

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC PHENIKAA

---



## BÁO CÁO BÀI TẬP LỚN

### HỌC PHẦN HỌC SÂU

---

#### DERMATOLOGIST-LEVEL CLASSIFICATION OF SKIN CANCER USING CONVOLUTIONAL NEURAL NETWORKS WITH IMBALANCED DATA PREPROCESSING TECHNIQUES

---

Giảng viên hướng dẫn: Ts. Nguyễn Văn Tới

Sinh viên thực hiện: Lê Bá Luật                      21010554    K15KHMT

Đào Quang Hiệp                      21013089    K15KHMT

Phạm Minh Tuấn                      21012400    K15KHMT

Khoa: Công Nghệ Thông Tin

HÀ NỘI, THÁNG 10 2023

# Mục lục

Lời mở đầu	1
1 Giới thiệu tổng quan	2
2 Các nghiên cứu gần đây	3
3 Mô hình mạng Convolutional Neural Networks	6
3.1 Lịch sử của CNN	6
3.2 Tổng quan về CNN	7
3.3 Các lớp trong mạng CNN	8
3.3.1 Lớp đầu vào (Input layer)	8
3.3.2 Lớp tích chập (Convolution Layer)	9
3.3.3 Lớp gộp (Pooling layer)	15
3.3.4 Lớp làm phẳng (Flatten layer)	15
3.3.5 Lớp kết nối đầy đủ (Fully Connected Layer)	15
3.4 Lan truyền ngược (backpropagation)	17
3.4.1 Backpropagation trên lớp Fully Connected (Dense)	17
3.4.2 Backpropagation trên lớp Pooling	19
3.4.3 Backpropagation trên lớp tích chập	20
4 Kết quả thực nghiệm và đánh giá	22
4.1 Về bộ dữ liệu HAM10000	22
4.2 Imbalanced-learn	23
4.3 Thiết kế mô hình mạng CNN	24
4.4 Kết quả	24
4.5 Đánh giá và nhận xét	26
Kết luận	28

# Mục lục hình ảnh

1	Cấu trúc cơ bản của mạng CNN.	8
2	Tích chập một chiều.	10
3	Phép tích chập hai chiều cho ảnh $6 \times 6$ ( $f = 3, s = 1, p = 0$ ).	11
4	Minh họa tích chập hai chiều đơn kênh với các cặp $(p, s)$ .	12

5	Ví dụ tính toán trong Conv2D. . . . .	13
6	Sử dụng hàm kích hoạt trong conv2D. . . . .	14
7	Average Pooling và Max Pooling với $f = 2, s = 2$ . . . . .	15
8	Lớp Fully Connected trong CNN. . . . .	16
9	Kiến trúc mạng CNN. . . . .	17
10	Ví dụ phép tích chập chuyển vị. . . . .	21
11	Ví dụ các mẫu ung thư da trong bộ dữ liệu HAM10000. . . . .	22
12	Biểu đồ histogram thể số lượng ảnh mỗi loại chẩn đoán trong dữ liệu. .	23
13	Áp dụng kỹ thuật Random Over-Sampling với hàm logistic. . . . .	24
14	Kết quả huấn luyện với mô hình mạng cơ sở. . . . .	25
15	Kết quả huấn luyện với mô hình trên bộ dữ liệu thu được sau khi xử lý mất cân bằng dữ liệu. . . . .	26

## Mục lục bảng

1	So sánh độ chính xác của các mô hình tốt nhất trong các nghiên cứu được đưa ra trước đó trên bộ dữ liệu HAM10000. . . . .	4
2	Cấu trúc mạng CNN sử dụng để huấn luyện. . . . .	25

## Lời mở đầu

Ung thư da (skin cancer) là một trong những loại bệnh ung thư khá phổ biến hiện nay. Ung thư da có thể xuất hiện ở bất kỳ vị trí nào trên cơ thể và dấu hiệu của ung thư da được biểu hiện thông qua một số dấu hiệu và triệu chứng trên da như mụn cóc hoặc vết loét không lành, nổi mẩn đỏ, sưng tấy hoặc chảy máu,... Việc phát hiện những dấu hiệu này sớm sẽ giúp ích rất nhiều cho việc chẩn đoán và phòng bệnh sớm tránh để tránh tình trạng bệnh trở nên nghiêm trọng hơn và giúp tăng tỷ lệ sống sót của người bị bệnh. Thường rất khó để phân biệt các khối u ác tính sớm với hình ảnh da của bệnh nhân, ngay cả với các bác sĩ da liễu chuyên nghiệp. Vì vậy, một số cách tiếp cận trong việc phân loại da đã được đề xuất và tăng tỉ lệ chính xác trong việc phân loại các mức độ ung thư da để từ đó đưa ra các phương pháp điều trị hợp lý.

Báo cáo này sẽ tập trung phát triển một số mô hình mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) kết hợp với một số kỹ thuật xử lý dữ liệu bất cân bằng (imbalanced data preprocessing techniques) để áp dụng vào bài toán chẩn đoán mức độ ung thư da cho bộ dữ liệu Skin Cancer MNIST: HAM10000 [1]. Từ đó, triển khai mô hình rồi đánh giá và so sánh hiệu suất của mô hình khi so sánh với các cách tiếp cận hiện tại trong bài toán phân loại các mức độ ung thư da.

# 1 Giới thiệu tổng quan

Ung thư là một trong những nguyên nhân gây tử vong hàng đầu trên thế giới. Theo số liệu thống kê từ GLOBOCAN 2020, trên toàn thế giới có khoảng 19.3 triệu ca ung thư mới được ghi nhận, trong đó có khoảng 10 triệu ca ung thư tử vong. Theo phân tích từ Sung, Hyuna và những người khác [2], vào năm 2020, có 1.522.708 ca ung thư da (bao gồm cả ung thư có khối u ác tính (melanoma) và khối u không ác tính (non-melanoma)), trong đó có đến 691.747 ca được chẩn đoán phát hiện ở khu vực Bắc Mỹ, chiếm hơn 45% tổng số ca mắc trên toàn thế giới, và đây cũng là căn bệnh ung thư được chẩn đoán phổ biến nhất ở nước Mỹ, theo số liệu thu thập từ Cancer Facts & Figures 2018 [3]. Vào cùng năm đó, bệnh ung thư da đã gây ra 120.774 ca tử vong, trong đó ung thư da có khối u ác tính (melanoma skin cancer) là một trong những loại ung thư da phổ biến, nguy hiểm nhất và đã mang đến một thử thách vô cùng to lớn đối với các hệ thống y tế từ trước đến nay [4].

Do đó, điều này sẽ tạo ra một nhu cầu cao trong việc kiểm tra và phát hiện sớm các loại ung thư da khác nhau để ngăn chúng trở nên nghiêm trọng hơn, từ đó tạo cơ hội cho dự đoán bệnh và phòng bệnh tốt hơn. Phương pháp truyền thống thường là bắt đầu bằng kiểm tra mắt thường của bác sĩ và sau đó sử dụng hình ảnh nội soi da (dermoscopy) để hỗ trợ chẩn đoán. Trong đó, nội soi da là một phương pháp không xâm lấn có thể chụp hình ảnh độ phân giải cao của da, cho phép các bác sĩ da liễu phát hiện những đặc điểm mà không thể nhìn thấy bằng mắt thường. Tuy nhiên, một số lượng lớn bệnh nhân bị ung thư da không kịp nhận được sự chẩn đoán sớm và điều trị kịp thời do thiếu sự có mặt của các bác sĩ chuyên nghiệp, chẩn đoán không chính xác hay sự chênh lệch về trình độ của các bác sĩ, và áp lực làm việc lặp đi lặp lại của các bác sĩ.

Với sự phát triển của trí tuệ nhân tạo (Artificial Intelligence - AI) trong lĩnh vực y học, học sâu (Deep Learning - DL) đã được sử dụng rộng rãi trong việc phát hiện và phân loại hình ảnh y học trong những năm gần đây [5]. Ứng dụng trí tuệ nhân tạo vào việc hỗ trợ chẩn đoán hình ảnh y học ngày càng đóng vai trò quan trọng trong lĩnh vực trí tuệ nhân tạo y học, đặc biệt trong việc nhận biết thông minh, hỗ trợ quyết định điều trị chính xác và các khía cạnh khác [6]. Thị giác máy tính (Computer Vision) là một lĩnh vực trong AI mà trong đó hệ thống học cách để tự động nhận biết, mô tả và xử lý những hình ảnh trực quan một cách chính xác và hiệu quả. Nó đã đẩy mạnh quá trình đánh giá hình ảnh y học với độ chính xác cao hơn và phân tích hiệu quả hơn [7]. Mạng nơ-ron tích chập (CNN) là một loại mạng nơ-ron nhân tạo đã cách mạng hóa trong việc phân tích, xử lý hình ảnh mà không cần trích xuất các đặc điểm thủ công

truyền thống như màu sắc, giá trị cường độ, cấu trúc tô-pô và thông tin về kết cấu của ảnh.

## 2 Các nghiên cứu gần đây

Mạng CNN cùng nhiều mô hình học máy khác đã được nghiên cứu và áp dụng cho bài toán phân loại mức độ ung thư da. Đã có nhiều nghiên cứu sử dụng các kỹ thuật học máy để giải quyết bài toán như nghiên cứu của M. Emre Celebi và những người khác [8] sử dụng thuật toán Support Vector Machine (SVM), nghiên cứu của Illias Maglogiannis [9] sử dụng thuật toán Naive Bayes và nghiên cứu của M. Emre Celebi [10] sử dụng cây quyết định (Decision Trees). Tuy nhiên, các nghiên cứu này chỉ áp dụng trên các bộ dữ liệu nhỏ và không mang lại độ chính xác cao. Với sự phát triển của học sâu, đặc biệt là sự xuất hiện của những dự án lớn như ImageNet (Deng J., 2009) [11], CIFAR (Canadian Institute For Advanced Research - Krizhevsky và Hinton), MNIST (Modified National Institute of Standfords and Technology - Deng L., 2012) [12], COCO (Common Objects in Context - Lin, 2014) [13], Open Images (Kuznetsova, 2020) [14] và SUN (Xiao J, 2010) [15], các mô hình mạng nơ-ron học sâu, đặc biệt là mạng CNN ngày càng được áp dụng nhiều cho các bài toán xử lý hình ảnh y tế bởi độ chính xác cao và thời gian xử lý nhanh chóng.

Năm 2018, Amirreza Rezvantlab và những người khác [16] đã áp dụng các mô hình pretrained DensNet 201, ResNet 152, Inception V3, InceptionResNet V2 trên bộ dữ liệu gồm 10135 ảnh nội soi da (HAM10000: 10015, PH2: 120). Nghiên cứu chỉ ra rằng mô hình DensNet 201 sau khi hiệu chỉnh với kích thước ảnh đầu vào là (224, 224), hệ số học learning rate 0.0006 và thuật toán stochastic gradient descent (SGD) với decay và momentum 0.9 mang lại hiệu suất tốt nhất trong cả các metric đánh giá như Precision (89.01% – 85.24%), F1-score (89.01% – 85.13%) và ROC AUC (98.79% – 98.16%).

Năm 2021, Soumyya K. D. và những người khác [17] đã áp dụng Soft-Attention (SA) vào các mô hình VGG, ResNet, Inception ResNet v2 (IRv2) và DenseNet rồi áp dụng vào bài toán phân loại da trên bộ dữ liệu HAM10000 và ISIC 2017 [18]. Việc áp dụng Soft-Attention giúp cải thiện hiệu suất của các mô hình tăng lên đến 4.7%, trong đó, mô hình IRv2+SA đem lại kết quả tốt nhất với 98.4% Avg AUC, 93.7% Precision, 93.4% Accuracy trên bộ dữ liệu HAM10000, trong khi trên bộ dữ liệu ISIC 2017 đạt 95.9% AUC và độ chính xác đạt 90.4%.

Wessam M. Salamaa và Moustafa H. Aly [19] đã đưa ra một nghiên cứu vào năm 2021 sử dụng kết hợp các mô hình ResNet50 và VGG-16 với các kỹ thuật tiền xử lý, thuật toán k-fold cross validation và SVM để áp dụng cho bài toán phân loại mức độ

ung thư da trên các bộ dữ liệu ISIC 2017, HAM10000 và ISBI 2016. Trong đó, các kỹ thuật tiền xử lý được sử dụng bao gồm: median filter, contrast enhancement và edge detection và các lớp Fully Connected (FC) được thay thế bởi thuật toán phân lớp SVM. Việc xử dụng các kỹ thuật tiền xử lý đã giúp cải thiện độ chính xác của mô hình lên đến hơn 5% cho tất cả các bộ dữ liệu. Nghiên cứu chỉ ra rằng mô hình ResNet50 mang lại hiệu suất tốt hơn VGG16 trong tất cả các bộ dữ liệu. Việc kết hợp thêm thuật toán phân lớp SVM, kết quả tốt nhất của mô hình ResNet50 có độ chính xác trung bình khoảng  $98.43\% \pm 0.19$ , AUC  $98.78\% \pm 0.23$ , Sensitivity  $98.42\% \pm 0.24$ , Precision  $96.31\% \pm 0.27$  và  $96.98\% \pm 0.29$  F1-score.

Năm 2022, Zhangli Lan và những người khác [20] đã đưa ra một mô hình mạng cải thiện của mạng CapsNets, gọi là FixCaps áp dụng cho bài toán phân loại ảnh nội soi da. FixCaps có kích thước kernel  $31 \times 31$ , lớn hơn và có hiệu suất tốt hơn kích thước  $9 \times 9$  thông thường, mang lại độ chính xác 96.43% trên bộ dữ liệu HAM10000 và cao hơn các mô hình đề xuất trước đó như GoogLeNet, Inception V3, MobileNet V3 và IRv2+SA. Ngoài ra trong nghiên cứu còn thực nghiệm kết hợp với deep-wise separable convolution (DS), gọi là FixCaps-DS, mang lại hiệu suất tốt và giảm số lượng biến phải sử dụng (0.14 triệu biến, chỉ xấp xỉ 10% so với MobileNet V3) và có độ chính xác lên đến 96.13%, cao hơn so với IRv2+SA được Soumyya K. D. đề xuất năm 2021 [17] là 93.4%.

Xinrong Lu và Y. A. Firoozeh Abolhasani Zadeh [21] cũng đưa ra một mô hình cải thiện của mạng Xception và áp dụng cho bộ dữ liệu MNIST10000 đạt Accuracy 100%, Sensitivity 94.05%, Precision 97.07% và F1-score 95.53%. Năm 2023, Maryam Tahir và một số người khác [22] cũng đưa ra một mô hình cải thiện của mạng DSCC\_Net mang lại kết quả Accuracy 94.17%, Precision 94.28% và 99.43% AUC.

Model	Accuracy (%)	AUC (%)	Micro Precision (%)
DensNet201 [16]	89.01	98.79	89.01
IRv2-SA [17]	93.40	98.40	93.70
ResNet50+SVM [19]	98.43	98.78	96.31
FixCaps [20]	96.49	-	-
XceptionNet [21]	<b>100</b>	-	<b>97.07</b>
DSCC_Net [22]	94.17	<b>99.43</b>	94.28

Bảng 1: So sánh độ chính xác của các mô hình tốt nhất trong các nghiên cứu được đưa ra trước đó trên bộ dữ liệu HAM10000.

Trong bài báo cáo này, ở phần tiếp theo chúng tôi sẽ giới thiệu tổng quan về mạng CNN và cấu trúc cơ bản của mạng CNN và các lớp, các bước khi huấn luyện mạng CNN. Tiếp đến, chúng tôi sẽ giới thiệu về bộ dữ liệu HAM10000 được sử dụng trong báo cáo này và kỹ thuật xử lý bất cân bằng được áp dụng. Sau đó, chúng tôi sẽ xây dựng và triển khai một mô hình mạng nơ-ron tích chập CNN, so sánh hiệu suất của mô hình lúc áp dụng và không áp dụng kỹ thuật xử lý bất cân bằng dữ liệu khi huấn luyện trên bộ dữ liệu HAM10000.



## 3 Mô hình mạng Convolutional Neural Networks

### 3.1 Lịch sử của CNN

Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) là một mô hình học máy có thể được sử dụng để xử lý dữ liệu hình ảnh và video. CNN được phát triển dựa trên nguyên tắc hoạt động của hệ thần kinh thị giác của con người.

Lịch sử phát triển của CNN bắt đầu từ những năm 1960, khi Hubel và Wiesel phát hiện ra rằng các tế bào thần kinh trong não bộ của động vật có vú có thể nhận biết các tính năng cụ thể trong hình ảnh. Những phát hiện này đã truyền cảm hứng cho các nhà khoa học máy tính phát triển các mô hình CNN có thể học cách nhận ra các tính năng trong hình ảnh.

Một trong những mô hình CNN đầu tiên được phát triển bởi Fukushima vào năm 1979 [23], mô hình này được sử dụng để nhận dạng chữ viết tay. Trong những năm 1980, CNN đã được sử dụng cho một số ứng dụng, bao gồm nhận dạng chữ viết tay, phân loại ảnh và phát hiện đối tượng. Tuy nhiên, CNN vẫn chưa được sử dụng rộng rãi vì các vấn đề về tính phức tạp và hiệu suất.

Năm 1998, LeCun et al. [24] đã phát triển một mô hình CNN mới có tên là LeNet. Mô hình này đã đạt được thành công đáng kể trong việc nhận dạng chữ viết tay.

Năm 2012, AlexNet [25] đã được phát triển bởi một nhóm nghiên cứu của Đại học Toronto gồm Alex Krizhevsky, Ilya Sutskever và Geoffrey E. Hinton. AlexNet là một mô hình CNN lớn hơn và phức tạp hơn LeNet. AlexNet đã đạt được thành công vang dội trong cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC), một cuộc thi hàng năm về nhận dạng hình ảnh. Thành công của AlexNet đã thúc đẩy sự phát triển của CNN. Trong những năm gần đây, CNN đã được sử dụng rộng rãi cho nhiều ứng dụng, bao gồm nhận dạng hình ảnh, phân loại ảnh, phát hiện đối tượng, phân loại video và tạo hình ảnh. Sau sự xuất hiện của mạng AlexNet, nhiều mô hình mạng CNN nổi tiếng khác được ra đời bao gồm: VGGNet (2014), GoogleNet (2014), ResNet (2015), Inception V3 (2015), Xception (2016), DenseNet (2016),...

Cho đến hiện nay, CNN hiện đang là một trong những mô hình học máy phổ biến nhất và được sử dụng trong nhiều ứng dụng thực tế, bao gồm: Nhận dạng khuôn mặt, Xác định đối tượng trong video, Phân loại ảnh y tế, Tạo hình ảnh chân dung,... CNN vẫn đang được tiếp tục nghiên cứu và phát triển. Các nhà khoa học vẫn đang liên tục tìm cách cải thiện hiệu suất của CNN và mở rộng ứng dụng của CNN sang các lĩnh vực mới.

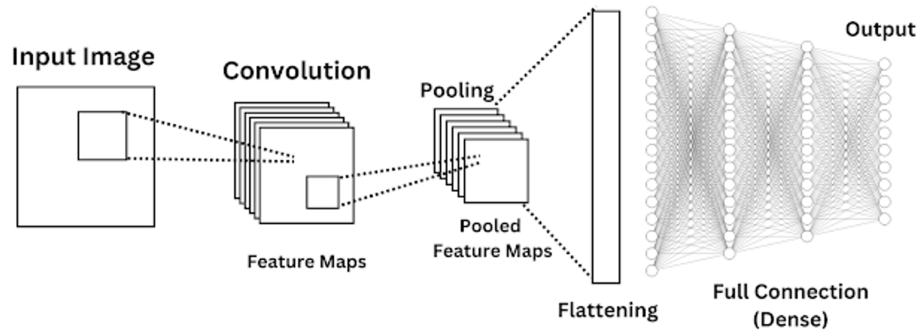
## 3.2 Tổng quan về CNN

Trong học máy, trình phân loại sẽ gán nhãn lớp cho điểm dữ liệu. Ví dụ: bộ phân loại hình ảnh tạo ra nhãn lớp (ví dụ: chó mèo) cho những đối tượng tồn tại trong một hình ảnh. Mạng neural tích chập, hay gọi tắt là CNN, là một loại máy phân loại có khả năng giải quyết vấn đề này rất tốt!

CNN là một mạng lưới thần kinh: một thuật toán được sử dụng để nhận dạng các mẫu trong dữ liệu. Mạng nơ-ron nói chung bao gồm một tập hợp các nơ-ron được tổ chức thành các lớp, mỗi lớp có trọng số và độ lệch có thể học được riêng. Hãy chia CNN thành các khối xây dựng cơ bản của nó:

1. Tensor (Tensor): Tensor có thể được hiểu như một cấu trúc dữ liệu đa chiều, tương tự như ma trận nhưng với số chiều  $n$ . Trong mạng CNN, Tensor thường có ba chiều (chiều cao, chiều rộng và chiều kênh) cho đến khi đến tầng đầu ra. Tensor là một khái niệm quan trọng trong xử lý dữ liệu đa chiều, như hình ảnh trong mạng CNN.
2. Neuron (Nơ-ron): Một neuron có thể được xem như một hàm số nhận nhiều đầu vào và cho ra một đầu ra duy nhất. Trong mạng CNN, các neuron thường biểu diễn bằng các bản đồ màu từ đỏ đến xanh lam, được gọi là các bản đồ hoạt động. Các neuron thực hiện các phép tính trên dữ liệu đầu vào để tạo ra các đặc trưng hữu ích cho việc phân loại ảnh.
3. Layer (Tầng): Tầng trong mạng CNN là một nhóm các neuron thực hiện cùng một phép toán. Các neuron trong cùng một tầng thường có các siêu tham số giống nhau, và tầng thường là các khối xây dựng cơ bản trong kiến trúc mạng CNN.
4. Kernel weights and biases (Trọng số và bias của kernel): Đây là các tham số cho mỗi neuron trong mạng CNN. Trọng số được sử dụng để trọng số hóa đầu vào, trong khi bias được sử dụng để điều chỉnh đầu ra của neuron. Các giá trị này được tinh chỉnh trong quá trình huấn luyện mạng và cho phép mạng thích nghi với dữ liệu cung cấp.
5. Class score: Một CNN truyền tải một hàm số điểm khả vi, được biểu diễn dưới dạng điểm số lớp trong tầng đầu ra của nó. Hàm số điểm này có tính chất liên tục và có thể tích hợp và tạo đạo hàm, điều này quan trọng trong quá trình huấn luyện mạng nơ-ron. Trong tầng đầu ra của mạng CNN, các class score được tính toán cho mỗi lớp hoặc nhãn mà mạng được đào tạo để phân loại. Điểm số này

thường đại diện cho mức độ tự tin của mạng đối với việc phân loại hình ảnh thành các lớp khác nhau. Bằng cách so sánh điểm số giữa các lớp, chúng ta có thể xác định lớp mà mạng dự đoán là chính xác nhất cho hình ảnh đầu vào.



Hình 1: Cấu trúc cơ bản của mạng CNN.

CNN đã chứng tỏ mình là một công cụ mạnh mẽ cho các ứng dụng thị giác máy tính và đã đạt được hiệu suất ấn tượng trong việc giải quyết các vấn đề liên quan đến xử lý hình ảnh. Tiếp sau đây ta sẽ đi vào việc hoạt động của các layers chính trong một mạng CNN.

### 3.3 Các lớp trong mạng CNN

#### 3.3.1 Lớp đầu vào (Input layer)

Tầng đầu vào (input layer) trong mạng CNN là phần đầu tiên của mạng, và nó có nhiệm vụ nhận và xử lý dữ liệu đầu vào, thường là hình ảnh hoặc dữ liệu có cấu trúc tương tự. Dưới đây là một số điểm quan trọng về tầng đầu vào trong CNN:

1. Dữ liệu đầu vào: Tầng đầu vào chứa dữ liệu đầu vào, thường là một hình ảnh hoặc một Tensor đa chiều thể hiện hình ảnh. Dữ liệu này thường có kích thước cố định và được xử lý bằng cách chuyển đổi nó thành các đặc trưng phù hợp cho việc trích xuất thông tin từ hình ảnh.
2. Kích thước đầu vào: Mạng CNN cần biết kích thước của dữ liệu đầu vào để thiết lập kích thước tầng đầu ra cho các tầng tích chập sau. Thông thường, kích thước đầu vào được chỉ định trước khi xây dựng mạng.

3. Kênh đầu vào: Hình ảnh màu thông thường có ba kênh (RGB), trong khi hình ảnh xám chỉ có một kênh. Tầng đầu vào phải có số kênh tương thích với đầu vào.
4. Tiền xử lý đầu vào: Trong một số trường hợp, tiền xử lý có thể được thực hiện trên dữ liệu đầu vào để chuẩn bị nó cho việc đưa vào mạng CNN. Điều này có thể bao gồm việc chuẩn hóa giá trị pixel hoặc thay đổi kích thước hình ảnh để phù hợp với kích thước mạng.
5. Thông tin đầu ra: Tầng đầu vào không thực hiện bất kỳ tính toán nào trên dữ liệu, nó chỉ đơn giản là nhận dữ liệu đầu vào và truyền nó xuống các tầng tích chập đầu tiên trong mạng CNN.

Tầng đầu vào là một phần quan trọng của mạng CNN vì nó đảm bảo rằng dữ liệu đầu vào được chuẩn bị đúng cách và truyền vào mạng để bắt đầu quá trình học tập và trích xuất đặc trưng từ hình ảnh.

### 3.3.2 Lớp tích chập (Convolution Layer)

CNN sử dụng một loại tầng đặc biệt gọi là tầng tích chập (convolutional layer), điều này làm cho chúng rất thích hợp để học từ dữ liệu hình ảnh và dữ liệu giống hình ảnh. Về dữ liệu hình ảnh, CNN có thể được sử dụng cho nhiều nhiệm vụ thị giác máy tính khác nhau, như xử lý hình ảnh, phân loại, phân đoạn và phát hiện đối tượng.

Tầng tích chập trong CNN cho phép mạng học cách phát hiện các đặc trưng cụ thể trong hình ảnh, như cạnh, góc, hoặc các đặc điểm quan trọng khác. Điều này làm cho CNN trở nên rất hiệu quả trong việc xử lý hình ảnh vì chúng có khả năng chia sẻ trọng số (weight sharing) và sử dụng các tầng tích chập để trích xuất thông tin cục bộ từ hình ảnh.

#### 3.3.2.1 Phép toán tích chập

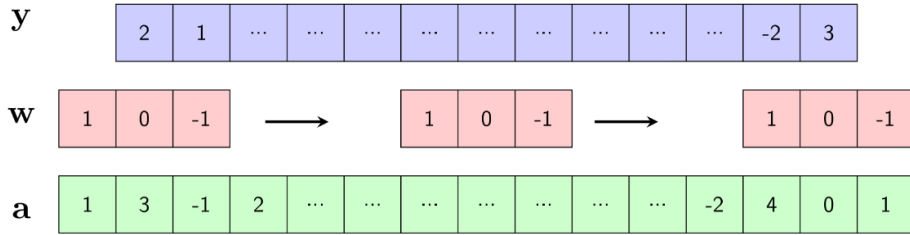
- Tích chập một chiều (conv1D): Xét tín hiệu một chiều  $a(t)$  và bộ lọc (filter)  $w(t)$ . Tích chập của tín hiệu và bộ lọc là một tín hiệu một chiều mới  $b(t)$  được xác định theo công thức:

$$b(t) = \int_u a(u)w(t-u)du \quad (1)$$

Nếu giả sử rằng  $a(t)$  và  $w(t)$  được định nghĩa chỉ trên giá trị nguyên của  $t$ , khi đó phép tích chập một chiều có thể được định nghĩa như sau:

$$b(t) = \sum_{u=-\infty}^{\infty} x(u)w(t-u) \quad (2)$$

Ví dụ về phép tích chập một chiều rời rạc (không có padding) được minh họa trong Hình 2.



Hình 2: Tích chập một chiều.

- Tích chập 2 chiều (conv2D): Tương tự ta xét tín hiệu đầu vào 2 chiều  $I(i, j)$  và kernel 2 chiều  $K(i, j)$ . Giả sử  $I, K$  được định nghĩa chỉ trên giá trị nguyên của  $i, j$ , tức là ma trận 2 chiều, khi đó, phép tích chập 2 chiều với các biến được định nghĩa như sau:

$$S(i, j) = (I * K)(i, j) = \sum_u \sum_v I(u, v)K(i-u, j-v) \quad (3)$$

Do tính giao hoán của phép tích chập, phép tích chập 2 chiều của  $I$  và  $K$  có thể được viết như sau:

$$S(i, j) = (K * I)(i, j) = \sum_u \sum_v I(i-u, j-v)K(u, v) \quad (4)$$

Khi bộ lọc hay kernel có tính đối xứng, phép tích chập trùng với phép tương quan (cross-correlation). Trong nhiều tài liệu học máy, phép tích chập được cài đặt bằng phép tương quan chéo:

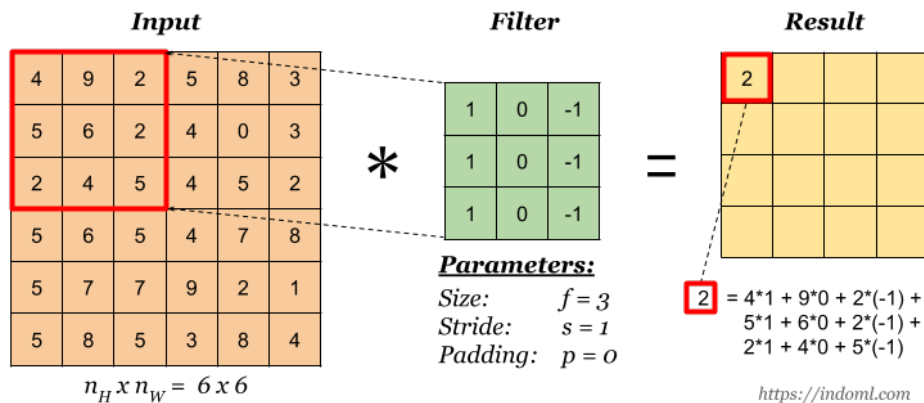
$$S(i, j) = (I * K)(i, j) = \sum_u \sum_v I(i+u, j+v)K(u, v) \quad (5)$$

Khi thực hiện tích chập, ta cần phải quan tâm tới một số hyperparameter cho filter đó là size, stride và padding của filter định rõ cách mà filter di chuyển qua đầu vào và

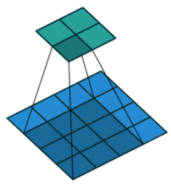
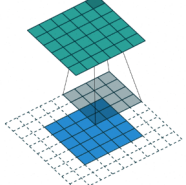
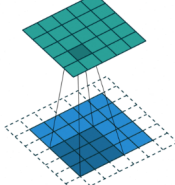
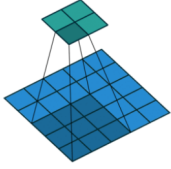
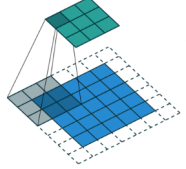
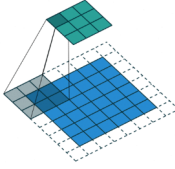
cách nó tương tác với dữ liệu để trích xuất đặc trưng.

1. Kích thước (Size): Đây là kích thước của filter, thường được định nghĩa bằng chiều rộng và chiều cao của filter. Ví dụ, filter 3x3 là ma trận gồm 3 dòng, 3 cột. Nếu kernel là ma trận vuông, ta xét kernel có kích thước  $f \times f$ .
2. Bước nhảy (Stride): Stride là khoảng cách giữa các vị trí liên tiếp mà filter được áp dụng trên đầu vào. Stride xác định cách mà filter di chuyển qua dữ liệu đầu vào. Ví dụ, với stride bằng 1, filter di chuyển mỗi pixel một lần, trong khi với stride bằng 2, filter nhảy qua một pixel mỗi lần. Ký hiệu là  $s$ .
3. Đệm (Padding), ký hiệu là  $p$ : Padding là việc thêm giá trị 0 hoặc giá trị khác vào xung quanh đầu vào trước khi filter được áp dụng. Padding có thể giúp duy trì kích thước đầu ra sau khi áp dụng filter, và nó cũng có thể giúp việc trích xuất đặc trưng từ biên của hình ảnh trở nên hiệu quả hơn. Có hai loại padding phổ biến: "same" (duy trì kích thước:  $p = (f - 1)/2$ ) và "valid" (không padding:  $p = 0$ ).

Sau khi áp dụng phép tích chập 2 chiều cho tín hiệu đầu vào  $I$  kích thước  $n \times n$  với kernel  $f \times f$ , bước nhảy  $s$  và giá trị đệm  $p$ , ta sẽ thu được ảnh mới có kích thước  $n_1 \times n_1$  với  $n_1 = \lfloor \frac{n+2p-f}{s} + 1 \rfloor$ . Ví dụ quá trình áp dụng phép tích chập 2 chiều cho một số giá trị  $f, s, p$  được trình bày trong Hình 3 và Hình 4.



Hình 3: Phép tích chập hai chiều cho ảnh  $6 \times 6$  ( $f = 3, s = 1, p = 0$ ).

		
$P = 0, S = 1$	$P \text{ bất kỳ}, S = 1$	$P = (f - 1)/2, S = 1$
		
$P = 0, S = 2$	$P = (f - 1)/2, S = 2$	$\frac{N+2P-f}{s} \notin \mathbb{N}$

Hình 4: Minh họa tích chập hai chiều đơn kênh với các cặp  $(p, s)$ .

Minh họa cụ thể hơn về phép tích chập 2 chiều với ảnh đầu vào  $X_{3 \times 3}$ , kernel  $w_{2 \times 2}$ :

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \end{bmatrix},$$

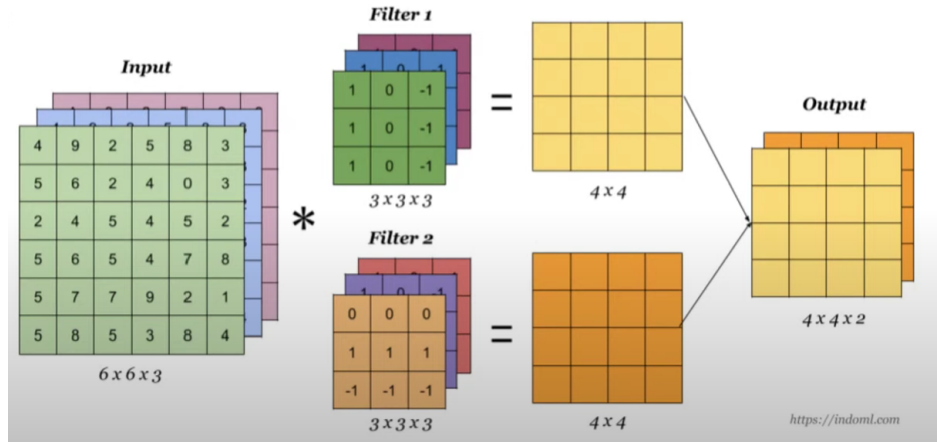
Chuyển ảnh đầu ra thành dạng vector với phép tương quan chéo:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_4 + w_4x_5 \\ w_1x_2 + w_2x_3 + w_3x_5 + w_4x_6 \\ w_1x_4 + w_2x_5 + w_3x_7 + w_4x_8 \\ w_1x_5 + w_2x_6 + w_3x_8 + w_4x_9 \end{bmatrix}$$

$$= \begin{bmatrix} w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \mathbf{C}\mathbf{x}$$

Từ ví dụ trên, ta có thể thấy tích chập 2 chiều tương đương phép biến đổi tuyến tính với ma trận trọng số thưa có chia sẻ (số phần tử khác 0 trên mỗi dòng bằng kích thước kernel,  $w_1, w_2, w_3, w_4$  xuất hiện trên các tất cả các dòng của ma trận).

### 3.3.2.2 Tính toán trong Conv2D



Hình 5: Ví dụ tính toán trong Conv2D.

Lớp Conv2D chứa  $n_f$  bộ lọc (kích thước  $3 \times 3 \times n_c, 5 \times 5 \times n_c, \dots$ ), với  $n_c$  là số kênh của đầu vào. Đầu vào nhân chập với từng bộ lọc và cho qua hàm kích hoạt  $\sigma_t$  tạo ra các bản đồ đặc trưng (feature map). Ghép các bản đồ đặc trưng lại với nhau ta được ảnh đầu ra của lớp conv2D gồm  $n_f$  kênh.

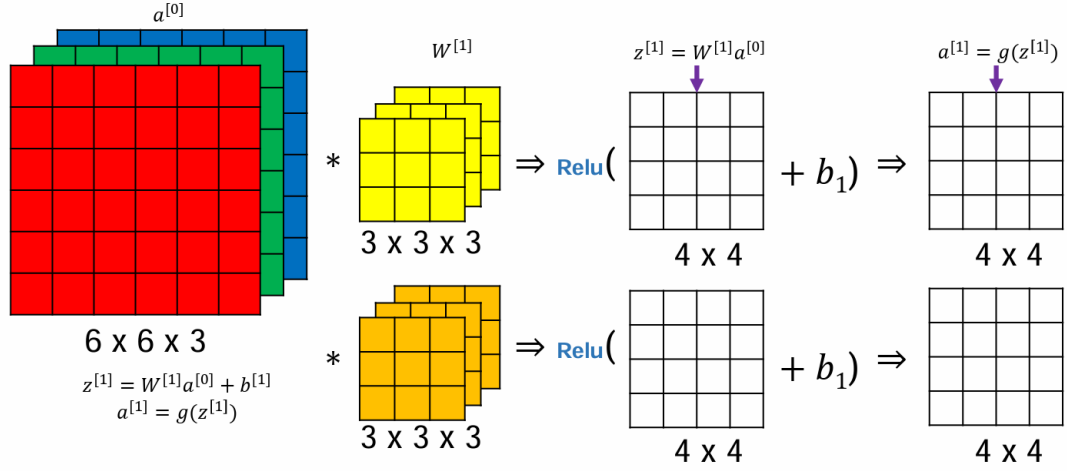
Ví dụ: Trong Hình 5, ảnh đầu vào có  $n_c = 3$  kênh và có  $n_f = 2$  bộ lọc, 16 nơ-ron trong bản đồ đặc trưng chia sẻ các trọng số trong bộ lọc 1, 16 nơ-ron trong bản đồ đặc trưng phía dưới chia sẻ trọng số trong bộ lọc 2.

Bản chất conv2D vẫn là 1 lớp mạng trong MLP (Multi layer perceptron) nhưng các trọng số được chia sẻ giữa các nơ-ron (làm giảm tổng số tham số). Đồng thời do tính chất của phép tích chập và kích thước bộ lọc nhỏ, kết quả tính toán thể hiện các tính chất cục bộ của ảnh đầu vào (dẫn đến tên gọi là bản đồ đặc trưng).

### 3.3.2.3 Hàm kích hoạt (activation function)

Sau khi tính toán tích chập, đầu ra sẽ được đưa qua 1 hàm activation function, minh họa trong Hình 6. Một phần lý do mà những mạng CNN đột phá này có thể đạt được độ chính xác khổng lồ là vì tính phi tuyến tính của hàm kích hoạt.





Hình 6: Sử dụng hàm kích hoạt trong conv2D.

Tính phi tuyến ở hàm kích hoạt là cần thiết để tạo ra biến quyết định phi tuyến tính, để đầu ra không thể được biểu diễn như một tổ hợp tuyến tính của các đầu vào. Nếu không có hàm kích hoạt phi tuyến tính, kiến trúc mạng CNN sâu sẽ trở thành một lớp tích chập đơn, không hoạt động hiệu quả bằng cách sử dụng hàm kích hoạt phi tuyến tính.

**Hàm kích hoạt ReLU:** Hàm kích hoạt ReLU được sử dụng cụ thể như một hàm kích hoạt phi tuyến tính, thay vì các hàm phi tuyến tính khác như Sigmoid, vì nhiều thực nghiệm trong các nghiên cứu chỉ ra rằng CNN sử dụng hàm kích hoạt ReLU huấn luyện có tốc độ nhanh hơn so với các hàm kích hoạt khác. Hàm kích hoạt này được áp dụng trên từng phần tử trên mọi giá trị từ Tensor đầu vào.

$$\text{ReLU}(x) = \max(0, x) \quad (6)$$

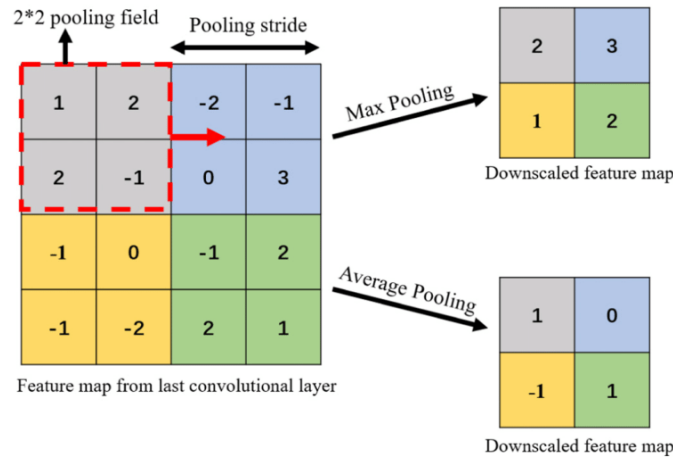
**Hàm kích hoạt Softmax:** Hàm Softmax là hàm đảm bảo tổng đầu ra của mạng CNN bằng 1. Vì điều này, hàm Softmax được sử dụng để chuyển đổi đầu ra của mạng thành một phân phối xác suất trên các lớp (hoặc các lựa chọn).

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (7)$$

Lớp hoặc lựa chọn với xác suất cao nhất sau khi áp dụng Softmax thường được chọn là dự đoán của mô hình cho đầu vào cụ thể. Hàm Softmax giúp chuyển đổi kết quả thành xác suất, và do đó, nó rất hữu ích trong các tác vụ phân loại, nơi chúng ta muốn biết xác suất của việc một mẫu thuộc về từng lớp. Trong mạng CNN, tầng cuối cùng thường sử dụng hàm Softmax để trả về xác suất cho các lớp phân loại.

### 3.3.3 Lớp gộp (Pooling layer)

Sau khi qua hàm kích hoạt từ Lớp tích chập, đầu ra (feature map) sẽ được cho qua 1 lớp gộp (Pooling Layer) để làm giảm kích thước dữ liệu tính toán xuống. Việc sử dụng lớp gộp giúp làm tăng tính bất biến, tính ổn định với các biến đổi nhỏ trong đầu vào. Một mạng CNN thường kết hợp nhiều lớp Conv2D và Pooling. Việc áp dụng Pooling sẽ áp dụng trên từng ma trận vuông con kích thước  $f \times f$  với bước nhảy  $s$ .



Hình 7: Average Pooling và Max Pooling với  $f = 2, s = 2$ .

Các lớp gộp phổ biến được sử dụng là Max Pooling, Average Pooling, Global Pooling. Ví dụ khi áp dụng Max Pooling và Average Pooling được minh họa trong Hình 7 với  $f = 2, s = 2$ . Còn Global Pooling đơn giản là áp dụng phép gộp trên ma trận vuông con có kích thước bằng feature map.

### 3.3.4 Lớp làm phẳng (Flatten layer)

Lớp làm phẳng (Flatten Layer) là lớp chuyển đổi lớp ba chiều trong mạng thành vector một chiều.

Ví dụ: một Tensor  $5 \times 5 \times 2$  sẽ được chuyển đổi thành vector có độ dài  $5 \cdot 5 \cdot 2 = 50$ .

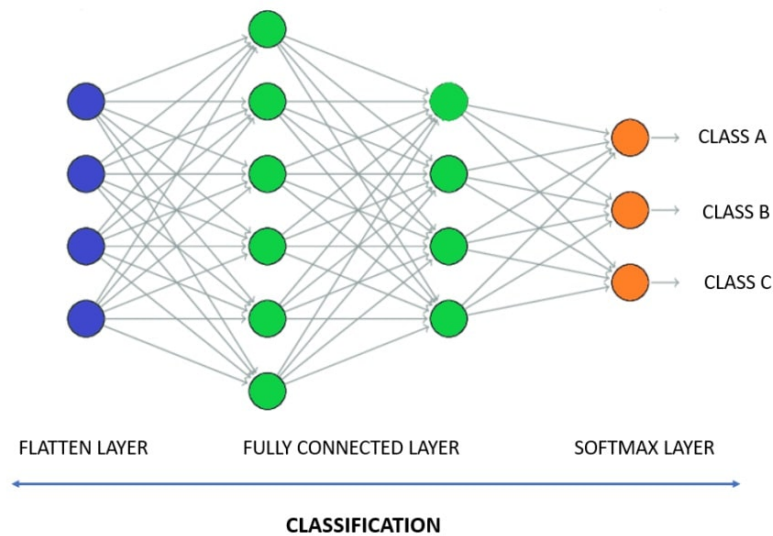
Các lớp chập trước đó của mạng đã trích xuất các đặc điểm từ hình ảnh đầu vào, và bây giờ là lúc phân loại các đặc điểm. Chúng ta sử dụng hàm logistic/softmax để xác định độ tin cậy của các nhãn, hàm này yêu cầu đầu vào 1 chiều. Đây là lý do tại sao lớp làm phẳng là cần thiết.

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

Lớp kết nối đầy đủ (còn được gọi là Lớp ẩn) là lớp cuối cùng trong mạng nơ-ron tích chập. Lớp này là sự kết hợp giữa hàm Affine và hàm phi tuyến tính:

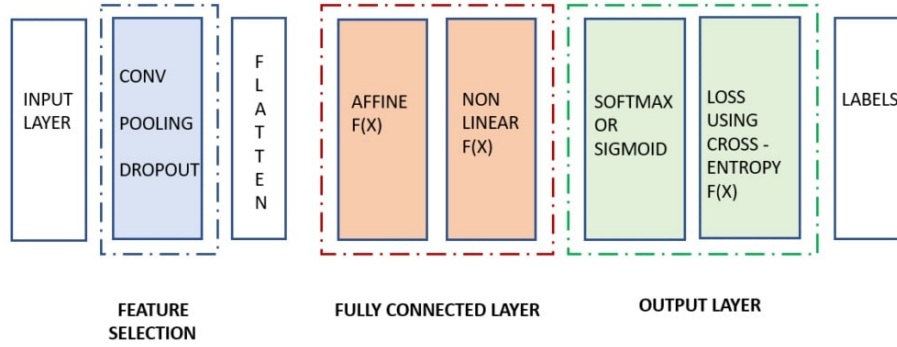
- Hàm Affine :  $y = Wx + b$ .
- Hàm phi tuyến: Sigmoid, Tanh, ReLU.

Lớp kết nối đầy đủ lấy đầu vào từ Lớp Flatten. Dữ liệu đến từ Lớp Flatten trước tiên được chuyển đến hàm Affine và sau đó đến hàm Phi tuyến. Sự kết hợp của 1 hàm Affine và 1 hàm phi tuyến tính được gọi là 1 Fully Connected (Được kết nối hoàn toàn) hoặc 1 lớp ẩn. Lớp kết nối đầy đủ có thể thêm nhiều hidden layer như vậy dựa trên độ sâu mà ta muốn áp dụng mô hình phân loại của mình. Điều này hoàn toàn phụ thuộc vào tập dữ liệu huấn luyện.



Hình 8: Lớp Fully Connected trong CNN.

Sự kết hợp của Lớp Flatten với Lớp Fully Connected và Lớp Softmax chính là mạng nơ-ron sâu (Deep Neural Network). Nếu chúng ta nhìn vào mạng lưới nơ-ron hoàn chỉnh, chúng ta sẽ thấy rằng các lớp ban đầu của mạng lưới nơ-ron tích chập bao gồm lớp tích chập Conv và lớp gộp Pooling. Lớp đầu ra trong mạng nơ-ron tích chập là lớp làm phẳng sử dụng hàm Softmax hoặc Sigmoid và tính toán chi phí (loss) bằng hàm Cross-entropy. Và cuối cùng sẽ có giá trị độ tin cậy của mỗi nhãn, nhãn có độ tin cậy cao nhất là nhãn cuối cùng của ảnh đầu vào. Kiến trúc mạng tổng quan của CNN được trình bày trong Hình 9.



Hình 9: Kiến trúc mạng CNN.

### 3.4 Lan truyền ngược (backpropagation)

#### 3.4.1 Backpropagation trên lớp Fully Connected (Dense)

Bước đầu tiên là tìm Hàm mất mát được sử dụng để đánh giá đầu ra được tạo ra từ lớp đầu ra của Dense Layer (Fully Connected Layer). Việc lựa chọn hàm Loss hoàn toàn phụ thuộc vào mục đích mà bạn muốn mạng thực hiện hay tính chất của bài toán, là phân loại 2 lớp, nhiều lớp hay hồi quy. Để minh họa, chúng ta hãy xem xét hàm loss Binary Cross Entropy:

$$L(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (8)$$

- Đạo hàm theo output  $\hat{y}$ :

$$\begin{aligned} \frac{\partial L}{\partial \hat{y}_i} &= \frac{\partial}{\partial \hat{y}_i} \left( -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \right) \\ &= -\frac{1}{n} \left( \frac{\partial}{\partial \hat{y}_i} (y_i \log(\hat{y}_i)) + \frac{\partial}{\partial \hat{y}_i} ((1 - y_i) \log(1 - \hat{y}_i)) \right) \\ &= -\frac{1}{n} \left( \frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right) \end{aligned}$$

- Đạo hàm theo trọng số  $w$ :

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_{ij}} \quad (9)$$

Ta biết rằng  $\hat{y}_i = x_j \cdot w_{ij} + b_j$ , vậy nên ta có:

$$\frac{\partial \hat{y}_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}(x_j \cdot w_{ij} + b_j) = x_j$$

Lúc đó,  $\frac{\partial L}{\partial w_{ij}}$  trở thành:

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial \hat{y}_i} \cdot x_j = -\frac{1}{n} \left( \frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right) \cdot x_j \quad (10)$$

- **Đạo hàm của loss theo input  $X$ :**

$$\frac{\partial L}{\partial x_j} = \sum_{i=1}^n \frac{\partial L}{\partial \hat{y}_i} \cdot w_{ij} = -\frac{1}{n} \sum_{i=1}^n \left( \frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right) \cdot w_{ij} \quad (11)$$

- **Đạo hàm của loss theo bias:** do  $\frac{\partial \hat{y}_i}{\partial b_j} = \frac{\partial}{\partial b_j}(x_j \cdot w_{ij} + b_j) = \frac{\partial b_j}{\partial b_j} = 1$  nên

$$\frac{\partial L}{\partial b_j} = \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial b_j} = \frac{\partial L}{\partial \hat{y}_i} = -\frac{1}{n} \left( \frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right) \quad (12)$$

### Cập nhật trọng số và bias sử dụng Gradient Descent:

Bước tiếp theo và quan trọng nhất trong quá trình lan truyền ngược là cập nhật trọng số và độ lệch bằng cách sử dụng đạo hàm của hàm loss theo trọng số và độ lệch.

- **Cập nhật trọng số:**

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} - \eta \cdot \frac{\partial L}{\partial w_{ij}} \quad (13)$$

$$= w_{ij}^{\text{old}} + \eta \cdot \left( \frac{1}{n} \left( \frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right) \right) \cdot x_j \quad (14)$$

Trong đó,  $\eta$  là learning rate,  $w_{ij}^{\text{new}}$  là trọng số mới được cập nhật,  $w_{ij}^{\text{old}}$  là trọng số cũ,  $\frac{\partial L}{\partial w_{ij}}$  là gradient của loss theo  $w$ .

- **Cập nhật bias:**

$$b_j^{\text{new}} = b_j^{\text{old}} - \eta \cdot \frac{\partial L}{\partial b_j} \quad (15)$$

$$= b_j^{\text{old}} + \eta \cdot \left( \frac{1}{n} \left( \frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right) \right) \quad (16)$$

Trong đó,  $\eta$  là learning rate,  $b_{ij}^{new}$  là bias mới được cập nhật,  $b_{ij}^{old}$  là bias cũ,  $\frac{\partial L}{\partial b_{ij}}$  là gradient của loss theo bias.

### 3.4.2 Backpropagation trên lớp Pooling

Lớp tiếp theo sau Lớp Dense là Pooling. Vì vậy, bước tiếp theo là chuyển gradient của đầu vào được tạo từ Lớp Dense sang Lớp Pooling. Tuy nhiên, giữa các lớp Dense và Pooling, có thêm một lớp nữa được gọi là lớp Reshape.

Khi nói đến forward, thường thì ta cần chuyển đổi đầu ra 2 chiều sau khi gộp chúng thành vector 1 chiều để áp dụng nó cho lớp Dense vì Dense layer xử lý các đầu vào vector 1 chiều. Vì vậy, trong quá trình ngược lại, chúng ta cần đảo ngược quá trình này, đó là chuyển đổi vector gradient 1 chiều từ Lớp Dense sang ma trận gradient 2D cho lớp Pooling và lớp tích chập.

$$\begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \\ \frac{\partial L}{\partial x_3} \\ \vdots \\ \frac{\partial L}{\partial x_j} \end{bmatrix} \xrightarrow{\text{Reshape}} \begin{bmatrix} \frac{\partial L}{\partial x_{11}} & \frac{\partial L}{\partial x_{12}} & \frac{\partial L}{\partial x_{13}} \\ \frac{\partial L}{\partial x_{21}} & \frac{\partial L}{\partial x_{22}} & \frac{\partial L}{\partial x_{23}} \\ \frac{\partial L}{\partial x_{31}} & \frac{\partial L}{\partial x_{32}} & \frac{\partial L}{\partial x_{33}} \\ \vdots & \vdots & \vdots \\ \frac{\partial L}{\partial x_{i,j}} & \dots & \dots \end{bmatrix}$$

Quá trình reshape trong quá trình backward có thể được xác định dựa trên kích cỡ của các lớp Pooling trong quá trình forward. Kích cỡ cụ thể của các lớp Pooling sẽ xác định cách chuyển đổi vector gradient 1 chiều từ lớp Dense trở lại thành ma trận gradient 2 chiều.

Khi xem xét Max-Pooling hoặc Pooling nói chung, cần lưu ý rằng không có phép tính gradient thực tế nào được thực hiện, thay vào đó, chuyển gradient ngược lại theo hàm Pooling được sử dụng. Trong trường hợp Max Pooling, bạn sẽ chọn giá trị tối đa từ mỗi ô và tạo đầu ra trong quá trình chuyển tiếp. Trong backpropagation, gradient tương ứng với các giá trị tối đa được gửi ngược và phần còn lại được coi là 0.

$$\frac{\partial L}{\partial x_{ij}} = \frac{\partial L}{\partial y_{ij}} \cdot \delta_{ij} \quad (17)$$

$$\delta_{ij} = \begin{cases} 1 & \text{nếu là vị trí chứa giá trị max.} \\ 0 & \text{nếu không phải.} \end{cases} \quad (18)$$

Ở đây  $\delta_{ij}$  được gọi là mask hoặc switch được sử dụng để chuyển đổi giữa 1 và 0 cho các vị trí khác nhau trong ma trận gradient 2 chiều. Nếu giá trị trở thành 1 thì gradient

được truyền ngược lại và nếu giá trị trở thành 0 thì gradient không được truyền. Giá trị trở thành 1 trên các vị trí có giá trị lớn nhất được tìm thấy và các giá trị còn lại bằng 0.

### 3.4.3 Backpropagation trên lớp tích chập

Sau khi đã trình bày các bước trước đó, chúng ta đã đến bước cuối cùng trong lan truyền ngược đó là xử lý trên Lớp tích chập. Ở đây, các đạo hàm backward cho Lớp Dense và Lớp tích chập có một số điểm tương đồng. Sự khác biệt duy nhất là ở đây chúng ta sử dụng phép tích chập thay vì phép nhân ma trận.

Tương tự như Dense layer, chúng ta cần tìm đạo hàm đối với trọng số (weight, ở đây là kernel/filter), đầu vào (input), và độ lệch (bias). Do tính chất 2 chiều của lớp tích chập, chúng ta cần sử dụng ba chỉ số để theo dõi các chiều. Các chỉ số này bao gồm chỉ số cho chiều cao và chiều rộng của kernel, hình ảnh đầu vào, và đầu ra, và sau đó là chỉ số được sử dụng để theo dõi vị trí của kernel trên hình ảnh đầu vào. Ngoài ra, thay vì thực hiện phép nhân ma trận như trong lớp Dense, lớp tích chập sử dụng một phép tích chập. Phép tích chập này bao gồm việc sử dụng kernel hoặc bộ lọc trên ma trận đầu vào.

- Đạo hàm theo filter/kernel:

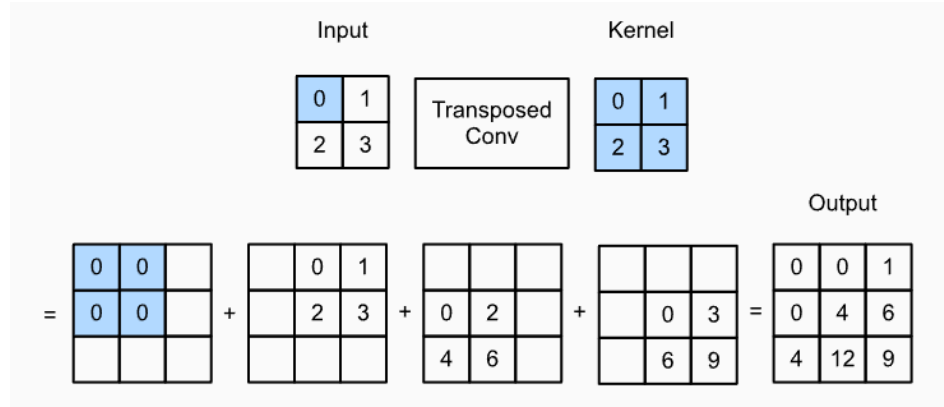
$$\frac{\partial L}{\partial K} = \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial K} = \frac{\partial L}{\partial y} \times X \quad (19)$$

Ở đây, phép  $\times$  là phép tích chập, phép toán trên tương đương với việc nhân chập ảnh đầu vào  $x$  với ảnh  $\frac{\partial L}{\partial y}$  (coi  $\frac{\partial L}{\partial y}$  là kernel).

- Đạo hàm của loss theo input  $X$ :

$$\begin{aligned} \frac{\partial y}{\partial X} &= K \\ \frac{\partial L}{\partial X} &= \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial X} = \frac{\partial L}{\partial y} \times K \end{aligned}$$

Phép toán tích chập ở trên còn được gọi là phép tích chập chuyển vị (Transposed convolution), giúp biến ảnh kích thước nhỏ thành ảnh kích thước lớn hơn. Ví dụ minh họa về phép tích chập chuyển vị được trình bày ở Hình 10.



Hình 10: Ví dụ phép tích chập chuyển vị.

- Đạo hàm của loss theo bias

$$\frac{\partial L}{\partial b_j} = \sum_{j=1}^m \frac{\partial L}{\partial y_{ij}}. \quad (20)$$

**Cập nhật Kernels và Bias sử dụng Gradient Descent:** Sau khi đã tính được đạo hàm theo kernel, input và bias, ta sẽ thực hiện cập nhật trọng số và bias dựa trên Gradient Descent.

- Cập nhật Kernels:

$$K_{ijk}^{\text{new}} = K_{ijk}^{\text{old}} - \eta \cdot \frac{\partial L}{\partial K_{ijk}}. \quad (21)$$

Trong đó,  $\eta$  là learning rate,  $K_{ijk}^{\text{new}}$  là kernel mới được cập nhật,  $K_{ijk}^{\text{old}}$  là kernel cũ,  $\frac{\partial L}{\partial K_{ijk}}$  là gradient của loss theo kernel.

- Cập nhật Bias

$$b_j^{\text{new}} = b_j^{\text{old}} - \eta \cdot \frac{\partial L}{\partial b_j}. \quad (22)$$

Trong đó,  $\eta$  là learning rate,  $b_j^{\text{new}}$  là bias mới được cập nhật,  $b_j^{\text{old}}$  là bias cũ,  $\frac{\partial L}{\partial b_j}$  là gradient của loss theo bias.

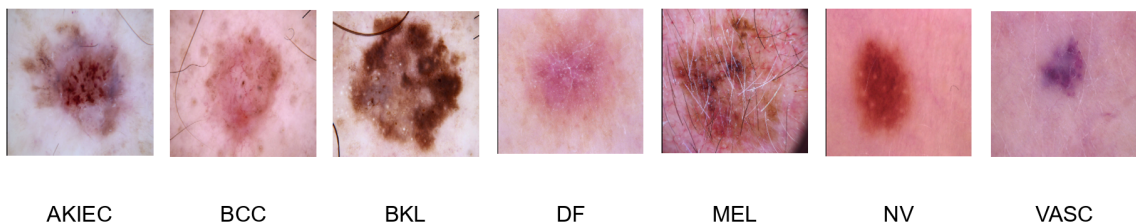


## 4 Kết quả thực nghiệm và đánh giá

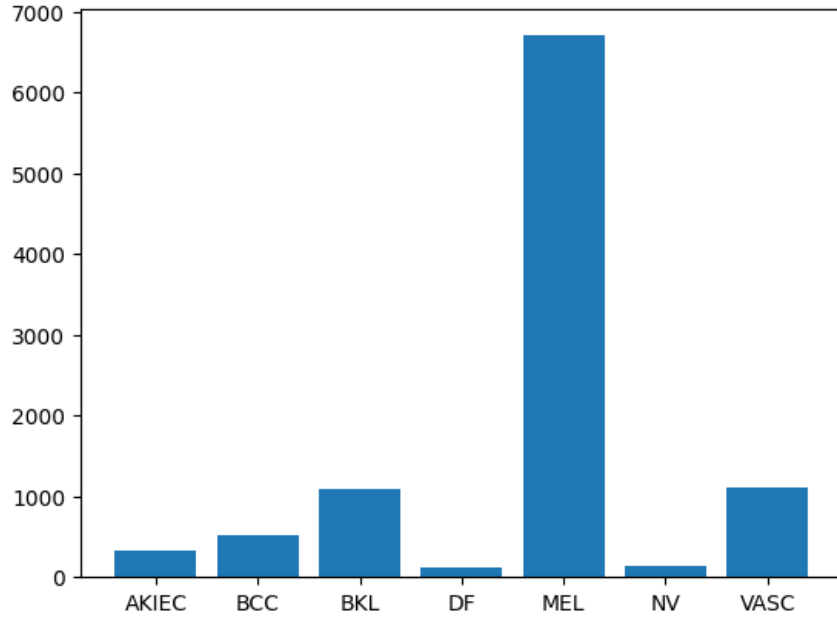
### 4.1 Về bộ dữ liệu HAM10000

Bộ dữ liệu Skin Cancer MNIST: HAM10000 [1] là bộ dữ liệu ảnh nội soi da, gồm 10015 ảnh nội soi kích thước  $450 \times 600$ . Bộ dữ liệu gồm 7 loại chẩn đoán (nhãn), hình ảnh mẫu được trình bày trong Hình 11 và biểu đồ thể hiện số lượng ảnh mỗi loại trong Hình 12:

1. Melanocytic nevi (NV): Bao gồm các nốt ruồi và khuyết điểm melanocytic.
2. Melanoma (MEL): Đây là một loại ung thư da, thường là do sự phát triển không bình thường của tế bào melanocytic. Nó là loại nguy hiểm nhất của ung thư da và cần được phát hiện và điều trị sớm.
3. Benign keratosis (BKL): Bao gồm các khuyết điểm da không ung thư. Đây có thể là những vết đốm, tảo biển, hoặc dấu vết khác trên da.
4. Basal cell carcinoma (BCC): BCC là một loại ung thư tuyến da cơ bản. BCC không phát triển nhanh và thường không lan rộng, nhưng cần được loại bỏ để tránh gây tổn thương hoặc biến chứng.
5. Actinic keratoses (AKIEC): Đây là một loại tác động tiền ung thư đối với da. Nó thường là kết quả của hư hại da do tác động của tia tử ngoại mặt trời.
6. Vascular lesions (VASC): VASC gồm bất kỳ sự thay đổi nào trong các mạch máu da, bao gồm các sự thay đổi về màu sắc và cấu trúc.
7. Dermatofibroma (DF): Đây là một khối u da, thường không nguy hiểm. Tuy nhiên, nó có thể gây khó chịu và thỉnh thoảng cần loại bỏ.



Hình 11: Ví dụ các mẫu ung thư da trong bộ dữ liệu HAM10000.



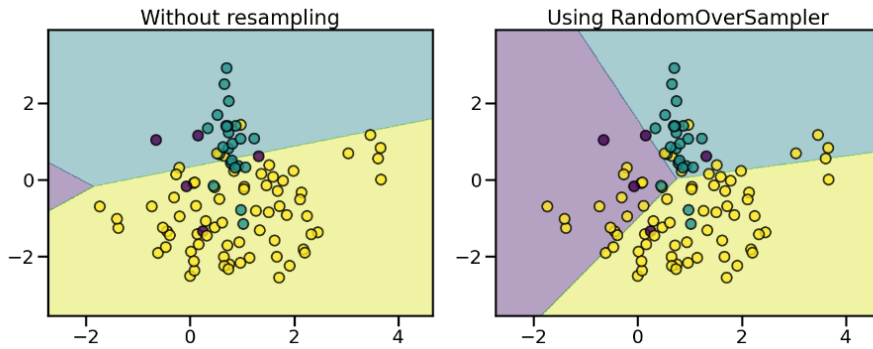
Hình 12: Biểu đồ histogram thể số lượng ảnh mỗi loại chẩn đoán trong dữ liệu.

Chúng tôi sẽ sử dụng bộ dữ liệu đã được thu nhỏ về kích thước  $28 \times 28 \times 3$  từ bộ dữ liệu HAM10000 trên Kaggle. Trong quá trình huấn luyện mô hình, bộ dữ liệu sẽ được chia ra thành 2 phần gồm 80% dữ liệu train (8012) và 20% dữ liệu test (2003).

## 4.2 Imbalanced-learn

Như biểu đồ histogram ở Hình 12 thể hiện, ta thấy đây là một bộ dữ liệu bất cân bằng, khi số lượng mẫu của da ung thư Melanoma nhiều hơn hẳn so với các loại dấu hiệu ung thư khác (xấp xỉ 66.9%). Vì vậy, trong quá trình huấn luyện mô hình để mang lại một mô hình có hiệu suất cao, việc xử vấn đề về mất cân bằng dữ liệu là điều cần thiết. Như trong nghiên cứu của Maryam Tahir [22] đã sử dụng kỹ thuật xử lý bất cân bằng dữ liệu SMOTE (synthetic minority oversampling technique) để tăng hiệu suất của mô hình DSCC\_Net.

Trong nghiên cứu này, chúng tôi sử dụng một kỹ thuật khác là Random Over-Sampling. Kỹ thuật này đơn giản là tạo ra các mẫu mới trong các lớp có ít dữ liệu. Chiến lược đơn giản nhất là tạo ra các mẫu mới bằng cách lấy mẫu ngẫu nhiên và thay thế các mẫu hiện có. Nhờ vậy, lớp đa số không thay thế các lớp khác trong quá trình huấn luyện. Do đó, tất cả các lớp đều được biểu diễn bằng hàm quyết định. Để thuận tiện, chúng tôi sử dụng hàm RandomOverSampler để sử dụng kỹ thuật Naive random over-sampling từ thư viện imbalanced-learn [26]. Kết quả sau khi sử dụng hàm RandomOverSampler với hàm quyết định Logistic được minh họa trong Hình 13.



Hình 13: Áp dụng kỹ thuật Random Over-Sampling với hàm logistic.

### 4.3 Thiết kế mô hình mạng CNN

Để thử nghiệm mô hình và áp dụng vào bộ dữ liệu HAM10000, chúng tôi xây dựng một mạng CNN sử dụng thư viện Tensorflow Keras. Mô hình xây dựng chứa tổng cộng 1275079 biến. Một số siêu tham số được đặt khi huấn luyện mô hình:

- Số lượng epochs: 50.
- Kích thước batch\_size khi huấn luyện: 128.
- Phương thức regularization: Reduce learning rate khi một metric không cải thiện.
- Optimizer: Adamax với learning\_rate = 0.001.
- Hàm mất mát: categorical crossentropy.

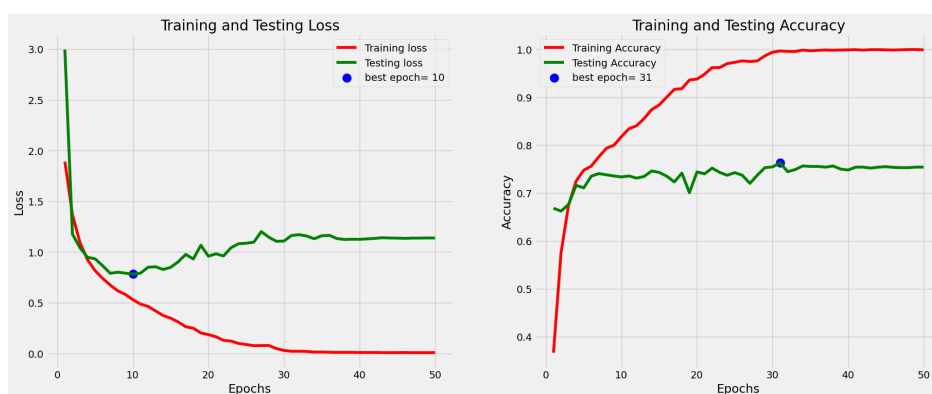
Mạng CNN được thiết kế bao gồm 7 lớp tích chập Conv2D, 4 lớp Max Pooling, 4 lớp Dense. Các lớp tích chập có hàm kích hoạt Relu, kỹ thuật padding là "same" (duy trì kích thước sau khi đệm), khởi tạo kernal bằng 'he\_normal'. Các lớp Dense cũng có hàm kích hoạt là Relu, trừ lớp Dense ở bước phân lớp có hàm kích hoạt là Softmax. Mô hình mạng CNN cụ thể được sử dụng được trình bày ở Bảng 2.

### 4.4 Kết quả

Mô hình được xây dựng khi không xử lý mất cân bằng dữ liệu được huấn luyện sau 50 epochs có kết quả accuracy tốt nhất đạt 99.90% trên tập dữ liệu train và 76.29% trên bộ dữ liệu test. Kết quả quá trình huấn luyện được hiển thị trong Hình 14.

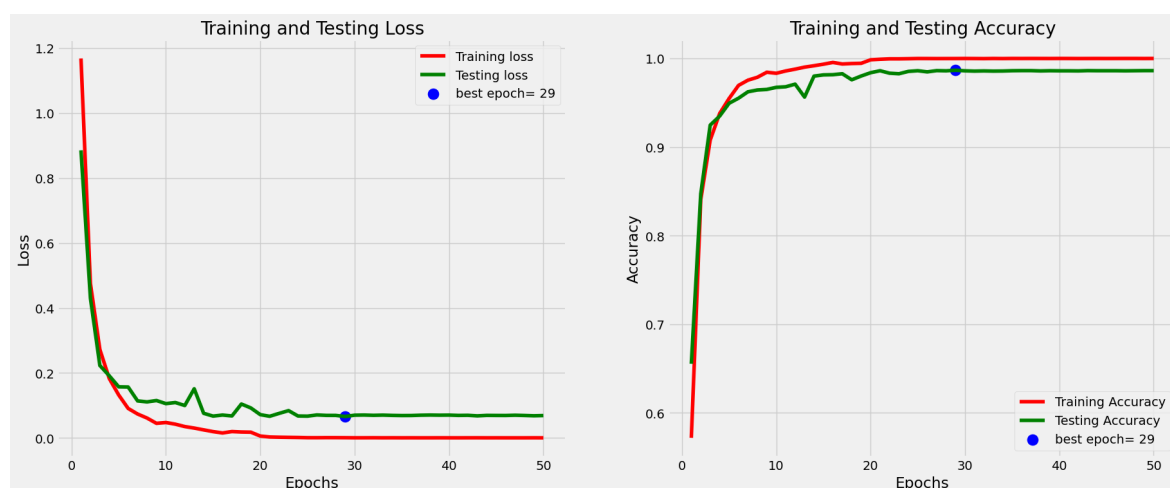
Layer (type)	Output Shape	Params
Conv2D	(None, 28, 28, 32)	896
MaxPooling2D	(None, 14, 14, 32)	0
BatchNormalization	(None, 14, 14, 32)	128
Conv2D	(None, 14, 14, 64)	18496
Conv2D	(None, 14, 14, 64)	36928
MaxPooling2D	(None, 7, 7, 64)	0
BatchNormalization	(None, 7, 7, 64)	256
Conv2D	(None, 7, 7, 128)	73856
Conv2D	(None, 7, 7, 128)	147584
MaxPooling2D	(None, 3, 3, 128)	0
BatchNormalization	(None, 3, 3, 128)	512
Conv2D	(None, 3, 3, 256)	295168
Conv2D	(None, 3, 3, 256)	590080
MaxPooling2D	(None, 1, 1, 256)	0
Flatten	(None, 256)	0
DropOut	(None, 256)	0
Dense Layer (None, 256)	65792	
BatchNormalization	(None, 256)	1024
Dense Layer (None, 128)	32896	
BatchNormalization	(None, 128)	512
Dense Layer (None, 64)	8256	
BatchNormalization	(None, 64)	256
Dense Layer	(None, 32)	2080
BatchNormalization	(None, 32)	128
classifier (Dense)	(None, 7)	231

Bảng 2: Cấu trúc mạng CNN sử dụng để huấn luyện.



Hình 14: Kết quả huấn luyện với mô hình mạng cơ sở.

Sau khi sử dụng kỹ thuật Random Over-Sampling, dữ liệu mới thu được bao gồm 46935 ảnh với 6705 ảnh cho mỗi loại bệnh. Mô hình được xây dựng ở trên được sử dụng để huấn luyện với bộ dữ liệu này (sau khi phân thành 2 phần gồm 80% train và 20% test) giúp cải thiện accuracy lên 100% đối với tập dữ liệu train và 98.67% đối với dữ liệu test. Điều này chứng tỏ hiệu quả của việc sử dụng kỹ thuật xử lý mất cân bằng dữ liệu khi áp dụng nó cho việc phân loại các loại bệnh ung thư da. Quá trình huấn luyện được trình bày cụ thể trong Hình 15.



Hình 15: Kết quả huấn luyện với mô hình trên bộ dữ liệu thu được sau khi xử lý mất cân bằng dữ liệu.

Chúng tôi thực nghiệm thêm bằng việc sử dụng mô hình huấn luyện trên 80% bộ dữ liệu train thu được từ việc xử lý bất cân bằng dữ liệu để dự đoán 20% dữ liệu test ban đầu, ta vẫn thu được kết quả tốt với accuracy tốt nhất đạt 98.75% trên 20% bộ dữ liệu test này.

## 4.5 Đánh giá và nhận xét

Qua thực nghiệm ở trên, ta thấy được độ hiệu quả của các mô hình mạng nơ-ron tích chập (CNN) vào các bài toán phân loại ảnh, ở nghiên cứu này là phân loại dựa trên các dấu hiệu ung thư da với bộ dữ liệu HAM10000. Mô hình mạng CNN đề xuất mang lại kết quả vô cùng tốt trên bộ dữ liệu train, tuy vậy vì thế dẫn đến hiện tượng overfitting và có kết quả không tốt trên bộ dữ liệu test, chỉ đạt 76.29%. Tuy vậy, mô hình huấn luyện tương đối nhanh, chỉ mất vọt vọt khoảng 1s đối với mỗi epoch.

Với việc nhận thấy đây là một bộ dữ liệu bất cân bằng, ta đã áp dụng thêm kỹ thuật Random Over-Sampling, một kỹ thuật đơn giản để xử lý bất cân bằng dữ liệu, ta thấy được sự hiệu quả sau khi xử lý bất cân bằng dữ liệu đối với bộ dữ liệu HAM10000,

nó giúp tăng hơn 20% độ chính xác của kết quả. Tuy vậy, không phải bộ dữ liệu nào cũng đem lại sự hiệu quả như vậy, với những bộ dữ liệu không bất cân bằng, kỹ thuật Random Over-Sampling sẽ không cải thiện được nhiều hiệu suất của mô hình.

Ngoài việc xử lý bất cân bằng dữ liệu, ở các bài toán phân loại ảnh như trên, chúng ta có thể sử dụng thêm những kỹ thuật tiền xử lý ảnh, hoặc áp dụng thêm một số thuật toán học máy khác như nghiên cứu của W. M. Salamaa [19] để tăng hiệu suất của mô hình. Tuy vậy ở thực nghiệm này, nhận thấy độ chính xác của mô hình đã rất tốt sau khi sử dụng kỹ thuật mất cân bằng nên chúng tôi đã lược bỏ đi các bước tiền xử lý ảnh và các kỹ thuật học máy để đơn giản hóa mô hình và giảm thiểu thời gian huấn luyện mô hình.

## Kết luận

Trong bài báo cáo này, chúng tôi đã tìm hiểu tổng quan về cấu trúc mạng nơ-ron tích chập (Convolutional Neural Networks - CNN), về bộ dữ liệu HAM10000. Sau đó, xây dựng một mạng nơ-ron tích chập để huấn luyện mô hình trên bộ dữ liệu HAM10000. Mạng nơ-ron được đề xuất được huấn luyện trong thời gian ngắn và mang lại độ chính xác cao trên bộ dữ liệu train. Để cải tiến độ chính xác của mô hình trên bộ dữ liệu test, chúng tôi đã sử dụng thêm kỹ thuật Random Over-Sampling để xử lý bất cân bằng dữ liệu giúp tăng độ chính xác của mô hình trên bộ dữ liệu test lên hơn 20% và đạt 98.67%. Mô hình được xây dựng dù không được áp dụng thêm các kỹ thuật tiền xử lý ảnh khác hay các thuật toán học máy, tuy vậy lại mang lại độ chính xác cao hơn các mô hình trong các nghiên cứu gần đây. Qua đó, ta thấy được sự hiệu quả của việc xử lý bất cân bằng dữ liệu khi phân loại các mức độ ung thư da trên bộ dữ liệu HAM10000.

Qua nghiên cứu này, chúng ta đã hiểu thêm về mạng CNN, cấu trúc mạng CNN và cách thức mạng CNN được huấn luyện và áp dụng vào các bài toán phân loại ảnh. Thông qua việc thực nghiệm trên bộ dữ liệu HAM10000, ta thấy độ hiệu quả của mạng CNN trong các bài toán phân loại ảnh, ở đây là phân loại các bệnh ung thư da để từ đó đưa ra chẩn đoán sớm và chính xác. Qua đây, ta cũng thấy được việc áp dụng các kỹ thuật, công nghệ của trí tuệ nhân tạo trong các bài toán thực tế nhằm giảm thiểu việc lao động chân tay của con người, tiết kiệm chi phí và tăng hiệu suất công việc ngày càng phổ biến và đem lại hiệu quả cao rõ rệt.

## Các nguồn tài liệu tham khảo

- [1] P. Tschandl, “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions,” 2018.
- [2] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, “Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA: A Cancer Journal for Clinicians*, vol. 71, no. 3, pp. 209–249, 2021.
- [3] A. C. Society, “Cancer facts & figures 2018,” *Atlanta, American Cancer Society*, 2018.
- [4] H. K. Koh, “Melanoma Screening: Focusing the Public Health Journey,” *Archives of Dermatology*, vol. 143, pp. 101–103, 01 2007.
- [5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–44, 05 2015.
- [6] B. Zhang, W. Yue, Q. Liu, and S. Hu, “Application of artificial intelligence in medical imaging diagnosis,” in *Proceedings of the 2021 International Conference on Bioinformatics and Intelligent Computing*, BIC 2021, (New York, NY, USA), p. 401–406, Association for Computing Machinery, 2021.
- [7] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, 02 2018.
- [8] M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, and R. H. Moss, “A methodological approach to the classification of dermoscopy images,” *Computerized Medical Imaging and Graphics*, vol. 31, no. 6, pp. 362–373, 2007.
- [9] I. Maglogiannis and C. N. Doukas, “Overview of advanced computer vision systems for skin lesions characterization,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 721–733, 2009.
- [10] M. E. Celebi, H. Iyatomi, W. V. Stoecker, R. H. Moss, H. S. Rabinovitz, G. Argenziano, and H. P. Soyer, “Automatic detection of blue-white veil and related structures in dermoscopy images,” *Computerized Medical Imaging and Graphics*, vol. 32, no. 8, pp. 670–677, 2008.



- [11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- [12] L. Deng, “The mnist database of handwritten digit images for machine learning research [best of the web],” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.
- [14] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, T. Duerig, and V. Ferrari, “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale,” *International Journal of Computer Vision*, vol. 128, 03 2020.
- [15] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.
- [16] A. Rezvantlab, H. Safigholi, and S. Karimijeshni, “Dermatologist level dermoscopy skin cancer classification using different deep learning convolutional neural networks algorithms,” 2018.
- [17] S. K. Datta, M. A. Shaikh, S. N. Srihari, and M. Gao, “Soft-attention improves skin cancer classification performance,” 2021.
- [18] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, and A. Halpern, “Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic),” in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 168–172, 2018.
- [19] W. Salama and M. Aly, “Deep learning design for benign and malignant classification of skin lesions: a new approach,” *Multimedia Tools and Applications*, vol. 80, pp. 1–17, 07 2021.

- [20] Z. Lan, S. Cai, X. He, and X. Wen, “Fixcaps: An improved capsules network for diagnosis of skin cancer,” *IEEE Access*, vol. 10, pp. 76261–76267, 2022.
- [21] X. Lu and Y. A. F. A. Zadeh, “Deep learning-based classification for melanoma detection using xceptionnet,” *Journal of Healthcare Engineering*, vol. 2022, 2022.
- [22] M. Tahir, A. Naeem, H. Malik, J. Tanveer, R. A. Naqvi, and S.-W. Lee, “Dscn\_net: Multi-classification deep learning models for diagnosing of skin cancer using dermoscopic images,” *Cancers*, vol. 15, no. 7, 2023.
- [23] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.
- [24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [25] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Neural Information Processing Systems*, vol. 25, 01 2012.
- [26] G. Lemaitre, F. Nogueira, and C. K. Aridas, “Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning,” *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.
- [27] L. Carin and M. J. Pencina, “On Deep Learning for Medical Image Analysis,” *JAMA*, vol. 320, pp. 1192–1193, 09 2018.
- [28] P. Lakhani and B. Sundaram, “Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks,” *Radiology*, vol. 284, no. 2, pp. 574–582, 2017. PMID: 28436741.