# **PHẦN 1: THÔNG TIN TÓM TẮT (18521422)**

|  |  |
| --- | --- |
| **Tên đề tài (IN HOA)** | PHÂN BIỆT CHÓ VÀ MÈO |
| **Họ và tên (IN HOA)** | TRẦN THỊ PHƯƠNG THẢO |
| **Lớp - MSSV** | CS114.K21.KHTN - 18521422 |
| **Ảnh** |  |
| **Link Github chứa repos CS114.K21** | * <https://github.com/shaking54/CS114.K21.KHTN> |
| **Điểm đánh giá giữa kỳ (A|B|C|D)** | * C |
| **Thành tích để tính điểm bonus** | Không có |
| **Tóm tắt Bài tập quá trình** | * Số lần nộp bài tập Quá trình trên Classroom: 36/36 * Số lần nộp bài Thực hành trên Classroom:7/7 * Tự đánh giá (85/100): |
| **Tóm tắt Đồ án Cuối kỳ (không quá 500 từ)** | * Input đầu vào của bài toán là một bức ảnh có chứa con chó hoặc con mèo, output là dự đoán bức ảnh là chó hay mèo. Đối với dữ liệu ảnh thì cần phải có các bước tiền xử lý dữ liệu và rút trích đặc trưng ảnh. Sau đó, sử dụng dữ liệu vừa được rút trích để huấn luyện mô hình. Các thách thức là dữ liệu chưa đủ đa dạng về giống, loại, kích thước, số con của mỗi lớp. Dataset chứa nhiều ảnh của 1 con vật được chụp dẫn đến dữ liệu trong dataset có nhiều điểm tương đồng với nhau. Tìm kiếm cách thức rút trích đặc trưng phù hợp vs dữ liệu khó. Dẫn đến khi đánh giá model trên tập test của dataset cho kết quả cao hơn so với kiểm tra trên tập dữ liệu ngẫu nhiên. Kết quả đánh giá model cho kết quả tốt nhất là 87.1% với dataset tự thu thập và 62.5 với tập test lấy từ google. * Tự đánh giá (85/100): |
| **Link khác** | * Link đến báo cáo chi tiết : <https://github.com/Phuongthao2k/CS114.K21.KHTN/blob/master/FinalReport.docx> * Link đến báo cáo slides (pdf): <https://github.com/Phuongthao2k/CS114.K21.KHTN/blob/master/FinalPresentation.pptx> * Link đến báo cáo video (YouTube): |

# **PHẦN 2: BÁO CÁO TÓM TẮT ĐỒ ÁN CUỐI KỲ**

## **I.Mô tả bài toán**

Bài toán: Phân biệt chó và mèo

* Input: Một bức ảnh có chứa con chó hoặc con mèo
* Output: Đó là chó hay mèo

Kết quả đạt được:

* Đã xây dựng được nhiều model với các phương pháp lấy đặc trưng Hog
* Model tốt nhất đạt được kết quả 87.1% với test của dữ liệu tự thu thập và 62.5% với dữ liệu test thu thập từ google.

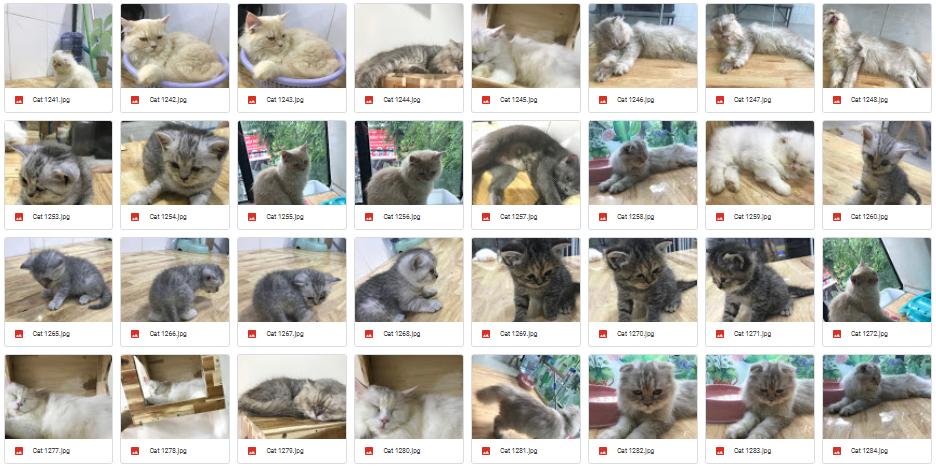
## **II. Mô tả bộ dữ liệu**

Về bộ dữ liệu:

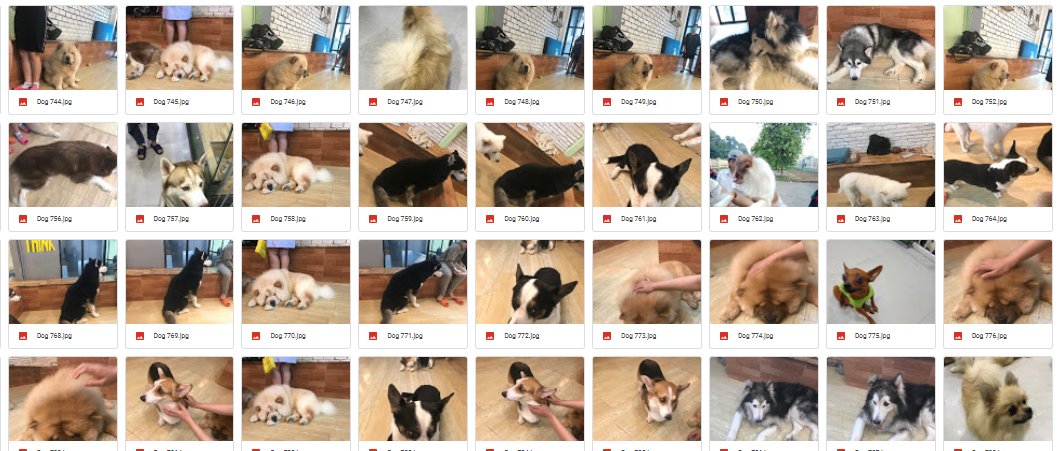
* + Dữ liệu tự thu thập tạo thành dataset.
  + Cat\_dog bộ dữ liệu cá nhân được lưu trữ trên drive.
  + Link: <https://drive.google.com/drive/folders/1rtY2aiZmCZezGejDAyXmVQbT7WFDywcf?usp=sharing>
  + Bộ dữ liệu dataset gồm 2600 bức ảnh, được chia đều cho 2 lớp chó / dog và mèo / cat.

Quá trình thu thập dữ liệu:

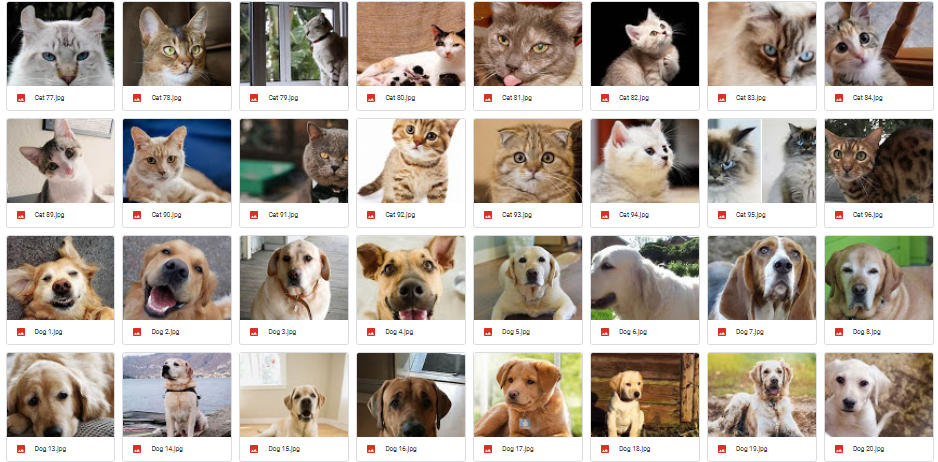
* Tìm kiếm các quán cà phê thú cưng, chó, mèo tại Sài Gòn. Tiến hành đi chụp ảnh trực tiếp tại quán. Em đã đi chụp được đi được 4 quán, trong đó 2 quán cà phê chó và 2 quán cà phê. Mỗi buổi chụp được hơn 1 nghìn tấm ảnh.
* Ngoài ra em còn chụp ảnh trực tiếp của những con chó, con mèo trên đường khi em gặp phải và trong những quán ăn nơi em ở.
* Sau mỗi buổi chụp, em tiến hành lựa chon, lọc những tấp ảnh phù hợp, ít bị mờ, loại bỏ hơn 1 nửa số ảnh đã chụp lưu trữ trên drive.
* Đảm bảo sự cân bằng giữa 2 lớp bằng cách chụp thêm hay lọc tiếp dữ liệu để tạo ra dataset hoàn chỉnh “cat\_dog”.
* Thêm đó em thu thập thêm dữ liệu từ google ảnh chó mèo, tạo ra tập dữ liệu kiểm thử [test\_cat\_dog](https://drive.google.com/drive/folders/1melOoH5HAp1qQe_70turtQP-9n3MQvDj?usp=sharing) gồm 200 ảnh (100 “dog” 100 “cat”) để đánh giá model và so sánh kết quả train model giữa 2 tập test tự thu thập và test dữ liệu từ google.
* **Một số ảnh từ bộ dữ liệu “cat\_dog”**
* **Cat**

****

* **Dog**

****

* **Một số ảnh dữ liệu từ google**

****

## **III. Tiền xử lý dữ liệu và rút trích đặc trưng**

Tiền xử lý dữ liệu:

* Tiền xử lý dữ liệu là một bước quan trọng trong bất kỳ dự án máy học nào.
* Ảnh được resize về kích thước 128x128, chuyển ảnh sang ảnh màu RGB

Trích xuất đặc trưng:

* Trong đồ án môn học, em sử dụng phương pháp rút trích đặc trưng Hog. Mã hóa hình ảnh thành một vector, và vector này mang những đặc trưng (các số thực) đại diện cho ảnh đó.
* Cách rút trích đặc trưng:

+ Tính toán gradient cho mỗi pixel trong ảnh: Gradients là sự thay đổi nhỏ theo hướng x và y.Quá trình này sẽ cung cấp cho chúng ta hai ma trận mới - một ma trận lưu trữ các gradient theo hướng x (G x) và một ma trận khác lưu trữ các gradient theo hướng y (G y).

+ Tính độ lớn và hướng:

* + - Áp dụng định lý Pythagoras để tính tổng độ lớn gradient:

Tổng độ lớn Gradient = √ [(G x) 2 + (G y) 2 ]

* + - Tính toán hướng, giá trị của góc sẽ là:

Φ = atan (Gy / Gx)

* + - Tạo biểu đồ bằng Gradients và Orientation:

+ Tính toán Biểu đồ Gradients trong các ô 8 × 8 (9 × 1): Chia hình ảnh thành các ô 8 × 8 và tạo các biểu đồ, chúng ta sẽ nhận được một ma trận 9 x 1 cho mỗi ô.

+ Chuẩn hóa gradient trong ô 16 × 16 (36 × 1):

* + - Kết hợp bốn ô 8 × 8 để tạo ra một khối 16 × 16, mỗi ô 8 × 8 có một ma trận 9 × 1 cho một biểu đồ. Vì vậy, chúng ta sẽ có bốn ma trận 9 × 1 hoặc một ma trận 36 × 1 duy nhất.
    - Chuẩn hoá ma trận 36 × 1

+ Các tính năng cho hình ảnh hoàn chỉnh:  kết hợp tất cả các khối 16 × 16 của hình ảnh.

+ Các tính năng cho hình ảnh hoàn chỉnh:  kết hợp tất cả các khối 16 × 16 của hình ảnh.

* Chúng ta sẽ có 255 (15 × 15) khối 16 × 16. Mỗi khối trong số 255 khối này có một vectơ 36 × 1 là các đối tượng địa lý. Do đó, tổng số đối tượng cho hình ảnh sẽ là 255 x 36 × 1 = 8100 đối tượng.
* Mỗi hình ảnh sẽ được mã hóa thành 1 vector 8100 (255 x 36 × 1) đặc trưng.
* Phân chia dữ liệu:

+ Em phân chia bộ dữ liệu dataset mình chụp thành 2 phần:

* 80% làm dữ liệu để train model
* 20% làm dữ liệu test để đánh giá model

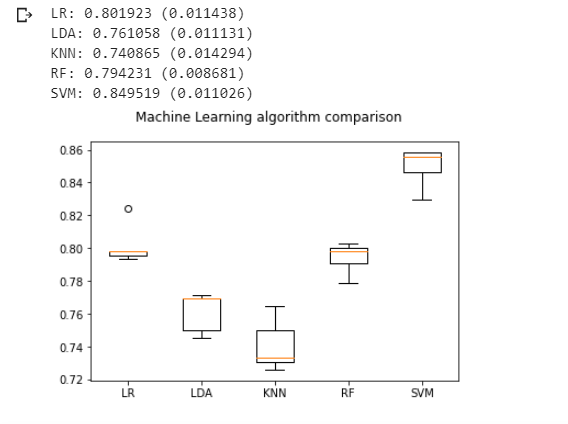
+ 200 bức ảnh cả em thu thâp trên google sử dụng để đánh giá lại model.

## **IV. Mô hình thuật toán máy học**

Trong Classifications, có nhiều thuật toán phân loại khác nhau như SVM, Decision Tree, …Trong đồ án, em chọn thuật toán Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbor, Random Forest Classifier, SVM để giải quyết bài toán.

## **V. Kết quả đánh giá model**

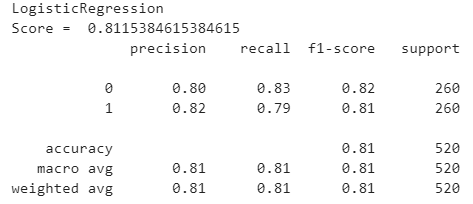
**1. Đánh giá model với K-FLod**



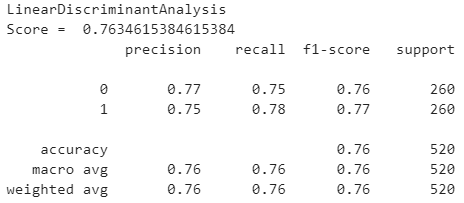
- SVM là model cho kết quả đánh giá cao nhất trong đánh giá model bằng xác thực chéo 84.95%, thấp nhất là KNN 74.1%.

**2. Đánh giá model với dữ liệu test**

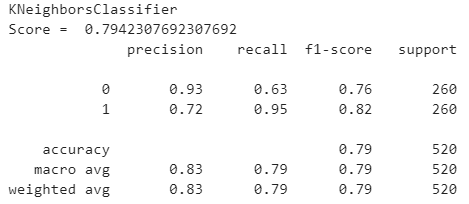
**a. Tập test từ dataset ảnh tự chụp**

\* Logistic Regressio

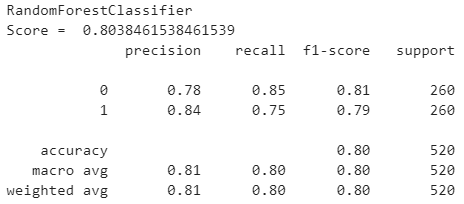
\* Linear Discriminant Analysis



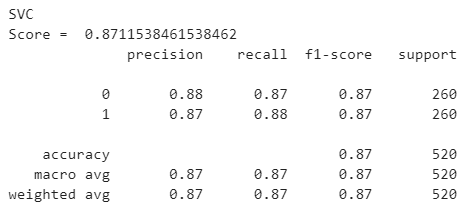
\* K-Nearest Neighbor



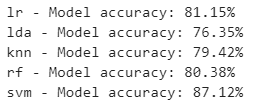
\* Random Forest Classifier



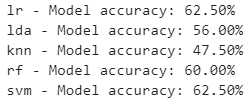
\* SVM



\* Độ chính xác của các model



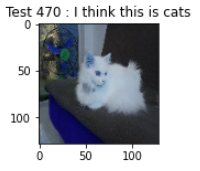
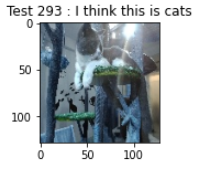
**b. Tập test ảnh từ google**



* SVM là model cho kết quả đánh giá cao nhất trong cả 2 tập test. Model có độ chính xác thấp nhất khi kiểm tra với tập test ảnh tự chụp và tập test ảnh từ google lần lượt là lda 76.35%, knn 47.5%.

## **VI. Kết luận**

Sau đây là một vài kết quả dự đoán của mô hình:



Chúng ta nhận thấy sự khác biệt giữa 2 tập dữ liệu tự chụp và thu thập từ google là: dữ liệu thu thập từ google cho độ đa dạng về giống, loài, kích thước, kiểu dáng hơn dữ liệu tự chụp. Song sáng 2 tập dữ liệu test, tập test từ dữ liệu tự chụp sát với tập train, ảnh có nhiều điểm tương đồng, còn tập test từ google cho dữ liệu không giống với tập train, ít điểm tương đồng.

       Mô hình cho kết quả khả quan khi kiểm tra với dữ liệu từ tập dataset tự chụp nhưng chưa cho kết quả tốt với tập dữ liệu kiểm tra từ google. Độ chính xác tốt nhất với tập test dataset tự chụp là 87.1% và là 62.5 với tập test lấy từ google. Những nguyên nhân dẫn đến là:

+       Bộ dữ liệu ít đa dạng.

+       Tiền xử lý dữ liệu và rút trích đặc trưng còn hạn chế về kiến thức.

+ Dataset tự chụp dẫn đến nhiều ảnh có nhiều điểm giống nhau, số lượng về loại trong từng lớp thấp, dataset giới hạn độ đa dạng về giống loài, kiểu dáng, kích thước trong từng lớp. Dataset lớn dữ liệu càng nhiều điểm tương đồng về 1 giống chó. Dẫn đến kiểm tra với đa dạng dữ liệu sẽ cho kết quả thấp.

## **VII. Hướng phát triển**

* Tiền xử lý dữ liệu tốt hơn
* Tăng độ đa dạng cho dữ liệu
* Tìm cách thức thức rút trích đặc trưng hiệu quả hơn cho model
* Có thể tiến hành giải quyết bài toán theo hướng:
* Using Convolutional Neural Network (CNN)
* Transfer Learning using Inception v3 Model