**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

Logo, company name

Description automatically generated

**BÁO CÁO MÔN HỌC**

**NHẬN DẠNG**

**CS338.N21.KHCL**

**ĐỀ TÀI: SKETCH SEARCH**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **ĐỖ VĂN TIẾN** |
| **SINH VIÊN THỰC HIỆN:** | **NGUYỄN VŨ DƯƠNG – 20520465**  **LÊ TRẦN QUỐC KHÁNH – 20520574**  **HUỲNH ĐÌNH KIM ĐIỀN - 20521183** |

**TP. HỒ CHÍ MINH, 06/2023**

LỜI CẢM ƠN

Tìm kiếm là một quá trình quan trọng giúp mở rộng kiến thức, tìm ra giải pháp cho các vấn đề, và phát triển bản thân thông qua việc thu thập thông tin, khám phá kiến thức mới, đánh giá tính đúng đắn và truyền đạt thông tin cho người khác. Tuy nhiên không phải chúng ta có thể biểu diễn được hình ảnh trong đầu mình để tìm kiếm.

Là sinh viên ngành KHMT, trong đồ án môn Nhận dạng, nhóm chúng em đã chọn và thực hiện đồ án “Sketch Search” dùng để tìm kiếm các thông tin của đối tượng tìm kiếm bằng những bản phác khảo giúp cho người dùng có thể tìm được thứ mình nghĩ. Bên cạnh đó, ứng dụng này cũng là thứ giúp cho những đứa trẻ có thể học hỏi nhiều thứ xung quanh mình thông qua những bức hình của mình vẽ để từ đó phát triển tư duy.

Nhóm xin gửi lời cảm ơn chân thành đến thầy Đỗ Văn Tiến đã tận tình giảng dạy, hướng dẫn chúng em trong suốt thời gian học vừa qua và các bạn học đã góp ý và giúp đỡ nhóm trong quá trình thực hiện đồ án này.

Do kiến thức và thời gian thực hiện hạn chế, đồ án của nhóm vẫn còn nhiều thiếu sót. Nhóm rất mong nhận được góp ý của thầy và các bạn để đồ án của nhóm được hoàn thiện.

TP. Hồ Chí Minh, tháng 6 năm 2023.

MỤC LỤC

[Chương 1: TỔNG QUAN 1](#_Toc137989279)

* 1. [Giới thiệu đề tài 1](#_Toc137989280)
  2. [Lý do chọn đề tài 1](#_Toc137989281)
  3. [Đối tượng và phạm vi đề tài 2](#_Toc137989282)
  4. [Thách thức đề tài 2](#_Toc137989283)
  5. [Hướng giải quyết tổng quát 3](#_Toc137989284)
  6. [Quá trình thực hiện đề tài 3](#_Toc137989285)

[Chương 2: TẬP DỮ LIỆU 4](#_Toc137989286)

[2.1) Giới thiệu bộ dữ liệu 4](#_Toc137989287)

[2.2) Lý do lựa chọn 5](#_Toc137989288)

[2.3) Phân tích bộ dữ liệu 6](#_Toc137989289)

[2.4) Xử lý bộ dữ liệu 8](#_Toc137989290)

[2.4.1) Thu thập dữ liệu 8](#_Toc137989291)

[2.4.2) Giải quyết nhược điểm 8](#_Toc137989292)

[Chương 3: CƠ SỞ LÝ THUYẾT 10](#_Toc137989293)

[3.1) Học đặc trưng 10](#_Toc137989294)

[3.1.1) Giới thiệu 10](#_Toc137989295)

[3.2) Phân loại hình ảnh 13](#_Toc137989296)

[3.2.1) Giới thiệu 13](#_Toc137989297)

[3.2.2) Hướng tiếp cận 13](#_Toc137989298)

[3.3) Tổng kết 15](#_Toc137989299)

[Chương 4: PHƯƠNG PHÁP 16](#_Toc137989300)

[4.1) Lựa chọn hướng học 16](#_Toc137989301)

[4.1.1) Mô hình CNN (Convolutional Neural Network) 16](#_Toc137989302)

[4.1.2) Mô hình FPN (Feature Pyramid Network) 17](#_Toc137989303)

[4.1.3) Mô hình Vision Transformer(VIT) 18](#_Toc137989304)

[4.1.4) So sánh các mạng 19](#_Toc137989305)

[4.2) Downsampling CNN 20](#_Toc137989306)

[4.2.1) Tìm hiểu về các phương pháp Downsampling 20](#_Toc137989307)

[4.2.2) Pooling và Inverted Residual Block 22](#_Toc137989308)

[4.2.3) Kết luận 25](#_Toc137989309)

[4.3) Kiến trúc Inverted Residual Block 26](#_Toc137989310)

[4.4) Dropout 26](#_Toc137989311)

[4.5) GlobalAveragePooling 27](#_Toc137989312)

[4.6) Kiến trúc mô hình 29](#_Toc137989313)

[Chương 5: HUẤN LUYỆN MÔ HÌNH 30](#_Toc137989314)

[5.1) Bộ dữ liệu 30](#_Toc137989315)

[5.2) Độ đo và hàm tối ưu 30](#_Toc137989316)

[5.2.1) Đo lường sự mất mát 30](#_Toc137989317)

[5.2.2) Đo lường kết quả 30](#_Toc137989318)

[5.2.3) Hàm tối ưu 31](#_Toc137989319)

[5.3) Hiệu chỉnh siêu tham số 31](#_Toc137989320)

[5.4) Môi trường huấn luyện 31](#_Toc137989321)

[5.5) Kết quả huấn luyện 32](#_Toc137989322)

[5.6) Kết quả trên tập test 33](#_Toc137989323)

[Chương 6: ỨNG DỤNG WEB 34](#_Toc137989324)

[6.1) Kiến trúc ứng dụng 34](#_Toc137989325)

[6.2) Công nghệ sử dụng 34](#_Toc137989326)

[6.1.1) FastAPI 34](#_Toc137989327)

[6.1.2) P5.js 35](#_Toc137989328)

[6.1.3) Char.js 35](#_Toc137989329)

[6.1.4) TensorFlow Lite 36](#_Toc137989330)

[6.3) Chức năng của web 36](#_Toc137989331)

[6.4) Quá trình xử lý 37](#_Toc137989332)

[6.4.1) Draw sketch 37](#_Toc137989333)

[6.4.2) Erase stroke 38](#_Toc137989334)

[6.4.3) Get bounding box 38](#_Toc137989335)

[6.4.4) Tạo ảnh 40](#_Toc137989336)

[6.4.5) Predict label 41](#_Toc137989337)

[6.4.6) Show chart 41](#_Toc137989338)

[6.4.7) Sử dụng Bing 42](#_Toc137989339)

[Chương 7: KẾT LUẬN 43](#_Toc137989340)

[Chương 8: Tài liệu tham khảo 44](#_Toc137989341)

[PHÂN CHIA CÔNG VIỆC 45](#_Toc137989342)

Chương 1: TỔNG QUAN

* 1. Giới thiệu đề tài

Ngày nay với sự phát triển của Internet, mọi người thường có xu hướng học tập qua mạng bằng cách sử dụng chiếc máy tính để làm việc và học tập. Không chỉ người lớn mà còn có cả trẻ em cũng đang dần làm quen với việc học tập mọi thứ qua các trang mạng giáo dục trên không gian Internet. Để đáp ứng được sự tò mò, học hỏi của trẻ em trong việc học thêm những vật thể trở nên đa dạng và phong phú hơn. Nhóm em đã đề xuất đề tài mang tên “Sketch Search” là trang web(hình bên dưới) với sứ mệnh tạo ra môi trường hội họa sáng tạo cho trẻ em, và dựa vào các nét vẽ để tìm kiếm các vật thể có khả năng liên quan đến.

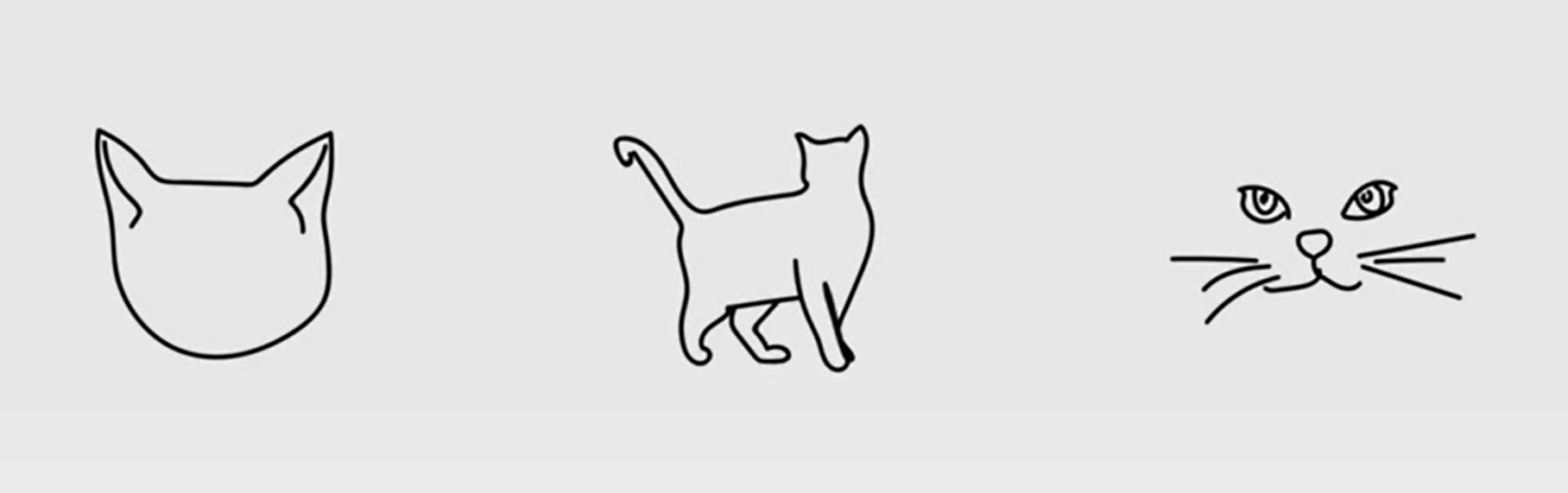


* 1. Lý do chọn đề tài

Nhóm chúng em thích khám phá và sáng tạo những điều mới thay vì lựa chọn một số đề tài cũ như nhận diện biển số xe hay nhận dạng trái cây,.... Tụi em muốn tạo ra một sản phẩm có thể chơi và nghịch được ngay trong lúc làm sản phẩm.  
Trẻ em ngày nay tiếp xúc với công nghệ từ khi còn nhỏ, với đồ án này giúp trẻ em có những trải nghiệm hội hoạ mới lạ và thú vị từ những vật thể được gợi ý từ bức vẽ, từ đó bọn trẻ có thể hiểu được cách những thứ mà nó vẽ sẽ trông như thế nào.Điều này tạo ra sự hứng thú cho trẻ nhỏ hơn, bởi vì con người chúng ta thường tiếp thu hình ảnh nhanh hơn so với các dạng kiến thức khác.Việc này sẽ giúp trẻ nhỏ có thể vừa chơi vừa học, vừa có thể vẽ, vừa có thể học thêm được những vật thể mới thông qua những bức ảnh.  
Ngoài ra, người dùng tìm hiểu về thứ mà họ đã nhìn thấy nhưng không biết tên gọi. Đồ án này cố gắng tìm ra những thông tin liên quan nhất hoặc cung cấp hình ảnh tương tự.

Nhóm em muốn khám phá xem thử liệu rằng 1 bức hình có cùng 1 chủ thể liệu vẽ bằng các cách khác nhau liệu nó còn dự đoán được hay không ?

Ví dụ: Đây là 3 bức ảnh về chú mèo được vẽ từ một người nhưng chúng ta có thể thấy chúng hoàn toàn khác nhau.



* 1. Đối tượng và phạm vi đề tài

Đối tượng: Tất cả mọi người đều có thể sử dụng Sketch Search để sử dụng tìm kiếm những hình ảnh thông qua bức vẽ của mình.Đặc biệt, trẻ em là đối tượng được khuyến khích sử dụng cho mục đích học tập và giải trí.

Phạm vi bài toán: Tìm kiếm hình ảnh của các vật thể liên quan đến bức vẽ được cung cấp bởi người dùng.

* 1. Thách thức đề tài

Khi xây dựng và phát triển đề tài này, nhóm đã gặp không ít khó khăn về tập dữ liệu, về vấn đề tập dữ liệu nhóm sẽ đề cập kĩ hơn ở chương 2. Nhìn chung bộ dữ liệu mà nhóm sử dụng cho đề tài “Sketch Search” là bộ dữ liệu được thu thập bởi hàng triệu nét vẽ tay của hàng triệu người. Vì bộ dữ liệu là hình ảnh được vẽ bằng tay nên sẽ có một số dữ liệu không chính xác ở một lớp nhất định



Có thể xem qua hình ảnh bên trên là lớp của con hổ, nhưng ta có thể thấy rằng người dùng khi vẽ khá nhiều hình ảnh không liên quan hoặc không giống so với con hổ. Đây gọi là sự nhiễu của dữ liệu mà nhóm đang gặp phải.

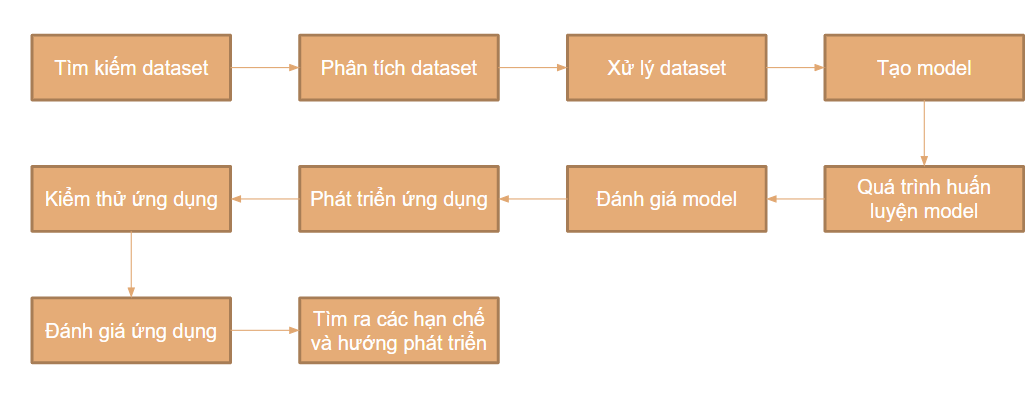
Đối mặt với sự khó khăn về mặt dữ liệu, nhóm hiển nhiên cũng gặp phải khó khăn kéo theo sau đó là lựa chọn mô hình và xây dựng kiến trúc model như thế nào ? Để đảm bảo được nhiệm vụ của hệ thống thực hiện tốt nhất. Về phần giải quyết mô hình sẽ được nhóm trình bày rõ hơn ở chương 3.

Một khó khăn không kém là lấy hình ảnh từ trang web được vẽ bằng tay bởi người dùng. Đây cũng là khó khăn không kém vì phần hình ảnh vào từ ứng dụng web phải có tính nhất quán với dữ liệu hình ảnh mà mô hình đã được học.Nếu không sẽ dẫn đến sự mất đi tính thống nhất giữa dữ liệu đã được học và dữ liệu thực tế cho mô hình kiểm thử.

* 1. Hướng giải quyết tổng quát

Với những khó khăn chung đã được trình bày ở trên, nhóm đã cùng nhau thảo luận để đưa ra giải pháp chung với việc sẽ không sử dụng mô hình được huấn luyện sẵn.Bởi vì các mô hình huấn luyện sẵn ở mảng Computer Vision hầu như được huấn luyện trên bộ dữ liệu ImageNet. Nhưng với tập dữ liệu trong đề tài lần này, thật sự không phù hợp với mô hình được huấn luyện sẵn.Vì vậy nhóm đã quyết định xây dựng mô hình kiến trúc lớp phù hợp với bộ dữ liệu dành cho đề tài Sketch Search.

* 1. Quá trình thực hiện đề tài



Chương 2: TẬP DỮ LIỆU

2.1) Giới thiệu bộ dữ liệu

Bộ dữ liệu được sử dụng trong đề tài Sketch Search là **QuickDraw**, đây là bộ dữ liệu khá nổi tiếng được thu thập và xây dựng từ nhiều hình ảnh được vẽ bằng tay của hàng triệu người trên thế giới về nhiều vật thể từ đơn giản đến phức tạp.Bộ dữ liệu này thường được các nhà nghiên cứu về lĩnh vực học sâu, học máy sử dụng để huấn luyện mạng nơ-ron thần kinh cho một số bài toán như nhận dạng và phân loại vật thể.

**QuickDraw** là một bộ dữ liệu lớn chứa hình vẽ đơn giản và mỗi hình ảnh có kích thước 28x28, được tạo ra bởi người dùng từ nhiều nền tảng và ứng dụng khác nhau. Bộ dữ liệu được Google phát triển để hỗ trợ nghiên cứu về trí tuệ nhân tạo và học máy. Bộ dữ liệu này bao gồm hơn 50 triệu hình vẽ từ 345 loại đối tượng khác nhau, được lưu dưới dạng chuỗi điểm trên không gian hai chiều. Mỗi hình vẽ được gắn nhãn với tên đối tượng tương ứng.  
QuickDraw dataset cung cấp độ đa dạng về hình dạng và phong cách vẽ, và có sẵn công khai để truy cập thông qua Google Cloud và các nguồn tài nguyên mở khác. Điều quan trọng nhất là bộ dữ liệu này bao gồm những hình vẽ của rất nhiều người trên thế giới đã vẽ ra, nên nó gần giống với những gì đứa trẻ có thể vẽ.

Có thể thấy việc lựa chọn bộ dữ liệu QuickDraw khá phù hợp với chủ đề Sketch Search theo như yêu cầu đã được giới thiệu ở chương 1.

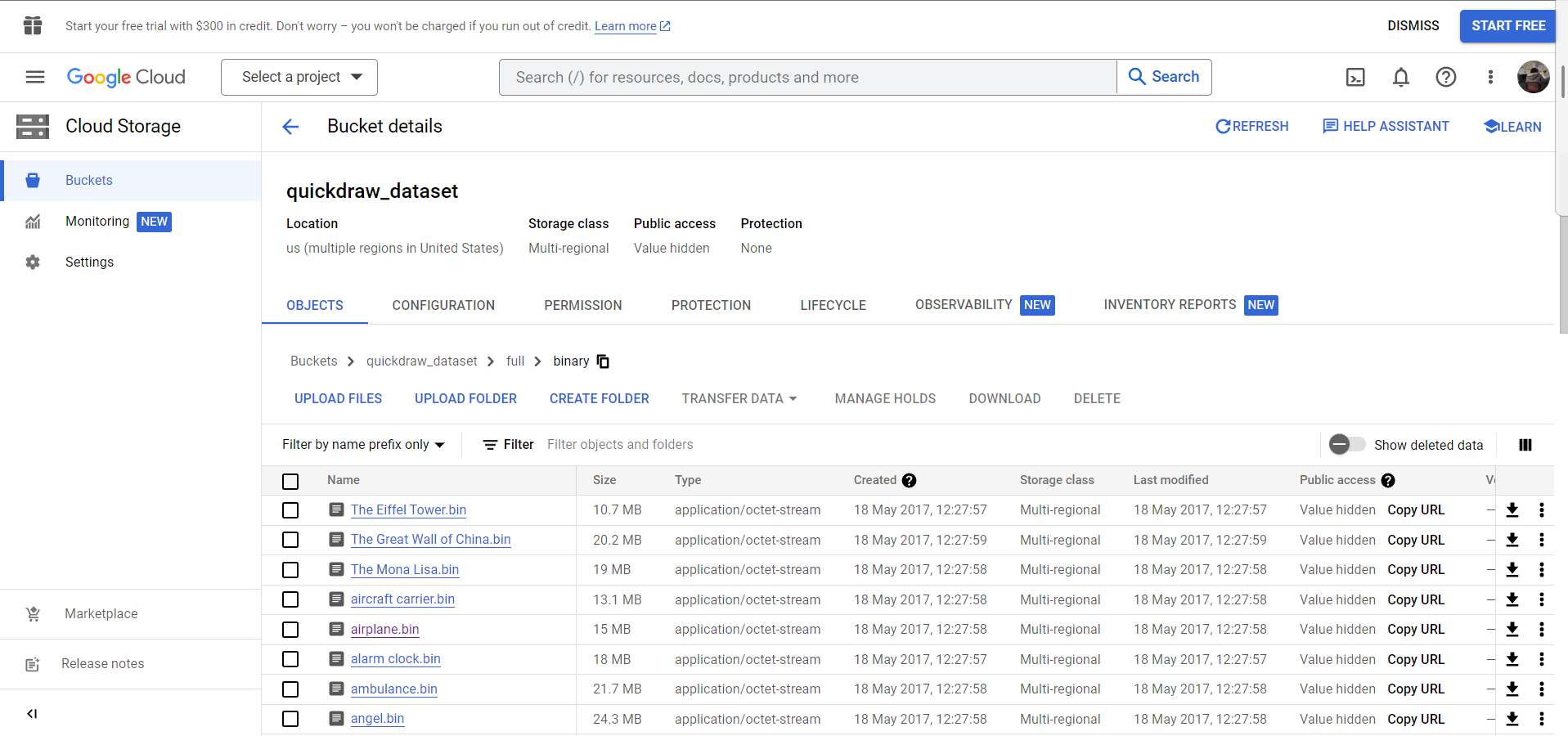


Nguồn dữ liệu: <https://quickdraw.withgoogle.com/data>

Bộ dữ liệu được lưu trữ dưới dạng một số dạng files như sau:

* [Raw files](https://console.cloud.google.com/storage/browser/quickdraw_dataset/full/raw) (.ndjson)
* [Simplified drawings files](https://console.cloud.google.com/storage/browser/quickdraw_dataset/full/simplified) (.ndjson)
* [Binary files](https://console.cloud.google.com/storage/browser/quickdraw_dataset/full/binary) (.bin)
* [Numpy bitmap files](https://console.cloud.google.com/storage/browser/quickdraw_dataset/full/numpy_bitmap) (.npy)

Nhóm chúng em sử dụng file kiểu .bin để lấy dữ liệu từ Google Cloud Storage



Với việc sử dụng file.bin mang lại lợi ích trong việc tải và đóng gói hiệu quả, các dữ liệu trong file.bin này đã được tiền xử lý.Đây là ví dụ về việc mở một file.bin sử dụng ngôn ngữ lập trình Python [examples/binary\_file\_parser.py](https://github.com/googlecreativelab/quickdraw-dataset/blob/master/examples/binary_file_parser.py).

2.2) Lý do lựa chọn

Dataset đóng vai trò cực kỳ quan trọng. Đúng như câu thành ngữ **"rubbish in, rubbish out"** (rác vào, rác ra), chất lượng và tính đại diện của dataset ảnh hưởng trực tiếp đến hiệu suất và độ chính xác của mô hình. Nó cung cấp dữ liệu để mô hình học và khám phá, đảm bảo áp dụng kiến thức vào các tình huống mới. Đa dạng và chất lượng dữ liệu đóng vai trò quan trọng để tránh overfitting và đạt được tổng quát hóa tốt hơn.  
Yêu cầu về dataset cho đề tài:  
 1. Nó phải dễ vẽ  
 2. Nó phải nhiều lớp  
 3. Số lượng dataset phải lớn và có sự cân bằng giữa các lớp  
 4. Các dataset phải có cùng kích thước   
Dù có nhiều dataset liên quan đến chủ đề của tụi em, tuy nhiên nó lại không phù hợp với yêu cầu trên.



Hình ảnh trên là một ví dụ về tập dữ liệu Imagenet-Sketch, có thể thấy rằng bộ dữ liệu với những hình vẽ vô cùng chi tiết và đầy đủ. Với lợi thế về sự dồi dào của bộ dữ liệu có 50000 ảnh với 1000 lớp, tuy có sự đáp ứng về số lượng và sự cân bằng của dữ liệu hình ảnh ở mỗi lớp.Nhưng khi xét về khía cạnh hình ảnh thì những bức ảnh này quá chi tiết và thực sự khó vẽ so với tiêu chí yêu cầu đã được đặt ra của đề tài.

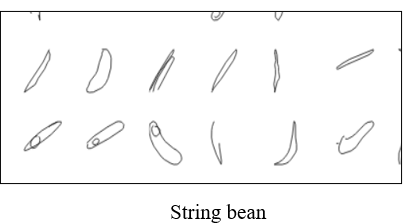
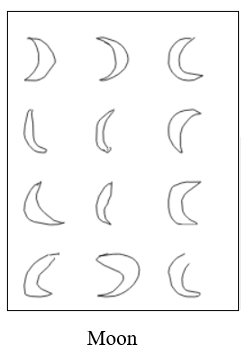
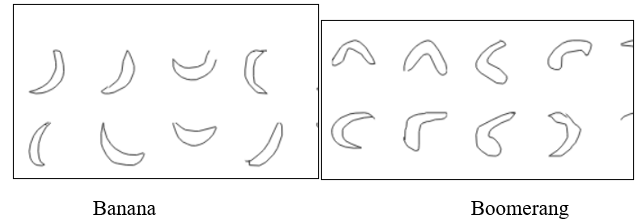
Vì vậy, nhóm đã cùng nhau thảo luận và thấy rằng tập dữ liệu QuickDraw rất thích hợp cho cho đề tài Sketch Search bởi tiêu chí đầu tiên cần được ưu tiên là hình ảnh phải dễ vẽ.Và QuickDraw đáp ứng được điều đó như đã giới thiệu ở phần 2.1.

2.3) Phân tích bộ dữ liệu

Với lợi thế về dữ liệu là những hình vẽ bằng tay khá phù hợp cho yêu cầu của ứng dụng web Sketch Search. Nhưng bộ dữ liệu vẫn còn tồn đọng một số vấn đề khó khăn. Bây giờ, chúng ta sẽ cùng nhau đi phân tích chi tiết hơn về ưu điểm và nhược điểm của bộ dữ liệu:

* Ưu điểm:
  + **Số lượng lớn:** Quickdraw dataset chứa hàng chục triệu hình vẽ từ hàng triệu người dùng, tạo ra một nguồn tài nguyên lớn để huấn luyện mô hình học máy.
  + **Đa dạng:** Với đa dạng hóa nguồn dữ liệu từ nhiều nguồn và người dùng, Quickdraw cung cấp sự đa dạng trong cách vẽ và phong cách của từng người. Điều này giúp mô hình học máy hiểu và nhận diện các đối tượng từ nhiều góc độ khác nhau.
  + **Tiện lợi và khả năng sử dụng:** Quickdraw dataset đã được xử lý và chuyển đổi thành dạng vector, giúp việc sử dụng nó trở nên thuận tiện và linh hoạt trong các nhiệm vụ học máy.
* Nhược điểm:
  + **Cần xử lý và làm sạch dữ liệu:** Do chất lượng không đồng nhất, người sử dụng Quickdraw cần tiến hành xử lý và làm sạch dữ liệu trước khi sử dụng. Điều này đòi hỏi thêm công đoạn tiền xử lý để loại bỏ dữ liệu không chính xác hoặc nhiễu.
  + **Sự thiếu cân đối:** Một số loại hình vẽ có số lượng nhiều hơn so với các loại khác, tạo ra sự không cân đối trong tập dataset. Điều này có thể làm giảm hiệu quả của một số mô hình huấn luyện trên tập dữ liệu này.
  + **Phụ thuộc vào phong cách vẽ người dùng:** Mỗi người dùng có một phong cách vẽ riêng, điều này có thể dẫn đến sự không nhất quán trong cách đại diện cho các đối tượng trong tập dataset.
  + **Nhược điểm về sự mất cân bằng dataset:** Không phải là một vấn đề quá lớn và dễ dàng giải quyết được có thể bằng cách tăng cường dữ liệu hoặc xóa bớt dữ liệu cho các lớp có số lượng hình bằng nhau.
  + **Nhược điểm về phụ thuộc vào phong cách vẽ người dùng**: Nó sẽ là một thử thách rất lớn về việc chọn mô hình phù hợp.

VD: Ở đây có 4 lớp tuy nhiên khi nhìn chúng thì khá giống nhau tuy nhiên vẫn thấy sự khác biệt:



* + **Nhược điểm về việc phải xử lý dữ liệu:** Đây là một vấn đề bình thường vì không ai cũng có thể có tài năng hội họa bẩm sinh và trí tưởng tượng phong phú được.

VD: Có thể thấy có rất nhiều người vẽ con hổ tuy nhiên không ra hình hài con hổ thực sự, điều này khác với bộ ImageNet-sketch về mặt hình ảnh nhưng chính sự khó khăn này mới phù hợp với sketch search



2.4) Xử lý bộ dữ liệu

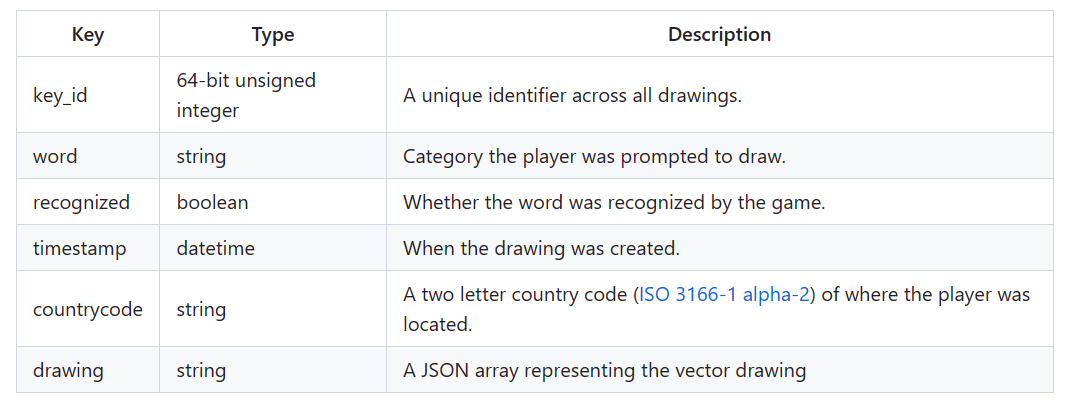
2.4.1) Thu thập dữ liệu

Vì số lượng dữ liệu rất lớn, để tiết kiệm dung lượng Google đã nén chúng lại thành các file có định dạng **.ndjson**, **.bin** và **.npy** để tiết kiệm không gian. Tuy nhiên, việc đọc các file này không quá phức tạp vì Google đã cung cấp hướng dẫn. Với cách lưu trữ ảnh dưới dạng vector điều này cũng giúp đảm bảo rằng tất cả các hình vẽ trong tập dữ liệu có cùng độ dày của đường vẽ khi sử dụng hàm draw trong thư viện **PIL**. Điều này, giúp cải thiện quá trình huấn luyện mô hình, vì mô hình có thể học được những đặc điểm chung và tinh chỉnh hơn.

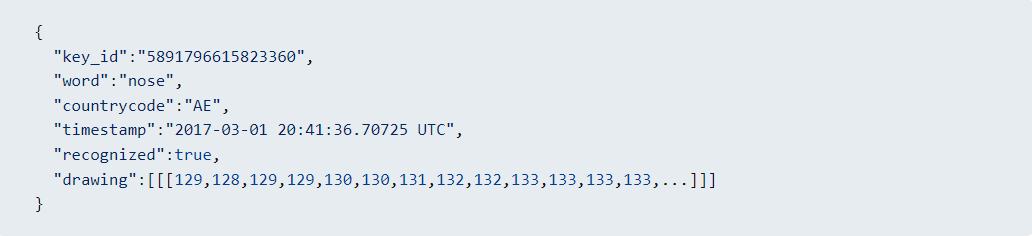
2.4.2) Giải quyết nhược điểm

Ở phần này, nhóm sẽ đề cập đến các cách giải quyết liên quan đến bộ dữ liệu.Như đã đề cập ở phần trên, chúng ta có thể thấy bộ dữ liệu có khá nhiều nhược điểm từ sự mất cân bằng, phụ thuộc vào người vẽ cho đến xử lý dữ liệu nhiễu ở mỗi lớp như thế nào cho phù hợp và đạt đúng yêu cầu đề ra của đề tài này.

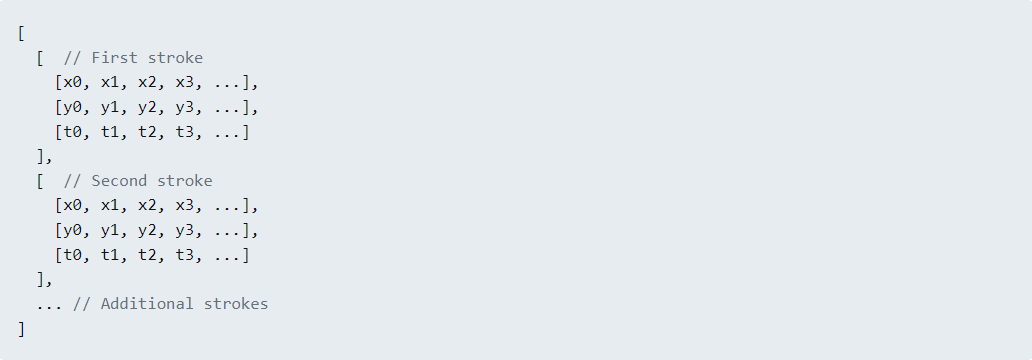
Đầu tiên, chúng ta cùng đến với giải quyết vấn đề về xử lý dữ liệu nhiễu. Vì đây là bộ dữ liệu của Google nên đã lường trước được mọi thứ sẽ xảy ra trong trường của dataset sẽ có thuộc tính như sau :



Dưới đây là cách lưu trữ dưới dạng file. njson theo thể loại. Mỗi dòng là thông tin của một dữ liệu ảnh



Hình thức lưu trữ của drawing như sau:



Trong đó:

* x0, x1, x2…., xn: tập các toạ độ x của một nét vẽ
* y0, y1, y2…., yn: tập các toạ độ y của một nét vẽ
* t0, t1, t2…., tn: là thời điểm của mỗi điểm ảnh xuất hiện khi được người dùng vẽ

Chúng ta dễ thấy rằng có 1 thuộc tính trong dataset đó là recognized. Thuộc tính này được sử dụng để phân loại và đánh giá độ chính xác của các bức vẽ trong quá trình huấn luyện và thử nghiệm mô hình nhận diện đối tượng của google và ngoài ra còn có sự đóng góp của người dùng nên có thể phần nào hạn chế tối đa được dữ liệu rác.

Nhược điểm thứ hai, mà nhóm muốn đề cập đến là việc giải quyết cho sự mất cân bằng của tập dữ liệu. Đối với vấn đề này, nhóm không gặp nhiều khó khăn:

* Vì tổng số lượng dataset là hơn 50 triệu cho 345 lớp tuy nhiên sự mất cân bằng ở đây không hơn là với tỉ là lớn mà mỗi lớp đều có nhiều hơn 100000 dataset.
* Không thể lấy hết toàn bộ số lượng dataset mà có thể huấn luyện được vì nó cả chứa những dữ liệu rác và dù lọc ra thì tài nguyên cũng sẽ không đủ để cung cấp để mô hình có thể học với số lượng dataset khổng lồ như thế. Nếu được thiết bị và tài nguyên tốt hơn thì nhóm có thể mở rộng số lượng hình ảnh ở mỗi lớp hơn.

Vì vậy, để phù hợp với điều kiện hiện tại mà nhóm đang có, ta chỉ có thể lấy vừa đủ để xây dựng tập train, valid và test.Về số lượng của tập sẽ được đề cập ở chương 5.

Nhược điểm thứ ba là vấn đề về phong cách vẽ của mỗi người, nhóm sẽ đề cập phần này chi tiết ở chương 4.

Chương 3: CƠ SỞ LÝ THUYẾT

3.1) Học đặc trưng

3.1.1) Giới thiệu

-Trong học máy, học tính năng hoặc học đại diện là một tập hợp các kỹ thuật cho phép hệ thống tự động khám phá các biểu diễn cần thiết để phát hiện hoặc phân loại tính năng từ dữ liệu thô. Điều này thay thế kỹ thuật tính năng thủ công và cho phép một máy vừa tìm hiểu các tính năng vừa học thực hiện một tác vụ cụ thể.

3.1.1.1) Hướng tiếp cận:

Học đặc trưng được học thông qua các mô hình được giám sát hoặc không được giám sát:

* Trong học đặc trưng có giám sát, các đặc trưng được học bằng cách sử dụng dữ liệu đầu vào được gắn nhãn. Các ví dụ bao gồm mạng nơ-ron được giám sát, perceptron nhiều lớp và học từ điển (có giám sát).
* Trong học đặc trưng không có giám sát, các đặc trưng được học với dữ liệu đầu vào không được gắn nhãn. Các ví dụ bao gồm học từ điển, phân tích thành phần độc lập, tự động mã hóa, phân tích nhân tử ma trận và các hình thức phân cụm khác nhau.

3.1.1.2) Học giám sát:

Học đặc trưng được giám sát là học các đặc trưng từ dữ liệu được gắn nhãn.Nhãn dữ liệu cho phép hệ thống tính toán thuật ngữ lỗi, mức độ mà hệ thống không tạo ra nhãn, sau đó có thể được sử dụng làm phản hồi để sửa quá trình học (giảm / giảm thiểu lỗi). Dưới đây là các mô hình học có giám sát sử dụng kĩ thuật học đặc trưng.

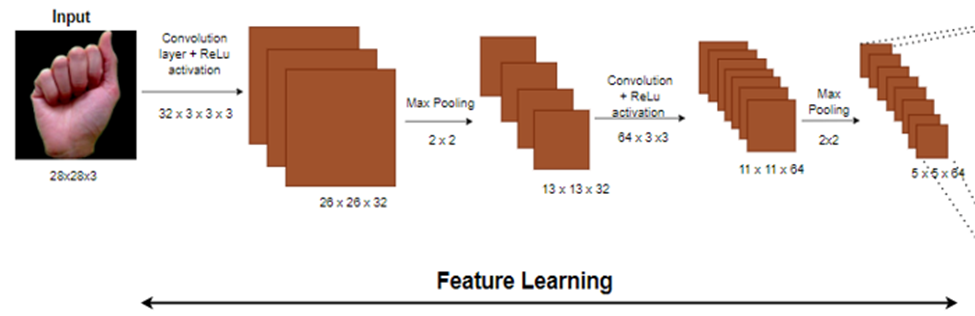
* Học từ điển có giám sát

Học từ điển phát triển một tập hợp (từ điển) các phần tử đại diện từ dữ liệu đầu vào sao cho mỗi điểm dữ liệu có thể được biểu diễn dưới dạng tổng trọng số của các phần tử đại diện. Các phần tử từ điển và trọng số có thể được tìm thấy bằng cách giảm thiểu sai số biểu diễn trung bình (trên dữ liệu đầu vào), cùng với sự chính xác hóa L1 trên các trọng số để tạo ra sự thưa thớt (tức là, biểu diễn của mỗi điểm dữ liệu chỉ có một vài trọng số khác không).

Học từ điển có giám sát khai thác cả cấu trúc bên dưới dữ liệu đầu vào và nhãn để tối ưu hóa các phần tử từ điển. Ví dụ, kỹ thuật học từ điển có giám sát này áp dụng việc học từ điển vào các vấn đề phân loại bằng cách cùng nhau tối ưu hóa các phần tử từ điển, trọng số để biểu diễn các điểm dữ liệu và các tham số của bộ phân loại dựa trên dữ liệu đầu vào. Cụ thể, một bài toán tối thiểu hóa được xây dựng, trong đó hàm mục tiêu bao gồm lỗi phân loại, lỗi biểu diễn, chính quy L1 trên các trọng số đại diện cho mỗi điểm dữ liệu (để cho phép biểu diễn dữ liệu thưa thớt) và chính quy L2 trên các tham số của trình phân loại.

* Mạng nơron

Mạng nơ-ron là một họ các thuật toán học sử dụng một "mạng" bao gồm nhiều lớp các nút được kết nối với nhau. Nó được lấy cảm hứng từ hệ thần kinh động vật, nơi các nút được xem như tế bào thần kinh và các cạnh được xem như khớp thần kinh. Mỗi cạnh có trọng số liên quan và mạng xác định các quy tắc tính toán để truyền dữ liệu đầu vào từ lớp đầu vào của mạng đến lớp đầu ra. Một chức năng mạng được liên kết với mạng nơ-ron đặc trưng cho mối quan hệ giữa các lớp đầu vào và đầu ra, được tham số hóa bởi các trọng số. Với các chức năng mạng được xác định phù hợp, các tác vụ học tập khác nhau có thể được thực hiện bằng cách giảm thiểu hàm chi phí trên chức năng mạng (trọng số). Mạng nơ ron nhiều lớp có thể được sử dụng để thực hiện việc học đặc trưng, vì chúng học cách biểu diễn đầu vào của chúng ở (các) lớp ẩn, sau đó được sử dụng để phân loại hoặc hồi quy ở lớp đầu ra. Kiến trúc mạng phổ biến nhất của loại này là mạng Siamese.

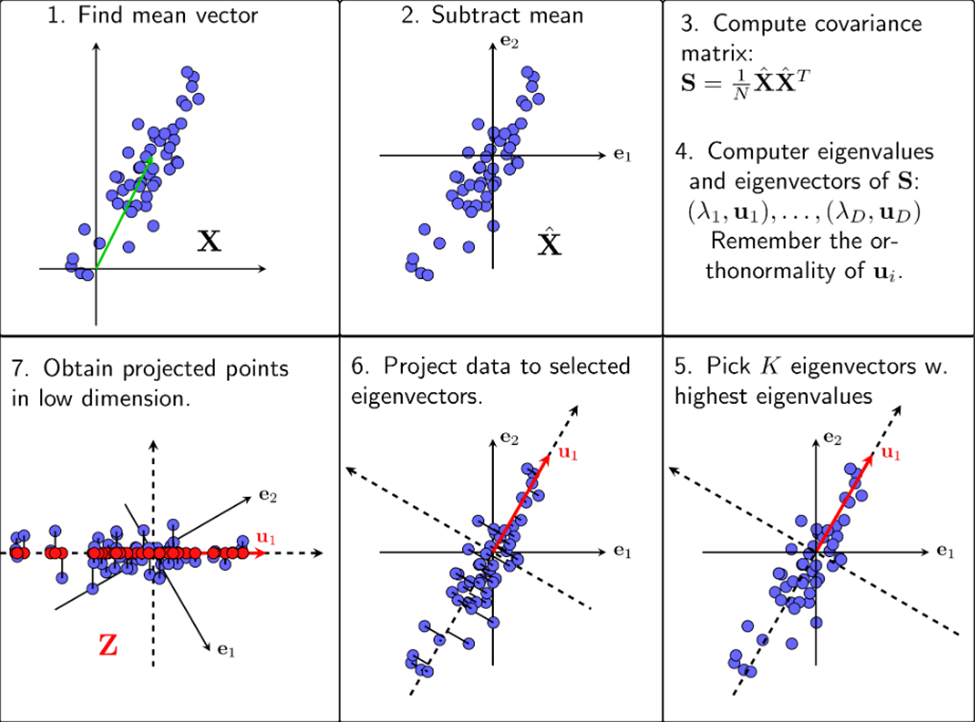


***Quá trình học đặc trưng của mô hình CNN***

3.1.1.3) Học không giám sát:

Học đặc trưng không giám sát là học các đặc trưng từ dữ liệu không được gắn nhãn. Khi việc học đặc trưng được thực hiện theo cách không được giám sát, nó sẽ cho phép một hình thức học tập bán giám sát trong đó các đặc trưng được học từ tập dữ liệu chưa được gắn nhãn sau đó sẽ được sử dụng để cải thiện hiệu suất trong cài đặt được giám sát với dữ liệu được gắn nhãn. Dưới đây là mô hình học không giám sát có sử dụng kĩ thuật học đặc trưng

* Phân tích thành phần chính(PCA)
* Phân tích thành phần chính (PCA) thường được sử dụng để giảm kích thước. Cho một tập hợp n vectơ dữ liệu đầu vào không được gắn nhãn, PCA tạo ra p (nhỏ hơn nhiều so với kích thước của dữ liệu đầu vào) vectơ đơn lẻ bên phải tương ứng với giá trị kỳ dị p lớn nhất của ma trận dữ liệu, trong đó hàng thứ k của ma trận dữ liệu là vectơ dữ liệu đầu vào thứ k được dịch chuyển bằng giá trị trung bình mẫu của đầu vào (tức là trừ giá trị trung bình mẫu cho vectơ dữ liệu). Tương tự, các vectơ đơn lẻ này là các ký hiệu riêng tương ứng với p giá trị riêng lớn nhất của ma trận hiệp phương sai mẫu của các vectơ đầu vào. P vectơ số ít này là các vectơ đặc trưng được học từ dữ liệu đầu vào và chúng đại diện cho các hướng mà dữ liệu có các biến thể lớn nhất.
* PCA là một phương pháp học đặc trưng tuyến tính vì các vectơ số ít p là các hàm tuyến tính của ma trận dữ liệu. Các vectơ số ít có thể được tạo ra thông qua một thuật toán đơn giản với p lần lặp. Trong lần lặp thứ i, phép chiếu của ma trận dữ liệu trên ký hiệu riêng thứ (i-1) bị trừ đi và vectơ số một thứ i được tìm thấy là vectơ số ít bên phải tương ứng với số ít lớn nhất của ma trận dữ liệu dư.
* PCA có một số hạn chế. Đầu tiên, nó giả định rằng các hướng có phương sai lớn được quan tâm nhất, điều này có thể không đúng. PCA chỉ dựa vào các phép biến đổi trực giao của dữ liệu gốc và nó chỉ khai thác các khoảnh khắc bậc nhất và bậc hai của dữ liệu, điều này có thể không đặc trưng cho việc phân phối dữ liệu. Hơn nữa, PCA có thể giảm thứ nguyên một cách hiệu quả chỉ khi các vectơ dữ liệu đầu vào có tương quan với nhau (dẫn đến một vài giá trị riêng chi phối).



***Quy trình học đặc trưng của PCA***

3.2) Phân loại hình ảnh

3.2.1) Giới thiệu

Phân loại hình ảnh (Image classification) là tác vụ của thị giác máy tính, ở đó thuật toán xem xét và dán nhãn cho hình ảnh từ một bộ dữ liệu được xác định và đào tạo trước.

Thực tế, thị giác góp phần tạo nên 80-85% nhận thức của con người về thế giới. Hàng ngày, mỗi người phải thực hiện phân loại trên bất kỳ dữ liệu hình ảnh nào mà chúng ta bắt gặp.Do đó, mô phỏng nhiệm vụ phân loại với sự trợ giúp của mạng nơ-ron là một trong những ứng dụng đầu tiên của thị giác máy tính mà các nhà nghiên cứu nghĩ đến.

Dưới đây là ảnh mô phỏng quá trình về quy trình nhận dạng tổng quát:

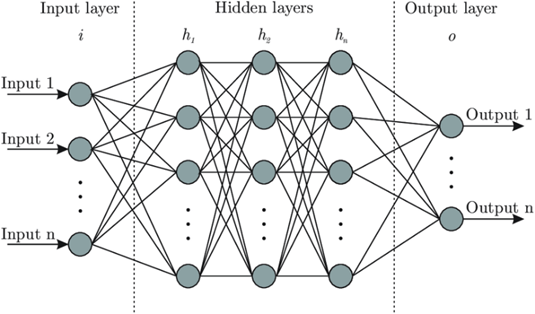


3.2.2) Hướng tiếp cận

Có nhiều thuật toán khác nhau được ứng dụng trong việc phân loại hình ảnh. Các thuật toán này được chia thành hai nhóm chính là học có giám sát (supervised learning) và học không giám sát (unsupervised learning).Ngoài ra, còn một cách học khác được gọi là bán giám sát (semi-supervised learning) nhưng ở chương này, nhóm sẽ không đề cập đến.

3.2.2.1 Phân loại có giám sát

* Trong học máy có giám sát, thuật toán được huấn luyện trên một tập hình ảnh đã được dán nhãn. Từ dữ liệu mẫu này, thuật toán có thể trích xuất thông tin, phục vụ phân loại ngay cả những hình ảnh chưa từng nhìn thấy trước đó.
* Xuyên suốt quá trình đào tạo, đặc điểm của ma trận hình ảnh sẽ được trích xuất dưới dạng dữ liệu quan trọng để đưa vào xử lý. Các đặc điểm này đại diện cho hình ảnh trong không gian chiều thấp (lower-dimensional feature space) và là cơ sở để thuật toán tiến hành phân loại.
* Trong quá trình đánh giá, các đặc điểm của ảnh thử nghiệm được thu thập và tái phân loại với sự hỗ trợ của mạng thần kinh nhân tạo. Hệ thống lúc này đã có thể nhận biết các đặc điểm điển hình của mọi lớp hình ảnh mà nó được đào tạo.
* Các phương pháp phân loại phổ biến dựa trên học có giám sát bao gồm:
  + Support Vector Machines
  + Decision Trees
  + K Nearest Neighbors
* Các mạng nơ-ron thường được sử dụng để phân loại hình ảnh có giám sát bao gồm AlexNet, ResNet, DenseNet và Inception.



***Phân loại hình ảnh sử dụng ANN***

Đối với phân loại có giám sát, việc dán nhãn dữ liệu đóng vai trò quan trọng. Độ chính xác của dữ liệu được dán nhãn quyết định phần lớn hiệu suất của mô hình học máy. Các thuật toán phân loại có giám sát có thể được chia thành hai mục nhỏ hơn dựa trên nhãn dữ liệu.

* Phân loại nhãn đơn
* Phân loại nhãn đơn (Single-label classification) là tác vụ phổ biến nhất trong phân loại ảnh có giám sát. Theo đó, mỗi hình ảnh được đại diện bởi một nhãn/chú thích (a single label or annotation). Mô hình xuất ra một giá trị hoặc dự đoán duy nhất cho mỗi hình ảnh mà nó xử lý.
* Đầu ra từ mô hình là mã hóa One-hot (từng giá trị được biến đổi thành các đặc trưng nhị phân chỉ chứa giá trị 1 hoặc 0). Mã hóa One-hot có độ dài bằng số lớp và giá trị biểu thị xác suất hình ảnh thuộc về lớp này.
* Hàm Softmax được sử dụng để đảm bảo các xác suất tổng bằng một và xác suất tối đa được chọn làm đầu ra của mô hình. Mặc dù Softmax không có giá trị về mặt dự đoán, nhưng nó giúp ràng buộc đầu ra giữa 1 và 0, nhờ vậy, có thể đánh giá độ tin cậy của mô hình từ điểm Softmax.
* Một số ví dụ về bộ dữ liệu phân loại nhãn đơn bao gồm MNIST, SVHN, ImageNet, v.v….
* Phân loại nhãn đơn có thể được xếp vào phân loại đa lớp (Multiclass classification) hoặc phân loại nhị phân (binary classification).
* Phân loại đa nhãn

- Phân loại đa nhãn là một tác vụ phân loại trong đó mỗi hình ảnh có thể chứa nhiều hơn một nhãn hoặc một số hình ảnh chứa đồng thời tất cả các nhãn.Phân loại đa nhãn xuất hiện phổ biến trong lĩnh vực xử lý hình ảnh y tế, khi một bệnh nhân có thể được chẩn đoán mắc nhiều bệnh dựa trên dữ liệu chụp X-quang.

3.2.2.2) Phân loại không giám sát

* Trong học máy không giám sát, thuật toán chỉ sử dụng dữ liệu thô để đào tạo.Các nhãn phân loại thường không xuất hiện trong kiểu học này và mô hình học bằng cách nhận dạng các mẫu trong tập dữ liệu huấn luyện.
* Giống như phân loại có giám sát, các phương pháp dựa trên không giám sát cũng liên quan đến bước trích xuất đặc điểm với các thông tin chi tiết nhất về hình ảnh. Các đặc điểm này sau đó được xử lý bằng các phương pháp phân cụm tham số (Gaussian Mixture Models) và phi tham số (K-means for Clustering) hoặc các thuật toán học không giám sát khác.
* Các thuật toán và kỹ thuật phân loại của thị giác máy tính không chỉ giới hạn trong dữ liệu hình ảnh 2D đơn giản mà còn mở rộng ra dưới dạng Video và ảnh 3D.

3.3) Tổng kết

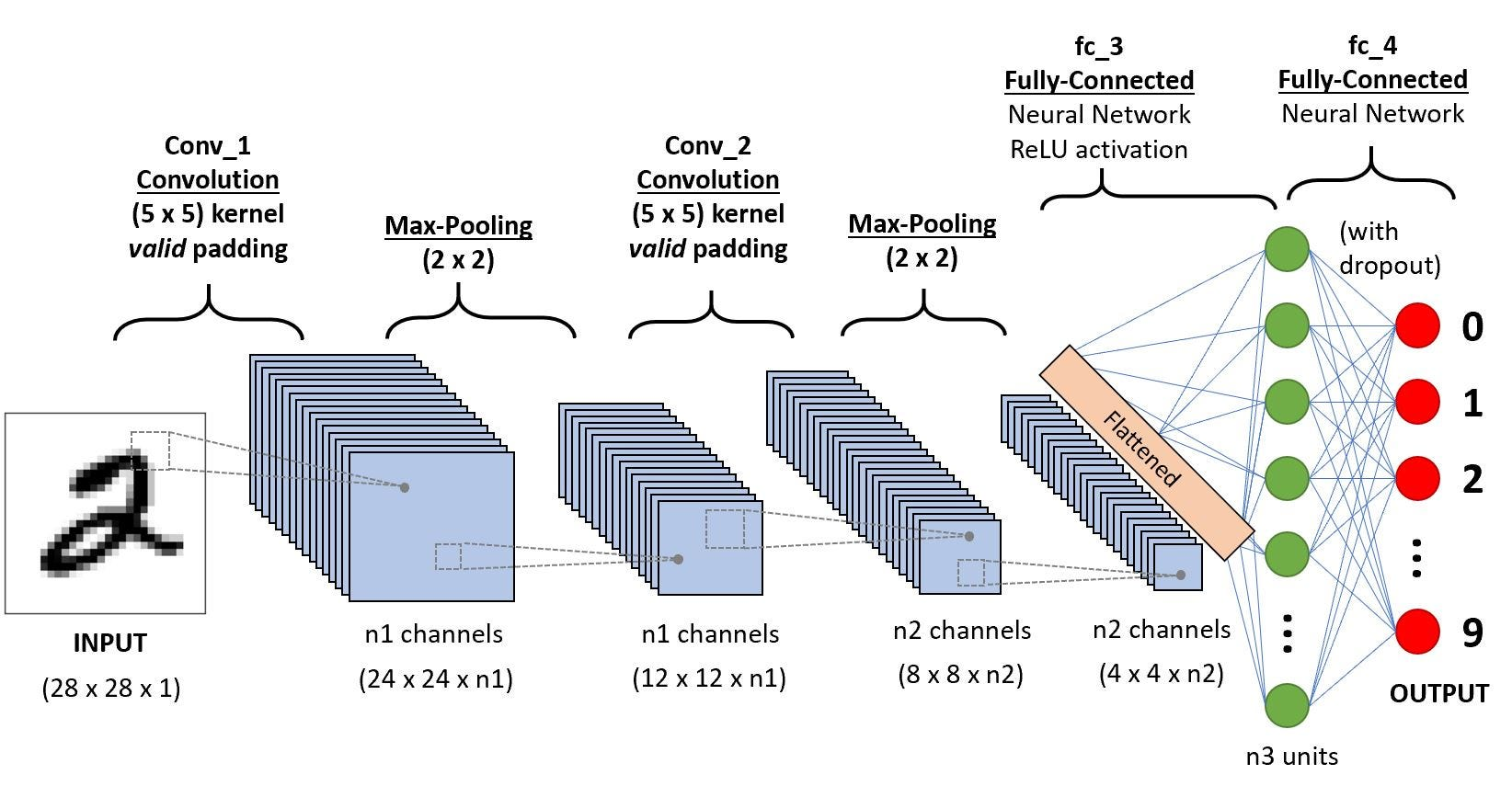
Với đề tài Sketch Search, như đã bàn luận ở các chương trước, bộ dữ liệu nhóm xây dựng là bộ dữ liệu đã được gán nhãn sẵn. Dựa vào các mục cơ sở lý thuyết đã bàn luận ở phía trên về Học đặc trưng và Phân loại hình ảnh, chúng ta kết luận với đề tài Sketch Search, sẽ được tiến hành tiếp cận Học đặc trưng và Phân loại hình ảnh theo hướng có giám sát. Điều này cũng dễ dàng chứng minh được là bởi vì bộ dữ liệu chúng ta xây dựng là bộ dữ liệu đã có nhãn và sau khi huấn luyện, chúng ta sẽ kiểm tra xem liệu mô hình của chúng ta đã dự đoán chính xác hay chưa ? Và việc lựa chọn tiếp cận bài toán này theo hướng học có giám sát là hoàn toàn hợp lý vì để đảm bảo yêu cầu tính chính xác của hệ thống như đã được trình bày. Sau khi cùng nhau tìm hiểu về cơ sở lý thuyết giúp ta tiếp cận bài toán này, chúng ta cùng đến chương tiếp theo của đề tài thú vị này nhé.

Chương 4: PHƯƠNG PHÁP

4.1) Lựa chọn hướng học

* Dựa vào bộ dữ liệu đã được đề cập ở chương 2, do số lượng hình ảnh nhiều nên lựa chọn deep learning sẽ tốt hơn so với machine learning vì Deep Learning có khả năng học từ dữ liệu lớn và rút trích đặc trưng phức tạp tự động, trong khi Machine Learning truyền thống yêu cầu việc chọn và xây dựng đặc trưng thủ công.
* Mô hình trong Deep Learning quan trọng bởi vì nó xác định cấu trúc và quy tắc để học từ dữ liệu, rút trích thông tin quan trọng, dự đoán và phân loại dữ liệu mới, cải thiện hiệu suất và khám phá tri thức ẩn.
* Như đã đề cập ở chương 3 phần Tổng kết, nhóm đã quyết định lựa chọn theo hướng học có giám sát, cụ thể là bài toán phân loại hình ảnh.Nhóm đề xuất một số kiến trúc mạng phổ biến của bài toán phân loại như sau:

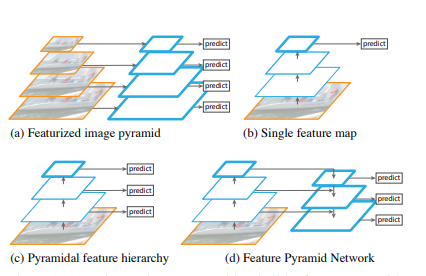
4.1.1) Mô hình CNN (Convolutional Neural Network)



Là một kiến trúc mạng nơ-ron sử dụng cho xử lý hình ảnh. Nó có khả năng rút trích đặc trưng và học các mẫu từ dữ liệu hình ảnh thông qua các lớp tích chập. CNN được sử dụng rộng rãi trong phân loại hình ảnh, phát hiện đối tượng và các tác vụ liên quan đến thị giác máy tính.

* Ưu điểm:
  + Cấu trúc dựa trên kiến trúc lưới thông qua việc sử dụng các lớp tích chập, giúp mô hình giữ được tính không gian của dữ liệu hình ảnh.
  + CNN có khả năng học được các đặc trưng cấp thấp và cấp cao từ dữ liệu hình ảnh thông qua các lớp tích chập và lớp gộp.
  + Có thể chia sẻ trọng số trong lớp tích chập để giảm số lượng tham số, làm giảm tình trạng overfitting.
  + Thích hợp cho việc xử lý hình ảnh với kích thước đầu vào cố định.
* Nhược điểm:
  + CNN không giữ được thông tin không gian trong cả quá trình học, có thể bỏ qua các mối quan hệ không gian giữa các vùng của hình ảnh.
  + Không thể chú ý đến mối quan hệ không gian xa giữa các điểm trong hình ảnh.

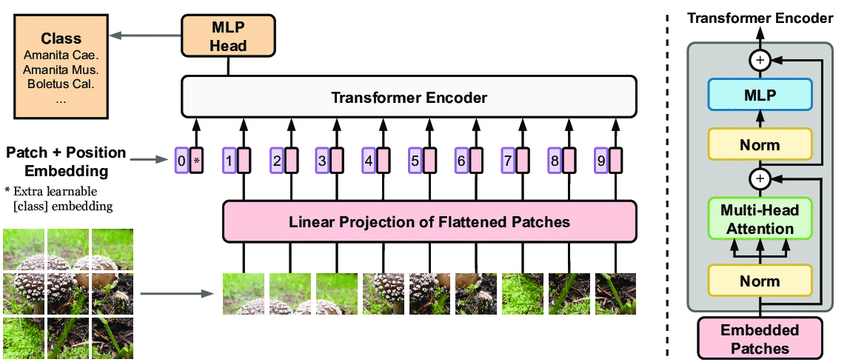
4.1.2) Mô hình FPN (Feature Pyramid Network)



Là một kiến trúc mạng nơ-ron được sử dụng để tạo ra các tầng đặc trưng đa tỉ lệ từ hình ảnh. Nó sử dụng thông tin từ các tầng đặc trưng khác nhau để nhận diện đối tượng ở các kích thước khác nhau trong hình ảnh và rút trích đặc trưng chi tiết.

* Ưu điểm:
  + Đa tầng đặc trưng: FPN tạo ra một hệ thống đa tầng các đặc trưng có kích thước khác nhau, giúp nhận diện đối tượng ở các tỷ lệ kích thước khác nhau trong hình ảnh. Điều này làm cho FPN rất hiệu quả trong việc nhận diện và phân đoạn các đối tượng có kích thước và tỷ lệ khác nhau.
  + Độ chính xác cao: FPN cung cấp các đặc trưng ở các tầng có độ chi tiết khác nhau, từ đó cung cấp thông tin chi tiết về các đối tượng trong hình ảnh. Điều này giúp tăng độ chính xác của các nhiệm vụ như nhận dạng đối tượng và phân đoạn hình ảnh.
* Nhược điểm:
  + Đòi hỏi thời gian huấn luyện và tính toán cao: FPN có cấu trúc phức tạp và yêu cầu nhiều tầng tích chập, điều này có thể làm tăng thời gian huấn luyện và tính toán so với các mô hình đơn giản hơn.
  + Độ phức tạp cài đặt: Cài đặt FPN có thể phức tạp hơn so với các mô hình khác, đòi hỏi kiến thức và kỹ năng cao hơn trong việc xây dựng và triển khai mạng.

4.1.3) Mô hình Vision Transformer(VIT)



Là một kiến trúc mạng nơ-ron sử dụng transformer để xử lý hình ảnh. Thay vì sử dụng các lớp tích chập truyền thống, VIT chia nhỏ hình ảnh thành các điểm chú ý và sử dụng transformer để học các mối quan hệ giữa chúng. Điều này cho phép VIT rút trích đặc trưng từ hình ảnh và đạt được kết quả tốt trong các tác vụ như phân loại hình ảnh và phát hiện đối tượng.

* Ưu điểm:
  + Vision Transformer giữ được thông tin không gian trong quá trình học bằng cách sử dụng phép biến đổi âm cực, cho phép mô hình nắm bắt mối quan hệ không gian xa trong hình ảnh.
  + Có thể áp dụng cho các bài toán về phân đoạn hình ảnh và phân loại đối tượng mà kích thước đầu vào không cố định.
  + Vision Transformer có khả năng mở rộng tốt đối với các tập dữ liệu lớn.
* Nhược điểm:
  + Yêu cầu tập dữ liệu huấn luyện lớn hơn so với CNN để đạt được hiệu suất tương tự.
  + Vision Transformer có số lượng tham số lớn hơn so với CNN, dẫn đến việc cần nhiều tài nguyên tính toán hơn để huấn luyện và triển khai mô hình.
  + Khả năng tổng quát hóa của Vision Transformer có thể không tốt bằng CNN đối với các tập dữ liệu nhỏ.

Sau khi xem xét về ưu điểm và nhược điểm của từng mạng, nhóm có kết luận như sau:

* Vision Transformer(ViT): Mô hình đạt kết quả tốt trên bộ dữ liệu lớn nhưng tốn nhiều tài nguyên để huấn luyện mô hình. Không thể sử dụng phương pháp học chuyển tiếp vì dữ liệu đầu vào của hình ảnh có kích thước 28x28 mà mô hình VIT lại cần 32x32 nên nếu muốn dùng mô hình VIT, phải tự tạo lại và huấn luyện từ đầu.Điều này tốn nhiều thời gian và tài nguyên huấn luyện.
* Feature Pyramid Network: Mạng được xây dựng cho bài toán phát hiện và nhận dạng đối tượng ở nhiều kích thước tỉ lệ khác nhau. Đồng thời như đã trình bày ở trên, mạng FPN có kiến trúc phức tạp và khó khăn trong việc xây dựng. Khi nhóm cân nhắc lại yêu cầu của đề tài Sketch Search, mục tiêu là phân loại hình ảnh với kích thước cố định nên việc sử dụng FPN không có nhiều lợi ích cho đề tài này.

Vì vậy nhóm quyết định lựa chọn mạng **Convolutional Neural Network** thuần, sau đó sẽ thiết kế cho phù hợp với bài toán đặt ra.

4.1.4) So sánh các mạng

Chúng ta cùng điểm lại bảng so sánh giữa 3 mạng đã nêu ở trên:

|  |  |  |  |
| --- | --- | --- | --- |
|  | CNN | VIT | FPN |
| Khả năng học | Tích chập cung cấp khả năng học các mẫu cục bộ và hiểu các đặc trưng không gian. | Transformer có khả năng học các mối quan hệ không gian và tương quan toàn cục trong hình ảnh. | Kết hợp thông tin từ các tầng đặc trưng khác nhau để học các đặc trưng ở các tỉ lệ khác nhau trong hình ảnh. |
| Tài nguyên | Tính toán phức tạp và yêu cầu tài nguyên cao, đặc biệt với các mạng CNN sâu và phức tạp. | Tính toán phức tạp hơn so với CNN truyền thống, do sử dụng transformer và yêu cầu tài nguyên lớn hơn. | Tính toán trung bình và yêu cầu tài nguyên trung bình. |
| Độ chính xác và hiệu suất | Được chứng minh là hiệu quả trong nhiều tác vụ xử lý hình ảnh và có thể đạt được độ chính xác cao. | Có hiệu suất cao trong một số tác vụ như phân loại hình ảnh trên các tập dữ liệu lớn, nhưng có thể yêu cầu nhiều dữ liệu huấn luyện hơn để đạt được kết quả tốt. | Đạt hiệu suất tốt trong việc phát hiện đối tượng và rút trích đặc trưng ở các tỉ lệ khác nhau trong hình ảnh. |

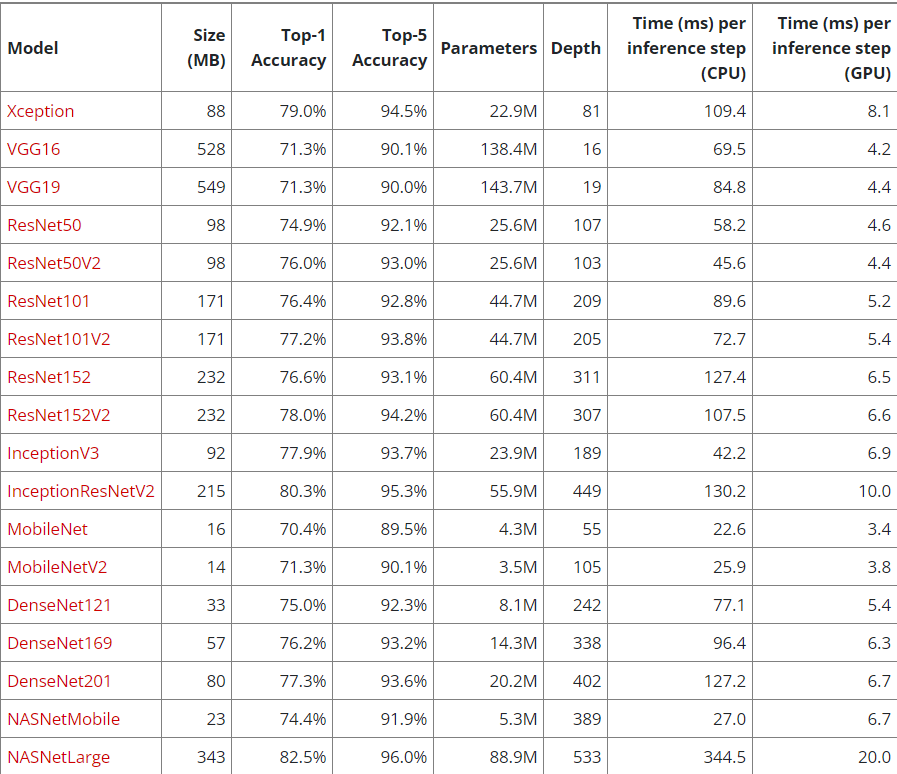
4.2) Downsampling CNN

* Downsampling trong CNN là giảm kích thước của đầu vào hình ảnh. Việc giảm kích thước giúp giảm độ phức tạp tính toán và số lượng tham số trong mô hình, từ đó tăng tốc độ huấn luyện và giảm nguy cơ overfitting. Downsampling loại bỏ thông tin dư thừa tuy nhiên nó cũng làm mất mát thông tin quan trọng.Vì vậy, lựa chọn cách downsampling có vai trò quan trọng trong kiến trúc mạng CNN.

4.2.1) Tìm hiểu về các phương pháp Downsampling

* Như vậy, chúng ta đã lựa chọn được mô hình CNN, trong phần này nhóm sẽ trình bày về các lớp phổ biến thường được sử dụng trong phương pháp Downsampling.
  + **Pooling**: Pooling là một quá trình tổng hợp thông tin từ các vùng dữ liệu nhỏ thành một đại diện tổng quát hơn, từ đó giảm kích thước dữ liệu và đồng thời giữ lại những thông tin quan trọng. Tuy nhiên, việc sử dụng pooling cũng có một số hạn chế. Bằng cách giảm kích thước dữ liệu, pooling có thể dẫn đến mất mát thông tin chi tiết, đặc biệt là khi các vùng nhỏ bị co lại quá mức. Điều này có thể làm giảm khả năng phân loại và phân đoạn chính xác của mô hình.
  + **Inverted residual blocks**: Inverted residual blocks là một cấu trúc quan trọng trong mạng neural, được sử dụng để giảm chiều dữ liệu và cải thiện hiệu suất mô hình. Chúng kết hợp việc sử dụng convolution và bottleneck layer để giảm số chiều dữ liệu và tạo ra biểu diễn đặc trưng tổng quát hơn. Việc sử dụng inverted residual blocks giúp giảm số lượng tham số và tính toán trong mô hình, đồng thời cải thiện khả năng học của mạng.
  + **Strided Convolution**: Thay vì sử dụng convolutional layer với stride bằng 1 và sau đó áp dụng pooling, có thể sử dụng convolutional layer với stride lớn hơn để đồng thời thực hiện phép giảm chiều và tính toán đặc trưng.
  + **Grouped Convolution:** Trong quá trình tính toán convolution, thay vì áp dụng một kernel lớn trên toàn bộ dữ liệu đầu vào, có thể chia nhỏ dữ liệu đầu vào thành các nhóm và áp dụng convolution độc lập trên từng nhóm. Sau đó, kết hợp các đầu ra từ các nhóm này. Phương pháp này có thể giảm chiều dữ liệu một cách hiệu quả.
  + **Compression:** Có thể sử dụng các phép toán như PCA (Principal Component Analysis) hoặc t-SNE (t-Distributed Stochastic Neighbor Embedding) để nén đặc trưng và giảm số chiều của dữ liệu. Tuy nhiên, việc này thường đòi hỏi tính toán phức tạp và có thể làm mất đi một số thông tin quan trọng.
  + **Autoencoders**: Mô hình autoencoder là một loại mạng nơ-ron có khả năng học biểu diễn sâu (deep representation learning) thông qua việc nén và tái tạo dữ liệu. Bằng cách huấn luyện một autoencoder với số lượng unit ở lớp ẩn giữa ít hơn số lượng unit ở lớp đầu vào, có thể giảm chiều dữ liệu mà vẫn giữ lại các đặc trưng quan trọng.
* Có thể thấy, có rất nhiều phương pháp Downsampling khác nhau được đề cập ở trên. Nhóm chỉ tập trung nghiên cứu 2 phương pháp: Pooling và Inverted Residual block, đây là 2 phương pháp downsampling được sử dụng trong những mạng học sâu nổi tiếng như:
  + Pooling: AlexNet, VGGNet, DenseNet,…
  + Inverted residual blocks: MobileNet, MnasNet, EfficientNet,…

Tất cả các mạng trên đều đạt kết quả **state-of-the-art** trong lĩnh vực phân loại hình ảnh, được huấn luyện và kiểm thử trên bộ dữ liệu ImageNet nổi tiếng. Dưới đây là kết quả được cung cấp trên Keras:

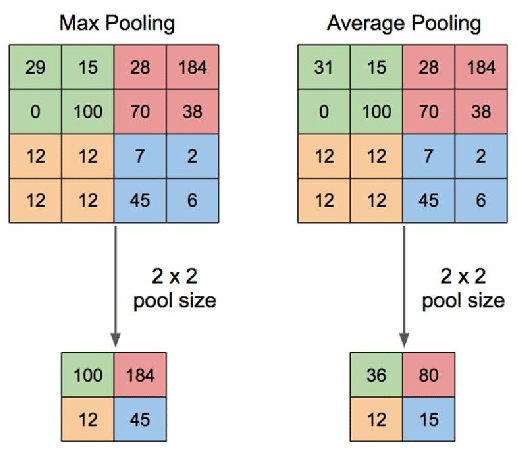


4.2.2) Pooling và Inverted Residual Block

4.2.2.1) Tìm hiểu về Pooling

Pooling là phương pháp giảm chiều dữ liệu phổ biến trong mạng thần kinh học sâu, ta thường thấy trong mạng như CNN. Pooling thường đứng sau lớp tích chập Convolution, với nhiệm vụ giảm chiều dữ liệu và gom tóm tắt lại đặc trưng tại 1 vùng nào đó của bức ảnh. Có 2 cách Pooling thường gặp như sau:

* Average Pooling: Tính trung bình cộng các giá trị của các kênh dựa trên phạm vi của kích thước ma trận tích chập vào.
* Max Pooling: Tính giá trị kênh màu lớn nhất trên phạm vi của kích thước ma trận tích chập vào.



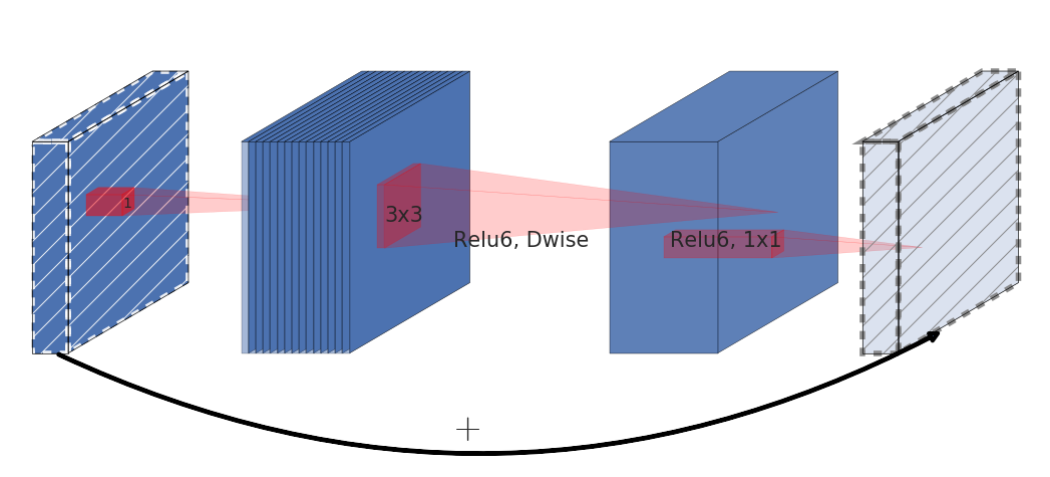
***Minh hoạ cho 2 cách Pooling***

Ngoài hai cách trên, ta có thể gặp phải nhiều kiểu pooling khác như: GlobalAveragePooling2D, GlobalMaxPooling2D, GlobalAveragePooling3D và GlobalMaxPoolin3D và nhiều kiểu Pooling khác nữa.

4.2.2.2) Tìm hiểu về Inverted Residual Block(IRB)

Là một khối xây dựng quan trọng trong kiến trúc mạng MobileNet, được đề xuất bởi tác giả Andrew G. Howard và cộng sự vào năm 2017. IRB là một khối cấu trúc đặc biệt trong việc giảm kích thước và tính toán của mạng CNN, đồng thời vẫn duy trì hiệu suất cao.

IRB được thiết kế để giảm số lượng tham số và tính toán trong quá trình huấn luyện và triển khai mạng. Điều này làm cho mô hình phù hợp với các thiết bị có tài nguyên hạn chế như điện thoại di động hoặc thiết bị nhúng. Khối IRB thường được sử dụng trong phần lõi của mạng MobileNet, nơi nó tạo ra hiệu năng tốt và kích thước mô hình nhỏ gọn.



***Kiến trúc IRB***

Một IRB có cấu trúc như sau:

o Bottleneck layer: Đầu tiên, đầu vào của IRB đi qua một lớp tích chập 1x1 với số lượng kênh (channels) giảm đi. Lớp tích chập này thường được gọi là bottleneck layer vì nó giúp giảm kích thước đặc trưng (feature map) và số lượng tham số. Mục đích của việc giảm số lượng kênh là để giảm tính toán và sử dụng tài nguyên hiệu quả hơn.

o Depthwise convolution: Sau đó, đầu ra của bottleneck layer đi qua một lớp tích chập 3x3 được gọi là depthwise convolution. Lớp tích chập này áp dụng một bộ lọc riêng lẻ cho từng kênh đặc trưng riêng biệt, mà không kết hợp thông tin giữa các kênh. Điều này giúp giảm tính toán so với tích chập thông thường, trong đó tất cả các kênh được kết hợp lại.

o Pointwise convolution: Tiếp theo, đầu ra của depthwise convolution đi qua một lớp tích chập 1x1 gọi là pointwise convolution. Lớp tích chập này giúp tăng lại số lượng kênh ban đầu, đồng thời cho phép mô hình học các mối tương quan phức tạp giữa các đặc trưng.

o Kết nối shortcut: Cuối cùng, IRB có thêm một kết nối shortcut để truyền thông tin từ đầu vào trực tiếp đến đầu ra. Thông thường, kết nối shortcut được thực hiện thông qua phép cộng hoặc các phép toán pooling. Mục đích của kết nối shortcut là tạo ra đường truyền ngắn cho thông tin ban đầu, giúp mô hình học nhanh hơn và có khả năng học các biến đổi phức tạp.

Tổng quát, IRB là một khối xây dựng được sử dụng trong mạng CNN để giảm số lượng tham số và tính toán, nhằm tối ưu hóa việc triển khai trên các thiết bị có tài nguyên hạn chế. Nó sử dụng bottleneck layer để giảm số lượng kênh, depthwise convolution để giảm tính toán, pointwise convolution để tăng lại số lượng kênh, và kết nối shortcut để tăng tốc độ học và khả năng học các biến đổi phức tạp.

4.2.2.3) So sánh Pooling và Inverted Residual Block

Mục tiêu chính:

* Các phép pooling: Mục tiêu chính của các phép pooling là giảm kích thước không gian của đầu vào thông qua việc lấy giá trị tối đa hoặc trung bình trong mỗi vùng. Điều này giúp giảm số lượng tham số, tính toán và tạo tính bất biến với dịch chuyển vị trí đối tượng trong ảnh.
* Inverted Residual Block: Mục tiêu chính của Inverted Residual Block là xây dựng kiến trúc mạng sâu với số lượng tham số ít hơn, nhưng vẫn duy trì độ chính xác và khả năng học đặc trưng phức tạp.

Tác động đến features:

* Các phép pooling: Trong quá trình pooling, một phần thông tin chi tiết có thể bị mất đi do việc lấy giá trị trung bình(Average Pooling) hoặc tối đa(Max Pooling). Điều này có thể làm giảm khả năng của mô hình trong việc phân loại các đối tượng có sự biến đổi nhỏ trong hình dạng hoặc kích thước.
* Inverted Residual Block: Inverted Residual Block cố gắng giảm số lượng tham số trong mô hình, nhưng vẫn giữ lại thông tin quan trọng bằng cách sử dụng các kênh tương ứng và các phép tích chập 1x1 trước và sau các phép tích chập 3x3.

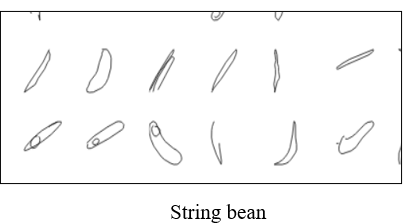
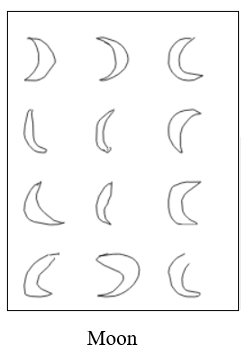
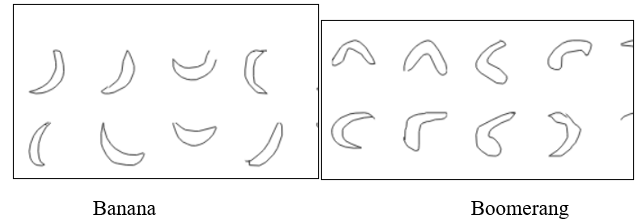
Khả năng học feature phức tạp:

* Các phép pooling: Các phép pooling chỉ tập trung vào việc giảm kích thước không gian và không thực hiện bất kỳ phép biến đổi hay học các đặc trưng phức tạp. Điều này có thể hạn chế khả năng của mô hình trong việc nhận dạng các đối tượng phức tạp và các tương quan không gian phức tạp hơn.
* Inverted Residual Block: Inverted Residual Block sử dụng các phép tích chập 1x1 và 3x3 để học các đặc trưng phức tạp và tạo ra khả năng học biểu diễn tốt hơn. Nó có khả năng học các mẫu phức tạp và tương quan không gian giữa các đặc trưng.

Tính toán:

* Các phép pooling: Các phép pooling có tính toán đơn giản và hiệu quả, giúp giảm tải tính toán cho mô hình. Điều này đặc biệt hữu ích khi làm việc với ảnh có độ phân giải cao hoặc khi tài nguyên tính toán bị hạn chế.
* Inverted Residual Block: Inverted Residual Block cũng có tính toán hiệu quả, nhưng có thể yêu cầu nhiều tham số hơn so với các phép pooling truyền thống. Tuy nhiên, việc tối ưu hóa và kết hợp với các kỹ thuật như **Depthwise Separable Convolution** có thể giúp giảm số lượng tham số và tăng tính hiệu quả tính toán.

Sau khi so sánh hai phương pháp downsampling qua từng khía cạnh ở trên, chúng ta có thể thấy Inverted Residual Block mang lại nhiều thuận lợi hơn. Cùng điểm lại vấn đề chưa được giải quyết ở chương 2.



Trên đây là những bức hình có hình dạng gần như là giống nhau, mặc dù nó nằm ở những lớp vật thể khác nhau. Để giải quyết được vấn đề nan giải này, sau khi đọc qua các phần so sánh giữa Pooling và Inverted Residual Block, chúng ta nhận thấy cả hai phương pháp đều là “giảm”:

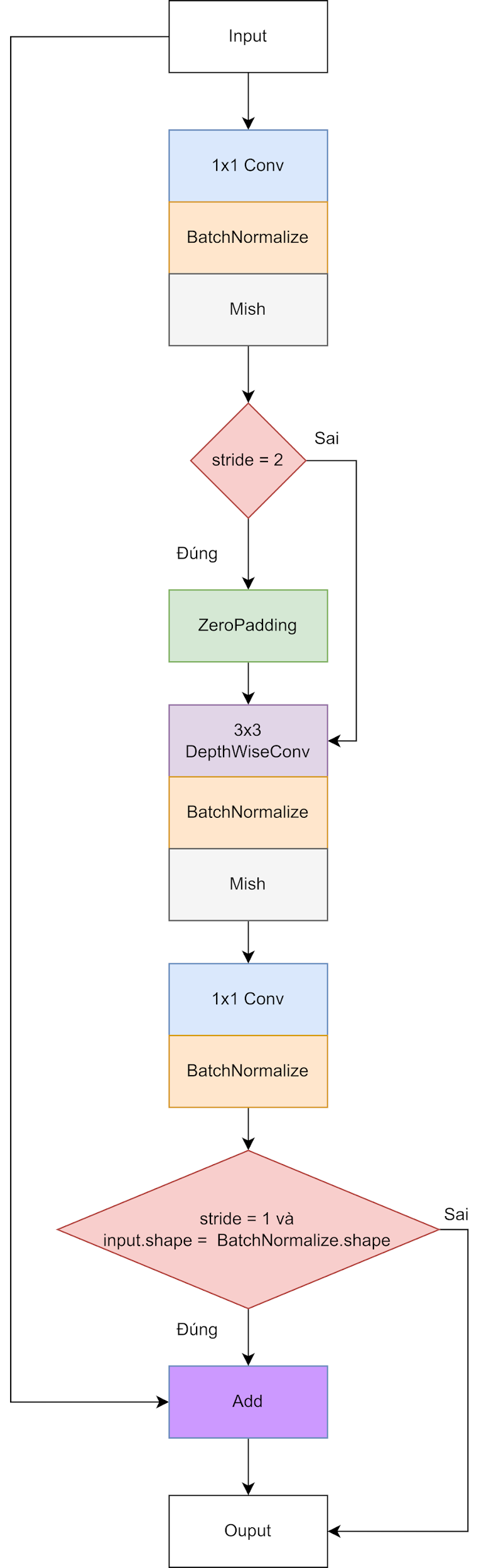
* Hiệu suất và tài nguyên tính toán: Inverted Residual Block nó đảm bảo được hiệu suất tốt, giảm thiểu chi phí và tài nguyên tốt hơn so với Pooling
* Trích xuất đặc trưng: Inverted Residual Block tận dụng được khả năng học qua nhiều lớp tăng giảm rồi lại tăng điều này đem lại khả năng giữ lại được những đặc trưng của bức ảnh, tránh mất mát đi lượng thông tin quan trọng. Đồng thời, đặc trưng được tổng quát hoá tốt hơn so với Pooling.

4.2.3) Kết luận

Nhằm xây dựng và hướng đến hệ thống nhận dạng đạt được các tiêu chí trên. Nhóm quyết định lựa chọn Inverted Residual Block với vai trò như bộ phận Downsampling trong kiến trúc mô hình.

4.3) Kiến trúc Inverted Residual Block

Trong phần này nhóm, sẽ giới thiệu bản thiết kế kiến trúc Inverted Residual Block cho mô hình của đề tài Sketch Search.

Depth-wise2D và Convolution2D là hai lớp tích chập để trích xuất đặc trưng. Tuy nhiên, sự khác nhau chính giữa hai lớp này là:

+ Convolution2D: Mỗi bộ lọc tích chập với tất cả các kênh của đầu vào.

+Depth-wise2D: Mỗi bộ lọc tích chập trên mỗi kênh riêng lẻ của đầu vào.

Lớp Zero Padding trong mạng neural được sử dụng để thêm các giá trị zero vào xung quanh dữ liệu đầu vào.Mục đích chính của việc thêm Zero Padding trước Depth-wise2D với bước nhảy là 2 dùng cho việc duy trì kích thước và thông tin không gian của đầu vào khi áp dụng phép tích chập, đồng thời tránh mất mát thông tin ,điều chỉnh kích thước và tăng cường trích xuất đặc trưng.

Linear Pointwise Convolution: Sử dụng Convolutional Layer 1x1 để kết hợp thông tin từ các kênh đã được tích chập riêng lẻ từ Depth-wise2D và tăng kích thước feature map trở lại.

Nhóm sử dụng hàm Mish vì:

+ Hàm kích hoạt Mish tính được đạo hàm tại mọi điểm trong miền giá trị của nó, có tính khả vi (smoothness)và không đơn điệu. Nó cũng là một hàm liên tục và tự điều chỉnh

+ Hàm kích hoạt Mish giới hạn giá trị đầu vào trong khoảng [-inf, +inf], giúp tránh tình trạng bão hòa và duy trì gradient lớn hơn, từ đó tránh tình trạng Gradient dying (hiện tượng gradient giảm mạnh trong quá trình lan truyền ngược, gây khó khăn trong việc cập nhật trọng số và làm mạng học chậm hoặc không hiệu quả).

Tuy nhiên chi phí tính toán của hàm mish khá cao so với các hàm kích hoạt khác.

4.4) Dropout

SpatialDropout2D và Dropoutlà kỹ thuật regularization (chống quá khích) trong mạng nơ-ron giúp tránh hiện tượng over fitting.  
 + Dropout: Loại bỏ ngẫu nhiên một phần các đơn vị đầu vào của một lớp.   
 + SpatialDropout2D: Loại bỏ ngẫu nhiên toàn bộ các kênh đầu vào của một lớp 2D (thường là Convolutional layer) trong quá trình huấn luyện. Sử dụng trong mạng CNN để xử lý tính chất không gian và tạo biểu diễn không gian của dữ liệu đa kênh.

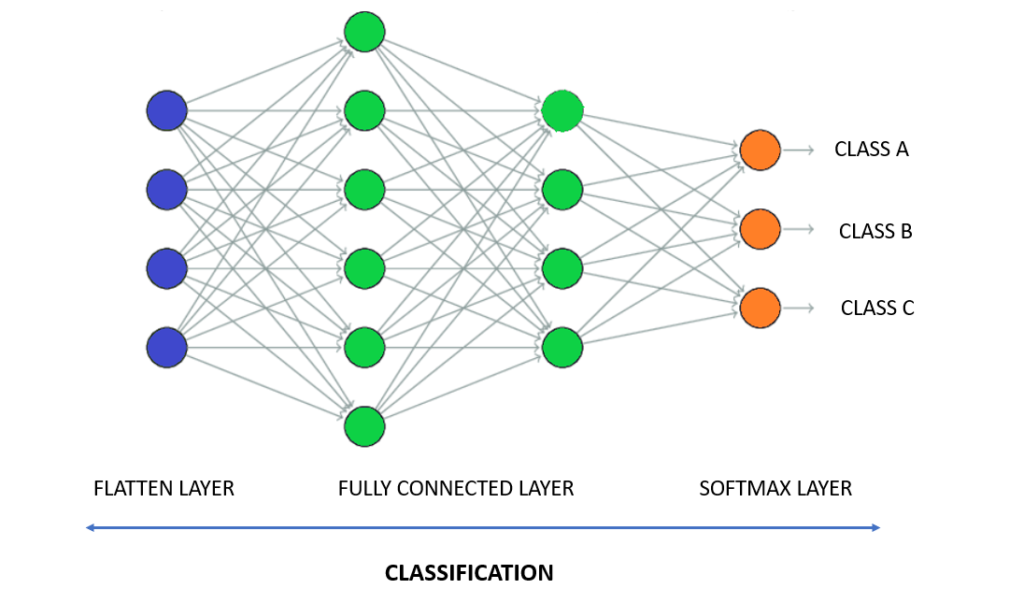
Trong đề tài lần này nhóm sử dụng lớp Spatial Dropout 2D thay vì sử dụng lớp Dropout thông thường hay còn gọi là Dropout 1D. Với một số lý do như sau:

* SpatialDropout2D và Dropout là hai phương pháp dropout thường được sử dụng trong mạng neural để tránh overfitting.
* Phương pháp dropout truyền thống, hay còn gọi là Dropout 1D, được áp dụng bằng cách tạm thời loại bỏ một số lượng ngẫu nhiên các đơn vị đầu vào trong quá trình huấn luyện. Cụ thể, mỗi đơn vị đầu vào (neuron) có một xác suất xác định bị tắt, và đầu ra của mô hình được tính toán dựa trên một tập dữ liệu đã bỏ qua một số đơn vị bị tắt này. Điều này giúp mô hình học được tính đa dạng và tránh việc phụ thuộc quá mức vào một số đặc trưng cụ thể.
* Tuy nhiên, trong các mạng neural sử dụng thông tin không gian như ảnh, các đơn vị đầu vào thường được tổ chức thành các cấu trúc 2D như ma trận. SpatialDropout2D được thiết kế đặc biệt cho việc này. Thay vì tắt các đơn vị đầu vào riêng lẻ như dropout thông thường, Spatial Dropout 2D tắt toàn bộ các kênh (channels) của ma trận đầu vào. Cụ thể, mỗi kênh của ma trận đầu vào có một xác suất tắt, và tất cả các giá trị trong kênh đó được đặt thành 0. Điều này giúp mô hình học được tính đa dạng không chỉ trong từng đơn vị riêng lẻ mà còn giữa các kênh.

Tóm lại, điểm khác nhau chính giữa Spatial Dropout 2D và dropout thông thường là Spatial Dropout 2D tắt các kênh (channels) của ma trận đầu vào, trong khi Dropout thông thường tắt các đơn vị đầu vào riêng lẻ. SpatialDropout2D thường được sử dụng trong các mô hình mạng neural sử dụng thông tin không gian như mạng neural tích chập (CNN).

4.5) GlobalAveragePooling

Global Average Pooling là phương pháp được giới thiệu trong bài báo Network in Network được nhóm tác giả đến từ trường đại học Singapore đề xuất năm 2014. Trước đó, khi xây dựng mạng CNN, sau khi xây dựng xong phần Feature Extraction, ta thường nghĩ đến việc xây dựng các lớp fully-connected để phục vụ cho việc phân loại.



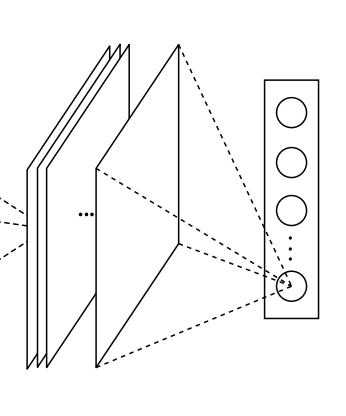
*Hình minh hoạ cho fully-connected-layers*

Nhìn qua, ta dễ dàng nhận ra các lớp cấu tạo nên phần Classification gồm:

* Flatten ⇔ Input
* Hidden layers
* Output layer

Ở đây, nhóm lựa chọn Global Average Pooling thay sử dụng cho các lớp flatten layer và fully-connected layer của mạng nơ-ron truyền thống.

Global Average Pooling sẽ tóm gọn tất cả đặc trưng trên ma trận đặc trưng(feature map) thành một giá trị trung bình mới, mỗi giá trị trung bình này sẽ được vectơ hoá để chuyển sang lớp output chứa hàm kích hoạt softmax để tính phân phối xác suất giữa các lớp trong bài toán.

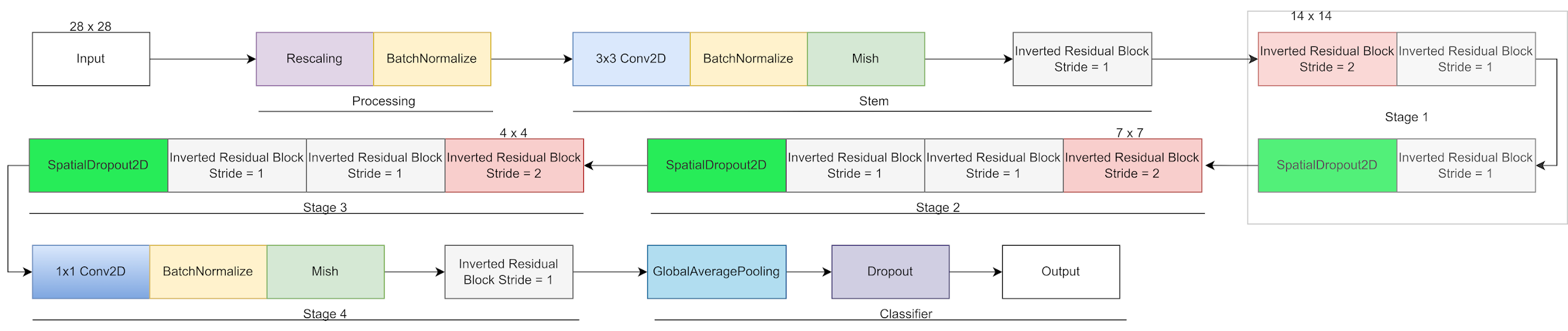


*Minh hoạ của Global Average Pooling*

Những điểm mạnh của Global Average Pooling:

* Quá trình tính toán nhanh dựa vào các ma trận đặc trưng để tính giá trị trung bình cho tất cả đặc trưng thuộc ma trận đặc trưng.
* Không sử dụng bộ tham số nên sẽ không cần tối ưu, tránh được sự overfitting của mô hình.
* Gia tăng thời gian huấn luyện và tối ưu mô hình, giảm thiểu tài nguyên lưu trữ vì không sử dụng bộ tham số
* Tổng quát được tất cả đặc trưng của hình ảnh, khái quát được thông tin đặc trưng của hình ảnh.
* Lớp flatten là lớp vecto hoá, nghĩa là có bao nhiêu đặc trưng sẽ được chuyển thành dưới dạng vectơ toàn bộ. Điều này không đảm bảo được tính tổng quát của dữ liệu.
* Tốn thời gian và kiểm thử trong việc thiết kế các lớp Hidden layer. Nếu thiết kế không hợp lý sẽ dẫn đến mất mát những thông tin đặc trưng quan trọng của hình ảnh, gây mất nhiều thời gian huấn luyện và tài nguyên sẵn có.

4.6) Kiến trúc mô hình



Mô hình được thống nhất thành 7 phần chính:

* Processing
* Stem
* Stage 1
* Stage 2
* Stage 3
* Stage 4
* Classification

Sau khi một loạt so sánh giữa ưu điểm và nhược điểm , nhóm đã chọn ra các lớp đã được đề cập ở phần trước. Có thể thấy, ở stage 1, 2 và 3 cuối mỗi stage đều có lớp Spatial Dropout 2D lớp này nhằm tắt đi ngẫu nhiên các giá trị kênh ở mỗi đặc trưng nhằm đưa lọc đi những đặc trưng gây nhiễu và thừa. Đồng thời trong lúc quá trình tối ưu tránh việc overfitting của mô hình.

Ở phần Classification, thay vì xây dựng các lớp fully-connected truyền thống của CNN, nhóm đã có sự điều chỉnh bằng Global Average Pooling, sau đó đi kèm 1 lớp Dropout nhằm tắt đi ngẫu nhiên 1 vài giá trị hay nơ-rin tránh việc overfitting khi mô hình học đi học lại nhiều lần.

Chương 5: HUẤN LUYỆN MÔ HÌNH

5.1) Bộ dữ liệu

Tập dataset gồm 345 lớp và các bức ảnh đều có thuộc tính **recognized = True**:  
 + Tập train: Mỗi lớp có 3200 bức ảnh-> tổng dataset để train là 1104000.  
 + Tập validation: Mỗi lớp có 800 bức ảnh-> tổng dataset để kiểm thử là 276000.  
 + Tập test: Mỗi lớp có 14000 bức ảnh-> tổng dataset để kiểm thử là 4830000.  
Các tập dữ liệu này không có sự trùng lặp về dữ liệu.

5.2) Độ đo và hàm tối ưu

5.2.1) Đo lường sự mất mát

Sử dụng Categorical Cross-entropy loss đo lường sai số giữa phân phối xác suất dự đoán của mô hình và phân phối xác suất thực tế của các lớp. Mục tiêu là tối thiểu hóa giá trị của hàm mất mát này để cải thiện hiệu suất phân loại đa lớp của mô hình.  
Dùng hàm loss này vì nhóm muốn sử dụng một kỹ thuật đó là Label smoothing là một kỹ thuật regularization giúp làm mờ phân phối xác suất nhãn để cải thiện khả năng tổng quát hóa của mô hình thay vì one-hot encoding cho nhãn và tránh overfitting .Điều này, giúp làm mô hình chủ động hơn trong việc khám phá các đặc trưng  
VD: Giả sử chúng ta có một bài toán phân loại hình ảnh gồm 3 lớp: mèo, chó và chim.  
 + Sử dụng label smoothing:Một bức ảnh của mèo được gán nhãn [1, 0, 0].  
 + Sử dụng label smoothing với epsilon = 0.1:Một bức ảnh của mèo được gán nhãn [0.9, 0.05, 0.05].  
Thông số được sử dụng cho label smoothing = 0.15

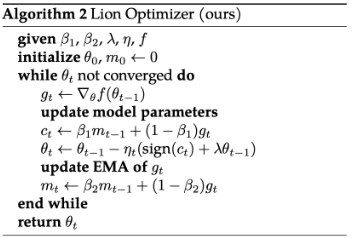
5.2.2) Đo lường kết quả

Các độ đo dưới đây được sử dụng trong quá trình huấn luyện và đánh giá mô hình trên tập validation. Một số độ đo như sau:

+ Top 1 Accuracy: Tỷ lệ dự đoán chính xác lớp cao nhất so với nhãn đúng.  
 + Top 5 Accuracy: Tỷ lệ dự đoán chính xác lớp cao nhất trong top 5 dự đoán so với nhãn đúng.  
 + Categorical Cross entropy: Đo lường độ tương đồng giữa phân phối xác suất dự đoán và phân phối xác suất thực tế bằng cách tính khoảng cách giữa hai phân phối.  
 + mAP@5 (mean Average Precision at 5): Đo lường trung bình độ chính xác trong việc xếp hạng các dự đoán theo thứ tự đúng trong top 5.  
 + NDCG@5 (Normalized Discounted Cumulative Gain at 5): Đo lường chất lượng xếp hạng của mô hình dựa trên sự ưu tiên đúng trong top 5 dự đoán.  
- Hàm callback đều dùng để theo dõi đối tượng “val\_ mAP@5 “:  
 + ReduceLROnPlateau: Giảm tỷ lệ học khi mô hình không cải thiện trong một số vòng lặp liên tiếp.  
 + ModelCheckpoint: Lưu lại phiên bản tốt nhất của mô hình dựa trên một chỉ số đánh giá sau mỗi vòng lặp.  
 + EarlyStopping: Dừng quá trình huấn luyện khi mô hình không cải thiện trong một số vòng lặp liên tiếp.

5.2.3) Hàm tối ưu

Hàm optimizer (hàm tối ưu): EvoLved Sign Momentum (Lion) đây là một hàm tối ưu mới và được phát triển Google Brain trên các thực nghiệm của bài báo và các người dùng thực nghiệm thì hàm tối ưu này đang tốt hơn so với các hàm tối ưu khác khi thực hiện cùng tác vụ trên cùng một mô hình và cùng số vòng lặp.



5.3) Hiệu chỉnh siêu tham số

Sử dụng thư viện Keras Tuner là một thư viện mã nguồn mở được sử dụng để tìm kiếm và tinh chỉnh tự động các siêu tham số cho mô hình học máy sử dụng thư viện Keras. Nó giúp tìm ra các siêu tham số tốt nhất để cải thiện hiệu suất của mô hình một cách tự động và tiết kiệm thời gian.

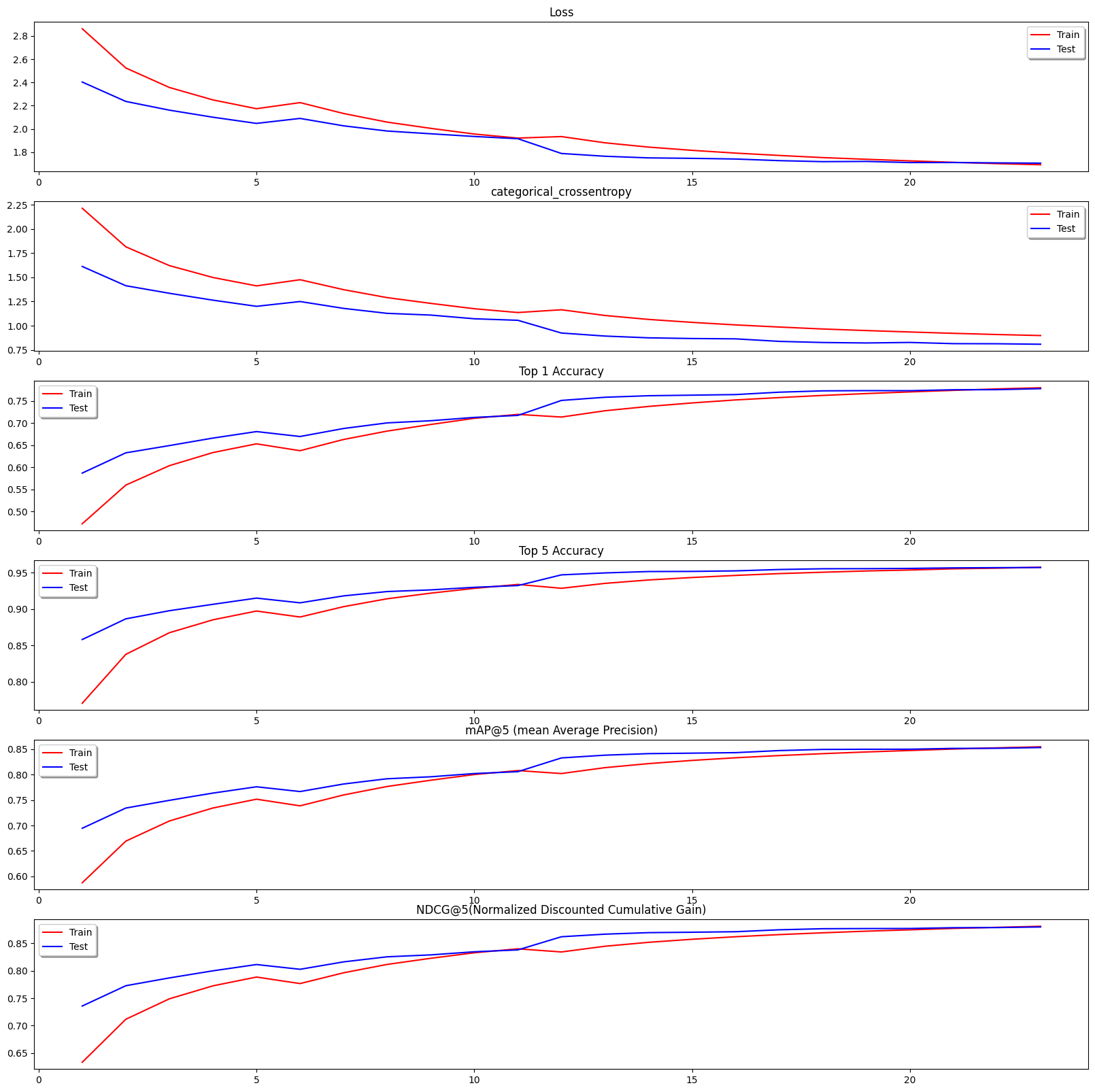
Chỉ có ràng buộc số lượng các filter của các Inverted Residual Blocks trong các stage phải giống nhau.

Bằng việc sử dụng mô hình ở trên và tập dataset dùng để tìm ra mô hình tối ưu và kết hợp với Keras Tuner đã giúp chọn được các hyperparameter cho bài toán.

5.4) Môi trường huấn luyện

Nhóm sử dụng Google Colab - một môi trường thí nghiệm dành cho các nhà nghiên cứu được xây dựng dựa trên nền tảng web cung cấp các tài nguyên như CPU, RAM, GPU và TPU - tổ chức và xây dựng từ dữ liệu cho đến mô hình và kiểm thử đánh giá trên Google Colab.

5.5) Kết quả huấn luyện



5.6) Kết quả trên tập test

|  |  |  |
| --- | --- | --- |
|  | Model -nhóm tự xây dựng | Model dựa vào VGGNet |
| Số lượng parameter | 3,351,005 | 6,661,989 |
| Loss | 0.7186 | 0.7223 |
| Accuracy | 80.63% | 79.63% |
| Top 3 accuracy | 94.32% | 93.44% |
| Top 5 accuracy | 96.70% | 95.96% |
| mAP@3 | 86.93% | 85.98% |
| mAP@5 | 87.48% | 86.56% |
| NDCG@3 | 88.84% | 87.91% |
| NDCG@5 | 89.82% | 88.95% |

**Nhận xét:**

Mô hình tự xây dựng có lượng tham số ít hơn so với model dựa VGGNET. Tuy có lượng tham số ít hơn nhưng mô hình vẫn đạt được hiệu suất đáng ngờ. Chúng ta dễ dàng thấy sự tốt hơn ở tất cả các độ đo mà nhóm sử dụng để đánh giá hệ thống. Tuy không có sự chênh lệch quá lớn giữa các thông số 2 mô hình nhưng nó vẫn chứng minh được hướng đi của nhóm trong việc xây dựng chọn phương pháp là phù hợp với đề tài.

Chương 6: ỨNG DỤNG WEB

6.1) Kiến trúc ứng dụng

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu tượng

Mô tả được tạo tự động

6.2) Công nghệ sử dụng

6.1.1) FastAPI

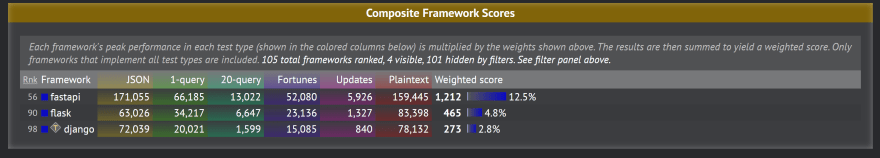


FastApi là 1 web framework dùng để build API có hiệu năng cao, code đơn giản nhưng cũng hỗ trợ tốt môi trường production.

FastAPI được phát từ năm 2018 bởi Sebastián Ramírez, bởi vì bản thân Ramírez cảm thấy khó chịu với việc sử dụng những framework hiện có như Flask và DRF nên đã tự tạo ra một khung công cụ cho riêng cho mình dựa trên Starlette và Pydantic. Và giờ đây những công ty lớn như Uber, Netflix, thậm chí là Microsoft, … đều đang sử dụng FastAPI cho việc xây dựng các app của họ.

Lợi ích của FastAPI:

1. Cung cấp hiệu suất cao sánh ngang với Node.js, GO và tự tin rằng là framework nhanh nhất của Python

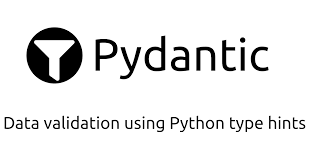


1. Hỗ trợ Async: async là một vấn đề nhức nhối tạo nên sự khác biệt giữa các trang web dùng Python với các ngôn ngữ còn lại, do async đã được hỗ trợ tốt trên các ngôn ngữ khác như javascript, java, php, …

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

1. Hỗ trợ Validation data: Pydantic là một thư viện phổ biến của Python cho việc kiểm thử dữ liệu đầu vào, và nhiều loại xác thực khác.



1. Hỗ trợ sinh Documment tự động: với mỗi API được viết bằng PastAPI sẽ được tự động cập nhập vào document, thuận tiện cho việc tra cứu, sử dụng của 2 bên front-end và back-end

6.1.2) P5.js

p5.js giúp người dùng có thể tương tác trực tiếp với website. Giống như một "cuốn sổ phác họa", p5.js có đầy đủ các bộ chức năng phục vụ cho việc vẽ vời animation trên website.

Trong ứng dụng web, p5.js được sử dụng để người dùng có thể thực hiện các thao tác vẽ, xóa đường vẽ trên khung vẽ.

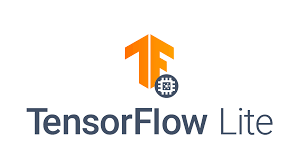
6.1.3) Char.js

Ảnh có chứa Đồ họa, trái tim, thiết kế, sáng tạo

Mô tả được tạo tự động

Chart.js là một dự án mã nguồn mở giúp cho mọi người có thể vẽ những biểu đồ thể hiện số liệu trên website một cách dễ dàng và đẹp nhất. Điểm mạnh nhất của Chart.js là: Responsive tự động điều chỉnh kích thước để có thể hiển thị đẹp nhất trên tất cả các thiết bị từ Desktop, Tablet, Mobile.

6.1.4) TensorFlow Lite



TensorFlow Lite là một phiên bản nhỏ gọn của TensorFlow cho mobile và thiết bị nhúng.

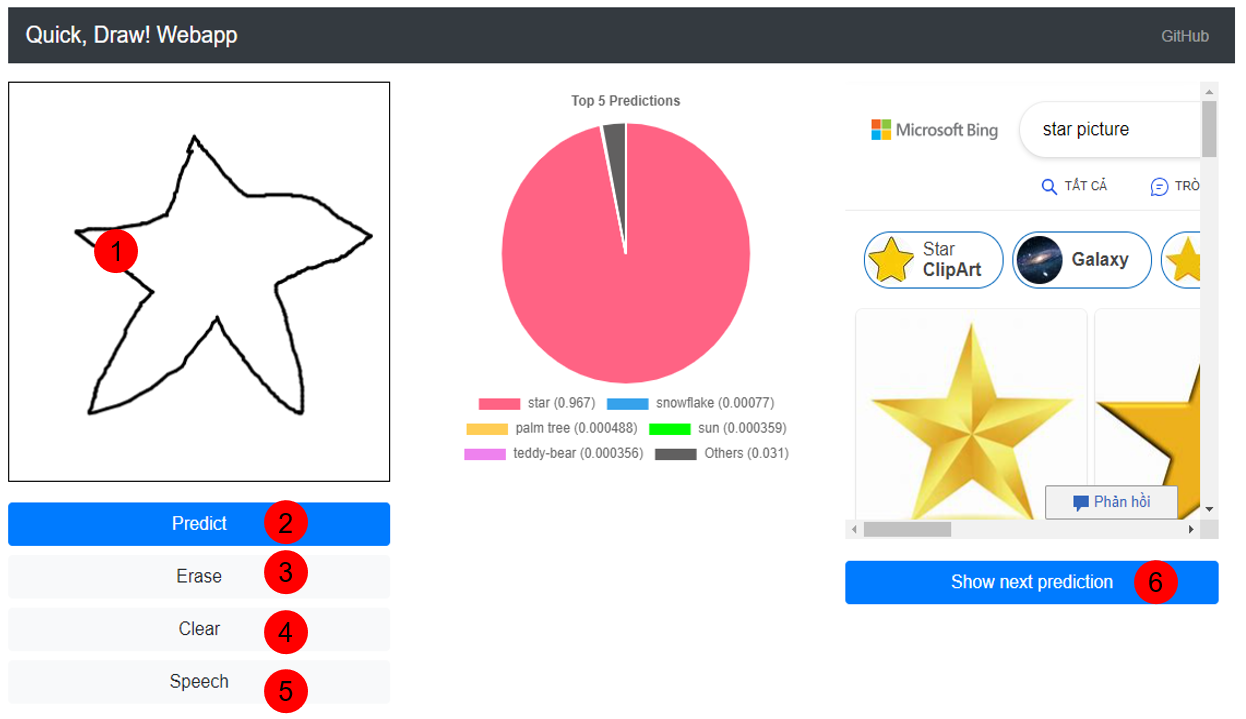
Lợi ích khi sử dụng TensorFlow Lite:

- Nhanh hơn, do TensorFlow Lite cho phép thực hiện machine. learning ngay trên device với độ trễ thấp.

- TensorFlow Lite tốn ít dung lượng nên khá tốt cho mobile, web.

- Tensorflow Lite hỗ trợ hardware acceleration.

6.3) Chức năng của web

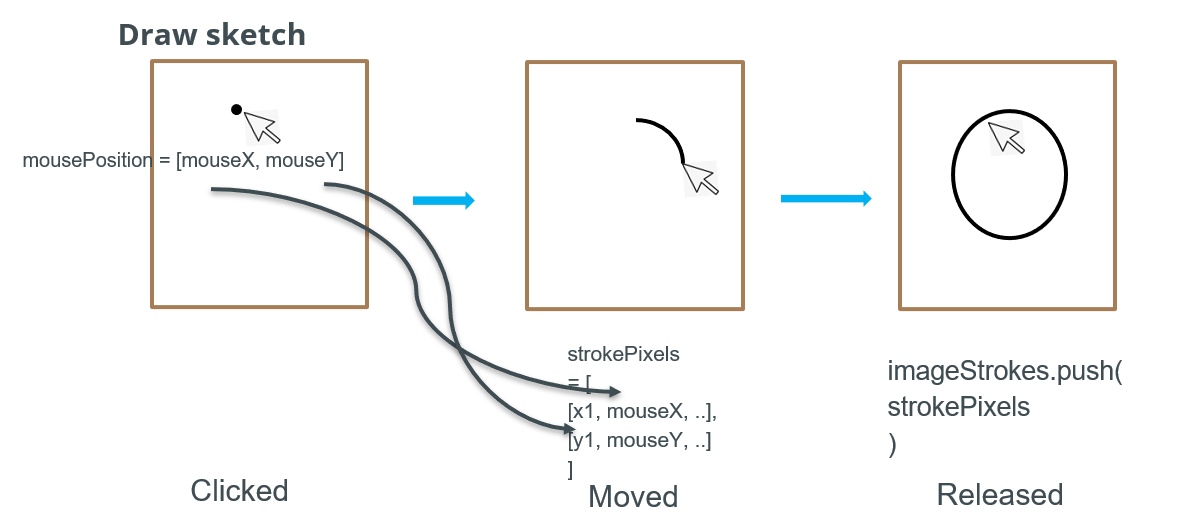


**Các chức năng của web:**

1. Click và giữ chuột phải để vẽ hình trong khung
2. Dự đoán ảnh đã vẽ
3. Xóa 1 nét vừa vẽ
4. Xóa toàn bộ khung
5. Đọc các kết quả dự đoán
6. Hiển thị kết quả tìm kiếm tiếp theo

6.4) Quá trình xử lý

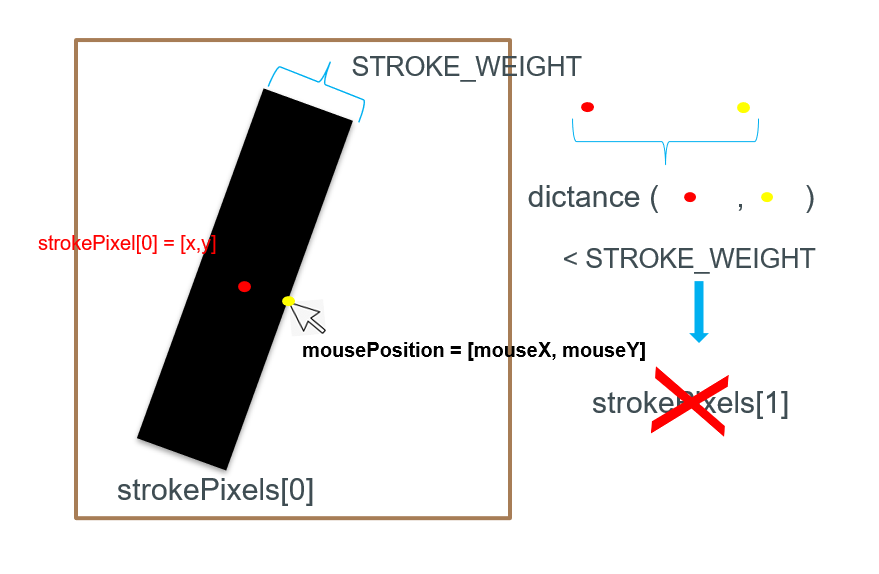
6.4.1) Draw sketch



Khi người dùng di chuyển con trỏ chuột vào trong khung vẽ (với chiều dài, và chiều rộng khung vẽ đã được thiết lập từ trước), trình duyệt sẽ bắt 3 sự kiện xảy ra từ con trỏ:

1. Clicked: Khi người dùng clicked chuột trái để bắt đầu vẽ, tọa độ của con trỏ được lưu lại vào biến mousePosition
2. Moved: Người dùng đã nhấp chuột trái và bắt đầu di chuyển con trỏ để vẽ. Liên tục giá trị của biến mousePosition sẽ được lưu lại vào mảng strokePixels và cập nhật theo vị trí con trỏ cho đến khi người dùng dừng vẽ
3. Released: sự kiện released xảy ra khi sau khi moved, biểu hiện cho biết người dùng đã dừng nét vẽ, lúc này mảng strokePixels sẽ được thêm vào mảng imageStrokes (mảng dùng để lưu lại các nét vẽ theo thứ tự từ lúc bắt đầu vẽ cho đến khi được dùng để dự đoán hình vẽ)

6.4.2) Erase stroke



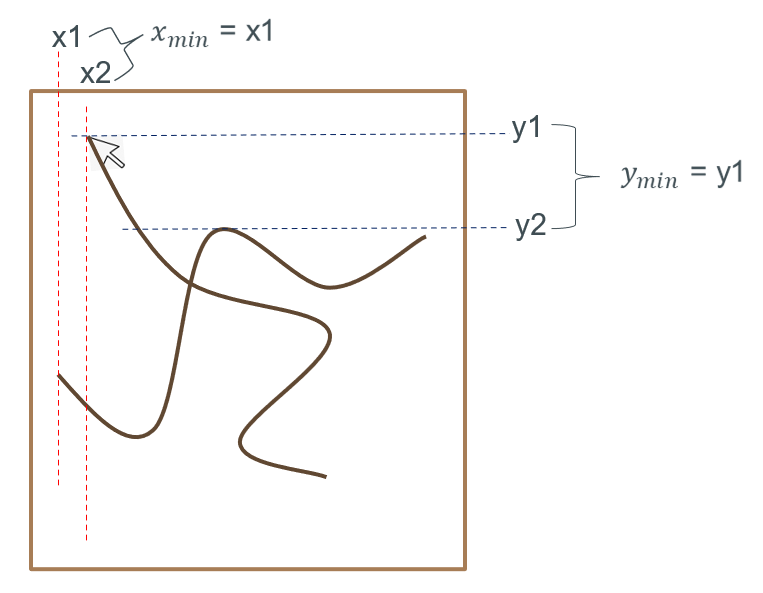
Erase stroke được kích hoạt khi clicked vào nút Erase trên màn hình.

Lúc này, mỗi sự kiện clicked, hoặc moved của con trỏ chuột đều được xem xét cho erase troke. Chúng ta sẽ tính khoảng cách giữa 2 tọa độ: con trỏ chuột và từng cặp pixel của mỗi strokePixels trong mảng imageStrokes, nếu khoản cách này nhỏ hơn giá trị STROKE\_WEIGHT (độ lớn của mỗi nét vẽ trong ảnh), nghĩa là đây là nét vẽ người dùng muốn xóa, ta sẽ bỏ strokePixels tương ứng của cặp pixel đang xét ra khỏi chuỗi imageStrokes.

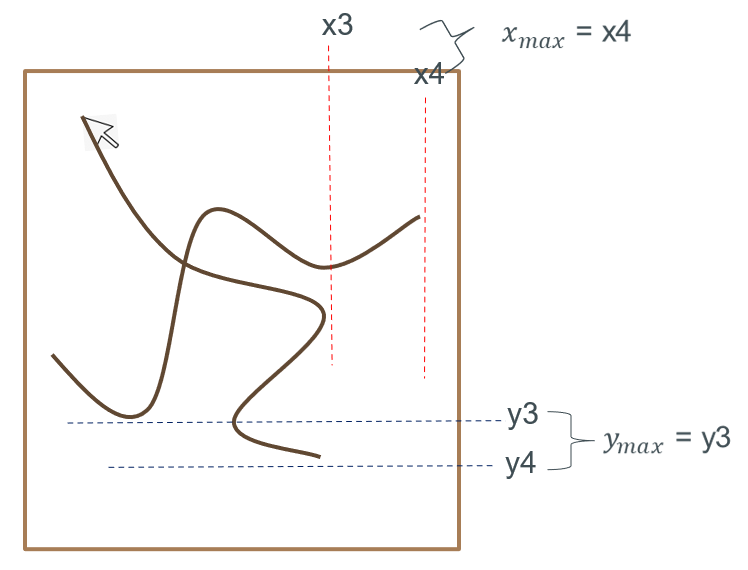
6.4.3) Get bounding box

Trong quá trình thử nghiệm, nhận thấy nếu sử dụng kích thước ban đầu của khung vẽ là sau đó resize về kích thước 28 dẫn đến độ chính xác của mô hình không cao, do đó cần thêm một bước xử lý là thu nhỏ bounding box để độ chính xác của hình vẽ trong ảnh được phóng to hơn, nhằm mục đích tăng độ chính xác của mô hình.

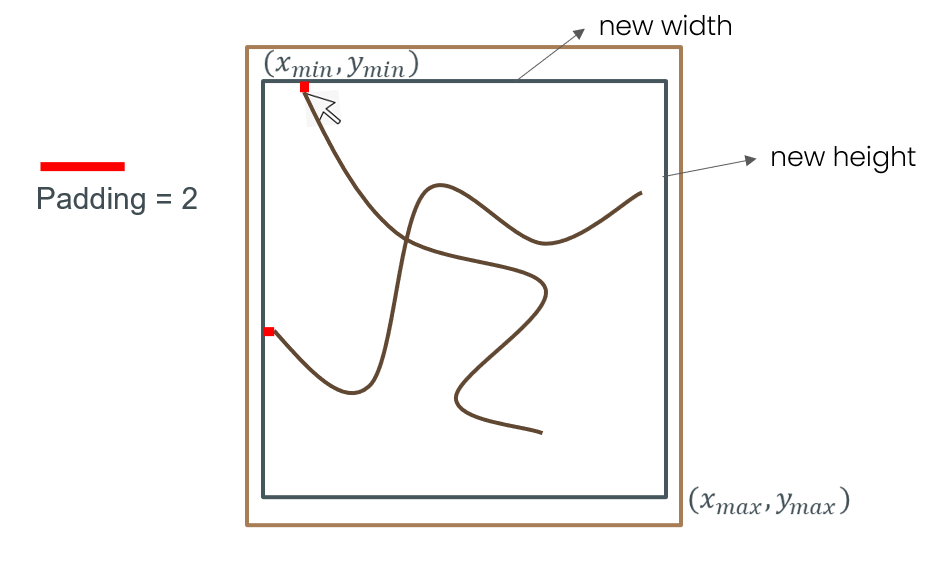
Đầu tiên ta sẽ xác định tọa độ trên cùng bên trái của các nét vẽ trong ảnh:



Tại mỗi strokePixels, đi tìm giá trị tọa độ nhỏ nhất theo hệ tọa độ x và y trong tất cả các cặp pixel, sau đó lấy tọa độ x và y nhỏ nhất trong tất cả các strokePixels của mảng imageStrokes sẽ thu được giá trị cho vị trí trên cùng bên trái



Tương tự tại mỗi strokePixels, đi tìm giá trị tọa độ lớn nhất theo hệ tọa độ x và y trong tất cả các cặp pixel, sau đó lấy tọa độ x và y lớn nhất trong tất cả các strokePixels của mảng imageStrokes sẽ thu được giá trị .



Bounding box sẽ là một hình vuông với vị trí bắt đầu là giá trị trên cùng bên trái đã tìm ở trên, độ dài cạnh hình vuông là giá trị lớn nhất giữa kết hợp với lớp padding 2 pixel.

6.4.4) Tạo ảnh

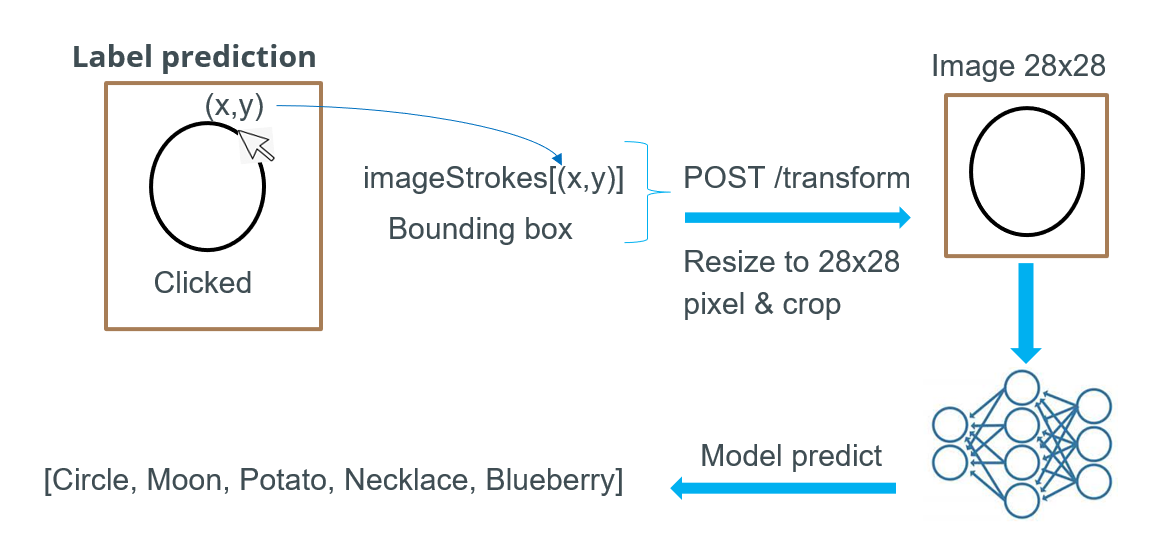


Mỗi request gửi về server đi kèm mảng imageStrokes chứa các strokes đã vẽ bởi người dùng và các giá trị bounding box đã được xử lý ở trên.

Từ đó một ảnh trắng với kích thước bằng bounding box được ra. Sau đó từng cặp pixel trong mỗi strokePixels của imageStrokes sẽ được duyệt qua, tương ứng với đó là mỗi point sẽ được vẽ trên ảnh trắng, sau khi duyệt hết tất cả các pixels, ta sẽ được ảnh người dùng đã vẽ ban đầu.

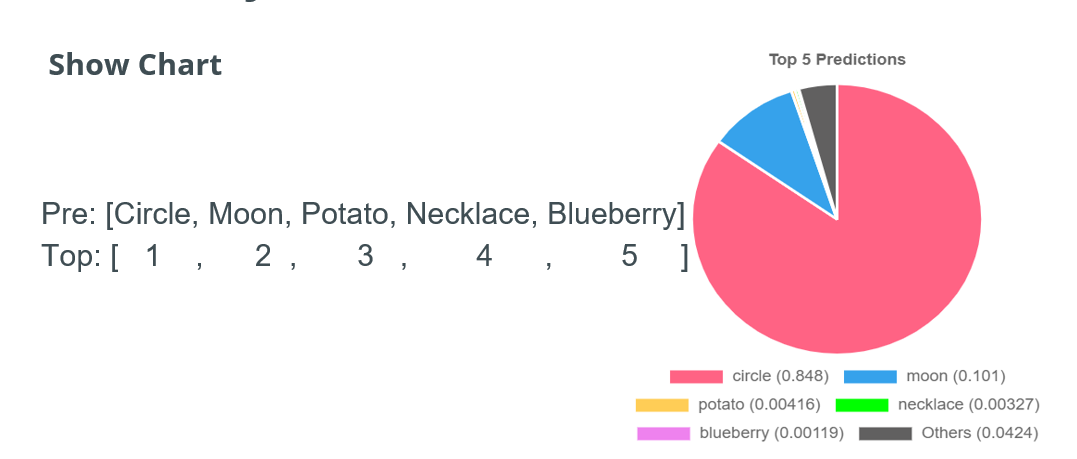
Ảnh này sẽ resize về kích thước 28 được gửi lại cho client thông qua response.

6.4.5) Predict label



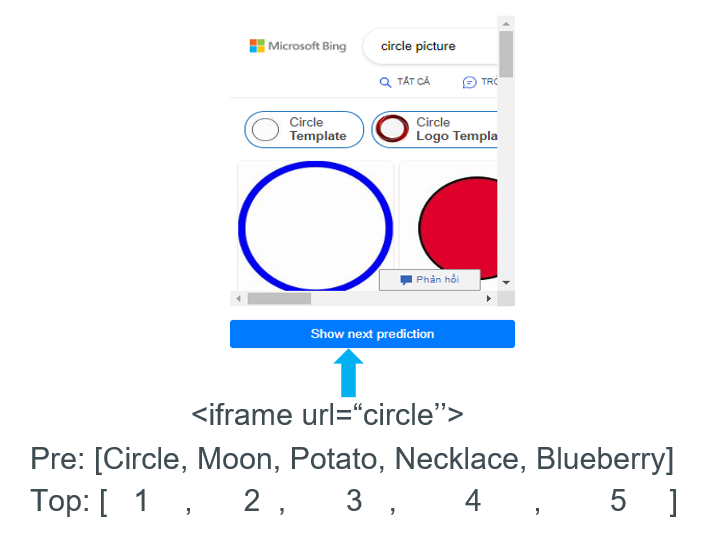
Sau khi nhận được ảnh có kích thước 28 từ sever, ảnh này được đưa vào model để predict các nhãn tương ứng với hình vẽ. Như ví dụ ở trên, đối với hình vẽ có dạng hình tròn, model sẽ cho ra 5 nhãn có giá trị gần nhất là: Circle, Moon, Potato, Necklace, Blueberry.

6.4.6) Show chart



Với 5 label được dự đoán từ model, một circle char được vẽ để dùng thể hiện với phần trăm accuracy tương ứng của mỗi label. Phần trăm lớn sẽ có chiếm nhiều diện tích trong chart.

6.4.7) Sử dụng Bing



Để trực quan hóa các label được dữ đoán, mỗi label sẽ được dùng để tìm kiếm các hình ảnh trên Bing. Tương tự như Show chart, với 5 label được dự đoán từ model, kết quả tìm kiếm trên Bing thể hiện những hình ảnh tương ứng với label. Phần trăm lớn sẽ được tìm kiếm trước và những label phía sau sẽ được tìm kiếm khi nút **Show next prediction** được clicked.

Chương 7: KẾT LUẬN

Đề tài Sketch Search là đề tài hay và mới lạ nhưng song song với đó là sự khó khăn gặp phải về bộ dữ liệu và mô hình đã được nhóm bàn luận và đưa ra hướng giải quyết ở trên. Sau khi hoàn thiện đề tài Sketch Search đã rút ra một số sai sót và kinh nghiệm khi làm những bài toán Nhận dạng vật thể. Trong báo cáo này, nhóm đã đưa ra các phương pháp trong việc xây dựng mô hình CNN sao cho phù hợp với bộ dữ liệu đang dùng. Nhóm đã đánh giá kết quả của mô hình tự xây dựng và mô hình dựa vào VGGNET trên tập dữ liệu test đã được đề cập ở chương 5. Kết quả qua các độ đo mà nhóm sử dụng cho thấy nhóm đã xây dựng mô hình đúng hướng và phù hợp với bộ dữ liệu QuickDraw. Tuy vậy, vẫn còn nhiều điểm hạn chế trong việc triển khai ứng dụng lên Server. Hi vọng thầy khi đọc đến đây sẽ thấu hiểu và bỏ qua thiếu sót của nhóm. Nhóm xin chân thành cảm ơn thầy đã tận tình hướng dẫn trong thời gian qua.

**Hướng phát triển:**

* Thêm CSS cho web, theme, animation nhằm tạo sự hứng thú cho người dùng đặc biệt là trẻ em.
* Giảm khác biệt giữa sketch và prediction trên Bing.
* Deploy lên server + phát triển app trên thiết bị mobile.
* Xây dựng cơ sở dữ liệu hình ảnh cho từng đối tượng để truy xuất hình ảnh liên quan hơn và không phụ thuộc vào tìm kiếm của trình duyệt Bing.

Github của nhóm: <https://github.com/duongve13112002/CS338.N21.KHCL>

Chương 8: Tài liệu tham khảo

* **Network In Network**, Min Lin, Qiang Chen, Shuicheng Yan Graduate School for Integrative Sciences and Engineering Department of Electronic & Computer Engineering National University of Singapore, Singapore, 2014
* **Feature Pyramid Networks for Object Detection**, Tsung-Yi Lin, Piotr Dollar ,Ross Girshick , Kaiming He, Bharath Hariharan, and Serge Belongie ,Facebook AI Research (FAIR) Cornell University and Cornell Tech, 2017
* **MobileNetV2: Inverted Residuals and Linear Bottlenecks**,Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen, Google Inc, 2019
* **Symbolic Discovery of Optimization Algorithms**, Xiangning Chen1 2 § ∗ Chen Liang1 § Da Huang1 Esteban Real1 Kaiyuan Wang1 Yao Liu1 † Hieu Pham1 Xuanyi Dong1 Thang Luong1 Cho-Jui Hsieh2 Yifeng Lu1 Quoc V. Le1, Google, 2023
* Github của thư viện Chart.js: <https://github.com/chartjs>
* Tài liệu Chart.js: <https://www.chartjs.org/>
* **VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION**, Karen Simonyan ∗ & Andrew Zisserman + Visual Geometry Group, Department of Engineering Science, University of Oxford, 2015

PHÂN CHIA CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| Họ và tên | Mã số sinh viên | Công việc |
| Nguyễn Vũ Dương | 20520465 | Làm slide, xây dựng mô hình, thu thập, xử lý dataset |
| Lê Trần Quốc Khánh | 20520574 | Làm slide,viết báo cáo, tìm hiểu dataset, xây dựng ứng dụng. |
| Huỳnh Đình Kim Điền | 20521183 | Làm slide, viết báo cáo, tìm hiểu dataset, xây dựng ứng dụng. |