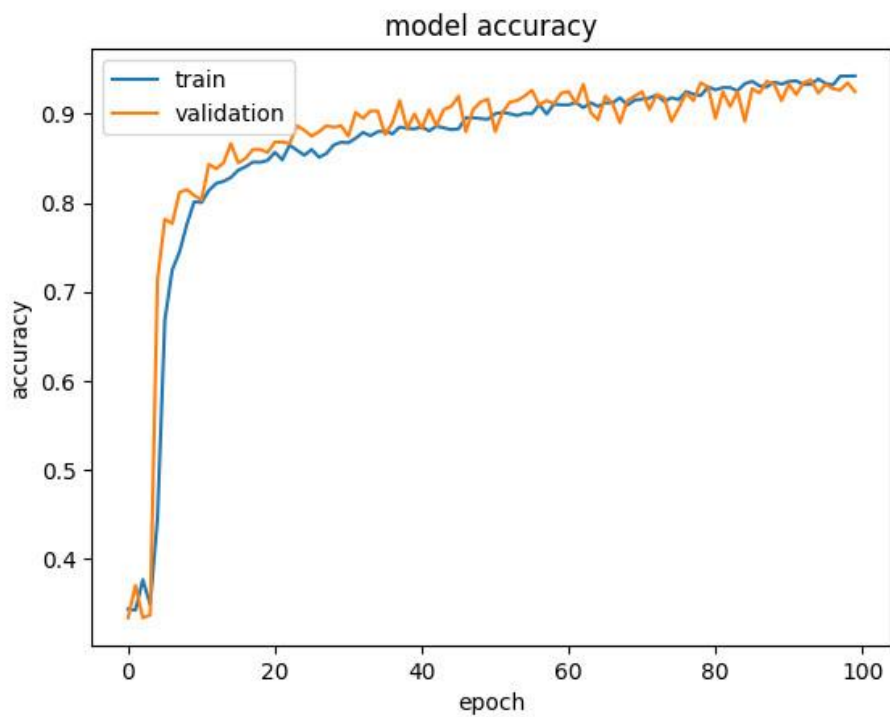


Hình 28. ResNet50V2 model accuracy

+ VGG16:



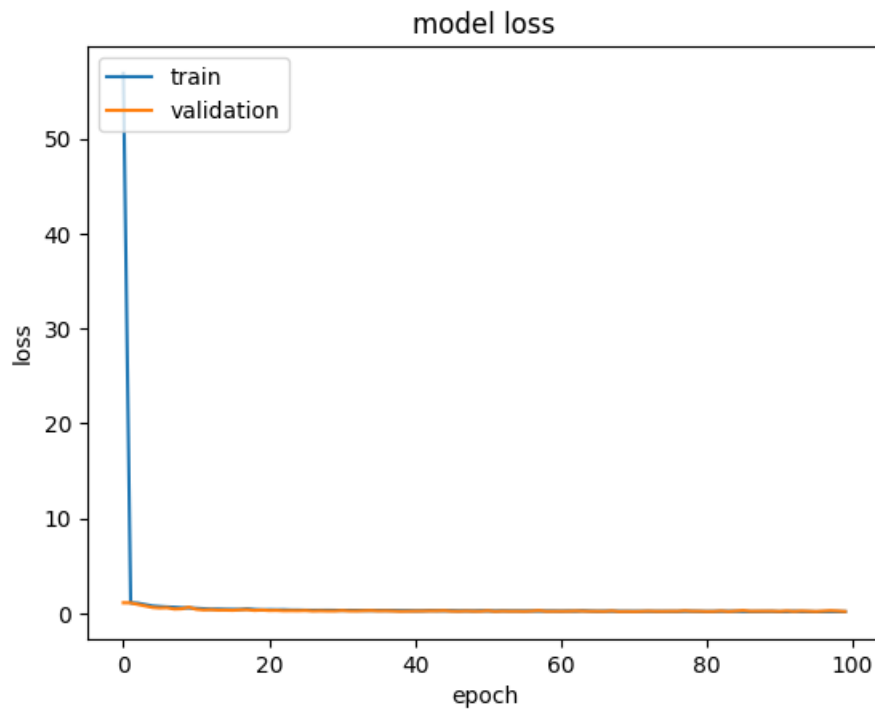
Hình 29. VGG16 model loss



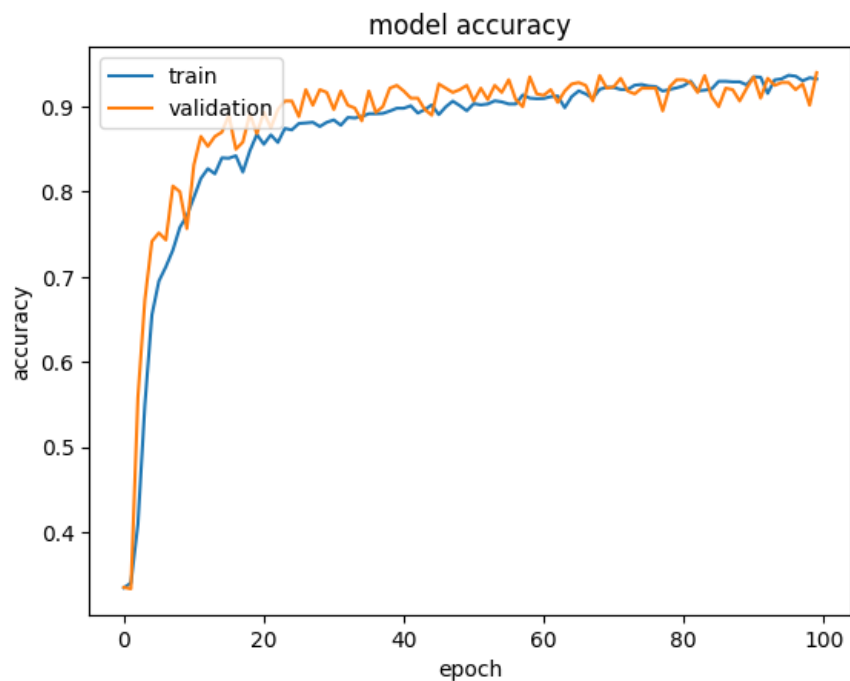
Hình 30. VGG16 model accuracy

---

+ VGG19:

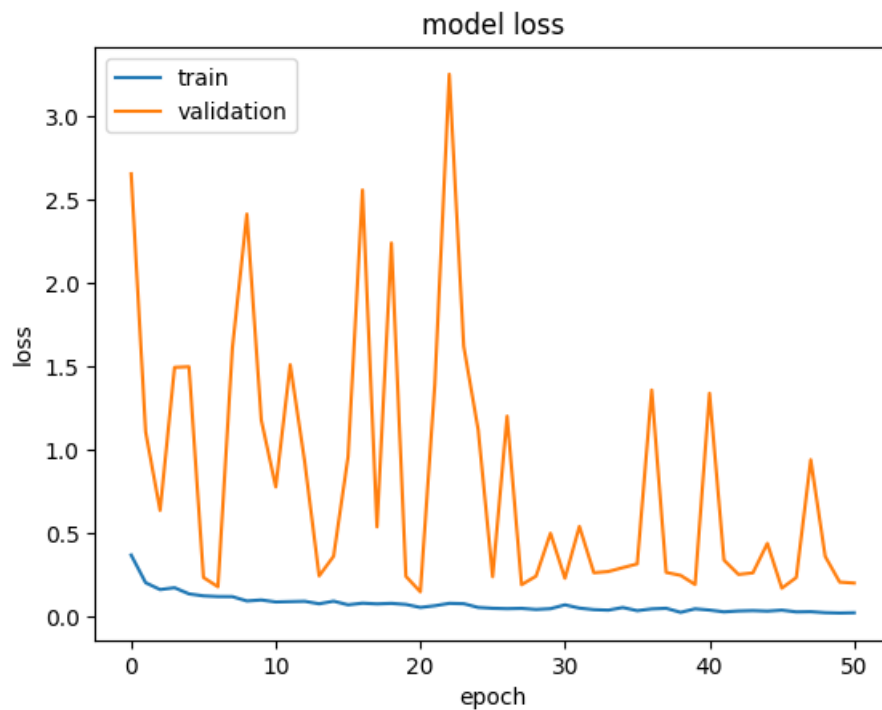


Hình 31. VGG19 model loss

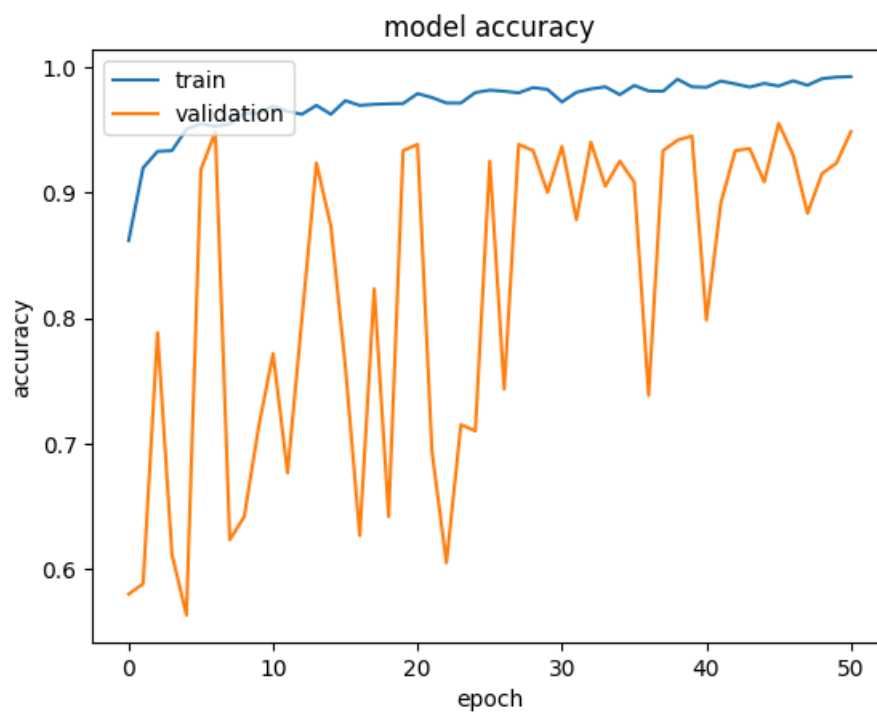


Hình 32. ResNet50V2 model accuracy

+ DenseNet121:



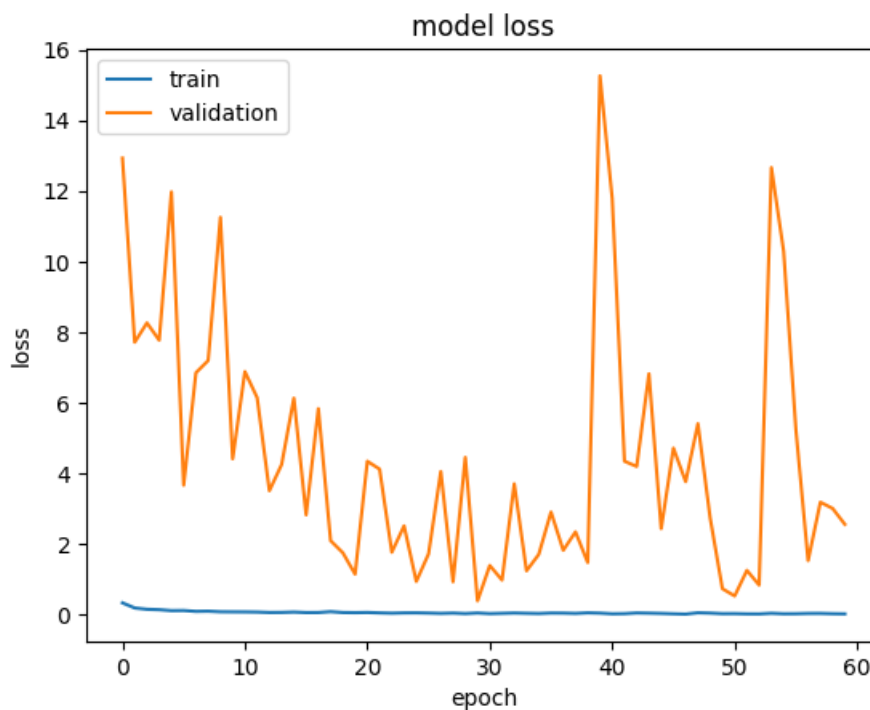
Hình 33. DenseNet121 model loss



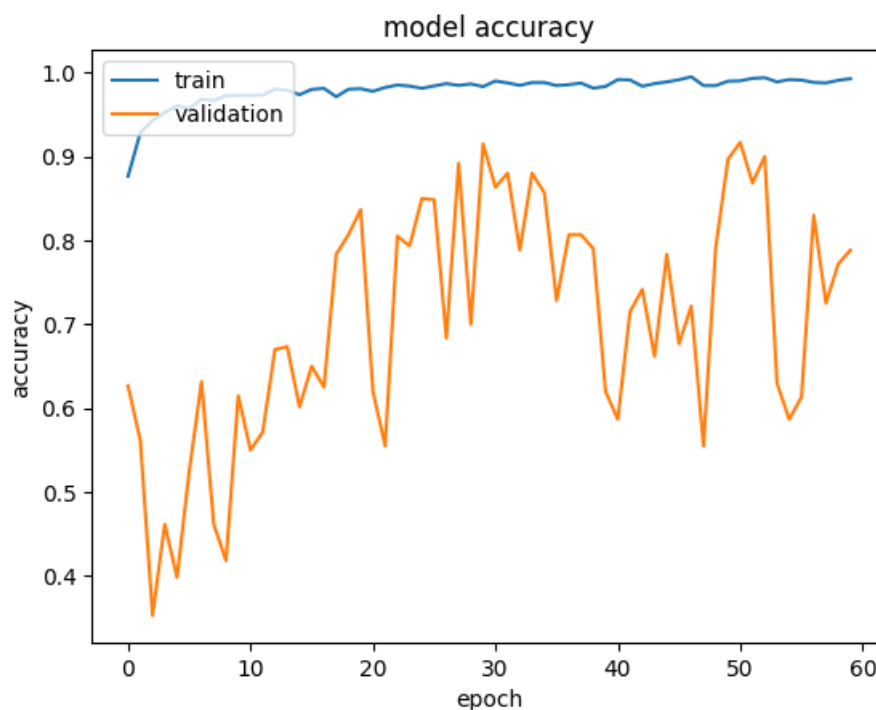
Hình 34. DenseNet121 model accuracy



+ MobileNetV2:



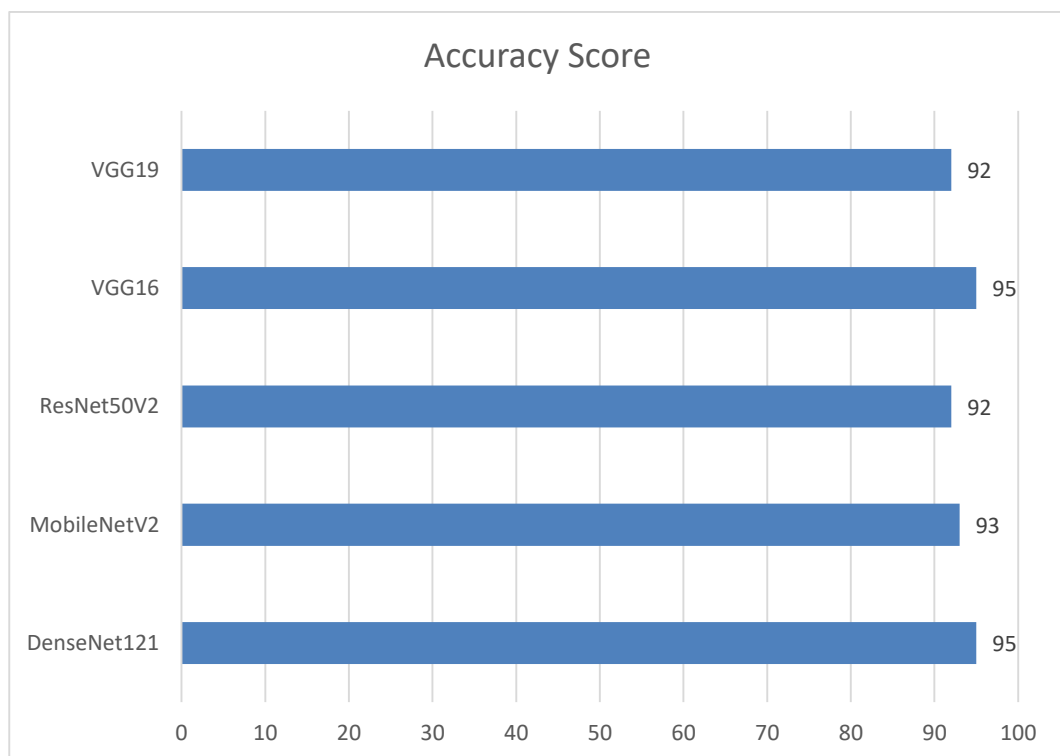
Hình 35. MobileNetV2 model loss



Hình 36. MobileNetV2 model accuracy

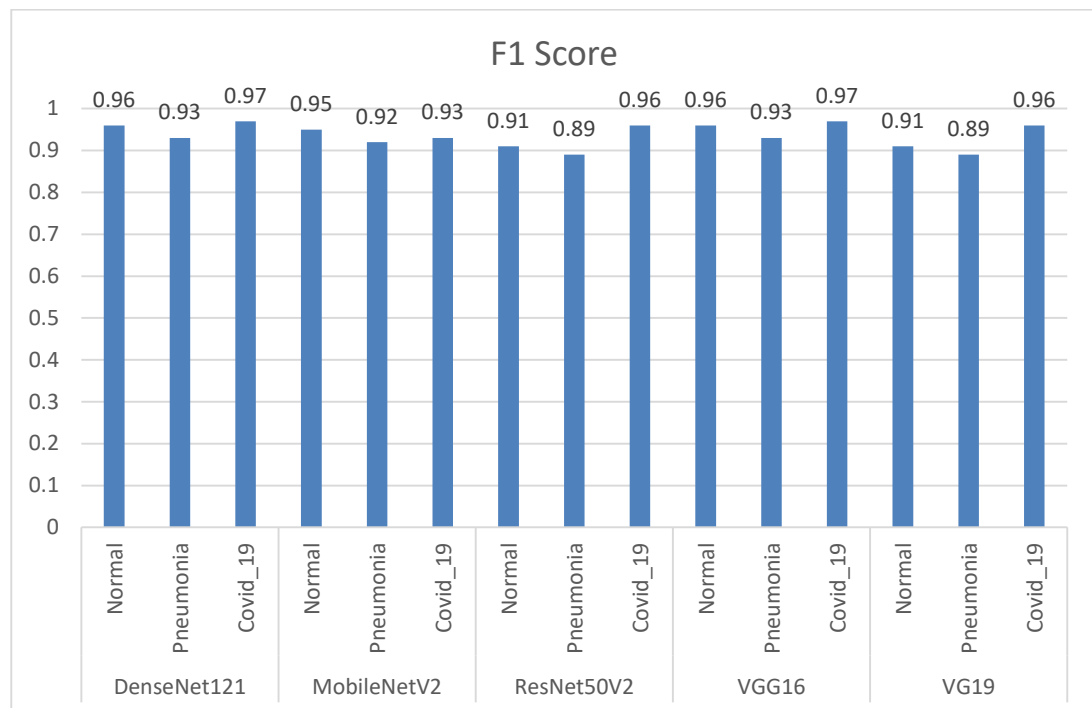
Dựa vào các biểu đồ về chỉ số loss và accuracy trong quá trình huấn luyện của các mô hình, chúng ta thấy được đối với tập dữ liệu đang sử dụng, sẽ thích hợp với các mô hình như: VGG16, VGG19 và ResNet50V2 do giá trị lỗi trên cả tập huấn luyện và đánh giá đều giảm đều theo thời gian và giá trị accuracy thì tăng đều theo thời gian. Ngược lại, đối với mô hình MobileNetV2 và DenseNet121, ta thấy rằng giá trị hàm lỗi trên tập huấn luyện thì giảm đều nhưng trên tập đánh giá lại tăng giảm thất thường và giá trị accuracy trên tập huấn luyện thì tăng nhưng trên tập đánh giá lại tăng giảm liên tục. Đây là một hiện tượng overfitting. Chúng ta sẽ có 2 hướng để khắc phục tình trạng này, một là tăng số lượng tập dữ liệu lên để tránh tình trạng trên, hai là giảm số lượng tham số của mô hình.

- Kết quả accuracy score trên 5 mô hình:



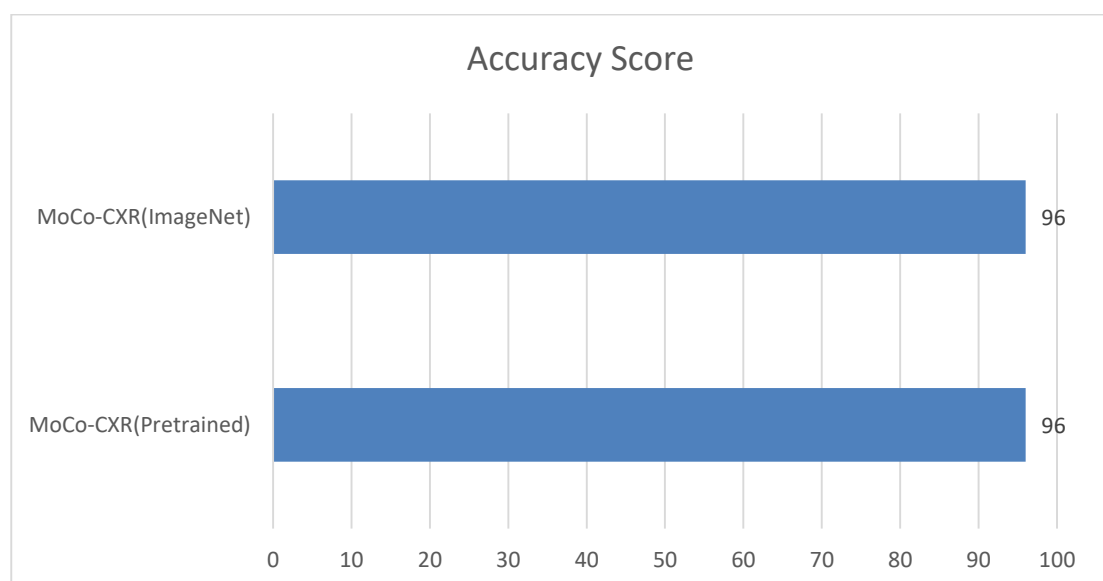
**Hình 37. Biểu đồ biểu thị giá trị Accuracy Score trên tập kiểm tra của 5 mô hình truyền thống**

Dựa vào biểu đồ bên trên ta thấy được rằng, hai mô hình VGG16 và DenseNet121 cho kết quả dự đoán tốt nhất lên đến 95%, kế đó là mô hình MobileNetV2 với độ chính xác là 93% và cuối cùng là hai mô hình VGG19 và ResNet50V2 chỉ đạt độ chính xác là 92%.

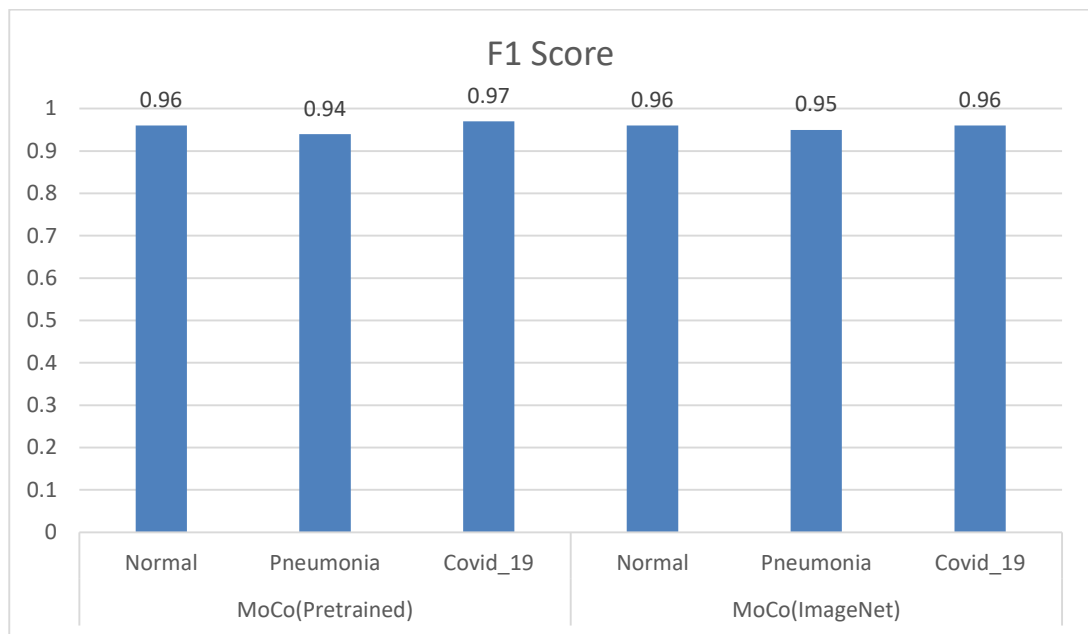


**Hình 38. Biểu đồ biểu thị giá trị F1-score trên tập kiểm tra của từng lớp của 5 mô hình truyền thống**

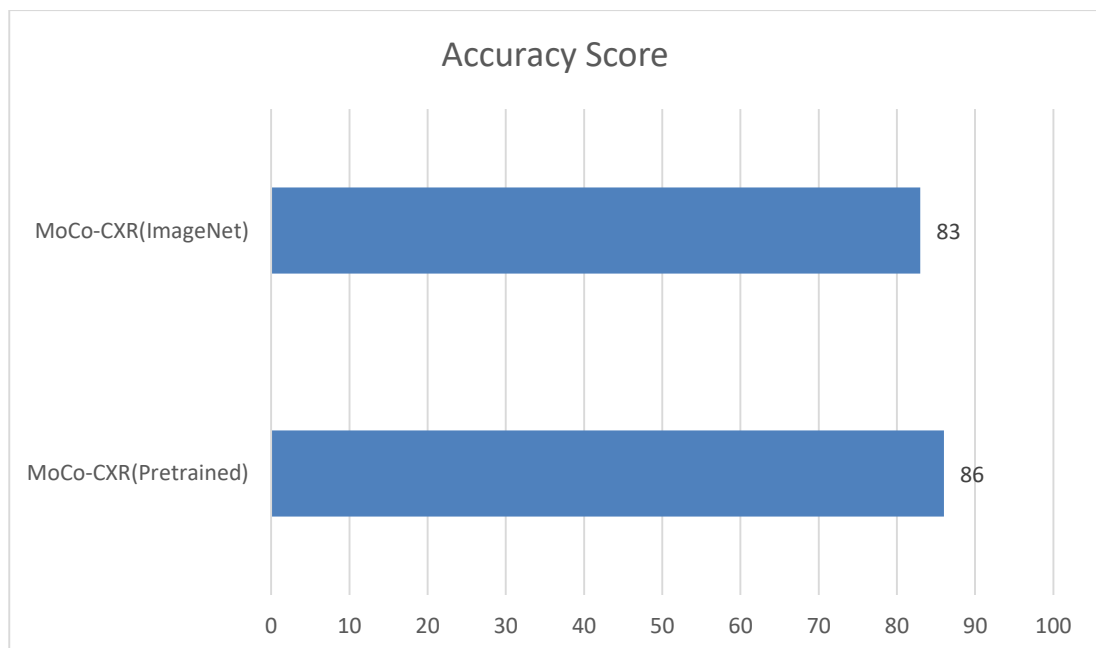
Biểu đồ bên trên cho ta thấy được rằng điểm F1-score trên tất cả các lớp của từng mô hình đều trên 0.89, do đó mô hình đạt được độ chính xác cao trong việc phân loại ảnh.



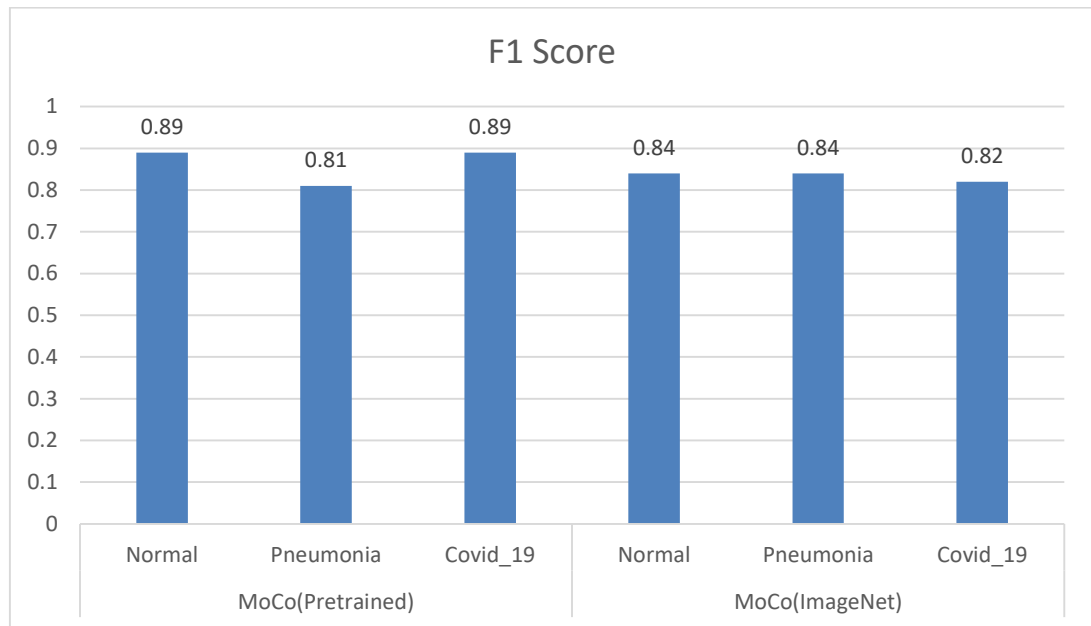
**Hình 39. Biểu đồ Accuracy Score của mô hình Moco sử dụng tập dữ liệu 4800 ảnh để huấn luyện**



**Hình 40. Biểu đồ F1 Score của mô hình Moco sử dụng tập dữ liệu 4800 ảnh để huấn luyện**



**Hình 41. Biểu đồ Accuracy Score của mô hình Moco sử dụng tập dữ liệu 300 ảnh để huấn luyện**



**Hình 42. Biểu đồ F1 Score của mô hình MoCo sử dụng tập dữ liệu 300 ảnh để huấn luyện**

Dựa vào các biểu đồ ở hình 38, 39, 40 41, cho chúng ta thấy được rằng, đối với mô hình MoCo sử dụng trọng số ImageNet với tập dữ liệu 300 ảnh mang lại accuracy thấp hơn so với mô hình sử dụng trọng số Pretrained được huấn luyện trên tập dữ liệu X-quang. Đối với mô hình sử dụng tập dữ liệu 6000 ảnh thì kết quả hai mô hình gần như bằng nhau, từ đó ta thấy được rằng đối với các bài toán có số lượng nhãn hạn chế thì mô hình sử dụng trọng số được huấn luyện trên các tập dữ liệu X-quang sẽ mang lại độ chính xác cao hơn so với mô hình sử dụng trọng số ImageNet.

---

## **PHẦN KẾT LUẬN**

### **1. Kết quả đạt được**

- Nghiên cứu và xây dựng được hệ thống dự đoán bệnh lý về phổi thông qua ảnh chụp X-quang. Thử nghiệm và đánh giá các mô hình cho thấy độ chính xác của mô hình khá cao lên đến 95%.
- Xây dựng được giao diện hệ thống giúp người dùng dễ dàng dự đoán được nhân của ảnh chụp X-quang mà bản thân mong muốn.
- Nghiên cứu và đánh giá kết quả thực nghiệm mô hình học tập tự giám sát (self-supervised learning) trên tập dữ liệu 6000 ảnh X-quang sử dụng trọng số từ mô hình đã .
- Sản phẩm demo có được sẽ làm tiền đề cho việc phát triển trong thời gian tới.

### **2. Hạn chế**

- Chưa thể áp dụng mô hình học tập tự giám sát vào hệ thống website.
- Chưa xây dựng trên nền tảng di động.
- Tốc độ dự đoán của hệ thống còn chưa được tốt.

### **3. Hướng phát triển**

- Nâng cao khả năng phân loại nhiều bệnh hơn.
- Nhận dạng vùng ảnh mang đặc trưng của bệnh lý.
- Cải thiện tốc độ dự đoán của mô hình.

---

## TÀI LIỆU THAM KHẢO

- [1] SOWRIRAJAN, Hari, et al. Moco pretraining improves representation and transferability of chest x-ray models. In: Medical Imaging with Deep Learning. PMLR, 2021. p. 728-744.
- [2] CHEN, Ting, et al. A simple framework for contrastive learning of visual representations. In: International conference on machine learning. PMLR, 2020. p. 1597-1607.
- [3] HE, Kaiming, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.
- [4] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [5] SANDLER, Mark, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 4510-4520.
- [6] HUANG, Gao, et al. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 4700-4708.
- [7] CHOWDHURY, Muhammad EH, et al. Can AI help in screening viral and COVID-19 pneumonia?. Ieee Access, 2020, 8: 132665-132676.
- [8] RAHMAN, Tawsifur, et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. Computers in biology and medicine, 2021, 132: 104319.
- [9] HE, Tong, et al. Bag of tricks for image classification with convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019. p. 558-567.
- [10] MISRA, Ishan; MAATEN, Laurens van der. Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. p. 6707-6717.

- 
- [11] IRVIN, Jeremy, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence. 2019. p. 590-597.
  - [12] IRVIN, Jeremy, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence. 2019. p. 590-597.
  - [13] RAJPURKAR, Pranav, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225, 2017.
  - [14] GUEFRECHI, Sarra, et al. Deep learning based detection of COVID-19 from chest X-ray images. Multimedia tools and applications, 2021, 80: 31803-31820.
  - [15] SZEGEDY, Christian, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 1-9.
  - [16] SZEGEDY, Christian, et al. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 2818-2826.
  - [17] TAN, Mingxing; LE, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR, 2019. p. 6105-6114.
  - [18] TAN, Mingxing; LE, Quoc. Efficientnetv2: Smaller models and faster training. In: International conference on machine learning. PMLR, 2021. p. 10096-10106.
  - [19] KERMANY, Daniel, et al. Labeled optical coherence tomography (oct) and chest x-ray images for classification. Mendeley data, 2018, 2.2: 651.



---

## ĐƯỜNG DẪN THAM KHẢO

1. Tập dữ liệu chứa các hình ảnh X-quang phổi có nhãn. Tuy nhiên, chúng ta chỉ lấy 2000 ảnh Covid-19 đầu tiên, 2000 ảnh Normal đầu tiên và 1345 ảnh Pneumonia.

<https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>

2. Đối với tập dữ liệu này, chúng ta chỉ lấy 655 ảnh Pneumonia đầu tiên trong tập huấn luyện.

<https://www.kaggle.com/datasets/jtiptj/chest-xray-pneumoniacovid19tuberculosis>

3. Trang chủ thư viện gradio, giúp bạn dễ dàng tạo ra giao diện website 1 cách đơn giản mà không cần kiến thức nền về giao diện.

<https://gradio.app/docs/>