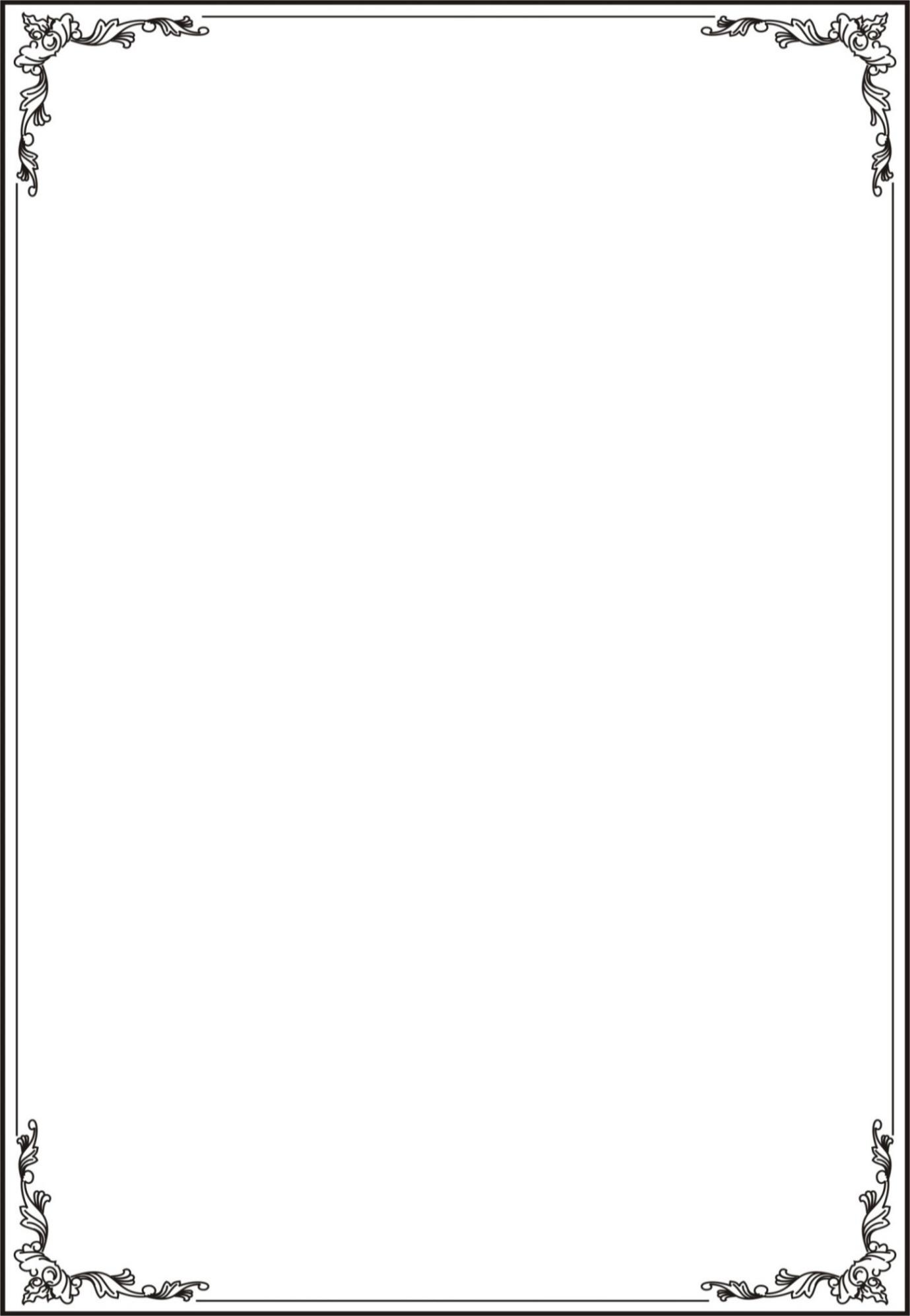
**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**



**THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG PHÂN LỚP ẢNH ĐỘNG VẬT**

**Sinh viên thực hiện:**

Nguyễn Ngọc Quang 18IT5

Nguyễn Trọng Tâm 18IT5

Nông Thị Hồng 18IT5

**Giáo viên hướng dẫn:** TS. Lê Thị Thu Nga

***Đà nẵng, tháng 12 năm 2021***

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**



**THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG PHÂN LỚP ẢNH ĐỘNG VẬT**

**Sinh viên thực hiện:**

Nguyễn Ngọc Quang 18IT5

Nguyễn Trọng Tâm 18IT5

Nông Thị Hồng 18IT5

**Giáo viên hướng dẫn:** TS. Lê Thị Thu Nga

***Đà nẵng, tháng 12 năm 2021***

**LỜI CẢM ƠN**

Chúng em xin được chân thành cảm ơn sự giúp đỡ tận tình của thầy giáo TS. Lê Thị Thu Nga đã cung cấp cho chúng em thêm những kiến thức về trí tuệ nhân tạo và xử lý ảnh, cũng như định hướng cho chúng em những phương pháp lập trình và cung cấp tài liệu để chúng em có thể hoàn thành tốt đồ án này.

Chúng em cũng xin trân trọng cảm ơn các thầy cô giáo giảng viên trong hội đồng sư phạm nhà Trường Đại Học Công Nghệ Thông Tin và Truyền Thông Việt - Hàn đã cung cấp cho chúng em nhiều kiến thức để phục vụ cho việc xây dựng đồ án môn học

Chúng em cũng xin gửi lời cảm ơn đến gia đình, bạn bè luôn động viên giúp đỡ chúng em trong suốt thời gian học tập và nghiên cứu, đóng góp những kinh nghiệm quý báu trong thời gian thực hiện đề tài này.

Kính chúc thầy cô mạnh khỏe, công tác tốt, tiếp tục giảng dạy và đào tạo thế hệ trẻ thành công. Một lần nữa chúng em trân trọng cảm ơn

MỤC LỤC

[Chương 1 TỔNG QUAN VỀ MẠNG THẦN KINH TÍCH CHẬP 2](#_Toc89638024)

[1.1 Giới thiệu 2](#_Toc89638025)

[1.2 Ảnh trong máy tính 3](#_Toc89638026)

[1.3 Phép tính convolution 6](#_Toc89638027)

[1.4 Convolutional neural network 8](#_Toc89638028)

[Chương 2 XÂY DỰNG ỨNG DỤNG PHÂN LỚP ẢNH ĐỘNG VẬT 11](#_Toc89638029)

[2.1 Phân tích và thiết kế 11](#_Toc89638030)

[2.2 Thêm thư viện 12](#_Toc89638031)

[2.3 Chuẩn bị dữ liệu 13](#_Toc89638032)

[2.4 Xây dựng mô hình 14](#_Toc89638033)

[2.5 Biểu diễn độ chính xác của mô hình 15](#_Toc89638034)

[2.6 Đánh giá mô hình 16](#_Toc89638035)

[2.7 Tạo ma trận nhầm lẫn 16](#_Toc89638036)

[2.8 Thử nghiệm hình ảnh trên mô hình 17](#_Toc89638037)

[Chương 3 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 18](#_Toc89638038)

[3.1 Kết luận 18](#_Toc89638039)

[3.2 Hướng phát triển 18](#_Toc89638040)

**DANH MỤC HÌNH**

[Hình 1.1 . Sơ đồ khối phân loại ảnh động vật 3](#_Toc89638052)

[Hình 1.2 Hệ màu RGB 4](#_Toc89638053)

[Hình 1.3 Hình hộp chữ nhật kích thước a\*b\*h 5](#_Toc89638054)

[Hình 1.4 Ảnh màu biểu diễn dưới dạng tensor 5](#_Toc89638055)

[Hình 1.5 Biểu diễn ảnh xám 6](#_Toc89638056)

[Hình 1.6 Phép tính Convolution 7](#_Toc89638057)

[Hình 1.7 Stride 8](#_Toc89638058)

[Hình 1.8 Mô hình mạng CNN 9](#_Toc89638059)

[Hình 1.9 Kiến trúc VGG16 9](#_Toc89638060)

[Hình 2.1 Sơ đồ hoạt động của dự án 11](#_Toc89638061)

[Hình 2.2 Mọt số thư viện được sử dụng 12](#_Toc89638062)

[Hình 2.3 chuẩn bị dữ liệu 13](#_Toc89638063)

[Hình 2.4 Xây dựng mô hình 14](#_Toc89638064)

[Hình 2.5 Tổng kết mô hình 15](#_Toc89638065)

[Hình 2.6 Biểu diễn độ chính xác mô hình 15](#_Toc89638066)

[Hình 2.7 1.1 Đánh giá mô hình 16](#_Toc89638067)

[Hình 2.8 Ma trận nhầm lẫn 17](#_Toc89638068)

[Hình 2.9 Thử nghiệm mô hình 17](#_Toc89638069)

**MỞ ĐẦU**

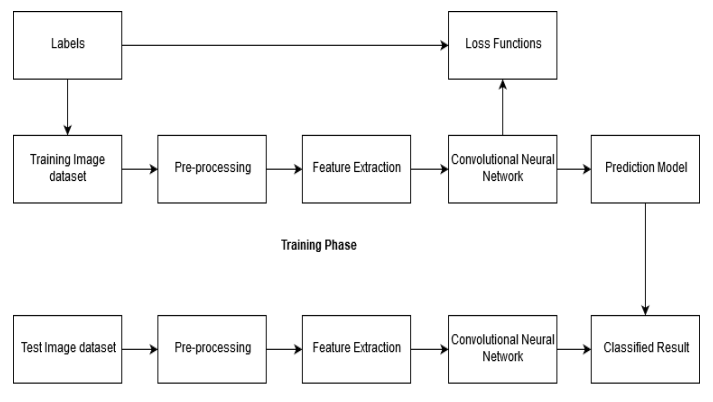
Ngày nay, với sự biến động ngày càng tăng, sự cần thiết và các ứng dụng của Trí tuệ nhân tạo, các lĩnh vực như Machine Learning và các tập con của nó, Deep Learning and Neural Networks đã đạt được sự thành công lớn. Neural Networks vẫn là mạng được tìm kiếm nhiều nhất cho sự lựa chọn cho các nhà khoa học máy tính để nhận dạng và xử lý hình ảnh và là một loại ANN (Mạng lưới thần kinh nhân tạo). Việc đào tạo cần có phần mềm và công cụ như bộ phân loại, dữ liệu, phân tích chúng và trích xuất các tính năng hữu ích. Các tính năng này sau đó được sử dụng để quan sát một mẫu và huấn luyện mạng sử dụng lại dữ liệu tương tự vào lần tiếp theo khi nó được cung cấp dữ liệu. Các các ứng dụng bao gồm các ứng dụng quân sự, ứng dụng công nghiệp, các ứng dụng IoT.... Đó cũng là lý do nhóm em chọn đề tài “Xây Dựng Ứng Dụng Phân Lớp Ảnh Động Vật “. Chúng em dự định đào tạo một mô hình học máy có độ chính xác cao bằng việc áp dụng mạng thần kinh tích hợp CNN.

# TỔNG QUAN VỀ MẠNG THẦN KINH TÍCH CHẬP

## Giới thiệu

Mục tiêu của dự án là phát triển một bộ phân loại hình ảnh động vật có độ chính xác cao, giúp xác định loài động vật dựa trên hình ảnh được đưa vào. Qua quá trình thực hiện dự án để tìm hiểu thêm nhiều kiến thức về machine learning củng như các kỹ thuật hiệu quả khác phục vụ cho dự án để đạt độ chính xác cao.

Ngày nay, với sự biến động ngày càng tăng, sự cần thiết và các ứng dụng của Trí tuệ nhân tạo, các lĩnh vực như Machine Learning và các tập con của nó, Deep Learning and Neural Networks đã đạt được thành công lớn. Nó đã trở thành một mô hình trung tâm dữ liệu. Các nhà phát triển đang "đào tạo" mạng trở nên "thông minh" và "độc lập". Việc đào tạo cần có phần mềm và công cụ như bộ phân loại, cung cấp một lượng lớn dữ liệu, phân tích chúng và trích xuất các tính năng hữu ích. Các tính năng này là sau đó được sử dụng để quan sát một mẫu và huấn luyện mạng sử dụng lại dữ liệu tương tự vào lần tiếp theo dữ liệu được cung cấp. Các ứng dụng bao gồm các ứng dụng quân sự, ứng dụng công nghiệp và IoT đó là lý do tại sao, chúng ta cần bộ phân loại để đạt được độ chính xác tối đa có thể. Theo dõi hiệu quả và đáng tin cậy các loài động vật hoang dã trong môi trường sống tự nhiên của chúng là điều cần thiết để cung cấp thông tin các quyết định bảo tồn và quản lý liên quan đến các loài động vật hoang dã, mô hình di cư, môi trường sống bảo vệ, và có thể là phục hồi và nhóm các loài động vật giống nhau lại với nhau. Xử lý một lượng lớn hình ảnh và video được chụp từ bẫy ảnh theo cách thủ công – là cực kỳ tốn kém, mất thời gian và cũng đơn điệu. Điều này gây trở ngại lớn cho các nhà khoa học và nhà sinh thái học để giám sát động vật hoang dã trong môi trường mở. Đặc biệt, chúng ta dự định sử dụng tập dữ liệu hình ảnh động vật và đào tạo một mạng nơ-ron phức hợp, có khả năng phân loại hình ảnh thành một lớp cụ thể với độ chính xác. Do đó, điều này có thể tăng tốc các kết quả nghiên cứu, cũng có thể dẫn đến việc khám phá các môi trường sống mới tiềm năng cũng như các loài động vật chưa từng thấy và / hoặc quý hiếm (hoặc đang trên bờ vực tuyệt chủng) - trong cùng một lớp, xây dựng các hệ thống giám sát hiệu quả hơn và các quyết định quản lý tiếp theo, có tiềm năng tạo ra những tác động đáng kể đến thế giới sinh thái và phân tích hình ảnh máy ảnh bẫy. Hình ảnh được chụp trong một trường đại diện cho một nhiệm vụ khó khăn trong khi phân loại vì chúng xuất hiện trong tư thế khác nhau, bối cảnh lộn xộn, ánh sáng khác nhau và điều kiện khí hậu, con người. Các lỗi chụp ảnh có thể gây ra sự biến dạng đáng kể, các góc khác nhau và khớp. Tất cả những thách thức này đòi hỏi một thuật toán hiệu quả để phân loại với hầu hết các độ chính xác tối ưu. Một trong những vấn đề lớn nhất là 'mất cân bằng giai cấp'. Vì có không đồng đều số lượng hình ảnh cho mỗi mẫu, thuật toán có thể đào tạo một số danh mục tốt hơn, như so với những người khác. Phạm vi của dự án là đào tạo mạng nơ-ron cho bộ phân loại hình ảnh động vật dựa trên mạng nơron phức hợp với kết quả chính xác, có thể được sử dụng trên hình ảnh động vật tập dữ liệu, cần phân bổ nhiều bộ nhớ hơn, độ phân giải tốt, tính nhất quán trong các thuộc tính và phân loại tốt.

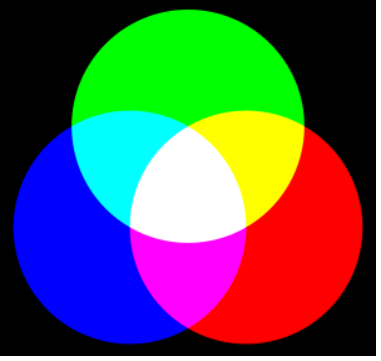


Hình 1.1 . Sơ đồ khối phân loại ảnh động vật

## Ảnh trong máy tính

**Hệ màu RGB**

RGB viết tắt của Red (đỏ), Green (xanh lục), Blue (xanh lam), là ba màu chính của ánh sáng khi tách ra từ lăng kính. Khi trộn ba màu trên theo tỉ lệ nhất định có thể tạo thành các màu khác nhau.



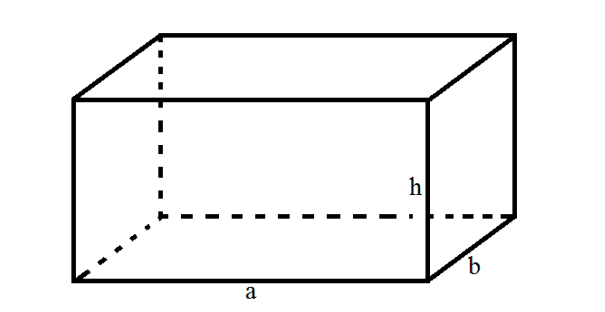
Hình 1.2 Hệ màu RGB

**Ảnh màu**

Ảnh màu là một ma trận các pixel mà mỗi pixel biểu diễn một điểm màu. Mỗi điểm màu được biểu diễn bằng bộ 3 số (r,g,b). Để tiện cho việc xử lý ảnh thì sẽ tách ma trận pixel ra 3 channel red, green, blue.

**Tensor**

Khi dữ liệu chiều hơn 2 nhiều thì sẽ được gọi là tensor, ví dụ như dữ liệu có 3 chiều. Để ý thì thấy là ma trận là sự kết hợp của các vector cùng kích thước. Xếp n vector kích thước m cạnh nhau thì sẽ được ma trận m\*n. Thì tensor 3 chiều cũng là sự kết hợp của các ma trận cùng kích thước, xếp k ma trận kích thước m\*n lên nhau sẽ được tensor kích thước m\*n\*k.

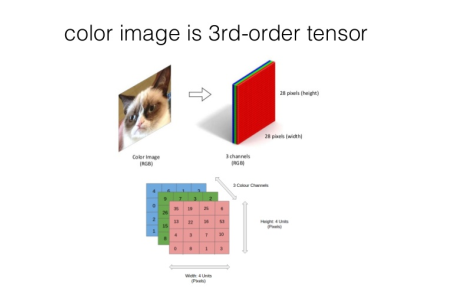


Hình 1.3 Hình hộp chữ nhật kích thước a\*b\*h

Tưởng tượng mặt đáy là một ma trận kích thước a \* b, được tạo bởi b vector kích thước a. Cả hình hộp là tensor 3 chiều kích thước a\*b\*h, được tạo bởi xếp h ma trận kích thước a\*b lên nhau.

Do đó biểu diễn ảnh màu trên máy tính ở phần trên sẽ được biểu diễn dưới dạng tensor 3 chiều kích thước 600\*800\*3 do có 3 ma trận (channel) màu red, green, blue kích thước 600\*800 chồng lên nhau.

Ví dụ biểu diễn ảnh màu kích thước 28\*28, biểu diễn dưới dạng tensor 28\*28\*3

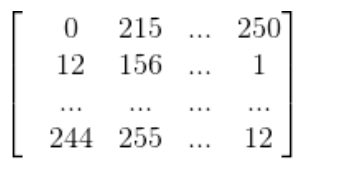


Hình 1.4 Ảnh màu biểu diễn dưới dạng tensor

**Ảnh xám**

Tương tự ảnh màu, ảnh xám nếu có kích thước 800 pixel \* 600 pixel, có thể biểu diễn dưới dạng một ma trận kích thước 600 \* 800 (vì định nghĩa ma trận là số hàng nhân số cột).

Tuy nhiên mỗi pixel trong ảnh xám chỉ cần biểu diễn bằng một giá trị nguyên trong khoảng từ [0,255] thay vì (r,g,b) như trong ảnh màu. Do đó khi biểu diễn ảnh xám trong máy tính chỉ cần một ma trận là đủ



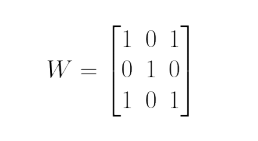
Hình 1.5 Biểu diễn ảnh xám

Giá trị 0 là màu đen, 255 là màu trắng và giá trị pixel càng gần 0 thì càng tối và càng gần 255 thì càng sáng

## Phép tính convolution

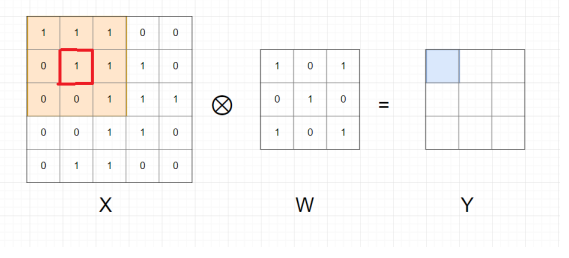
**Convolution**

Ta định nghĩa kernel là một ma trận vuông kích thước k\*k trong đó k là số lẻ. k có thể bằng 1, 3, 5, 7, 9,... Ví dụ kernel kích thước 3\*3



Kí hiệu phép tính convolution (⊗), kí hiệu Y = X ⊗W

Với mỗi phần tử xi j trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W có phần tử xi j làm trung tâm (đây là vì sao kích thước của kernel thường lẻ) gọi là ma trận A. Sau đó tính tổng các phần tử của phép tính element-wise của ma trận A và ma trận W, rồi viết vào ma trận kết quả Y.



Hình 1.6 Phép tính Convolution

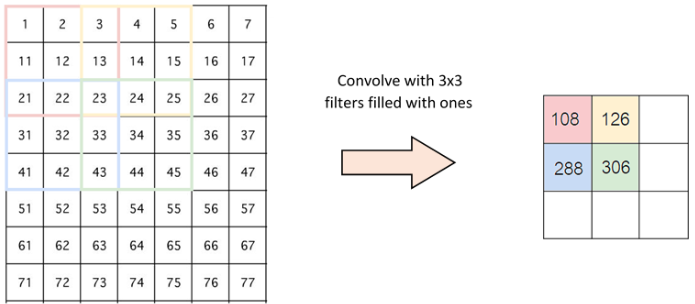
**Padding**

Như ở trên thì mỗi lần thực hiện phép tính convolution xong thì kích thước ma trận Y đều nhỏ hơn X. Tuy nhiên giờ ta muốn ma trận Y thu được có kích thước bằng ma trận X => Tìm cách giải quyết cho các phần tử ở viền Thêm giá trị ở viền ngoài ma trận X.

Mục đích của phép tính convolution trên ảnh là làm mờ, làm nét ảnh; xác định các đường,….

**Stride**

Stride là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy. Hình dưới là lớp tích chập hoạt động với stride là 2.



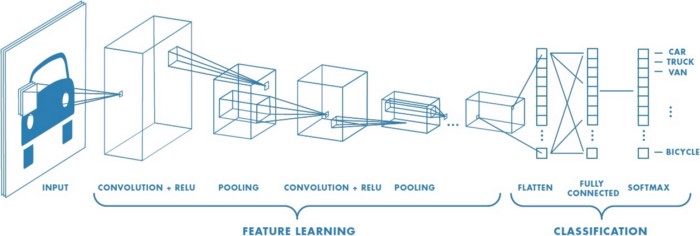
Hình 1.7 Stride

## Convolutional neural network

Trong mạng neural, mô hình mạng neural tích chập (CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là 1 trong số những lĩnh vực mà CNN được sử dụng rộng rãi.

CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao, W: Chiều rộng, D: Độ dày). Ví dụ: Hình ảnh là mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).

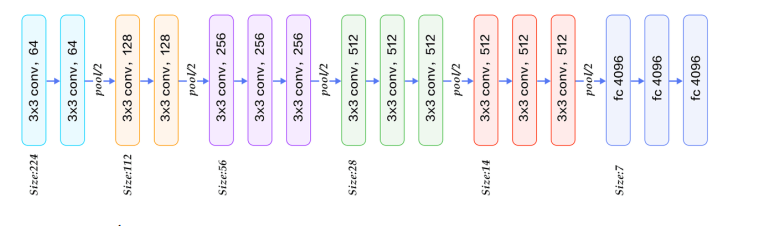
Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernals), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.



Hình 1.8 Mô hình mạng CNN

**Mạng VGG 16**

VGG16 là mạng convolutional neural network được đề xuất bởi K. Simonyan and A. Zisserman, University of Oxford. Model sau khi train bởi mạng VGG16 đạt độ chính xác 92.7% top-5 test trong dữ liệu ImageNet gồm 14 triệu hình ảnh thuộc 1000 lớp khác nhau. Giờ áp dụng kiến thức ở trên để phân tích mạng VGG 16.



Hình 1.9 Kiến trúc VGG16

* Convolutional layer: kích thước 3\*3, padding=1, stride=1. Tại sao không ghi stride, padding mà vẫn biết? Vì mặc định sẽ là stride=1 và padding để cho output cùng width và height với input.
* Pool/2: max pooling layer với size 2\*2
* 3\*3 conv, 64: thì 64 là số kernel áp dụng trong layer đấy, hay depth của output của layer đấy
* Càng các convolutional layer sau thì kích thước width, height càng giảm nhưng depth càng tăng.
* Sau khá nhiều convolutional layer và pooling layer thì dữ liệu được flatten và cho vào fully connected layer.

**Visualizing Convolutional Neural Network**

Bên cạnh những thứ rất thú vị mà mô hình học sâu mang lại, thứ mà thực sự bên trong mô hình học sâu vẫn được coi là một chiếc hộp đen. Rõ ràng chúng ta luôn muốn giải mã chiếc hộp đen đó, rằng tại sao mô hình lại phân biệt được các chữ số, hay tại sao tìm ra vị trí của chú chó bull và xa hơn là tại sao mô hình lại có thể chuẩn đoán một tế bào là tế bào ung thư hay không. Càng ứng dụng các mô hình học sâu vào đời sống, việc giải mã chiếc hộp đen càng quan trọng. Nó dẫn đến 2 vấn đề mà ta cần biết:

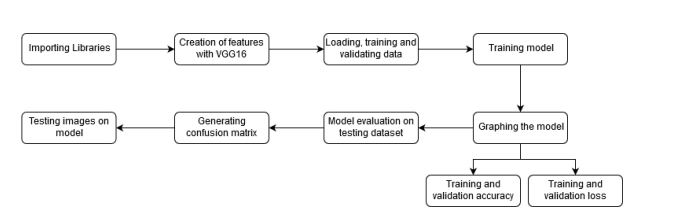
* Để hiểu mô hình học sâu làm gì, từ đó có những giải pháp cải tiến mô hình.
* Nhiều bài toán liên quan đến pháp lý cần lời giải thích hợp lý cho phương hướng giải quyết hơn là chấp nhận chiếc hộp đen

Thật may mắn khi mạng CNN có thể trực quan hoá mô hình làm những gì. Có 3 thành phần quan trọng trong mô hình CNN mà ta có thể visualize để hiểu mô hình CNN thực hiện việc gì:

* Features Map
* Convnet Filters
* Class Output

# XÂY DỰNG ỨNG DỤNG PHÂN LỚP ẢNH ĐỘNG VẬT

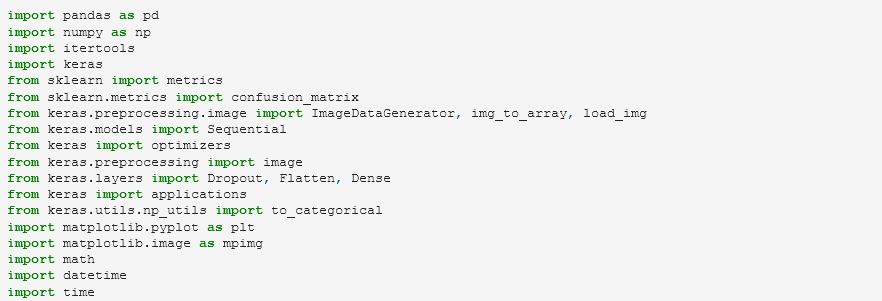
## Phân tích và thiết kế



Hình 2.1 Sơ đồ hoạt động của dự án

Ở đây, chúng ta trình bày một phương pháp phân loại các hình ảnh động vật. Hình 2.1. hiển thị thiết kế được đề xuất của mô hình cho Phân loại hình ảnh động vật bằng cách sử dụng CNN. Bộ dữ liệu động vật được sử dụng để phân loại là nguồn mở có sẵn từ Kaggle. Chúng ta đã sử dụng CNN với chức năng kích hoạt rò rỉ ReLU (Leaky ReLU) và kiến trúc VGG16 cho mô hình của chúng ta. Bước đầu tiên hệ thống thực hiện nhằm mục đích tạo ra các tính năng với mô hình VGG16. Ứng dụng Xử lý hình ảnh cùng với Tải, Kiểm tra, Đào tạo và Xác thực tập dữ liệu trước bước đào tạo giúp loại bỏ nhiễu, chướng ngại vật, biến dạng khỏi hình ảnh. Bước tiếp theo sử dụng CNN cùng với Leaky ReLU để huấn luyện mô hình phân loại chính xác và chính xác các lớp động vật. Sau khi đào tạo mô hình, chúng ta lập biểu đồ về độ chính xác và mất mát trong quá trình đào tạo và xác thực của mô hình để có thông tin chi tiết về mức độ được đào tạo của mô hình. Ít mất mát hơn, độ chính xác cao hơn. Bước tiếp theo là tạo ma trận phân loại và ma trận nhầm lẫn để có chi tiết chính xác về mức độ chính xác của mô hình được đào tạo và phân loại, vì chúng ta không thể chỉ dựa vào độ chính xác. Cuối cùng, chúng ta đã kiểm tra mô hình của mình với dữ liệu mẫu và nhận thấy nó được phân loại như thế nào.

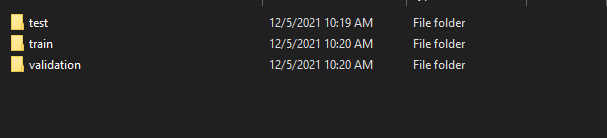
## Thêm thư viện



Hình 2.2 Mọt số thư viện được sử dụng

Nhập các thư viện cần thiết cho mạng nơ-ron, chẳng hạn như thư viện Panda, được sử dụng để cung cấp cấu trúc dữ liệu hiệu suất cao, dễ sử dụng và phân tích dữ liệu, NumPy được sử dụng để thực hiện các phép toán và logic trên mảng, Keras được thiết kế để cho phép thử nghiệm nhanh với các mạng nơ-ron sâu, nó tập trung vào việc thân thiện với người dùng, mô-đun và có thể mở rộng, v.v. Các hình ảnh đã được thay đổi kích thước, chiều rộng = 224 và chiều cao = 224 theo yêu cầu của mô hình VGG16. Một tệp nút cổ chai đã được tạo để khuyến khích mạng nén các biểu diễn tính năng để phù hợp nhất với không gian có sẵn, nhằm giảm thiểu tổn thất trong quá trình đào tạo. Hệ thống tệp Bottleneck được sử dụng để chuyển đổi tất cả các pixel hình ảnh thành số của chúng (mảng NumPy) và lưu trữ trong hệ thống lưu trữ. Chúng tôi chỉ đơn giản là cho mạng của mình biết vị trí của hình ảnh trong bộ nhớ của chúng tôi để máy biết đâu là cái gì. Chúng tôi đã xác định kích thước kỷ nguyên và lô cho mạng nơ-ron của mình. Chúng tôi đã phân loại tập dữ liệu hình ảnh thành tập, xác nhận và kiểm tra và tải nó vào mạng. Chúng tôi cũng đã tải một mô hình VGG16 được đào tạo trước vì nó chỉ sử dụng 11 lớp phức hợp và khá dễ làm việc với

## Chuẩn bị dữ liệu



Hình 2.3 chuẩn bị dữ liệu

Đầu tiên là tập training set. Đây thường là một tập dữ liệu có kích thước lớn, được dùng để training trong quá trình huấn luyện máy học. Nôm na dễ hiểu là, đây chính là tập dữ liệu máy dùng để học và rút trích được những đặc điểm quan trọng để ghi nhớ lại. Tập training set sẽ gồm 2 phần: - Input: sẽ là những dữ liệu đầu vào. Ví dụ với bài toán nhận dạng hình ảnh chẳng hạn: input sẽ là những bức hình. - Output: sẽ là những kết quả tương ứng với tập input. Ví dụ là những nhãn dán. Nếu input là ảnh con mèo, thì output sẽ là "con mèo". Tóm cái đuôi lại:

Testing set là tập dữ liệu dùng để test sau khi máy đã học xong. Một mô hình máy học sau khi được huấn luyện, sẽ cần phải được kiểm chứng xem nó có đạt hiểu quả ko. Cũng giống như con người, sau mỗi khóa học bạn phải có một bài kiểm tra cuối kì để lấy kết quả. Ai điểm cao thì có thưởng, ai điểm thấp thì nhịn chơi game... Mô hình máy học cũng vậy, sau mỗi quá trình huấn luyện gian khổ, các mô hình này sẽ được kiểm chứng độ chính xác, nếu đáp ứng được yêu cầu thì ok, không thì quăng. Và để kiểm nghiệm được độ chính xác của mô hình này, người ta dùng tập Testing set. Khác với Training set, Testing set chỉ gồm các giá trị input mà không có các giá trị output. Máy tính sẽ nhận những giá trị input này, và xử lý các giá trị, sau đó đưa ra output tương ứng cho giá trị input. Ví dụ, bạn đưa cho máy tính 1 lá bài hình con mèo : đây chính là giá trị input. Máy tính sẽ xử lý các chi tiết trên lá bài này và in ra màn hình: "con mèo"

Validation set là tập các giá trị input đi kèm với giá trị output và được dùng để kiểm thử độc chính xác của mô hình máy học trong quá trình huấn luyện.

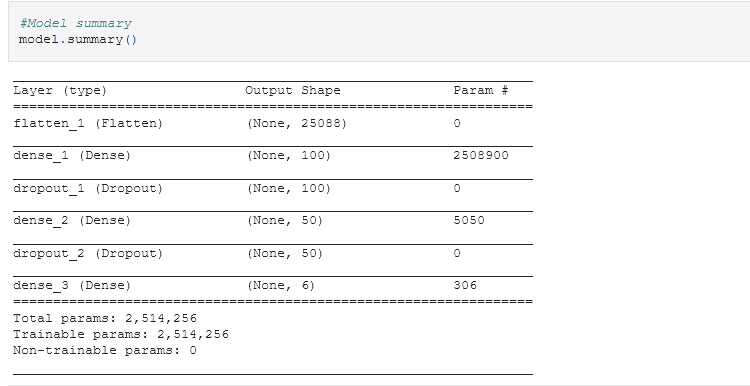
## Xây dựng mô hình

Đào tạo một mạng nơ-ron liên quan đến việc tìm một tập hợp các trọng số để bản đồ đầu vào tốt nhất cho đầu ra. Hàm Loss là một hàm cho chúng ta biết, mạng nơ-ron của chúng ta tốt như thế nào cho một nhiệm vụ nhất định. Vấn đề đào tạo tương đương với vấn đề giảm thiểu độ sai của mô hình. Thủ tục được sử dụng để thực hiện quá trình học tập trong mạng nơ-ron được gọi là thuật toán tối ưu hóa (hoặc trình tối ưu hóa).

Bước đầu tiên là khởi tạo mô hình với Sequential (). Sau đó, chúng ta làm phẳng dữ liệu của mình và thêm bổ sung 3 lớp ẩn. Sau đó, sau khi chúng ta đã tạo và biên dịch mô hình của mình, chúng ta phù hợp với đào tạo và xác nhận dữ liệu cho nó. Cuối cùng, chúng ta tạo một bước đánh giá, để kiểm tra độ chính xác của tập hợp đào tạo mô hình của chúng ta so với tập hợp xác thực.

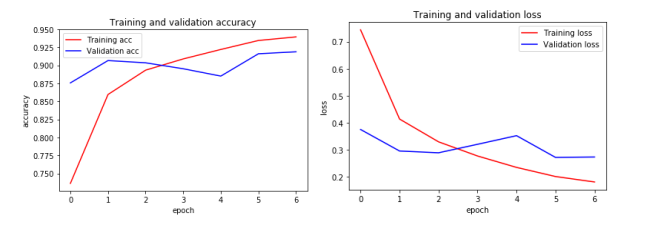


Hình 2.4 Xây dựng mô hình



Hình 2.5 Tổng kết mô hình

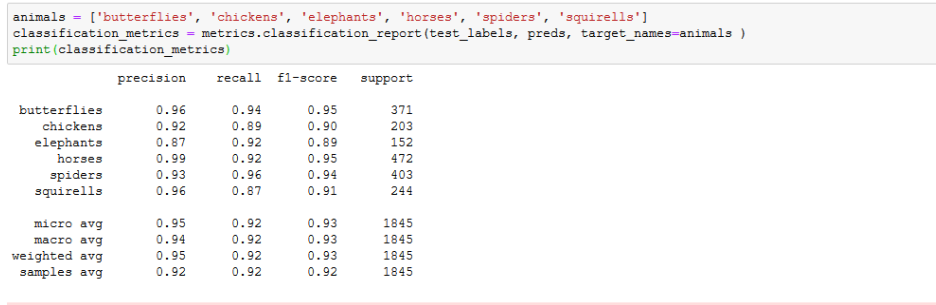
## Biểu diễn độ chính xác của mô hình



Hình 2.6 Biểu diễn độ chính xác mô hình

Chúng ta đang vẽ biểu đồ về độ chính xác và mất mát của quá trình đào tạo và xác thực cho từng kỷ nguyên. Trong một kỷ nguyên, hàm mất mát được tính toán trên mọi mục dữ liệu và nó được đảm bảo cung cấp định lượng số đo tổn thất tại một kỷ nguyên đã cho và đường cong biểu đồ qua các lần lặp lại chỉ cho biết tổn thất trên một tập hợp con của toàn bộ tập dữ liệu. Chức năng đánh giá mô hình cho biết máy chúng ta hoạt động tốt như thế nào được thực hiện có thể dự đoán dựa trên dữ liệu không nhìn thấy được. Đánh giá mô hình cho biết mức độ tốt của máy chúng ta được thực hiện có thể dự đoán dựa trên dữ liệu không nhìn thấy được

## Đánh giá mô hình



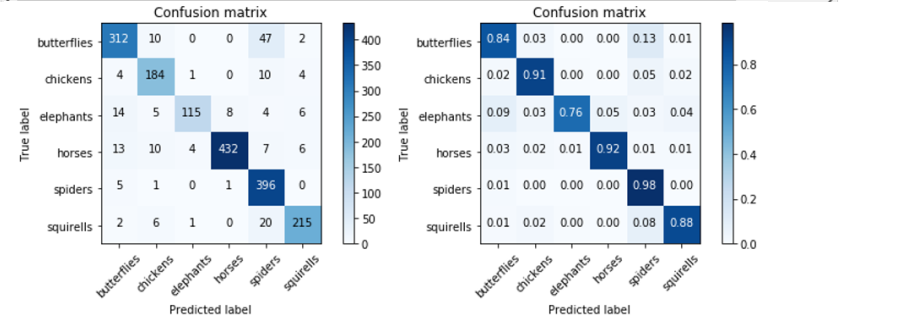
Hình 2.7 1.1 Đánh giá mô hình

Chúng ta đã triển khai và hiển thị các Chỉ số phân loại (precision, recall, F1-score, support), cùng với trung bình vi mô và vĩ mô và trung bình có trọng số và mẫu của chúng. Để sử dụng số liệu phân loại, chúng tôi phải chuyển đổi dữ liệu thử nghiệm của mình sang một định dạng NumPy khác, mảng NumPy để đọc

Precision được định nghĩa là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm mô hình dự đoán là Positive. Recall được định nghĩa là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm thật sự là Positive (hay tổng số điểm được gán nhãn là Positive ban đầu). Một số liệu phổ biến kết hợp độ chính xác và thu hồi được gọi là điểm F1. Support là số lượng mẫu phản hồi thực sự nằm trong lớp đó.

## Tạo ma trận nhầm lẫn

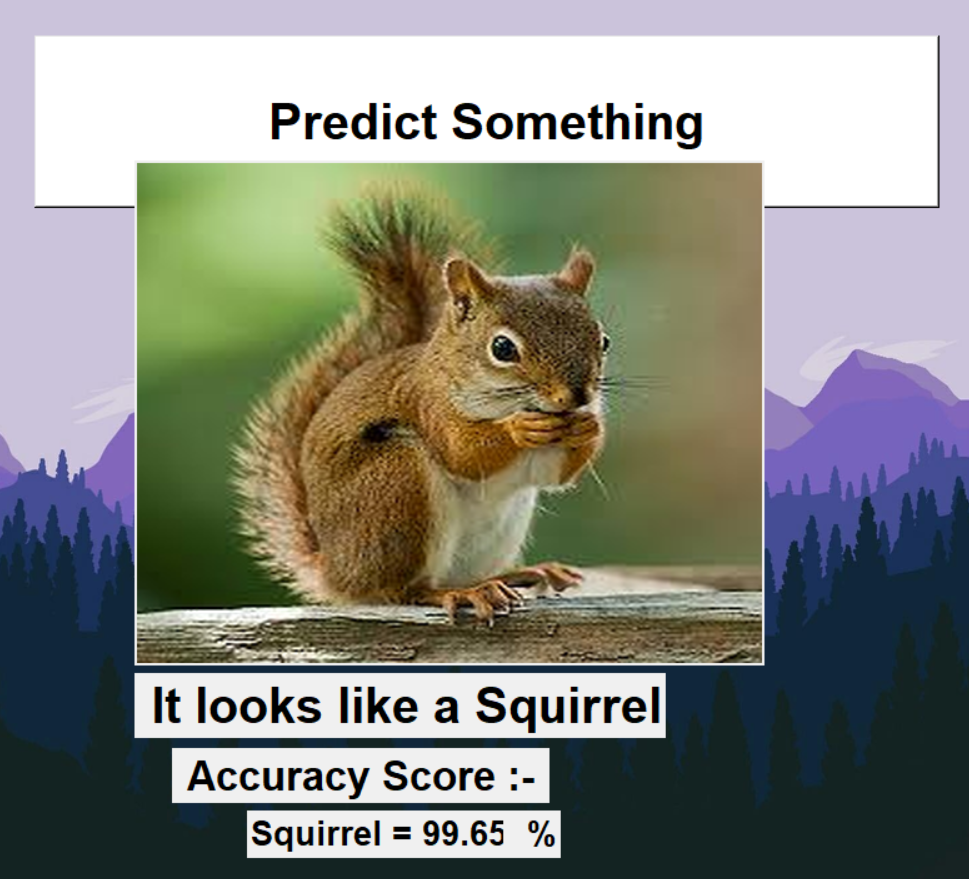
Chúng ta đã triển khai ma trận nhầm lẫn vì chúng ta không thể chỉ dựa vào độ chính xác. Ma trận nhầm lẫn là được sử dụng để đánh giá chất lượng đầu ra của bộ phân loại. Ma trận nhầm lẫn là một bảng trực quan hóa các dự đoán của mô hình so với các nhãn xác thực



Hình 2.8 Ma trận nhầm lẫn

## Thử nghiệm hình ảnh trên mô hình

Cuối cùng, giai đoạn cuối cùng là kiểm tra mô hình được đào tạo trên hình ảnh mẫu để kiểm tra xem mạng nơ-ron được huấn luyện chính xác và hoạt động thế nào



Hình 2.9 Thử nghiệm mô hình

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

**Ưu điểm:**

Xây dụng được mô hình phân loại hình ảnh một cách hoàn chỉnh, có độ chính xác cao với dữ liệu test được sử dụng là trên 80%.

Hiểu và áp dụng được quy trình xây dựng mô hình học máy trong CNN để hoàn thiện dự án phân lớp ảnh động vật

Hoàn tất dự án với chức năng đầy đủ đáp ứng được bài toán đặt ra.

**Nhược điểm:**  Dữ liệu chưa được phong phú nên có thể dẫn đến sai lệnh trong quá trình dự đoán.

## Hướng phát triển

Cần phát triển dự án bằng cách thêm nhiều data để tăng độ chính xác cho mô hình. Cần phát triển thêm ứng dụng để có thể mở rộng ra đa nền tảng phục vụ cho nhiều thiết bị khác nhau.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich. “Going deeper with convolutions”, IEEE Conference on Computer Vision and Pattern Recognition, 2015

[2] Manali Shaha, Meenakshi Pawar, “Transfer learning for image classificaion”, 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2018

[3]Nguyễn Thanh Tuấn, “Deep Learning Cơ Bản”, 2019