

**TRƯỜNG ĐẠI HỌC VĂN LANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO NHẬP MÔN PHÂN TÍCH DỮ LIỆU VÀ HỌC SÂU**

NGÀNH: CÔNG NGHỆ THÔNG TIN

***Đề tài:* COVID-19 IN LUNGS**

**Nhóm sinh viên thực hiện (Họ tên - Mã SV):**

**Nguyễn Trọng Nghĩa – 207CT40473**

**Đỗ Quang Dũng – 207CT27663**

**Lê Đặng Kim Lân – 207CT47823**

**Nguyễn Lê Khải Tân – 207CT28480**

**Lục Lê Anh Dũng – 207CT40181**

**Giảng viên hướng dẫn:** **Nguyễn Thái Anh**

**TP. Hồ Chí Minh – năm 2024**



**MỤC LỤC**

[CHƯƠNG 1: MỞ ĐẦU 3](#_Toc172542222)

[**1.** **Lý do chọn chủ đề:** 3](#_Toc172542223)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5](#_Toc172542224)

[**1.** **Khái niệm về CNN** 5](#_Toc172542225)

[**2.** **Các Thành phần Chính của CNN** 5](#_Toc172542226)

[**3.** **Hoạt động của CNN** 6](#_Toc172542227)

[**4.** **Ứng dụng của CNN** 6](#_Toc172542228)

[**5.** **Thư viện kết hợp Keras** 6](#_Toc172542229)

[**6.** **Đánh giá mô hình** 6](#_Toc172542230)

[CHƯƠNG 3: TRIỂN KHAI MÔ HÌNH 8](#_Toc172542231)

[**1.** **Các bước triển khai** 8](#_Toc172542232)

[**2.** **Phân tích đồ thị** 12](#_Toc172542233)

[CHƯƠNG 4: KẾT LUẬN 20](#_Toc172542234)

[CHƯƠNG 5: ĐÁNH GIÁ KHẢ NĂNG LÀM VIỆC NHÓM CỦA MỖI THÀNH VIÊN 20](#_Toc172542235)

[CHƯƠNG 6: LỜI CẢM ƠN 20](#_Toc172542236)

# 

# **CHƯƠNG 1: MỞ ĐẦU**

1. **Lý do chọn chủ đề:**

Dịch COVID-19, do virus corona gây ra, bắt đầu vào tháng 12 năm 2019 tại Vũ Hán, Trung Quốc, và nhanh chóng lan rộng toàn cầu. Bệnh này gây ra các triệu chứng hô hấp nghiêm trọng và liên quan đến tỷ lệ nhập viện ICU và tử vong cao. Phương pháp lấy mẫu từ miệng và mũi có thể ảnh hưởng đến kết quả xét nghiệm. Nếu không được thực hiện đúng cách, các xét nghiệm có thể cho ra kết quả không chính xác. Hơn nữa, có báo cáo cho rằng độ nhạy của các bộ xét nghiệm không đủ cao để phát hiện sớm và điều trị cho bệnh nhân nghi ngờ. Vì vậy, hình ảnh chụp cắt lớp vi tính (CT) phổi đóng vai trò quan trọng trong việc phát hiện sớm COVID-19.

Đại dịch COVID-19 đã ảnh hưởng nghiêm trọng đến sức khỏe toàn cầu và gây ra khủng hoảng y tế. Việc phát hiện và chẩn đoán nhanh chóng COVID-19 là cực kỳ cần thiết để kiểm soát sự lây lan của virus. Sử dụng các công cụ học máy và mạng nơ-ron tích chập (CNN) để phân tích hình ảnh X-quang ngực có thể giúp phát hiện sớm các dấu hiệu nhiễm COVID-19 trong phổi, ngay cả khi các triệu chứng chưa rõ ràng hoặc trong giai đoạn đầu của bệnh. Mô hình AI này có thể hỗ trợ đắc lực cho các bác sĩ và chuyên gia y tế trong việc chẩn đoán và điều trị bệnh nhân, đặc biệt quan trọng trong bối cảnh thiếu hụt nhân lực y tế và số lượng lớn bệnh nhân cần được kiểm tra.

Sử dụng mô hình CNN để phân tích hình ảnh X-quang có thể tăng tốc độ chẩn đoán, giảm tải công việc cho các chuyên gia y tế, và cho phép xử lý số lượng lớn bệnh nhân một cách hiệu quả hơn. Chủ đề này mở ra cơ hội để nghiên cứu và phát triển các phương pháp chẩn đoán và điều trị mới, góp phần vào sự tiến bộ của y học và công nghệ y tế. Hình ảnh X-quang là một phương pháp chẩn đoán phổ biến và có sẵn ở nhiều nơi trên thế giới. Việc phát triển một mô hình AI để phân tích các hình ảnh này là khả thi và có thể triển khai rộng rãi.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **Khái niệm về CNN**

Convolutional Neural Network (CNN) hay Mạng Nơ-ron Tích chập là một loại mạng nơ-ron sâu được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc dạng lưới, chẳng hạn như hình ảnh. CNN đã trở thành một công cụ quan trọng trong lĩnh vực thị giác máy tính và nhiều ứng dụng khác. Dưới đây là các thành phần và khái niệm cơ bản về CNN:

## **Các Thành phần Chính của CNN**

1. **Convolutional Layer (Lớp tích chập):**
   * Đây là lớp chính của CNN, nơi các bộ lọc (Filter) được áp dụng lên dữ liệu đầu vào (Thường là hình ảnh) để trích xuất các đặc trưng cục bộ.
   * Bộ lọc là một ma trận nhỏ, thường là 3x3 hoặc 5x5, trượt qua ảnh đầu vào và thực hiện phép nhân chập (Convolution) giữa bộ lọc và vùng tương ứng của ảnh.
   * Kết quả của phép nhân chập này là một bản đồ đặc trưng (Feature map) chứa thông tin về các đặc trưng phát hiện được từ ảnh.
2. **Activation Function (Hàm kích hoạt):**
   * Sau mỗi lớp tích chập, một hàm kích hoạt, thường là ReLU (Rectified Linear Unit), được áp dụng. ReLU chuyển đổi tất cả các giá trị âm trong bản đồ đặc trưng thành 0 và giữ nguyên các giá trị dương. Điều này giúp mô hình học được các đặc trưng phi tuyến.
3. **Pooling Layer (Lớp gộp):**
   * Lớp gộp được sử dụng để giảm kích thước không gian của bản đồ đặc trưng, giúp giảm số lượng tham số và tính toán trong mạng.
   * Max pooling là kỹ thuật phổ biến nhất, lấy giá trị lớn nhất trong một vùng nhỏ (Thường là 2x2) của bản đồ đặc trưng.
4. **Fully Connected Layer (Lớp kết nối đầy đủ):**
   * Ở phần cuối của CNN, các lớp kết nối đầy đủ được sử dụng. Mỗi neuron trong lớp này kết nối với tất cả các neuron ở lớp trước.
   * Lớp này tổng hợp các đặc trưng đã được trích xuất và thực hiện phân loại hoặc dự đoán.
5. **Softmax Layer:**
   * Thường được sử dụng ở lớp cuối cùng cho các bài toán phân loại. Softmax chuyển đổi đầu ra của mạng thành xác suất cho các lớp khác nhau.

## **Hoạt động của CNN**

CNN hoạt động bằng cách xếp chồng các lớp tích chập, lớp kích hoạt và lớp gộp để trích xuất các đặc trưng từ hình ảnh đầu vào. Các đặc trưng này sau đó được đưa qua các lớp kết nối đầy đủ để thực hiện phân loại hoặc dự đoán. Quá trình huấn luyện CNN bao gồm việc tối ưu hóa các bộ lọc và trọng số trong các lớp kết nối đầy đủ để giảm thiểu hàm mất mát (Loss function), chẳng hạn như cross-entropy cho bài toán phân loại.

## **Ứng dụng của CNN**

CNN được sử dụng rộng rãi trong nhiều ứng dụng như:

* **Nhận diện và phân loại hình ảnh**: Nhận diện đối tượng, phân loại ảnh theo các danh mục.
* **Xử lý video**: Nhận diện và theo dõi đối tượng trong video.
* **Y tế**: Chẩn đoán bệnh từ hình ảnh y tế như X-quang, MRI.
* **Xử lý ngôn ngữ tự nhiên (NLP)**: Phân loại văn bản, nhận diện thực thể.
* **Thị giác máy tính trong xe tự lái**: Nhận diện biển báo giao thông, người đi bộ.

1. **Thư viện kết hợp Keras**

Keras là gì hay Keras python là gì là câu hỏi được nhiều người quan tâm. Keras là một thư viện mã nguồn mở được sử dụng rộng rãi trong lĩnh vực deep learning (học sâu) và mạng nơ-ron. Nó được thiết kế để giúp các nhà phát triển xây dựng và thử nghiệm các mô hình học sâu một cách dễ dàng và linh hoạt. Mục tiêu chính của Keras là cung cấp một API đơn giản, trực quan và dễ sử dụng, giúp người dùng tập trung vào việc xây dựng mô hình mà không cần lo lắng về chi tiết kỹ thuật phức tạp.Một trong những điểm mạnh của Keras là khả năng tích hợp với các framework học sâu khác như TensorFlow và Theano, cho phép người dùng tận dụng sức mạnh của cả hai thế giới. Đây chính là điều giúp Keras trở thành một lựa chọn hàng đầu cho các dự án học sâu từ phân loại ảnh, dự đoán chuỗi thời gian, đến các ứng dụng trong lĩnh vực ngôn ngữ tự nhiên và nhiều lĩnh vực khác.

1. **Đánh giá mô hình**

Đánh giá mô hình CNN (Convolutional Neural Network) trong đồ án COVID-19 in lungs có thể bao gồm các tiêu chí sau:

1. **Hiệu suất (Performance):**
   * **Độ chính xác (Accuracy):** Tỷ lệ dự đoán đúng trên tổng số dự đoán.
   * **Độ nhạy (Sensitivity) hoặc Recall:** Tỷ lệ bệnh nhân COVID-19 được xác định đúng trên tổng số bệnh nhân thực sự mắc bệnh.
   * **Độ đặc hiệu (Specificity):** Tỷ lệ người không mắc bệnh được xác định đúng trên tổng số người không mắc bệnh.
   * **F1-Score:** Trung bình hài hòa của Precision và Recall.
2. **Thời gian và tài nguyên tính toán (Computational Time and Resources):**
   * Thời gian huấn luyện mô hình.
   * Số lượng epochs và batch size.
   * Sử dụng tài nguyên (CPU, GPU).
3. **Phân tích hình ảnh (Image Analysis):**
   * Khả năng mô hình phát hiện các đặc điểm đặc trưng của COVID-19 trong hình ảnh phổi.
   * Sử dụng các kỹ thuật visualization như Grad-CAM, LIME để xem mô hình tập trung vào những vùng nào của ảnh.
4. **Đánh giá qua Cross-validation:**
   * Sử dụng kỹ thuật K-fold cross-validation để đảm bảo mô hình không bị overfitting và đánh giá độ ổn định của mô hình.
5. **So sánh với các mô hình khác:**
   * So sánh hiệu suất của mô hình CNN với các mô hình khác (SVM, Random Forest, các mô hình deep learning khác) để xem mô hình CNN có thực sự tốt hơn hay không.
6. **Đánh giá tổng quan:**
   * Mức độ ứng dụng thực tế của mô hình: Mô hình có thể áp dụng vào việc chẩn đoán COVID-19 trong môi trường thực tế hay không.
   * Tính khả thi và độ tin cậy của mô hình trong điều kiện thực tế.

# **CHƯƠNG 3: TRIỂN KHAI MÔ HÌNH**

1. **Các bước triển khai**

**Bước 1: Tạo các đối tượng ImageDataGenerator**

1. **ImageDataGenerator cho tập huấn luyện:**

* **“rescale=1./255”:** Chia tất cả các giá trị pixel của hình ảnh cho 255 để chuẩn hóa các giá trị pixel từ 0-255 về khoảng 0-1.
* **“shear\_range=0.2”:** Áp dụng phép biến đổi xiên lên hình ảnh trong khoảng 0.2.
* **“zoom\_range=0.2”:** Áp dụng phép biến đổi phóng to lên hình ảnh trong khoảng 0.2.
* **“horizontal\_flip=True”:** Cho phép lật ngang các hình ảnh một cách ngẫu nhiên.

Các phép biến đổi này giúp tăng cường dữ liệu (data augmentation), làm cho mô hình huấn luyện được trên nhiều biến thể của dữ liệu gốc, từ đó tăng tính tổng quát của mô hình.

1. **ImageDataGenerator cho tập kiểm tra:**

* **“rescale=1./255”:** Chỉ chuẩn hóa các giá trị pixel từ 0-255 về khoảng 0-1. Không áp dụng các phép biến đổi khác vì tập kiểm tra cần giữ nguyên hình ảnh gốc để đánh giá chính xác hiệu suất của mô hình.

A screenshot of a computer program

Description automatically generated

**Bước 2: Chuẩn bị dữ liệu huấn luyện và kiểm tra**

1. **training\_set để huấn luyện:**

* **“train\_datagen”:** Một instance của ImageDataGenerator cho dữ liệu huấn luyện.
* **“flow\_from\_directory('dataset2/two/train')”:** Chỉ định thư mục chứa hình ảnh huấn luyện.
* **“target\_size=(64, 64)”:** Tất cả hình ảnh sẽ được thay đổi kích thước thành 64x64 pixel.
* **“batch\_size=32”:** Số lượng hình ảnh sẽ được sinh ra từ bộ sinh dữ liệu mỗi đợt.
* **“class\_mode='binary'”:** Cho biết đây là bài toán phân loại nhị phân (ví dụ: hai lớp: mèo và chó).

1. **test\_set để kiểm tra**

* **“test\_datagen”:** Một instance của ImageDataGenerator cho dữ liệu kiểm tra.
* **“flow\_from\_directory('dataset2/two/test')”:** Chỉ định thư mục chứa hình ảnh kiểm tra.
* Các tham số khác (Target\_size, batch\_size, và class\_mode) giống với tập huấn luyện, đảm bảo tính nhất quán trong tiền xử lý dữ liệu.

A computer screen shot of a program

Description automatically generated

**Bước 3: Huấn luyện mô hình**

1. **Tham số**

* **“classifier.fit\_generator”:** Phương thức này huấn luyện mô hình sử dụng các bộ sinh dữ liệu.
* **“training\_set”:** Bộ sinh dữ liệu cho tập huấn luyện.
* **“samples\_per\_epoch=130”:** Số lượng mẫu (Hình ảnh) được sử dụng trong mỗi kỷ nguyên (Epoch).
* **“nb\_epoch=100”:** Số lượng kỷ nguyên để huấn luyện mô hình.
* **“validation\_data=test\_set”:** Bộ sinh dữ liệu cho tập kiểm tra.
* **“nb\_val\_samples=18”:** Số lượng mẫu (Hình ảnh) kiểm tra được sử dụng để đánh giá mô hình sau mỗi kỷ nguyên.

A screen shot of a computer

Description automatically generated

Đoạn mã dưới đây được sử dụng để trực quan hóa hiệu suất của mô hình CNN trong quá trình huấn luyện và kiểm tra. Cụ thể, nó vẽ các đồ thị cho độ chính xác (Accuracy) và hàm mất mát (Loss) theo số lượng epochs.

**Bước 4: Thiết lập để tạo biểu đồ**

1. **Thêm thư viện matplotlib.pyplot được sử dụng để tạo ra các biểu đồ trong Python.**
2. **Hàm plot\_acc\_loss nhận hai tham số:**

* **“results”:** Đối tượng lưu trữ dữ liệu lịch sử của mô hình.
* **“epochs”:** Số lượng epochs huấn luyện mô hình.

1. **Bên trong hàm, các bước sau được thực hiện:**

* Lấy ra các giá trị độ chính xác và mất mát trên tập dữ liệu huấn luyện từ “results.history”.
* Lấy ra các giá trị độ chính xác và mất mát trên tập dữ liệu kiểm tra từ “results.history”.

A screen shot of a computer code

Description automatically generated

**Bước 5: Tạo đồ thị**

* **Tạo hình ảnh:** Sử dụng hàm plt.figure để tạo ra một hình ảnh mới với kích thước được chỉ định (15 inch x 5 inch).
* **Tạo các trục phụ:** Sử dụng hàm “plt.subplot” để tạo ra hai trục phụ trong hình ảnh, mỗi trục phụ chiếm một nửa diện tích theo chiều ngang.
* Sử dụng hàm “plt.plot” để vẽ đường biểu đồ.
* Sử dụng hàm “plt.title” để đặt tiêu đề cho biểu đồ.
* Sử dụng hàm “plt.legend” để hiển thị chú thích cho các đường biểu đồ.
* Sử dụng hàm “plt.show” để hiển thị biểu đồ.
* Sử dụng hàm “plot\_acc\_loss(Results, 100)” để vẽ biểu đồ (Plot) cho độ chính xác (Accuracy) và mất mát (Loss) của mô hình trong suốt quá trình huấn luyện.

A computer screen shot of text

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

1. **Phân tích đồ thị**

**Đồ Thị Độ Chính Xác (Accuracy over 100 Epochs)**

1. **Độ chính xác trên tập huấn luyện (Train\_acc) và tập kiểm tra (Test\_acc)**:
   * **Độ chính xác trên tập huấn luyện**: Dao động mạnh qua các epochs nhưng nhìn chung đạt rất cao, dao động xung quanh 0.94 đến 1.0.
   * **Độ chính xác trên tập kiểm tra**: Dường như ổn định ở mức rất cao, gần 1.0, cho thấy mô hình hoạt động rất tốt trên tập kiểm tra.

**Đồ Thị Hàm Mất Mát (Loss over 100 Epochs)**

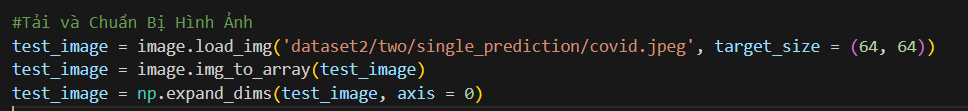
1. **Hàm mất mát trên tập huấn luyện (Train\_loss) và tập kiểm tra (Test\_loss)**:
   * **Hàm mất mát trên tập huấn luyện**: Ban đầu khá cao nhưng giảm nhanh chóng và dao động ở mức thấp sau đó. Tuy nhiên, có nhiều điểm dao động mạnh qua các epochs.
   * **Hàm mất mát trên tập kiểm tra**: Ổn định ở mức rất thấp, thấp hơn so với tập huấn luyện sau một số epochs ban đầu.

**Nhận Xét**

1. **Hiện Tượng Overfitting**:
   * Mặc dù độ chính xác trên tập kiểm tra rất cao và hàm mất mát rất thấp, nhưng độ chính xác trên tập huấn luyện dao động mạnh và có nhiều điểm không ổn định.
   * Điều này có thể là dấu hiệu của overfitting, khi mô hình quá khớp với dữ liệu huấn luyện nhưng hoạt động rất tốt trên dữ liệu kiểm tra.
2. **Mô hình Quá Khớp Dữ Liệu**:
   * Độ chính xác và hàm mất mát trên tập kiểm tra đều rất cao và thấp, tương ứng, có thể do dữ liệu kiểm tra và dữ liệu huấn luyện có nhiều đặc điểm tương tự hoặc mô hình học được quá kỹ các đặc điểm của dữ liệu huấn luyện.
3. **Đánh Giá Hiệu Suất**:
   * Hiệu suất cao trên tập kiểm tra cho thấy mô hình có thể dự đoán chính xác trên dữ liệu chưa từng thấy, nhưng cũng cần kiểm tra thêm trên các tập dữ liệu kiểm tra khác để đảm bảo tính tổng quát của mô hình.

Đoạn mã dưới dùng để thực hiện dự đoán trên một hình ảnh đơn lẻ bằng mô hình CNN đã được huấn luyện.

* **“image.load\_img”:** Hàm này từ thư viện Keras dùng để tải hình ảnh từ đường dẫn cụ thể.
* **“dataset2/two/single\_prediction/covid.jpeg”:** Đường dẫn đến hình ảnh cần dự đoán.
* **“target\_size=(64, 64)”:** Thay đổi kích thước của hình ảnh về 64x64 pixel.
* **“image.img\_to\_array”:** Hàm này chuyển đổi hình ảnh từ định dạng PIL thành một mảng numpy (array).
* **“np.expand\_dims”:** Hàm này từ thư viện NumPy dùng để mở rộng chiều của mảng.
* **“axis=0”:** Thêm một chiều mới ở vị trí 0 (đầu tiên), điều này làm cho mảng từ dạng (64, 64, 3) thành (1, 64, 64, 3), phù hợp với đầu vào của mô hình học sâu Keras (batch size, height, width, channels).



Đoạn mã dưới dùng đê dự đoán và in ra các chỉ số

* **“classifier.predict(test\_image)”:** Hàm này sử dụng mô hình classifier đã huấn luyện để dự đoán kết quả cho hình ảnh test\_image.
* **“test\_image”:** Hình ảnh đã được chuẩn bị và tiền xử lý ở các bước trước.
* **“result”:** Biến này sẽ chứa kết quả dự đoán của mô hình. Kết quả này thường là một mảng xác suất mà mô hình gán cho mỗi lớp.
* **“training\_set.class\_indices”:** Thuộc tính này của training\_set là một từ điển (dictionary) ánh xạ tên lớp thành chỉ số tương ứng mà mô hình đã học.

A screen shot of a computer code

Description automatically generated

Đoạn mã dưới thực hiện dự đoán và in ra kết quả dự đoán của mô hình.

* **“if result[0][0] == 1”:** Kiểm tra giá trị dự đoán đầu tiên (result[0][0]). Nếu giá trị này bằng 1, thì kết quả dự đoán là 'normal'.
* **“prediction = 'normal'”:** Nếu điều kiện trên đúng, gán giá trị 'normal' cho biến prediction.
* **“else”:** Nếu điều kiện trên sai, tức là result[0][0] không bằng 1, thì gán giá trị 'covid' cho biến prediction.
* **“prediction = 'covid'”:** Gán giá trị 'covid' cho biến prediction.
* **“print("AI dự đoán là: " + prediction)”:** In ra kết quả dự đoán, bao gồm chuỗi "AI dự đoán là: " và giá trị của biến prediction.

A computer screen with colorful text

Description automatically generated

Đoạn mã này sử dụng thư viện matplotlib để hiển thị hình ảnh X-quang của một người dương tính với COVID-19.

* **“import matplotlib.pyplot as plt”:** Nhập thư viện matplotlib.pyplot và đặt tên tắt là plt. Thư viện này cung cấp các hàm để vẽ biểu đồ và hình ảnh.
* **“import matplotlib.image as mpimg”:** Nhập thư viện matplotlib.image và đặt tên tắt là mpimg. Thư viện này cung cấp các hàm để đọc và hiển thị hình ảnh.
* **“mpimg.imread('dataset2/two/single\_prediction/covid2.jpeg')”:** Hàm imread đọc hình ảnh từ đường dẫn "dataset2/two/single\_prediction/covid2.jpeg" và lưu nó vào biến img.
* **“plt.imshow(img)”:** Hàm imshow hiển thị hình ảnh được lưu trong biến img. Kết quả được lưu vào biến imgplot.
* **“plt.title('Hình chụp X-quang của người dương tính')”:** Hàm title thêm tiêu đề "Hình chụp X-quang của người dương tính" vào hình ảnh.

A screen shot of a computer program

Description automatically generated

X-ray of a person's chest

Description automatically generated

* **“result[0][0]”:** Biến này đại diện cho kết quả dự đoán của mô hình AI.
* **“if result[0][0] == 1”:** Kiểm tra xem kết quả dự đoán có bằng 1 hay không. Nếu bằng 1, có nghĩa là mô hình dự đoán tình trạng là "normal".
* **“prediction = 'normal'”:** Nếu kết quả dự đoán bằng 1, biến prediction được gán giá trị "normal".
* **“Else”:** Câu lệnh này được thực thi nếu kết quả dự đoán không bằng 1.
* **“prediction = 'covid'”:** Nếu kết quả dự đoán không bằng 1, biến prediction được gán giá trị "covid".
* **“print("AI dự đoán là: "+ prediction)”:** Câu lệnh này in ra kết luận của AI.

A computer screen shot of a program

Description automatically generated

Đoạn mã này sử dụng thư viện matplotlib để hiển thị hình ảnh X-quang của một người âm tính với COVID-19.

* **“import matplotlib.pyplot as plt”:** Dòng này import thư viện matplotlib.pyplot và đặt biệt danh là plt. Thư viện này được sử dụng để tạo đồ thị và hiển thị hình ảnh.
* **“import matplotlib.image as mpimg”:** Dòng này import thư viện matplotlib.image và đặt biệt danh là mpimg. Thư viện này được sử dụng để tải và xử lý hình ảnh.
* **“test\_image = image.load\_img('dataset2/two/single\_prediction/normal.jpeg', target\_size = (64, 64))”**: Dòng này tải hình ảnh từ đường dẫn 'dataset2/two/single\_prediction/normal.jpeg' và thay đổi kích thước của hình ảnh thành 64x64 pixel. Hình ảnh này được gán vào biến test\_image.
* **“test\_image = image.img\_to\_array(test\_image)”:** Dòng này chuyển đổi hình ảnh từ dạng PIL Image thành mảng NumPy.
* **“test\_image = np.expand\_dims(test\_image, axis=0)”:** Dòng này thêm một chiều vào mảng NumPy đại diện cho batch size của hình ảnh.
* **“result = classifier.predict(test\_image)”:** Dòng này sử dụng mô hình học máy classifier để dự đoán dựa trên hình ảnh test\_image. Kết quả của dự đoán được gán vào biến result.
* **“img=mpimg.imread('dataset2/two/single\_prediction/normal.jpeg')”:** Dòng này tải hình ảnh từ đường dẫn 'dataset2/two/single\_prediction/normal.jpeg' và gán vào biến img.
* **“imgplot = plt.imshow(img)”:** Dòng này hiển thị hình ảnh img trên màn hình.
* **“plt=plt.title('Hình chụp X-quang của người âm tính ')”:** Dòng này đặt tiêu đề cho hình ảnh là 'Hình chụp X-quang của người âm tính'.

A computer screen shot of a code

Description automatically generated

A x-ray of a person's chest

Description automatically generated

A screen shot of a computer program

Description automatically generated

# **CHƯƠNG 4: KẾT LUẬN**

Đồ án này đã nghiên cứu và áp dụng các mô hình mạng nơ-ron tích chập (CNN) để phân tích hình ảnh CT phổi, nhằm hỗ trợ quá trình chẩn đoán COVID-19. Kết quả cho thấy, việc sử dụng các mô hình học sâu có thể cải thiện đáng kể độ chính xác trong việc phát hiện các dấu hiệu nhiễm COVID-19 so với các phương pháp truyền thống.

Tóm lại, việc kết hợp công nghệ AI và hình ảnh y học đã mở ra những triển vọng mới trong việc chẩn đoán và điều trị COVID-19. Đồ án này không chỉ khẳng định vai trò quan trọng của hình ảnh CT trong chẩn đoán COVID-19 mà còn đề xuất những hướng đi mới cho việc ứng dụng học máy trong y học, góp phần vào cuộc chiến chống lại đại dịch toàn cầu này.

# **CHƯƠNG 5: ĐÁNH GIÁ KHẢ NĂNG LÀM VIỆC NHÓM CỦA MỖI THÀNH VIÊN**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Student ID** | **Student Name** | **Average Grade by Team** |
| **1** | **207CT40473** | **Nguyễn Trọng Nghĩa** | **100%** |
| **2** | **207CT27663** | **Đỗ Quang Dũng** | **100%** |
| **3** | **207CT47823** | **Lê Đặng Kim Lân** | **100%** |
| **4** | **207CT28480** | **Nguyễn Lê Khải Tân** | **100%** |
| **5** | **207CT40181** | **Lục Lê Anh Dũng** | **100%** |

# **CHƯƠNG 6: LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Thầy vì những lời hướng dẫn, góp ý quý báu cho bài báo cáo của nhóm em vừa qua. Nhờ sự tận tâm và nhiệt tình của Thầy, nhóm em đã hoàn thành bài báo cáo một cách tốt đẹp.

Bài báo cáo này không thể hoàn thành tốt đẹp nếu không có sự hỗ trợ và hướng dẫn nhiệt tình của Thầy. Chúng em xin hứa sẽ tiếp thu những góp ý của Thầy để hoàn thiện bản thân và đạt được nhiều kết quả tốt hơn trong học tập và nghiên cứu.

Cuối cùng, em xin kính chúc Thầy luôn dồi dào sức khỏe, hạnh phúc và thành công trong sự nghiệp giảng dạy.

Link: <https://github.com/trongnghia2510/Covid-19-Team-03>