

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

------*---*---*---*



BÁO CÁO ĐỒ ÁN

CHỦ ĐỀ : FASHION CLASSIFICATION

Môn học: Thị Giác Máy Tính

Giảng viên hướng dẫn: Th.Mai Tiến Dũng

Sinh viên:

- Ngô Đức Lộc – 18520924
- Lê Thành Đạt – 17520332

Thành phố Hồ Chí Minh, ngày 09 tháng 01 năm 2022

Lời cảm ơn

Lời đầu tiên nhóm xin chân thành cảm ơn thầy Mai Tiến Dũng-khoa Khoa Học Máy Tính – trường Đại học Công nghệ Thông tin, đã truyền đạt kiến thức bổ ích, hỗ trợ và giải đáp thắc mắc cho nhóm trong suốt quá trình học tập và thực hiện đồ án môn Thị Giác Máy Tính. Trong quá trình hiện thực đồ án, nhóm đã tổng hợp nhiều kiến thức từ những nguồn thông tin tin cậy nhưng cũng không thể tránh được hết tất cả những sai sót. Nhóm rất mong nhận được những góp ý từ thầy và các bạn.

Thành phố Hồ Chí Minh, ngày 09 tháng 01 năm 2022

Mục lục

Lời cảm ơn.....	2
Mục lục	3
Danh sách hình ảnh minh họa.....	5
Bảng phân công.....	6
Danh mục từ viết tắt.....	6
I.GIỚI THIỆU	7
1.1. Đặt vấn đề	7
1.2. Lý do thực hiện đề tài.....	7
II.HƯỚNG TIẾP CẬN	8
2.1. Mục đích nghiên cứu.....	8
2.2. Phạm vi và ngôn ngữ lập trình sử dụng	8
III.Bộ dữ liệu và nội dung lý thuyết	8
3.1. Mô tả bài toán.....	8
3.2. Về bộ dữ liệu	9
3.3. Tổng quan về mạng nơ ron tích chập được đề xuất	9
3.4. Mạng nơ ron tích chập CNNs	10
3.4.1. Convolutional layer	11
3.4.2. Pooling layer	12
3.4.3. Fully connected layer.....	12
IV.Mô phỏng	13
4.1. Khám phá bộ dữ liệu Fashion-MNIST	13
4.1.1 Import libraries	13
4.1.2 Trích xuất tập dữ liệu.....	13
4.1.3 Tính năng	13

4.1.4 Iexamine dimensions	14
4.1.5 Kiểm tra các giá trị null	14
4.2. Trực quan hóa dữ liệu	14
4.2.1 Lập đồ thị hình ảnh ngẫu nhiên	14
4.2.2 Phân phối mô tả	14
4.3. Chuẩn bị dữ liệu	15
4.3.1 Tách dữ liệu	15
4.3.2 Định hình lại kích thước hình ảnh:	15
4.3.3 Chuẩn hóa:	15
4.4. Huấn luyện mô hình	16
4.4.1 Xây dựng mô hình	16
4.4.2 Biên dịch mô hình.....	16
4.4.3 Tóm tắt mô hình.....	17
4.4.4 Đánh giá mô hình.....	18
4.4.5 Báo cáo dự đoán và phân loại.....	18
V. KẾT QUẢ.....	18
VI.DEMO	19
VII. HÌNH ẢNH THỰC TIỄN.....	20
VII. TỔNG QUÁT	21
VII. TÀI LIỆU THAM KHẢO	22

Danh sách hình ảnh minh họa

Hình 1 : Minh họa bộ dữ liệu Fashion-MNIST

Hình 2 : Tập các nhãn

Hình 3 : Biểu diễn kiến trúc mạng CNN

Hình 4 : Hiển thị một trong những hình ảnh từ tập dữ liệu đào tạo

Hình 5 : Bảng tóm tắt mô hình

Hình 6 : Biểu đồ training and validation accuracy, loss

Hình 7 : Demo

Hình 8 : Ảnh túi xách

Hình 9 : Kết quả nhãn

Hình 10 : Biểu đồ kết quả dự đoán hình ảnh thực tế

Bảng phân công

Sinh viên	Phân Công
Ngô Đức Lộc	Thực hiện code, viết báo cáo, thuyết trình
Lê Thành Đạt	Chỉnh sửa, làm slide, thuyết trình

Danh mục từ viết tắt

Từ viết tắt	Ý nghĩa
CV	Computer Vision
ML	Machine Learning

I.GIỚI THIỆU

1.1. Đặt vấn đề

Trong thời đại công nghệ số hiện nay, những gì liên quan đến công nghệ đều thu hút sự chú ý của mọi người, đặc biệt là những công nghệ phục vụ tích cực trong cuộc sống hằng ngày. Trong số đó, Machine Learning là một trong những nền tảng công nghệ không thể không nhắc đến. Ứng dụng của Computer Vision không ngừng phát triển và đóng vai trò chủ chốt trong nhiều ứng dụng quan trọng.

Hầu hết mọi ngành công nghiệp làm việc liên quan đến nhận diện đều nhận ra tầm quan trọng của Computer Vision. Những cái nhìn sáng suốt từ nguồn dữ liệu này – chủ yếu dưới dạng thời gian thực – sẽ giúp các tổ chức vận hành hiệu quả hoặc tạo lợi thế lớn giữa các đối thủ cạnh tranh.

Các ứng dụng của Computer Vision đã quá quen thuộc đối với con người :, hệ thống tag khuôn mặt trên Facebook ,hệ thống nhận diện dấu vân tay, hệ thống phân loại hình ảnh ,...chỉ là một trong những ứng dụng trong vô vàn những ứng dụng khác của Computer Vision trong cuộc sống hằng ngày.

Xu hướng phát triển công nghệ thông tin ngày càng tăng, song song với nó lượng dữ liệu sinh ra cũng ngày càng lớn. Vì vậy, nhu cầu để giải quyết dữ liệu sinh ra cũng ngày càng lớn hơn, CV đang góp phần giải quyết những vấn đề này. Một trong những bài toán đang được nhóm chúng em quan tâm trong lĩnh vực CV này đó là Phân loại trang phục.

Ứng dụng của bài toán này là trong lĩnh vực thời trang. Thời trang là một sự thể hiện thẩm mỹ phổ biến tại một thời gian, địa điểm cụ thể, trong một bối cảnh cụ thể bằng cách sử dụng quần áo, giày dép,...Thời trang rất đa dạng, nó phụ thuộc vào từng người, từng phong cách. Vì vậy, việc phân loại thời trang là rất cần thiết.

1.2. Lý do thực hiện đề tài

Công nghệ thông tin ngày càng phát triển và có vai trò hết sức quan trọng không thể thiếu trong cuộc sống hiện đại. Con người ngày càng tạo ra những cỗ máy thông minh có khả năng tự nhận biết và xử lý được các công việc một cách tự động, phục vụ cho lợi ích của con người.

Trong những năm gần đây, một trong những bài toán nhận được nhiều sự quan tâm chính là bài toán phân loại và nó đã được áp dụng ở rất nhiều lĩnh vực như phân loại xe, con vật, đồ vật.. Và đối với phân loại trang phục chúng ta có thể cho máy tính học được cách phân loại các loại trang phục khác nhau...

Với sự hấp dẫn của bài toán và những kiến thức được học, nhóm chúng em đã chọn đề tài “Fashion Classification” (Phân loại trang phục) để làm đồ án môn học của mình.

II.HƯỚNG TIẾP CẬN

2.1. Mục đích nghiên cứu

- _ Tìm hiểu sơ về bộ dữ liệu fashion- mnist.
- _ Nghiên cứu, tìm hiểu và đề xuất hướng giải quyết về vấn đề phân loại các loại trang phục trong bộ dữ liệu.
- _ Đánh giá các vấn đề xảy ra đối với dữ liệu trong bộ dữ liệu và dữ liệu ngoài.
- _ Nghiên cứu này nhằm mục đích phân loại chính xác hình ảnh của các loại trang phục với độ chính xác từ 90% trở lên.

2.2. Phạm vi và ngôn ngữ lập trình sử dụng

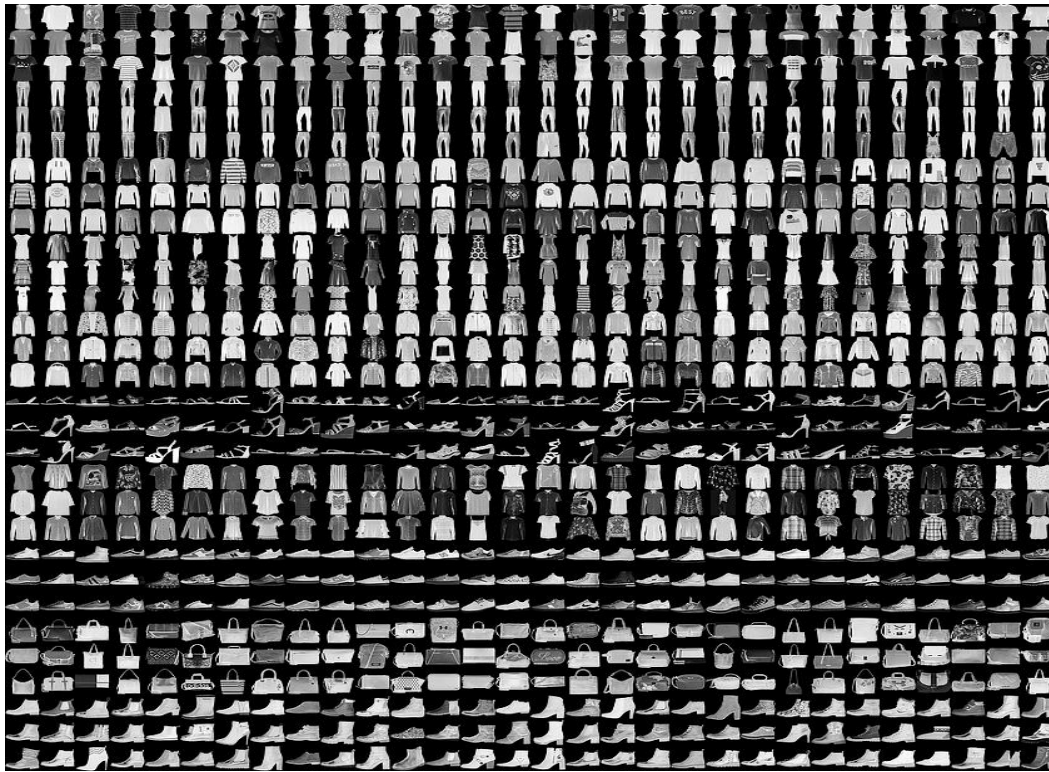
- _ Phạm vi nghiên cứu: Thử nghiệm trên Google Colaboratory
- _ Ngôn ngữ lập trình sử dụng: Python.

III.Bộ dữ liệu và nội dung lý thuyết

3.1. Mô tả bài toán

Ở đồ án này, chúng ta sẽ xây dựng mô hình mạng Neuron phục vụ cho việc phân loại các kiểu trang phục cơ bản trong cuộc sống, qua đó biết được nhu cầu, sở thích của khách hàng trong việc lựa chọn trang phục.

Input : Dataset Fashion-mnist (70000 ảnh trắng đen)



Hình 1 : Minh họa bộ dữ liệu Fashion-MNIST

Mục tiêu : Xây dựng mô hình mạng neuron để phân loại trang phục

Output : Dự đoán loại trang phục

3.2. Về bộ dữ liệu

Fashion MNIST là tập dữ liệu được dùng để thay thế cho tập dữ liệu MNIST kinh điển thường dùng cho các chương trình "Hello, World" của machine learning trong lĩnh vực thị giác máy tính. Tập dữ liệu kinh điển vừa đề cập gồm ảnh của các con số (ví dụ 0, 1, 2) được viết tay. Các ảnh này có cùng định dạng tệp và độ phân giải với các ảnh về quần áo và giày dép chúng ta sắp dùng.

Với tập dữ liệu này, 60.000 ảnh sẽ được dùng để huấn luyện và 10.000 ảnh sẽ thường dùng để đánh giá khả năng phân loại nhận diện ảnh của mạng neuron.

Mỗi ảnh là một mảng NumPy 2 chiều, 28x28, với mỗi pixel có giá trị từ 0 đến 255.

Sau đây là bảng các lớp:

Nhãn	0	1	2	3	4	5	6	7	8	9
Loại	Áo thun	Quần dài	Áo len	Váy	Áo khoác	Sandal	Áo sơ mi	Giày	Túi xách	Ủng

Hình 2 . Tập các nhãn

3.3. Tổng quan về mạng nơ ron tích chập được đề xuất

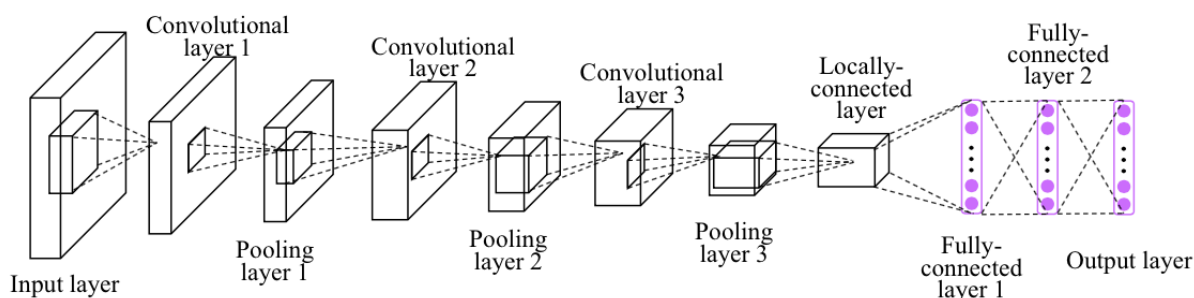
Phân loại hình ảnh là một vấn đề học máy mà các chuyên gia ML để phân loại hình ảnh bằng cách sử dụng phân loại pixel và nhãn của chúng. Thông thường, con người hiểu mô hình dưới dạng ấn tượng trực quan trong khi máy tính hiểu hình ảnh theo cách của pixel. Máy chỉ có thể nhìn thấy số nhị phân từ bất kỳ mô hình nào của một vật thể, giả sử như hình ảnh quả cam. Theo cách hiểu của con người, hình ảnh có thể được biểu diễn bằng các số thập phân từ 0 đến 255. Có những thuật ngữ thiết yếu được sử dụng trong phân loại hình ảnh học sâu. Theo Nair (2018), sau đây là những mối quan tâm quan trọng trong khi xây dựng mạng nơ-ron:

- i. Convolution: đề cập đến hoạt động trên tập dữ liệu hình ảnh để xác định các tính năng cụ thể của hình ảnh đó. Nó giúp làm mờ hình ảnh, làm sắc nét, phát hiện cạnh và giảm nhiễu để cho phép máy học các tính năng hình ảnh cần thiết.
- ii. Pooling: thông thường, hình ảnh thô có thể có kích thước rộng. Do đó, gộp chung sẽ giúp giảm kích thước hình ảnh để tránh làm mất thông kê dữ liệu cần thiết.
- iii. Flattening: Mạng nơ-ron sử dụng một vector đặc trưng để tìm hiểu và phân loại hình ảnh. Mục đích của việc làm phẳng là biến ma trận 2D của các đối tượng hình ảnh thành một đại lượng vector để xây dựng mạng nơ-ron.
- iv. Full connection: đề cập đến quy trình cung cấp các tính năng hình ảnh phẳng vào mạng nơ-ron.

3.4. Mạng nơ ron tích chập CNNs

CNN giống với cách nhận diện hình ảnh của con người. Nó nhận các pixel làm đầu vào hình ảnh, biến đổi chúng bằng cách sử dụng các phép tính toán học và chức năng kích hoạt, sau đó xây dựng một bộ phân loại cho các lớp đầu vào. CNN có một tính năng mã hóa cụ thể trong kiến trúc của nó phù hợp với nó để thực hiện các tác vụ xử lý hình ảnh và thị giác máy tính (O'shea và Nash, 2015).

CNN có thể áp dụng trong các lĩnh vực như nhận dạng hình ảnh, phân tích video, xử lý ngôn ngữ tự nhiên (NLP), khám phá thuốc và phân tích rủi ro trong y tế. Kiến trúc CNN bao gồm ba lớp thiết yếu, đó là the convolutional layer, the pooling layer, and the fully-connected layer. Sử dụng bộ phân loại MNIST, kiến trúc CNN có thể được biểu diễn như sau:



Hình 3 : Biểu diễn kiến trúc mạng CNN

Các lớp kiến trúc và chức năng được giải thích như sau.

3.4.1. Convolutional layer

Lớp Convolution là phần đầu tiên và là một phần quan trọng của kiến trúc CNN tập trung vào việc sử dụng kernels hoặc các tham số bộ lọc của đầu vào hình ảnh. Các kernels có kích thước nhỏ và trải rộng theo chiều sâu của các thông số đầu vào. Ngay lập tức các tham số đầu vào đến lớp convolutional layer, lớp mà nó trải từng pixel trên kích thước đầu vào và ánh xạ chúng thành hai bản đồ định hướng bằng cách sử dụng chức năng kích hoạt.

Ví dụ: hãy xem xét một hình ảnh có kích thước 5 x 5 pixel, lớp tích chập nhân ma trận hình ảnh với ma trận bộ lọc (3 x 3) được gọi là feature map và tạo đầu ra được trình bày bên dưới.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

*

1	0	1
0	1	0
1	0	1

5 x 5 – Image Matrix

3 x 3 – Filter Matrix

Quá trình tích chập (convolution) có ý nghĩa quan trọng trong việc phát hiện cạnh của hình ảnh, làm mờ và làm sắc nét hình ảnh bằng cách kết hợp chúng với ma trận bộ lọc. Chức năng kích hoạt (map) áp dụng cho các bộ lọc tương ứng và quá trình này được lặp lại.

Đầu ra của lớp convolution layer thường được tối ưu hóa để sử dụng các thông số về độ sâu, bước nhảy(stride) và lớp đệm để tránh phức tạp. Bước nhảy (stride) đề cập đến số lượng pixel hình ảnh xoay quanh ma trận đầu vào. Độ sâu của đầu ra tương ứng với số lượng neurons trong lớp tích chập. Giảm tham số độ sâu có thể làm giảm tổng độ phức tạp của nơ-ron, cũng có thể làm giảm độ chính xác của mô hình để nhận dạng chính xác một hình ảnh. Hơn nữa, padding cung cấp khả năng kiểm soát hiệu quả kích thước đầu ra bằng cách đệm các cạnh pixel (O'shea và Nash, 2015).

3.4.2. Pooling layer

Đây là lớp thứ hai của kiến trúc CNN nhằm mục đích giảm thiểu các tham số trong trường hợp pixel ảnh cực kỳ lớn đối với mô hình. Do đó, độ phức tạp tính toán của mô hình cũng được giảm bớt. Lớp này chấp nhận activation map của lớp trước làm tham số đầu vào của nó và chia tỷ lệ nó. Đây được gọi là tổng hợp tối đa (max pooling), một loại tổng hợp không gian sử dụng hàm tối đa. Các kiểu gộp (pooling) khác có thể áp dụng là gộp tổng và gộp trung bình, làm giảm kích thước của activation map trong khi vẫn duy trì các tính năng cần thiết (O's Hea và Nash, 2015).

3.4.3. Fully connected layer

Fully connected layer là lớp cuối cùng của kiến trúc CNN làm phẳng đầu ra của ma trận gộp và đưa nó vào mạng nơ-ron. Các neurons được kết nối với nhau và tạo thành mạng lưới thần kinh nhân tạo. Trong lớp này, có các chức năng kích hoạt SoftMax và sigmoid tóm tắt tất cả chi phí phân loại hình ảnh. Tóm lại, việc phân loại hình ảnh bằng cách sử dụng mạng nơ-ron tích chập là một quy trình chuyển tiếp nguồn cấp dữ liệu theo quy trình sau (Prabhu, 2018):

- Ban đầu, nạp kích thước hình ảnh (pixel) vào lớp convolution layer.
- Chọn chính xác các đặc điểm hình ảnh (thông số), áp dụng các stride và padding nếu cần thiết, và chỉnh sửa hình ảnh bằng cách sử dụng activation map.
- Áp dụng pooling để giảm thiểu kích thước hình ảnh.

- Thêm nhiều convolution layers cho đến khi mô hình đạt được độ chính xác tốt nhất mong muốn.
- Ngoài ra, nó làm phẳng các kết quả và đưa vào lớp cuối cùng được gọi là lớp được kết nối đầy đủ (fully connected layer).
- Cuối cùng, tạo ra kết quả bằng cách sử dụng một chức năng kích hoạt, chức năng này sẽ phân loại hình ảnh đích vào lớp tương ứng.

IV. Mô phỏng

Chương này mô tả việc triển khai mạng nơ-ron tích chập chuyên tiếp với một lớp ẩn duy nhất có khả năng phân loại hình ảnh quần áo thời trang thành mười nhóm riêng biệt bằng cách sử dụng tập dữ liệu Fashion MNIST. Mạng nơ-ron này được viết bằng ngôn ngữ python và sử dụng thư viện học sâu (TensorFlow, Keras). Các thư viện python khác như pandas, NumPy, seaborn và matplotlib cũng được sử dụng để chuẩn bị và trực quan hóa dữ liệu.

4.1. Khám phá bộ dữ liệu Fashion-MNIST

4.1.1 Import libraries:

Như đã đề cập trước đó, mạng nơ-ron tích chập này được thiết kế bằng python 3, Keras và TensorFlow. Thiết kế được thực hiện trên Google Colaborary.

4.1.2 Trích xuất tập dữ liệu:

Tập dữ liệu Fashion - MNIST được truy xuất từ trang web Kaggle. Tập dữ liệu bao gồm dữ liệu về huấn luyện và dữ liệu thử nghiệm. Thay vì tải dữ liệu qua website Kaggle, ta sẽ load dữ liệu trực tiếp từ chính thư viện Keras đã được tích hợp sẵn.

4.1.3 Tính năng:

Có hai tính năng thiết yếu của tập dữ liệu Fashion-MNIST. Chúng bao gồm các nhãn hình ảnh và các pixel thể hiện kích thước hình ảnh. Thông thường, con người sử dụng mắt để nhìn, xử lý và phân biệt các hình ảnh vật thể. Trong mắt người tồn tại các tế bào thụ cảm có chức năng xử lý các pixel đối tượng và thể hiện nó dưới dạng hình ảnh.

Tương tự như vậy, máy tính (máy) hiểu các bức ảnh là một mảng hai chiều các số nguyên (2D pixel). Một máy có thể sử dụng thang độ xám hoặc thang đo RGB để hiểu ranh giới hình ảnh của một đối tượng.

4.1.4 Iexamine dimensions

: Có 60.000 hình ảnh cho tập huấn luyện và 10.000 hình ảnh cho tập thử nghiệm. Tất cả các hình ảnh được thể hiện là hình có thang độ xám 28 x 28 tương ứng với nhãn quần áo của tập dữ liệu. Các pixel là các giá trị số nguyên từ 0 đến 255. Các nhãn được mã hóa bằng các giá trị nguyên từ 0 đến 9, mỗi giá trị đại diện cho các loại quần áo khác nhau, như được hiển thị trong mô tả tập dữ liệu.

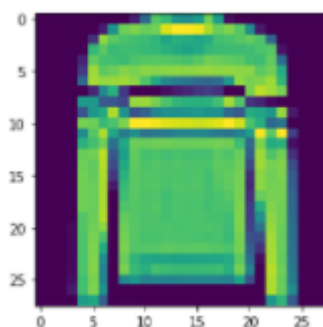
4.1.5 Kiểm tra các giá trị null:

Đây là một quá trình làm sạch dữ liệu thiết yếu đòi hỏi việc kiểm tra các giá trị Null trong tập dữ liệu. Giá trị rỗng có thể là kết quả của lỗi đầu vào và có thể gây ra sự bất thường trong mô hình. May mắn thay, tập dữ liệu MNIST thời trang không có bất kỳ giá trị rỗng nào.

4.2. Trực quan hóa dữ liệu

4.2.1 Lập đồ thị hình ảnh ngẫu nhiên:

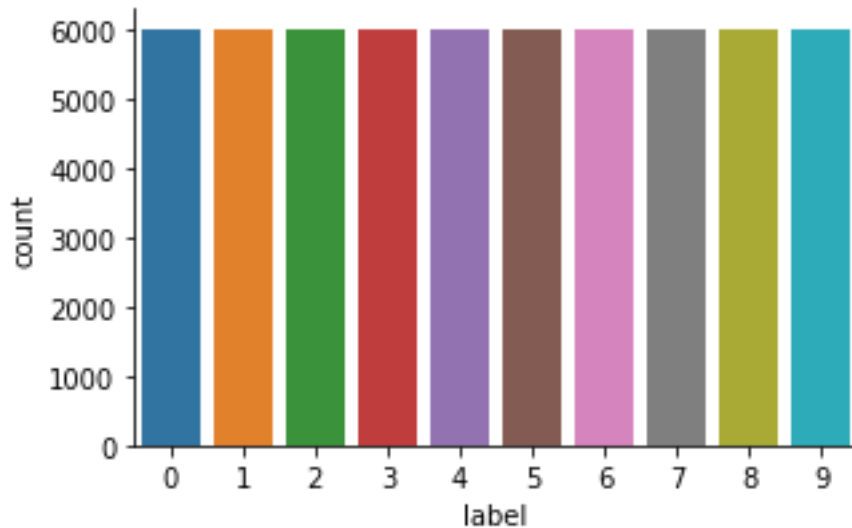
Điều cần thiết là phải nhìn sâu vào những gì pixel đại diện. Điều này có thể đạt được bằng cách xác định các lớp và khớp chúng với các pixel bằng cách sử dụng hàm ngẫu nhiên.



Hình 4 : Hiển thị một trong những hình ảnh từ tập dữ liệu đào tạo

4.2.2 Phân phối mô tả:

Điều này chỉ để đánh giá xem tất cả các nhãn lớp có được phân phối đồng đều hay không để tránh lấy mẫu quá mức. Đầu ra của phân phối cho thấy rằng tất cả các lớp được phân phối đồng đều.



4.3. Chuẩn bị dữ liệu

4.3.1 Tách dữ liệu:

Điều cần thiết là phải có dữ liệu huấn luyện và thử nghiệm riêng biệt để tránh overfitting và underfitting. Scikit-learn cung cấp phương pháp đơn giản nhất để chia nhỏ dữ liệu. Dữ liệu huấn luyện được sử dụng để đào tạo, xác thực (validation) được sử dụng để xác nhận độ chính xác của mô hình.

4.3.2 Định hình lại kích thước hình ảnh:

Thư viện học sâu Keras chấp nhận một đối tượng thêm chiều làm đầu vào channel của nó. Tuy nhiên, các pixel được biểu diễn trong một vector 1 chiều 28 x 28 (784 pixel). Tập dữ liệu được định hình lại thành (-1, 28,28,1), trong đó -1 đại diện cho dimension đầu tiên, chưa được biết và 1 đại diện cho dimension bổ sung cho Keras channel.

4.3.3 Chuẩn hóa:

Chuẩn hóa là điều cần thiết để có một thuật toán tối ưu hóa nhanh hơn. Các giá trị pixel là số nguyên 8 byte cần được chia tỷ lệ thành $[0,1]$ để đạt được mục tiêu tối ưu hóa.

4.4. Huấn luyện mô hình

4.4.1 Xây dựng mô hình:

Đây là phần cốt lõi mà mô hình mạng nơ ron tích chập được xây dựng. Các bước sau đây trình bày quy trình xây dựng mô hình CNN.

- I. Ban đầu, một API Keras tuân tự được định nghĩa để thêm lần lượt các lớp kiến trúc.
- II. Sau đó, thêm lớp CNN đầu tiên (lớp phức hợp), là một khối xây dựng thiết yếu của kiến trúc CNN. Các siêu tham số mục tiêu như số và kích thước bộ lọc (F), stride (S), đệm (P) và chức năng kích hoạt được nhập thủ công vào mô hình. Độ sâu của đầu ra phải bằng với số đầu vào của bộ lọc. Về mặt toán học, điều này được biểu diễn như sau:
- III. $(\text{Chiều cao}, \text{Chiều rộng}) = ((W-F + 2P) / S) + 1$
- IV. Tiếp theo là thêm Pooling layer để giảm kích thước của đầu vào lấy mẫu. Lớp Pooling giảm thiểu các tham số và tính toán, do đó giảm tỷ lệ overfitting cho mô hình.
- V. Sau đó, batch normalization được thêm vào để giảm các giá trị ngoại lệ và buộc mô hình học đồng đều mà không cần dựa vào các trọng số cụ thể.
- VI. Ngoài ra, điều cần thiết là phải có một phần tử bỏ để điều chỉnh lại trọng lượng của mô hình bằng cách thả một số neurons một cách ngẫu nhiên.
- VII. Và cuối cùng, mô hình được làm phẳng và bằng cách thêm lớp làm phẳng(flattening layer) và lớp đầu ra(output) để phân loại hình ảnh bằng chức năng kích hoạt SoftMax.

4.4.2 Biên dịch mô hình:

Để tính toán độ chính xác của mô hình, trước tiên mô hình phải được biên dịch. Biên dịch mô hình bao gồm chỉ định trình tối ưu hóa, chức năng mất mát và phương pháp đánh giá số liệu (the optimizer, loss function và the metric evaluation method). Trình tối ưu hóa được sử dụng trong thử nghiệm nghiên cứu này là Adam. Ngoài ra, thử

nhị phân và categorical_crossentropy để phân loại nhiều lớp. Cuối cùng, mô hình sử dụng độ chính xác để đánh giá các chỉ số của mô hình.

4.4.3 Tóm tắt mô hình:

Đây là giai đoạn kiểm tra xem xét mô hình để xác nhận rằng mọi thứ đã như mong đợi. Hình bên dưới mô tả tóm tắt mô hình.

Model: "sequential"		
<hr/>		
Layer (type)	Output Shape	Param #
=====		
==		
conv2d (Conv2D)	(None, 28, 28, 64)	320
<hr/>		
max_pooling2d (MaxPooling2D)	(None, 14, 14, 64)	0
<hr/>		
dropout (Dropout)	(None, 14, 14, 64)	0
<hr/>		
conv2d_1 (Conv2D)	(None, 14, 14, 32)	8224
<hr/>		
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 32)	0
<hr/>		
dropout_1 (Dropout)	(None, 7, 7, 32)	0
<hr/>		
flatten (Flatten)	(None, 1568)	0
<hr/>		
dense (Dense)	(None, 256)	401664
<hr/>		
dropout_2 (Dropout)	(None, 256)	0
<hr/>		
dense_1 (Dense)	(None, 10)	2570
=====		
==		
Total params: 412,778		

<p>Trainable params: 412,778</p> <p>Non-trainable params: 0</p>

Hình 5 : Bảng tóm tắt mô hình

4.4.4 Đánh giá mô hình:

Vẽ một ma trận confusion tóm tắt kết quả hoạt động của mô hình trên dữ liệu thử nghiệm.

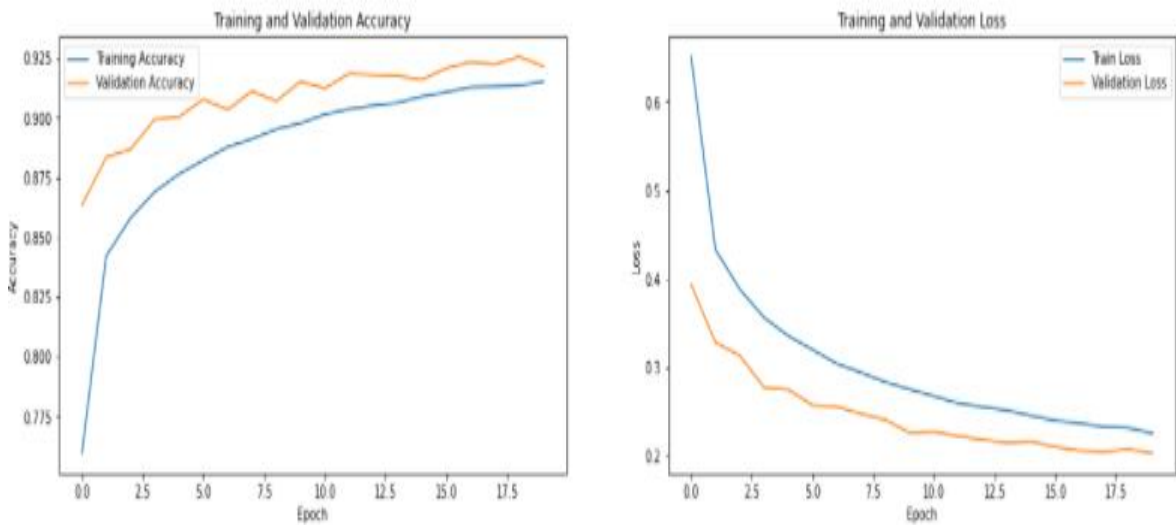
4.4.5 Báo cáo dự đoán và phân loại:

Báo cáo phân loại thể hiện trực quan độ chính xác, recall , điểm F1 và hỗ trợ.

- Độ chính xác đề cập đến tỷ lệ của true positive với tổng của true và false positive. Nó đại diện cho tỷ lệ phần trăm của các giá trị được dự đoán chính xác.
- Việc recall thể hiện khả năng của một mô hình phân loại để xác định tất cả các dự đoán đúng có thể có. Đó là tỷ lệ giữa true positive với tổng các giá trị true positive và false negative.
- Điểm F1 thể hiện giá trị trung bình tổng hợp của độ chính xác và recall. Mô hình có điểm F1 là 1,0 có mức độ chính xác cao nhất trong khi mô hình với điểm không có độ chính xác kém hơn.
- Support đề cập đến sự xuất hiện thực tế của một lớp trong một tập dữ liệu xác định. Imbalance support cho thấy điểm yếu của mô hình phân loại và có thể được sửa chữa bằng cách phân tầng lấy mẫu.

V. KẾT QUẢ

Sự thành công của mô hình được xác định bởi tỷ lệ phần trăm các dự đoán phù hợp do CNN cung cấp trong quá trình nghiên cứu. Dự đoán đúng có nghĩa là mạng phân loại một hình ảnh là lớp mà nó thuộc về một cách hợp lý. Kết quả được đánh giá cho từng loại quần áo thời trang và kết quả trung bình được xác định cho toàn bộ tập hợp con. Độ chính xác bộ training là ~ 92%, trong khi độ chính xác trên bộ validation là ~93%. Dưới đây là phần trình bày về sự phát triển mất mát và mức độ chính xác của mô hình. Giá trị mất mát có nghĩa là một mô hình nhất định hoạt động hiệu quả như thế nào trong mỗi lần lặp lại tối ưu hóa.



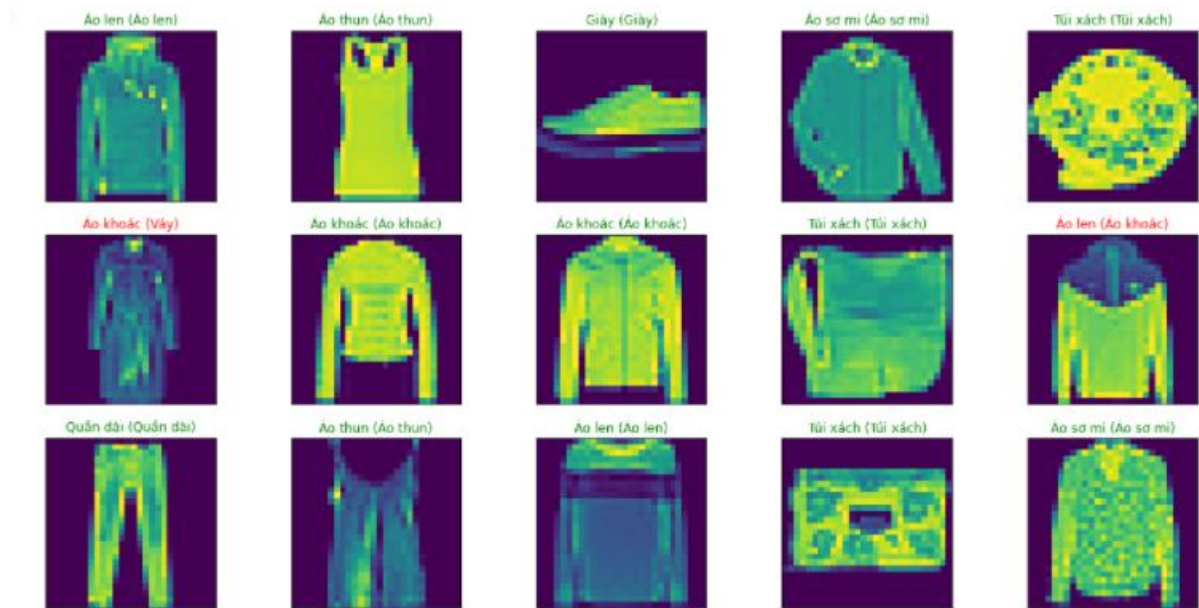
Hình 6 : Biểu đồ training and validation accuracy, loss

Biểu đồ chỉ ra rằng độ chính xác của tập dữ liệu validation (92%) kém hơn một chút so với độ chính xác của tập dữ liệu testing (93%). Sự khác biệt giữa tính nhất quán của việc chuẩn bị và tính nhất quán của thử nghiệm là kết quả của việc overfitting. Việc overfitting xảy ra khi một mô hình học máy hoạt động kém trên các đầu vào mới mà trước đó chưa được biết đến so với kết quả training. Sự mất mát của mô hình CNN là một đồ thị có độ trễ âm. Điều này cho thấy rằng các mô hình hoạt động như mong đợi với mức giảm tổn thất sau mỗi epoch.

Thông thường, độ chính xác của một mô hình được tính toán sau khi các tham số của mô hình được biết và thiết lập, và không có quá trình học nào xảy ra. Sau đó, các mẫu thử nghiệm được đưa vào máy và sau khi so sánh với các mục tiêu thực, số lỗi (zero-one loss) mà máy tạo ra sẽ được registered. Sau đó, nó đo lường phân loại sai.

VI.DEMO

Thử in ra 15 hình ảnh từ tập dữ liệu test set và đặt tiêu đề với dự đoán. Nếu dự đoán đúng (giống nhãn), thì tiêu đề sẽ có màu xanh và ngược lại là màu đỏ.



Hình 7 : Demo

Nhận thấy rằng mô hình đôi khi vẫn dự đoán sai.

VII. HÌNH ẢNH THỰC TIỄN

Để thêm thách thức cho mô hình mạng nơ ron, chúng ta sẽ đưa cho mô hình phân tích và đưa ra dự đoán cho dữ liệu thực tiễn.



Hình 8. Ảnh túi xách

Để cho mô hình dự đoán với ảnh thực tế, cần phải điều chỉnh hình dạng, kiểu dữ liệu ảnh phù hợp với lớp input của mô hình.

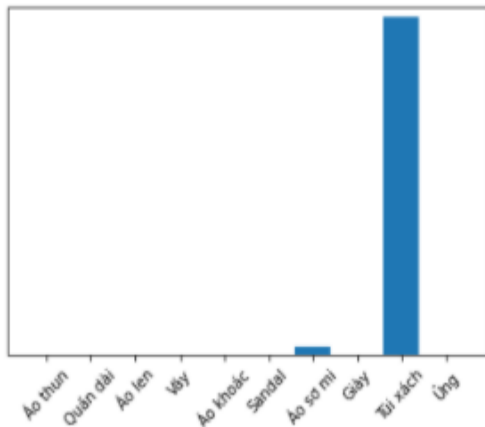
- Kiểu dữ liệu của ảnh là kiểu mảng 'numpy.ndarray'.

- Kích thước của ảnh ban đầu là 28 x 28 pixel.
- Kích thước của ảnh khi đã điều chỉnh hình dạng là (1, 28, 28).

Kết quả nhận và biểu đồ dự đoán của hình ảnh thực tiễn .

```
class_names[np.argmax(result)]
'Túi xách'
```

Hình 9 : Kết quả nhận



Hình 10 : Biểu đồ kết quả dự đoán hình ảnh thực tế

VII. TỔNG QUÁT

Huấn luyện thành công Convolutional Neural Network (CNN) để phân loại Fashion-MNIST với độ chính xác khoảng 91%, cao hơn độ chính xác giả định. Tuy nhiên, vì vấn đề overfitting mà mô hình ghi nhớ các ví dụ huấn luyện và trở nên không hiệu quả đối với tập thử nghiệm, khiến độ chính xác của dự đoán thấp hơn một chút so với độ chính xác huấn luyện. Tái tạo dữ liệu sáng tạo và backpropagation có thể sửa chữa sự mất cân bằng về độ chính xác của dự đoán.

Đối với các dữ liệu đầu vào là các ảnh không thuộc bộ dữ liệu Fashion – MNIST, cần phải thay đổi kiểu dữ liệu ảnh, kích thước , .. và mô hình sẽ chỉ dự đoán chính xác đối với các ảnh chỉ có đơn thuần là một loại trang phục trong danh sách các nhãn. Ngược lại, với các ảnh có background phức tạp, hoặc có nhiều loại trang phục khác nhau, mô hình sẽ đưa ra kết quả không chính xác.

VII. TÀI LIỆU THAM KHẢO

Zalando Research. (2017). Fashion MNIST dataset. Kaggle. Available at <https://www.kaggle.com/zalando-research/fashionmnist>

Zalando Research. (2017). Fashion clothing categories and description. GitHub. Available at <https://github.com/zalando-research/fashion-mnist/blob/master/doc/img/fashion-mnist-sprite.png>

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Yang, S. Luo, P. Loy, C. K. W. Shum, X. (2015). Deep Representation Learning with Target Coding. CNN architecture. Tang in Proceedings of AAAI Conference on Artificial Intelligence, 2015

Aurélien Géron - Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition (September 2019). Dr Daniel Bell – Epochs in Neural Network

<https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-fashion-mnist-clothing-classification>