ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Môn học: Nhập môn học máy

Báo cáo đồ án thực hành 2: Classification

Nhóm sinh viên
Nguyễn Huy Hải – 18120023
Phạm Công Minh – 18120058
Nguyễn Thanh Tùng – 18120104
Lê Minh Đức – 18120164
Trần Đại Tài – 18120543

Contents

1 Phân công công việc của mỗi thành viên

- **Nguyễn Huy Hải - 18120023**
  - Tìm hiểu mô hình InceptionV3.
  - Xây dựng mô hình InceptionV3.

  Đánh giá: 100%
- **Phạm Công Minh - 18120058**
  - Tìm hiểu mô hình ResNet.
  - Xây dựng mô hình ResNet.

  Đánh giá: 100%
- **Nguyễn Thanh Tùng - 18120104**
  - Tổng hợp các mô hình.
  - Huấn luyện các mô hình và đánh giá.
  - Viết báo cáo

  Đánh giá: 100%
- **Lê Minh Đức - 18120164**
  - Khám phá tập train và test.
  - Khám phá hình ảnh.
  - Tìm hiểu Data Augmentation và áp dụng.

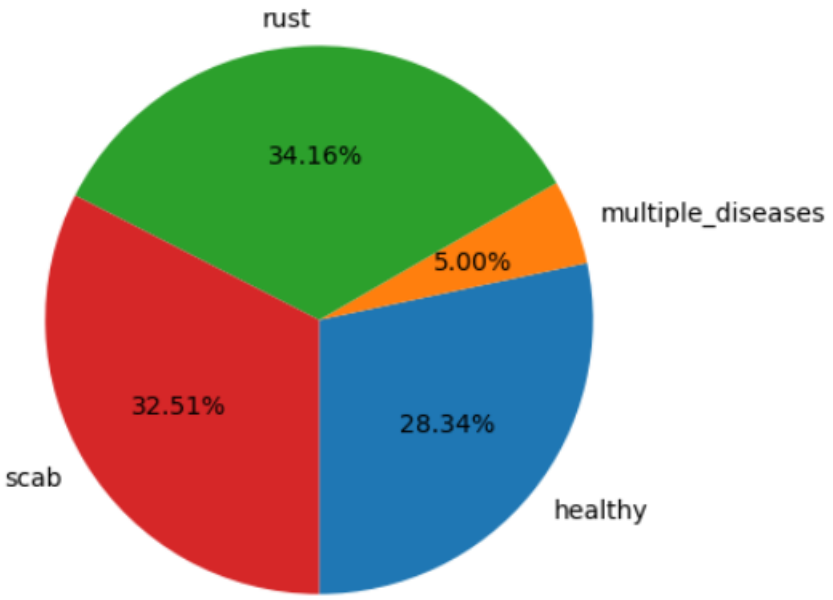
  Đánh giá: 100%
- **Trần Đại Tài - 18120543**
  - Tìm hiểu mô hình DenseNet.
  - Xây dựng mô hình DenseNet.

  Đánh giá: 100%

2 Phân tích tập dữ liệu.

2.1 Khám phá tập train và test.

- Tập train.
  - Dữ liệu có 1821 dòng, 5 cột. Trong đó:
    + Cột **image_id**: id của ảnh
    + Cột **healthy**: lá khỏe mạnh bình thường (1/0).
    + Cột **multiple_diseases**: lá có nhiều bệnh (1/0).
    + Cột **rust**: lá bị úa vàng (1/0).
    + Cột **scab**: lá bị sần sùi (1/0).
  - Nhận xét:
    * Các lá chỉ thuộc một lớp trong bốn lớp (healthy, multiple_diseases, rust, scab).
    * Sự phân bố của các lớp:

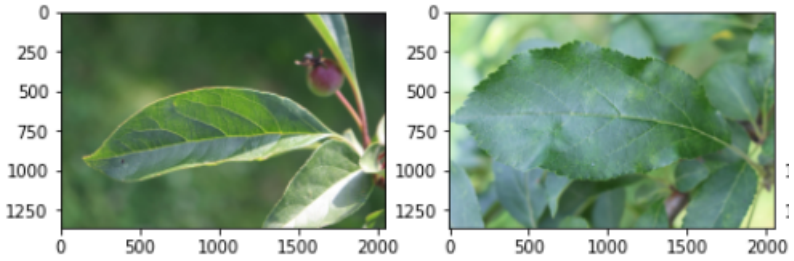


Ba lớp scab, healthy, rust có tỉ lệ gần như nhau. Chỉ có lớp multiple_diseases chiếm tỉ lệ rất thấp (chỉ 5%) điều này có thể ảnh hưởng đến việc dự đoán của lớp này.

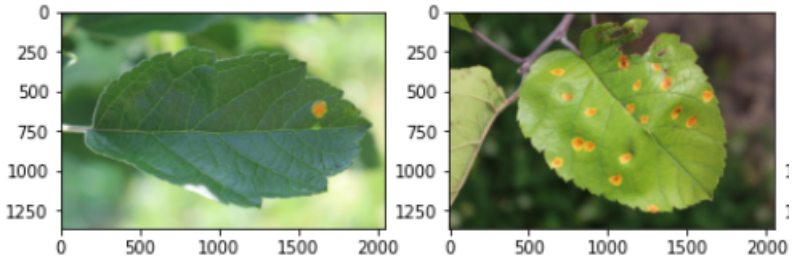
- Tập test.
  - Dữ liệu có 1821 dòng, 1 cột. Trong đó:
    + Cột **image_id**: id của ảnh

2.2 Khám phá tập ảnh.

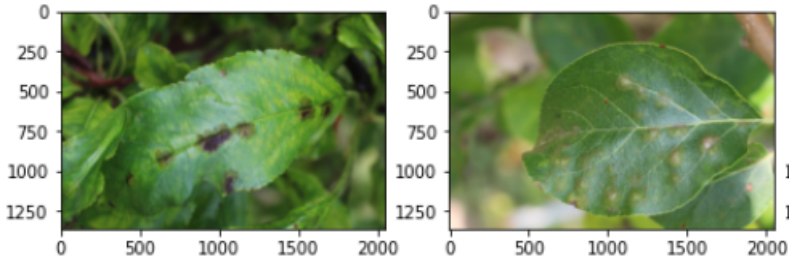
- Tất cả các ảnh đều có kích thước 2048x1365.
- Vì là tập ảnh về lá cây nên màu chính là màu xanh lá.
- Một số hình ảnh của từng lớp:
  + Lớp **healthy**:



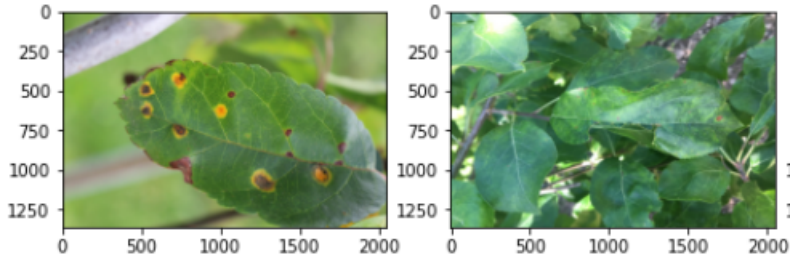
  Nhận xét: Lá cây không có dấu hiệu gì bất thường.
  + Lớp **rust**:



  Nhận xét: Lá cây bị úa vàng ở nhiều chỗ.
  + Lớp **scab**:



  Nhận xét: Lá cây bị sần sùi.
  + Lớp **multiple_diseases**:



  Nhận xét: Lá cây bị những tình trạng trên.

3 Trình bày các mô hình được cài đặt trong bài toán.

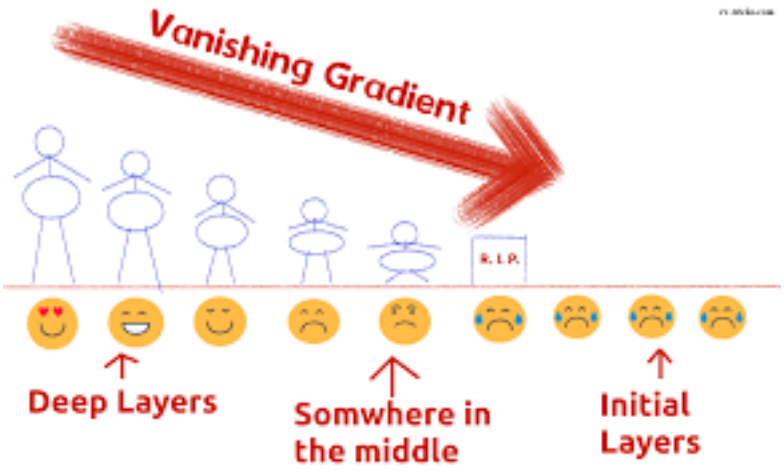
3.1 Mô hình Resnet.

a) Giới thiệu
ResNet (Residual Network) là một mạng CNN được giới thiệu đến công chúng vào năm 2015 và thậm chí đã giành được vị trí thứ 1 trong cuộc thi ILSVRC 2015 với tỉ lệ lỗi top 5 chỉ 3.57%.

Hiện tại thì có rất nhiều biến thể của kiến trúc ResNet với số lớp khác nhau như ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152,...Với tên là ResNet theo sau là một số chỉ kiến trúc ResNet với số lớp nhất định.

b) Đặc điểm của mô hình.

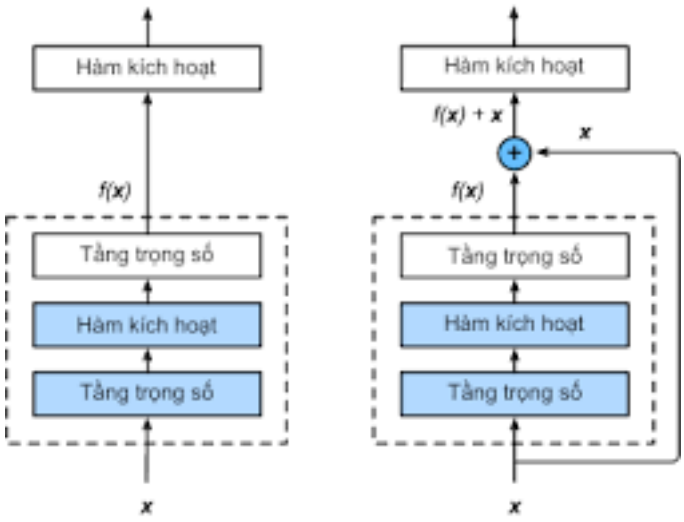
Một vấn đề xảy ra khi xây dựng mạng CNN với nhiều lớp chập sẽ xảy ra hiện tượng Vanishing Gradient. Vì độ dốc được truyền ngược trở lại các lớp trước đó, phép nhân lặp đi lặp lại có thể làm cho độ dốc cực nhỏ. Kết quả là, hiệu suất của mạng bị bão hòa hoặc giảm hiệu quả nhanh chóng dẫn tới quá trình học tập không tốt.



Giải pháp mà ResNet đưa ra là sử dụng kết nối "tắt" đồng nhất để xuyên qua một hay nhiều lớp. Một khối như vậy được gọi là một **Residual Block**.

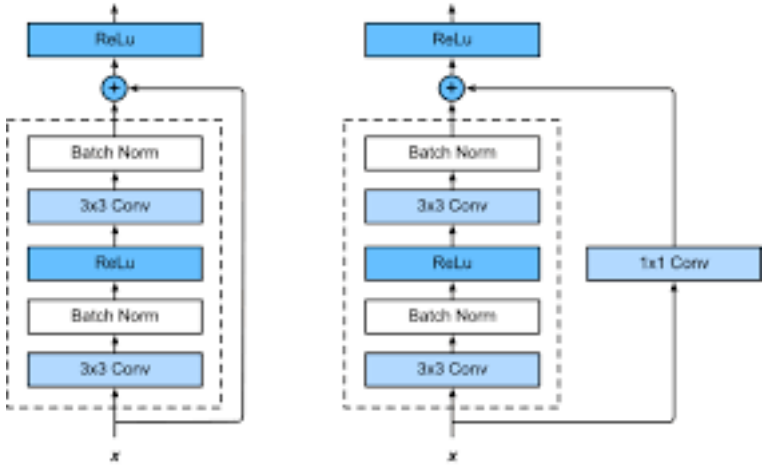
Ký hiệu đầu vào là x. Giả sử ánh xạ lý tưởng muốn học được là f(x), và được dùng làm đầu vào của hàm kích hoạt. Phần nằm trong viền nét đứt bên trái phải khớp trực tiếp với ánh xạ f(x). Điều này có thể không đơn giản nếu chúng ta không cần khối đó và muốn giữ lại đầu vào x. Khi đó, phần nằm trong viền nét đứt bên phải chỉ cần tham số

hoá độ lệch khỏi giá trị x , bởi vì ta đã trả về $x + f(x)$. Trên thực tế, ánh xạ phần dư thường dễ tối ưu hơn, vì chỉ cần đặt $f(x) = 0$.



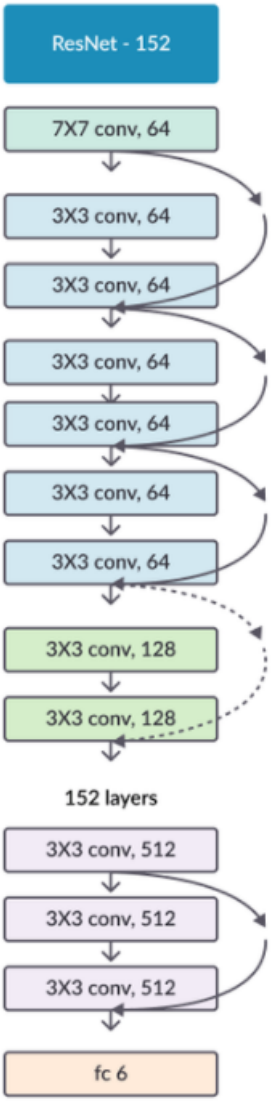
c) Cấu trúc của mô hình.

- Mô hình chung của ResNet:
  ResNet gần như tương tự với các mạng đều gồm có convolution, pooling, activation và fully-connected layer. ResNet có thiết kế tầng tích chập 3×3 giống VGG. Khối phần dư có hai tầng tích chập 3×3 với cùng số kênh đầu ra. Mỗi tầng tích chập được theo sau bởi một tầng chuẩn hóa theo batch và một hàm kích hoạt ReLU. Ta đưa đầu vào qua khối phần dư rồi cộng với chính nó trước hàm kích hoạt ReLU cuối cùng.



- Mô hình ResNet-152 V2: là một biến thể của kiến trúc ResNet nên vẫn tuân theo kiến trúc chung của ResNet
  Mô hình ResNet-152 V2 là mô hình có tổng số tầng là 152 (số tầng trong mỗi mô-đun + tầng tích chập đầu tiên + tầng kết nối đầy

đủ cuối cùng) Ảnh bên phải hiển thị khối dư được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư ResNet-152 V2 sử dụng các kết nối tắt (kết nối trực tiếp đầu vào của lớp (n) với (n+x) được hiển thị dạng mũi tên cong. Qua mô hình nó chứng minh được có thể cải thiện hiệu suất trong quá trình training model khi mô hình có nhiều lớp. Giải pháp ResNet đưa ra đơn giản hơn và tập trung vào cải thiện thông tin thông qua độ dốc của mạng



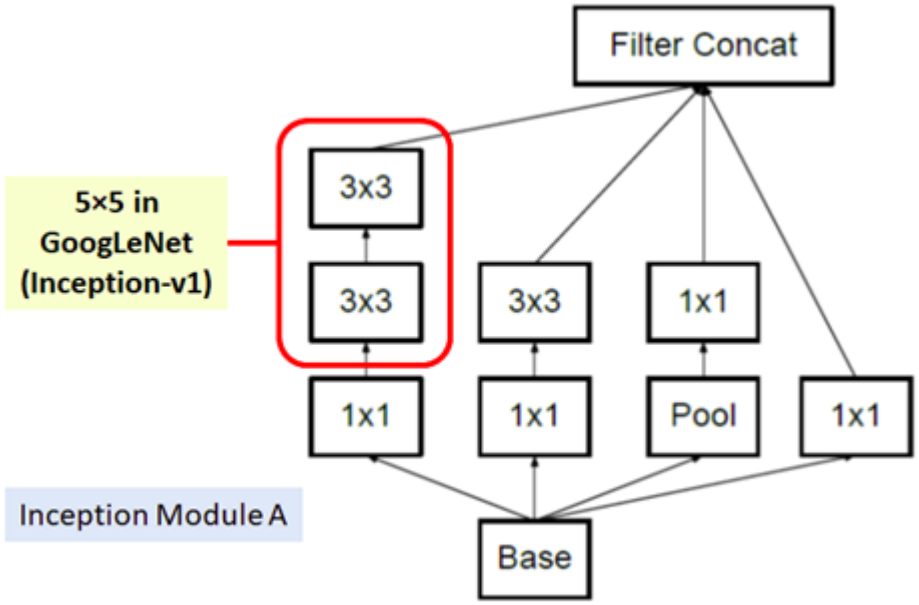
3.2 Mô hình InceptionV3.

a) Giới thiệu.
Inception-v3 là một mô hình nhận dạng hình ảnh được sử dụng rộng rãi đã được chứng minh là đạt độ chính xác hơn 78,1% trên tập dữ liệu ImageNet. Nó dựa trên bài báo gốc: "Rethinking the Inception Architecture for Computer Vision" của Szegedy.

b) Đặc điểm của mô hình.
Mô hình được tạo thành từ các block đối xứng và bất đối xứng, bao gồm các lớp convolutions, average pooling, max pooling, concats, dropouts, and fully connected. Batchnorm (Chuẩn hóa hàng loạt) được sử dụng rộng rãi trong toàn bộ mô hình và được áp dụng cho các đầu vào kích hoạt (activation inputs). Hàm mất mát được tính qua hàm Softmax.

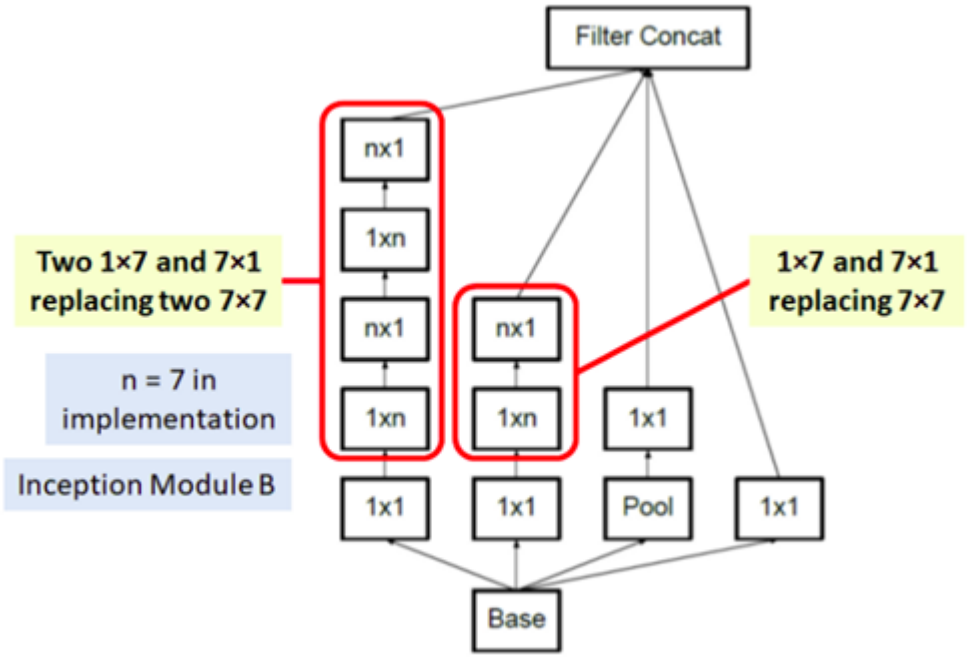
Mô hình có 3 phần chính:

- Factorizing Convolutions:
  Mục đích chính của Factorizing Convolutions là để giảm số lượng kết nối / tham số (connections/parameters) mà không làm giảm độ hiệu quả của mạng.
  - Factorizing to Smaller Convolutions.
    Hai lớp 3x3 convolutions có thể thay thể một lớp 5x5 convolutions.
    + Nếu dùng 1 lớp 5x5, số tham số cần thiết là 5*5 = 25
    + Nếu dùng 2 lớp 3x3, số tham số cần thiết là 2*3*3 = 18
    Ta đã giảm số tham số đi 28%.
    Từ đó ta có một Inception module mới (Gọi là Inception module A).



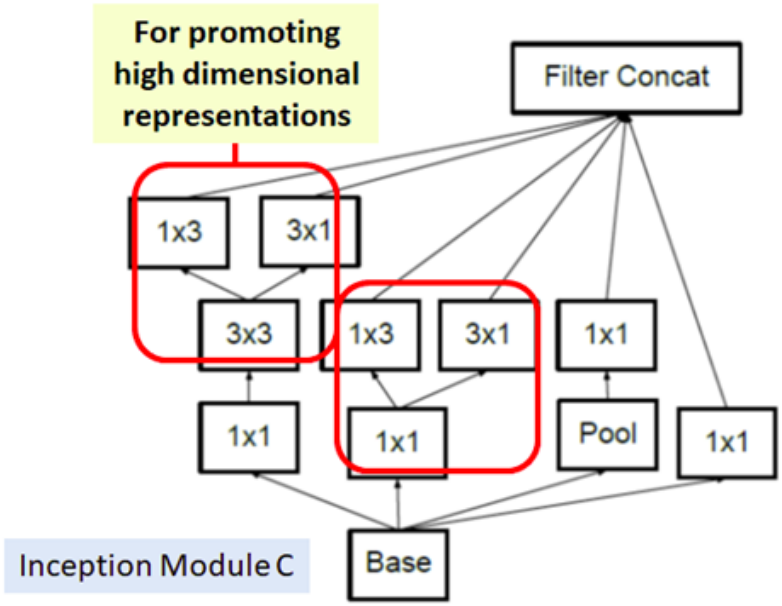
Inception Module A using factorization

– Factorizing into Asymmetric Convolutions.
Một lớp 3x1 convolution theo sau một lớp 1x3 convolution thay thế cho một lớp 3x3 convolution. + Nếu dùng lớp 3x3, số tham số cần thiết là: 3*3 = 9 + Nếu dùng kết hợp 3x1 và 1x3, số tham số cần thiết là: 3*1 + 1*3 = 6 Ta đã giảm được số tham số đi 33%. Từ đó ta có một Inception module mới (Gọi là Inception module B):



Inception Module B using asymmetric factorization

Và thêm một Inception module C được đề xuất như sau:

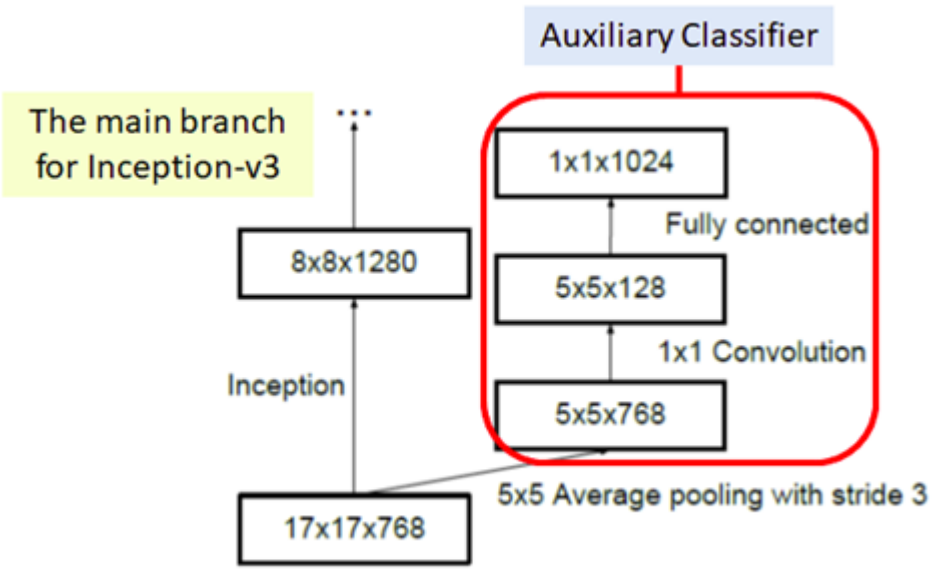


Inception Module C using asymmetric factorization

Nhờ Factorizing, số lượng tham số được giảm trong toàn mạng, giúp tránh tình trạng overfitting, từ đó mạng có thể đi sâu hơn.

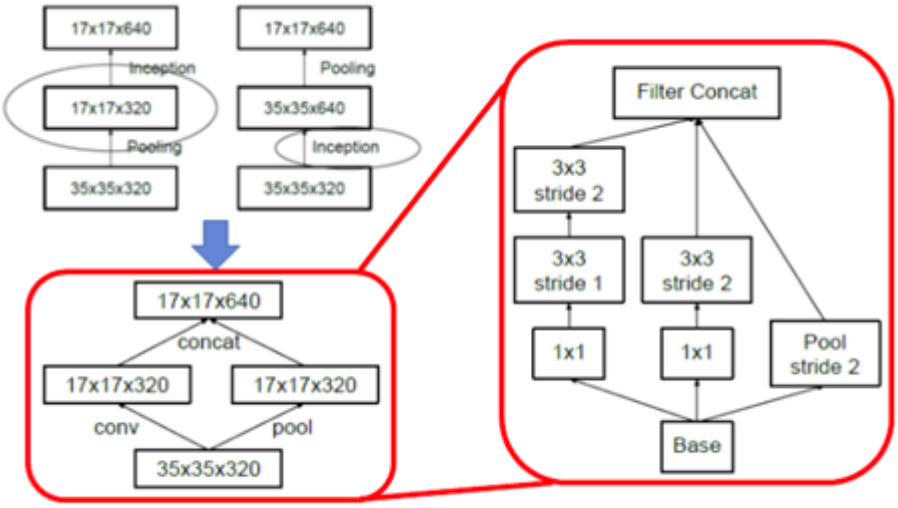
- Auxiliary Classifier:
  Auxiliary Classifier là một CNN nhỏ được chèn vào giữa các lớp trong quá trình huấn luyện (train) và tổn thất (loss) phát sinh được thêm vào tổn thất mạng chính. Trong GoogLeNet, Auxiliary Classifier được sử dụng cho một mạng sâu hơn, trong khi với Inception v3, nó hoạt động như một bộ điều chỉnh.

  Auxiliary Classifier đã được đề xuất trong GoogLeNet/Inception-v1. Có một số sửa đổi ở trong Inception-v3. Chỉ có một bộ Auxiliary Classifier được sử dụng ở trên đầu lớp 17x17 cuối cùng thay vì sử dụng 2 bộ. Với Inception-v3, Auxiliary Classifier được sử dụng làm bộ điều chỉnh. Batch Normalization (Đã được nhắc tới ở trên, được đề xuất trong Inception-v2) cũng được sử dụng ở đây.



Auxiliary Classifier act as a regularization

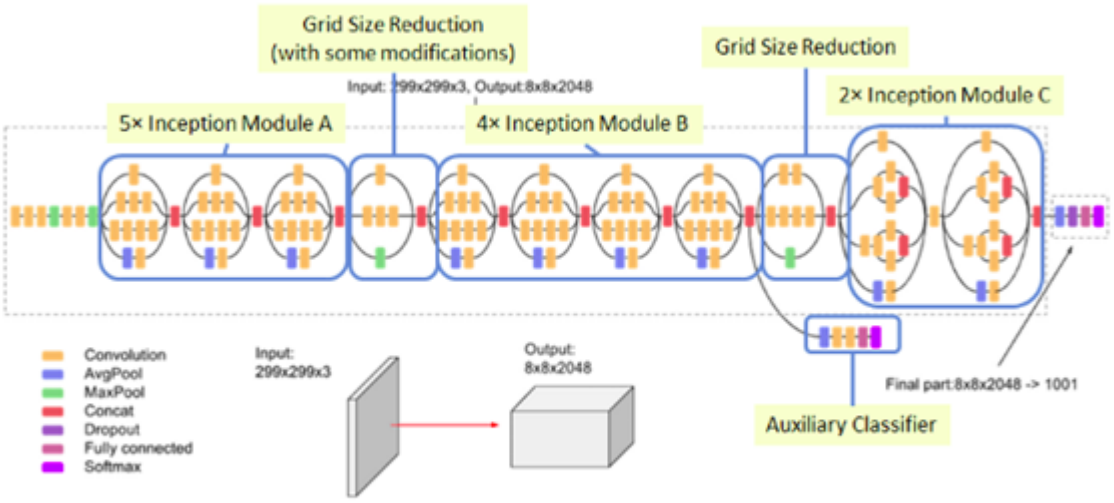
- Efficient Grid Size Reduction:
  Thông thường ta sẽ thực hiện việc giảm kích thước bằng max pooling. Nhưng việc sử dụng max pooling có những nhược điểm nhất định về chi phí tính toán. Sau đây là một cách khác được đề xuất, giúp giảm chi phí tính toán nhưng vẫn đạt được hiệu quả cao.



Conventional downsizing (Top Left), Efficient Grid Size Reduction (Bottom Left), Detailed Architecture of Efficient Grid Size Reduction (Right)

c) Cấu trúc mô hình.

Tổng thể lại ta có mô hình InceptionV3:



Inception-v3 Architecture (Batch Norm and ReLU are used after Conv)

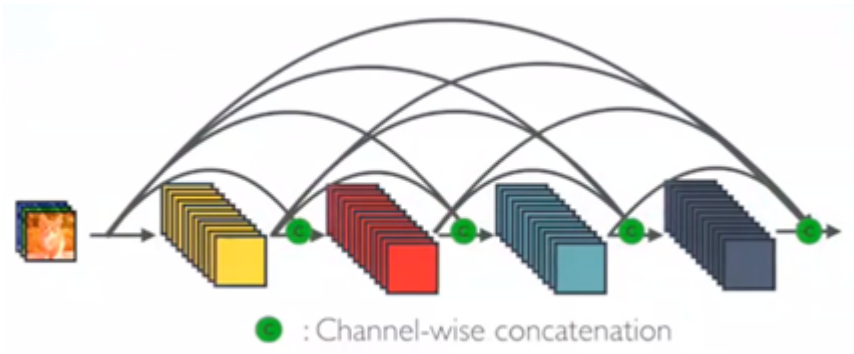
Với 42 lớp sâu (layer deep), chi phí tính toán chỉ cao hơn khoảng 2,5 lần so với GoogLeNet, và hiệu quả hơn rất nhiều so với VGGNet.

3.3 Mô hình Densenet.

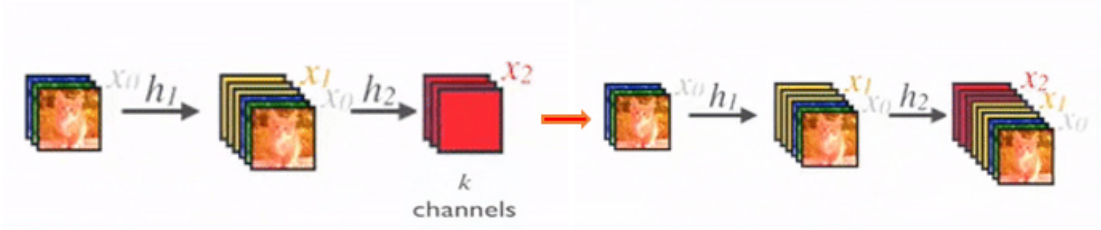
a) Giới thiệu
Densenet(Dense connected convolutional network) là một trong những netwok mới nhất cho visual object recognition. Ở một mức độ nào đó, DenseNet có thể được coi là phiên bản mở rộng hợp lý của ResNet.

b) Đặc điểm của mô hình.
Densenet có cấu trúc gồm các dense block và các transition layer. Các dense block định nghĩa cách các đầu vào và đầu ra được nối với nhau, trong khi các transition layers kiểm soát số lượng kênh sao cho nó không quá lớn.

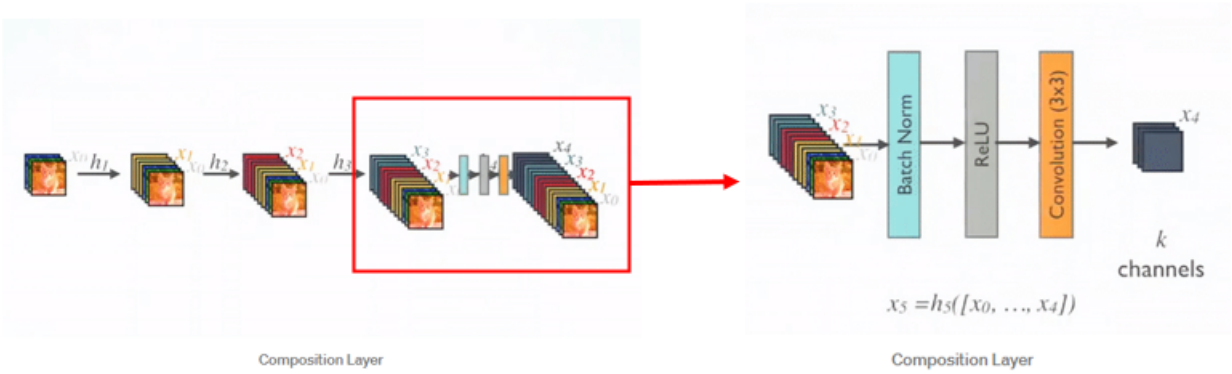
- Dense Block.
  Trong DenseNet, mỗi layer nhận thêm các input từ các layer trước đó và truyền feature-map của chính nó đến các layer phía sau. Với CNN truyền thống nếu chúng ta có L layer thì sẽ có L connection, còn trong densenet sẽ có L(L+1)/2 connection.



Sử dụng Concatenation, growth rate k sẽ là số lượng channel tăng lên qua mỗi lớp.

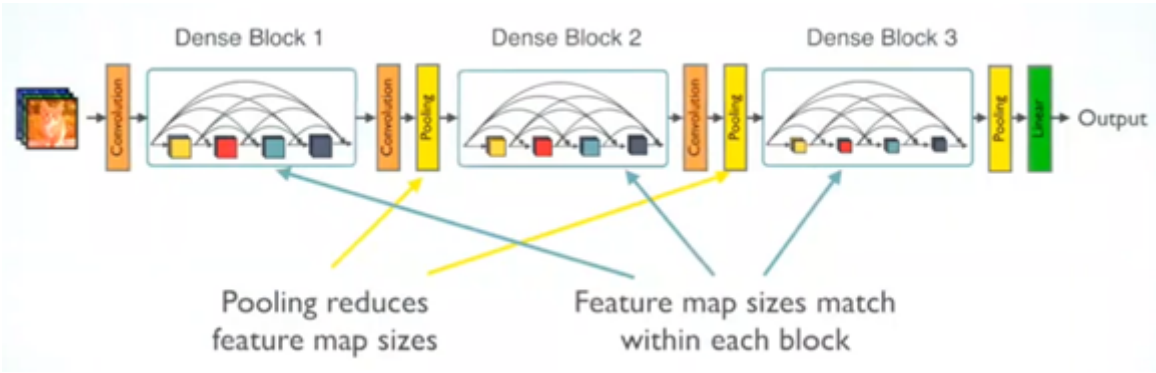


DenseNet sử dụng kiến trúc "chuẩn hóa theo batch, hàm kích hoạt và phép tích chập" đã qua sửa đổi của ResNet.



- Transition layer.
  Mỗi dense block sẽ làm tăng thêm số lượng channel. Nhưng việc thêm quá nhiều channel sẽ tạo nên một mô hình phức tạp quá mức. Do đó, transition layer sẽ được sử dụng để kiểm soát độ phức tạp của mô hình. Tầng này dùng một 1×1 Convolution để giảm số lượng kênh, theo sau là một 2×2 average pooling để giảm một nửa chiều cao và chiều rộng, từ đó giảm độ phức tạp của mô hình hơn nữa.

c) Cấu trúc của mô hình: Multiple Dense Blocks with Transition Layers.



- Đầu tiên, DenseNet sử dụng một Convolution và một Max Pooling như trong ResNet.
- Sau đó là các Dense Block xen kẽ với các Transition Layer.
- Ở cuối là một global average pooling và sau đó là một softmax classifier sẽ được sử dụng để cho ra Output.

4 Cài đặt mô hình.

4.1 Tiền xử lí dữ liệu ảnh.

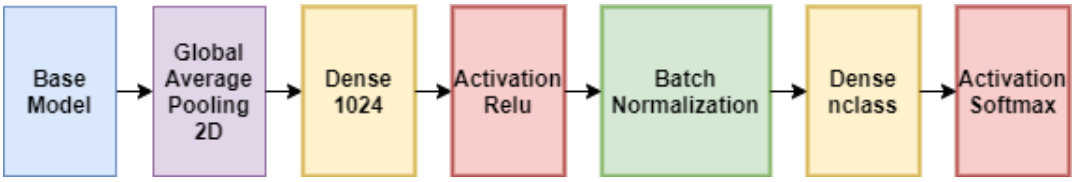
- Để chỉnh dữ liệu ảnh về khoảng 0-1 ta sẽ chia cho 255.
- Ta sử dụng kĩ thuật Data Augmentation để tạo thêm dữ liệu bằng các phương pháp:
  - Rotation: Xoay ảnh.
  - Zoom: Phóng to, thu nhỏ ảnh.
  - Shear: Làm méo ảnh.
  - Horizontal flip: Lật ảnh theo chiều ngang.
  - Vertical flip: Lật ảnh theo chiều dọc.
  - Width shift: Dịch ảnh theo chiều ngang.
  - Height shift: Dịch ảnh theo chiều dọc.

  Tensorflow có cung cấp một class **ImageDataGenerator** để thực hiện những phương pháp trên theo từng batch và dữ liệu ở những batch đó sẽ được sử dụng lại trong mỗi epoch.

```
train_datagen = ImageDataGenerator(rescale=1/255.0,
                                   rotation_range=5,
                                   zoom_range=0.1,
                                   shear_range=0.05,
                                   horizontal_flip=True,
                                   vertical_flip =True,
                                   width_shift_range=0.1,
                                   height_shift_range=0.1)
```

4.2 Xây dựng mô hình.

- Thư viện sử dụng xây dựng mô hình: tensorflow, keras.
- Sử dụng các mô hình ResNet, InceptionV3, DenseNet được xây dựng sẵn trong thư viện tensorflow.
- Sử dụng bộ trọng số có sẵn (bộ trọng số đã huấn luyện tập ImageNet) của các mô hình để làm trọng số ban đầu.
  Chọn bộ trọng số như vậy sẽ giúp cho mô hình sẽ huấn luyện cho kết quả chính xác cao hơn và tốc độ huấn luyện sẽ nhanh hơn.
- Sử dụng hàm loss: categorical_crossentropy.
- Sử dụng metric: accuracy.
- Ngoài ra trong mô hình, ta còn thêm vào **Model Checkpoint** và **Early Stopping**.
  - Model Checkpoint: Lưu lại mô hình, hoặc tham số của mô hình.
  - Early Stopping: Dừng train nếu mô hình không cải thiện.
- Mô hình cài đặt tổng quát:
  Coi các mô hình trên là Base Model, ta có mô hình như sau:



- Số lượng tham số của từng mô hình:

Model	Số lượng tham số
Base Model: ResNet152V2	18,329,668
Base Model: InceptionV3	21,810,980
Base Model: DenseNet201	58,339,844

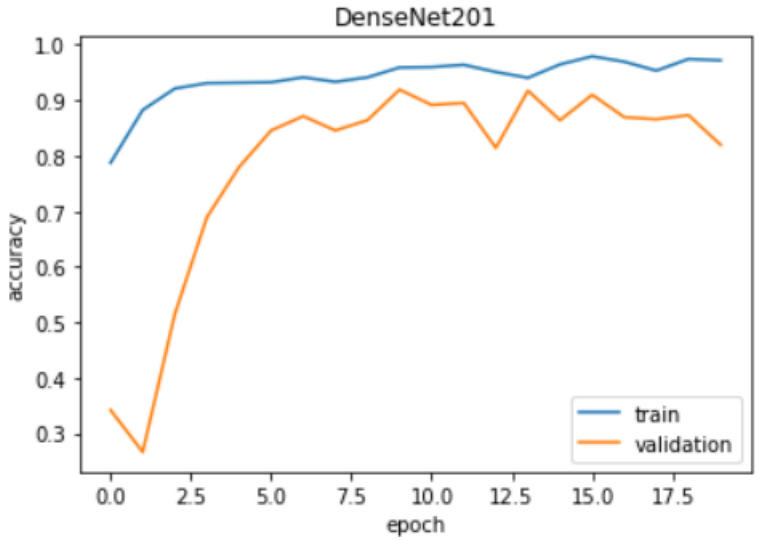
5 Báo cáo kết quả.

5.1 Kết quả của mô hình.

a) Dựa trên mô hình ResNet

	Tập train	Tập validation
Accuracy	93.8776%	92.5046%
Loss	0.201362	0.297874

Table 1: Kết quả của mô hình dựa trên ResNet

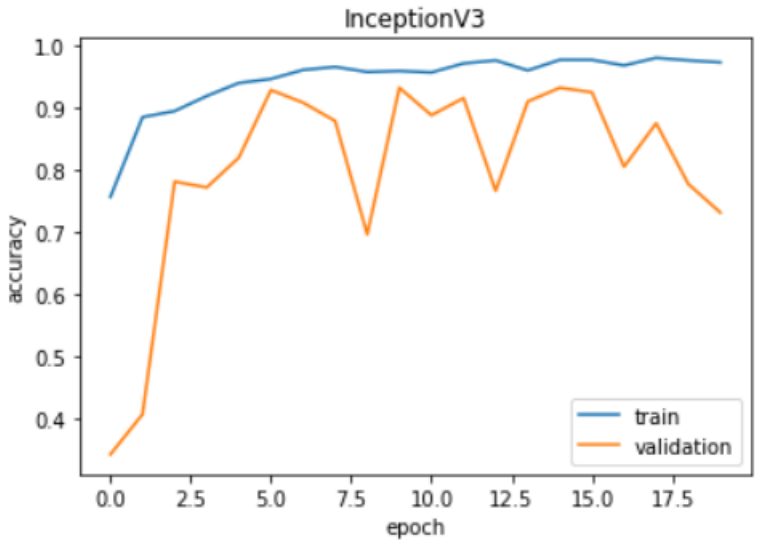


Lịch sử huấn luyện mô hình ResNet qua từng Epoch

b) Dựa trên mô hình InceptionV3

	Tập train	Tập validation
Accuracy	95.4474%	93.2358%
Loss	0.288272	0.297874

Table 2: Kết quả của mô hình dựa trên InceptionV3

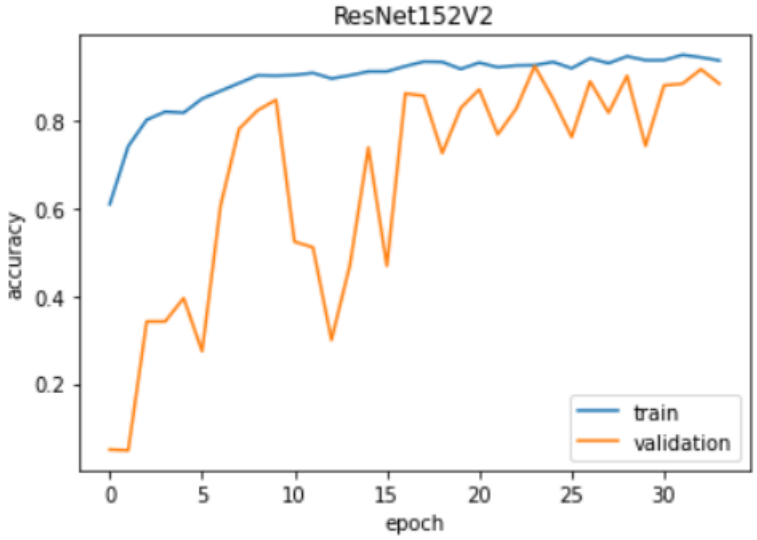


Lịch sử huấn luyện mô hình InceptionV3 qua từng Epoch

c) Dựa trên mô hình DenseNet.

	Tập train	**Tập validation**
Accuracy	95.5259%	91.9561%
Loss	0.179632	0.352452

Table 3: Kết quả của mô hình dựa trên DenseNet



Lịch sử huấn luyện mô hình DenNet qua từng Epoch

d) Đánh giá qua kết quả dự đoán tập test.

- Ba mô hình cho kết quả dự đoán giống nhau ở 1420 hình (tỉ lệ 78%).

- Ba mô hình cho kết quả dự đoán giống nhau ở 401 hình (tỉ lệ 22%).

6 Tài liệu tham khảo.

- Thư viện sử dụng trong đồ án:
  - pandas.
  - seaborn.
  - matplotlib.
  - tensorflow
  - cv2.
  - os.