**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG ĐIỆN TỬ- VIỄN THÔNG**

****

**BÁO CÁO BÀI TẬP LỚN**

**Môn: Trí tuệ nhân tạo và ứng dụng**

**Giảng viên hướng dẫn: TS. Võ Lê Cường**

|  |  |  |
| --- | --- | --- |
| Phạm Thị Anh | 20182360 | ET1 – 04– K63 |
| Nguyễn Quang Thu Phương | 20182725 | ET1 – 03 – K63 |
| Mai Danh Quý | 20182746 | ET1 – 04 – K63 |
| Hồ Sỹ Thạch | 20182769 | ET1 – 04 – K63 |

**Nhóm sinh viên thực hiện:**

Hà Nội, 7 – 2022

# PHÂN CHIA CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| Họ và tên | MSSV | Nhiệm vụ |
| Phạm Thị Anh | 20182360 | Tìm hiểu Vgg16, xây dựng kiến trúc mạng noron để huấn luyện mô hình. Đánh giá kết quả. |
| Nguyễn Quang Thu Phương | 20182725 | Tìm hiểu CNN, xây dựng kiến trúc mạng noron để huấn luyện mô hình. Đánh giá kết quả. |
| Mai Danh Quý | 20182746 | Xử lý dữ liệu, xây dựng kiến trúc mạng noron để huấn luyện mô hình. Đánh giá kết quả. |
| Hồ Sỹ Thạch | 20182769 | Tìm kiếm, xử lý dữ liệu, xây dựng kiến trúc mạng noron để huấn luyện mô hình, huấn luyện mô hình, đánh giá kết quả. Xây dựng giao diện sử dụng mô hình. |

**MỤC LỤC**

[PHÂN CHIA CÔNG VIỆC 2](#_Toc109538837)

[LỜI MỞ ĐẦU 5](#_Toc109538838)

[DANH MỤC HÌNH ẢNH 6](#_Toc109538839)

[Chương 1: TỔNG QUAN ĐỀ TÀI 8](#_Toc109538840)

[1.1 Đặt vấn đề: 8](#_Toc109538841)

[1.2 Mục tiêu: 9](#_Toc109538842)

[Chương 2: CƠ SỞ LÍ THUYẾT 10](#_Toc109538843)

[2.1. Giới thiệu về CNN: 10](#_Toc109538844)

[2.1.1. Mạng neural tích chập: 10](#_Toc109538845)

[2.1.2. Feature: 12](#_Toc109538846)

[2.1.3. Convolutional: 12](#_Toc109538847)

[2.2. Các lớp cơ bản trong CNN: 13](#_Toc109538848)

[2.2.1. Lớp tích chập – Convolution Layer: 13](#_Toc109538849)

[2.2.2. Lớp Pooling [4]: 14](#_Toc109538850)

[2.2.3. ReLU layer: 15](#_Toc109538851)

[2.2.4. Fully connected layer: 15](#_Toc109538852)

[2.3. Cách chọn tham số phù hợp cho CNN [3]: 16](#_Toc109538853)

[**2.4. Mạng VGG 16:** 16](#_Toc109538854)

[2.5. Một số thư viện và thuật toán sử dụng trong mô hình: 17](#_Toc109538855)

[2.5.1. Thư viện Tensorflow: 17](#_Toc109538856)

[2.5.2. Thư viện Keras: 18](#_Toc109538857)

[2.5.3. Thuật toán Gradient Descent: 19](#_Toc109538858)

[Chương 3: THUẬT TOÁN NHẬN DIỆN BIỂN BÁO GIAO THÔNG 21](#_Toc109538859)

[3.1. Sơ đồ tổng quan giải quyết bài toán: 21](#_Toc109538860)

[3.2. Chuẩn bị dữ liệu: 22](#_Toc109538861)

[3.3. Xây dựng kiến trúc mô hình 25](#_Toc109538862)

[3.4. Tiến hành huấn luyện: 26](#_Toc109538863)

[3.4. Kết quả huấn luyện: 27](#_Toc109538864)

[3.5. Giao diện GUI: 30](#_Toc109538865)

[KẾT LUẬN 32](#_Toc109538866)

[TÀI LIỆU THAM KHẢO 33](#_Toc109538867)

[PHỤ LỤC 34](#_Toc109538868)

# LỜI MỞ ĐẦU

Trí tuệ nhân tạo là một trong những bước nhảy lớn về mặt Công nghệ của cả thế giới nói chung và Việt Nam nói riêng. Nó tạo ra những nghiên cứu mới nhằm cải thiện và nâng cao chất lượng cuộc sống của con người như xe tự vận hành, nhận diện khuôn mặt, nhà thông minh…Trí tuệ nhân tạo còn được áp dụng vào mọi lĩnh vực của đời sống con người từ kinh tế đến giao thông vận tải ngay cả những giá trị tinh thần.

Như chúng ta đã biết, tình hình giao thông đường bộ nước ta ngày càng trở nên phức tạp, hàng năm số ca tai nạn giao thông không ngừng tăng. Nguyên nhân xảy ra sự việc đáng tiếc đó một phần do có nhiều biển báo đã mờ, bong tróc các đường sơn khiến cho người tham gia giao thông khó quan sát dẫn đến việc đi sai làn đường, đi vượt quá tốc độ…gây ra tai nạn thương tâm.

Việc ứng dụng công nghệ thông tin để giải quyết các vấn đề trong lĩnh vực giao thông luôn là điều vô cùng cần thiết. Vấn đề phát triển giao thông thông minh để giảm thiểu ùn tắc, tai nạn và tiết kiệm nguồn lực cũng được thảo luận trong nhiều diễn đàn. Nhận dạng biển báo giao thông là một công cụ hỗ trợ trong hệ thống giao thông thông minh. Từ những vấn đề trên nhóm em đã quyết định nghiên cứu đề tài nhận diện biển báo giao thông.

Cảm ơn sự hỗ trợ giảng dạy của TS. Võ Lê Cường, cùng các anh, các bạn trợ giảng đã giúp đỡ nhóm chúng em hoàn thành báo cáo này.

Do kiến thức còn hạn chế, nên trong quá trình tìm hiểu không thể tránh khỏi những sai sót rất mong nhận được sự đóng góp, phản hồi của thầy cô và các bạn để bài báo cáo của nhóm đạt được kết quả tốt hơn.

Xin chân thành cảm ơn!

# DANH MỤC HÌNH ẢNH

[Hình 1. Mô tả hình ảnh mảng ma trận RGB 6x6x3 10](#_Toc109538869)

[Hình 2. Mô hình CNN 11](#_Toc109538870)

[Hình 3. Lớp tích chập 13](#_Toc109538871)

[Hình 4. Ví dụ về các loại của lớp Pooling 14](#_Toc109538872)

[Hình 5. Một số hàm kích hoạt 15](#_Toc109538873)

[Hình 6. Fully connected 16](#_Toc109538874)

[Hình 7. Kiến trúc VGG 16 17](#_Toc109538875)

[Hình 8. Sơ đồ tổng quan giải quyết bài toán 21](#_Toc109538876)

[Hình 9. Các loại biển báo trong điểm chuẩn GTSRB 22](#_Toc109538877)

[Hình 10. Biểu đồ mô tả số lượng các biển báo trong điểm chuẩn GTSRB 23](#_Toc109538878)

[Hình 11. Phân chia dữ liệu 23](#_Toc109538879)

[Hình 12. Bình tròn biểu đồ phân chia dữ liệu 24](#_Toc109538880)

[Hình 13. Code kiến trúc mô hình 25](#_Toc109538881)

[Hình 14. Kiến trúc trực quan của mô hình 25](#_Toc109538882)

[Hình 15. Thông số của mạng 26](#_Toc109538883)

[Hình 16. Huấn luyện 27](#_Toc109538884)

[Hình 17. Kết quả đồ thị Train Accuracy & Validation Acuracy 27](#_Toc109538885)

[Hình 18. Kết qua đồ thị Train loss & validation loss 28](#_Toc109538886)

[Hình 19. Kết quả test data accuracy 28](#_Toc109538887)

[Hình 20. Kết quả thực tế nhân diện biển báo giao thông 29](#_Toc109538888)

[Hình 21. Ma trận kết nối, dự đoán của mô hình trên dữ liệu test 30](#_Toc109538889)

[Hình 22. Giao diện tải tệp lên để phân loại 30](#_Toc109538890)

[Hình 23. Giao diện Realtime 31](#_Toc109538891)

# Chương 1: TỔNG QUAN ĐỀ TÀI

*Chương này sẽ giới thiệu cho bạn đọc về lí do và mục đích nhóm chọn đề tài này.*

## Đặt vấn đề:

Giao thông Việt Nam luôn là một trong những vấn đề nhức nhối của xã hội, số lượng các vụ tai nạn giao thông ngày càng trở nên phức tạp và không ngừng gia tăng. Theo báo cáo của Ủy ban an toàn giao thông (ATGT) Quốc gia tại Hội nghị trực tuyến toàn quốc sơ kết công tác bảo đảm trật tự an toàn giao thông Quý I, phương hướng, nhiệm vụ công tác Quý II/2022. Theo đó, tai nạn giao thông trong Quý I/2022 trên cả nước tính từ ngày 15/12/2021 đến 14/3/2022, xảy ra 2.762 vụ tai nạn giao thông, làm chết 1.676 người, bị thương 1.741 người. Trong tổng số vụ tai nạn giao thông có 15,47% số vụ do người điều khiển phương tiện vi phạm làn đường, phần đường; 7,49% do chuyển hướng không đảm bảo an toàn; 3,69% do vượt xe sai quy định; 3,28% do vi phạm tốc độ; 1,76% do sử dụng rượu, bia, chất kích thích có cồn; 1,52% do người đi bộ sang đường sai quy định; 0,12% do công trình giao thông đường bộ không đảm bảo an toàn; 0,52% do dừng đỗ sai quy định; 0,08% do phương tiện không đảm bảo an toàn kỹ thuật; 27,42% do các nguyên nhân khác; 38,61% chưa xác định được nguyên nhân [1]. Nhận thấy, nguyên nhân phần lớn xảy ra các vụ tai nạn là do ý thức của người tham gia giao thông như uống rượu bia mất kiểm soát, hay đi sai làn đường quy định…Việc biển báo bị mờ, bong tróc gây khó nhìn cho người đi đường cũng là nguyên nhân gây tai nạn giao thông khiến cho người tham gia giao thông đi sai luật. Trên cơ sở đó nhóm đã chọn xây dựng và phát triển đề tài nhận diện biển báo giao thông để phù hợp với tình hình giao thông hiện tại.

## Mục tiêu:

* Xây dựng phương pháp phát hiện và nhận dạng biển báo giao thông.
* Hỗ trợ người tham gia giao thông ý thức và chủ động trong việc xử lí tình huống nguy hiểm tiềm ẩn khi tham gia giao thông.
* Tích hợp vào hệ thống xe tự hành.
* Hỗ trợ người khiếm thị khi tham gia giao thông.
* Xây dựng hệ thống tối ưu, có độ chính xác cao > 90%.
* Thời gian nhận diện ngắn, độ sử dụng lâu dài.
* Giảm được độ phức tạp thuật toán, thời gian học máy, nhận diện kết quả.

# Chương 2: CƠ SỞ LÍ THUYẾT

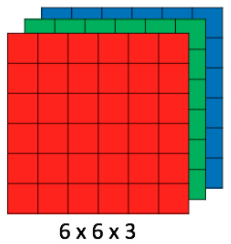
*Trong chương này, nhóm sẽ trình bày về cơ sở lí thuyết về mạng tích chập CNN từ đó thiết kế hệ thống nhận diện biển báo giao thông.*

## 2.1. Giới thiệu về CNN:

### 2.1.1. Mạng neural tích chập:

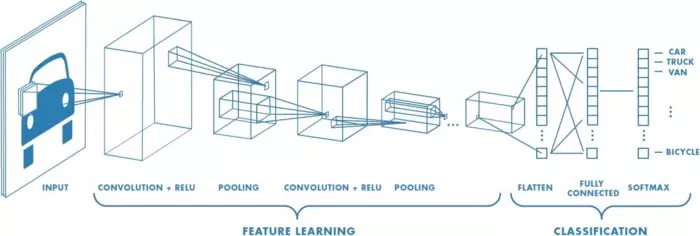
Trong mạng neural, mô hình mạng neural tích chập (Convolutional Neural Network được viết tắt là CNN) là một trong những mô hình để nhận dạng và phân loại hình ảnh [2] .

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). [2]



Hình . Mô tả hình ảnh mảng ma trận RGB 6x6x3

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. [2]



Hình . Mô hình CNN

Mạng CNN gồm nhiều lớp Convolution chồng lên nhau sử dụng các hàm để kích hoạt trọng số. Mỗi một lớp sau khi được kích hoạt sẽ cho ra một kết quả trừu tượng cho các lớp tiếp theo. Mỗi layer kế tiếp chính là thể hiện kết quả của layer tiếp theo. Trong CNN model có 2 cạnh cần quan tâm là tính **bất biến** (Bất biến vị trí) và kết **hợp tính toán** (Điều kiện). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các mức độ khác nhau (dịch, xoay, mở rộng quy mô), thì thuật toán chính thức sẽ bị ảnh hưởng đáng kể. Cấu trúc cơ bản của CNN gồm 3 phần chính: local receptive field, shared weights và bias, và pooling layer. Local receptive field (trường cục bộ) giúp chúng ta tách lọc các dữ liệu, thông tin của ảnh và chọn những vùng có giá trị tốt nhất. Shared weights và bias (Trọng số chia sẻ) làm giảm tối đa số lượng các tham số. Còn Pooling layer (lớp tổng hợp) sẽ cho bạn tính toán bất kỳ biến đối với phép chuyển đổi (dịch), phép quay (quay) và phép co giãn (mở rộng). Tính toán cục bộ kết hợp cho ta biểu diễn thông tin từ mức thấp đến mức cao và vật thể hơn tích chập thông qua các bộ lọc. Đó là lý do tại sao CNN cho ra model với độ chính xác rất cao như cách con người nhận biết các vật thể trong tự nhiên.

### 2.1.2. Feature:

Như được biết CNN so sánh hình ảnh theo từng mảnh, mỗi mảnh đó được gọi là Feature. Mỗi Feature được coi như một hình ảnh mini, những mảnh hai chiều nhỏ. Các Feature được khớp với các khía cạnh chung của bức ảnh nghĩa là feature này sẽ tương ứng với khía cạnh nào đó của hình ảnh và chúng được khớp lại với nhau. So với việc khớp các bức ảnh với nhau thì CNN làm việc nhìn ra sự tương đồng trong tìm kiếm thô các Feature khớp với nhau trong hai hình ảnh tốt hơn. [3]

### 2.1.3. Convolutional:

Convolutional hay còn được gọi là tích chập. Khi xem một hình ảnh mới, CNN không biết nó ở vị trí nào, các Feature sẽ khớp với nhau ở đâu vì vậy chúng tiến hành thử ở tất cả các vị trí khác nhau. Trong quá trình đó chúng tạo ra được bộ lọc (Filter). Và để thực hiện được điều này, chúng ta đã sử dụng phần toán noron tích chập [3].

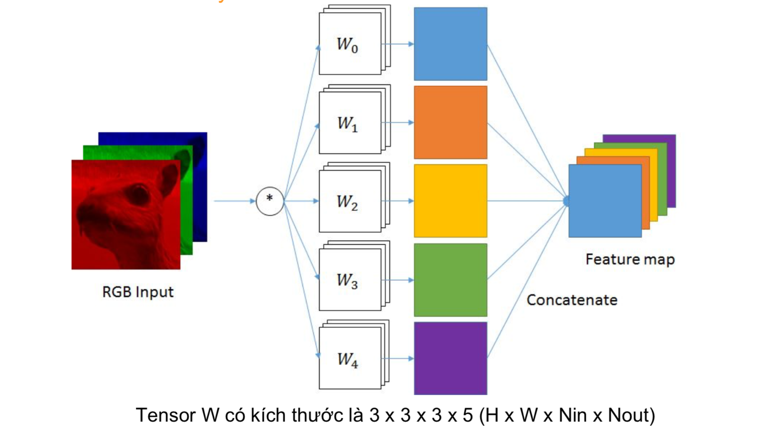
Để tính toán sự khớp của cá Feature với mỗi mảnh của hình ảnh thì ta lấy kết quả của các phép tính giữa mỗi điểm ảnh trong Feature nhân với giá trị của điểm ảnh tương ứng trong hình ảnh đó, sau đó cộng hết với nhau rồi chia cho số lượng các điểm ảnh có trong Feature đó. Nếu kết quả khớp thì kết quả bằng 1 còn không thì kết quả là (-1).

Để hoàn tất quá trình tích chập ta phải lặp lại hành động sắp xếp tất cả các Feature vào tất cả mọi mảnh hình ảnh có thể thực hiện được. Kết quả của quá trình này chính là chúng ta có những hình ảnh được lọc, mỗi cái sẽ có filter tương ứng, Quá trình tích chập diễn ra theo từng lớp mội và được gọi là layer [3].

## 2.2. Các lớp cơ bản trong CNN:

### 2.2.1. Lớp tích chập – Convolution Layer:

Đây là lớp quan trọng nhất của CNN, lớp này có nhiệm vụ thực hiện mọi tính toán.



Hình . Lớp tích chập

Những yếu tố quan trọng của một Convolutional layer là:

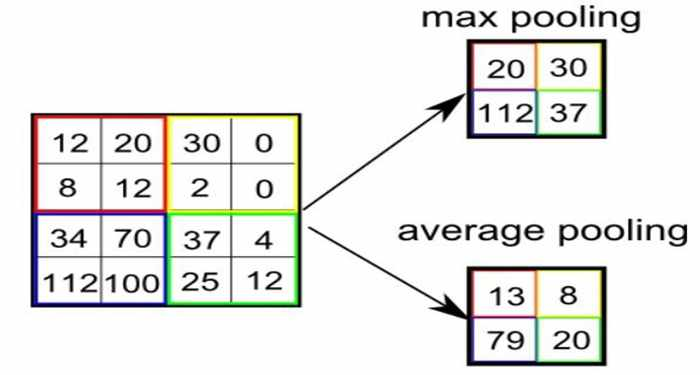
* + - * CNN sử dụng những filter để áp dụng vào những vùng của hình ảnh. Các filter map này chính là ma trận 3 chiều trong đó bao gồm những con số được gọi là parameter.
* Bước nhảy- Stride: là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy [2].
* Đường viền – Padding: là những giá trị 0 được thêm vào lớp input hoặc cắt bớt hình ảnh tại những điểm không phù hợp với kernel do kernel đôi khi không phù hợp với hình ảnh đầu vào [3].
* Feature map: thể hiện kết quả mỗi lần filter map quét qua input. Mỗi lần quét như thế sẽ xảy ra quá trình tính toán.

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 vuông nhỏ của dữ liệu đầu vào. Nó là một phép toán có hai đầu vào như ma trận hình ảnh và một bộ lọc hoặc hạt nhân.

### 2.2.2. Lớp Pooling [4]:

Khi đầu vào quá lớn, những lớp pooling sẽ được xếp vào giữa những lớp convolutional để làm giảm parameter.

Pooling layer được biết đến với 2 loại phổ biến là: max pooling và average pooling.



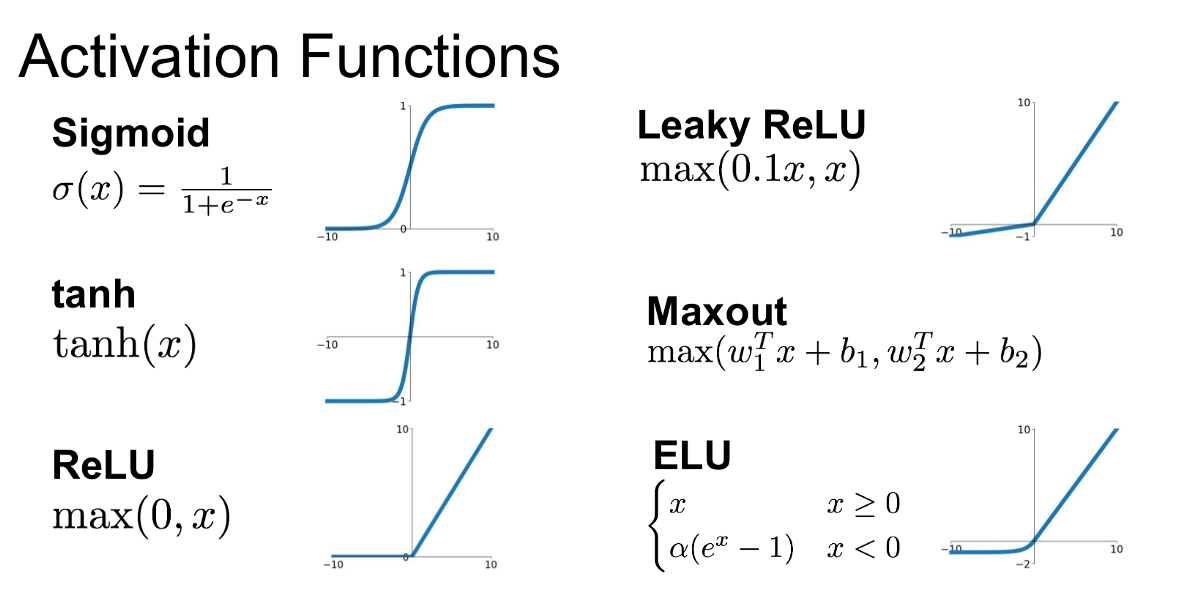
Hình . Ví dụ về các loại của lớp Pooling

Tại Pooling layer, khi sử dụng lớp Max pooling thì số lượng parameter có thể sẽ giảm đi. Vì vậy, Convolutional Neural Network sẽ xuất hiện nhiều lớp filter map và mỗi filter map đó sẽ cho ra một Max Pooling khác nhau.

Trong một số model người ta dùng convolutional layer với stride > 1 để giảm kích thước dữ liệu thay cho pooling layer.

### 2.2.3. ReLU layer:

ReLU (Rectified Linear Unit) là 1 hàm phi tuyến với đầu ra là: ƒ (x) = max (0, x), là một hàm kích hoạt trong neural network. Hàm này giúp mô phỏng các neuron có tỉ lệ truyền xung qua axon. Một số hàm cơ bản như: Signmoid, Tanh, Maxout…ReLU được sử dụng khá phổ biến hiện nay do có hiệu suất tốt, tính toán nhanh [2] [3].



Hình . Một số hàm kích hoạt

### 2.2.4. Fully connected layer:

Lớp này có nhiệm vụ đưa ra kết quả sau khi lớp Convolutional và pooling đã nhận được ảnh truyền. Lúc này, ta thu được kết quả là model đã đọc được thông tin của ảnh và để liên kết chúng cũng như cho ra nhiều output hơn thì ta sử dụng Fully Connected layer.



Hình . Fully connected

## 2.3. Cách chọn tham số phù hợp cho CNN [3]:

Để chọn được các tham số phù hợp cho CNN thì chúng ta cần chú ý đến số lượng các mục sau: số convolution layer, filter size, poolong size và việc train test.

* Số convolution layer: lớp này càng nhiều càng tốt khi đó chương trình càng được cải thiện. Việc sử dụng các layer với số lượng sẽ dẫn đến các tác động có thể được giảm một cách đáng kể.
* Filter size: thông thường các filter size có kích thước là 3x3 hoặc 5x5.
* Pooling size: nếu hình ảnh thông thường sẽ sử dụng kích thước 2x2 còn nếu đầu vào hình ảnh lớn ta có thể sử dụng 4x4.
* Train test: nên được thực hiên nhiều lần sẽ cho ra các parameter tốt nhất.

**2.4. Mạng VGG 16:**

* VGG 16 là mạng convolutional neural network được đề xuất bởi K. Simonyan and A. Zisserman, University of Oxford.
* Model sau khi train bởi mạng VGG 16 đạt độ chính xác 92.7% top-5 test trong dữ liệu ImageNet gồm 14 triệu hình ảnh thuộc 1000 lớp khác nhau.

Diagram

Description automatically generated with low confidence

Hình . Kiến trúc VGG 16

## 2.5. Một số thư viện và thuật toán sử dụng trong mô hình:

### 2.5.1. Thư viện Tensorflow:

* **Giới thiệu về Tensorglow**

Khi yêu cầu về dữ liệu của con người ngày càng lớn, các thuật toán machine learning không thể đáp ứng, chỉ có deep learning mới mang đến hiệu năng đủ dùng khi xử lý khối lượng thông tin khổng lồ nên Google quyết định cải tiến các dịch vụ của mình bằng cách sử dụng deep learning neural network.

Vì vậy tensorflow xuất hiện với mục đích giúp các thành viên trong một team phát triển phần mềm có thể làm việc cùng nhau trên model A.I. Nó có một hệ sinh thái toàn diện, linh hoạt gồm các công cụ, thư viện và tài nguyên cộng đồng cho phép các nhà nghiên cứu thúc đẩy tính năng tiên tiến trong ML và các nhà phát triển dễ dàng xây dựng và triền khai các ứng dụng hỗ trợ ML.

* **Kiến trúc của Tensorflow**

Kiến trúc hoạt động của Tensorflow được chia thành 3 phần:

* Tiền xử lý dữ liệu
* Dựng model
* Train và ước tính model
* **Ưu điểm của Tensorflow**

Tensorflow được dùng như một công cụ đắc lực mang đến cái nhìn tổng quan cho người dùng. Nếu người dùng cảm thấy chưa hài lòng về cách thức thực hiện cũng như là kết quả mà Tensorflow mang lại, họ có thể trực tiếp chỉnh sửa các operation trong biểu đồ hướng xử lý dữ liệu. Thay vì phải khởi tạo lại toàn bộ một biểu đồ, cách chỉnh sửa theo chi tiết thế này tiết kiệm được rất nhiều thời gian và công sức. Người dùng còn có thể đánh giá, nhận xét và cài đặt thêm các tính năng khác cho Tensorflow để có thể mang lại kết quả tốt hơn cho sau này.

Tensorflow có tính năng Tensorboard cho phép người dùng quan sát một cách trực tiếp quy trình làm việc của Tensorflow. Để dễ dàng phát hiện ra những sai lầm của quy trình vận hành hệ thống để tiến đến bước chỉnh sửa.

### 2.5.2. Thư viện Keras:

* **Giới thiệu về Keras**

Keras là một open source cho Neural Network, được viết bởi ngôn ngữ Python. Nó là một thư viện được phát triển bởi Francois Chollet, là một kỹ sư nghiên cứu Deep learning.

Keras chạy trên nền tảng máy học Tensorflow và nó được phát triển với trọng tâm là cho phép thử nghiệm nhanh. Có thể đi từ ý tưởng đến kết quả càng nhanh càng tốt và là chìa khóa quan trọng để thực hiện nghiên cứu tốt.

* **Ưu điểm của Keras**

Đơn giản – nhưng không đơn giản. Keras giảm tải nhận thức của nhà phát triển để giải phóng, tập trung vào các vấn đề thực sự quan trọng.

Linh hoạt – Keras áp dụng nguyên tắc bộc lộ dần sự phức tạp: quy trình công việc đơn giản phải nhanh chóng và dễ dàng, trong khi quy trình công việc nâng cao tùy ý có thể thực hiện được thông qua một con đường rõ rang dựa trên những gì đã học.

Mạnh mẽ - Keras cung cấp khả năng mở rộng và hiệu suất toàn ngành: nó được các tổ chức và công ty bao gồm NASA, Youtube, Waymo sử dụng.

### 2.5.3. Thuật toán Gradient Descent:

Gradient Descent là mộ kỹ thuật tối ưu hóa phổ biển trong Machine Learning và Deep Learning. Gradient là độ dốc của một hàm. Nó đo lường mức độ thay đổi của một biến để đáp ứng với những thay đổi của một biến khác. Về mặt toán học, Gradient Descent là một hàm lồi có đầu ra là đạo hàm riêng của một tập các tham số đầu vào của nó. Gradient càng tăng thì độ dốc càng lớn. Thuật toán sẽ lặp lại nhiều lần để tìm ra được kết quả tối ưu.

Thuật toán Gradient Descent có một tham số là learning rate. Learning rate (tốc độ học) là một siêu tham số sử dụng trong việc huấn luyện các mạng nơ ron, có giá trị dương nhỏ, thường nằm trong khoảng giữa 0 và 1. Tốc độ học kiểm soát tốc độ mô hình thay đổi các trọng số để phù hợp với bài toán. Tốc độ học lớn giúp mạng nơ ron được huấn luyện nhanh hơn nhưng cũng có thể làm giảm độ chính xác, không hội tụ hay không tìm được cực tiểu. Nếu tốc độ học quá thấp, mô hình sẽ huấn luyện rất chậm vì các bản cập nhật tối thiểu được thực hiện đối với trọng số qua mỗi lần lặp. Do đó, sẽ mất nhiều bản cập nhật trước khi đạt đến điểm tối thiểu.

Khi dữ liệu lớn, thuật toán Gradient Descent có sử dụng thêm các thuật ngữ epochs, batch size, iterations để giải quyết các vấn đề của bài toán. Trong đó, Một Epoch được tính là khi chúng ta đưa tất cả dữ liệu trong tập train vào mạng neural network 1 lần.  
Batch size là số lượng mẫu dữ liệu trong một batch. Ở đây, khái niệm batchsize và số lượng batch là hoàn toàn khác nhau. Chúng ta không thể đưa hết toàn bộ dữ liệu vào huấn luyện trong 1 epoch, vì vậy chúng ta cần phải chia tập dữ liệu thành các phần, mỗi phần có kích thước là batch size. Còn Iterations là số lượng batchs cần để hoàn thành 1 epoch.

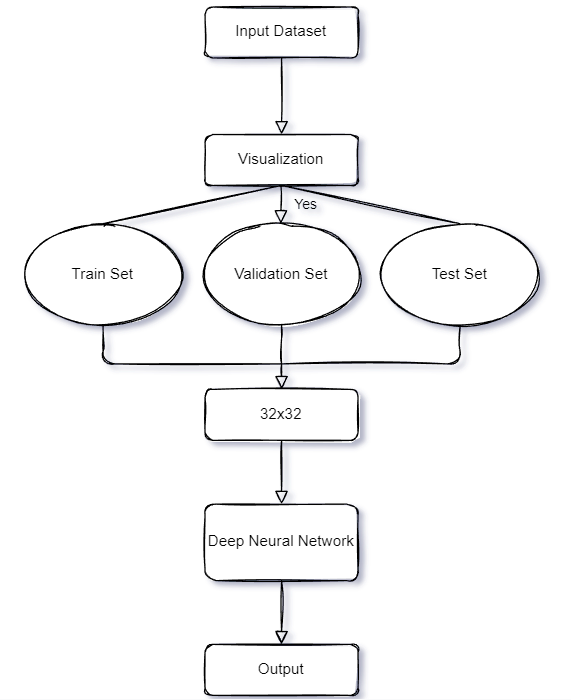
Thông thường có 3 loại Gradient Descent:

* Batch Gradient Descent
* Stochastic Gradient Descent
* Độ dốc màu theo lô nhỏ.

# Chương 3: THUẬT TOÁN NHẬN DIỆN BIỂN BÁO GIAO THÔNG

*Dựa trên cơ sở lý thuyết đã tìm hiểu kết hợp với ngôn ngữ lập trình python và Google colab nhóm đã tiến hành xây dựng hệ thống dưới đây*

## 3.1. Sơ đồ tổng quan giải quyết bài toán:



Hình . Sơ đồ tổng quan giải quyết bài toán

## 3.2. Chuẩn bị dữ liệu:

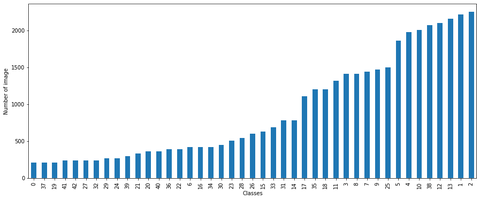
Dựa vào điểm chuẩn bộ dữ liệu nhận dạng biển báo giao thông của Đức là GTSRB (được viết tắt là German Traffic Sign Recognition Benchmark) để nhóm thực hiện xây dựng hệ thống. Bộ dữ liệu GTSRB gồm:

* 51.840 hình ảnh về biển báo hiệu đường bộ của Đức bao gồm 43 loại biển báo
* Hình ảnh có sẵn dưới dạng hình ảnh màu (RGB)
* Kích thước của traffic signs trong tập này dao động từ 15×15 đến 222×193 pixel.



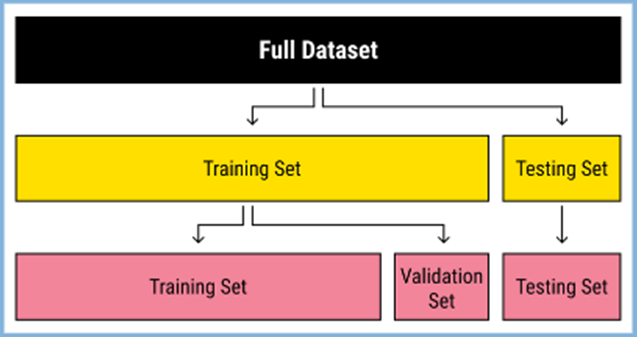
Hình . Các loại biển báo trong điểm chuẩn GTSRB

Số lượng các hình ảnh trong mỗi loại biển báo được phân bố như biểu đồ dưới đây.



Hình . Biểu đồ mô tả số lượng các biển báo trong điểm chuẩn GTSRB

Mô hình phân chia dữ liệu:



Hình . Phân chia dữ liệu

Trong bộ dữ liệu GTSRB, nhóm thực hiện lấy 31367 ảnh để train chiếm 60,5%, 7842 ảnh dùng để validation chiếm 15,1% của bộ dữ liệu, còn lại để sử dụng test 12630 ảnh.

Chart, pie chart

Description automatically generated

Hình . Bình tròn biểu đồ phân chia dữ liệu

Bài toán đặt ra là xây dựng mô hình nhận diện, phân loại biển báo giao thông. Đầu vào của mô hình là các ảnh có kích thước 32 x 32 x 3 và đầu ra của mô hình là loại biển báo của ảnh đó.

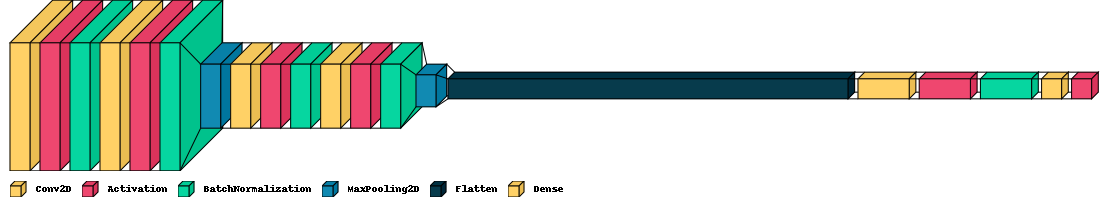
Chúng ta sẽ đọc, lưu ảnh dưới dạng các ma trận số nguyên. Do ảnh từ bộ dữ liệu GTSRB có kích thước khác nhau nên ta cần resize hết tất cả các ảnh về kích thước 32 x 32 x 3. Dễ nhận thấy tập dữ liệu là một mảng 3 chiều có thước 31367 x 32 x 32 với các giá trị số nguyên (0-255). Các giá trị số nguyên này cần được chuẩn hóa về trong khoảng (0,1). Nhãn là một số nguyên. Với 43 ngõ ra, nhãn cần được mã hóa sử dụng mã one-hot. Mã one-hot sử dụng 0 và 1 để mã hóa các số nguyên. Giá trị 1 cho tích cực và 0 cho không tích cực. Ví dụ, để biểu diễn nhãn số 5. Thay vì dùng giá trị số nguyên là 5, mã one-hot cho số nguyên 5 là “0,0,0,0,0,1,0…0”. Để có dữ liệu đào tạo đa dạng hơn, ta tăng cường ảnh bằng cách dịch, quay, làm mờ…

## 3.3. Xây dựng kiến trúc mô hình

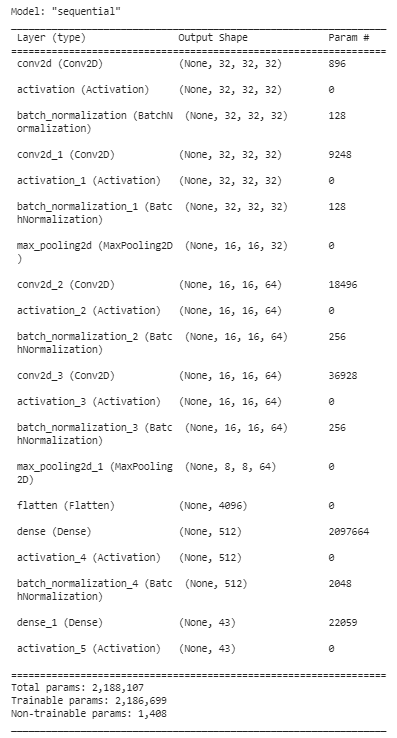
Dựa theo kiến trúc VGG16, nhóm đã xây dựng kiến trúc sau:



Hình . Code kiến trúc mô hình



Hình . Kiến trúc trực quan của mô hình



Hình . Thông số của mạng

Thông số của mô hình, chúng ta quan tâm đầu vào của mô hình là 32 x 32 và đầu ra là 43 tương ứng 43 loại biển báo.

## 3.4. Tiến hành huấn luyện:

Các bước trên đã thực hiện xong , bây giờ chúng ta sẽ tiến hành huấn luyện mô hình:

* Sử dụng framework Tensorflow 2.x
* Learning rate: 0.0001
* Optimizer: Adam
* Epochs: 100
* batch size: 32

Text

Description automatically generated

Hình . Huấn luyện

## 3.4. Kết quả huấn luyện:

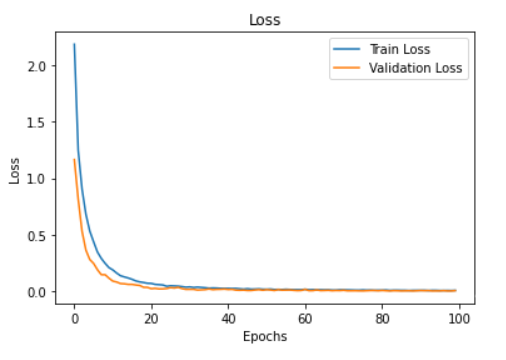
* Sau 20 phút đào tạo, Google Colab sử dụng GPU T4
* Kết quả huấn luyện được mô tả bằng các đồ thị dưới đây :

A picture containing chart

Description automatically generated

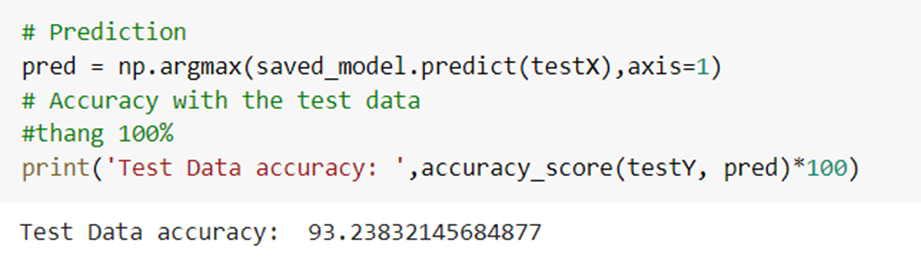
Hình . Kết quả đồ thị Train Accuracy & Validation Acuracy

* Valid Accuracy: 0.9981
* Train Accuracy: 0.9964



Hình . Kết qua đồ thị Train loss & validation loss

* Mô hình cho kết quả nhận diện biển báo trên tập test với độ chính xác là 93.24%.



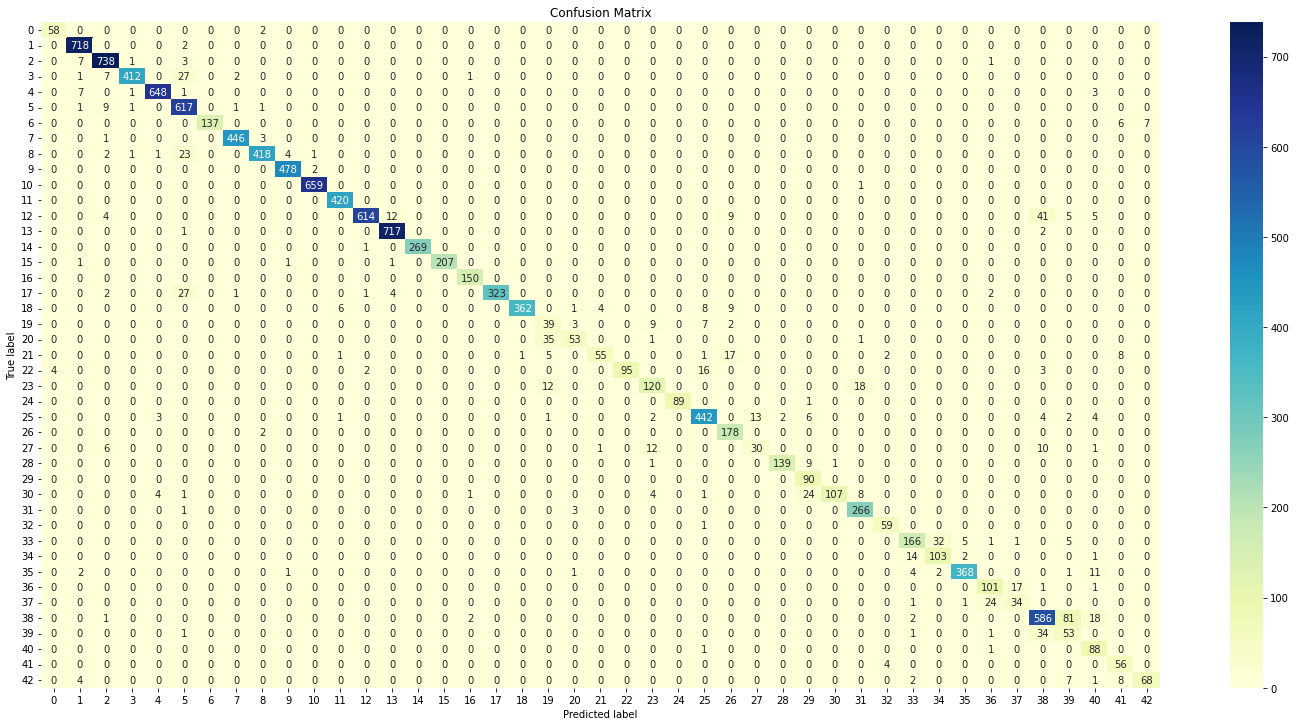
Hình . Kết quả test data accuracy

Dưới đây là kết quả thực tế khi nhận diện trên bộ test, những hình ảnh có nhãn màu xanh là đúng, màu đỏ là mô hình nhận diện sai:



Hình . Kết quả thực tế nhân diện biển báo giao thông

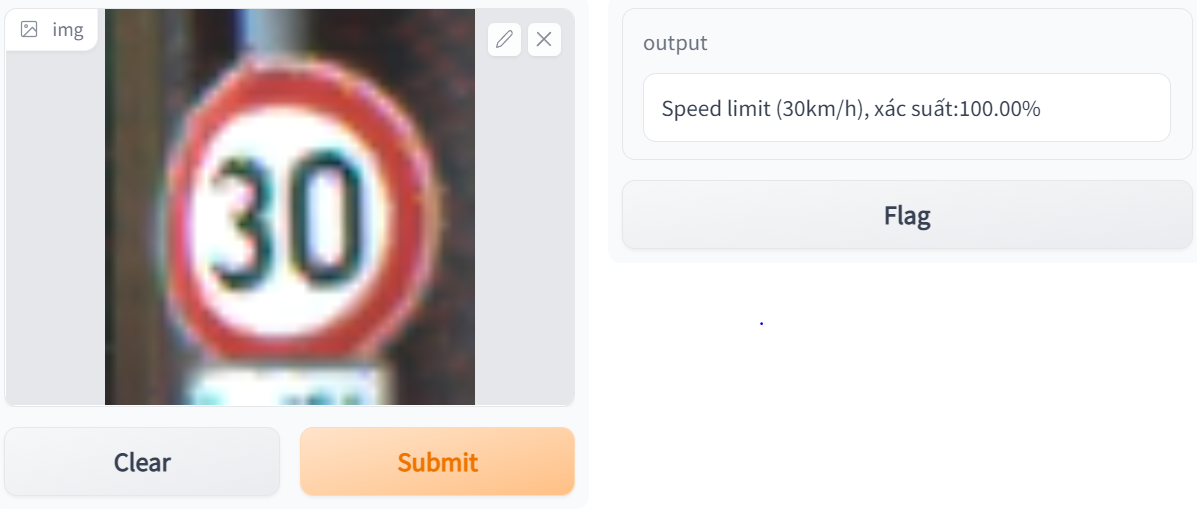
Từ ma trận dự đoán dưới đây chúng ta có thể thấy rõ mô hình nhận diện sai nhiều ở biển báo nào. Từ ma trận này chúng ta có thể thấy với số lượng ảnh trong mỗi loại biển báo là nhiều thì độ chính xác càng cao.



Hình . Ma trận kết nối, dự đoán của mô hình trên dữ liệu test

## 3.5. Giao diện GUI:

Xây dựng một giao diện đơn giản để sử dụng mô hình nhận diện biển báo giao thông.



Hình . Giao diện tải tệp lên để phân loại

Giao diện người dùng tải file ảnh lên, mô hình sẽ phân loại loại biển báo.

Mô hình trung bình phân loại 1 ảnh biển báo trong 0.5s, Nhóm cũng xây dựng ứng dụng test realtime chạy bằng python. Tuy nhiên độ chính xác của nó không được ổn đinh.



Hình . Giao diện Realtime

# KẾT LUẬN

Hệ thống nhận diện biển báo giao thông là một trong những đề tài gần gũi và cần thiết trong xã hội hiện nay khi mà vấn đề giao thông của nước ta luôn là vấn đề nóng, cần được giải quyết cải thiện một cách nhanh chóng. Qua quá trình tìm hiểu và nghiên cứu, nhóm đã hoàn thành tốt hệ thống nhận diện biển báo giao thông. Áp dụng và phát triển dựa trên các mô hình đã có từ lâu, mà hệ thống đã đạt được độ chính xác cao hơn và có tính ứng dụng tốt hơn các dự án nghiên cứu trước đó. Việc thực hiện đề tài không chỉ giúp các thành viên có thêm những kiến thức mới mà còn giúp mỗi cá nhân rèn luyện khả năng làm việc nhóm, thuyết trình, báo cáo và nghiên cứu.

Cảm ơn sự hỗ trợ và giúp đỡ của thầy và các anh chị trợ giảng, cùng cảm ơn các dự án nghiên cứu đã có để nhóm có thể tham khảo và phát triển dự án tốt hơn. Nhóm em rất mong nhận được sự phản hồi, đóng góp của bạn đọc để hệ thống có thể hoàn thiện hơn.

Nhóm em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | N. T. Xuân, "Thông tin nhanh về tình hình tai nạn giao thông Quý I/2022.," [Online]. Available: https://binhphuoc.gov.vn/vi/stp/an-toan-giao-thong/thong-tin-nhanh-ve-tinh-hinh-tai-nan-giao-thong-quy-i-2022-1290.html. |
| [2] | C. P. Van, "[Deep Learning] Tìm hiểu về mạng noron tích chập CNN," [Online]. Available: https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2. |
| [3] | P. A. Nguyễn, "[GÓC CHIA SẺ] Tìm hiểu về Convolution neural network là gì?," [Online]. Available: https://timviec365.vn/blog/convolutional-neural-network-new9565.html. |
| [4] | N. T. Tuấn, "Deep Learning cơ bản," in *Phần V: Convolutional Neural Network*. |

# PHỤ LỤC

1. Phụ lục code:

* Link code của hệ thống: <https://colab.research.google.com/drive/1hYnHwfAQr8_CmgTlrE19Cf1BYFATKwK0?usp=sharing>