**TRƯỜNG ĐẠI HỌC THỦY LỢI**

****

**ĐỀ TÀI: Sử dụng mô hình CNN trong dự đoán các bệnh lá lúa**

**Sinh viên thực hiện: Trần Chí Tuấn**

**Lớp: 61TH**

**Khoa: CNTT**

**Giáo viên hướng dẫn: Nguyễn Thị Phương Thảo**

MỤC LỤC

[LỜI NÓI ĐẦU 2](#_Toc133967910)

[PHẦN 1: TỔNG QUAN 4](#_Toc133967911)

[**1. Giới thiệu về Deep Learning** 4](#_Toc133967912)

[**1.1 Lịch sử phát triển của Deep learning** 5](#_Toc133967913)

[**1.2 Các thuật toán trong Deep learning** 6](#_Toc133967914)

[PHẦN 2: GIỚI THIỆU BÀI TOÁN 8](#_Toc133967915)

[**1. Bài toán dự đoán bệnh lá lúa** 8](#_Toc133967916)

[**2. Phương pháp học sâu sử dụng trong bài toán** 9](#_Toc133967917)

[**2.1** **Mô tả tập dữ liệu** 9](#_Toc133967918)

[**2.2 Xử lý ảnh** 10](#_Toc133967919)

[**2.2.1 Ảnh trong máy tính** 10](#_Toc133967920)

[**2.2.2 Phép tích chập trong xử lý ảnh** 12](#_Toc133967921)

[**2.3 Mô hình CNN** 14](#_Toc133967922)

[**2.3.1 Convolutional Layer** 15](#_Toc133967923)

[**2.3.2 Rectified Linear Unit (ReLU) Layer** 16](#_Toc133967924)

[**2.2.3 Pooling Layer** 16](#_Toc133967925)

[**2.3.4** **Fully Connected (FC)** 17](#_Toc133967926)

[PHẦN 3: THỰC NGHIỆM 18](#_Toc133967927)

[**1 Chương trình demo** 18](#_Toc133967928)

[**2 Kết quả chạy chương trình** 22](#_Toc133967929)

[**3 Ứng dụng thực tế và kết luận** 23](#_Toc133967930)

[KẾT LUẬN 25](#_Toc133967931)

# 

# **LỜI NÓI ĐẦU**

Những năm gần đây, trí tuệ nhân tạo (artificial intelligence–AI) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1–động cơ hơi nước, 2–năng lượng điện, 3–công nghệ thông tin). Nó đang dần trở thành một thế lực mạnh mẽ mang lại sự thay đổi đáng kể cho nhiều lĩnh vực đời sống của con người. Với khả năng xử lý dữ liệu nhanh chóng, AI đã được ứng dụng mạnh mẽ trong các lĩnh vực như y tế, tài chính, hàng không vũ trụ, và cả trong cuộc sống hàng ngày của chúng ta.

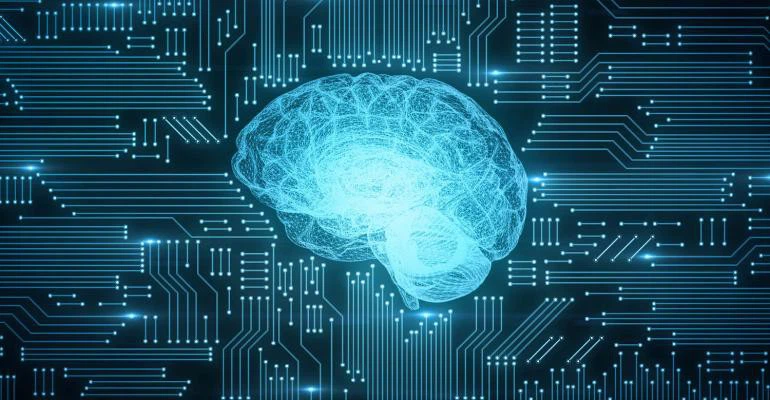
Cũng phải nói, trong những năm gần đây, học sâu (deep learning) đã trở thành một công cụ quan trọng để khai phá thêm nhiều ứng dụng của AI. Sâu hơn so với các phần mềm truyền thống, học sâu có thể học hỏi các mô hình trừu tượng từ dữ liệu, giúp giải quyết những vấn đề phức tạp và truyền tải thông tin nhanh hơn.

Trong học sâu, một thuật toán quan trọng là Convolutional Neural Network (CNN), một mạng nơ-ron với cấu trúc đặc biệt được sử dụng để xử lý các dữ liệu có tính chất như hình ảnh, video và âm thanh. CNN có thể học hỏi các đặc trưng cục bộ từ dữ liệu đầu vào, giúp nó trở nên hiệu quả hơn trong việc phát hiện và nhận dạng đối tượng trong ảnh và video.

Vì vậy, nghiên cứu và phát triển thuật toán CNN là rất quan trọng vì nó đóng vai trò chủ chốt trong việc áp dụng học sâu cho các lĩnh vực mang tính cấu trúc cao. Báo cáo nghiên cứu này sẽ tập trung đánh giá và phân tích tầm quan trọng của thuật toán CNN trong nghiên cứu học sâu, cùng với các ứng dụng thực tế và lợi ích mang lại cho cuộc sống của chúng ta.

# **PHẦN 1: TỔNG QUAN**

## **1. Giới thiệu về Deep Learning**

****

Hình 1. Deep learning là gì?

Deep learning là một phương pháp học máy sâu sử dụng các mạng nơ-ron nhân tạo để học các mô hình phức tạp. Với các mạng nơ-ron nhân tạo, deep learning có thể học và hiểu những dữ liệu lớn và phức tạp hơn bao giờ hết. Deep learning hoạt động dựa trên mô hình của một mạng nơ-ron (hoặc nhiều mạng nơ-ron) trong đó các nơ-ron được cấu hình để tìm kiếm các đặc trưng tốt nhất cho một tác vụ cụ thể.

Các ứng dụng của deep learning rất phong phú và đa dạng, bao gồm nhận dạng ảnh, xử lý ngôn ngữ tự nhiên, phân tích âm thanh và video, và các ứng dụng trên thị trường tài chính và y tế. Deep learning được sử dụng để đọc kết quả x-quang và MRI trong y tế, giúp kiểm tra chất lượng và an toàn trong các nhà máy sản xuất, tối ưu hoá quy trình sản xuất cho các công ty, và cải thiện đáng kể trải nghiệm người dùng cho các sản phẩm công nghệ như ứng dụng điện thoại thông minh và trợ lý ảo.

Tóm lại, deep learning đang ngày càng được sử dụng rộng rãi trong nhiều lĩnh vực vì khả năng xử lý các dữ liệu lớn và phức tạp, đẩy mạnh nỗ lực của chúng ta trong việc giải quyết các vấn đề phức tạp trong đời sống.

### **1.1 Lịch sử phát triển của Deep learning**

Khoảng 10 năm trở lại đây, deep learning mới nổi lên như là một giải pháp cho những vấn đề phức tạp và hao tốn tài nguyên. Sự phát triển mạnh mẽ của phần cứng máy tính khiến cho những chương trình deep learning chạy nhanh hơn, nhận được nhiều dữ liệu hơn và có thể tiếp cận đến những vấn đề lớn hơn.

Thế nhưng, những mô hình deep learning vẫn luôn được nghiên cứu và phát triển từ rất lâu trước đó. Một vài dấu mốc quan trọng trong quá trình hình thành và phát triển của deep learning có thể kể đến như:

1. McCulloch-Pitts neuron (1943): là mô hình neuron đầu tiên được đề xuất.

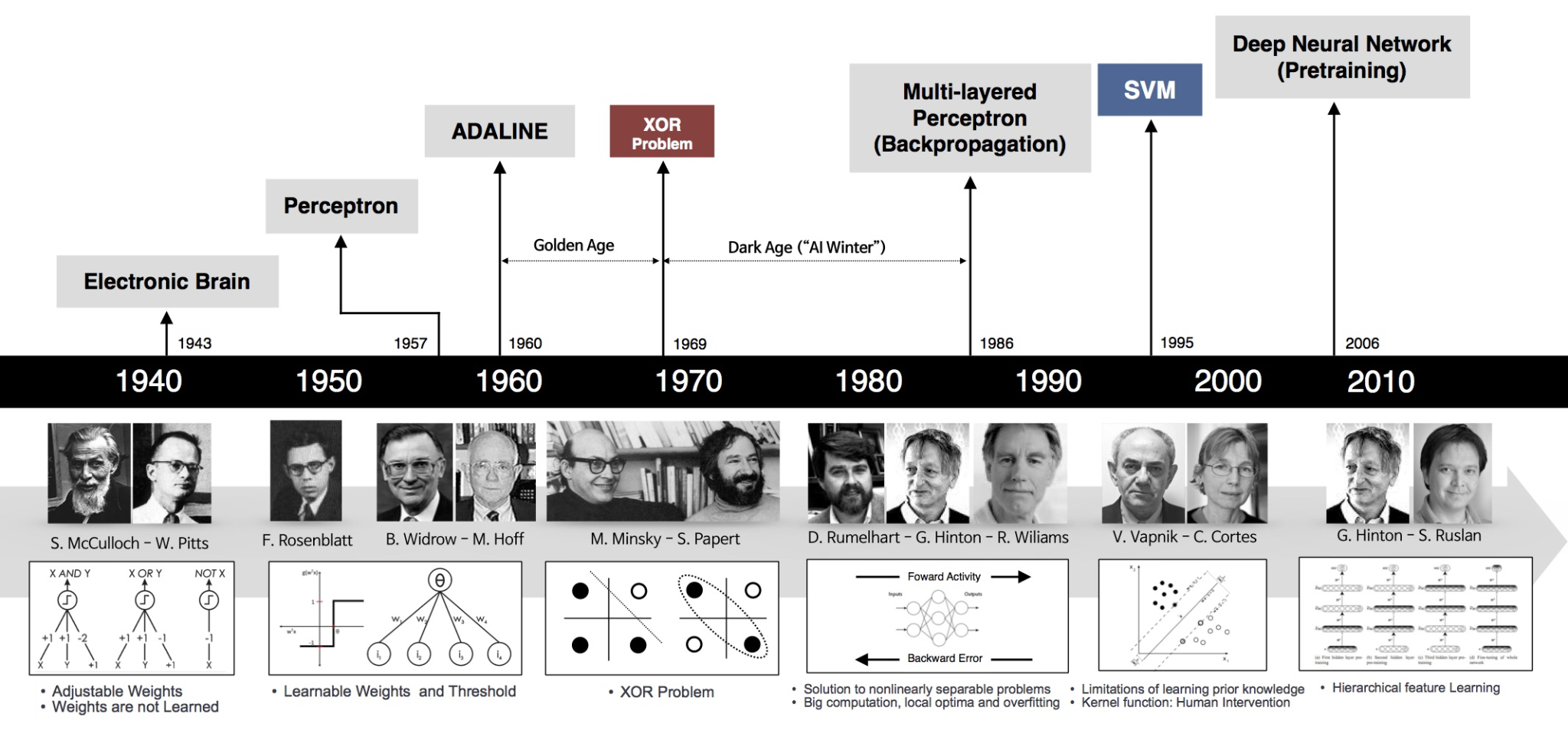
2. Perceptron (1957): là mô hình đầu tiên của deep learning, được phát triển bởi Rosenblatt, có thể học các hàm tuyến tính.

3. Backpropagation (1974): đây là phương pháp huấn luyện mạng neuron đa lớp đầu tiên, cho phép cập nhật các trọng số theo thời gian.

4. Convolutional Neural Networks (CNNs) (1998): kiến ​​trúc mạng neuron được thiết kế để phân loại hình ảnh, cho phép mô hình học được các đặc trưng trừu tượng từ dữ liệu hình ảnh.

5. Recurrent Neural Networks (RNNs) (1997): kiến trúc mạng neuron cho phép mô hình học được các chuỗi dữ liệu, như dữ liệu văn bản hay âm thanh.

6. AlphaGo (2016): đây là lần đầu tiên mô hình deep learning đã đánh bại một vận động viên chuyên nghiệp trong trò chơi cờ vây.



Hình 2. Lịch sử phát triển của Deep learning

Càng về sau, deep learning đã dần cho thấy tầm ảnh hưởng của nó đến nhiều lĩnh vực của cuộc sống như khoa học, khai thác dữ liệu và công nghệ thông tin. Hiện nay, nó được coi là một trong những xu hướng công nghệ quan trọng nhất của thế giới ngày nay

### [**1.2 Các thuật toán trong Deep learning**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.3dy6vkm)

Các thuật toán trong deep learning rất đa dạng, mỗi thuật toán lại có một cơ chế hoạt động khác nhau, nhưng điểm chung là đều được xây dựng dựa trên một hay nhiều neuron nhân tạo.

Một neuron nhân tạo hoạt động tương tự như một neuron sinh học, bao gồm các thành phần chính như dendrite, axon, soma và synapse. Đầu vào của một neuron là tín hiệu đầu ra từ các neuron khác thông qua các dendrite, nó sẽ giải mã và tính toán tín hiệu đó và tạo ra tín hiệu đầu ra thông qua axon. Tiếp tục như thế, tín hiệu đầu ra sẽ được chuyển đến các neuron khác thông qua các synapse.

Trong quá trình này, neuron nhân tạo sẽ tính toán tổng trọng số đầu vào và đưa ra kết quả đầu ra bằng cách sử dụng một hàm kích hoạt, như sigmoid hoặc ReLU. Tuy nhiên, quá trình tính toán này sẽ có sai số.

Để giảm sai số, chúng ta cần cập nhật lại trọng số. Cách cập nhật trọng số của một neuron nhân tạo là sử dụng thuật toán lan truyền ngược (backpropagation). Theo cách này, các trọng số có ảnh hưởng lớn đến sai số của mạng sẽ được điều chỉnh để giảm thiểu nó.

Điều kiện dừng vòng lặp của một neuron nhân tạo có thể được xác định bởi một số yếu tố khác nhau, như số lần lặp, trọng số đầu vào, độ chính xác của đầu ra, độ chịu lỗi, cũng như tốc độ học của mạng. Thông thường, khi độ chịu lỗi đạt đến một mức độ chấp nhận được thì việc huấn luyện sẽ dừng lại.



Hình 3. Các thuật toán trong Deep learning

Dưới đây là một số thuật toán được cho là quan trọng trong deep learning:

1. Convolutional Neural Networks (CNNs): CNNs là một loại mạng nơ-ron sử dụng trong xử lý ảnh và nhận dạng đối tượng. CNNs áp dụng các bộ lọc (filters) trên các điểm ảnh trên đầu vào (input), cho phép bỏ qua các chi tiết không quan trọng và tập trung vào các đặc trưng của đối tượng.

2. Recurrent Neural Networks (RNNs): RNNs là một loại mạng nơ-ron thường được sử dụng cho các tác vụ liên quan đến ngôn ngữ tự nhiên như dịch thuật hay ngữ cảnh. Cơ chế hoạt động của RNNs là giá trị đầu ra của mạng phụ thuộc vào đầu ra của nó ở bước trước đó, cho phép nó lưu giữ thông tin từ các bước trước đó và sử dụng thông tin đó để đưa ra dự đoán.

3. Generative Adversarial Networks (GANs): GANs là một loại mạng nơ-ron được sử dụng để tạo ra các khối dữ liệu giống hệt với các tập dữ liệu đã huấn luyện. GANs bao gồm hai mạng nơ-ron tương tác với nhau, generator và discriminator, cho phép tạo ra các hình ảnh, âm thanh hay văn bản mới dựa trên các đặc trưng của dữ liệu đã huấn luyện.

4. Deep Reinforcement Learning (DRL): DRL là một lớp các thuật toán học tăng cường được sử dụng để xử lý các tác vụ yêu cầu khả năng tương tác động. DRL kết hợp các thuật toán học máy với các kỹ thuật học tăng cường như đánh giá (policy evaluation), mô tả (model description) và tối ưu hóa (value optimization) để dẫn dắt quá trình học.

Các thuật toán trên có những cơ chế hoạt động khác nhau tùy thuộc vào mục đích sử dụng của chúng. Tuy nhiên, các thuật toán này đều được sử dụng để giải quyết các vấn đề phức tạp và nắm giữ vai trò quan trọng trong các ứng dụng của deep learning.

# [**PHẦN 2: GIỚI THIỆU BÀI TOÁN**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.lnxbz9)

## [**1. Bài toán dự đoán bệnh lá lúa**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.35nkun2)

Áp dụng khoa học kĩ thuật trong nhận dạng các loại bệnh trên lá lúa đang là một lĩnh vực nghiên cứu mới phù hợp với thời đại và xu hướng phát triển hiện nay. Các loại bệnh về lá có sức tàn phá đáng kể và có thể lây lan trên diện rộng, nếu không phát hiện và giải quyết kịp thời sẽ gây ra tổn hại lớn, ảnh hưởng rất nhiều đến kinh tế của người nông dân.

Các phương pháp thăm đồng truyền thống và để phát hiện bệnh trên cây trồng tốn nhiều thời gian và công sức. Tuy nhiên, việc sử dụng thuật toán CNN trong nhận dạng bệnh trên cây trồng sẽ cung cấp một giải pháp chính xác và hiệu quả hơn để giải quyết vấn đề này.

Thuật toán CNN là một kiến trúc deep learning được sử dụng rộng rãi trong phân tích ảnh và video. Nó được thiết kế gồm nhiều lớp có khả năng học tập các đặc trưng từ ảnh đầu vào một cách tự động. Lý do chọn thuật toán CNN cho nghiên cứu này là khả năng tự động học hỏi và nhận biết của nó với các mẫu huấn luyện, vì vậy, nó là hướng đi hợp lý để giải quyết một cách chính xác bài toán chúng ta đã đặt ra

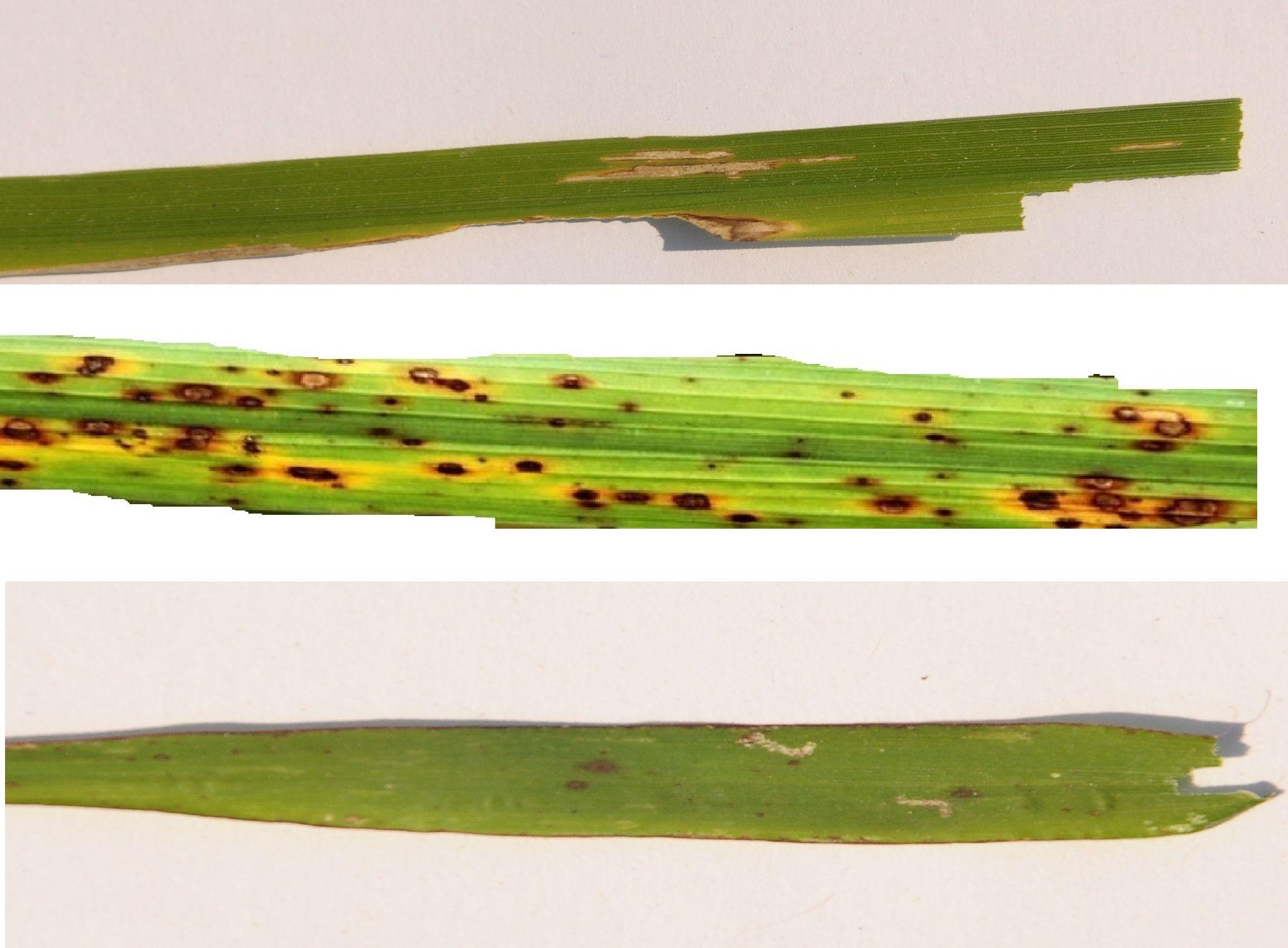
Với phương pháp này, người nông dân có thể nhanh chóng xác định loại bệnh ảnh hưởng đến lá lúa, sớm có các biện pháp thích hợp để kiểm soát sự lây lan của bệnh. Phát hiện sớm các bệnh sẽ giới hạn thiệt hại do nó gây ra, giảm thiểu mất mát năng suất và bảo vệ thu nhập kinh tế.

Nhìn chung, nhận dạng các loại bệnh trên lá lúa bằng thuật toán CNN là một chủ đề nghiên cứu quan trọng có thể có tác động lớn đến ngành nông nghiệp, đặc biệt là với các nước đang phát triển như Việt Nam.

## [**2. Phương pháp học sâu sử dụng trong bài toán**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.1ksv4uv)

### [**2.1**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.44sinio) **Mô tả tập dữ liệu**

Tại Việt Nam, có khoảng hơn 30 loại bệnh lá lúa đã được xác định bao gồm cả các bệnh do nấm, vi khuẩn và virus gây ra. Một số trong số các bệnh lá lúa phổ biến ở Việt Nam bao gồm đốm nâu, vàng lá, bạc lá, đốm lá, lá xoăn.... Nhưng trong báo cáo này, chúng ta sẽ tập trung vào 3 loại bệnh phổ biến hiện nay là bạc lá, đốm nâu và đốm lá.



Hình 4. (a) Bạc lá, (b) Đốm nâu, (c) Đốm lá

Vấn đề đặt ra là cần phải tạo ra một mô hình có thể phát hiện được các loại bệnh trên lá lúa này bằng thuật toán học máy.

Bộ dữ liệu trên được thu thập từ Kho lưu trữ học máy của UCI [1] với mỗi loại bệnh sẽ có 40 hình ảnh làm dữ liệu huấn luyện.

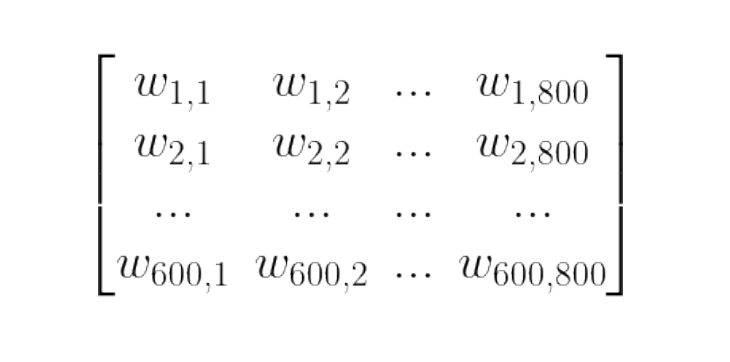
### [**2.2 Xử lý ảnh**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.44sinio)

#### [**2.2.1**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.44sinio) **Ảnh trong máy tính**

Một hình ảnh trong máy tính được tạo ra từ một số lượng lớn các điểm ảnh. Mỗi điểm ảnh (hay còn gọi là pixel) là điểm nhỏ nhất của hình ảnh có thể được thu nhận và xử lý bởi máy tính.

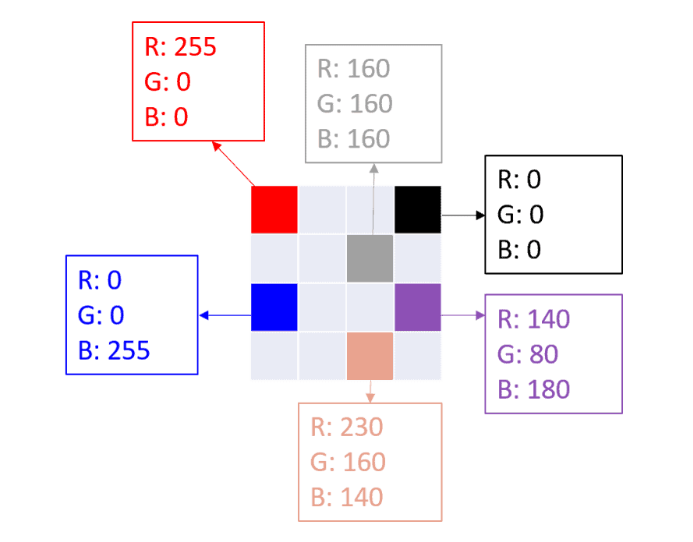
Pixel (viết tắt của Picture Element) là đơn vị cơ bản để tạo ra hình ảnh trong máy tính. Mỗi pixel được đại diện bằng một số nguyên dương và được sắp xếp thành một mảng hai chiều hay ma trận.

Để dễ hiểu hơn, giả sử ta có một ảnh với kích thước 800 \* 600 tức chiều dài của ảnh là 800px, chiều rộng 600px. Vậy bức ảnh trên có thể biểu diễn bằng một ma trận có kích thước 600 \* 800 (vì định nghĩa ma trận là số hàng nhân số cột)



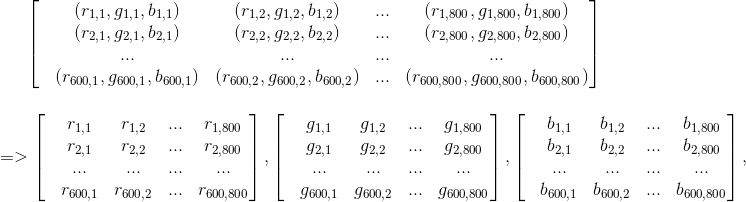
Trong đó mỗi phần tử wi j là một pixel.

Như vậy có thể hiểu là mỗi pixel thì biểu diễn một màu và bức ảnh trên là sự kết hợp rất nhiều pixel.

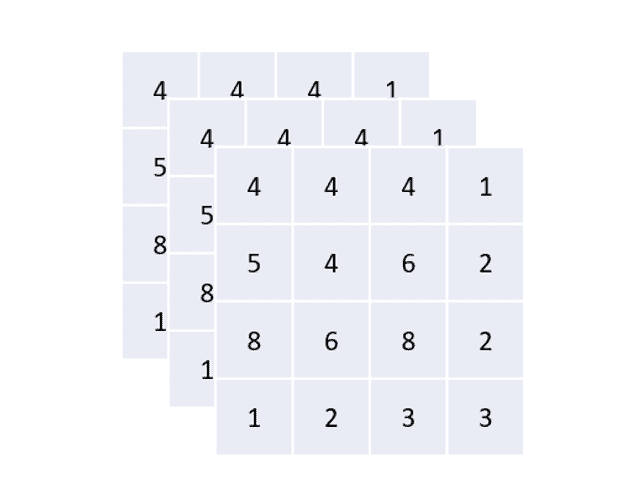
Tuy nhiên để biểu diễn 1 màu ta cần 3 thông số (r,g,b) nên gọi wij = (rij,gij,bij). 

Hình 5. Ma trận ảnh màu RGB

Thông thường, để tiện lưu trữ và xử lý, không thể lưu chúng trong 1 ma trận mà sẽ phải tách mỗi giá trị màu trong mỗi pixel ra một ma trận riêng



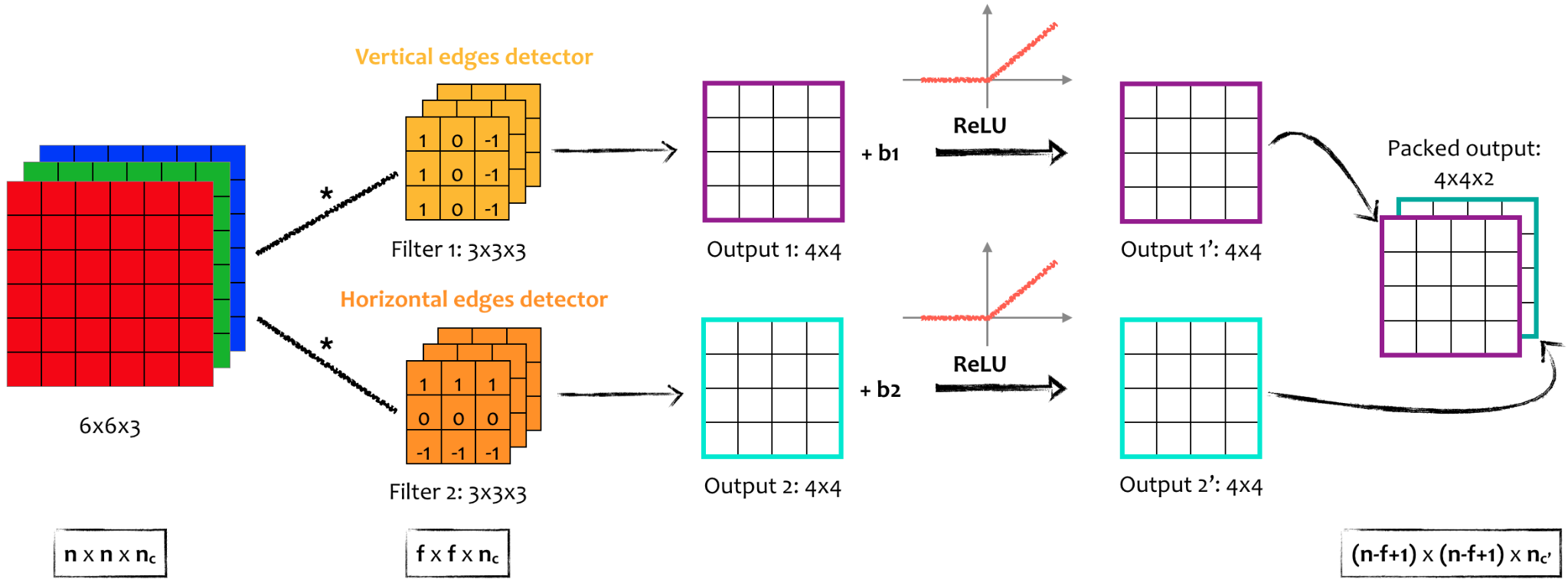
Hình 3. Tách ma trận biểu diễn màu ra 3 ma trận, mỗi ma trận lưu giá trị 1 màu.

Như vậy, một ảnh màu rgb trên máy tính là một ma trận 3D với các màu đỏ (R), xanh lá (G) và xanh dương (B)

Hình 6. Ma trận 3D RGB

#### [**2.2.2**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.44sinio) **Phép tích chập trong xử lý ảnh**

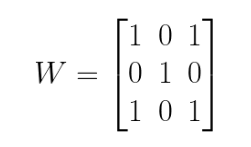
Trong xử lý hình ảnh, tích chập là một phép tính cơ bản được sử dụng để lọc và chuyển đổi hình ảnh với các mục đích khác nhau bao gồm khử nhiễu, phát hiện đường biên và nén ảnh. Mạng neuron tích chập (CNN) đang ngày càng phổ biến trong những năm gần đây như một công cụ mạnh mẽ cho các tác vụ phân tích hình ảnh và thị giác máy tính mà cốt lõi của nó là phép tích chập, góp phần đáng kể vào hiệu quả và độ chính xác của thuật toán này.



Hình 7. Phép tích chập trong xử lý ảnh

Để cho dễ hình dung ta sẽ lấy ví dụ trên ảnh xám, tức là ảnh được biểu diễn dưới dạng ma trận A kích thước m\*n.

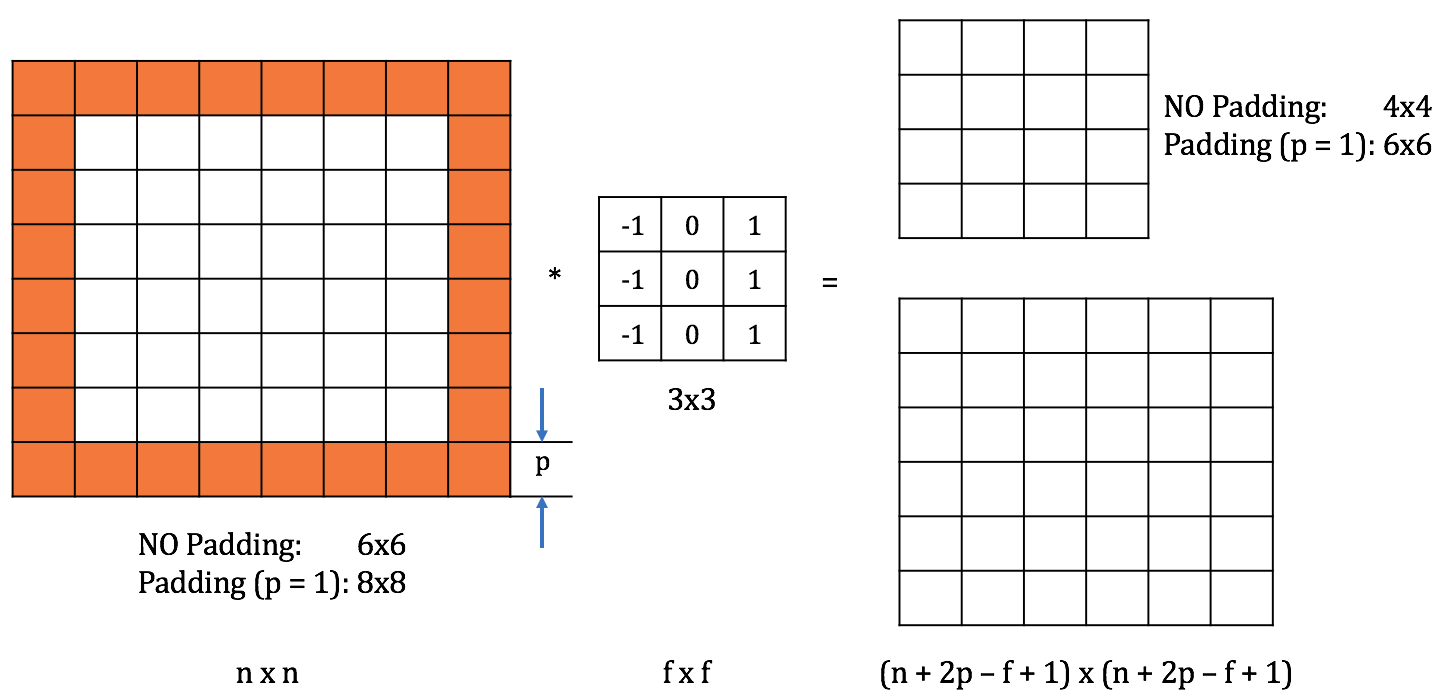
Ta định nghĩa kernel là một ma trận vuông kích thước k\*k trong đó k là số lẻ. k có thể bằng 1, 3, 5, 7, 9,... Ví dụ kernel kích thước 3\*3



Với mỗi phần tử xi j trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W có phần tử xi j làm trung tâm (đây là vì sao kích thước của kernel thường lẻ) gọi là ma trận A. Sau đó tính tổng các phần tử của phép tính element-wise của ma trận A và ma trận W, rồi viết vào ma trận kết quả Y.

Mỗi lần thực hiện phép tính convolution xong thì kích thước ma trận Y đều nhỏ hơn

X. Tuy nhiên giờ ta muốn ma trận Y thu được có kích thước bằng ma trận X => Tìm cách giải quyết cho các phần tử ở viền => Thêm giá trị 0 ở viền ngoài ma trận X.



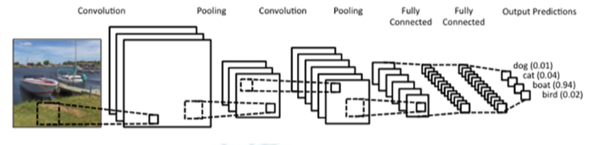
Hình 8. Padding, No Padding

Phép tính này gọi là convolution với padding=1. Padding=k nghĩa là thêm k vector 0 vào mỗi phía (trên, dưới, trái, phải) của ma trận.

### **2.3 Mô hình CNN**

Mô hình CNN chỉ đơn giản gồm một vài layer của convolution kết hợp với các hàm kích hoạt phi tuyến như ReLU hay tanh để tạo ra thông tin ở mức trừu tượng hơn cho các layer tiếp theo. Trong mô hình mạng nơ-ron truyền thẳng (feedforward neural network), các layer kết nối trực tiếp với nhau thông qua một trọng số(weighted vector). Các layer này còn được gọi là có kết nối đầy đủ(fully connected layer) hay affine layer. Trong mô hình CNN thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơron ở layer tiếp theo sinh ra từ filter áp đặt lên một vùng ảnh cục bộ của nơron layer trước đó. Mỗi layer như vậy được áp đặt các filter khác nhau, thông thường có vài trăm đến vài nghìn filter như vậy. Một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong suốt quá trình huấn luyện, CNN sẽ tự động học được các thông số cho các filter. Ví dụ,trong nhiệm vụ phân lớp ảnh như được minh họa trong hình dưới, CNN sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự draw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



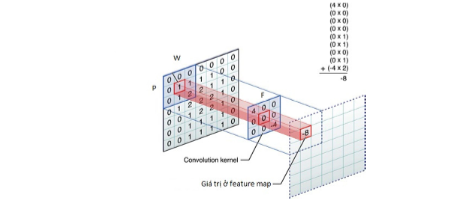
Hình 9. Mô hình mạng Nơ-ron tích chập

CNN có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer cho tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter. Đó là lý do tại sao CNN cho ra mô hình với độ chính xác rất cao.

Các lớp trong mô hình mạng CNN bao gồm:

#### **2.3.1 Convolutional Layer**

Layer này chính là nơi thể hiện tư tưởng ban đầu của CNN. Thay vì kết nối toàn bộ điểm ảnh, layer này sẽ sử dụng một tập các bộ lọc (filters) có kích thước nhỏ so với ảnh (thường là 5×5 hoặc 3×3) áp vào một vùng trong ảnh và tiến hành tính tích chập giữa bộ lọc và giá trị điểm ảnh trong vùng cục bộ đó. Bộ Lọc sẽ lần lượt được dịch chuyển theo một giá trị bước trượt (stride) chạy dọc theo ảnh và quét toàn bộ ảnh.



Hình 10. Lớp tích chập

Như vậy, với một bức ảnh 32×32 và một filter 3×3, ta sẽ có kết quả là một tấm ảnh mới có kích thước 32×32 (với điều kiện đã thêm padding vào ảnh gốc để tính tích chập cho các trường hợp filter quét ra các biên cạnh) là kết quả tích chập của filter và ảnh. Với bao nhiêu filter trong lớp này thì ta sẽ có bấy nhiêu ảnh tương ứng mà lớp này trả và được truyền vào lớp tiếp theo. Các trọng số của filter ban đầu sẽ được khởi tạo ngẫu nhiên và sẽ được học dần trong quá trình huấn luyện mô hình. Hình 2.3 minh họa của một phép tính convolution với bộ lọc có kích thước 3×3.

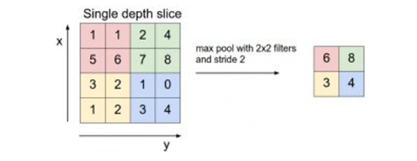
#### **2.3.2 Rectified Linear Unit (ReLU) Layer**

Layer này thường được cài đặt ngay sau layer Convolution. Layer này sử dụng hàm kích hoạt f(x) = max(0, x). Nói một cách đơn giản, layer này có nhiệm vụ chuyển toàn bộ giá trị âm trong kết quả lấy từ lớp Convolution thành giá trị 0. Ý nghĩa của cách cài đặt này chính là tạo nên tính phi tuyến cho mô hình. Tương tự như trong mạng truyền thẳng, việc xây dựng dựa trên các phép biến đổi tuyến tính sẽ khiến việc xây dựng đa tầng đa lớp trở nên vô nghĩa. Có rất nhiều cách để khiến mô hình trở nên phi tuyến như sử dụng các hàm kích hoạt sigmoid, tanh,... nhưng hàm f(x) = max(0, x) dễ cài đặt, tính toán nhanh mà vẫn hiệu quả.

#### **2.2.3 Pooling Layer**

Layer này sử dụng một cửa sổ trượt quét qua toàn bộ ảnh dữ liệu, mỗi lần trượt theo một bước trượt (stride) cho trước. Khác với layer Convolutional, layer Pooling không tính tích chập mà tiến hành lấy mẫu (subsampling). Khi cửa sổ trượt trên ảnh, chỉ có một giá trị được xem là giá trị đại diện cho thông tin ảnh tại vùng đó (giá trị mới) được giữ lại. Các phương thức lấy phổ biến trong layer Pooling là MaxPooling (lấy giá trị lớn nhất), MinPooling (lấy giá trị nhỏ nhất) và Average Pooling(lấy giá trị trung bình).

Xét một ảnh có kích thước 32×32 và layer Pooling sử dụng bộ lọc có kích thước 2×2 với bước trượt stride là 2, phương pháp sử dụng là MaxPooling. Bộ Lọc sẽ lần lượt trượt qua ảnh, với mỗi lần trượt chỉ có giá trị lớn nhất trong 4 giá trị nằm trong vùng cửa sổ 2×2 của bộ lọc được giữ lại và đưa vào ma trận đầu ra. Như vậy,sau khi qua layer Pooling, ảnh sẽ giảm kích thước xuống còn 16×16 (kích thước mỗi chiều giảm 2 lần).



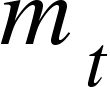
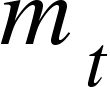
Hình 11. Lớp Pooling

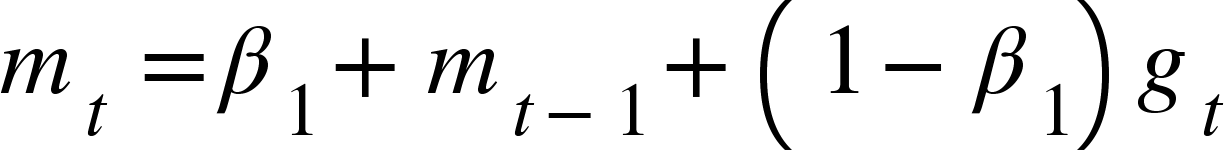
Pooling Layer có vai trò giảm kích thước dữ liệu. Với một bức ảnh kích thước lớn qua nhiều Pooling Layer sẽ được thu nhỏ lại tuy nhiên vẫn giữ được những đặc trưng cần cho việc nhận dạng (thông qua cách lấy mẫu). Việc giảm kích thước dữ liệu sẽ làm giảm lượng tham số, tăng hiệu quả tính toán và góp phần kiểm soát hiện tượng quá khớp (overfitting).

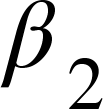
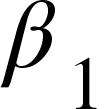
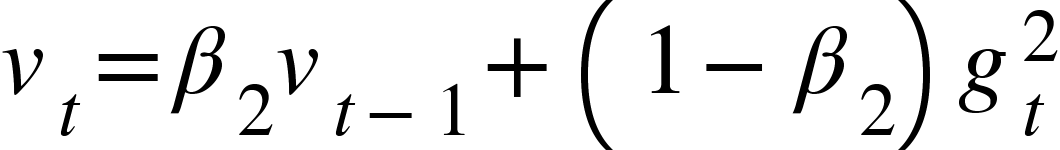
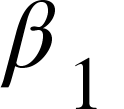
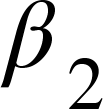
#### **2.3.4** **Fully Connected (FC)**

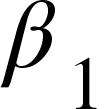
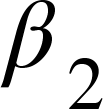
Layer Layer này tương tự với layer trong mạng nơron truyền thẳng, các giá trị ảnh được liên kết đầy đủ của các nơron trong layer tiếp theo. Sau khi ảnh được xử lý và rút trích đặc trưng của các layer trước đó, dữ liệu ảnh sẽ không còn quá lớn so với mô hình truyền thẳng nên ta có thể sử dụng mô hình truyền thẳng để tiến hành nhận dạng

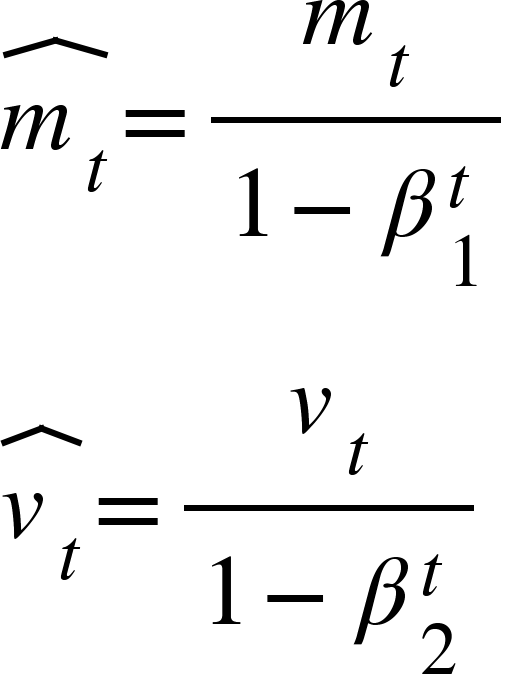
[**2.4 Thuật toán Adam cập nhật trọng s**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.z337ya)**ố**

Adam (Adaptive Moment Estimation) [8] là một thuật toán cho phép tính tốc độ học thích ứng với mỗi trọng số. Adam không chỉ lưu trữ trung bình bình phương các gradient trước đó như Adadelta mà còn lưu cả giá trị trung bình mô-men . Các giá trị và được tính bởi công thức:





trong đó và là các trọng số không âm, thường được chọn là = 0.9 và = 0.999 . Nếu khởi tạo mt và t v là các vector 0, các giá trị này có khuynh hướng nghiêng về 0, đặc biệt là khi và xấp xỉ bằng 1. Do vậy, để khắc phục, các giá trị này được ước lượng bằng cách:

Sau đó cập nhật các trọng số theo công thức:

# [**PHẦN 3: THỰC NGHIỆM**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.1y810tw) {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>&#x3B8;</mi><mrow><mi>t</mi><mo>+</mo><mn>1</mn></mrow></msub><mo>=</mo><msub><mi>&#x3B8;</mi><mi>t</mi></msub><mo>-</mo><mfrac><mi>&#x3B7;</mi><mrow><msqrt><mover><msub><mi>v</mi><mi>t</mi></msub><mo>^</mo></mover></msqrt><mo>+</mo><mi>&#x3B5;</mi></mrow></mfrac><mover><msub><mi>m</mi><mi>t</mi></msub><mo>^</mo></mover><mo>,</mo><mo>&#xA0;</mo><mi>&#x3B5;</mi></mstyle></math>","truncated":false}

## [**1 Chương trình demo**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.4i7ojhp)

Các bước giải quyết bài toán:

* Đọc dữ liệu vào
* Chuẩn hóa dữ liệu về khoảng [0,1]
* Chia bộ train, test (Mỗi loại bệnh 30 mẫu huấn luyện và 10 mẫu test)
* Xây dựng mô hình:

Data augmentation layer (optional): Bao gồm thu phóng ngẫu nhiên, xoay và lật ngẫu nhiên các hình ảnh đầu vào để tạo ra các biến thể mới của dữ liệu phòng ngừa overfitting

Lớp Conv2D 1: Áp dụng 16 bộ lọc có kích thước 3x3 với hàm kích hoạt ReLU

Lớp MaxPooling2D 1: Giảm kích thước đối tượng bằng cách lấy giá trị max là 2x2, giảm độ phức tạp tính toán và tăng tính bất biến.

Conv2D layer 2: Áp dụng 32 bộ lọc kích thước 3x3 với hàm kích hoạt ReLU, xử lý dữ liệu ở cấp cao hơn.

MaxPooling2D lớp 2: Có tác dụng giống như lớp thứ 3, nhưng được sử dụng làm hàm đầu ra của lớp thứ 4.

Lớp Conv2D 3: Áp dụng 64 bộ lọc có kích thước 3x3 với hàm kích hoạt ReLU, tiếp tục khai thác những đặc trưng của mẫu ở mức cao hơn nữa.

MaxPooling2D lớp 3: Cùng mục đích với lớp thứ 3 và 5, nhưng được áp dụng làm đầu ra của lớp thứ 6.

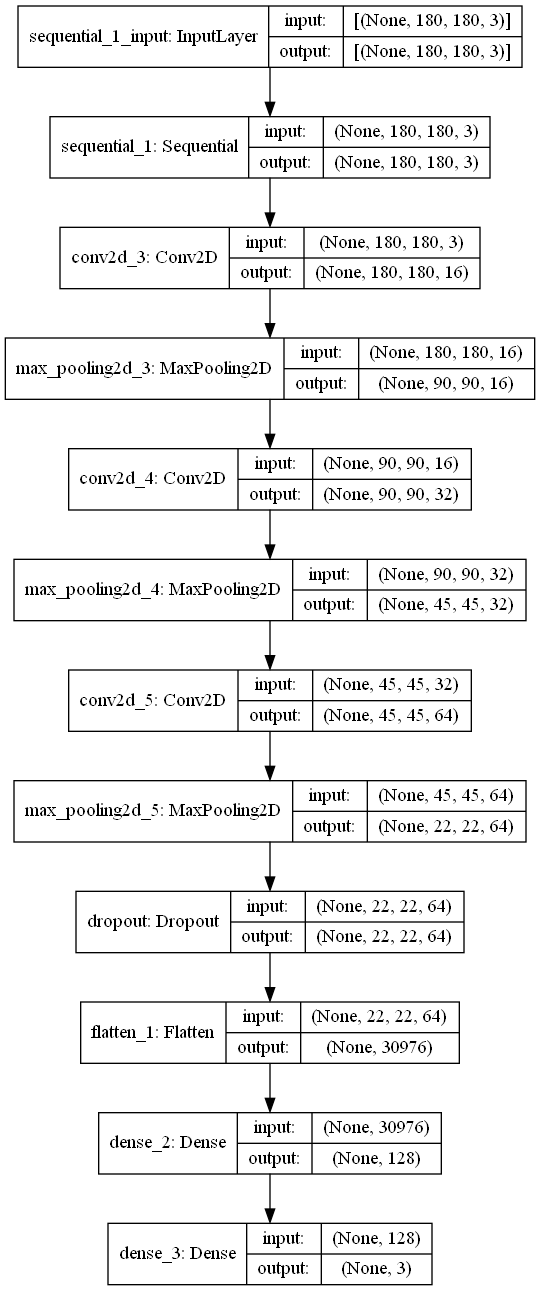
Dropout layer (optional): Loại bỏ ngẫu nhiên 10% số nơ-ron trong quá trình huấn luyện, phòng ngừa overfitting và thúc đẩy quá trình khái quát hóa.

Flatten layer: Chuyển đổi ma trận 3D từ các lớp tích chập trước đó thành một mảng 1D.

Dense layer 1: Lớp được kết nối đầy đủ với 128 nơ-ron và hàm kích hoạt ReLU, tổng hợp các thuộc tính học được bởi các lớp tích chập và tạo ra các khối trừu tượng ở cấp cao hơn.

Dense layer 2 (output layer): Là lớp liên kết đầy đủ với 3 nơron, mỗi nơron tương ứng với một trong các loại bệnh trên lá lúa. Là lớp đầu ra, dự đoán xác suất của từng loại bệnh.

* Mô hình thuật toán:

**

Hình 12. Sơ đồ khối mô hình thuật toán

* Huấn luyện và đánh giá mô hình

## [**2 Kết quả chạy chương trình**](https://docs.google.com/document/d/1RSpWLExr6tjz34YlEAm9t-Dsm_2vb8fM/edit#heading=h.2xcytpi)

* Kết quả: 2 lần chạy
  + Lần 1: chạy với bộ dữ liệu ban đầu, ko có hiệu chỉnh:

Hàm mất mát: 0.9988975524902344

Độ chính xác: 0.7333333492279053

Kết quả chính xác: 22/30

* + Lần 2: chạy với bộ dữ liệu có hiệu chỉnh (dùng augumentation)

Hàm mất mát: 0.3825152814388275

Độ chính xác: 0.8666666746139526

Kết quả chính xác: 26/30

* Có thể thấy sự khác biết rõ ràng giữa thông số của 2 lần chạy. Đối với tập dữ liệu đã được tăng cường, sai số giảm dần về 0 (Từ 0.999 còn 0.383). Độ chính xác tăng lên (Từ 0.733 lên 0.867) và kết quả dự đoán đã tăng thêm 4 mẫu (Từ 22 lên 26)
* So sánh mô hình có và không có augumentation:
* Đối với dữ liệu không có augumentation: Chỉ sử dụng dữ liệu gốc từ tập dữ liệu huấn luyện.

Điểm yếu: Nếu tập dữ liệu huấn luyện nhỏ hoặc không đủ đa dạng, mô hình có thể dễ bị overfitting (quá khớp) và khó khăn trong việc tổng quát hóa với dữ liệu chưa thấy.

Hiệu suất của mô hình có thể giảm khi gặp dữ liệu mới hoặc nếu dữ liệu có nhiều biến thể.

* CNN với augmentation:

Trong trường hợp này, mô hình sử dụng augmentation để tăng cường dữ liệu huấn luyện. Augmentation là quá trình tạo ra những biến thể mới của hình ảnh thông qua các phép biến đổi như xoay, phóng to, lật hay thay đổi độ sáng, độ tương phản,...

Điểm mạnh: Giúp tăng độ đa dạng của dữ liệu, giúp mô hình học được nhiều đặc điểm hơn từ các tình huống khác nhau, giúp mô hình tránh tình trạng overfitting.

Cải thiện hiệu suất của mô hình trên dữ liệu chưa thấy và khi dữ liệu có nhiều biến thể.

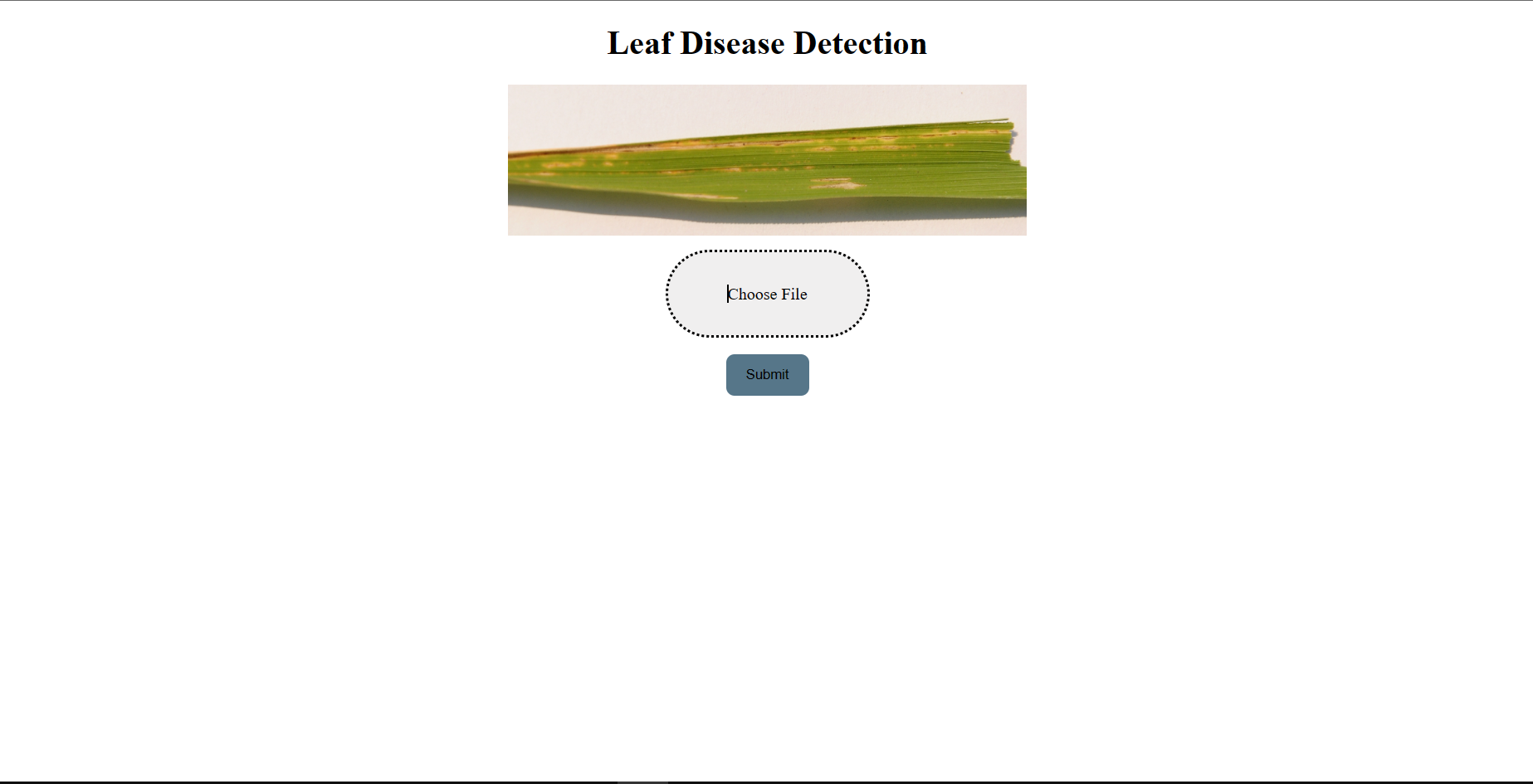
* Kết luận:

Việc sử dụng augmentation trong CNN sẽ giúp cải thiện đáng kể hiệu suất của mô hình khi tập dữ liệu gốc nhỏ, bằng cách tránh overfitting và tăng độ tổng quát hóa.

Mặt khác, nếu dữ liệu gốc đã đủ lớn và đa dạng, tác động của augmentation sẽ giảm, nhưng vẫn có thể cải thiện đôi chút hiệu suất của mô hình trên dữ liệu mới.

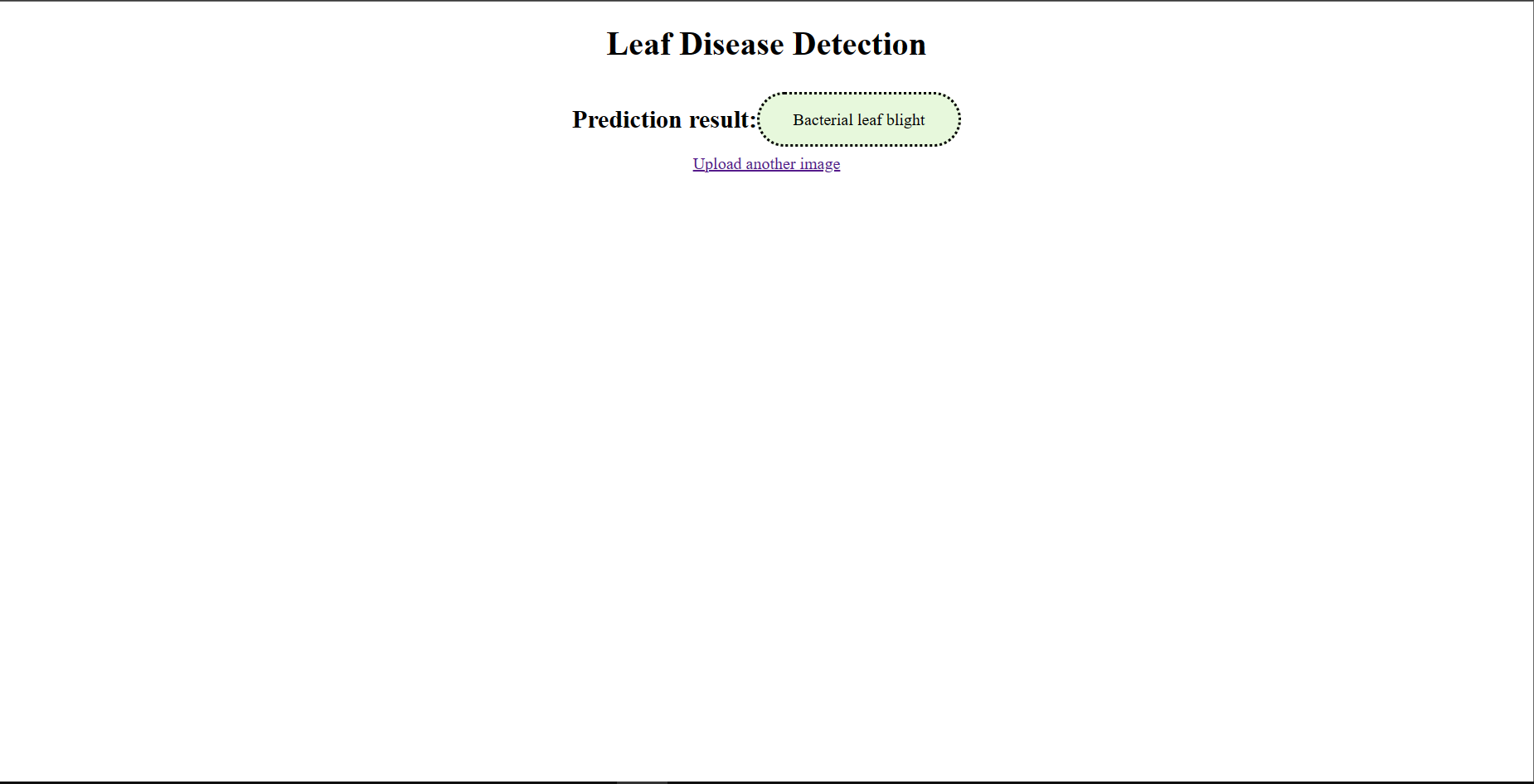
## **3 Ứng dụng thực tế và kết luận**

Bước 1: chạy file ‘app.py’

Bước 2: mở trình duyệt và truy cập đường dẫn ‘<http://127.0.0.1:5000>’

Hình 13. Giao diện web ở màn import

Bước 3: Import ảnh sau đó nhấn submit và xem kết quả dự đoán.



Hình 14. Giao diện web ở màn result

# **KẾT LUẬN**

Mô hình CNN trên đã cho thấy tiềm năng của việc áp dụng kĩ thuật khoa học trong phát triển kinh tế và mọi mặt trong đời sống nói chung, tự động hóa quy trình và giảm thiểu công sức là điều mà chúng ta luôn luôn hướng đến. Việc nghiên cứu sâu hơn có thể khám phá thêm những kiến trúc, mô hình khác nhau, cập nhật siêu trọng số và học tăng cường có thể cải thiện hơn nữa hiệu suất của mô hình.

Việc nghiên cứu và phát triển các mô hình học sâu đã giúp chúng em hiểu hơn về các thuật toán AI mạnh mẽ hiện nay, nắm được cơ chế vận hành và cấu trúc cốt lõi của nó.

Tuy nhiên, đó cũng là một thách thức lớn, khi em tìm hiểu một kỹ thuật mới cần đòi hỏi nhiều thời gian, công sức và tính tự giác trong công việc. Trong thời gian ngắn, chúng em chưa thể phát triển công trình lớn như kỳ vọng.

Trong thời gian tới, chúng em rất mong có được sự quan tâm, giúp đỡ từ phía nhà trường cùng các thầy cô hướng dẫn để mở rộng công trình nghiên cứu này và đưa nó đi vào thực tế.