TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN VĂN THẢO - 52000805**

**LÊ ĐÀO TẤN THÀNH - 52000804**

**CHÚ THÍCH HÌNH ẢNH CHÓ VÀ MÈO**

**BÁO CÁO CUỐI KÌ MÔN NHẬP MÔN THỊ GIÁC MÁY TÍNH**

**TÍNH TOÁN THÔNG MINH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN VĂN THẢO - 52000805**

**LÊ ĐÀO TẤN THÀNH - 52000804**

**CHÚ THÍCH HÌNH ẢNH CHÓ VÀ MÈO**

**BÁO CÁO CUỐI KÌ MÔN NHẬP MÔN THỊ GIÁC MÁY TÍNH**

**TÍNH TOÁN THÔNG MINH**

Người hướng dẫn

**TS. Phạm Văn Huy**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn thầy Phạm Văn Huy đã dạy em những kiến thức về thị giác máy tính qua đó giúp nhóm em hoàn thành bài báo cáo này.

*TP. Hồ Chí Minh, ngày 06 tháng 03 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Thảo*

*Trần Văn Thảo*

*Thành*

*Lê Đào Tấn Thành*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Phạm Văn Huy. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 06 tháng 03 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Thảo*

*Trần Văn Thảo*

*Thành*

*Lê Đào Tấn Thành*

**CHÚ THÍCH HÌNH ẢNH CHÓ VÀ MÈO**

**TÓM TẮT**

Báo cáo tập trung vào mô tả hình ảnh cho chó và mèo, sử dụng mô hình CNN và LSTM. Nói về lý thuyết, xây dựng mô hình, và ứng dụng thực tế trong xây dựng trang web thú cưng. Báo cáo đánh giá hiệu suất mô hình và đề xuất hướng phát triển tương lai. Kết hợp giữa lý thuyết và ứng dụng, mang lại giá trị trong lĩnh vực trí tuệ nhân tạo và xử lý hình ảnh.

**CHÚ THÍCH HÌNH ẢNH CHÓ VÀ MÈO**

**ABSTRACT**

Bài báo cáo này tập trung vào việc giải quyết vấn đề mô tả hình ảnh cho chó và mèo, với sự áp dụng của lý thuyết và phương pháp xây dựng mô hình hiệu quả. Bước đầu tiên của bài báo cáo là giới thiệu về lý do chọn đề tài, với nhấn mạnh vào sự quan trọng của mô tả hình ảnh trong bối cảnh dữ liệu ngày càng lớn và ngày càng quan trọng của trí tuệ nhân tạo. Phần lớn của báo cáo tập trung vào lý thuyết và kỹ thuật xây dựng mô hình. Giải thích chi tiết về sự lựa chọn kiến trúc mạng nơ-ron, đặc biệt là việc sử dụng mô hình CNN và cách áp dụng LSTM để giải quyết các thách thức trong xử lý chuỗi dữ liệu hình ảnh. Ứng dụng thực tế của mô hình được mô tả chi tiết trong báo cáo, với những ví dụ cụ thể về cách nó đã giúp tự động mô tả hình ảnh chó và mèo một cách chính xác và tự nhiên. Bài báo cáo cũng bao gồm những ứng dụng tiềm năng trong các lĩnh vực khác như giáo dục, quảng cáo thú cưng và trải nghiệm tương tác động vật ảo. Cuối cùng, bài báo cáo đánh giá sự hiệu quả của mô hình và đề xuất những hướng phát triển tiếp theo. Tổng cộng, nó là một báo cáo toàn diện với sự kết hợp giữa lý thuyết, kỹ thuật và ứng dụng thực tế, mang lại giá trị lớn trong lĩnh vực trí tuệ nhân tạo và xử lý hình ảnh.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vii](#_Toc677850654)

[DANH MỤC BẢNG BIỂU ix](#_Toc929706566)

[DANH MỤC CÁC CHỮ VIẾT TẮT x](#_Toc485334591)

[CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI 1](#_Toc1494229047)

[1.1 Lý do chọn đề tài 1](#_Toc2128486259)

[1.2 Mục tiêu thực hiện đề tài 1](#_Toc1730539442)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 2](#_Toc17407236)

[2.1 Mạng nơ-ron tích chập (CNN) 2](#_Toc310592063)

[2.1.1 Tổng quan về mạng nơ-ron tích chập 2](#_Toc1914295056)

[2.1.2 Xception 12](#_Toc1840356129)

[2.2 Mạng nơ-ron hồi quy (RNN) 16](#_Toc1856222575)

[2.2.1 Tổng quan về mạng nơ-ron hồi quy 16](#_Toc1581986035)

[2.2.2 Long Short-term Memory 19](#_Toc1616292495)

[CHƯƠNG 3. MÔ HÌNH ĐỀ XUẤT 22](#_Toc1628710794)

[3.1 Xây dụng mô hình kết hợp CNN - LSTM 22](#_Toc1570252581)

[3.1.1 Tổng quan 22](#_Toc1668380377)

[3.1.2 Kiến trúc 22](#_Toc1978170634)

[CHƯƠNG 4. THỰC NGHIỆM 23](#_Toc1206561197)

[4.1 Dữ liệu thực nghiệm 23](#_Toc177053541)

[4.2 Cài đặt thực nghiệm 24](#_Toc2015374167)

[4.2.1 Cài đặt mô hình cho đề tài 24](#_Toc499608898)

[4.2.2 Triển khai ứng dựng thực tế 26](#_Toc1263538175)

[CHƯƠNG 5. KẾT LUẬN 26](#_Toc967633872)

[5.1 Kết luận 26](#_Toc2092881408)

[5.2 Hướng phát triển 26](#_Toc737766889)

[TÀI LIỆU THAM KHẢO 28](#_Toc260498628)

# DANH MỤC HÌNH VẼ

[Hình 1. 1: Kiến trúc cơ bản của CNN 2](#_Toc2116647894)

[Hình 1. 2: Các activation function 3](#_Toc550252858)

[Hình 1. 3: Các bộ lọc 4](#_Toc1378501563)

[Hình 1. 4: (a) bước nhảy là 1, (b) bước nhảy là 2 5](#_Toc1241376822)

[Hình 1. 5: Các bộ lọc 6](#_Toc1677407402)

[Hình 1. 6: Tích chập 7](#_Toc902837494)

[Hình 1. 8: Trượt 8](#_Toc320280906)

[Hình 1. 9: Đệm 8](#_Toc1561030284)

[Hình 1. 10: Tích chập nhiều kênh màu 9](#_Toc665443211)

[Hình 1. 11: Tích chập nhiều bộ lọc 9](#_Toc1679159326)

[Hình 1. 12: Tích chập hợp nhất 10](#_Toc140944762)

[Hình 1. 13: Lớp tích chập hoàn chỉnh 10](#_Toc37908866)

[Hình 1. 14: Lớp gộp 11](#_Toc729644740)

[Hình 1. 15: Lớp kết nối đầy đủ 11](#_Toc1354658406)

[Hình 1. 16: Kiến trúc mạng Xception 13](#_Toc1502007400)

[Hình 1. 17: Lớp tích chập tiêu chuẩn và phân tách theo chiều sâu 13](#_Toc522977164)

[Hình 1. 18: Khối dư 14](#_Toc840759223)

[Hình 2. 1: Kiến trúc cơ bản của RNN 16](#_Toc860404092)

[Hình 2. 2: Các loại trong RNN 18](#_Toc402606219)

[Hình 2. 3: Kiến trúc của LSTM 20](#_Toc1369516411)

[Hình 3. 1: Kiến trúc kiến hợp của CNN và LSTM 22](#_Toc1199316841)

[Hình 4. 1: Kiến trúc của LSTM sử dụng trong mô hình chung 24](#_Toc1126954770)

[Hình 4. 2: Đánh giá kết quả bằng BLEU score 25](#_Toc2140617611)

# DANH MỤC BẢNG BIỂU

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| CNN | Convolutional Neural Network |
| RNN | Recurrent Neural Network |
| LSTM | Long Short-term Memory |
| GRU | Gated Recurrent Unit |
| BLEU | Bilingual Evaluation Understudy |

# MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài

Đề tài "Mô tả hình ảnh cho chó và mèo" được lựa chọn với những lý do đáng chú ý. Đầu tiên, với sự bùng nổ của dữ liệu hình ảnh và video trên Internet, khả năng tự động mô tả hình ảnh đặc biệt quan trọng để hiểu và tận dụng nguồn thông tin lớn này. Mô hình mô tả hình ảnh cho chó và mèo không chỉ hỗ trợ trong việc tự động hóa quy trình này mà còn đưa ra những thách thức mới đối với lĩnh vực trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên.

## Mục tiêu thực hiện đề tài

Mục tiêu của nghiên cứu là xây dựng một mô hình có khả năng mô tả chính xác và tự nhiên cho hình ảnh chó và mèo. Ngoài ra, việc tìm hiểu về cách mô hình có thể tự động học được đặc điểm quan trọng của chó và mèo từ hình ảnh giúp mở rộng kiến thức về học máy và trí tuệ nhân tạo. Điều này không chỉ mang lại giá trị nghiên cứu mà còn có ứng dụng rộng rãi trong nhiều lĩnh vực như giáo dục, quảng cáo thú cưng, và tạo ra các ứng dụng di động hoặc trang web hữu ích cho người dùng. Đề tài còn mở ra cơ hội mở rộng sang việc mô tả hình ảnh cho nhiều loài động vật khác, đặt ra những thách thức mới và đưa ra những đóng góp quan trọng cho cộng đồng nghiên cứu và ứng dụng công nghệ.

# CƠ SỞ LÝ THUYẾT

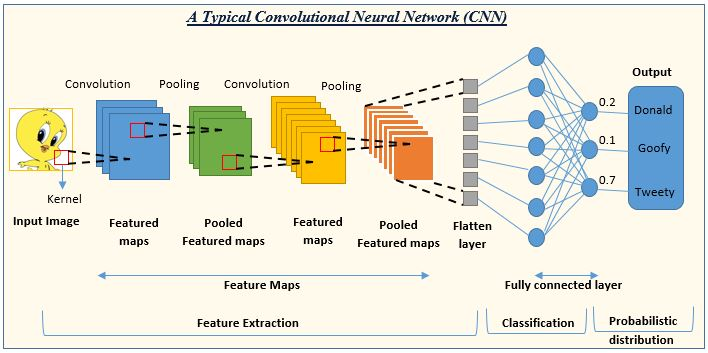
## Mạng nơ-ron tích chập (CNN)

### Tổng quan về mạng nơ-ron tích chập

#### Khái niệm

CNN là viết tắt của Convolutional Neural Network (Mạng nơ ron tích chập). Nó là một lớp mạng thần kinh sâu thường được áp dụng để phân tích hình ảnh trực quan. CNN đã đạt được hiệu quả tốt trong các bài toán như nhận dạng và phân loại hình ảnh.

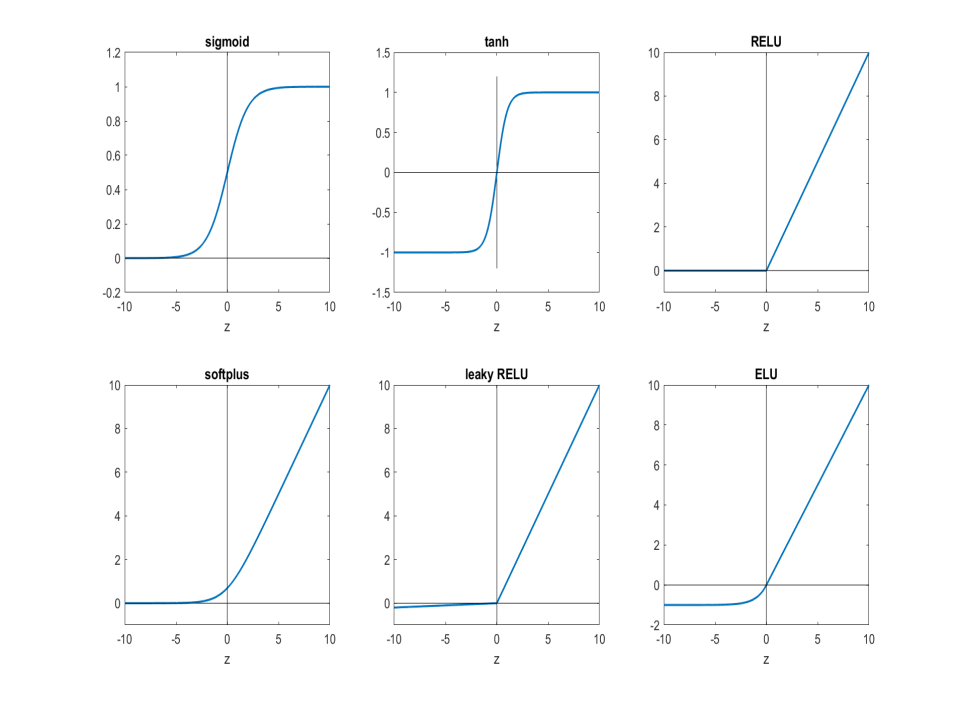
#### Sơ đồ hoạt động của CNN



Hình 1. 1: Kiến trúc cơ bản của CNN

(Nguồn: [link](https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/))

#### Hàm kích hoạt (Activation function)



Hình 1. 2: Các activation function

(Nguồn: [link](https://www.researchgate.net/figure/Common-activation-functions-in-ANN_fig3_314972496))

Sigmoid: Nén giá trị đầu vào thành 0 và 1 và hữu ích trong các bài toán phân loại có 2 lớp.

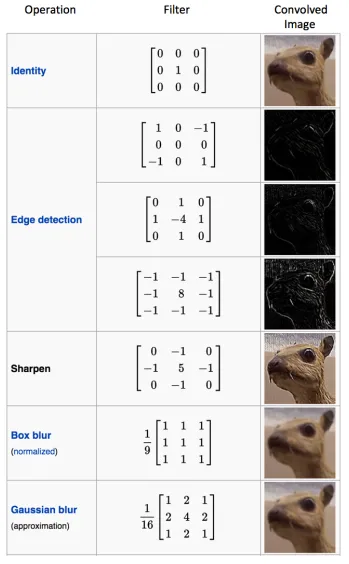
Tanh: Nén giá trị đầu vào giữa -1 và 1, giảm độ biến động của đầu vào, kiểm soát quá trình học và làm mô hình ổn định hơn.

Softmax: Sử dụng ở tầng output của mạng neural phục vụ cho các bài toán phân loại đa lớp.

ReLU (Rectified Linear Unit): Thay thế các giá trị âm thành 0, làm giảm lượng tham số cần học.

Leaky ReLu: Để giải quyết vấn đề “dying ReLU” thì Leaky ReLU không loại bỏ hết số âm mà giữ lại các số âm không đáng kể.

#### Bộ lọc (Filters)



Hình 1. 3: Các bộ lọc

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

Bộ lọc là ma trận trọng số được áp dụng cho dữ liệu đầu vào, trích xuất các đặc trưng, đặc biệt là trong xử lý ảnh, thông qua các phép toán tích chập.

Một số bộ lọc phổ biến được dùng trong CNN:

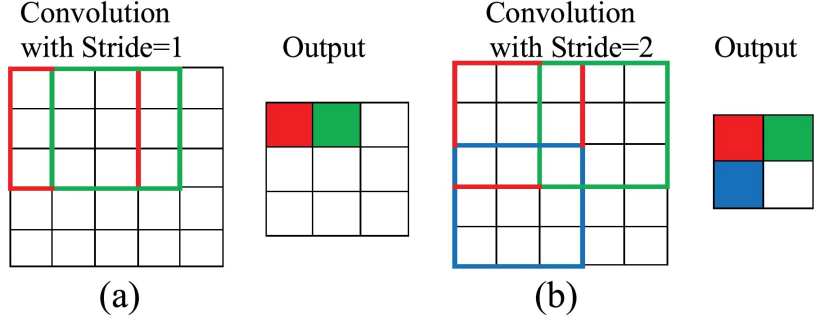
- Bộ lọc nhận dạng (Identity): Giữ nguyên các đặt trưng đầu vào.

- Bộ lọc phát hiện cạnh (Edge): Phát hiện các cạnh theo các hướng khác nhau.

- Bộ lọc làm mờ (Blur): Làm giảm nhiễu và mịn ảnh.

- Bộ lọc làm sắc nét (Sharpen): Tăng cường các góc cạnh và làm nổi bật chi tiết trong hình ảnh.

#### Bước nhảy (Stride)



Hình 1. 4: (a) bước nhảy là 1, (b) bước nhảy là 2

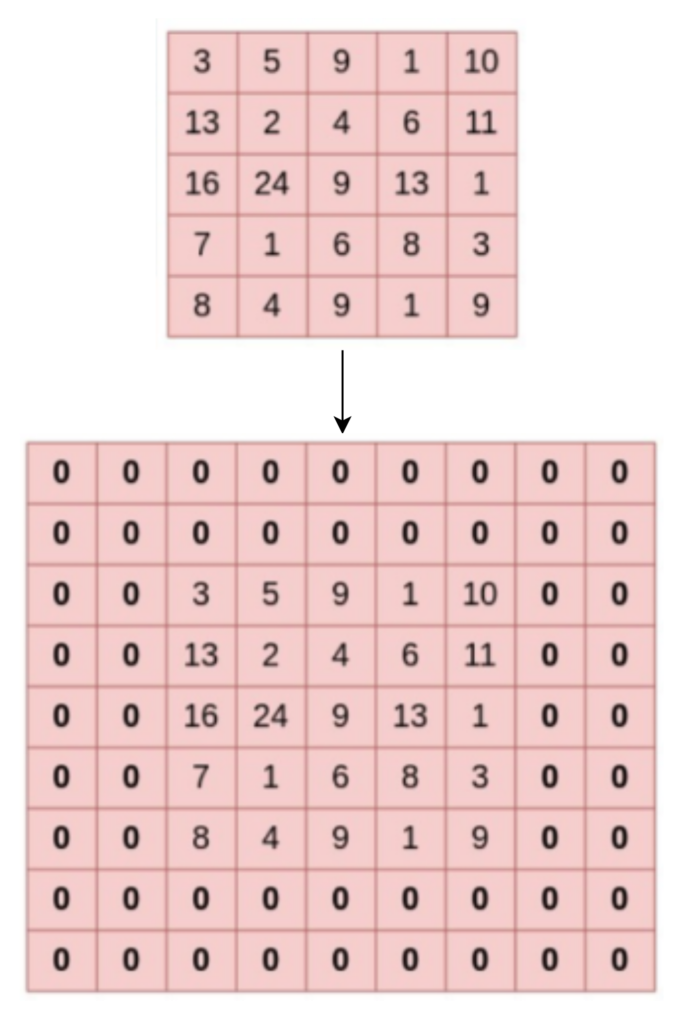
(Nguồn: [link](https://www.analyticsvidhya.com/blog/2022/03/basics-of-cnn-in-deep-learning/))

Stride xác định số bước trượt, áp dụng cho bộ lọc trượt trên ma trận đầu vào trong quá trình tích chập, cũng như áp dụng trong các lớp gộp.

Ưu điểm: Kiểm soát độ phân giải, tình trạng Overfitting, hiệu quả tính toán.

Nhược điểm: Giảm tính chi tiết, tăng độ phức tạp tính toán.

#### Đệm (Padding)



Hình 1. 5: Các bộ lọc

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

Đệm là thêm các pixel bổ sung xung quanh các cạnh ngoài của hình ảnh đầu vào hoặc feature map khi áp dụng thao tác tích chập.

Vai trò: Bảo toán tính chi tiết của hình anh, ngăn ngừa mất thông tin.

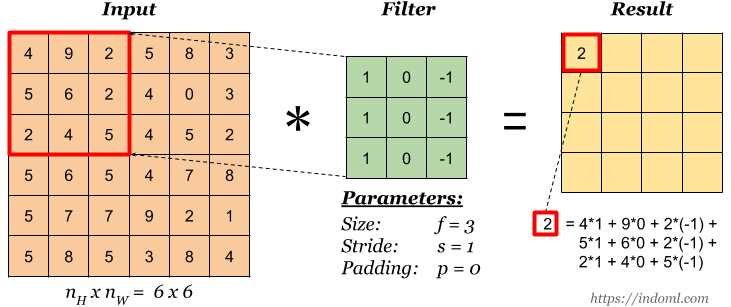
Được sử dụng trong CNN gồm:

- “Valid” padding: Không cần padding

- “Same” padding: để giữ nguyên kích thước đầu ra như kích thước của đầu vào trước khi tích chập.

#### Phép toán tích chập đơn giản (Basic Convolution Operation)

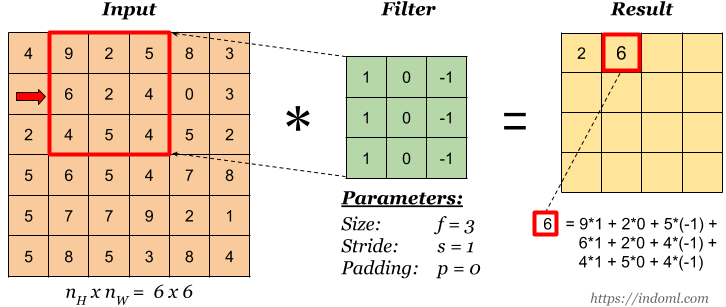
Bước 1: Đặt bộ lọc lên đầu vào, thực hiện phép nhân phần tử và cộng kết quả.



Hình 1. 6: Tích chập

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

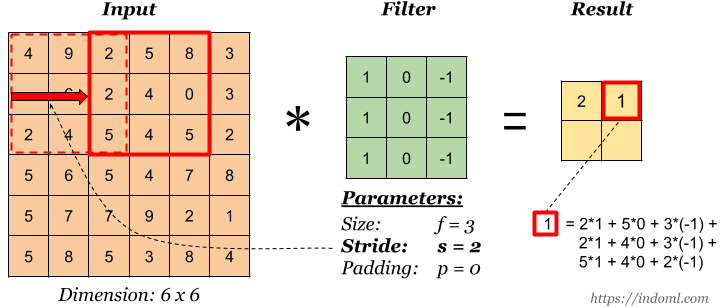
Bước 2: Trược dựa trên stride và tiếp tục thực hiện phép toán ở bước 1.



Hình 1. 7: Trượt

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

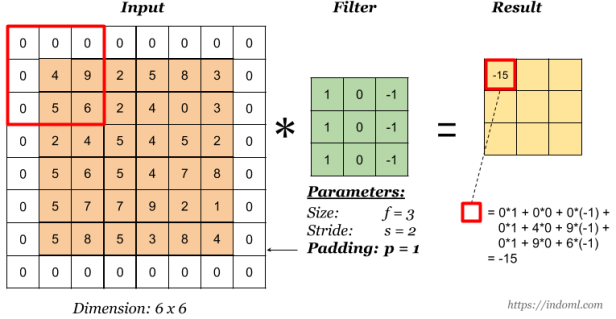
Stride - Basic Convolution Operation: Stride chi phối số lượng ô mà bộ lọc được di chuyển trong đầu vào để tính toán ô tiếp theo trong kết quả.



Hình 1. 8: Trượt

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

Padding - Basic Convolution Operation: Nó giúp chúng ta giữ được nhiều thông tin hơn ở viền của hình ảnh. Nếu không có phần đệm, rất ít giá trị ở lớp tiếp theo sẽ bị ảnh hưởng bởi các pixel làm cạnh của hình ảnh.

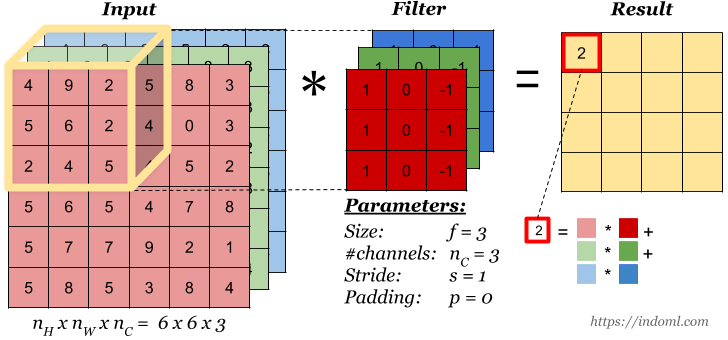


Hình 1. 9: Đệm

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

#### Tích chập nhiều kênh màu (Convolution Operation on Volume)

Khi đầu vào có nhiều kênh màu (ví dụ: hình ảnh RGB), bộ lọc phải có số kênh phù hợp. Để tính toán một ô đầu ra, hãy thực hiện tích chập trên từng kênh tương ứng, sau đó cộng kết quả lại với nhau.

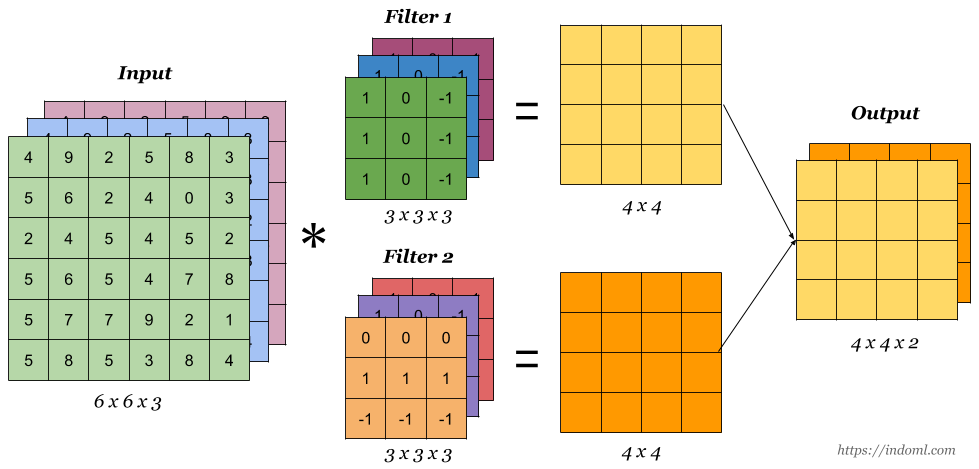


Hình 1. 10: Tích chập nhiều kênh màu

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

#### Tích chập với nhiều bộ lọc (Convolution Operation With Multiple Filters)

Nhiều bộ lọc có thể được sử dụng trong lớp tích chập để detect ra nhiều đặc trưng hơn. Đầu ra của lớp khi đó sẽ có số kênh bằng số lượng bộ lọc sử dụng trong lớp tích chập.

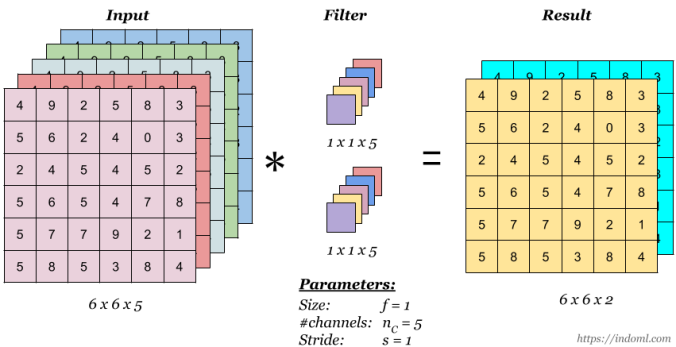


Hình 1. 11: Tích chập nhiều bộ lọc

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

#### Tích chập 1x1 (1x1 Convolution)

Đây là tích chập với bộ lọc 1 x 1. Hiệu quả là làm phẳng hoặc “hợp nhất” các kênh lại với nhau.

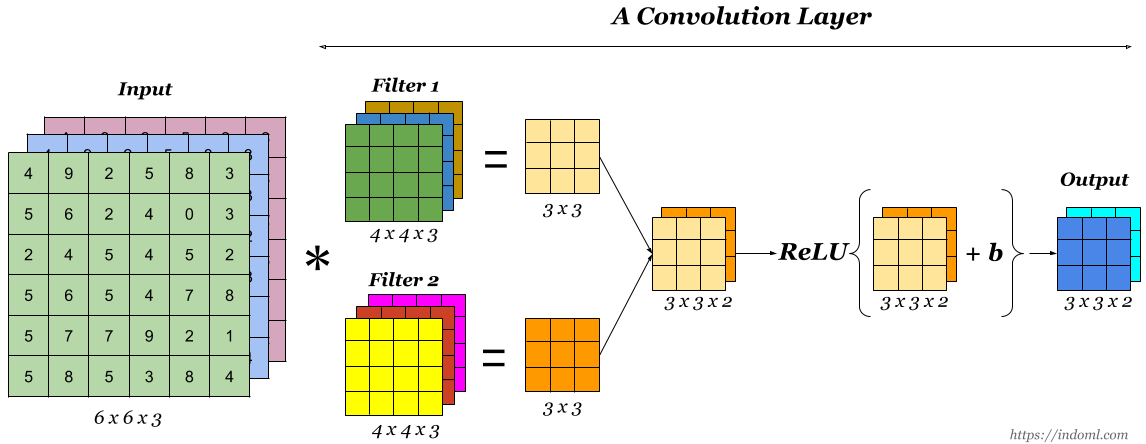


Hình 1. 12: Tích chập hợp nhất

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

#### Một lớp tích chập (One Convolution Layer)

Cuối cùng, để tạo thành lớp tích chập, độ lệch (ϵ R) được thêm vào và chức năng kích hoạt như ReLU hoặc tanh được áp dụng.



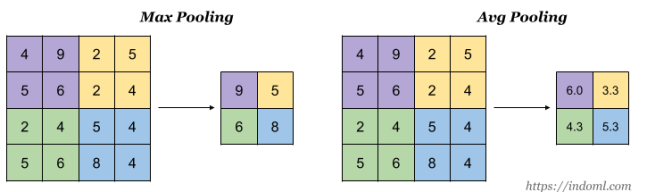
Hình 1. 13: Lớp tích chập hoàn chỉnh

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

#### Lớp gộp (Pooling Layer)

Lớp gộp được sử dụng để giảm kích thước của hình ảnh và tăng tốc độ tính toán, cũng như làm cho một số đặc trưng mà nó detect được trở nên mạnh mẽ hơn một chút.

Các loại gộp mẫu là gộp tối đa và gộp trung bình.

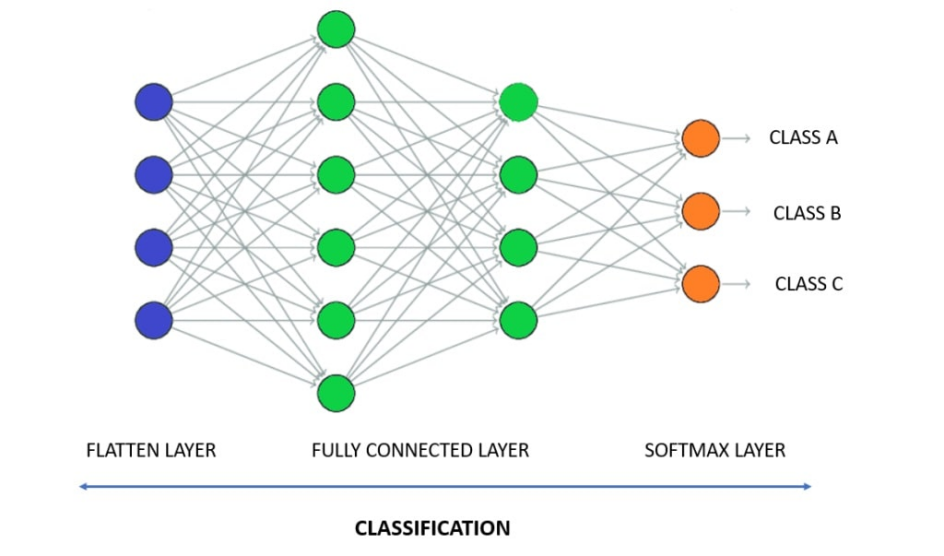


Hình 1. 14: Lớp gộp

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

#### Lớp kết nối đầy đủ (Fully Connected Layer)

Nằm ở cuối mạng. Trong phân loại hình ảnh, lớp cuối cùng có kích hoạt softmax cho phép đưa ra các quyết định các đặc trưng đã học.



Hình 1. 15: Lớp kết nối đầy đủ

(Nguồn: [link](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))

#### Ứng dụng

CNN có thể giải quyết các vấn đề liên quan đến dữ liệu liên tục, chẳng hạn như văn bản, âm thanh và video.

Chúng có thể được kết hợp với các mạng khác hoặc chuyển đổi dữ liệu chuỗi thành hình ảnh.

Các tác vụ cụ thể mà CNN có thể thực hiện trong lĩnh vực Thị giác máy tính bao gồm: Phân loại, Vị trí hóa, Phát hiện và Phân đoạn.

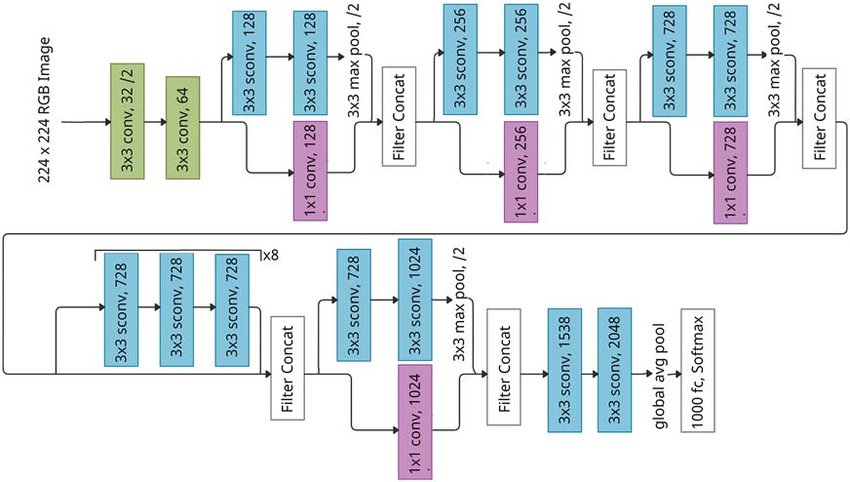
### Xception

#### Khái niệm

Mạng Xception là viết tắt của Extreme Inception (một biến thể nổi bật của mạng nơ-ron tích chập CNN). Nó là một kiến trúc mạng nơ-ron sâu được giới thiệu bởi François Chollet, một nhà nghiên cứu của Google Brain, vào năm 2017. Xception được thiết kế để cải thiện hiệu suất của các mạng nơ-ron sâu trong việc nhận diện ảnh và các nhiệm vụ liên quan đến xử lý hình ảnh.

#### Sơ đồ hoạt động của Xception

Xception là mạng nơ-ron được xây dựng trên cơ sở của kiến trúc Inception.



Hình 1. 16: Kiến trúc mạng Xception

(Nguồn: [link](https://www.researchgate.net/figure/Architecture-of-the-Xception-deep-CNN-model_fig2_351371226))

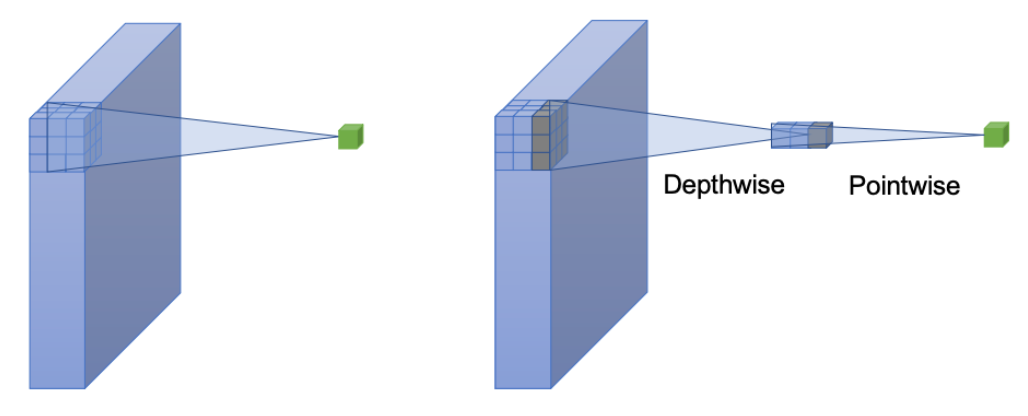
Khác biệt chính giữa Xception và Inception là ở việc Xception sử dụng các module depthwise separable convolution thay vì convolution thông thường như trong Inception.

Module depthwise separable convolution chia mỗi lớp convolution thành hai lớp riêng biệt: convolution theo chiều sâu (depthwise convolution) và convolution theo chiều rộng và chiều cao (pointwise convolution). Điều này giúp giảm số lượng trọng số cần học và tăng tính hiệu quả tính toán.

#### Lớp tích chập phân tách theo chiều sâu (Depthwise Separable Convolution)

Lớp này thực hiện convolution trên từng kênh độ sâu riêng biệt của dữ liệu đầu vào.

Sau đó, một lớp convolution 1x1 (gọi là pointwise convolution) được áp dụng để kết hợp thông tin từ các kênh đã được xử lý riêng biệt.

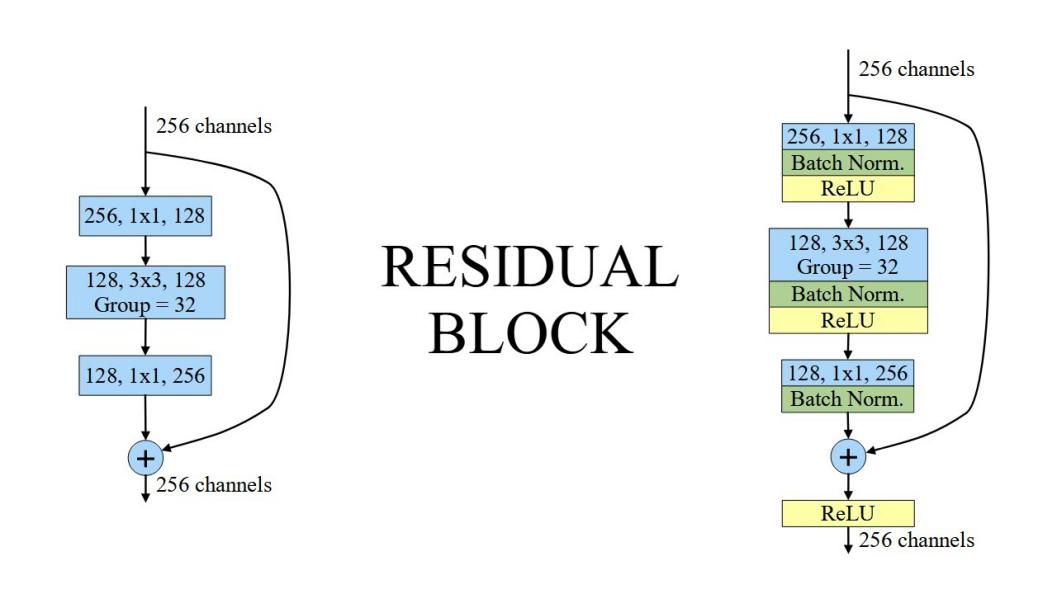


Hình 1. 17: Lớp tích chập tiêu chuẩn và phân tách theo chiều sâu

(Nguồn: [link](https://viblo.asia/p/separable-convolutions-toward-realtime-object-detection-applications-aWj534bpK6m))

#### Khối dư (Block Residual)

Mạng Xception thường sử dụng các khối residual (giống như ResNet) để giảm thiểu vấn đề về gradient vanishing và exploding trong quá trình huấn luyện mạng sâu.



Hình 1. 18: Khối dư

(Nguồn: [link](https://www.youtube.com/watch?v=r0HvOIjziw4))

#### Cải thiện của Xception so với Inception

Xception là một bước tiến đáng kể so với kiến trúc Inception, với các cải tiến đáng chú ý sau:

Depthwise Separable Convolution:

* Cải tiến chính nhất của Xception so với Inception là việc sử dụng các lớp depthwise separable convolution.
* Trong Inception, mỗi lớp convolution được áp dụng cho toàn bộ kênh đầu vào. Trong khi đó, Xception chia mỗi lớp convolution thành hai phần: depthwise convolution và pointwise convolution.
* Lớp depthwise convolution thực hiện convolution trên từng kênh độ sâu của dữ liệu đầu vào.
* Sau đó, lớp pointwise convolution thực hiện convolution trên không gian của dữ liệu đầu ra từ lớp depthwise convolution.
* Phương pháp này giúp giảm số lượng trọng số cần học và tăng tính hiệu quả tính toán, đồng thời giữ nguyên hoặc cải thiện hiệu suất so với Inception.

Giảm kích thước mô hình:

* Bằng cách sử dụng các lớp depthwise separable convolution, Xception giảm được số lượng trọng số cần học so với Inception.
* Điều này làm cho mô hình nhỏ gọn hơn và dễ huấn luyện hơn, đồng thời giảm bớt áp lực về tài nguyên tính toán.

Tính hiệu quả tính toán:

* Nhờ sự phân rã lớp convolution thành hai phần riêng biệt, Xception giúp giảm đi số lượng phép tính so với Inception.
* Điều này dẫn đến tăng hiệu suất tính toán và giảm thời gian huấn luyện mô hình.

Tính linh hoạt:

* Xception có thể được áp dụng cho nhiều loại dữ liệu đầu vào và nhiều nhiệm vụ khác nhau trong xử lý hình ảnh.
* Kiến trúc linh hoạt của nó cho phép tích hợp vào các ứng dụng cụ thể và điều chỉnh cho phù hợp với yêu cầu cụ thể của dự án.

#### Ứng dụng

Xception thường được sử dụng cho các nhiệm vụ nhận diện hình ảnh, phân loại hình ảnh, phát hiện đối tượng và nhiều tác vụ liên quan đến xử lý hình ảnh.

Đặc biệt, trong các bài toán yêu cầu độ chính xác cao và tài nguyên tính toán hạn chế, Xception thường là một lựa chọn phổ biến.

## Mạng nơ-ron hồi quy (RNN)

### Tổng quan về mạng nơ-ron hồi quy

#### Khái niệm

Mạng nơ-ron hồi quy (RNN) là một loại kiến trúc mạng nơ-ron được thiết kế để xử lý dữ liệu có tính chất tuần tự hoặc có mối quan hệ thời gian. RNN có khả năng làm việc với dữ liệu đầu vào và đầu ra có kích thước thay đổi, phù hợp cho nhiều ứng dụng như xử lý ngôn ngữ tự nhiên, dự đoán chuỗi thời gian, và nhiều tác vụ khác.

#### Sơ đồ hoạt động của RNN

Hình 2.1 cho thấy sơ đồ của 1 RNN



Hình 2. 1: Kiến trúc cơ bản của RNN

(Nguồn: <https://aws.amazon.com/vi/what-is/recurrent-neural-network/> )

RNN được tạo thành từ các nơ-ron: các nút xử lý dữ liệu kết hợp cùng nhau để thực hiện các tác vụ phức tạp. Các nơ-ron được tổ chức dưới dạng lớp đầu vào, đầu ra và ẩn. Lớp đầu vào nhận thông tin để xử lý và lớp đầu ra cung cấp kết quả. Quá trình xử lý dữ liệu, phân tích và dự đoán diễn ra trong lớp ẩn.

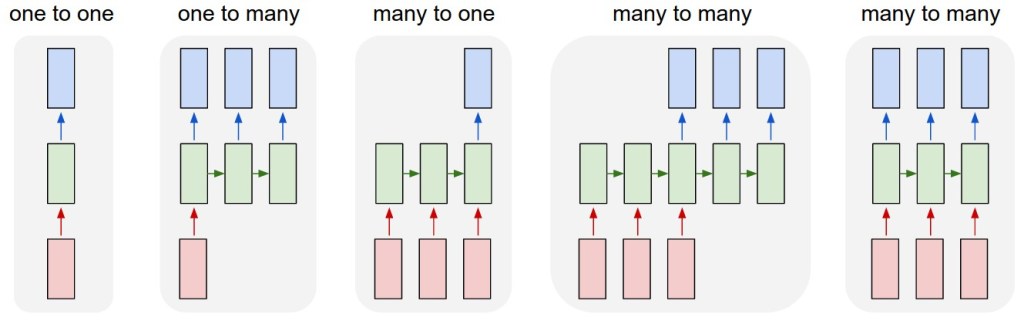
#### Lớp ẩn trong RNN

RNN hoạt động bằng cách truyền dữ liệu tuần tự qua các lớp ẩn, nhưng cũng tích hợp quy trình lặp lại hay hồi quy. Quy trình này cho phép RNN ghi nhớ thông tin từ các đầu vào trước đó và sử dụng nó để dự đoán các phần tiếp theo của chuỗi.

Ví dụ, xét chuỗi: "The cat is sitting on the \_\_\_" (Con mèo đang ngồi trên cái gì đó). Nếu RNN đã học qua đầu vào "The cat is sitting on," lớp ẩn sẽ lưu trữ thông tin về việc mèo đang ngồi. Khi nhận được từ tiếp theo, ví dụ như "mat," RNN sử dụng thông tin đã lưu trữ để hiểu rằng câu đầy đủ là "The cat is sitting on the mat" (Con mèo đang ngồi trên chiếc thảm). Sau đó, nó có thể dự đoán từ tiếp theo hoặc thậm chí hoàn thành câu.

#### Các loại trong mạng nơ-ron hồi quy

RNN thường có đặc trưng là kiến trúc một-một, nghĩa là một chuỗi đầu vào được liên kết với một đầu ra. Tuy nhiên, có khả năng linh hoạt trong việc điều chỉnh cấu trúc để đáp ứng các mục đích cụ thể. Hình 2. 2 cho thấy các loại có trong RNN:



Hình 2. 2: Các loại trong RNN

(Nguồn: <https://nttuan8.com/bai-13-recurrent-neural-network/> )

Một-nhiều (One-to-Many): Loại RNN này dẫn một đầu vào đến một số đầu ra. Thích hợp cho các ứng dụng như chú thích hình ảnh, nơi một câu có thể được tạo ra từ một từ khóa duy nhất.

Nhiều-nhiều (Many-to-Many): Mô hình sử dụng nhiều đầu vào để dự đoán nhiều đầu ra.

Nhiều-một (Many-to-One): Một số đầu vào được ánh xạ đến một đầu ra. Hữu ích trong các ứng dụng như phân tích cảm xúc, nơi mô hình dự đoán cảm xúc từ lời chứng thực đầu vào, chẳng hạn như xác định tích cực, tiêu cực hay trung lập.

#### Vấn đề Vanishing Gradient và Exploding Gradient

Hiểu quả của RNN trong việc học mối quan hệ dài hạn có thể bị ảnh hưởng bởi vấn đề Vanishing Gradient và Exploding Gradient. Đối với Vanishing Gradient, đây là hiện tượng khi hàm kích hoạt (như tanh hoặc sigmoid) giới hạn kết quả đầu ra trong một khoảng nhỏ như [-1, 1] hoặc [0, 1], đồng nghĩa với việc đạo hàm của nó chỉ nằm trong khoảng [0, 1] (đối với sigmoid, khoảng này có thể là [0, 0.25]). Khi ta nhân các giá trị nhỏ nhau trong ma trận, đạo hàm sẽ giảm đáng kể sau mỗi bước nhân, dẫn đến việc đạo hàm "biến mất" và không truyền được thông tin xa.

Vấn đề này có thể làm cho RNN không thể học các mối quan hệ dài hạn, vì các đối tượng xa nhau trong chuỗi sẽ không còn ảnh hưởng đủ lớn để được học. Vanishing Gradient không chỉ áp dụng cho RNN mà còn cho các mạng neural truyền thống với nhiều lớp.

Ngược lại, Exploding Gradient là hiện tượng khi giá trị của đạo hàm trở nên rất lớn, thậm chí vô hạn. Điều này có thể dẫn đến vấn đề khi các giá trị trọng số tăng lên quá nhanh, gây ra sự không ổn định và khó kiểm soát.

Mặc dù cả hai vấn đề này đều có thể ảnh hưởng đến hiệu suất của mô hình, nhưng Vanishing Gradient thường được đề cập đến nhiều hơn trong thực tế, có lẽ vì nó thường khó phát hiện và kiểm soát Exploding Gradient dễ hơn thông qua kỹ thuật như Gradient Clipping.

Trong thực tế, nhiều phương pháp đã được đề xuất để giải quyết vấn đề Vanishing Gradient, bao gồm sử dụng hàm kích hoạt khác như ReLU, Leaky ReLU, hay sử dụng các kiến trúc mạng như LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Unit) giúp giảm thiểu vấn đề này.

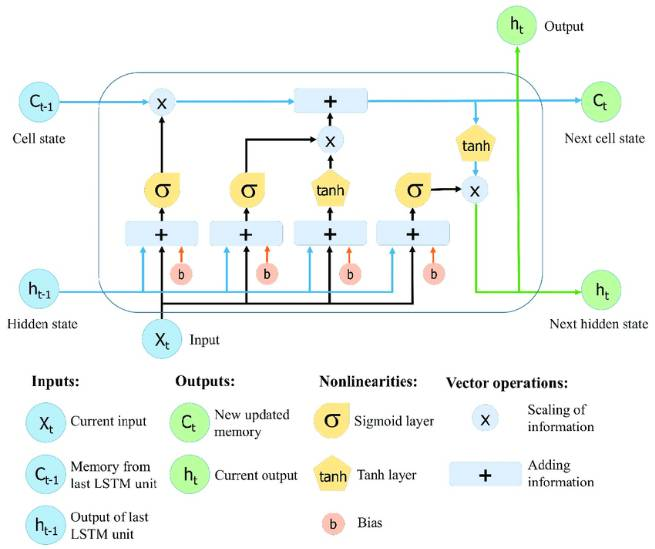
### Long Short-term Memory

#### Khái niệm

LSTM là một kiến trúc mạng nơ-ron đặc biệt, được thiết kế để xử lý và giữ thông tin trong thời gian. Trong RNN truyền thống, có khả năng xảy ra hiện tượng "vanishing gradient," nơi các gradient giảm đi rất nhanh khi lan truyền ngược qua các lớp, làm cho mô hình khó học được các phụ thuộc dài hạn. LSTM giữ lại thông tin qua các "cổng" có thể kiểm soát được, giúp mô hình học và lưu trữ thông tin theo cách hiệu quả hơn.

#### Sơ đồ hoạt động của LSTM

Hình 2.3 cho thấy sơ đồ của 1 LSTM:



Hình 2. 3: Kiến trúc của LSTM

(Nguồn: <https://www.researchgate.net/figure/The-structure-of-the-Long-Short-Term-Memory-LSTM-neural-network-Reproduced-from-Yan_fig8_334268507> )

Khi nhận đầu vào, LSTM xử lý dữ liệu hiện tại (Xt) và trạng thái ẩn từ đơn vị LSTM trước đó (ht-1).

Cổng quên (Forget Gate): Sử dụng hàm sigmoid, LSTM quyết định thông tin nào từ trạng thái ô nhớ trước đó (Ct-1) sẽ được giữ lại hoặc quên đi. Cổng quên giúp kiểm soát lượng thông tin cần giữ lại và thông tin nào cần bị loại bỏ.

Cổng đầu vào (Input Gate): LSTM quyết định thông tin nào từ đầu vào hiện tại (Xt) sẽ được lưu vào trạng thái ô nhớ. Sử dụng hàm sigmoid và hàm tanh, cổng đầu vào kiểm soát việc cập nhật thông tin mới vào trạng thái ô nhớ.

Cập nhật trạng thái ô nhớ (Cell State): Dựa trên kết quả từ cổng quên và cổng đầu vào, LSTM cập nhật trạng thái ô nhớ hiện tại (Ct). Quá trình này giúp ô nhớ của mô hình tổ chức và lưu trữ thông tin theo thời gian.

Cổng đầu ra (Output Gate): Cuối cùng, LSTM quyết định thông tin nào từ trạng thái ô nhớ hiện tại (Ct) sẽ được đưa ra dưới dạng đầu ra hiện tại (ht). Cổng đầu ra sử dụng hàm sigmoid để quyết định độ quan trọng của thông tin trong trạng thái ô nhớ và hàm tanh để tạo ra đầu ra.

Tổng cộng, các cổng và trạng thái ô nhớ trong LSTM tạo ra một cơ chế kiểm soát mạnh mẽ, giúp mô hình xử lý hiệu quả thông tin theo thời gian và tránh được vấn đề vanishing gradient trong quá trình huấn luyện.

#### Điểm Cải Tiến Từ RNN:

#### Cổng Quên (Forget Gate): LSTM sử dụng cổng quên để quyết định thông tin nào nên được giữ lại và thông tin nào nên bị quên đi từ quá khứ. Cổng quên giúp mô hình quyết định xem thông tin nào quan trọng và cần được giữ lại.

#### Cổng Nhớ (Memory Cell): LSTM có một ô nhớ chính để lưu trữ thông tin. Cổng này cho phép cập nhật giá trị của ô nhớ dựa trên các thông tin mới và thông tin cũ.

#### Cổng Đầu Ra (Output Gate): Cổng này quyết định thông tin nào sẽ trở thành đầu ra của LSTM. Nó đảm bảo rằng chỉ những thông tin quan trọng nhất sẽ được truyền đi.

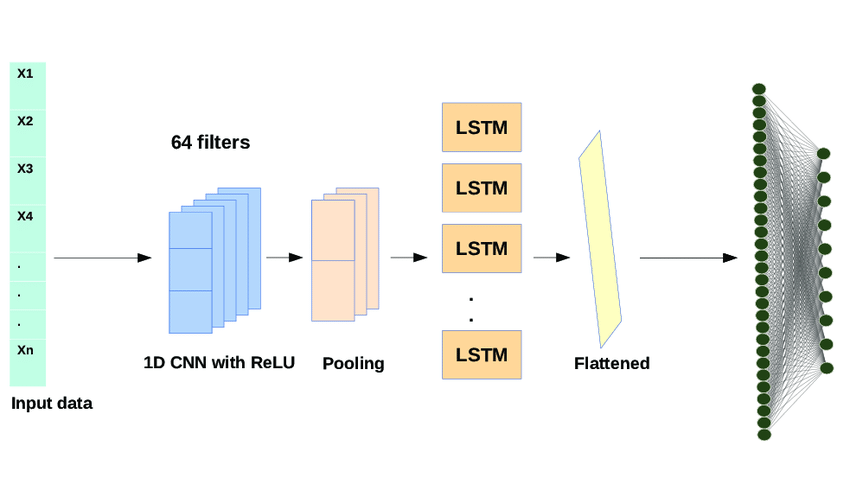
# MÔ HÌNH ĐỀ XUẤT

## Xây dụng mô hình kết hợp CNN - LSTM

### Tổng quan

Mô hình kết hợp CNN - LSTM được thiết kế để hiểu và tạo mô tả cho hình ảnh. Kết hợp giữa Convolutional Neural Network (CNN) và Long Short-Term Memory (LSTM), mô hình này có khả năng trích xuất đặc trưng hình ảnh thông qua CNN và sau đó sử dụng LSTM để xử lý mô tả chuỗi liên quan.

### Kiến trúc



Hình 3. 1: Kiến trúc kiến hợp của CNN và LSTM

(Nguồn: [Joint Learning of Temporal Models to Handle Imbalanced Data for Human Activity Recognition](https://www.researchgate.net/publication/343341551_Joint_Learning_of_Temporal_Models_to_Handle_Imbalanced_Data_for_Human_Activity_Recognition) )

Các bước trong quá trình xây dựng mô hình kết hợp CNN - LSTM từ hình 3.1:

* Trích xuất đặc trưng hình ảnh (CNN): Sử dụng mạng CNN để nhận diện và trích xuất các đặc trưng chính từ hình ảnh. Áp dụng lớp Convolutional và MaxPooling để tạo ra biểu diễn số hóa của đặc trưng hình ảnh.
* Xử lý mô tả chuỗi (LSTM): Sử dụng mô hình LSTM, một kiến trúc chuỗi thời gian, để xử lý dữ liệu chuỗi mô tả ngôn ngữ tự nhiên. Sử dụng lớp Embedding để nhúng từ vựng và lớp LSTM để hiểu cấu trúc và mối quan hệ giữa các từ trong mô tả.
* Kết hợp đặc trưng và chuỗi: Ghép đầu ra của mô hình CNN và LSTM để tạo ra biểu diễn tổng hợp của đặc trưng. Quá trình này giúp mô hình có cái nhìn tổng thể về nội dung của cả hình ảnh và mô tả.
* Huấn luyện tổng hợp: Huấn luyện mô hình bằng cách điều chỉnh trọng số của cả hai thành phần, CNN và LSTM, dựa trên dữ liệu đầu vào. Mục tiêu là học được các mối quan hệ phức tạp giữa hình ảnh và mô tả, tạo ra một biểu diễn chung mạnh mẽ.

# THỰC NGHIỆM

## Dữ liệu thực nghiệm

Tên bộ dữ liệu: CocoDS

* Nguồn gốc: Kaggle dataset
* Loại dữ liệu: Bao gồm ảnh của chó và mèo, kèm theo mô tả về hành động của chúng.
* Kích thước: 83 MB
* Định dạng: PNG, JPG
* Số lượng mẫu: 8,639
* Số lượng thuộc tính: 3 (image, caption, keypoint)

Chia tập dữ liệu:

* Phương pháp chia tập dữ liệu: Chia thành hai phần bình thường.
* Tỷ lệ chia tập: 8,192 mẫu được sử dụng để huấn luyện và 348 mẫu được sử dụng để kiểm thử.

## Cài đặt thực nghiệm

### Cài đặt mô hình cho đề tài

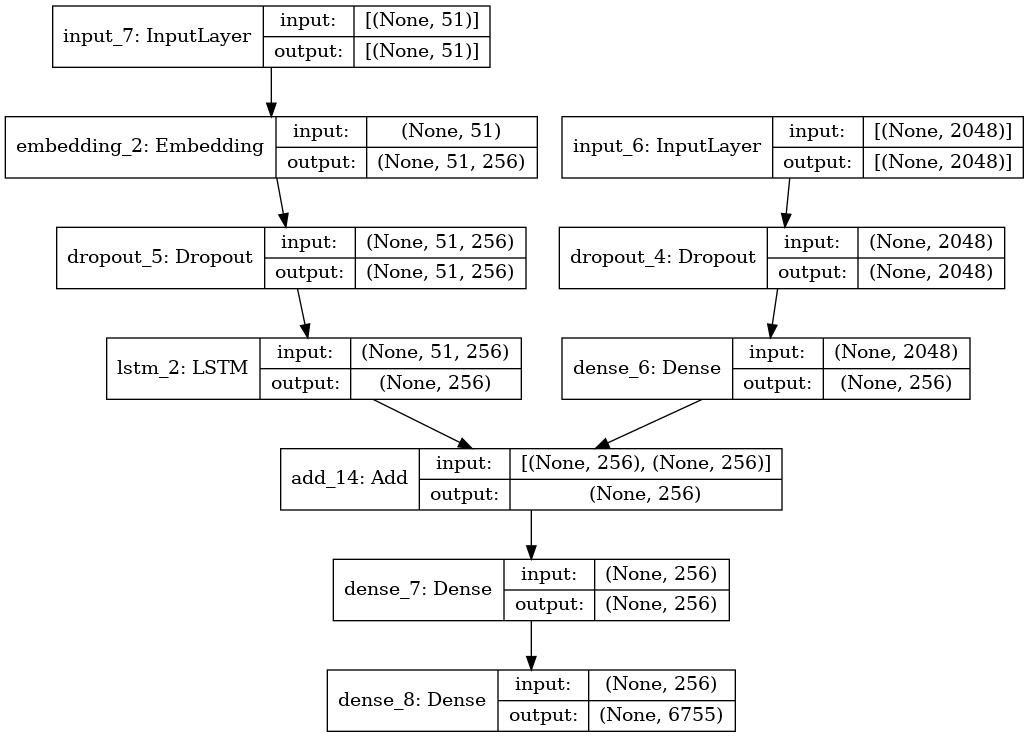
#### Chuẩn bị Môi trường và cài đặt thư viện

Sử dụng Google Colab và Python 3.10.4, cài đặt thư viện TensorFlow và các thư viện quan trọng khác như pandas, numpy,...

#### Xây dựng mô hình

Xception: được sử dụng từ thư viện Tensorflow

LSTM: Tiến hành xây dựng và kiến trúc của nó như sau:

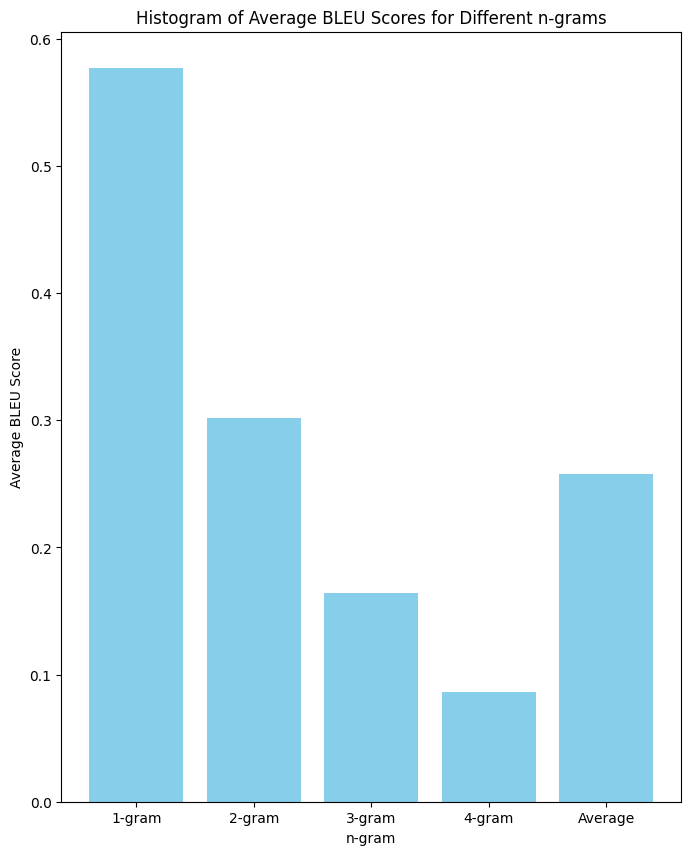


Hình 4. 1: Kiến trúc của LSTM sử dụng trong mô hình chung

#### Đánh giá Mô hình

Đánh giá mô hình được thực hiện bằng cách sử dụng độ đo BLEU score trên tập kiểm thử.

Dưới đây là kết quả đánh giá của mô hình:



Hình 4. 2: Đánh giá kết quả bằng BLEU score

Tổng kết chung, mô hình đánh giá sử dụng BLEU score cho công tác tạo chú thích ảnh của chúng ta đã cho thấy những kết quả đáng chú ý. Trong quá trình đánh giá, mô hình thể hiện khả năng tốt trong việc dự đoán từ đơn (1-gram), với điểm BLEU đáng kể là 0.577, chỉ ra sự chính xác trong việc dự đoán từng từ một. Tuy nhiên, mô hình gặp khó khăn khi dự đoán các cặp từ liên tiếp (2-gram), chuỗi ba từ (3-gram) và chuỗi bốn từ (4-gram), với các BLEU score lần lượt là 0.302, 0.165, và 0.087.

Đặc biệt, điểm BLEU trung bình (0.258) là một chỉ số tổng thể, cho thấy mức hiệu suất trung bình của mô hình trong việc tạo chú thích ảnh. Để cải thiện mô hình, chúng ta cần tập trung vào việc nâng cao khả năng hiểu ngữ cảnh và mối quan hệ giữa các từ, đồng thời tối ưu hóa khả năng dự đoán chuỗi từ dài hạn.

### Triển khai ứng dựng thực tế

# KẾT LUẬN

## Kết luận

Trong đề tài "Mô tả hình ảnh cho chó và mèo", chúng em đã đặt ra mục tiêu xây dựng một mô hình CNN - LSTM có khả năng mô tả chính xác và tự nhiên cho hình ảnh chó và mèo. Nghiên cứu này có ý nghĩa lớn trong bối cảnh tăng cường khả năng hiểu và tận dụng thông tin từ dữ liệu hình ảnh ngày càng lớn trên Internet. Qua quá trình nghiên cứu, chúng em đã áp dụng các kỹ thuật xử lý ngôn ngữ tự nhiên và học máy để đạt được mục tiêu đề ra. Kết quả của đề tài không chỉ giúp tự động hóa quy trình mô tả hình ảnh mà còn mang lại những đóng góp quan trọng cho lĩnh vực trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên.

## Hướng phát triển

Để mở rộng và cải thiện đề tài, có một số hướng phát triển có thể được xem xét:

* Mở rộng Loại Động vật: Nghiên cứu có thể được mở rộng để mô tả hình ảnh cho nhiều loài động vật khác nhau, tạo ra một hệ thống đa dạng và đa loài.
* Tích hợp Nguyên tắc Học Sâu: Áp dụng các nguyên tắc học sâu để cải thiện khả năng tự động học và hiểu sâu sắc hơn về đặc trưng của chó và mèo từ dữ liệu hình ảnh.
* Phân loại Tình cảm Động vật: Nghiên cứu mở rộng sang việc nhận diện tình cảm của động vật từ hình ảnh, tạo ra khả năng mô tả thêm về trạng thái tinh thần của chúng.

Những hướng phát triển này có thể giúp đưa đề tài vào một chiều sâu và ứng dụng rộng rãi hơn trong cả lĩnh vực nghiên cứu và thực tế.

# TÀI LIỆU THAM KHẢO

Tiếng Anh

https://www.analyticsvidhya.com/blog/2021/12/step-by-step-guide-to-build-image-caption-generator-using-deep-learning/

https://www.researchgate.net/profile/Jameer-Kotwal/publication/377852268\_IMAGE\_CAPTION\_GENERATOR\_USING\_CONVOLUTIONAL\_NEURAL\_NETWORKS\_AND\_LONG\_SHORT-TERM\_MEMORY/links/65bb6cdb1e1ec12eff684fd8/IMAGE-CAPTION-GENERATOR-USING-CONVOLUTIONAL-NEURAL-NETWORKS-AND-LONG-SHORT-TERM-MEMORY.pdf

https://www.ijitee.org/wp-content/uploads/papers/v10i3/C83830110321.pdf