

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO ĐỒ ÁN**

**MÁY HỌC – CS114.M11**

**Đề tài: ỨNG DỤNG QUẢN LÝ HÌNH  
ẢNH TRÊN THIẾT BỊ DI ĐỘNG**

**GVHD: PGS.TS. Lê Đình Duy**

**Ths. Phạm Nguyễn Trường An**

**Sinh viên thực hiện: Nhóm TSQ**

**Nguyễn Đình Sang                      19522120**

**Phan Lê Xuân Trường                19522444**

**Lương Tường Quy                      19522108**

**Thành phố Hồ Chí Minh – 01/2022**

**NHẬN XÉT**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Với lòng biết ơn sâu sắc nhất, đầu tiên nhóm chúng em xin gửi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý Thầy Cô khoa Khoa học máy tính đã giúp cho nhóm em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành tới PGS.TS. Lê Đình Duy - giảng viên lý thuyết và Ths. Phạm Nguyễn Trường An – giảng viên thực hành môn Máy học đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn nhóm trong suốt quá trình làm đồ án môn học. Nhờ đó, chúng em đã tiếp thu được nhiều kiến thức bổ ích trong việc vận dụng cũng như kỹ năng làm đồ án. Nếu không có những lời hướng dẫn, dạy bảo của các thầy thì nhóm chúng em nghĩ đồ án này của nhóm rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn Thầy. Ngoài ra, để đồ án được hoàn thành thì không thể nào cảm ơn những người đã làm ra đó, cảm ơn các bạn các thành viên trong nhóm đã chăm chỉ và chịu khó hoàn thành nhiệm vụ đúng tiến độ.

Xuất phát từ mục đích học tập của môn học và yêu cầu của đồ án. Dựa trên những kiến thức được các Thầy cung cấp trên trường kết hợp với việc tự tìm hiểu những công cụ và kiến thức mới, nhóm đã cố gắng thực hiện đồ án một cách tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm chúng em không tránh khỏi những thiếu sót. Chính vì vậy, nhóm chúng em rất mong nhận được những sự góp ý từ phía các Thầy nhằm hoàn thiện những kiến thức mà chúng em đã học tập và là hành trang để chúng em thực hiện tiếp các đề tài khác trong tương lai.

Sau cùng, nhóm em xin kính chúc Thầy thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp là truyền đạt kiến thức cho các bạn sinh viên.

Xin chân thành cảm ơn Thầy!

**Nhóm thực hiện**

## MỤC LỤC

NHẬN XÉT .....	2
LỜI CẢM ƠN .....	3
MỤC LỤC .....	5
CHƯƠNG 1: TỔNG QUAN .....	6
1. MÔ TẢ BÀI TOÁN.....	6
2. MÔ TẢ DỮ LIỆU.....	7
CHƯƠNG 2: XÂY DỰNG BỘ DỮ LIỆU .....	8
1. THU THẬP DỮ LIỆU.....	8
2. CẤU TRÚC THƯ MỤC CHỨA DỮ LIỆU .....	8
3. THÔNG SỐ BỘ DỮ LIỆU.....	8
CHƯƠNG 3: ĐỀ XUẤT MÔ HÌNH .....	10
1. MÔ HÌNH DENSENET .....	10
2. MÔ HÌNH INCEPTION VERSION 3 (INCEPTIONV3) .....	15
CHƯƠNG 4: TRAINING VÀ ĐÁNH GIÁ MODEL .....	17
1. MÔ HÌNH DENSENET .....	17
2. MÔ HÌNH INCEPTION V3 .....	24
3. KẾT QUẢ MÔ HÌNH.....	33
4. NHẬN XÉT .....	33
CHƯƠNG 5: ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN .....	34
1. ỨNG DỤNG .....	35
2. HƯỚNG PHÁT TRIỂN.....	35
CHƯƠNG 6: KẾ HOẠCH THỰC HIỆN .....	36
1. THÔNG TIN NHÓM.....	36
2. PHƯƠNG THỨC LÀM VIỆC .....	36
3. PHÂN CÔNG NHIỆM VỤ, ĐÁNH GIÁ.....	37
CHƯƠNG 7: TÀI LIỆU THAM KHẢO .....	38

# CHƯƠNG 1: TỔNG QUAN

## 1. MÔ TẢ BÀI TOÁN

Trong thế giới công nghệ phát triển liên tục như ngày nay thì việc sở hữu riêng cho bản thân một chiếc điện thoại thông minh với nhiều tính năng hữu ích là một điều vô cùng dễ dàng. Trong đó phổ biến và ưu chuộng nhất là tính năng chụp ảnh, với một chiếc điện thoại di động ai ai cũng có thể chụp và lưu trữ cho mình rất nhiều hình ảnh trong cuộc sống hằng ngày từ các cuộc du lịch với gia đình bạn bè, đến các buổi tiệc tùng, hẹn hò; các hoá đơn mua sắm, selfie, các tài liệu giấy tờ,...

Việc chụp nhiều hình ảnh đòi hỏi mỗi khi muốn lục tìm lại một tấm ảnh nào đó khá mất thời gian vì phải duyệt tuần tự, tìm lần lượt từng hình ảnh này sang hình ảnh khác.

Vì vậy, nhóm chúng em muốn tạo ra một ứng dụng tự động phân loại các hình ảnh. Ứng dụng này sẽ phân chia hình ảnh đã được lưu trữ thành các thư mục (hoặc album) một cách dễ dàng và thuận tiện để giúp tất cả mọi người sử dụng điện thoại di động có thể tìm kiếm hình ảnh nhanh hơn.

- ❖ Input của bài toán: Một hình ảnh được lưu trữ trong điện thoại di động.
- ❖ Output của bài toán: Class (loại) của hình ảnh (Ở bài toán này sẽ có 21 loại)
  - 21 loại ảnh được phân loại đó là:
    - Beach (Biển)
    - Bill (Hoá đơn)
    - Bookjacket (Bìa sách)
    - Car (Ô tô)

- Cat (Con mèo)
- Clothes (Quần áo)
- Dog (Con chó)
- Drink (Nước uống)
- Feast (Tiệc tùng)
- Field (Cánh đồng)
- Flower (Hoa)
- Food (Thức ăn)
- Fruits (Trái cây)
- House (Nhà cửa)
- Jewels (Trang sức)
- Motorcycle (Mô tô, xe gắn máy)
- Mountain (Núi đồi)
- Papers (Tài liệu, giấy tờ)
- Person (Ảnh người)
- Shoe (Giày dép)
- Truck (Xe tải)

## 2. MÔ TẢ DỮ LIỆU

Dữ liệu của bài toán được thu thập từ nhiều nguồn ảnh khác nhau: tự chụp, chụp màn hình, ảnh từ Google images và ảnh từ bộ dữ liệu có sẵn Kaggle.

Các hình ảnh đa dạng và phong phú về nội dung và đối tượng để phù hợp với yêu cầu của bài toán.

## CHƯƠNG 2: XÂY DỰNG BỘ DỮ LIỆU

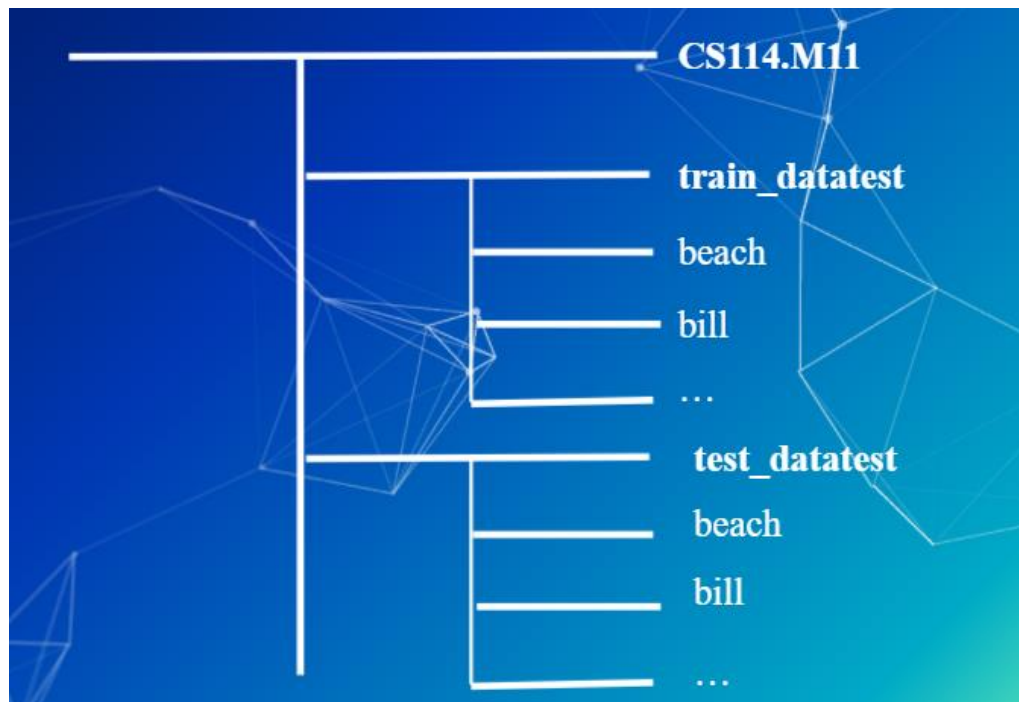
### 1. THU THẬP DỮ LIỆU

Để mô phỏng cho ứng dụng phân loại thực tế, nhóm quyết định chia hình ảnh thành 21 loại cơ bản nhất. Sau đó tiến hành thu thập dữ liệu cho mỗi class.

Dữ liệu thu thập từ camera điện thoại, ảnh chụp màn hình điện thoại.

Ảnh từ Google Images và bộ dữ liệu có sẵn Kaggle.

### 2. CẤU TRÚC THƯ MỤC CHỨA DỮ LIỆU



### 3. THÔNG SỐ BỘ DỮ LIỆU

Hình ảnh thuộc ba định dạng sau: JPG, JPEG và PNG

Nhóm đã thu thập được tất cả 4536 hình ảnh, trong đó:

- Tập train có 3936 ảnh
- Tập test có 600 ảnh



Liệt kê chi tiết số lượng dữ liệu thu thập được cho bộ dữ liệu training:

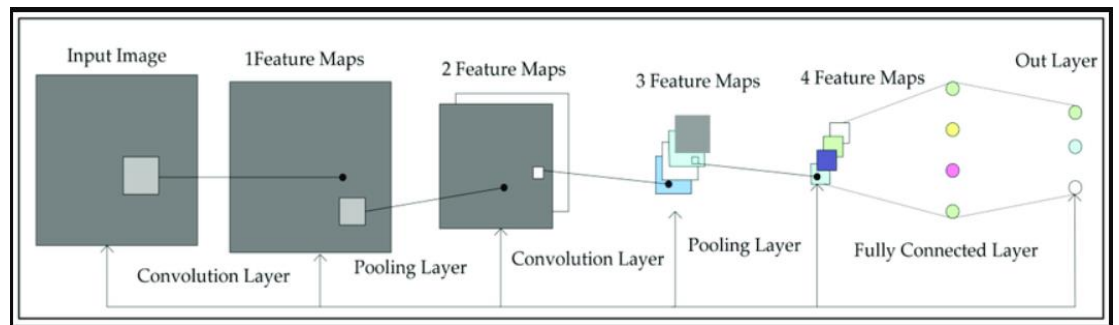
- Total training fruits images: 157
- Total training cat images: 316
- Total training dog images: 293
- Total training beach images: 287
- Total training bill images: 108
- Total training bookjacket images: 217
- Total training car images: 114
- Total training clothes images: 188
- Total training drink images: 167
- Total training feast images: 110
- Total training field images: 126
- Total training flower images: 276
- Total training food images: 218
- Total training house images: 111
- Total training jewels images: 117
- Total training motorcycle images: 211
- Total training mountain images: 107
- Total training papers images: 125
- Total training person images: 184
- Total training shoe images: 355
- Total training truck images: 149

## CHƯƠNG 3: ĐỀ XUẤT MÔ HÌNH

### 1. MÔ HÌNH DENSENET

❖ Giới thiệu mô hình:

DenseNet<sup>121</sup>: Trong một mạng thần kinh convolutional feed-forward truyền thống (CNN), mỗi lớp phức tạp ngoại trừ lớp đầu tiên (lấy đầu vào), nhận đầu ra của lớp phức tạp trước đó và tạo ra một bản đồ tính năng đầu ra sau đó được chuyển sang lớp phức tạp tiếp theo. Do đó, đối với các lớp 'L', có các kết nối trực tiếp 'L'; một giữa mỗi lớp và lớp tiếp theo.



Tuy nhiên, khi số lượng lớp trong CNN tăng lên, tức là khi chúng sâu hơn, vấn đề 'gradient biến mất' phát sinh. Điều này có nghĩa là khi đường dẫn thông tin từ đầu vào đến các lớp đầu ra tăng lên, nó có thể khiến một số thông tin nhất định 'biến mất' hoặc bị mất làm giảm khả năng đào tạo hiệu quả của mạng.

DenseNet giải quyết vấn đề này bằng cách sửa đổi kiến trúc CNN tiêu chuẩn và đơn giản hóa mô hình kết nối giữa các lớp. Trong kiến trúc DenseNet, mỗi lớp được kết nối trực tiếp với mọi lớp khác, do đó có tên Densely Connected Convolutional Network. Đối với các lớp 'L', có kết nối trực tiếp  $L(L+1)/2$ .

❖ Lý do chọn mô hình:

Một số ưu điểm của DenseNet121:

- Accuracy : Densenet training tham số ít hơn 1 nửa so với Resnet nhưng có same accuracy so trên ImageNet classification dataset.
- Overfitting : DenseNet resistance overfitting rất hiệu quả.
- Giảm được vashing gradient.
- Sử dụng lại feature hiệu quả hơn.

❖ Cấu trúc mô hình:

Các thành phần của DenseNet121 bao gồm:

- Kết nối
- Các block dày đặc
- Tốc độ tăng trưởng
- Các lớp tắc nghẽn

### Kết nối

Trong mỗi lớp, bản đồ tính năng của tất cả các lớp trước đó không được tóm tắt, mà được ghép nối và sử dụng làm đầu vào. Do đó, DenseNet yêu cầu ít tham số hơn so với CNN truyền thống tương đương và điều này cho phép tái sử dụng tính năng vì bản đồ tính năng dư thừa bị loại bỏ. Vì vậy,  $l$ th lớp nhận bản đồ tính năng của tất cả các lớp trước,  $x_0, \dots, x_{l-1}$ , như đầu vào:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]).$$

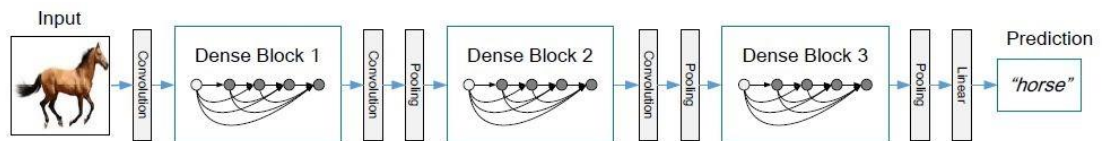
nơi  $[x_0, x_1, \dots, x_{l-1}]$  là sự kết hợp của các bản đồ tính năng, tức là đầu ra được tạo ra trong tất cả các lớp trước  $l$  ( $0, \dots, l-1$ ). Nhiều đầu vào của  $H_l$  được kết hợp thành một tensor duy nhất để dễ dàng thực hiện.

### Các block dày đặc

Việc sử dụng thao tác kết hợp là không khả thi khi kích thước của bản đồ tính năng thay đổi. Tuy nhiên, một phần thiết yếu của CNN là việc lấy mẫu xuống của các lớp làm giảm kích thước của bản đồ tính năng thông qua giảm kích thước để đạt được tốc độ tính toán cao hơn.

Để kích hoạt điều này, DenseNets được chia thành DenseBlock, trong đó kích thước của bản đồ tính năng vẫn không đổi trong một khối, nhưng số lượng bộ lọc giữa chúng được thay đổi. Các lớp giữa các khối được gọi là Lớp chuyển tiếp làm giảm số lượng kênh xuống còn một nửa so với các kênh hiện có.

Đối với mỗi lớp, từ phương trình trên, H1 Được định nghĩa là một hàm tổng hợp áp dụng ba hoạt động liên tiếp: chuẩn hóa hàng loạt (BN), một đơn vị tuyến tính được sửa chữa (ReLU) và một convolution (Conv).



Trong hình trên, một DenseNet sâu với ba khối dày đặc được hiển thị. Các lớp giữa hai khối liên tiếp là các lớp chuyển tiếp thực hiện downsampling (tức là thay đổi kích thước của các bản đồ tính năng) thông qua các hoạt động phức tạp và gộp, trong khi trong khối dày đặc kích thước của bản đồ tính năng là như nhau để cho phép ghép các tính năng.

### Tốc độ tăng trưởng

Người ta có thể nghĩ về các tính năng như một trạng thái toàn cầu của mạng. Kích thước của bản đồ tính năng phát triển sau khi đi qua mỗi lớp dày đặc với mỗi lớp thêm các tính năng 'K' lên trên trạng thái toàn cầu

(các tính năng hiện có). Tham số này 'K' được gọi là tốc độ tăng trưởng của mạng, điều chỉnh lượng thông tin được thêm vào trong mỗi lớp của mạng. Nếu mỗi hàm H l tạo ra bản đồ tính năng k, sau đó lth lớp có

$$k_l = k_0 + k * (l - 1)$$

nhập bản đồ tính năng, trong đó k0 là số lượng kênh trong lớp đầu vào.

Không giống như các kiến trúc mạng hiện có, DenseNet có thể có các lớp rất hẹp.

### Các lớp tắc nghẽn

Mặc dù mỗi lớp chỉ tạo ra bản đồ tính năng đầu ra k, số lượng đầu vào có thể khá cao, đặc biệt là đối với các lớp tiếp theo. Do đó, một lớp phức tạp 1x1 có thể được giới thiệu như một lớp nút cổ chai trước mỗi phức tạp 3x3 để cải thiện hiệu quả và tốc độ tính toán.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Một bản tóm tắt các kiến trúc khác nhau được thực hiện cho cơ sở dữ liệu ImageNet đã được cung cấp trong bảng trên. Stride là số lượng pixel thay

đổi qua ma trận đầu vào. Một bước tiến của 'n' (giá trị mặc định là 1), chỉ ra rằng các bộ lọc được di chuyển 'n' pixel tại một thời điểm.

Sử dụng kiến trúc DenseNet-121 để hiểu bảng, chúng ta có thể thấy rằng mỗi khối dày đặc có số lớp (lặp lại) khác nhau có hai phức tạp mỗi lớp; một hạt nhân có kích thước 1x1 làm lớp nút cổ chai và nhân 3x3 để thực hiện thao tác phức tạp.

Ngoài ra, mỗi lớp chuyển tiếp có một lớp phức tạp 1x1 và một lớp gộp trung bình 2x2 với sải chân là 2. Như vậy, các lớp có mặt như sau:

1. Lớp phức tạp cơ bản với 64 bộ lọc có kích thước 7X7 và sải chân 2
2. Lớp pooling cơ bản với pooling tối đa 3x3 và sải chân 2
3. Khối 1 dày đặc với 2 phức tạp lặp lại 6 lần
4. Lớp chuyển tiếp 1 (1 Conv + 1 AvgPool)
5. Khối 2 dày đặc với 2 phức tạp lặp lại 12 lần
6. Lớp chuyển tiếp 2 (1 Conv + 1 AvgPool)
7. Khối 3 dày đặc với 2 phức tạp lặp lại 24 lần
8. Lớp chuyển tiếp 3 (1 Conv + 1 AvgPool)
9. Khối 4 dày đặc với 2 phức tạp lặp lại 16 lần
10. Lớp Pooling trung bình toàn cầu - chấp nhận tất cả các bản đồ tính năng của mạng để thực hiện phân loại
11. Lớp đầu ra

Do đó, DenseNet-121 có các lớp sau:

- 1 7x7 Convolution

- 58 3x3 Convolution
- 61 1x1 Convolution
- 4 AvgPool
- 1 Lớp kết nối đầy đủ

Tóm lại, DenseNet-121 có 120 convolutions và 4 AvgPool.

Tất cả các lớp tức là những lớp trong cùng một khối dày đặc và các lớp chuyển tiếp, trải trọng lượng của chúng trên nhiều đầu vào cho phép các lớp sâu hơn sử dụng các tính năng được chiết xuất sớm.

Vì các lớp chuyển tiếp xuất ra nhiều tính năng dư thừa, các lớp trong khối dày đặc thứ hai và thứ ba gán trọng lượng ít nhất cho đầu ra của các lớp chuyển tiếp.

Ngoài ra, mặc dù trọng lượng của toàn bộ khối dày đặc được sử dụng bởi các lớp cuối cùng, vẫn có thể có nhiều tính năng cấp cao hơn được tạo sâu hơn vào mô hình vì dường như có nồng độ cao hơn đối với các bản đồ tính năng cuối cùng trong các thí nghiệm.

## 2. MÔ HÌNH INCEPTION VERSION 3 (INCEPTIONV3)

### ❖ Giới thiệu mô hình:

Authors: Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Google, University College London

Inception-V3 là kế thừa của Inception-V1 bao gồm 24 triệu tham số.

Toàn bộ các layer tích chập của Inception-V3 được theo sau bởi một layer batch normalization và một ReLU activation. Batch normalization là kỹ thuật chuẩn hóa đầu vào theo từng minibatch tại mỗi layer theo phân phối chuẩn hóa  $N(0,1)$ , giúp cho quá trình huấn luyện thuật toán nhanh hơn.

❖ Lý do chọn mô hình:

Inception-V3 giải quyết được vấn đề thắt cổ chai (representational bottlenecks). Tức là kích thước của các layers không bị giảm một cách đột ngột. Đồng thời Inception-V3 có một cách tính toán hiệu quả hơn nhờ sử dụng phương pháp nhân tố (factorisation methods).

❖ Cấu trúc mô hình:

Hiện tại Inception module bao gồm 4 version. Chúng ta hãy cùng xem qua các điểm đặc biệt ở từng version.

- Inception-A: Cải tiến so với Inception module V1. Tại nhánh thứ nhất thay 1 layer tích chập  $5 \times 5$  bằng 2 layer tích chập  $3 \times 3$  liên tiếp giúp giảm số lượng tham số từ 25 về 18 và tăng độ sâu cho mô hình.
- Inception-B: Cải tiến so với Inception-A. Thay tích chập  $3 \times 3$  bằng tích chập  $7 \times 7$  ở nhánh thứ nhất và nhánh thứ 2. Đồng thời chúng ta phân tích nhân tố tích chập  $7 \times 7$  thành 2 tích chập liên tiếp  $7 \times 1$  và  $1 \times 7$  số lượng tham số sẽ ít hơn so với tích chập  $2 \times 2$  tích chập  $3 \times 3$  liên tiếp. Nhờ đó số lượng tham số giảm từ 18 xuống còn 14.
- Inception-C: Cải tiến so với Inception-B. Thay tích chập  $7 \times 1$  bằng tích chập  $3 \times 1$  và  $1 \times 7$  bằng  $1 \times 3$  và đồng thời thay vì đặt layer  $3 \times 1$  và  $1 \times 3$  liên tiếp thì đặt chúng song song. Kiến trúc này giúp giảm số lượng tham số từ 14 về còn 6.

Ngoài ra ở Inception-V3 chúng ta còn sử dụng 2 kiến trúc giảm chiều dữ liệu là Reduction-A và Reduction-B.



## CHƯƠNG 4: TRAINING VÀ ĐÁNH GIÁ MODEL

### 1. MÔ HÌNH DENSENET

DenseNet121: Sử dụng model training data với kích thước ảnh (224,224)

- Thuật toán: adam
- batch\_size=128
- verbose=2
- callback=[anne,checkpoint]

Tham số thuật toán adam lần lượt là: learning\_rate=0.001, beta\_1=0.9, beta\_2=0.999, epsilon=1e-07, amsgrad=False

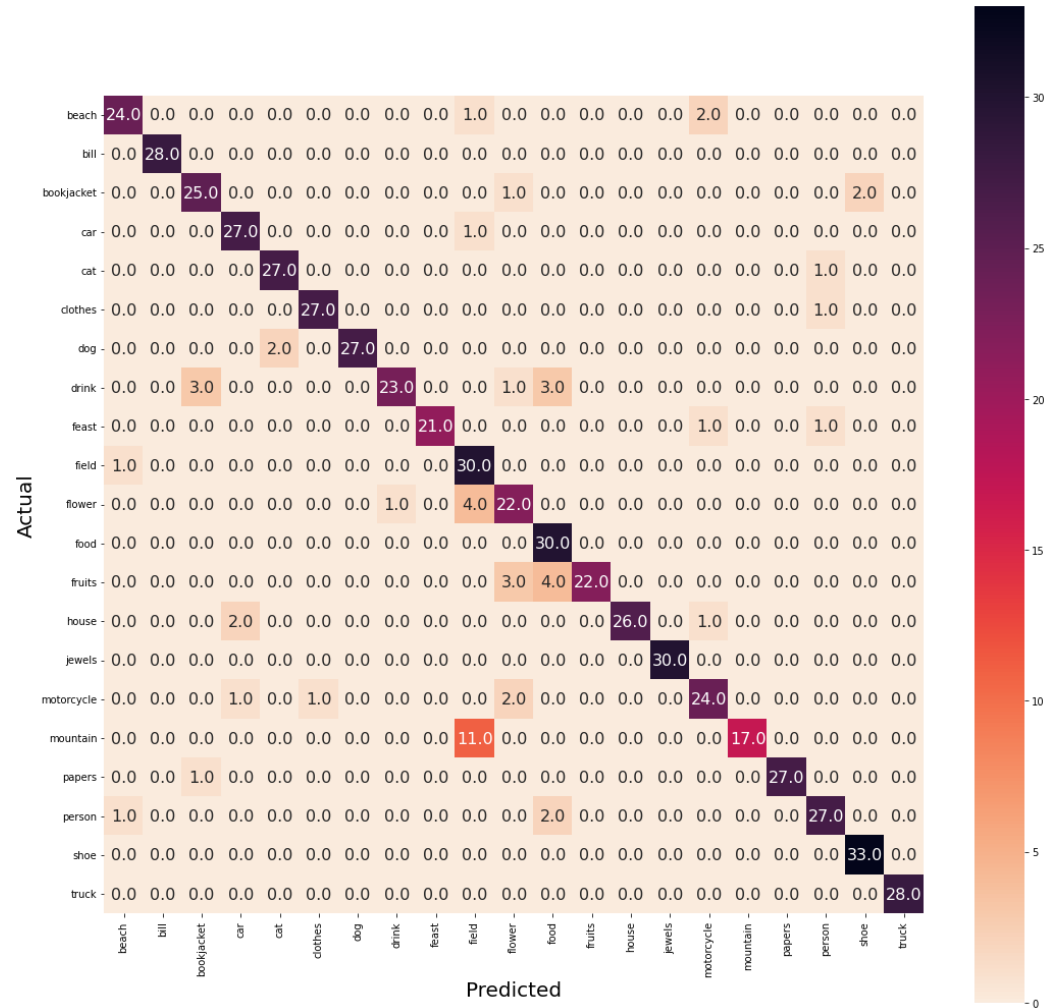
❖ **epochs = 10**

Total-test-data: 600

accurately-predicted-data: 545

wrongly-predicted-data: 55

accuracy = 90.8333%



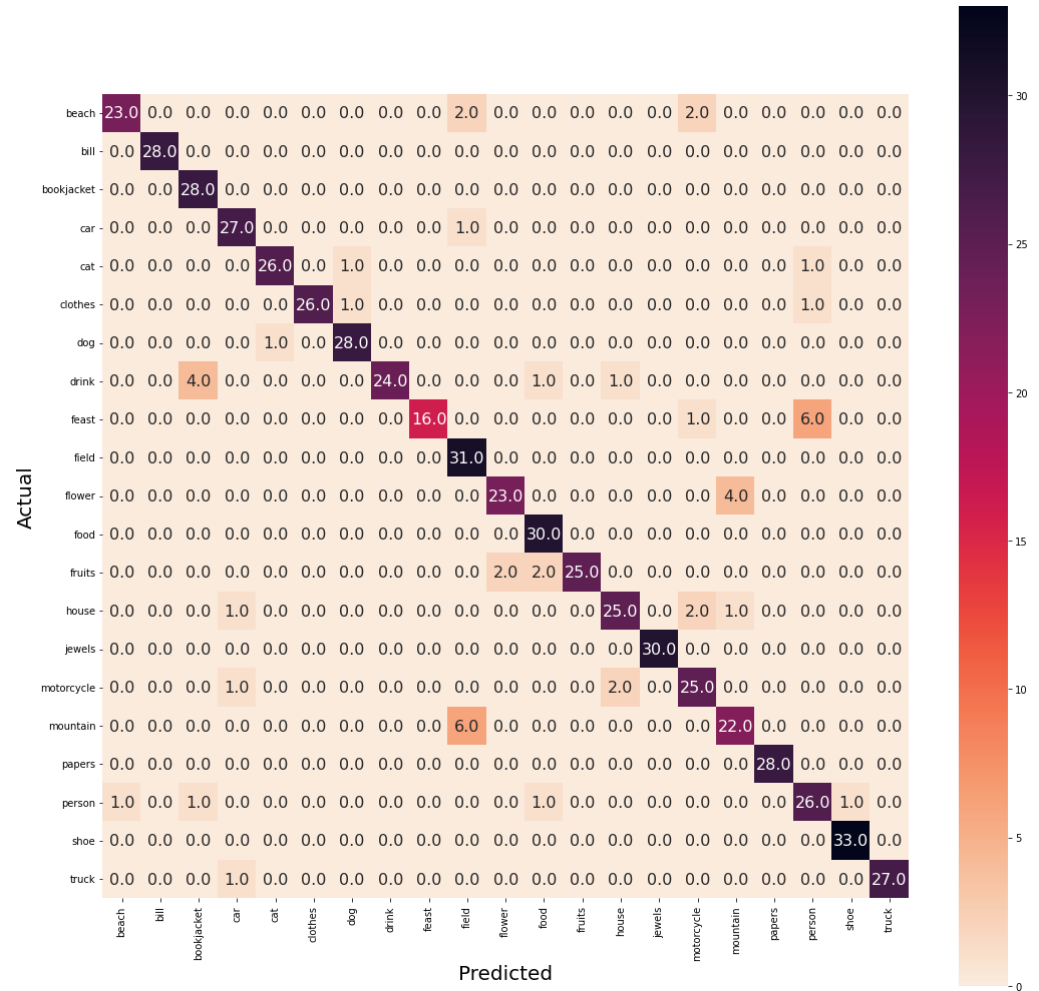
❖ epochs=15

Total-test-data: 600

accurately-predicted-data: 551

wrongly-predicted-data: 49

accuracy = 91.8333%



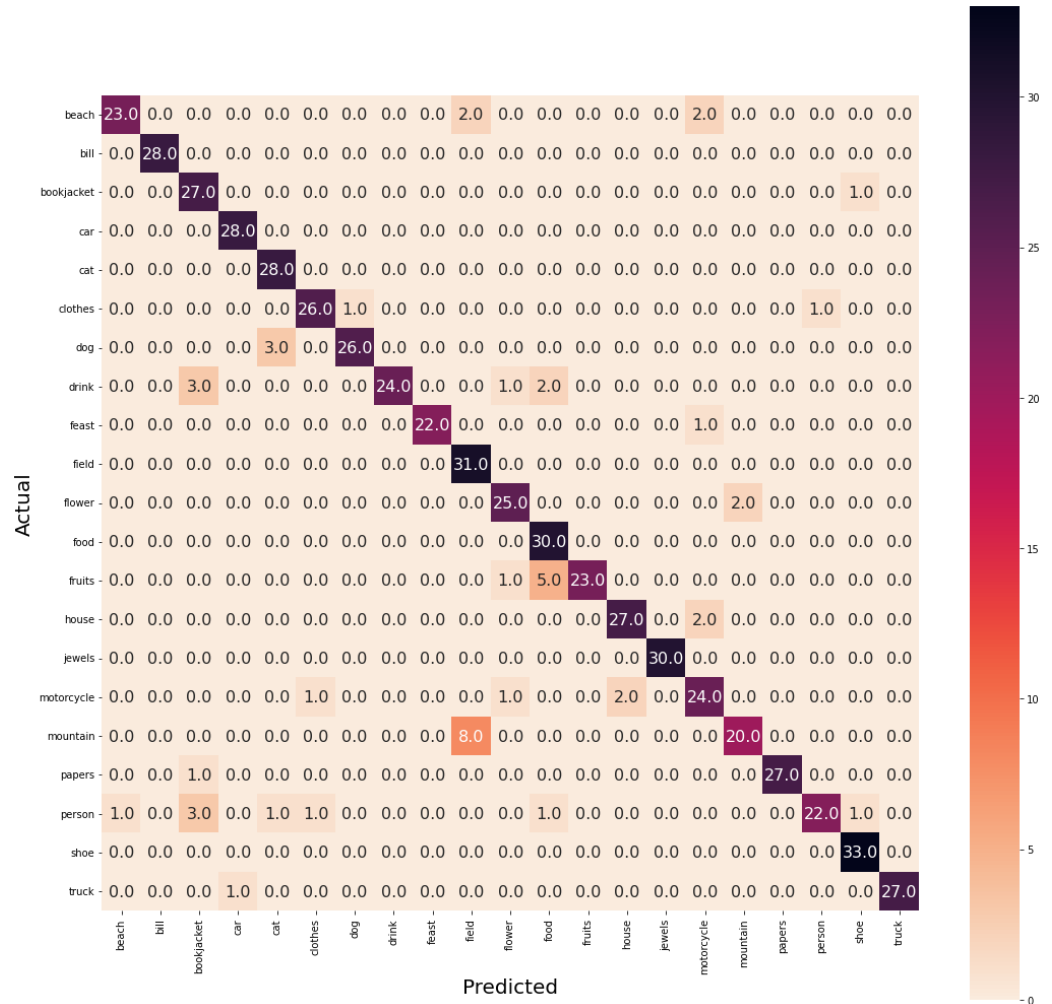
❖ epochs=50

Total-test-data: 600

accurately-predicted-data: 551

wrongly-predicted-data: 49

accuracy = 91.8333%



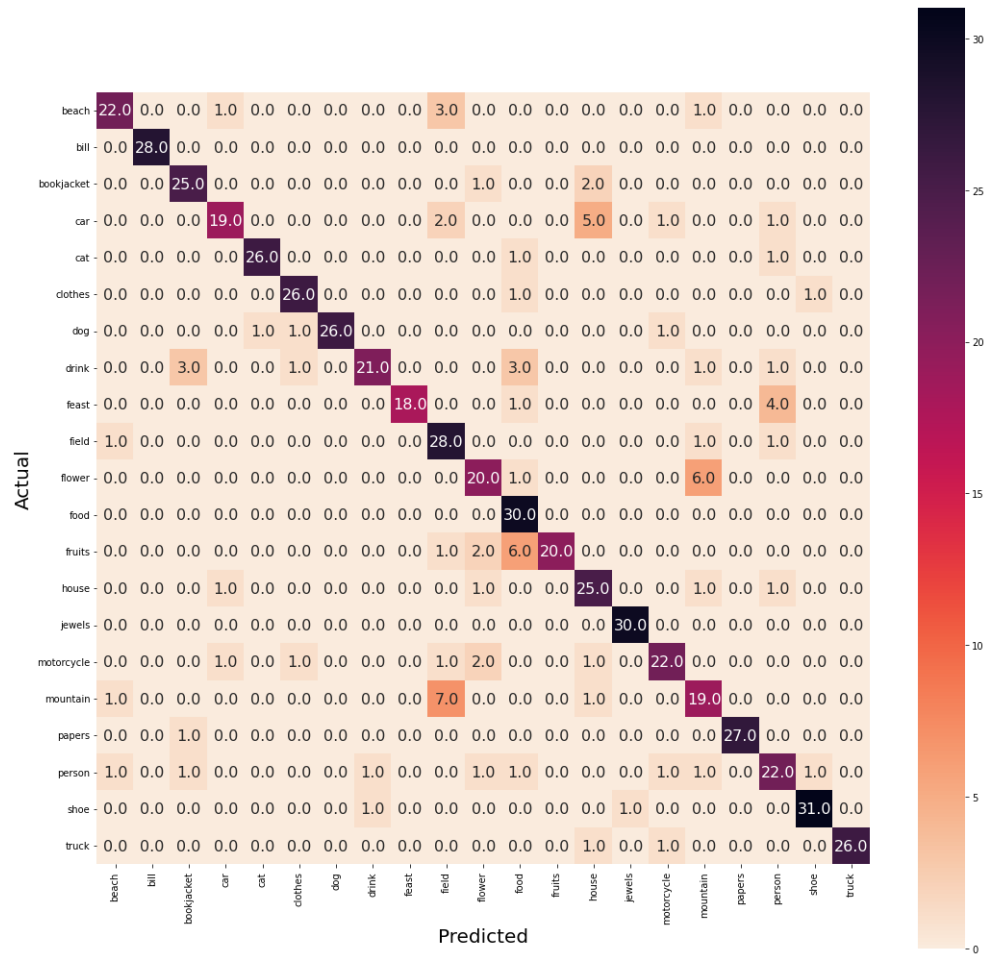
❖ **Training với các kích thước khác nhau:**

Số lượng epochs = 10

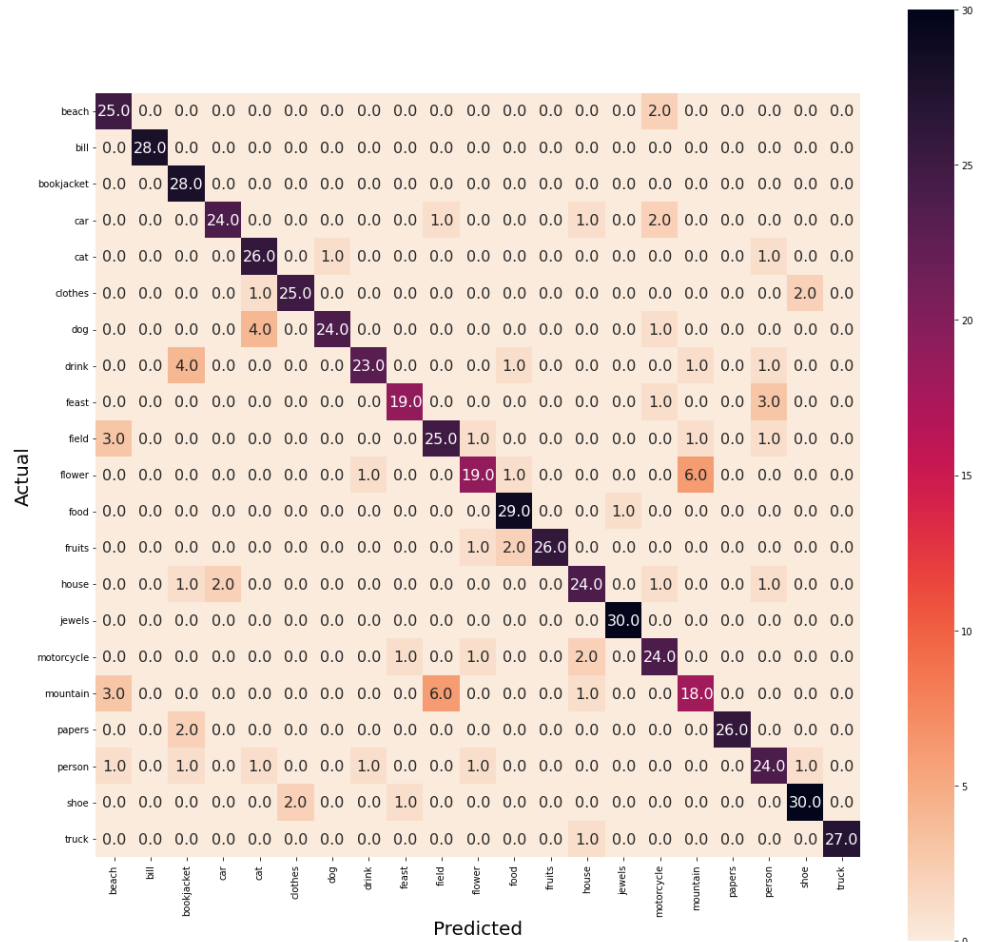
Các kích thước ảnh lần lượt là **124x124, 150x150, 224x224, 255x255**

Size	Total test data	Accurately predicted data	Wrongly predicted data	Accuracy
124x124	600	503	97	83.833%
150x150	600	524	76	87.33%
224x224	600	545	55	90.83%
255x255	600	545	55	90.83%

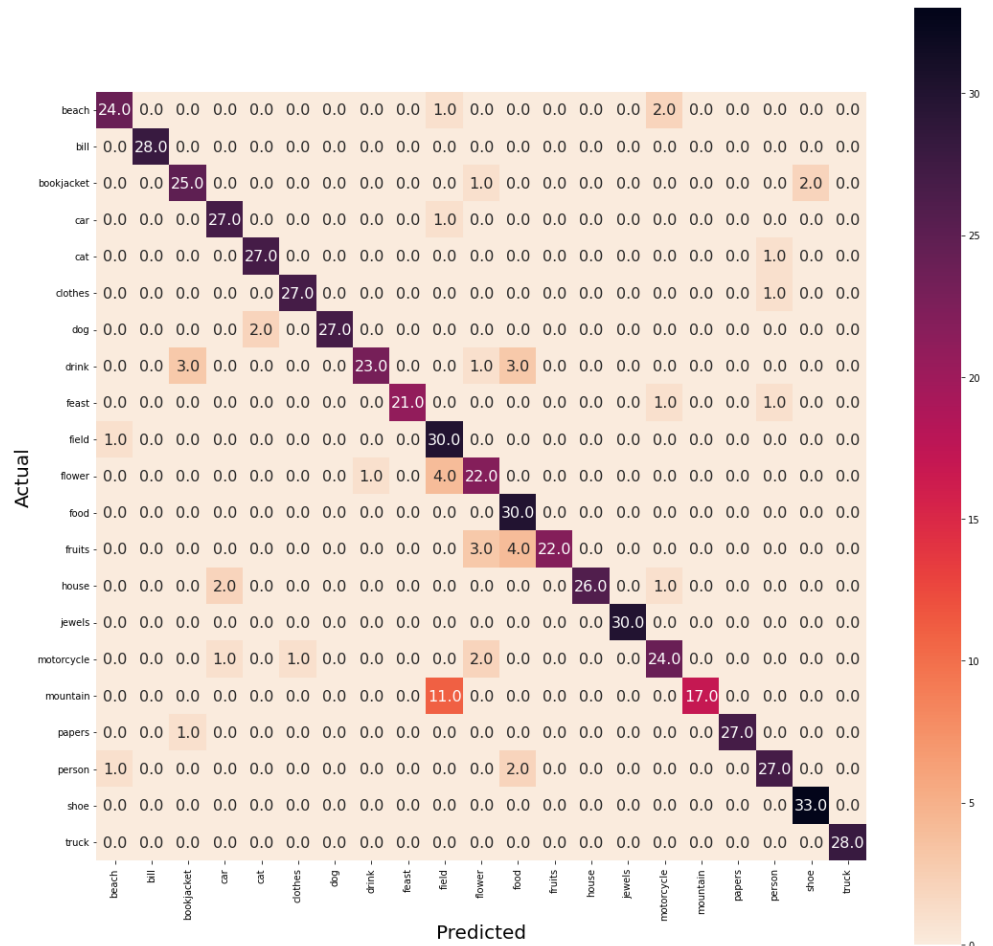
- Kích thước ảnh 124x124



- Kích thước ảnh 150x150



- Kích thước ảnh 224x224



## 2. MÔ HÌNH INCEPTION V3

Sử dụng tensorflow.keras.applications.InceptionV3

Thiết lập các tham số:

- batch\_size: 150
- optimizer: adam
- Hàm loss: CategoricalCrossentropy
- Learning rate: 0.001



## ❖ Training với số lượng epochs khác nhau

Size ảnh khi được training là 224\*224

### ➤ epochs = 20

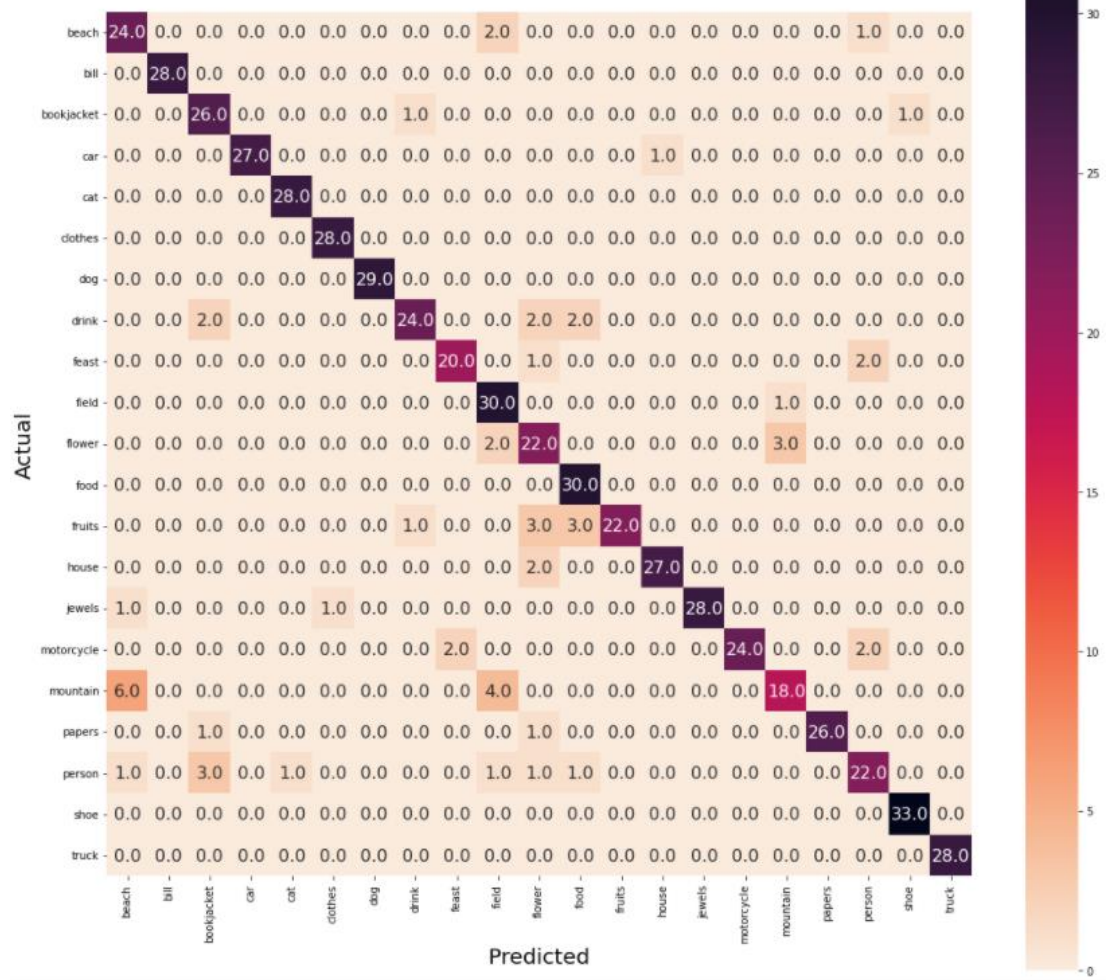
Total-test-data: 600

accurately-predicted-data: 544

wrongly-predicted-data: 56

Accuracy: 90.667 %

	precision	recall	f1-score	support
0	0.75	0.89	0.81	27
1	1.00	1.00	1.00	28
2	0.81	0.93	0.87	28
3	1.00	0.96	0.98	28
4	0.97	1.00	0.98	28
5	0.97	1.00	0.98	28
6	1.00	1.00	1.00	29
7	0.92	0.80	0.86	30
8	0.91	0.87	0.89	23
9	0.77	0.97	0.86	31
10	0.69	0.81	0.75	27
11	0.83	1.00	0.91	30
12	1.00	0.76	0.86	29
13	0.96	0.93	0.95	29
14	1.00	0.93	0.97	30
15	1.00	0.86	0.92	28
16	0.82	0.64	0.72	28
17	1.00	0.93	0.96	28
18	0.81	0.73	0.77	30
19	0.97	1.00	0.99	33
20	1.00	1.00	1.00	28
accuracy			0.91	600
macro avg	0.91	0.91	0.91	600
weighted avg	0.91	0.91	0.91	600



➤ **epochs = 50**

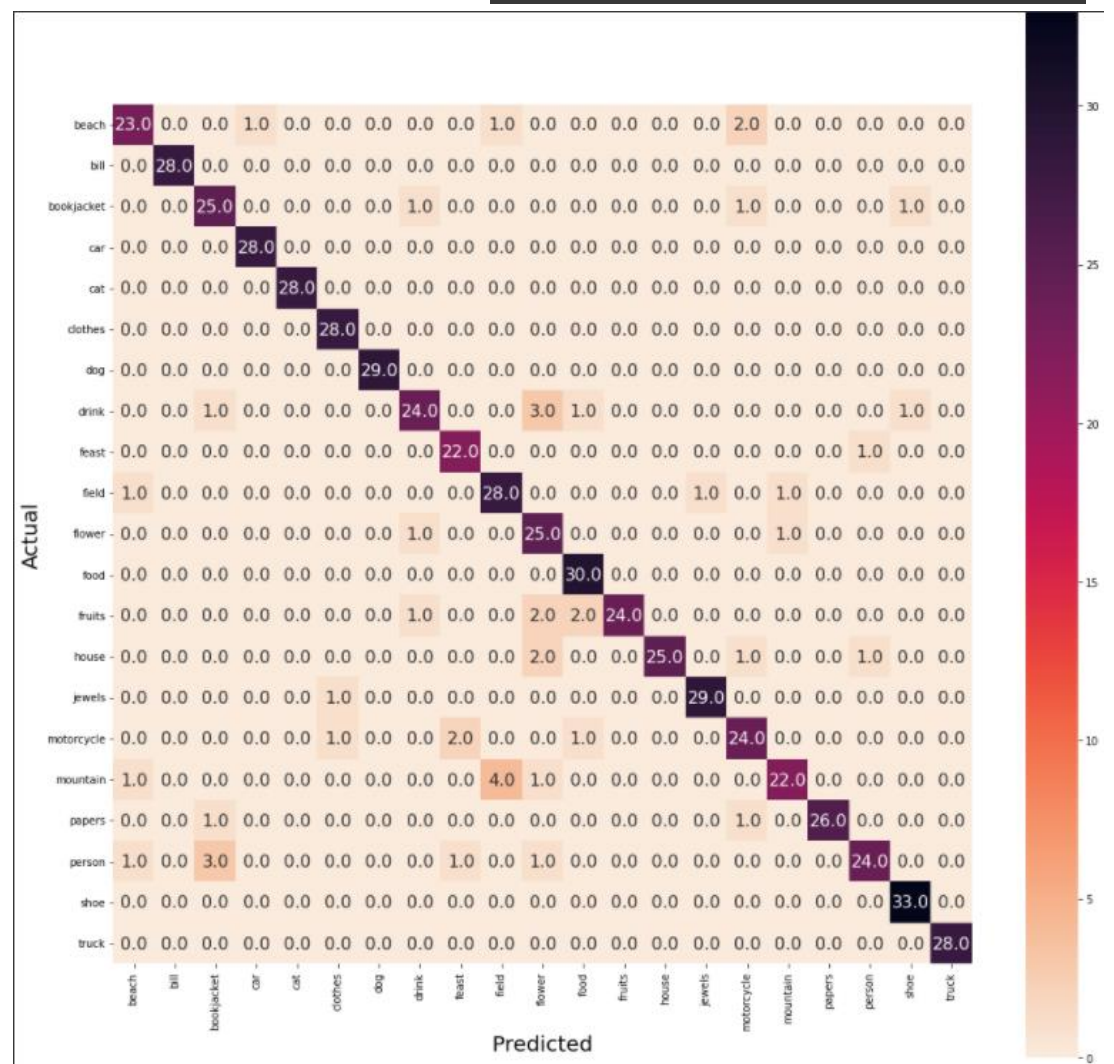
Total-test-data; 600

accurately-predicted-data: 553

wrongly-predicted-data: 47

Accuracy: 92.167 %

	precision	recall	f1-score	support
0	0.88	0.85	0.87	27
1	1.00	1.00	1.00	28
2	0.83	0.89	0.86	28
3	0.97	1.00	0.98	28
4	1.00	1.00	1.00	28
5	0.93	1.00	0.97	28
6	1.00	1.00	1.00	29
7	0.89	0.80	0.84	30
8	0.88	0.96	0.92	23
9	0.85	0.90	0.88	31
10	0.74	0.93	0.82	27
11	0.88	1.00	0.94	30
12	1.00	0.83	0.91	29
13	1.00	0.86	0.93	29
14	0.97	0.97	0.97	30
15	0.83	0.86	0.84	28
16	0.92	0.79	0.85	28
17	1.00	0.93	0.96	28
18	0.92	0.80	0.86	30
19	0.94	1.00	0.97	33
20	1.00	1.00	1.00	28
accuracy			0.92	600
macro avg	0.93	0.92	0.92	600
weighted avg	0.93	0.92	0.92	600



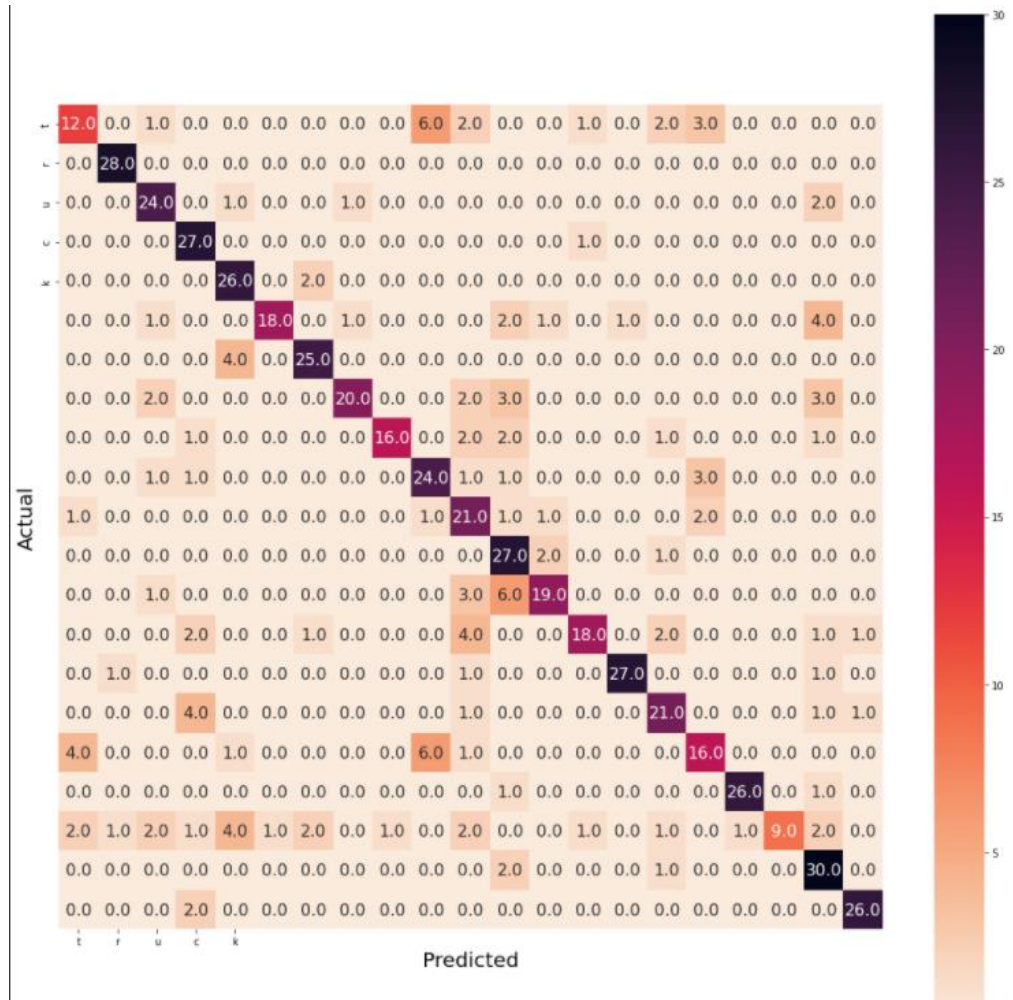
❖ **Training với các kích thước ảnh khác nhau**

Số lượng epochs = 10

Các kích thước ảnh lần lượt là **124x124, 150x150, 224x224, 255x255**

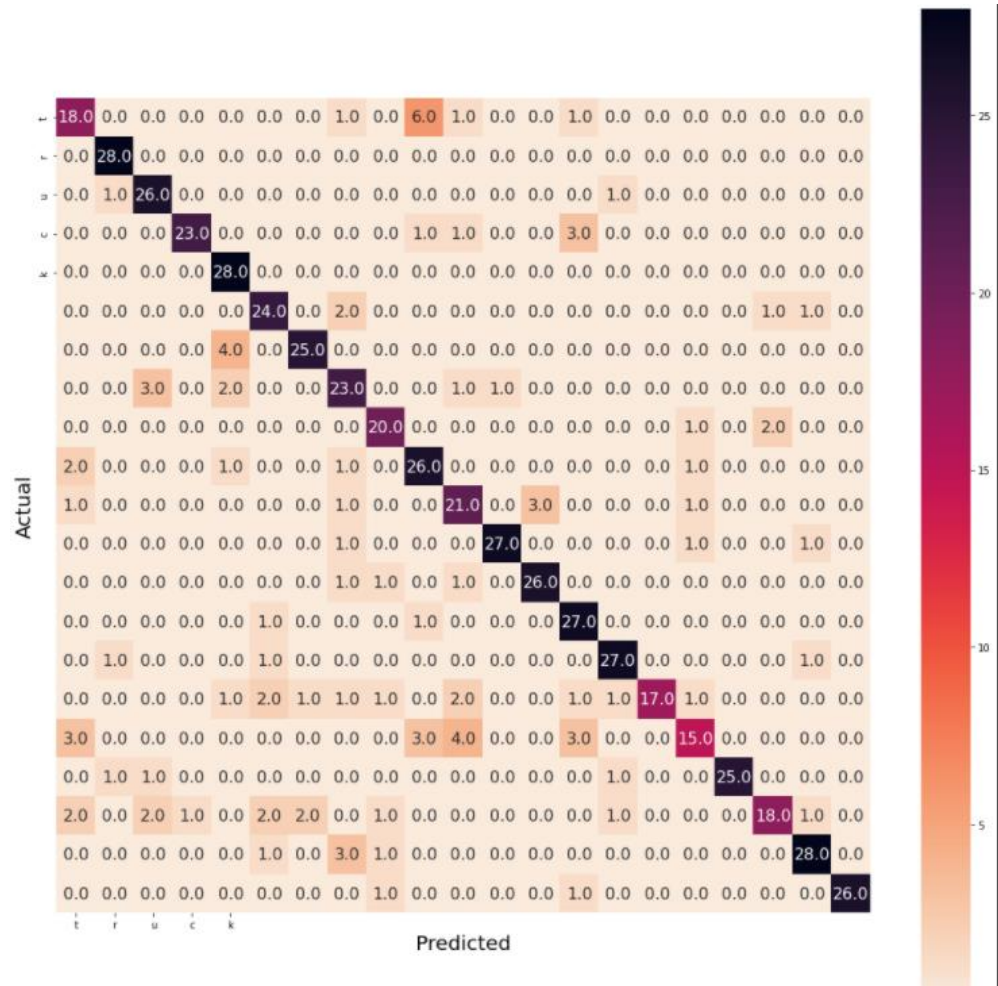
Size	Total test data	Accurately predicted data	Wrongly predicted data	Accuracy
124x124	600	460	140	76.7%
150x150	600	498	102	83.3%
224x224	600	539	61	89.8%
255x255	600	556	44	92.7%

- Kích thước ảnh 124x124

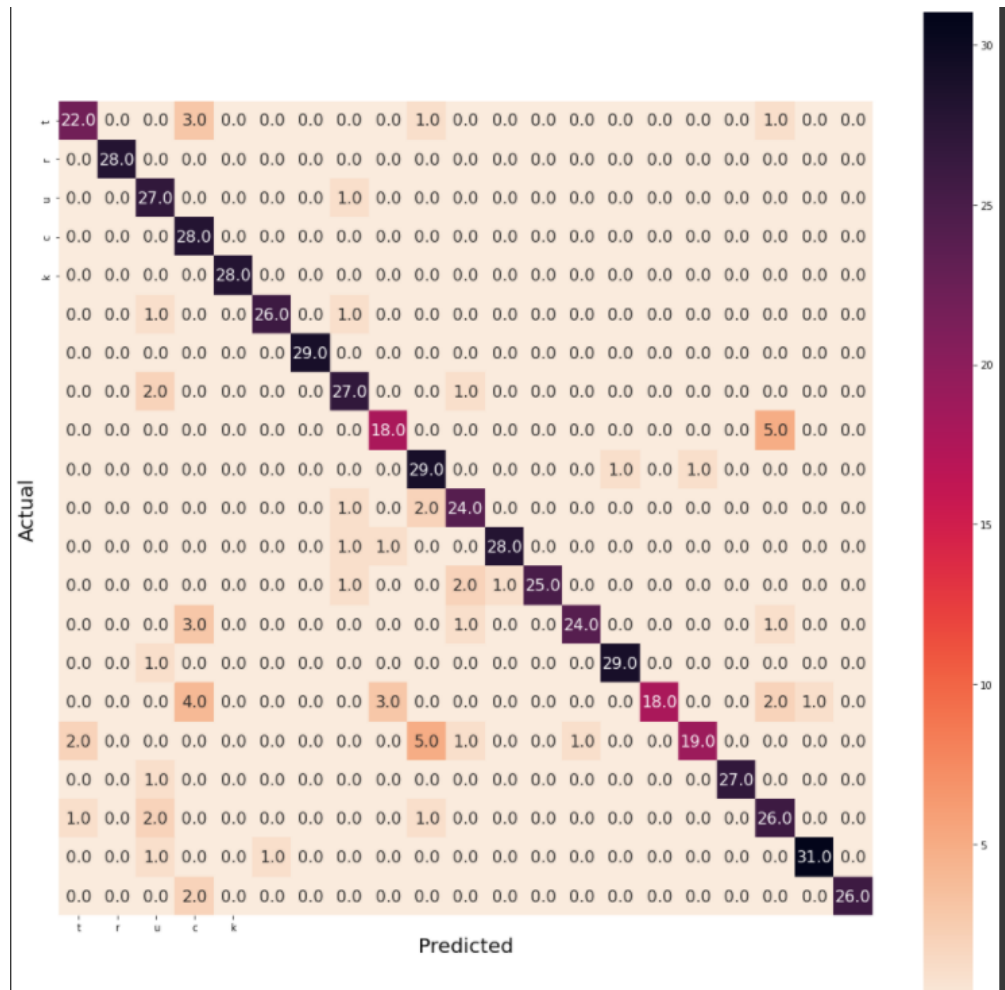




- Kích thước ảnh 150x150



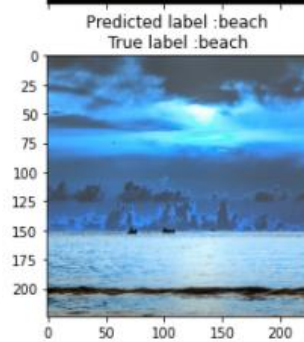
- Kích thước ảnh 224x224



-

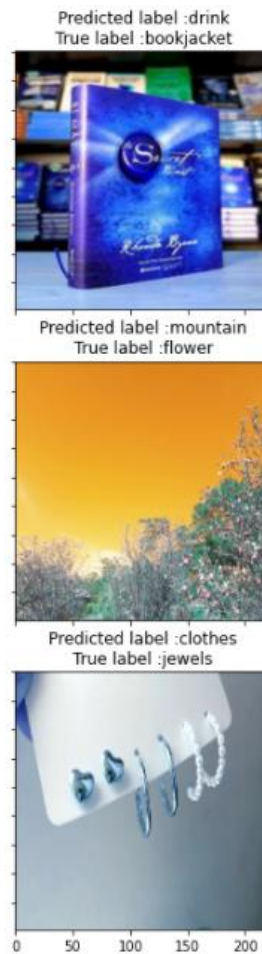
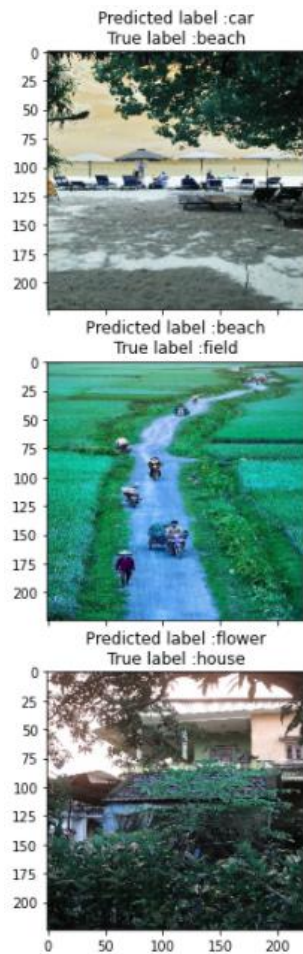
❖ Một số kết quả thu được khi test

- Một số trường hợp đúng:





- Một số trường hợp bị sai:



### 3. KẾT QUẢ MÔ HÌNH

Accuracy khi test trên 500 ảnh

	epochs	InceptionV3	DenseNet121
1	10	0.9050	0.9083
2	15	0.9050	0.9183
3	20	0.9067	0.9333
4	50	0.9100	0.9183

#### 4. NHẬN XÉT

- Cả 2 mô hình cho kết quả tương đối khả quan
- Với các kích thước ảnh càng lớn khi training thì accuracy cho kết quả càng cao.
- Mô hình DenseNet121 cho ra kết quả luôn cao hơn so với InceptionV3
- Một số class gây nhầm lẫn cho nhau như:
  - Cảnh đồng với đồi núi(ở DenseNet sai nhiều nhất là 11 mức trong số 28 bức ảnh test)
  - Hoa với núi(sai nhiều nhất 6 bức trong 25 bức ảnh test)
  - Trang sức với người(do có nhiều bức hình trang sức có chứa người trong đó)

## CHƯƠNG 5: ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN

### 1. ỨNG DỤNG

Như đã nêu ở trên, mục đích ứng dụng của mô hình này nhằm hướng đến nhu cầu lưu trữ và tìm kiếm ảnh nhanh chóng, dễ dàng cho người dùng điện thoại di động. Hình ảnh được lưu trữ trong điện thoại sẽ tự động được lưu vào một album riêng, nơi chứa những hình ảnh cùng một loại. Tuy nhiên, việc phân loại ảnh cần độ chính xác cao nên model cần cải tiến rất nhiều về tốc độ xử lý và khả năng xử lý chính xác (phân loại nhiều đối tượng trên hình,...)

### 2. HƯỚNG PHÁT TRIỂN

Dữ liệu:

- Cần nhiều dữ liệu và nhiều class (loại) dữ liệu hơn nữa giúp cho model trở nên thực tế, hữu ích, đa dạng hơn với người dùng.
- Cải thiện môi trường thu thập dữ liệu sát với thực tế.

Model:

- Thử nghiệm một số model khác cũng như nghiên cứu thêm thông tin về các thông số ảnh hưởng thế nào đến các bộ dữ liệu khác nhau nhằm đưa ra cách điều chỉnh phù hợp.

## CHƯƠNG 6: KẾ HOẠCH THỰC HIỆN

### 1. THÔNG TIN NHÓM

MSSV	Họ tên	Email	Vai trò
<b>19522444</b>	Phan Lê Xuân Trường	<a href="mailto:19522444@gm.uit.edu.vn">19522444@gm.uit.edu.vn</a>	Trưởng nhóm
<b>19522120</b>	Nguyễn Đình Sang	<a href="mailto:19522120@gm.uit.edu.vn">19522120@gm.uit.edu.vn</a>	Thành viên
<b>19522108</b>	Lương Tường Quy	<a href="mailto:19522108@gm.uit.edu.vn">19522108@gm.uit.edu.vn</a>	Thành viên

### 2. PHƯƠNG THỨC LÀM VIỆC

#### 1. Quy trình

- Thống nhất đề tài đồ án
- Phân công nhiệm vụ cho từng thành viên
- Tìm kiếm tài liệu, lên kế hoạch cho các mốc thời gian cho đồ án
- Tiến hành các buổi họp nhóm, trao đổi thông tin, quy trình, bàn bạc kế hoạch thực hiện, phân công nhiệm vụ cho mỗi thành viên và ra hạn deadline
- Hoàn thành đồ án, cho ra sản phẩm hoàn chỉnh
- Đánh giá và thử nghiệm sản phẩm, hoàn thành báo cáo đồ án

#### 2. Công cụ

- Google Colab
- Google Drive
- Group FaceBook và Skype
- Office 365
- Github

### 3. PHÂN CÔNG NHIỆM VỤ, ĐÁNH GIÁ

Họ tên	MSSV	Mức độ hoàn thành	Công việc được giao
Phan Lê Xuân Trường	19522444	100%	Thu thập data Tìm hiểu, lựa chọn model (InceptionV3, DenseNet121) Training model InceptionV3 Viết báo cáo
Nguyễn Đình Sang	19522120	100%	Thu thập data Tìm hiểu model DenseNet121 Training model DenseNet121 Viết báo cáo
Lương Tường Quy	19522108	100%	Thu thập data Chạy thử nghiệm Tìm những bức ảnh gây nhiễu So sánh giữa hai model Làm slide báo cáo Viết báo cáo

## CHƯƠNG 7: TÀI LIỆU THAM KHẢO

- G. Huang, Z. Liu, van, và Weinberger, Kilian Q, "Densely Connected Convolutional Networks," arXiv.org, 2016. [arxiv.org/abs/1608.06993](https://arxiv.org/abs/1608.06993).
- Keras API reference. Link: [Keras Applications](#)
- Introduction to DenseNet with TensorFlow. Pluralsight. Link: [Introduction to DenseNet with TensorFlow | Pluralsight](#)
- `tf.keras.Model`. TensorFlow. Link: [tf.keras.Model | TensorFlow Core v2.8.0](#)