TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue and white logo

Description automatically generated with low confidence

**BÀI TẬP LỚN**

HỌC PHẦN: HỌC SÂU VÀ ỨNG DỤNG

**ĐỀ TÀI: PHÂN LOẠI GIÀY – SANDAL – BOOT BẰNG PHƯƠNG PHÁP CONVOLUTIONAL NEURAL NETWORK (CNN)**

Giáo viên hướng dẫn: Nguyễn Thị Kim Ngân

Sinh viên/nhóm sinh viên thực hiện:

1. Trần Thanh Thảo, lớp 62TH-NB

2. Nguyễn Hà Linh, lớp 63CNTT2

**Hà Nội, tháng 6 năm 2023**

**MỤC LỤC**

[**Phần 1: Lý thuyết** 3](#_Toc137333787)

[***1.*** ***Giới thiệu*** 3](#_Toc137333788)

[***2.*** ***Lớp tích chập – Convolution Layer*** 4](#_Toc137333789)

[***3.*** ***Bước nhảy – Stride*** 5](#_Toc137333790)

[***4.*** ***Đường viền – Padding*** 6](#_Toc137333791)

[***5.*** ***Các kiểu tầng*** 6](#_Toc137333792)

[5.1 **Tầng tích chập (CONV)** 6](#_Toc137333793)

[5.2 **Pooling (POOL)** 6](#_Toc137333794)

[5.3 Fully Connected (FC) 7](#_Toc137333795)

[***6.*** ***Các hàm kích hoạt thường gặp*** 7](#_Toc137333796)

[6.1 Rectified Linear Unit (ReLu) 7](#_Toc137333797)

[**6.2 Softmax** 8](#_Toc137333798)

[**Phần 2: Ứng dụng trong thực tế** 9](#_Toc137333799)

[***1.*** ***Mô tả bài toán*** 9](#_Toc137333800)

[***2.*** ***Mô tả tập dữ liệu của bài toán*** 10](#_Toc137333801)

[***3.*** ***Thiết kế mô hình CNN*** 11](#_Toc137333802)

[***4.*** ***Kết quả thực nghiệm*** 14](#_Toc137333803)

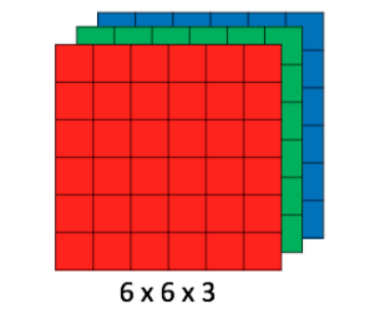
[**Kết luận** 15](#_Toc137333804)

[**Tài liệu tham khảo** 15](#_Toc137333805)

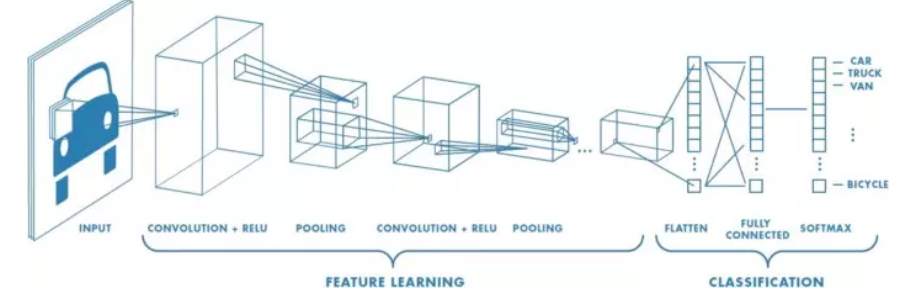
# **Phần 1: Lý thuyết**

# ***Giới thiệu***

* Trong mạng neural, mô hình mạng neural tích chập (CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là 1 trong số những lĩnh vực mà CNN được sử dụng rộng rãi.
* CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao, W: Chiều rộng, D: Độ dày). Ví dụ: Hình ảnh là mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).

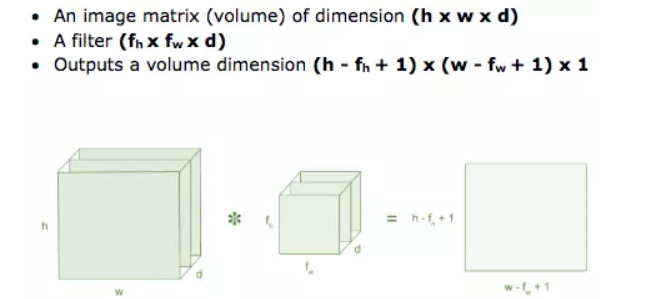


* Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernals), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.

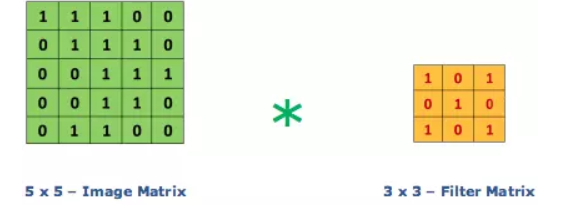


# ***Lớp tích chập – Convolution Layer***

* Tích chập là lớp đầu tiên để trích xuất các tính năng từ hình ảnh đầu vào. Tích chập duy trì mối quan hệ giữa các pixel bằng cách tìm hiểu các tính năng hình ảnh bằng cách sử dụng các ô vương nhỏ của dữ liệu đầu vào. Nó là 1 phép toán có 2 đầu vào như ma trận hình ảnh và 1 bộ lọc hoặc hạt nhân.



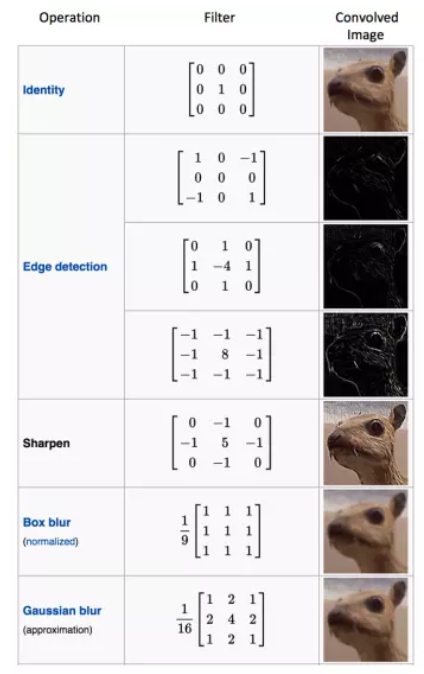
* Xem xét 1 ma trận 5 x 5 có giá trị pixel là 0 và 1. Ma trận bộ lọc 3 x 3 như hình bên dưới.



* Sau đó, lớp tích chập của ma trận hình ảnh 5 x 5 nhân với ma trận bộ lọc 3 x 3 gọi là 'Feature Map' như hình bên dưới.

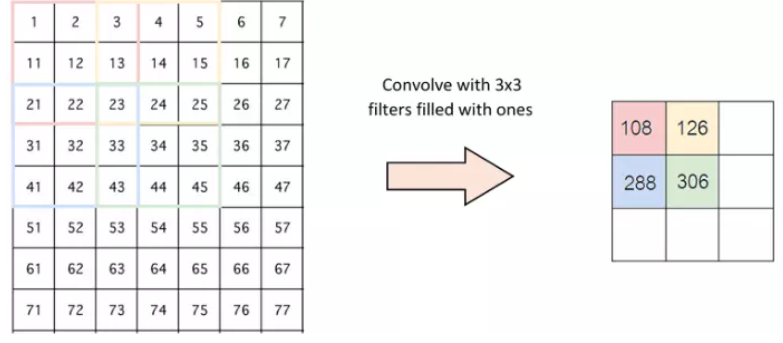
|  |  |  |
| --- | --- | --- |
| (1) | (2) | (3) |
| (6) | (5) | (4) |

* Sự kết hợp của 1 hình ảnh với các bộ lọc khác nhau có thể thực hiện các hoạt động như phát hiện cạnh, làm mờ và làm sắc nét bằng cách áp dụng các bộ lọc. Ví dụ dưới đây cho thấy hình ảnh tích chập khác nhau sau khi áp dụng các Kernel khác nhau.



# ***Bước nhảy – Stride***

Stride là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy. Hình dưới là lớp tích chập hoạt động với stride là 2.



# ***Đường viền – Padding***

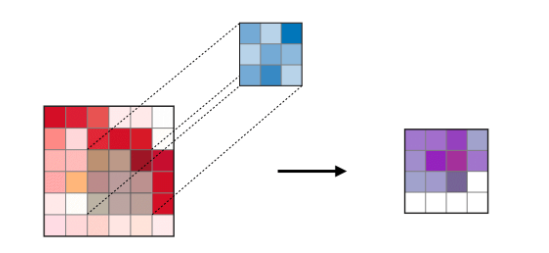
Đôi khi kernel không phù hợp với hình ảnh đầu vào. Ta có 2 lựa chọn:

* Chèn thêm các số 0 vào 4 đường biên của hình ảnh (padding).
* Cắt bớt hình ảnh tại những điểm không phù hợp với kernel.

# ***Các kiểu tầng***

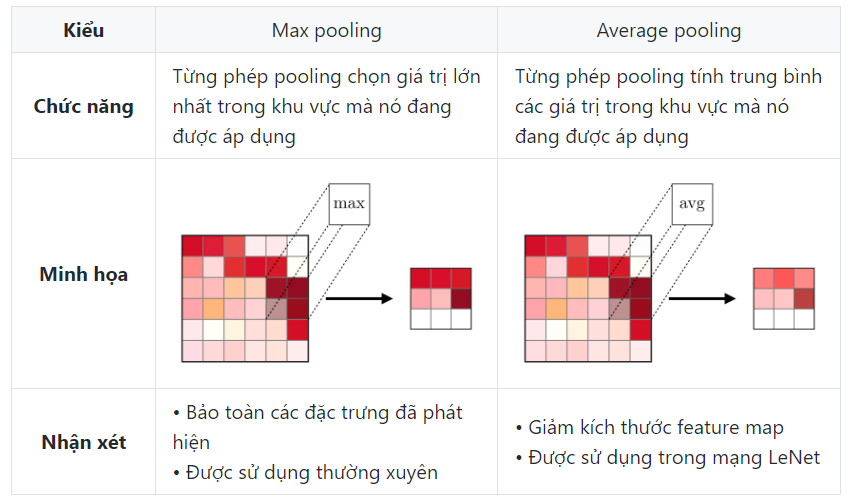
## **Tầng tích chập (CONV)**

Sử dụng các bộ lọc để thực hiện phép tích chập khi đưa chúng đi qua đầu vào *I* theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước bộ lọc *F* và độ trượt (stride) *S*. Kết quả đầu ra *O* được gọi là feature map hay activation map.



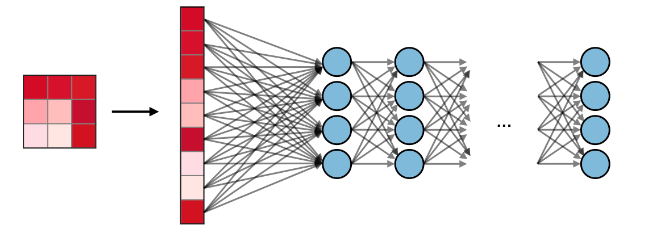
## **Pooling (POOL)**

Là một phép downsampling, thường được sử dụng sau tầng tích chập, giúp tăng tính bất biến không gian. Cụ thể, max pooling và average pooling là những dạng pooling đặc biệt, mà tương ứng là trong đó giá trị lớn nhất và giá trị trung bình được lấy ra.



## Fully Connected (FC)

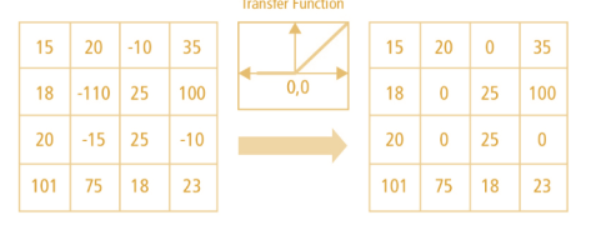
Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.



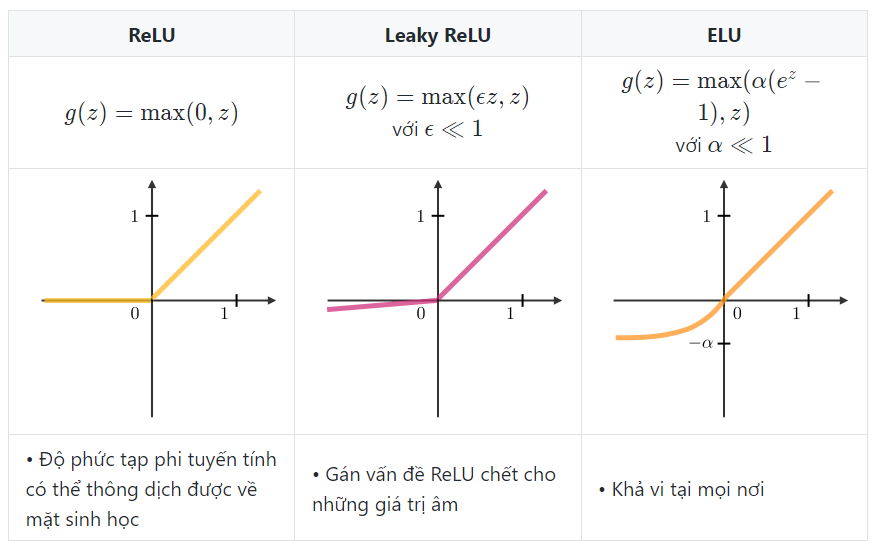
# ***Các hàm kích hoạt thường gặp***

## 6.1 Rectified Linear Unit (ReLu)

* Là 1 hàm phi tuyến, với đầu ra là f(x): max(0,x)
* Tại sao ReLU lại quan trọng: ReLU giới thiệu tính phi tuyến trong ConvNet. Vì dữ liệu trong thế giới mà chúng ta tìm hiểu là các giá trị tuyến tính không âm.

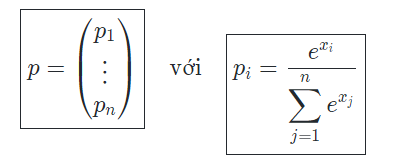
****

* Có 1 số hàm phi tuyến khác như tanh, sigmoid cũng có thể được sử dụng thay cho ReLU. Hầu hết người ta thường dùng ReLU vì nó có hiệu suất tốt.
* Những biến thể khác của ReLU được tổng hợp ở bảng dưới:



## **6.2 Softmax**

Bước softmax có thể được coi là một hàm logistic tổng quát lấy đầu vào là một vector chứa các giá trị và cho ra là một vector gồm các xác suất   thông qua một hàm softmax ở cuối kiến trúc. Nó được định nghĩa như sau:



# **Phần 2: Ứng dụng trong thực tế**

## ***Mô tả bài toán***

* **Tên bài toán**: Phân loại Giày, Sandal và Boot
* **Mục đích của bài toán**: Mục tiêu của bài toán nhằm xây dựng một mô hình học máy sử dụng Convolutional Neural Network (CNN) để phân loại ảnh thuộc vào một trong 3 lớp: Shoe, sandal và boot.
* **Input**:

Dữ liệu đầu vào là các hình ảnh từ bộ dữ liệu “Shoe vs Sandal vs Boot Image Dataset”. Đây là tập hợp các file hình ảnh trong định dạng JPEG (.jpg). Mỗi hình ảnh trong tập dữ liệu được đại diện bởi một file đơn lẻ và thường là các hình ảnh của giày, sandal và boot

* **Output**:

Đầu ra của tập dữ liệu là nhãn lớp (class label) tương ứng với mỗi hình ảnh. Các nhãn lớp được xác định dựa trên 3 danh mục khác nhau: "shoe", "sandal", "boot". Mỗi hình ảnh được gán một nhãn lớp duy nhất, chỉ ra loại hình ảnh mà nó biểu thị.

* **Tóm tắt công việc thực hiện của bài toán**:
* Xử lí dữ liệu:
  + Load dữ liệu từ tập dữ liệu được cung cấp
  + Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra
  + Chuẩn hóa dữ liệu, resize ảnh
  + Chuyển đổi thành định dạng cho phù hợp mô hình
* Xây dựng mô hình CNN:
  + Sử dụng kiến trúc CNN để xây dựng mô hình phân loại
  + Gồm các lớp: Convolutional, Pooling, Fully Connected và lớp đầu ra Softmax.
* Huấn luyện mô hình: Đưa dữ liệu huấn luyện vào mô hình và điều chỉnh các tham số để mô hình học được từ dữ liệu và tối ưu hóa hiệu suất phân loại.
* Đánh giá mô hình: Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra độc lập. Sử dụng các độ đo như độ chính xác (accuracy), độ phủ (recall), và độ chính xác trung bình (average precision), F1-score để đánh giá khả năng phân loại ảnh của mô hình.
* Kiểm tra và triển khai mô hình: Kiểm tra mô hình trên các hình ảnh mới để đánh giá hiệu suất thực tế. Sau đó, mô hình có thể được triển khai vào ứng dụng thực tế, ví dụ như hệ thống nhận diện và phân loại hình ảnh tự động.

## ***Mô tả tập dữ liệu của bài toán***

* **Link download dữ liệu của bài toán:**

<https://www.kaggle.com/datasets/hasibalmuzdadid/shoe-vs-sandal-vs-boot-dataset-15k-images>

* **Mô tả tập dữ liệu của bài toán:**
* Tập dữ liệu có 15.000 ảnh.
* Gồm 3 lớp đối tượng:
  + Shoe
  + Sandal
  + Boot
* Mỗi đối tượng gồm 5000 ảnh.
* Hình ảnh có độ phân giải 136x102 pixel trong mô hình màu RGB.

|  |  |  |
| --- | --- | --- |
| **Shoe** | **Sandal** | **Boot** |

* **Phân chia huấn luyện và kiểm tra:**

Chia tập dữ liệu thành 3 tập con, số mẫu dữ liệu của mỗi tập: training data (dùng để huấn luyện mô hình), validation data (dùng để kiểm thử mô hình), test data (dùng để đánh giá mô hình).

|  |  |
| --- | --- |
| * Trainning data: gồm 9000 mẫu dữ liệu * Validatin data (development): gồm 3000 mẫu dữ liệu * Test data: gồm 3000 mẫu dữ liệu |  |

## ***Thiết kế mô hình CNN***

* **Mô hình số 1:**

|  |  |
| --- | --- |
| + Tổng quan: 3 lớp tích chập, 3 lớp gộp (MaxPool), 3 lớp full connected.  + Thiết kế cụ thể: (theo thứ tự từ  input->ouput)   * Conv: 8 bộ lọc, cửa sổ tích chập (7x7), đệm, bước sải 2, hàm kích hoạt: “relu” * MaxPool: 2x2 * Conv: 16 bộ lọc, cửa sổ tích chập (3x3), đệm, bước sải 1, hàm kích hoạt : “relu” * MaxPool: 2x2 * Conv: 32 bộ lọc, cửa sổ tích chập (5x5), đệm, bước sải 1, hàm kích hoạt : “relu” * MaxPool: 2x2 * Dense: 128 node, hàm kích hoạt : “relu” * Dense:32 node, hàm kích hoạt : “relu” * Dense:3 node, hàm kích hoạt : “softmax” | + Biểu đồ tóm tắt thiết kế:  Diagram  Description automatically generated |

* **Mô hình số 2:**

|  |  |
| --- | --- |
| + Tổng quan: 2 lớp tích chập, 2 lớp gộp (MaxPool), 3 lớp full connected.  + Thiết kế cụ thể: (theo thứ tự từ  input->ouput)   * Conv: 32 bộ lọc, cửa sổ tích chập (7x7), đệm, hàm kích hoạt : “relu” * MaxPool: 2x2 * Conv: 64 bộ lọc, cửa sổ tích chập (3x3), hàm kích hoạt : “relu” * MaxPool: 2x2 * Dense: 120 node, hàm kích hoạt : “sigmoid” * Dense:84 node, hàm kích hoạt : “sigmoid” * Dense:3 node, hàm kích hoạt : “softmax” | + Biểu đồ tóm tắt thiết kế:  Diagram  Description automatically generated |

* **Mô hình số 3:**

|  |  |
| --- | --- |
| + Tổng quan: 3 lớp tích chập, 3 lớp gộp (2MaxPool, 1AveragePool), 3 lớp full connected.  + Thiết kế cụ thể: (theo thứ tự từ  input->ouput)   * Conv: 16 bộ lọc, cửa sổ tích chập (3x3), đệm, bước sải 1, hàm kích hoạt : “relu” * MaxPool: 2x2 * Conv: 32 bộ lọc, cửa sổ tích chập (3x3), bước sải 1, hàm kích hoạt : “relu” * MaxPool: 2x2 * Conv: 64bộ lọc, cửa sổ tích chập (3x3), bước sải 1, hàm kích hoạt : “relu” * 2x2 AveragePool * Dense: 128 node, hàm kích hoạt : “sigmoid” * Dense:64 node, hàm kích hoạt : “sigmoid” * Dense:3 node, hàm kích hoạt : “softmax” | + Biểu đồ tóm tắt thiết kế:  Diagram  Description automatically generated |

## ***Kết quả thực nghiệm***

* Accuracy: Đây là tỷ lệ giữa số lượng mẫu dữ liệu được dự đoán đúng và tổng số mẫu dữ liệu trong tập kiểm tra.
* Precision: Đây là tỷ lệ giữa số lượng mẫu dữ liệu được dự đoán đúng thuộc lớp positive và tổng số mẫu dữ liệu được dự đoán thuộc lớp positive.
* Recall: Đây là tỷ lệ giữa số lượng mẫu dữ liệu được dự đoán đúng thuộc lớp positive và tổng số mẫu dữ liệu thực tế thuộc lớp positive.
* F1-score: Đây là một độ đo kết hợp giữa Precision và Recall, tính theo công thức 2 \* (Precision \* Recall) / (Precision + Recall).
* Kết quả:

**+ Mô hình số 1:**

Text

Description automatically generated with medium confidence

**+ Mô hình số 2:**

Text

Description automatically generated

**+ Mô hình số 3:**

**Text

Description automatically generated**

## 

# **Kết luận**

Qua bài làm về đề tài: Ứng dụng mạng noron tích chập (CNN) để phân loại giày, sandal & boot, nhóm chúng em hi vọng mô hình này có thể phát triển hơn nữa. Đồng thời thông qua bài làm giúp nhóm em có những kiến thức về môn học máy cũng như áp dụng kiến thức đó vào phân tích các bài toán thực tế.

Trải qua quá trình huấn luyện và đánh giá, mô hình CNN đã cho thấy khả năng phân loại chính xác và hiệu suất đáng kể trong việc nhận diện và phân loại các loại giày, sandal & boot khác nhau. Điều này chỉ ra rằng CNN là một phương pháp mạnh mẽ và hiệu quả trong lĩnh vực phân loại ảnh. Kết quả này có thể ứng dụng rộng rãi trong các ứng dụng thực tế như phân loại ảnh, nhận dạng vật thể và phân tích hình ảnh tự động.

Mặc dù đã có nhiều cố gắng trong suốt quá trình làm bài tập, nhưng kiến thức còn hạn chế, cũng như chưa có kinh nghiệm trong thực tế nên không thể tránh khỏi những sai sót. Vì vậy, nhóm em mong thầy, cô góp ý để có thể chỉnh sửa, hoàn thiện bài làm tốt hơn.

# **Tài liệu tham khảo**

1. VIBLO:

<https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>

1. Topdev:

<https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/>

1. ShervineAmidi:

<https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-convolutional-neural-networks>

1. Kaggle:

<https://www.kaggle.com/datasets/hasibalmuzdadid/shoe-vs-sandal-vs-boot-dataset-15k-images>

1. Slide bài giảng