**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN MÔN HỌC**

**NHẬN DẠNG**

Đề tài: Nhận dạng và phân loại một số loài động vật

Nhóm thực hiện: GVHD: Ths. Đỗ Văn Tiến

Trần Nguyên Thuận 16521534

Nguyễn Minh Đức 16520242

Thành phố Hồ Chí Minh, tháng 12 năm 2019

**Phụ lục**

[I. Phân công công việc 3](#_Toc28702644)

[II. Giới thiệu bài toán 4](#_Toc28702645)

[**1.** **Mô tả bài toán:** 4](#_Toc28702646)

[**2.** **Tính ứng dụng:** 4](#_Toc28702647)

[**3.** **Ý tưởng thực hiện:** 4](#_Toc28702648)

[III. Giải quyết bài toán 5](#_Toc28702649)

[**1.** **Kiến thức tổng quan** 5](#_Toc28702650)

[**1.1.** **Convolutional Neural Network** 5](#_Toc28702651)

[**2.** **Thực hiện** 8](#_Toc28702652)

[**2.1.** **Thu thập, phân tích, phân chia dữ liệu** 8](#_Toc28702653)

[**2.2.** **Tiền xử lý tăng cường dữ liệu.** 10](#_Toc28702654)

[**2.3.** **Xây dựng mô hình** 11](#_Toc28702655)

[**2.4.** **Đánh giá mô hình mạng.** 12](#_Toc28702656)

[**2.5.** **Xây dựng mô hình nhận dạng.** 14](#_Toc28702657)

[IV. Kết quả, hạn chế. 17](#_Toc28702658)

[**1.** **Kết quả** 17](#_Toc28702659)

[**2.** **Hạn chế** 18](#_Toc28702660)

[V. Hướng phát triển. 19](#_Toc28702661)

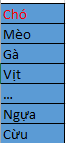
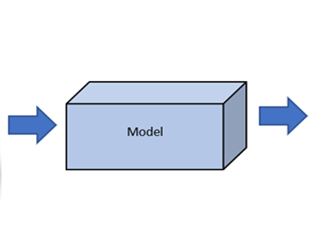
[VI. Chú thích, tài liệu tham khảo 20](#_Toc28702662)

1. **Phân công công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Công việc riêng | Công việc chung | Mức độ hoàn thành |
| Trần Nguyên Thuận | Code phân chia dữ liệu, train mô hình | - Tìm tài liệu  - Kiểm tra đánh giá lại mô hình | Hoàn thành |
| Nguyễn Minh Đức | Code phần kiểm tra đánh giá mô hình và nhận diện ảnh | Hoàn thành |

1. **Giới thiệu bài toán**
2. **Mô tả bài toán:**

* Bài toán đề cập tới việc phân loại hình ảnh con vật
* Input: Hình con vật
* Output: Trả về kết quả hình con vật đó có tên là gì



1. **Tính ứng dụng:**

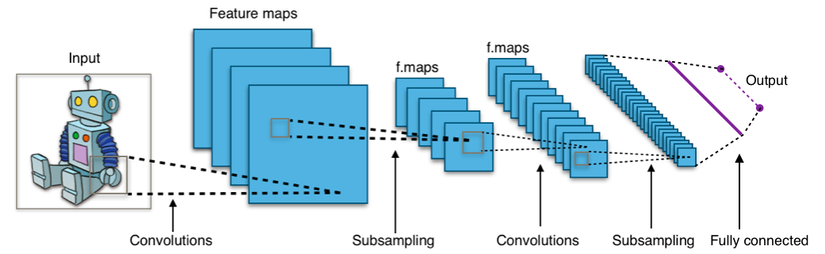
* Bài toán tuy chỉ mang tính phân loại nhưng có nhiều ứng dụng khác nhau. Nhóm đề cập tới một số vấn đề như sau:
* Áp dụng vào phát triển giáo dục cho trẻ em. Giúp trẻ em nhận diện và làm quen với một số loài động vật một cách dễ dàng, nhanh chóng, thông qua các ứng dụng liên quan

1. **Ý tưởng thực hiện:**

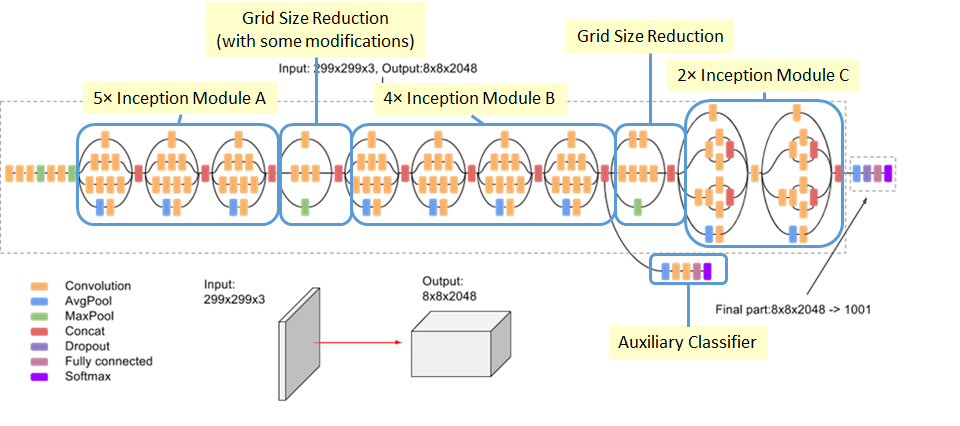
* Sử dụng neuron network để rút trích đặc trưng của tập dữ liệu data
* Train mô hình
* Nhập ảnh và nhận dạng

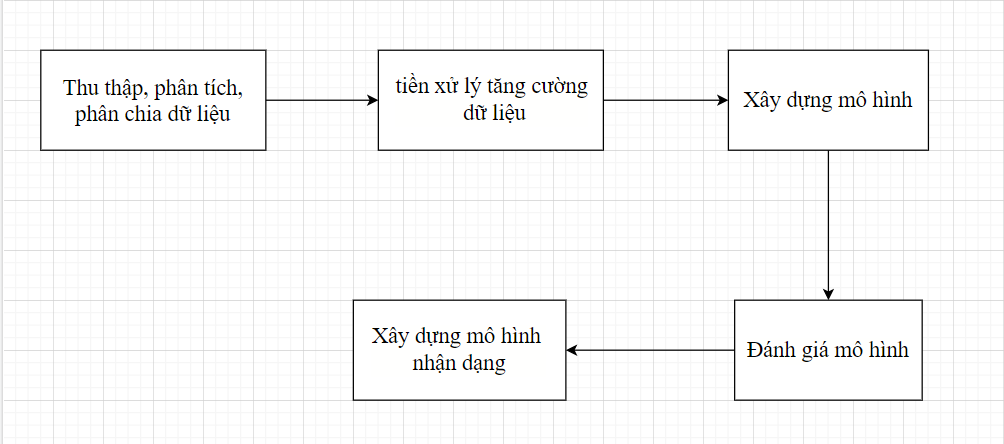
1. **Giải quyết bài toán**
2. **Kiến thức tổng quan**
   1. **Convolutional Neural Network**
      1. **Giới thiệu.**

* Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Như hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động.
* Trong lĩnh vực thị giác máy tính, CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh.
  + 1. **Convolutional là gì?**
* Là một cửa sổ trượt (Sliding Windows) trên một ma trận
* Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature bằng cách sử dụng các ma trận filter.
  + 1. **Kiến trúc mạng CNN**
* Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.
* Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.
* Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.
* Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.
* Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



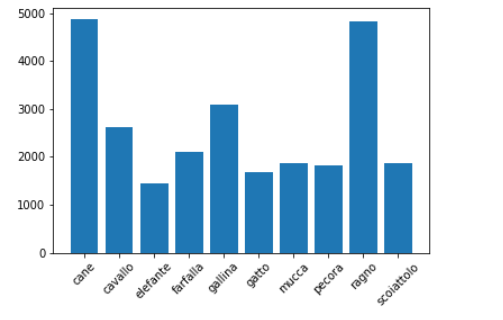
* Trong mô hình CNN có 2 khía cạnh cần quan tâm là **tính bất biến** (Location Invariance) và **tính kết hợp** (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.
* Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.
* Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.
* Mạng CNN sử dụng 3 ý tưởng cơ bản:
  + **các trường tiếp nhận cục bộ** (local receptive field)
  + **trọng số chia sẻ** (shared weights)
  + **tổng hợp** (pooling)
    1. **Mô hình CNN được sử dụng - InceptionV3**
* Inception v3 là một mô hình nhận dạng hình ảnh được sử dụng rộng rãi đã được chứng minh là đạt được độ chính xác cao hơn 78,1% trên bộ dữ liệu ImageNet. Mô hình này là đỉnh cao của nhiều ý tưởng được phát triển bởi nhiều nhà nghiên cứu trong những năm qua.
* Bản thân mô hình được tạo thành từ các khối symmetric and asymmetric building blocks, bao gồm các cấu trúc : Convolution, Average pooling, Max pooling, concats, dropouts, and fully connected layer. Batch-normalization được sử dụng rộng rãi và áp dụng cho các đầu vào. Loss function được tính thông qua Softmax.



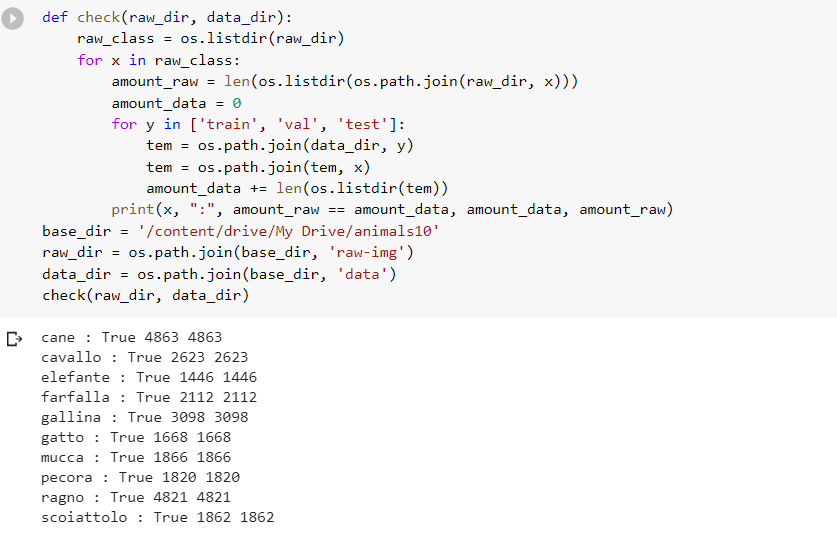
1. **Thực hiện**
   1. **Thu thập, phân tích, phân chia dữ liệu**

* Bộ dữ liệu được lấy từ Kaggle[1]
* Bộ dữ liệu gồm 28.000 ảnh phân làm 10 lớp tương ứng với 10 loại động vật.

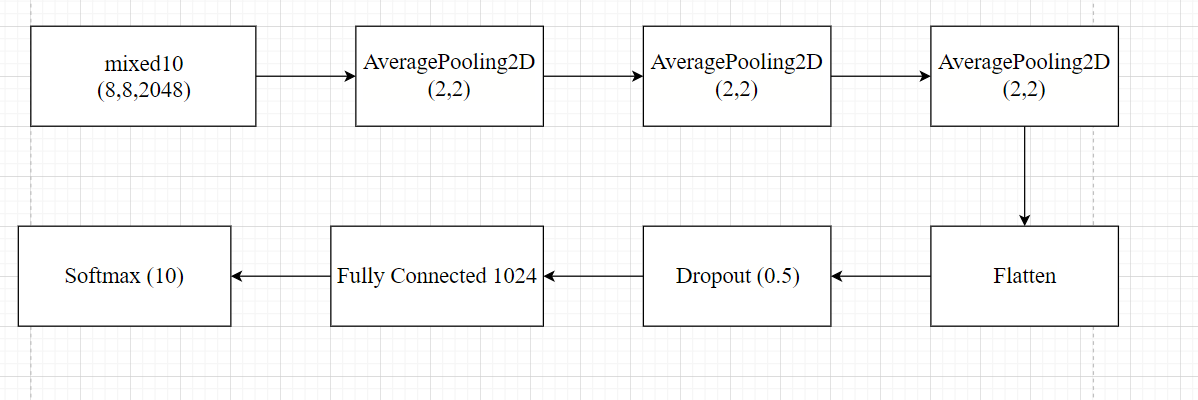
|  |  |
| --- | --- |
| Lớp | Loại động vật |
| Cane | Dog |
| Cavallo | Horse |
| Elefante | elephant |
| Farfalle | Butterfly |
| Gallina | Chicken |
| Gatto | Cat |
| Mucca | cow |
| Pecora | Sheep |
| Ragno | Spyder |
| Scoiattolo | squirrel |

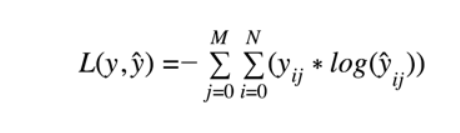
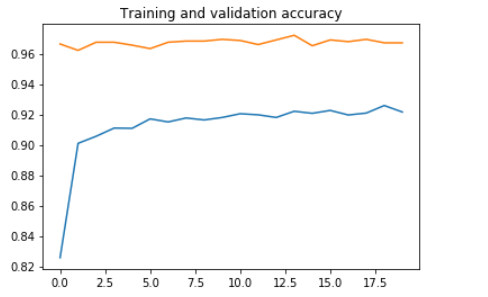


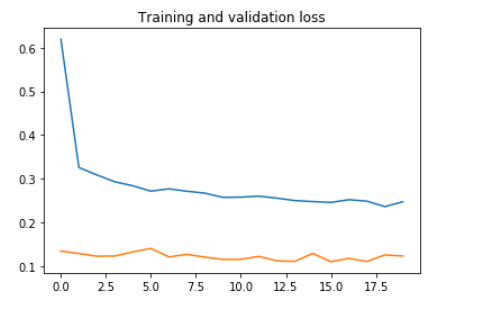
* Nhận xét: Bộ dữ liệu bị mất cân bằng.
* Để giải quyết bài toán ta chia bộ dữ liệu thành 3 phần với cách chọn ngẫu nhiên:
  + Phần Trainning: 70% dữ liệu tương ứng với 18322 ảnh.
  + Phần Testing: 20% dữ liệu tương ứng với 5242 ảnh.
  + Phần Validation: 10% dữ liệu tương ứng với 2615 ảnh.



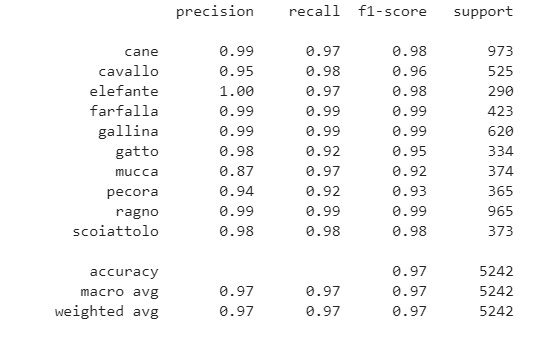
* 1. **Tiền xử lý tăng cường dữ liệu.**
* Dữ liệu sau khi được phân chia sẽ được tiếp tục xử lý qua các bước sau:
* Tăng cường dữ liệu tập train với các phép xoay, dịch trái, dịch phải, lật, zoom, tăng chỉnh độ sáng.
* Sau đó ảnh sẽ được resize về kích thước (299, 299, 3) để phù hợp với mô hình mạng Inception đang sử dụng ( được sử dụng cho cả 3 tập dữ liệu Train, Validation và Test ).
* Rescale: ảnh sẽ được đưa về dưới dạng ma trận ảnh nhị phân( được sử dụng cho cả 3 tập dữ liệu Train,Validation và Test).
  1. **Xây dựng mô hình**
* Sau khi phân tích và phân chia dữ liệu ta bắt đầu xây dựng mô hình giải quyết bài toán.
* Ta xây dựng mô hình mạng InceptionV3 để rút trích đặc chưng.
* Sau đó ta thiết kết thêm một mạng nữa để phân loại ảnh với đầu vào là mixed10 của mô hình mạng InceptionV3.

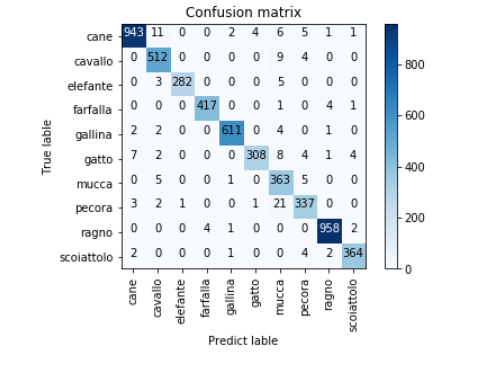


* Mô hình được xây dựng có 23.911.210 tham số, trong đó số lượng tham số Trainable là 2.108.426 tham số, số lượng tham số Non-trainable là 21,802,784 tham số.
  + 1. **Training**
* Epochs=20
* steps\_per\_epoch = 367
* validation\_steps = 131
* Hàm tối ưu hóa tự điều chỉnh learning rate được sử dụng là Adam.
* Hàm loss được sử dụng là categorical\_crossentropy vì dữ liệu ở đây là dữ liệu nhãn đơn.
* Hàm đánh giá được sử dụng là Accuracy, Confussion matrix.
  1. **Đánh giá mô hình mạng.**
* Để đánh giá mô hình mạng có tốt hay không ta đánh giá bằng các độ đo : 
* Theo độ đo accuracy:
  + Độ chính xác của quá trình Training đạt khoảng 0.9671.
  + Độ chính xác của quá trình Validation đạt khoảng 0.9216.



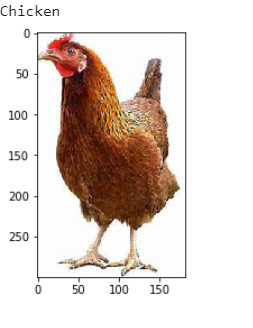
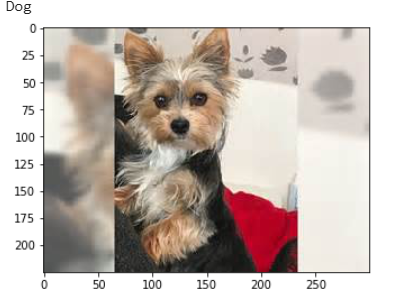
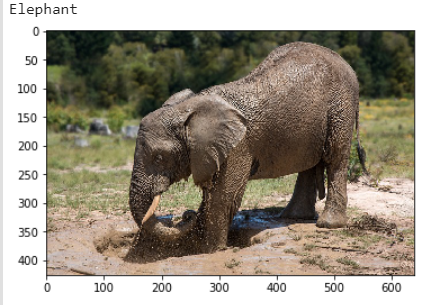
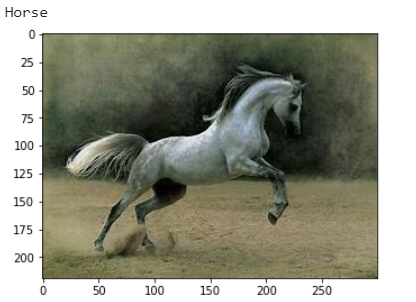
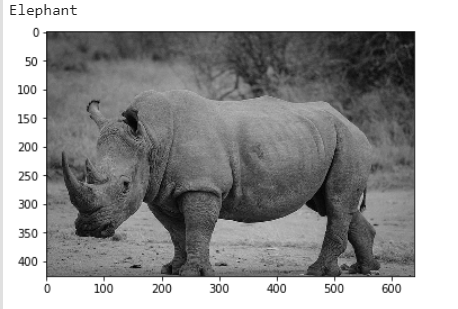
* Theo độ đo accuracy:
  + Loss function của quá trình Training đạt khoảng 0.2472.
  + Loss function của quá trình Validation đạt khoảng 0.1227.





* Ta có thể thấy mô hình cho ra kết quả có độ chính xác cao.
  1. **Xây dựng mô hình nhận dạng.**
* Sau khi sử dụng mô hình InceptionV3 để rút trích đặc chưng và phân loại ảnh ta lưu kết quả của mô hình và thiết kế 1 model để có thể nhận dạng ảnh để có thể sử dụng nhiều lần và nhanh chóng .
* Ta sử dụng hàm upload() của package google.colab.files để có thể upload ảnh lên colab, sau đó ta resize và rescale ảnh để tiến hành nhận dạng. Ta sử dụng hàm predict để so sánh và lấy label của ảnh.

1. **Kết quả, hạn chế.**
   * + 1. **Kết quả**

* Bài toán hoàn toàn dựa trên mô hình IncetionV3 nên kết quả đánh giá IncetionV3 chính là kết quả của bài toán. Vì vậy, độ chính xác của bài toán khoảng 0.97195727
* Để có thể xác thực hơn ta xem 1 số kết quả dưới đây: 
* tuy nhiên với 1 số trường hợp nhiễu mô hình cho ra kết quả chưa đúng.
* 
  + - 1. **Hạn chế**
* Mất cân bằng dữ liệu.
* Mất dữ liệu khi resize ảnh.

1. **Hướng phát triển.**

* Giải quyết vấn đề mất cân bằng dữ liệu
* Giải quyết vấn đề resize ảnh.

1. **Chú thích, tài liệu tham khảo**
2. <https://www.kaggle.com/alessiocorrado99/animals10>
3. <https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c>