**Ngày bắt đầu:** 20/5/2024

**Người thực hiện:** Võ Hoài Trọng (Github: HiTrong)

Xây dựng mô hình nhận dạng ung thư phổi thông qua ảnh chụp CT-Scan ở ngực

# Lý do và mục tiêu thực hiện đề tài

Hiện nay trên thế giới ung thư phổi là một trong những nguyên nhân gây tử vong hàng đầu trên thế giới. Ung thư phổi là một căn bệnh nguy hiểm, gây tử vong nhiều nhất trên toàn thế giới. Tại Việt Nam, mỗi năm ghi nhận khoảng 26.262 ca mắc mới ung thư phổi và 23.797 ca tử vong vì căn bệnh này.

Các nguyên nhân phổ biến cho ung thư phổi:

* Hút thuốc lá: là nguyên nhân chính gây ra ung thư phổi. Hiện nay, 90% bệnh nhân bị ung thư phổi do hút thuốc lá, và 4% bệnh nhân hít phải khói thuốc hàng ngày.
* Môi trường làm việc: Các tác nhân từ môi trường như khói bụi, làm việc trong môi trường luyện thép, niken, crom và khí than cũng có thể gây ra bệnh ung thư phổi.
* Tiếp xúc với tia phóng xạ: Đây là nguyên nhân gây ra nhiều bệnh ung thư, trong đó có ung thư phổi. Bệnh nhân làm việc trong các mỏ uranium, fluorspar và hacmatite có thể tiếp xúc với tia phóng xạ do hít thở không khí có chứa khi radon.

Chính vì vậy nếu ta có thể chẩn đoán sớm và chính xác khả năng ung thư phổi có thể giúp cải thiện tỷ lệ sống sót và chất lượng cuộc sống của bệnh nhân. Đó chính là một trong những lí do hàng đầu để đề tài này được thực hiện.

Ngoài ra đề tài còn giúp chúng ta tìm hiểu và ứng dụng thực tế đối với các mô hình học máy và deep learning để phân tích hình ảnh y tế có thể giúp các bác sĩ chẩn đoán bệnh nhanh hơn và chính xác hơn, giảm thiểu sai sót do con người gây ra. Giải quyết được vấn đề thiếu hụt nguồn lực y tế. Ở nhiều nơi, đặc biệt là các khu vực nông thôn và các quốc gia đang phát triển, việc tiếp cận các chuyên gia y tế có thể rất hạn chế. Một hệ thống tự động có thể giúp giảm tải cho các chuyên gia và cung cấp hỗ trợ chẩn đoán ở những nơi thiếu hụt nguồn lực y tế. Tất cả đã làm nên ý nghĩa của đề tài này.

# Tìm hiểu về các loại ung thư phổi

Để có thể xây dựng được một mô hình nhận dạng ung thư phổi thông qua ảnh chụp CT-Scan ở ngực thì với vị trí là đại diễn kĩ thuật ta rất cần hiểu sơ lược về kiến thức y khoa đối với loại bệnh này. Điều này sẽ giúp ích cho chúng ta trong quá trình xây dựng mô hình, kiểm nghiệm, hiểu rõ bản chất hơn. Ung thư phổi phổ biến bao gồm 3 loại: Adenocarcinoma, Large Cell Carcinoma, Squamous Cell Carcinoma.

## Adenocarcinoma – Ung thư biểu mô tuyến

Đây là dạng ung thư phổi phổ biến nhất, chiếm khoảng 30% trong tổng số các trường hợp ung thư phổi và khoảng 40% trong các trường hợp ung thư phổi không phải tế bào nhỏ (NSCLC). Thường được tìm thấy ở vùng ngoại biên của phổi, trong các tuyến tiết ra chất nhầy và giúp chúng ta thở.

## Large Cell Carcinoma – Ung thư tế bào lớn

Loại ung thư phổi này phát triển và lan rộng nhanh chóng, chiếm khoảng 10-15% trong tổng số các trường hợp ung thư phổi không phải tế bào nhỏ (NSCLC). Có thể xuất hiện ở bất kỳ vị trí nào trong phổi.

## Squamous Cell Carcinoma – Ung thư biểu mô tế bào vảy

Chiếm khoảng 30% trong tổng số các trường hợp ung thư phổi không phải tế bào nhỏ (NSCLC). Thường liên quan đến việc hút thuốc lá và được tìm thấy ở trung tâm của phổi, nơi các phế quản lớn nối với khí quản và phổi.

# Tìm hiểu về nền tảng hỗ trợ xây dựng mô hình nhận dạng

Hiện nay với sự phát triển của AI, các hệ thống nhận dạng hình ảnh đã đạt được những tiến bộ vượt bậc. Máy tính đã có thể xử lý và phân tích ảnh, học hỏi từ dữ liệu đó để giải quyết các bài toán bằng Deep Learning. Deep Learning, một nhánh của học máy, đặc biệt mạnh mẽ trong việc trích xuất và học các đặc trưng phức tạp từ dữ liệu hình ảnh.

Các Framework và công cụ phổ biến và mạnh mẽ nhất hiện nay là:

* TensorFlow: Được phát triển bởi Google, TensorFlow là một framework mã nguồn mở mạnh mẽ và linh hoạt, được sử dụng rộng rãi trong cộng đồng AI và học máy. TensorFlow hỗ trợ việc xây dựng và huấn luyện các mô hình deep learning phức tạp.
* Keras: Là một API cấp cao chạy trên nền TensorFlow, Keras giúp đơn giản hóa quá trình xây dựng và huấn luyện các mô hình học sâu. Keras cung cấp các công cụ dễ sử dụng và nhanh chóng, giúp nhà phát triển tập trung vào thiết kế mô hình mà không cần quá lo lắng về các chi tiết kỹ thuật phức tạp.
* PyTorch: Được phát triển bởi Facebook's AI Research lab (FAIR), PyTorch là một framework mã nguồn mở linh hoạt và dễ sử dụng cho các mô hình deep learning. PyTorch hỗ trợ tính toán động (dynamic computation graph), giúp việc debug và phát triển mô hình trở nên dễ dàng hơn.
* OpenCV: Là thư viện mã nguồn mở mạnh mẽ dành cho xử lý ảnh và thị giác máy tính. OpenCV cung cấp nhiều công cụ và thuật toán để tiền xử lý và phân tích ảnh trước khi đưa vào các mô hình deep learning.
* scikit-learn: Là một thư viện học máy mã nguồn mở cho Python, scikit-learn cung cấp các công cụ đơn giản và hiệu quả cho phân tích và khai thác dữ liệu. Mặc dù không chuyên về deep learning, scikit-learn có thể được tích hợp với các framework khác để xây dựng các pipeline học máy phức tạp.
* MXNet: Được phát triển bởi Apache Software Foundation, MXNet là một framework deep learning mã nguồn mở, được tối ưu hóa cho hiệu năng cao và hỗ trợ tính toán phân tán. MXNet cũng hỗ trợ tính toán động và được sử dụng rộng rãi trong nghiên cứu và phát triển AI.
* FastAI: Là một thư viện mã nguồn mở xây dựng trên nền PyTorch, FastAI cung cấp các công cụ đơn giản và mạnh mẽ để xây dựng và huấn luyện các mô hình deep learning. FastAI hướng đến việc làm cho deep learning trở nên dễ tiếp cận hơn cho mọi người, từ người mới bắt đầu đến các chuyên gia.

Sự kết hợp của các framework và công cụ này giúp các nhà nghiên cứu và phát triển có thể xây dựng, huấn luyện và triển khai các mô hình deep learning một cách hiệu quả và nhanh chóng. Chúng cung cấp các công cụ cần thiết để giải quyết các bài toán phức tạp trong nhận dạng hình ảnh và các lĩnh vực khác của AI.

# Các phương pháp và kỹ thuật trong việc xây dựng mô hình

## Thu thập và tiền xử lý dữ liệu

* Một trong những điều cần thiết nhất trong việc xây dựng một mô hình học máy đó chính là dữ liệu. Có thể nói nó là gốc rễ cho mọi tri thức mọi nguồn dinh dưỡng tài nguyên để máy tính có thể học được. Chính vì vậy việc thu thập dữ liệu là việc quan trọng nhất và ta cần đảm bảo rằng dữ liệu đủ lớn và liên quan mật thiết đến cái mục tiêu mà ta xây dựng mô hình. Ở đây ta xây dựng một mô hình nhận dạng ung thư phổi qua ảnh chụp CT-Scan ở ngực, vì thế dữ liệu mà chúng ta cần nhất chính là ảnh chụp CT-Scan ở ngực của các nhóm người từ người không bệnh đến người mắc các loại bệnh ung thư phổi. Đó chính là một bước quan trọng khởi đầu cho mô hình này. Trong dự án này chúng ta sẽ sử dụng một tập dữ liệu có sẵn được lấy từ một nguồn đáng tin cậy được đề cập bên dưới.
* Sau khi đã có được dữ liệu liên quan mật thiết đến mục tiêu xây dựng mô hình ta cần xử lý dữ liệu đó. Trong quá trình thu thập có thể có sai sót cũng như là sai sót trong quá trình ghi nhận kết quả,… có rất nhiều trường hợp có thể xảy ra chính vì vậy ta cần tiền xử lý dữ liệu đối với chúng. Ngoài ra việc tiền xử lý có thể giúp dữ liệu phù hợp hơn đối với mô hình chúng ta, giảm thiểu sự phức tạp nhưng vẫn giữ được thậm chí là rõ hơn đối với các đặc trưng có trong dữ liệu. Ví dụ trong dữ liệu ảnh màu ta có một điểm màu sẽ có 3 kênh (RGB) nhưng để đơn giản hóa tính toán nhưng thể hiện rõ đặc trưng hơn ta sẽ xử lý ảnh màu này thành GREY (1 kênh màu) giúp giảm lượng thông tin cần xử lý hơn. Hoặc là đơn giản dữ liệu ảnh có một kích thước khá lớn việc xử lý khá khó khăn thì ta sẽ nén ảnh lại chỉ đơn giản vậy!

## Mạng nơ-ron tích chập (CNN)

* Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) là một loại mạng nơ-ron nhân tạo đặc biệt, được thiết kế để xử lý dữ liệu dạng lưới, chẳng hạn như hình ảnh. CNN được sử dụng rộng rãi trong các nhiệm vụ như phân loại hình ảnh, nhận dạng đối tượng, phân đoạn hình ảnh, và nhiều ứng dụng khác trong thị giác máy tính.

### Cấu trúc của CNN

* CNN thường bao gồm ba loại lớp chính:

1. **Lớp tích chập (Convolutional Layer):**

* Tích chập (Convolution): Lớp này áp dụng các bộ lọc (filters) hay còn gọi là kernel lên đầu vào để tạo ra các bản đồ đặc trưng (feature maps). Bộ lọc này sẽ quét qua toàn bộ đầu vào, thực hiện phép tính chập để trích xuất các đặc trưng quan trọng.
* Các tham số chính:
* Số lượng bộ lọc: Xác định số lượng bản đồ đặc trưng được tạo ra.
* Kích thước bộ lọc: Xác định kích thước của bộ lọc (ví dụ: 3x3, 5x5).
* Stride: Bước dịch chuyển của bộ lọc khi quét qua đầu vào.
* Padding: Thêm viền cho đầu vào để kiểm soát kích thước của bản đồ đặc trưng đầu ra.

1. **Lớp pooling (Pooling Layer):**

* Max Pooling / Average Pooling: Lớp này giảm kích thước không gian của các bản đồ đặc trưng, giữ lại các thông tin quan trọng và giảm bớt số lượng tham số và tính toán trong mạng. Max pooling chọn giá trị lớn nhất trong vùng cửa sổ, còn average pooling lấy giá trị trung bình.
* Kích thước và bước của cửa sổ pooling: Xác định kích thước của vùng pooling và bước dịch chuyển của cửa sổ này.

1. **Lớp kết nối đầy đủ (Fully Connected Layer):**

* Các neuron trong lớp này được kết nối đầy đủ với tất cả các neuron trong lớp trước đó. Lớp này thực hiện chức năng giống như mạng nơ-ron truyền thống và thường được sử dụng ở cuối mạng để đưa ra quyết định phân loại.

### Đặc điểm nổi bật của CNN:

* Tự động trích xuất đặc trưng: Thay vì phải thiết kế các đặc trưng thủ công, CNN học cách trích xuất các đặc trưng quan trọng từ dữ liệu đầu vào thông qua quá trình huấn luyện.
* Tính địa phương và tính chia sẻ tham số: Các bộ lọc được áp dụng trên toàn bộ đầu vào, giúp giảm số lượng tham số và tăng hiệu quả tính toán.
* Chống lại dịch chuyển và biến đổi: Nhờ vào các lớp pooling và tính chia sẻ tham số, CNN có khả năng nhận diện các đối tượng bất kể sự dịch chuyển hoặc biến đổi nhỏ trong hình ảnh.

### Ứng dụng của CNN:

* Nhận diện hình ảnh: Phân loại các đối tượng trong ảnh (ví dụ: nhận diện chó, mèo).
* Nhận diện đối tượng: Xác định và định vị các đối tượng trong ảnh (ví dụ: phát hiện người đi bộ, xe hơi).
* Phân đoạn hình ảnh: Phân chia một hình ảnh thành các khu vực tương ứng với các đối tượng khác nhau.
* Nhận dạng khuôn mặt: Xác định và xác thực danh tính của một người dựa trên khuôn mặt.
* Xử lý ngôn ngữ tự nhiên: CNN cũng được áp dụng trong một số nhiệm vụ xử lý ngôn ngữ tự nhiên, như phân loại văn bản và phân tích cảm xúc.

## Giải quyết vấn đề overfitting của CNN

### Sử dụng lớp Dropout

* Dropout là một kỹ thuật phổ biến được sử dụng trong huấn luyện mạng nơ-ron nhằm giảm thiểu hiện tượng overfitting. Overfitting xảy ra khi mô hình học quá mức các chi tiết và nhiễu trong dữ liệu huấn luyện, dẫn đến hiệu suất kém khi áp dụng lên dữ liệu mới (dữ liệu kiểm tra hoặc thực tế). Dropout giúp giải quyết vấn đề này bằng cách tạm thời loại bỏ một số neuron ngẫu nhiên cùng với các kết nối của chúng trong mỗi lần lặp của quá trình huấn luyện.
* Lợi ích của Dropout
* Giảm overfitting: Bằng cách ngẫu nhiên bỏ bớt các neuron, dropout ngăn cản mạng học thuộc dữ liệu huấn luyện một cách máy móc, giúp mạng học các đặc trưng tổng quát hơn.
* Tăng khả năng tổng quát hóa: Dropout làm cho mạng trở nên ít phụ thuộc vào bất kỳ đặc trưng cụ thể nào, giúp cải thiện hiệu suất trên dữ liệu chưa thấy.
* Tăng tính ngẫu nhiên và đa dạng hóa: Mỗi lần huấn luyện với dropout tạo ra một kiến trúc mạng khác nhau, do đó mô hình cuối cùng giống như tập hợp của nhiều mô hình khác nhau, điều này tương tự với kỹ thuật ensemble.

### Tăng cường dữ liệu - Data Augmentation

* Tăng cường dữ liệu (Data Augmentation) là một kỹ thuật được sử dụng trong học sâu để tăng cường kích thước và sự đa dạng của tập dữ liệu huấn luyện mà không cần thu thập thêm dữ liệu mới. Kỹ thuật này đặc biệt hữu ích trong các ứng dụng như thị giác máy tính, nơi mà dữ liệu huấn luyện có thể bị hạn chế hoặc không đủ phong phú để xây dựng một mô hình mạnh mẽ và tổng quát.
* Mục đích của Tăng Cường Dữ Liệu
* Giảm overfitting: Bằng cách tạo ra nhiều biến thể của dữ liệu huấn luyện, mô hình sẽ ít bị học thuộc vào dữ liệu huấn luyện ban đầu và có khả năng tổng quát hóa tốt hơn trên dữ liệu mới.
* Tăng cường độ đa dạng: Các biến thể của dữ liệu huấn luyện giúp mô hình học được nhiều đặc trưng hơn từ dữ liệu, cải thiện khả năng nhận diện và phân loại.
* Hiệu quả về chi phí: Giúp tăng kích thước tập dữ liệu mà không cần thu thập thêm dữ liệu mới, tiết kiệm thời gian và chi phí.

# Xây dựng mô hình

## Thiết lập môi trường GPU cho Tensorflow

* Ta cần cài đặt các chương trình sau:
* GeForce Experience: Đối với các máy sử dụng GPU NVDIA
* Anaconda
* Tạo môi trường ảo trong Anaconda (Vì hiện tại Tensorflow chỉ hỗ trợ GPU cho python 3.10 nên ta cần điểu chỉnh version python)

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Dùng conda-forge để download các package hỗ trợ GPU: cudatoolkit=11.2 cudnn=8.1.0

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

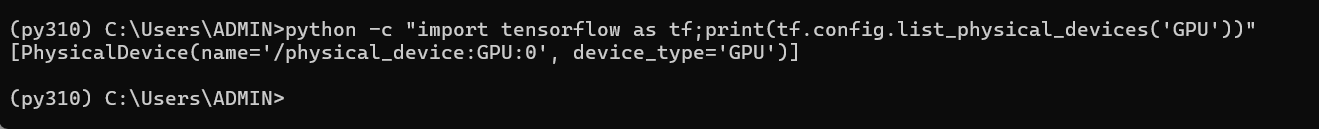
Mô tả được tạo tự động

* Download Tensorflow (Ở đây dự án dùng bản 2.10)

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thực đơn

Mô tả được tạo tự động

* Test thử xem Tensorflow có sử dụng GPU hay chưa



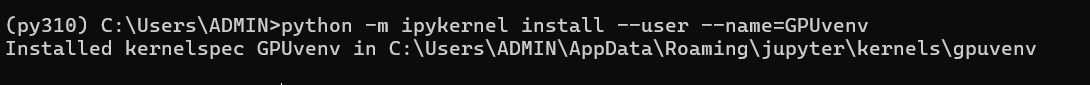
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Tiếp theo ta sẽ tạo ra ipykernel để sử dụng trong notebook

Ảnh có chứa Phông chữ, văn bản, ảnh chụp màn hình, màu đen

Mô tả được tạo tự động



* Tải Jupyter notebook và chạy thử ipykernel





Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

* Như vậy ta đã setup thành công môi trường GPU cho Tensorflow

## Xây dựng mô hình với Jupyter Notebook

* Mở file Jupyter notebook được đính kèm Project

# Đánh giá mô hình và kết luận

* Mở file Jupyter notebook được đính kèm Project