

ĐẠI HỌC ĐÀ NẴNG

**TRƯỜNG ĐẠI HỌC SƯ PHẠM ĐÀ NẴNG**

**KHOA TIN HỌC**

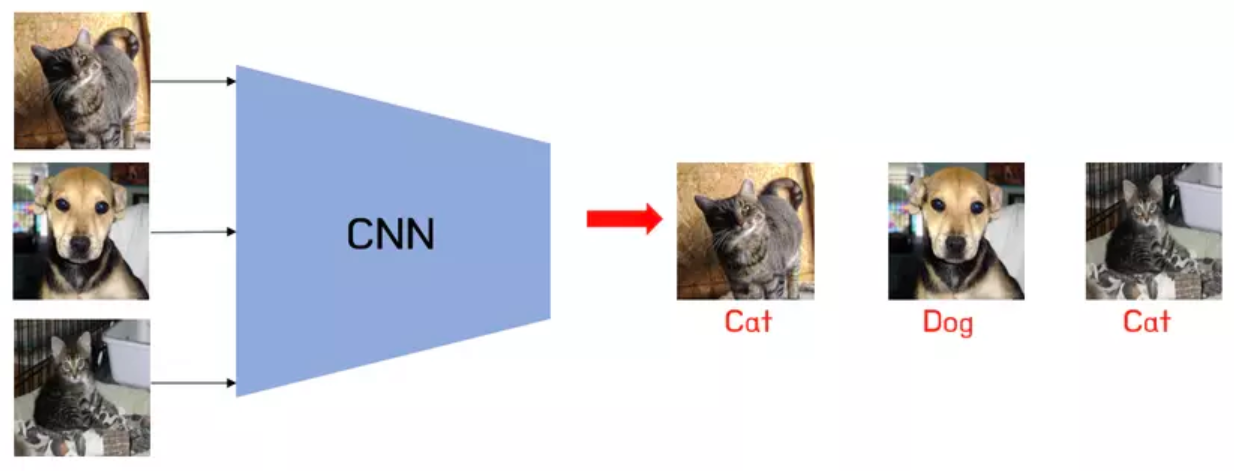


**ĐỒ ÁN CHUYÊN NGÀNH**

**ĐỀ TÀI:**

**PHÂN LOẠI ẢNH MÈO VÀ CHÓ BẰNG**

**THUẬT TOÁN CNN**

****

**Sinh viên thực hiện: Phạm Anh Kiệt**

**Lớp: 21CNTT3**

**Giảng viên hướng dẫn** : **T.s Vũ Thị Trà**

**Đà Nẵng,11/2023**

**LỜI NÓI ĐẦU**

Hiện nay, việc ứng dụng trí tuệ nhân tạo vào nhận diện sự vật quanh ta đang cực kì phổ biến và phát triển, đặc biệt là những thứ rất gần gũi và thân thuộc với cuộc sống chúng ta. Chính vì lý do đó mà tôi đã quyết định lựa chọn phát triển phần mềm nhận diện mèo và chó và phần mềm này có thể nâng cấp lên nhận diện nhiều động vật và đồ vật khác. Đây là tài liệu dùng để nhận diện mèo và chó dựa trên hình ảnh đã được cung cấp. Trong bài báo cáo này có sử dụng thuật toán CNN Keras và TensorFlow. Tài liệu này giúp ta có một cái nhìn tổng quát về việc áp dụng thuật toán vào phần mềm phân biệt mèo và chó thông qua hình ảnh đã được cung cấp trước.

Tôi xin chân thành cảm ơn giảng viên T.s Vũ Thị Trà người đã hướng dẫn, và giảng viên Nguyễn Thị Ngọc Anh đã trang bị cho tôi những kiến thức cần thiết qua môn “Trí tuệ nhân tạo" để có thể hoàn thành dự án nay.

Tuy nhiên do thời gian cùng với trình độ hiểu biết còn hạn hẹp nên khó tránh khỏi những thiếu sót. Rất mong nhận được sự đóng góp của quý thầy cô và các bạn để dự án được hoàn thiện hơn.

Xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... ..................................................................................................................................... .....................................................................................................................................

.....................................................................................................................................

Đà Nẵng, ngày…. tháng…. năm 2023

Giảng viên hướng dẫn

**TS. Vũ Thị Trà**

MỤC LUC

[**LỜI MỞ ĐẦU** 6](#_Toc152146819)

[**1. Lý do chọn đề tài** 6](#_Toc152146820)

[**2. Mục tiêu của đồ án** 6](#_Toc152146821)

[**3. Phương pháp nghiên cứu** 6](#_Toc152146822)

[**Chương I: Tìm hiểu về thuật toán CNN?** 7](#_Toc152146823)

[**1. Giới thiệu mạng CNN** 7](#_Toc152146824)

[**2. Những lớp cơ bản của mạng CNN** 8](#_Toc152146825)

[**3. Kiến trúc của mạng CNN** 8](#_Toc152146826)

[**4. Cách lựa chọn tham số của mạng CNN** 10](#_Toc152146827)

[**Chương II: Tìm hiểu về thư viện lớn TensorFlow** 10](#_Toc152146828)

[**1. Lịch sử ra đời của TensorFlow** 10](#_Toc152146829)

[**2. Ví dụ TensorFlow** 11](#_Toc152146830)

[**3. Các option tải dữ liệu vào TensorFlow** 11](#_Toc152146831)

[**Chương III: Tìm hiểu về thư viện Keras** 12](#_Toc152146832)

[**1. Giới thiệu về thư viện Keras** 12](#_Toc152146833)

[**2. MNIST Dataset** 13](#_Toc152146834)

[**3. Ứng dụng của việc phân loại ảnh** 13](#_Toc152146835)

[**Chương IV: Ứng dụng nhận diện mèo và chó** 14](#_Toc152146836)

[**1. Giới thiệu về dự án** 14](#_Toc152146837)

[**2. Bộ dữ liệu ( Mèo và Chó)** 14](#_Toc152146838)

[**3. Các bước xây dựng ứng dụng** 14](#_Toc152146839)

[**3.1 Tạo thư mục Kaggle và sao chép tệp kaggle.json vào thư mục này.** 14](#_Toc152146840)

[**3.2 Tải tập dữ liệu từ Kaggle bằng lệnh kaggle datasets download.** 14](#_Toc152146841)

[**3.3 Giải nén tệp zip chứa tập dữ liệu.** 15](#_Toc152146842)

[**3.4 Import các thư viện cần thiết (tensorflow, keras) và các lớp mạng nơ-ron.** 15](#_Toc152146843)

[**3.5 Tạo các đối tượng train\_ds và validation\_ds để tải dữ liệu huấn luyện và kiểm tra từ thư mục tương ứng.** 15](#_Toc152146844)

[**3.6 Chuẩn hóa dữ liệu bằng cách chia tỷ lệ giá trị pixel về khoảng [0, 1].** 16](#_Toc152146845)

[**3.7 Xây dựng mô hình CNN bằng cách tạo một đối tượng Sequential và thêm các lớp Conv2D, BatchNormalization, MaxPooling2D, Flatten, Dense và Dropout.** 16](#_Toc152146846)

[**3.8 Biên dịch mô hình bằng cách chọn bộ tối ưu hóa, hàm mất mát và các độ đo đánh giá.** 17](#_Toc152146847)

[**3.9 Tiến hành huấn luyện mô hình bằng cách gọi phương thức fit trên dữ liệu huấn luyện và kiểm tra.** 18](#_Toc152146848)

[**3.10 Vẽ đồ thị biểu đồ để quan sát quá trình huấn luyện và kiểm tra của mô hình.** 18](#_Toc152146849)

[**3.11 Đọc và chuẩn bị hình ảnh kiểm tra bằng OpenCV.** 18](#_Toc152146850)

[**3.12 Thay đổi kích thước hình ảnh kiểm tra và chuẩn bị đầu vào cho mô hình.** 18](#_Toc152146851)

[**3.13 Dự đoán nhãn của hình ảnh kiểm tra bằng phương thức predict của mô hình.** 18](#_Toc152146852)

[**4. Thử nghiệm** 19](#_Toc152146853)

[**Chương V: Tài liệu tham khảo** 21](#_Toc152146854)

# **LỜI MỞ ĐẦU**

## **1. Lý do chọn đề tài**

Phân loại ảnh là một trong những bài toán cơ bản và quan trọng trong lĩnh vực thị giác máy tính. Bài toán này có nhiều ứng dụng thực tế trong đời sống như nhận dạng khuôn mặt, phân loại vật thể, tìm kiếm hình ảnh,...

Trong số các phương pháp phân loại ảnh, mạng nơ-ron tích chập (CNN) là một trong những phương pháp hiệu quả nhất. CNN có khả năng học các đặc trưng từ dữ liệu hình ảnh một cách tự động, giúp phân loại ảnh một cách chính xác.

## **2. Mục tiêu của đồ án**

Mục tiêu của đồ án là xây dựng một mô hình CNN hiệu quả để phân loại ảnh mèo và chó. Mô hình CNN được xây dựng sẽ có khả năng phân loại ảnh mèo và chó với độ chính xác cao.

## **3. Phương pháp nghiên cứu**

Được thực hiện theo các bước sau:

1. Thu thập và chuẩn bị dữ liệu hình ảnh

2. Xây dựng kiến trúc mô hình CNN

3. Huấn luyện mô hình CNN

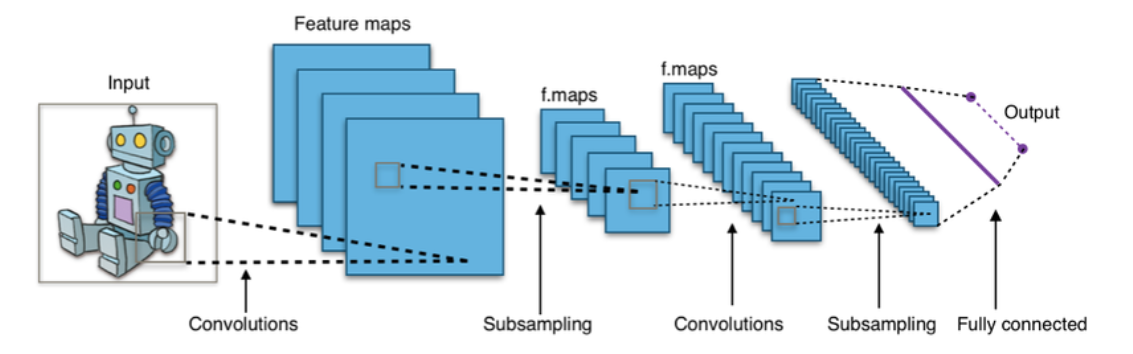
4. Đánh giá hiệu quả của mô hình CNN

# **Chương I: Tìm hiểu về thuật toán CNN?**

## **1. Giới thiệu mạng CNN**

CNN là từ viết tắt của cụm Convolutional Neural Network hay là mạng nơ ron tích chập. Đây là một loại mạng nơ-ron nhân tạo (ANN) được sử dụng rộng rãi trong lĩnh vực thị giác máy tính. CNN có khả năng học các đặc trưng từ dữ liệu hình ảnh một cách tự động, giúp phân loại ảnh, nhận dạng đối tượng, theo dõi chuyển động,...

CNN hoạt động dựa trên nguyên tắc tích chập. Tích chập là một phép toán toán học được sử dụng để tìm các đặc trưng cục bộ trong một hình ảnh. CNN sử dụng các bộ lọc tích chập để tìm các đặc trưng này. Mỗi bộ lọc tích chập được thiết kế để tìm một loại đặc trưng cụ thể, chẳng hạn như cạnh, góc, đường viền,...

****

**Convolutional trong CNN**

Đây là một “cửa sổ” sử dụng trượt trên ma trận nhằm lấy được những tin chính xác và cần thiết nhất mà không phải chọn đặc trưng (feature).

Convolution hay phân tích chập là cách mà những lớp Convolutional này nhân những phần tử trong ma trận. Sliding Window hay kernel là dạng ma trận có kích thước nhỏ, sử dụng trong nhân tích chập với ma trận hình ảnh.

**Feature trong CNN**

Feature là đặc trưng, mạng CNN sẽ so sánh dựa vào từng mảnh và các mảnh như vậy được gọi là feature. Thay vì phải tiến hành khớp các bức ảnh lại với nhau thì mạng CNN sẽ xác định được sự tương đồng thông qua tìm kiếm thô những đặc trưng khớp với nhau bằng hai hình ảnh tốt hơn. Một feature là một hình ảnh dạng mini (những mảng 2 chiều nhỏ). Những feature này đều tương ứng với một khía cạnh nào đó của hình ảnh và chúng có thể khớp lại được với nhau.

## **2. Những lớp cơ bản của mạng CNN**

Các lớp cơ bản của mạng CNN bao gồm:

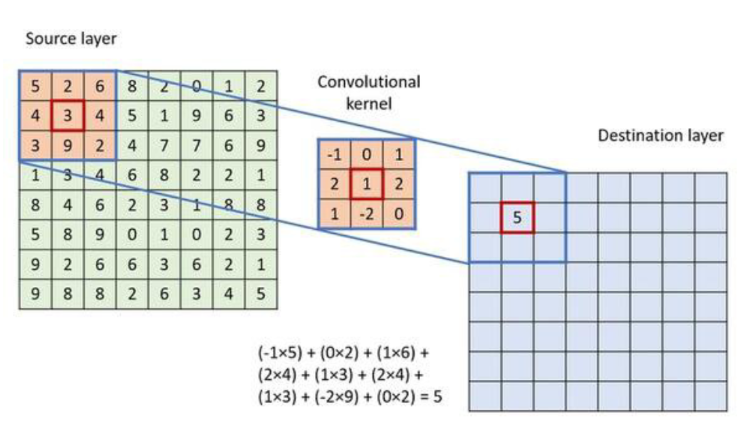
**+** Tầng đầu vào: Tầng này nhận đầu vào là hình ảnh. Hình ảnh được biểu diễn dưới dạng ma trận, trong đó mỗi pixel là một giá trị số.

**+** Tầng tích chập: Tầng này sử dụng các bộ lọc tích chập để tìm các đặc trưng cục bộ trong hình ảnh. Bộ lọc tích chập là một ma trận có kích thước nhỏ hơn kích thước của hình ảnh. Bộ lọc tích chập được di chuyển qua hình ảnh, và các giá trị của hình ảnh tại các vị trí tương ứng với các giá trị của bộ lọc tích chập sẽ được cộng lại với nhau.

**+** Tầng Pooling: Tầng này giảm kích thước của hình ảnh đầu ra từ tầng tích chập. Điều này giúp giảm số lượng nơ-ron trong mạng, đồng thời giúp CNN trở nên hiệu quả hơn.

**+** Tầng Fully Connected: Tầng này kết nối tất cả các nơ-ron của các tầng trước đó. Tầng này tạo ra một biểu diễn toàn diện của hình ảnh.

**+** Tầng Output: Tầng này tạo ra đầu ra của CNN. Đầu ra của CNN thường là một giá trị hoặc một biểu diễn phân loại của hình ảnh.



## **3. Kiến trúc của mạng CNN**

Mạng CNN là gì? Đó là tập hợp những Convolutional layer xếp chồng lên nhau, đồng thời mạng sử dụng những hàm như ReLU và Tanh để kích hoạt các trọng số trong các node. Các lớp này sau khi qua các hàm activation sẽ có trọng số trong những node và có thể chế tạo ra những thông tin trừu tượng hơn đến với các lớp kế tiếp trong mạng.

Mạng CNN có tính kết hợp và tính bất biến. Tức là, nếu cùng một đối tượng mà sử dụng chiếu theo các góc độ khác nhau thì sẽ có ảnh hưởng đến độ chính xác. Với dịch chuyển, co giãn hay quay ma trận ảnh thì lớp Pooling sẽ được dùng để hổ trợ làm bất biến các tính chất này. Chính vì vậy mà mạng CNN sẽ đưa ra những kết quả có độ chính xác tương ứng với từng mô hình.

Trong đó, lớp Pooling sẽ có khả năng tạo tính bất biến với phép dịch chuyển, co giãn và quay. Còn tính kết hợp cục bộ sẽ cho thấy những cấp độ biểu diễn, dữ liệu từ thấp đến cao với mức trừu tượng thông qua Convolution từ filter. Mạng CNN có những lớp liên kết nhau dựa vào cơ chế Convolution.

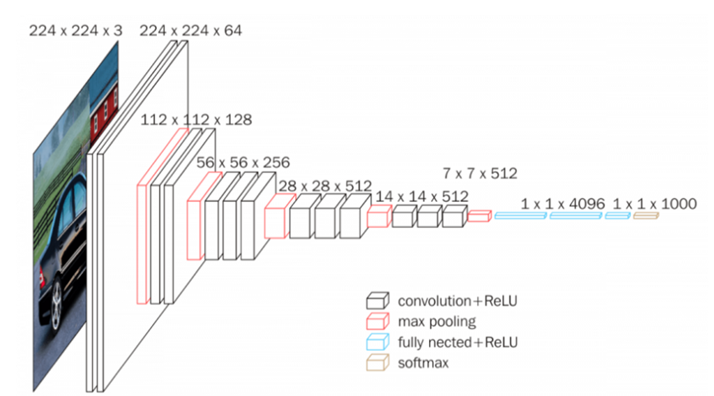
Các lớp tiếp theo sẽ là kết quả từ những lớp trước đó, vì vậy mà bạn sẽ có những liên kết cục bộ phù hợp nhất. Trong quá trình huấn luyện mạng, CNN sẽ tự học hỏi những giá trị thông qua filter layer dựa theo cách thức mà bạn thực hiện.

Cấu trúc cơ bản của một mô hình mạng CNN thường bao gồm 3 phần chính bao gồm:

**+** Trường cục bộ/ Local receptive field: Lớp này sử dụng để tách lọc dữ liệu, thông tin hình ảnh để từ đó có thể lựa chọn các vùng có giá trị sử dụng hiệu quả cao nhất.

**+** Trọng số chia sẻ/ Shared weights and bias: Lớp này hỗ trợ làm giảm các tham số đến mức tối thiếu trong mạng CNN. Trong từng lớp convolution sẽ chứa các feature map riêng và từng feature thì sẽ có khả năng phát hiện một vài feature trong hình ảnh.

**+** Lớp tổng hợp/ Pooling layer: Đây là lớp cuối cùng và sử dụng để làm đơn giản các thông tin output. Tức là, sau khi tính toán xong và quét qua các layer trong mạng thì pooling layer sẽ được dùng để lược bỏ các thông tin không hữu ích. Từ đó cho ra kết quả theo kỳ vọng người dùng.



## **4. Cách lựa chọn tham số của mạng CNN**

Để chọn tham số phù hợp nhất cho mạng CNN thì bạn cần chú ý đến những yếu tố như: filter size, so convolution, pooling size và việc train - test.

**+** Lớp Convolution: Số lượng lớp này càng nhiều thì sẽ giúp cải thiện đượchoạt động của chương trình. Sử dụng những lớp với số lượng lớn thì khảnăng hạn chế các tác động các tốt. Thông thường, chỉ sau khoảng 3 đến 4lớp bạn sẽ đạt được kết quả như kỳ vọng.

**+** Filter size: Kích thước thường chọn là ma trận 3 \* 3 hoặc ma trận 5 \* 5

**+** Pooling size: Với những hình ảnh thông thường, bạn nên chọn ma trậnpooling kích thước 2 \* 2 Với những ảnh kích thước lớn thì nên chọn matrận kích thước 3 \* 3

**+** Train - test: Cần thực hiện train - test nhiều lần để có thể cho ra nhữngparameter tốt nhất.

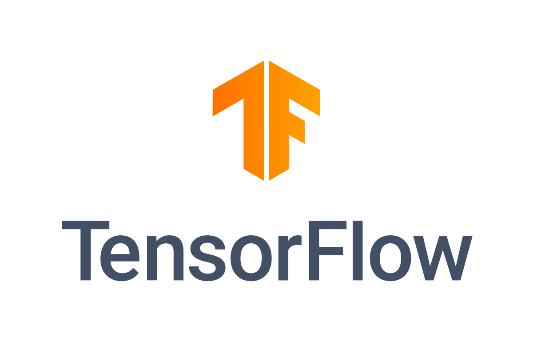
# **Chương II: Tìm hiểu về thư viện lớn TensorFlow**

## **1. Lịch sử ra đời của TensorFlow**

TensorFlow là một thư viện mã nguồn mở được phát triển bởi Google AI. TensorFlow được sử dụng để xây dựng các mô hình học máy, bao gồm các mô hình mạng nơ-ron nhân tạo (ANN).

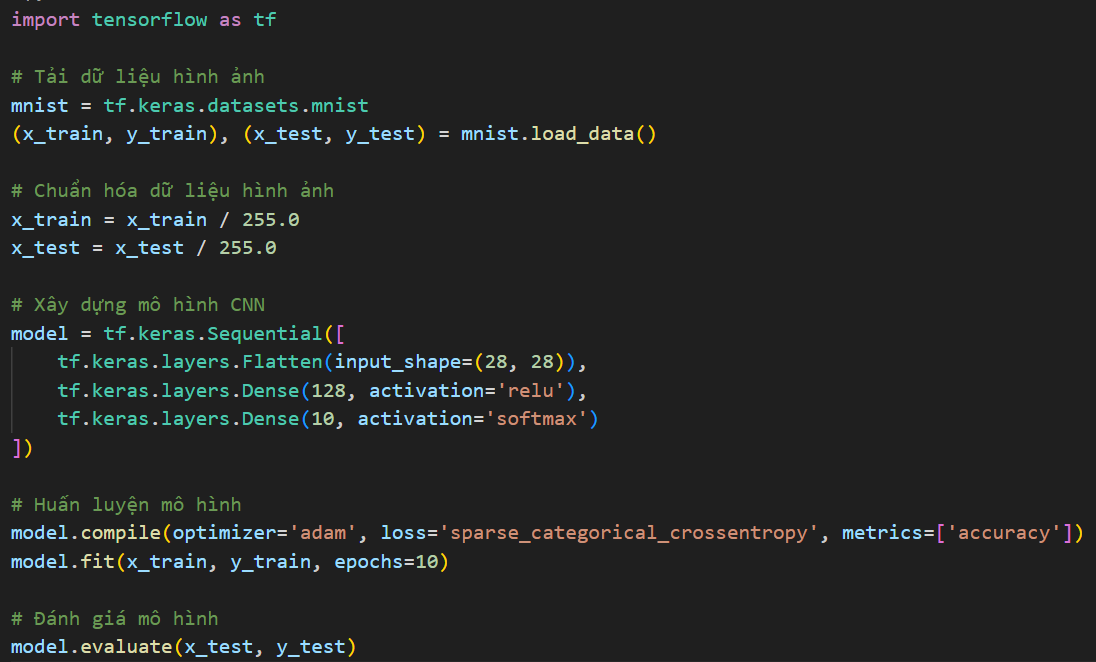
TensorFlow được phát triển bởi một nhóm các nhà nghiên cứu tại Google AI vào năm 2015. Nhóm nghiên cứu này được dẫn dắt bởi Jeff Dean, một trong những nhà tiên phong của lĩnh vực học máy.

TensorFlow được phát hành dưới dạng mã nguồn mở vào năm 2015. Ngay lập tức, TensorFlow đã trở thành một trong những thư viện học máy phổ biến nhất trên thế giới.



## **2. Ví dụ TensorFlow**

Dưới đây là một ví dụ đơn giản về cách sử dụng TensorFlow để xây dựng một mô hình mạng nơ-ron tích chập (CNN) để phân loại ảnh:

****

Kết quả của ví dụ này cho thấy mô hình CNN có thể đạt độ chính xác lên tới 98% trong việc phân loại ảnh số.

## **3. Các option tải dữ liệu vào TensorFlow**

TensorFlow cung cấp nhiều cách khác nhau để tải dữ liệu vào mô hình. Một số option phổ biến bao gồm:

**+** Tải dữ liệu từ tệp: TensorFlow hỗ trợ nhiều định dạng tệp dữ liệu, bao gồm CSV, JSON, và TFRecord.

**+** Tải dữ liệu từ cơ sở dữ liệu: TensorFlow có thể truy cập các cơ sở dữ liệu phổ biến, chẳng hạn như MySQL và PostgreSQL.

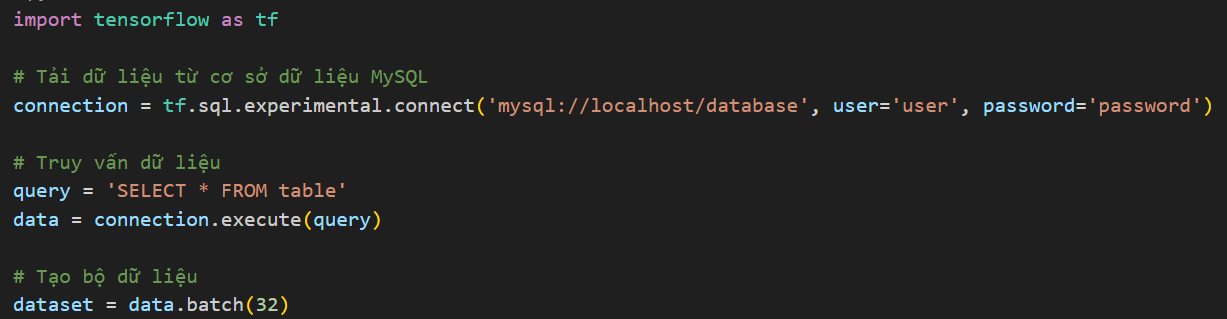
**+** Tải dữ liệu từ mạng: TensorFlow có thể tải dữ liệu từ các nguồn dữ liệu trực tuyến, chẳng hạn như API.

Dưới đây là một số ví dụ cụ thể về cách tải dữ liệu vào TensorFlow:

* Tải dữ liệu từ tệp CSV:



* Tải dữ liệu từ cơ sở dữ liệu MySQL:



* Tải dữ liệu từ mạng:



# **Chương III: Tìm hiểu về thư viện Keras**

## **1. Giới thiệu về thư viện Keras**

Keras là một thư viện mã nguồn mở được phát triển bởi Google AI. Keras được sử dụng để xây dựng các mô hình học máy, bao gồm các mô hình mạng nơ-ron nhân tạo (ANN).

Keras được thiết kế để đơn giản hóa việc xây dựng và huấn luyện các mô hình học máy. Keras cung cấp một API cao cấp, dễ sử dụng, giúp người dùng tập trung vào việc xây dựng mô hình mà không cần phải lo lắng về các chi tiết kỹ thuật của TensorFlow, thư viện học máy mà Keras dựa trên.

Keras bao gồm một số lớp và hàm được định nghĩa sẵn, giúp người dùng dễ dàng xây dựng các mô hình học máy phức tạp. Ngoài ra, Keras cũng cung cấp một số công cụ để giúp người dùng huấn luyện và đánh giá các mô hình học máy.



## **2. MNIST Dataset**

MNIST Dataset là một bộ dữ liệu hình ảnh chữ số viết tay. Bộ dữ liệu này bao gồm 60.000 hình ảnh chữ số được viết tay bởi người, được chia thành 60.000 hình ảnh huấn luyện và 10.000 hình ảnh kiểm tra.

MNIST Dataset là một bộ dữ liệu phổ biến được sử dụng để đào tạo các mô hình phân loại ảnh. Bộ dữ liệu này có kích thước nhỏ, dễ sử dụng và có kết quả phân loại chính xác cao, khiến nó trở thành một lựa chọn tốt cho người mới bắt đầu.

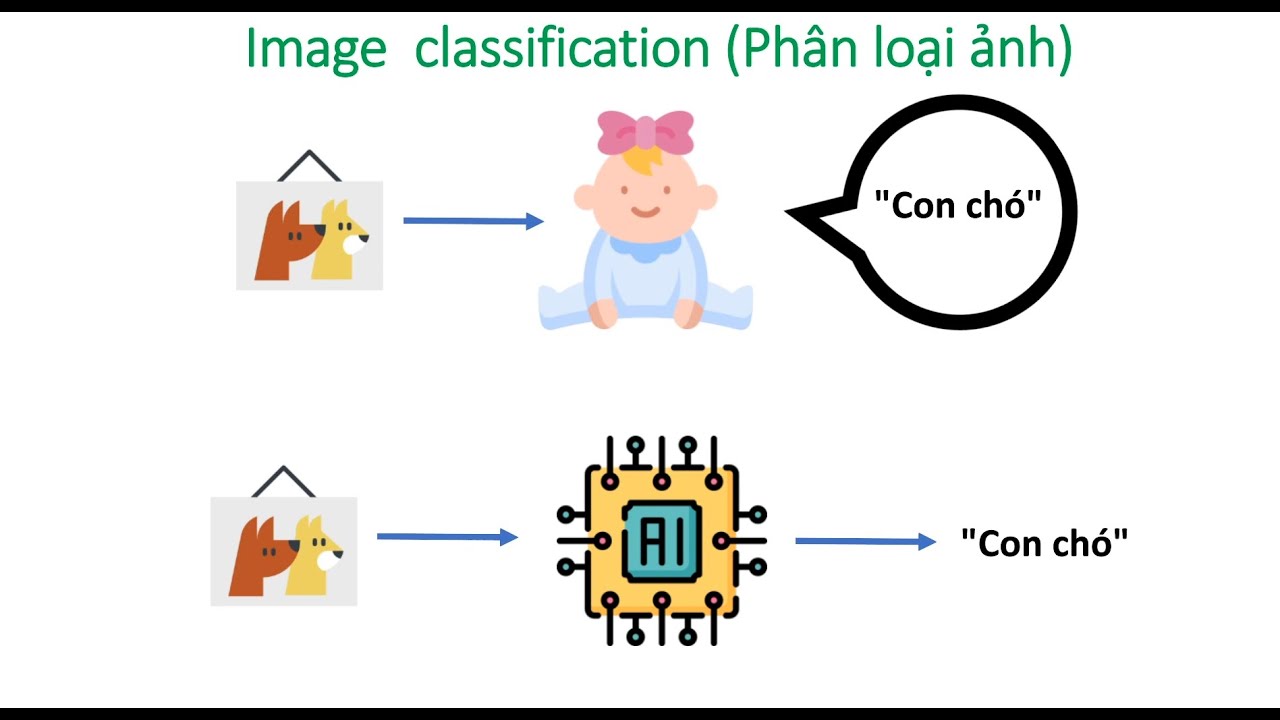
## **3. Ứng dụng của việc phân loại ảnh**

Phân loại ảnh là một bài toán học máy được sử dụng để phân loại các hình ảnh thành các lớp khác nhau. Bài toán này có nhiều ứng dụng trong thực tế, chẳng hạn như:

**+** Phân diện đối tượng: Phân loại ảnh có thể được sử dụng để phát hiện các đối tượng trong hình ảnh, chẳng hạn như con người, xe cộ, hoặc vật thể tự nhiên.

**+** Phân loại hình ảnh y tế: Phân loại ảnh có thể được sử dụng để phân loại các hình ảnh y tế, chẳng hạn như hình ảnh X-quang, hình ảnh MRI, hoặc hình ảnh CT.

**+** Phân loại hình ảnh sản phẩm: Phân loại ảnh có thể được sử dụng để phân loại các hình ảnh sản phẩm, chẳng hạn như hình ảnh quần áo, hình ảnh đồ điện tử, hoặc hình ảnh thực phẩm.



# **Chương IV: Ứng dụng nhận diện mèo và chó**

## **1. Giới thiệu về dự án**

Dự án này xây dựng một ứng dụng nhận diện mèo và chó sử dụng mô hình mạng nơ-ron tích chập (CNN). Ứng dụng cho phép người dùng tải lên hình ảnh mèo hoặc chó và ứng dụng sẽ phân loại hình ảnh đó là mèo hay chó. Chúng ta sẽ cùng khám phá cách xây dựng và training một mạng nơ-ron tích chập (CNN) để phân loại hình ảnh chó mèo.

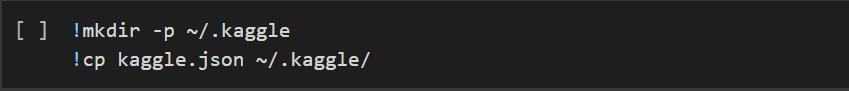
## **2. Bộ dữ liệu ( Mèo và Chó)**

Bộ dữ liệu “dogs vs cats” được tôi tham khảo và sử dụng trên kaggle tại link:” <https://www.kaggle.com/datasets/salader/dogs-vs-cats/data>” . Bộ dữ liệu này bao gồm 2 thư mục là train và test. Mục train có 20.000 ảnh bao gồm 10.000 ảnh mèo và 10.000 ảnh chó. Tương tự nhưng ít hơn, mục test có 5000 ảnh bao gồm 2500 ảnh mèo và 2500 ảnh chó.

## **3. Các bước xây dựng ứng dụng**

Ứng dụng được cho chạy trên Google Colab

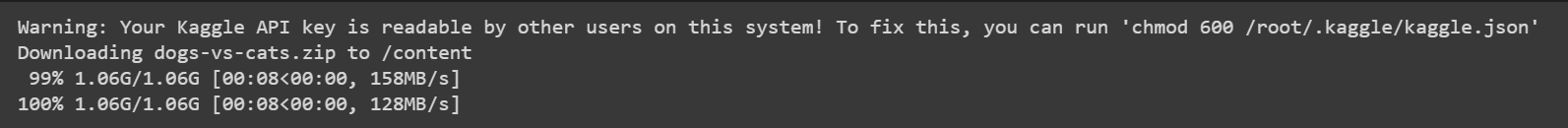
### **3.1 Tạo thư mục Kaggle và sao chép tệp kaggle.json vào thư mục này.**

****

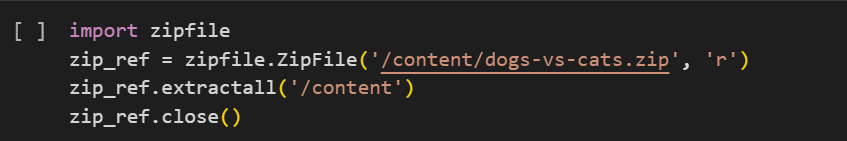
### **3.2 Tải tập dữ liệu từ Kaggle bằng lệnh kaggle datasets download.**

****

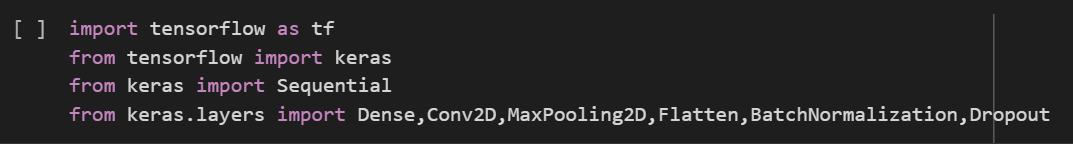
Sau khi cho chạy sẽ hiện ra dòng sau:

****

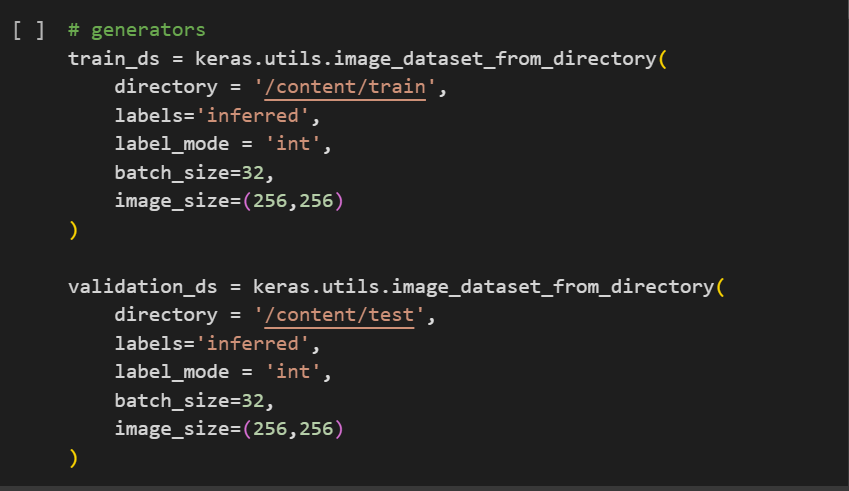
### **3.3 Giải nén tệp zip chứa tập dữ liệu.**

****

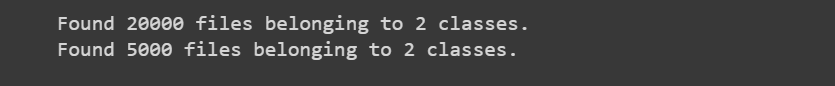
### **3.4 Import các thư viện cần thiết (tensorflow, keras) và các lớp mạng nơ-ron.**

****

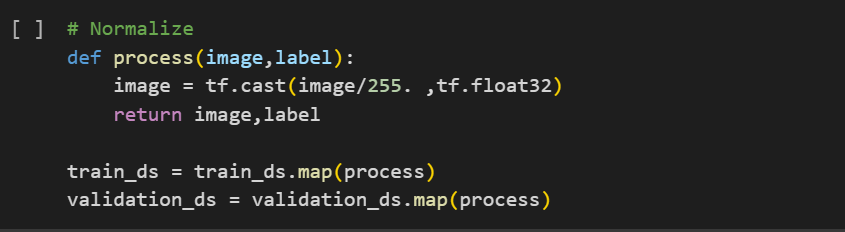
### **3.5 Tạo các đối tượng train\_ds và validation\_ds để tải dữ liệu huấn luyện và kiểm tra từ thư mục tương ứng.**

****

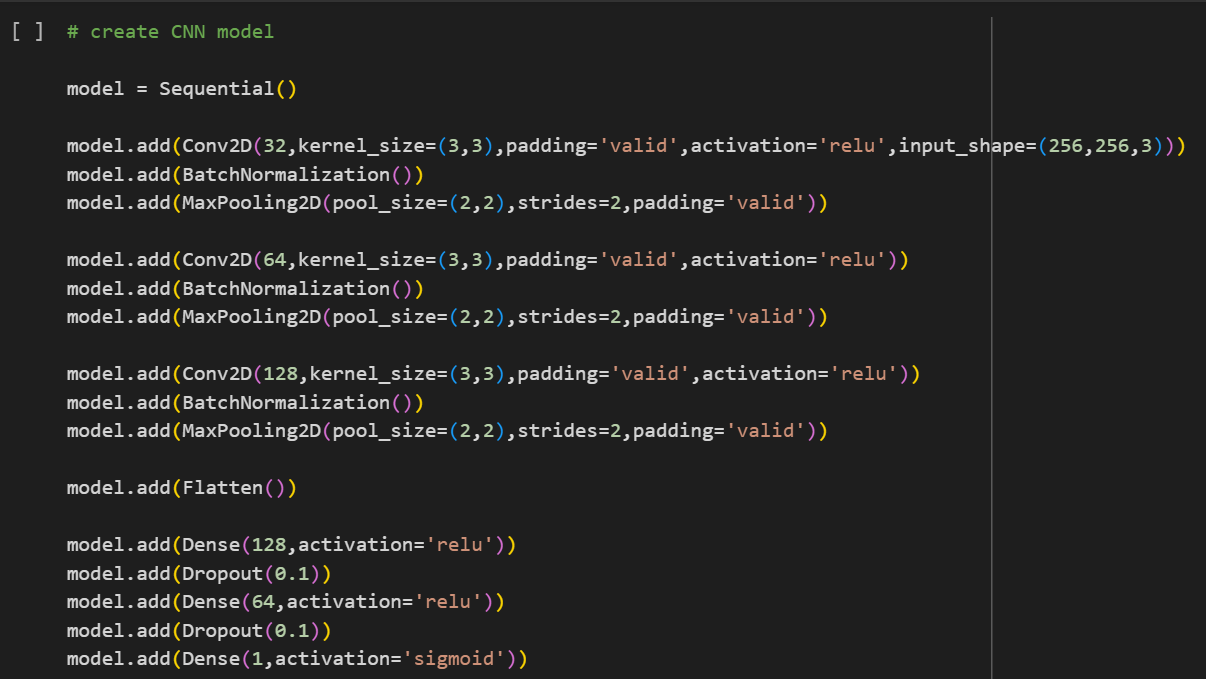
Cho chạy, nó sẻ hiển thị dòng:

****

### **3.6 Chuẩn hóa dữ liệu bằng cách chia tỷ lệ giá trị pixel về khoảng [0, 1].**

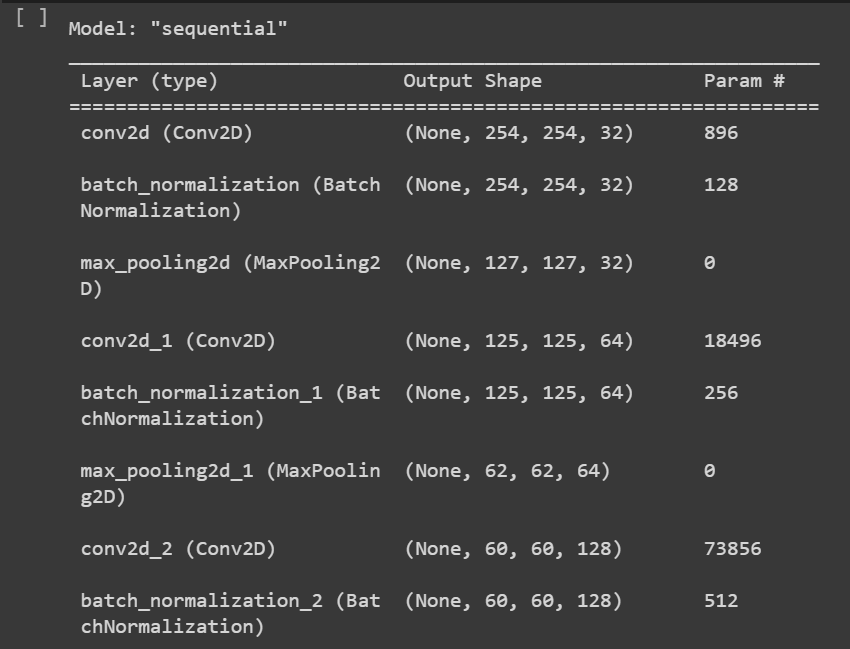
****

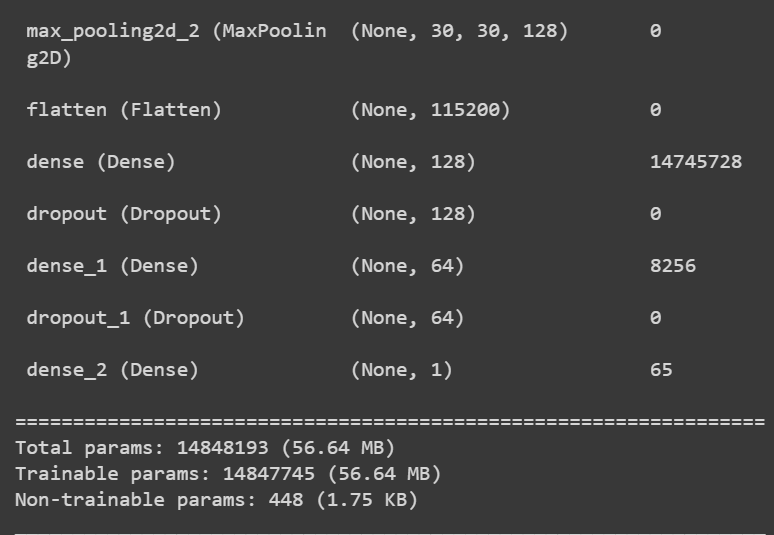
### **3.7 Xây dựng mô hình CNN bằng cách tạo một đối tượng Sequential và thêm các lớp Conv2D, BatchNormalization, MaxPooling2D, Flatten, Dense và Dropout.**

****

****

Sau khi cho chạy sẻ hiển thị:

****

****

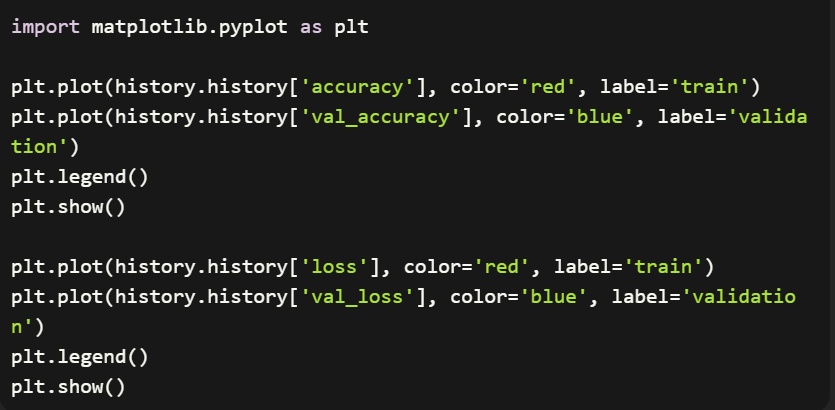
### **3.8 Biên dịch mô hình bằng cách chọn bộ tối ưu hóa, hàm mất mát và các độ đo đánh giá.**

****

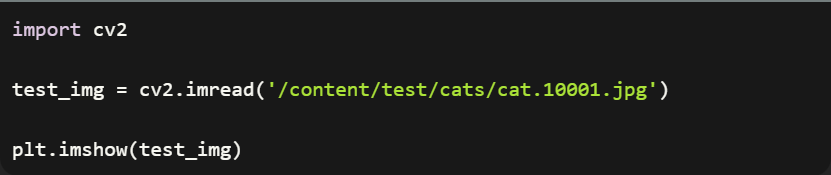
### **3.9 Tiến hành huấn luyện mô hình bằng cách gọi phương thức fit trên dữ liệu huấn luyện và kiểm tra.**

****

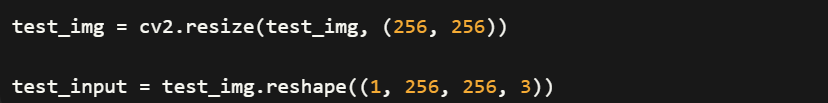
### **3.10 Vẽ đồ thị biểu đồ để quan sát quá trình huấn luyện và kiểm tra của mô hình.**

****

### **3.11 Đọc và chuẩn bị hình ảnh kiểm tra bằng OpenCV.**

****

### **3.12 Thay đổi kích thước hình ảnh kiểm tra và chuẩn bị đầu vào cho mô hình.**

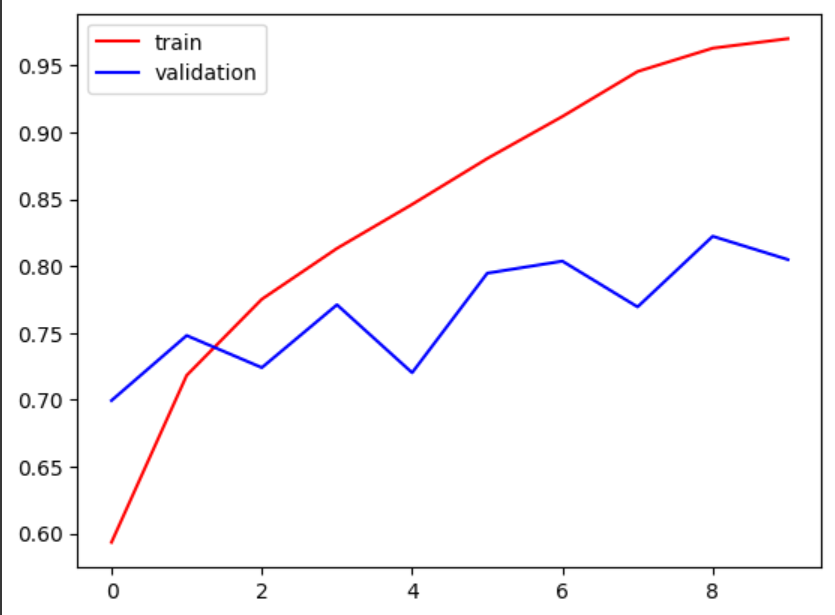
****

### **3.13 Dự đoán nhãn của hình ảnh kiểm tra bằng phương thức predict của mô hình.**

****

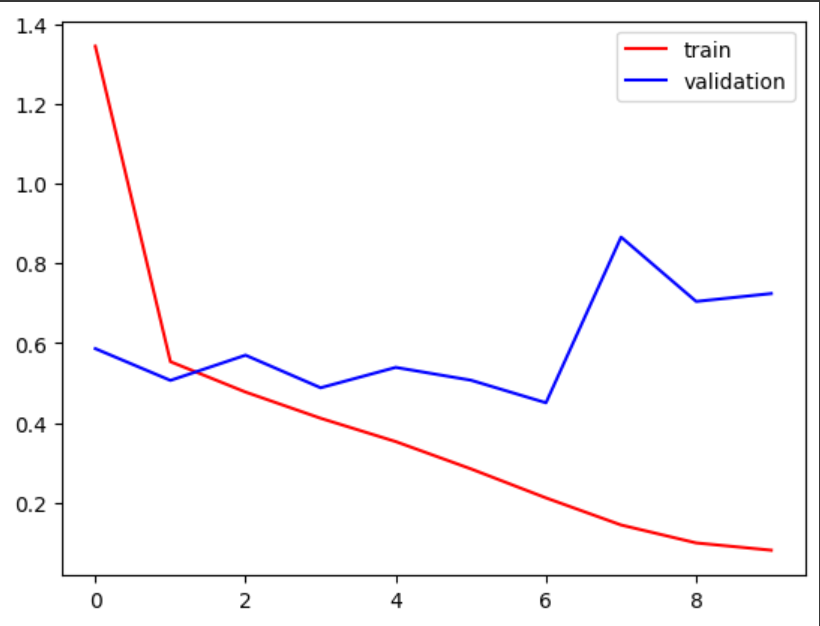
## **4. Thử nghiệm**

**Hàm vẽ biểu đồ cho độ chính xác (Accuracy):**

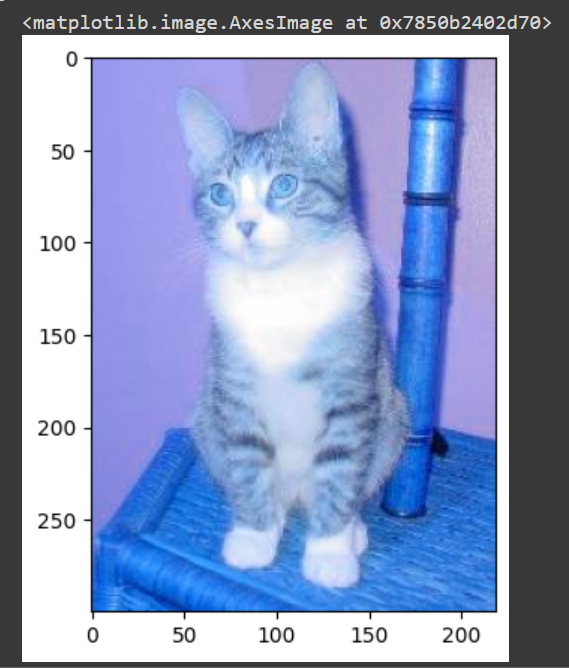
****

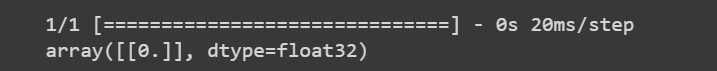
Biểu đồ này sẽ hiển thị giá trị độ chính xác của mô hình trên tập huấn luyện ('train') và tập kiểm tra ('validation') qua các epoch. Nó giúp theo dõi sự cải thiện hoặc overfitting của mô hình trong quá trình huấn luyện. Nếu đường cong 'train' và 'validation' gần nhau và tiến gần đến 1, điều đó cho thấy mô hình đang học tốt.

**Hàm vẽ biểu đồ cho hàm mất mát (Loss):**

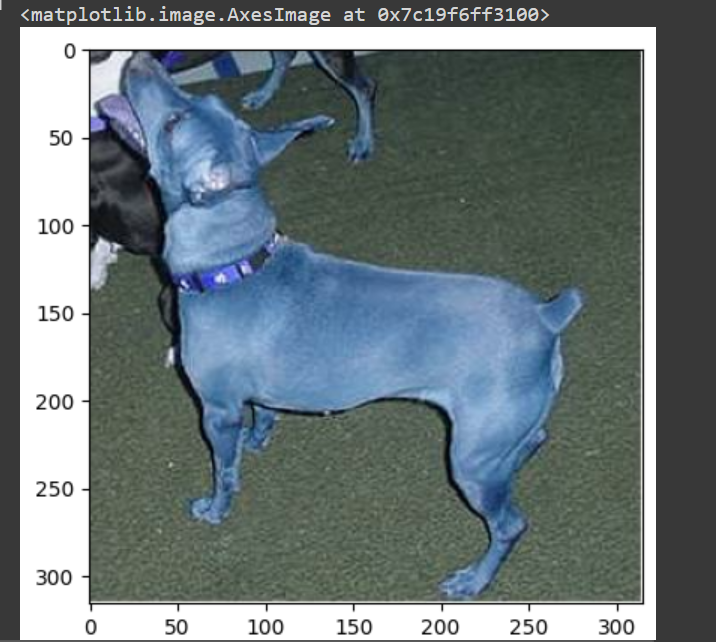
****

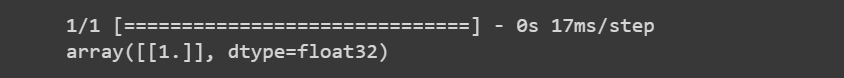
Biểu đồ này hiển thị giá trị hàm mất mát của mô hình trên tập huấn luyện ('train') và tập kiểm tra ('validation') qua các epoch. Nó giúp theo dõi sự giảm thiểu lỗi và đánh giá hiệu suất của mô hình. Nếu đường cong 'train' giảm và đường cong 'validation' giảm nhưng không quá nhanh, điều đó cho thấy mô hình đang học tốt và không bị overfitting.

****

****

**Theo như kết quả đã hiển thị “0” là 1 con mèo.**

****

****

**Theo như kết quả đã hiển thị “1” là 1 con chó.**

# **Chương V: Tài liệu tham khảo**

* [**www.nengo.ai/keras-spiking/examples/spiking-fashion-mnist.html**](http://www.nengo.ai/keras-spiking/examples/spiking-fashion-mnist.html)
* [**https://nttuan8.com/sach-deep-learning-co-ban/**](https://nttuan8.com/sach-deep-learning-co-ban/)
* [**https://www.tensorflow.org/tutorials/images/cnn?hl=vi**](https://www.tensorflow.org/tutorials/images/cnn?hl=vi)
* [**http://opencv.org/**](http://opencv.org/)
* [**http://www.numpy.org/**](http://www.numpy.org/)
* [**https://keras.io/**](https://keras.io/)