

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KỲ

MÔN HỌC: NHẬP MÔN THỊ GIÁC MÁY TÍNH

ĐỀ TÀI: PHÂN LOẠI RÁC THẢI (GARBAGE CLASSIFICATION)

Giảng viên hướng dẫn : TS.Lê Minh Hưng

Sinh viên thực hiện : Võ Phạm Duy Đức – 19521383

Trịnh Công Danh – 19521326

Đỗ Trọng Khánh – 19521676

Lớp : CS231.M13.KHCL

I. MÔ TẢ BỘ DỮ LIỆU

- Bộ dữ liệu của của nhóm gồm 6 lớp: cardboard, glass, metal, paper, plastic, trash.
- Tổng số hình nhóm thu thập được là: 4801 hình ➔ Trung bình mỗi lớp 800 hình.

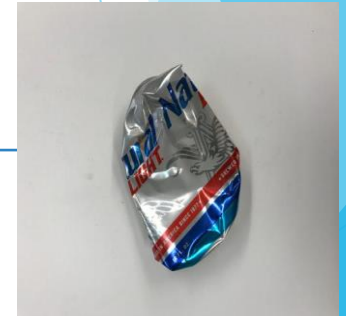
Cardboard: 800 hình



Glass: 801 hình



Metal: 799 image



GARBAGE

Trash: 801 hình

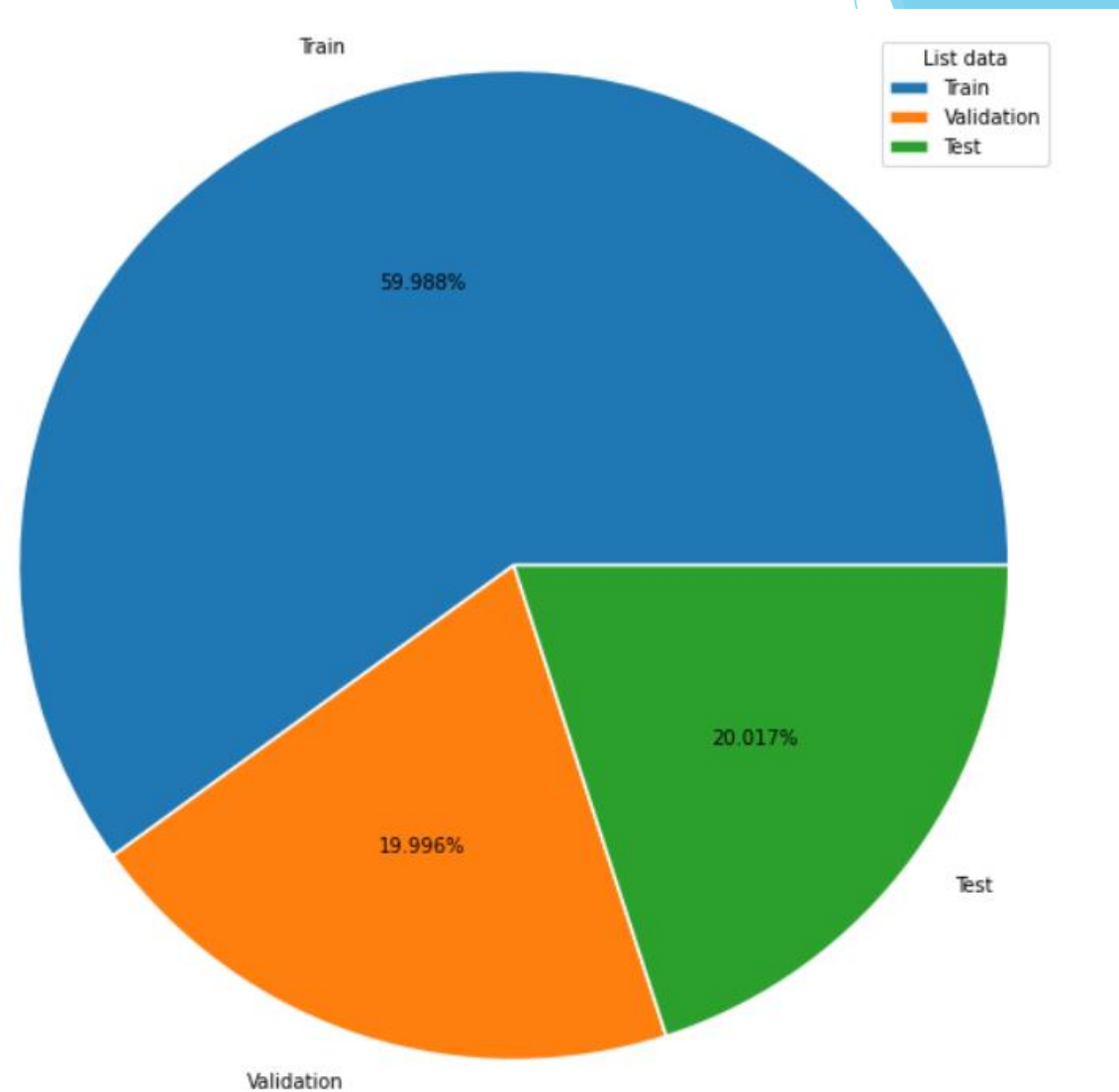


Plastic: 800 hình



Paper: 800 hình

- Nhóm sẽ chia bộ dữ liệu thành các bộ train, validation và test (6:2:2)
 - Train: mỗi lớp sẽ có 480 hình.
 - Validation: Mỗi lớp sẽ có 160 hình.
 - Test: Mỗi lớp sẽ có 160 hình.



- Sử dụng các phép biến đổi ảnh có sẵn trong keras để augment data cho tập train.

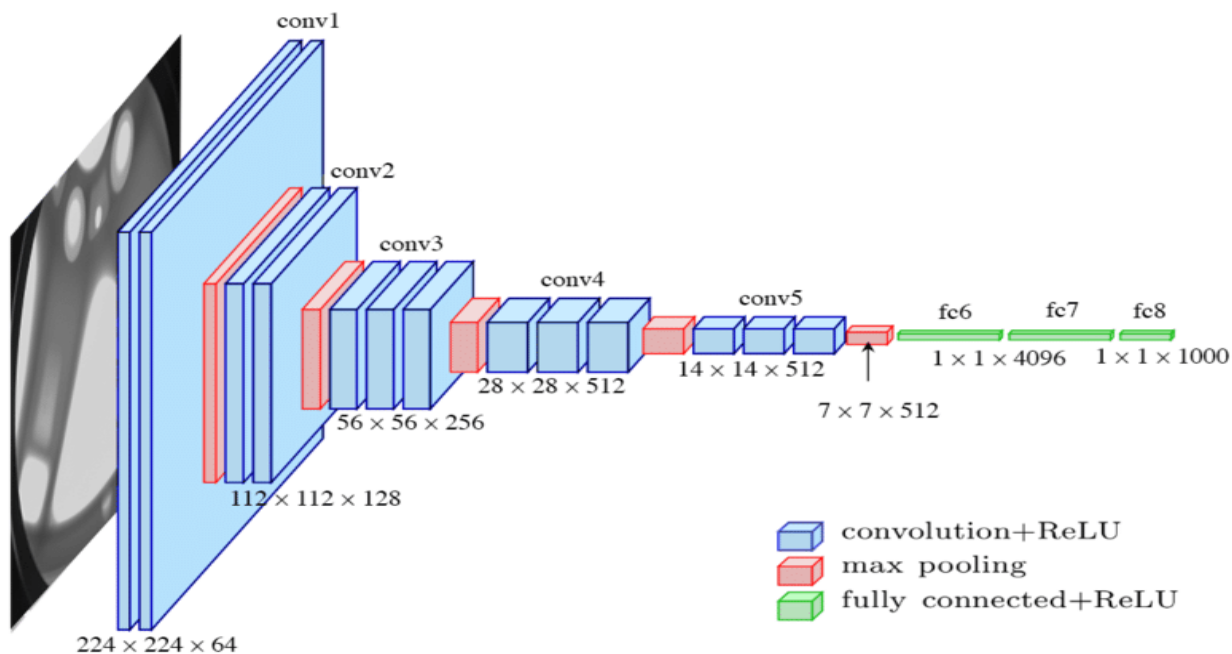
```
train_datagen = ImageDataGenerator(  
    rescale = 1./255,  
    rotation_range = 20, # Dữ liệu được tạo ra sẽ được xoay ngẫu nhiên một góc trong phạm vi +rotation_range đến -rotation_range (tính bằng độ)  
    width_shift_range = 0.2, # Dịch theo chiều ngang ngẫu nhiên trong một phạm vi nào đó  
    height_shift_range = 0.2, # Dịch ảnh theo chiều dọc trong một phạm vi nào đó  
    horizontal_flip = True, # lật ảnh ngẫu nhiên theo chiều ngang  
    vertical_flip = True, # lật ảnh ngẫu nhiên theo chiều dọc  
)
```

II. TRÍCH XUẤT ĐẶC TRƯNG VÀ HUẤN LUYỆN MODEL

1. MÔ HÌNH MẠNG VGG16

a. Giới thiệu về mô hình

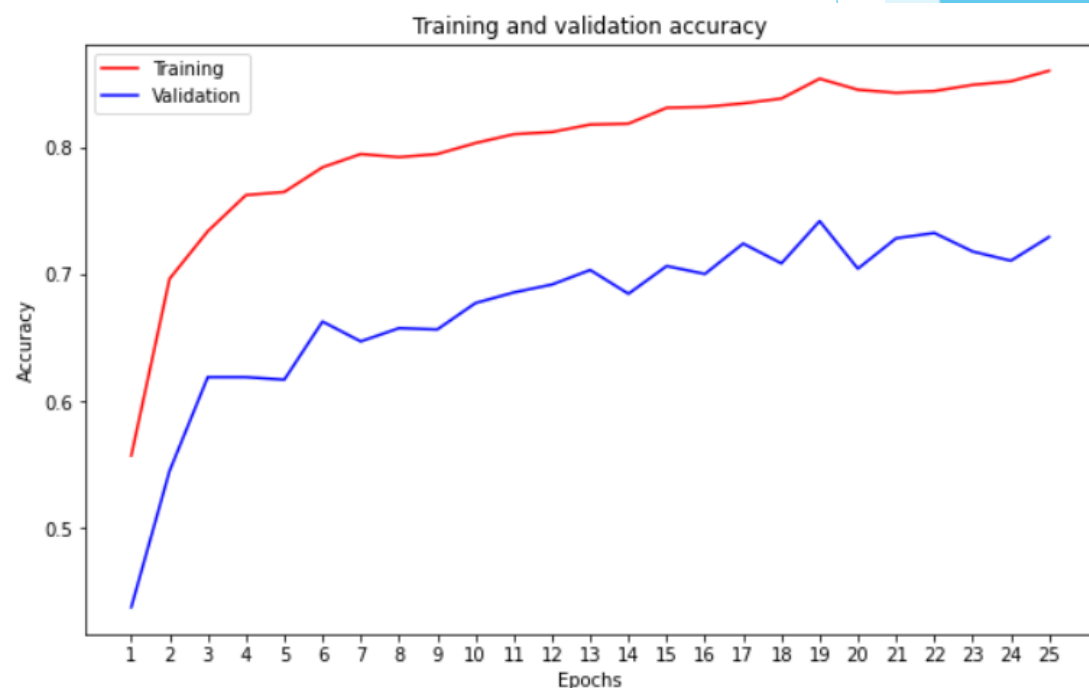
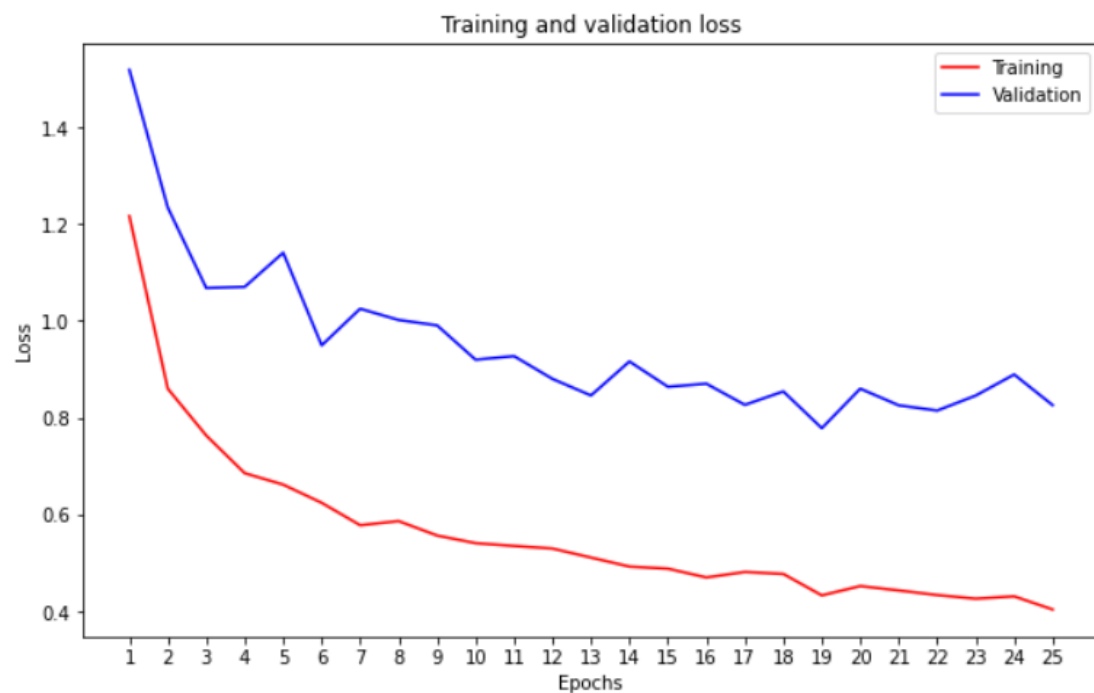
- **VGG16** bao gồm 16 layers: 13 layers **Conv** (2 layers conv-conv, 3 layers conv-conv-conv) đều có kernel 3x3, sau mỗi layer Conv là **maxpooling** downsize xuống 0.5 và 3 layer **fully connected**.



- Nhờ việc sử dụng 3 lớp conv layer 3x3 làm tăng tính **effective receptive field** hơn so với lớp conv layer 11x11 của mô hình **AlexNet**. Ngoài ra kích thước bộ lọc nhỏ sẽ giúp giảm số lượng tham số cho mô hình và đem lại kết quả tốt hơn.

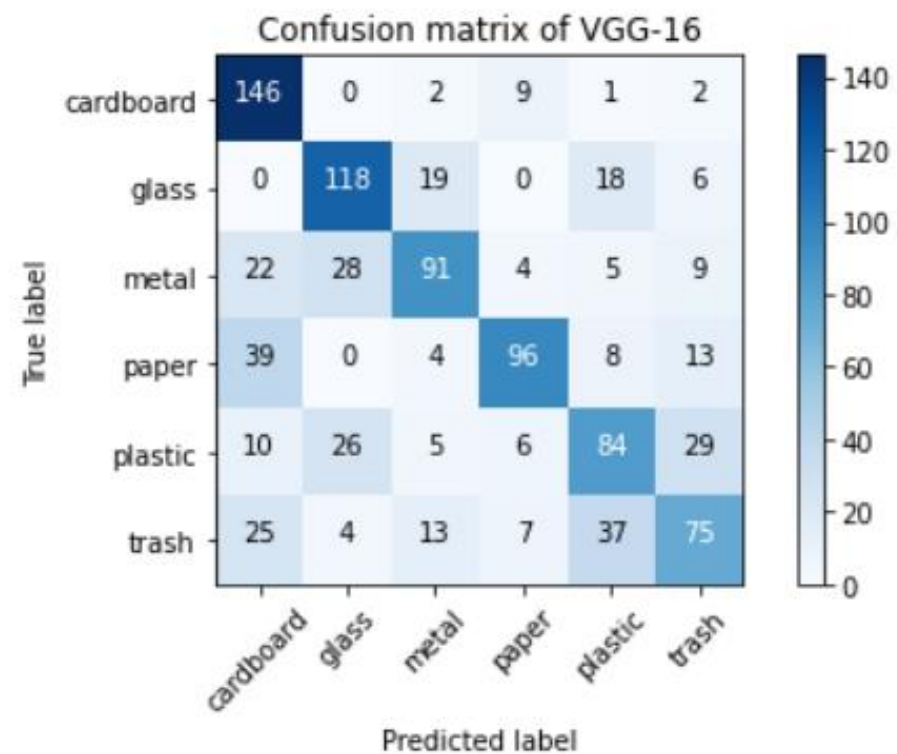
b. Ý tưởng và kết quả

- Hướng 1: Fine tune
 - Bỏ các lớp FC ban đầu và thay thế bằng các lớp FC phù hợp với bài toán và sử dụng activation softmax cho việc phân loại.
 - Đóng băng các lớp trước khi thêm lớp FC vào. Sau khi train được 20 epoch, mở băng tất cả để train tiếp.



Visualize accuracy và loss của 2 tập train và test

	precision	recall	f1-score	support
cardboard	0.60	0.91	0.73	160
glass	0.67	0.73	0.70	161
metal	0.68	0.57	0.62	159
paper	0.79	0.60	0.68	160
plastic	0.55	0.53	0.54	160
trash	0.56	0.47	0.51	161
accuracy			0.63	961
macro avg	0.64	0.63	0.63	961
weighted avg	0.64	0.63	0.63	961

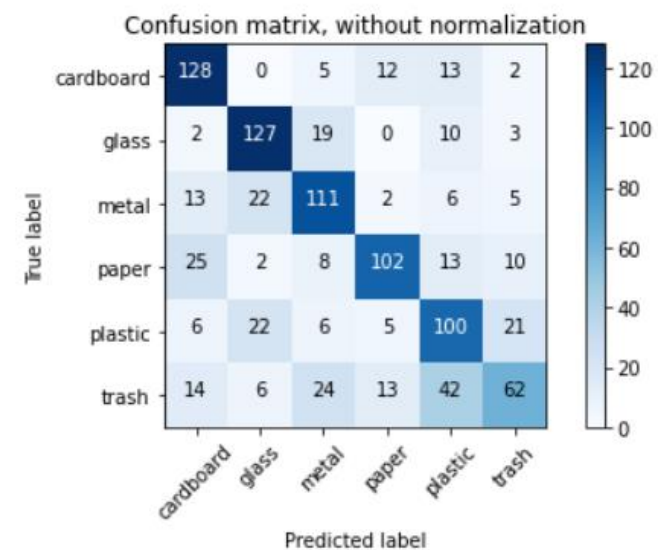
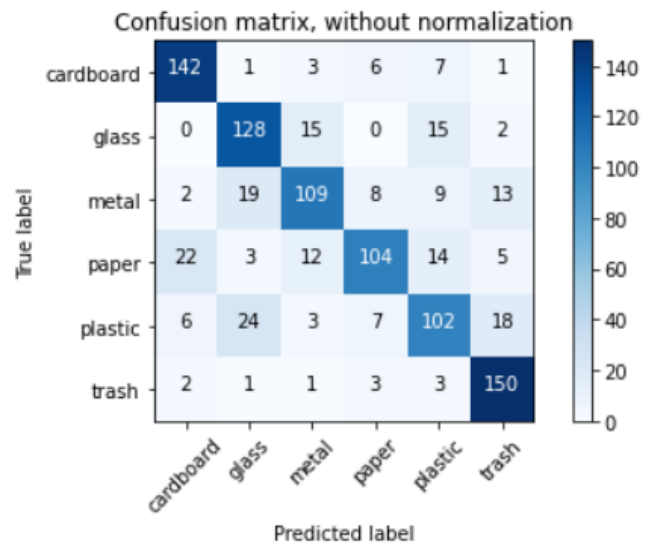


Kết quả đánh giá trên tập test

- Hướng 2: Trích xuất đặc trưng ở lớp FC7 và đưa vào mô hình **Support Vector Machine (SVM)**.

----- Validation -----				
	precision	recall	f1-score	support
cardboard	0.82	0.89	0.85	160
glass	0.73	0.80	0.76	160
metal	0.76	0.68	0.72	160
paper	0.81	0.65	0.72	160
plastic	0.68	0.64	0.66	160
trash	0.79	0.94	0.86	160
accuracy			0.77	960
macro avg	0.77	0.77	0.76	960
weighted avg	0.77	0.77	0.76	960

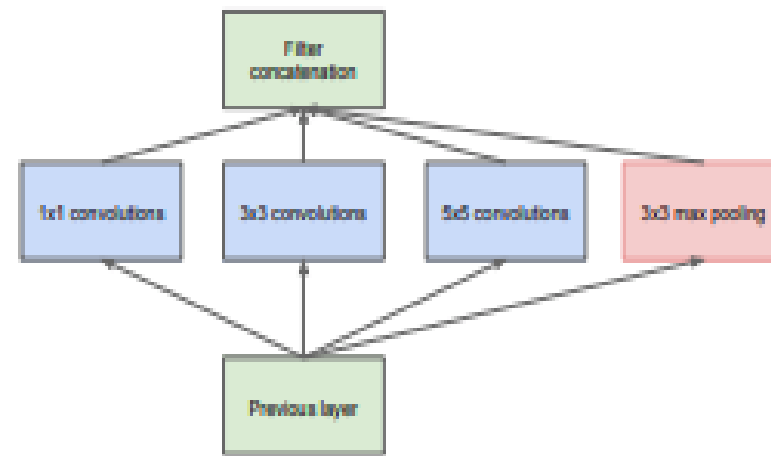
----- Test -----				
	precision	recall	f1-score	support
cardboard	0.68	0.80	0.74	160
glass	0.71	0.79	0.75	161
metal	0.64	0.70	0.67	159
paper	0.76	0.64	0.69	160
plastic	0.54	0.62	0.58	160
trash	0.60	0.39	0.47	161
accuracy			0.66	961
macro avg	0.66	0.66	0.65	961
weighted avg	0.66	0.66	0.65	961



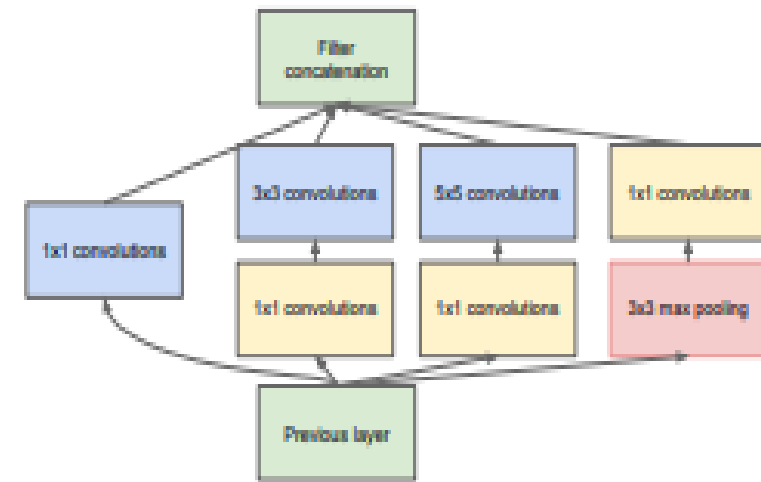
2. MÔ HÌNH MẠNG GOOGLE-NET

a. Giới thiệu về mô hình

- **GoogleNet** được tạo thành từ các inception module rất hiệu quả.
- Ý tưởng của **GoogleNet** là kết hợp các kernel khác nhau trong cùng một module tăng số lượng thông tin từ vùng ảnh trong ảnh.
- Với kiểu thiết kế naïve version: Gồm có 4 kernel để trích xuất đặc trưng ảnh: 1x1, 3x3, 5x5 và một 3x3 maxpooling. Mỗi kernel đều thêm padding để đảm bảo đầu ra feature map có cùng kích thước để concatenate lại với nhau.



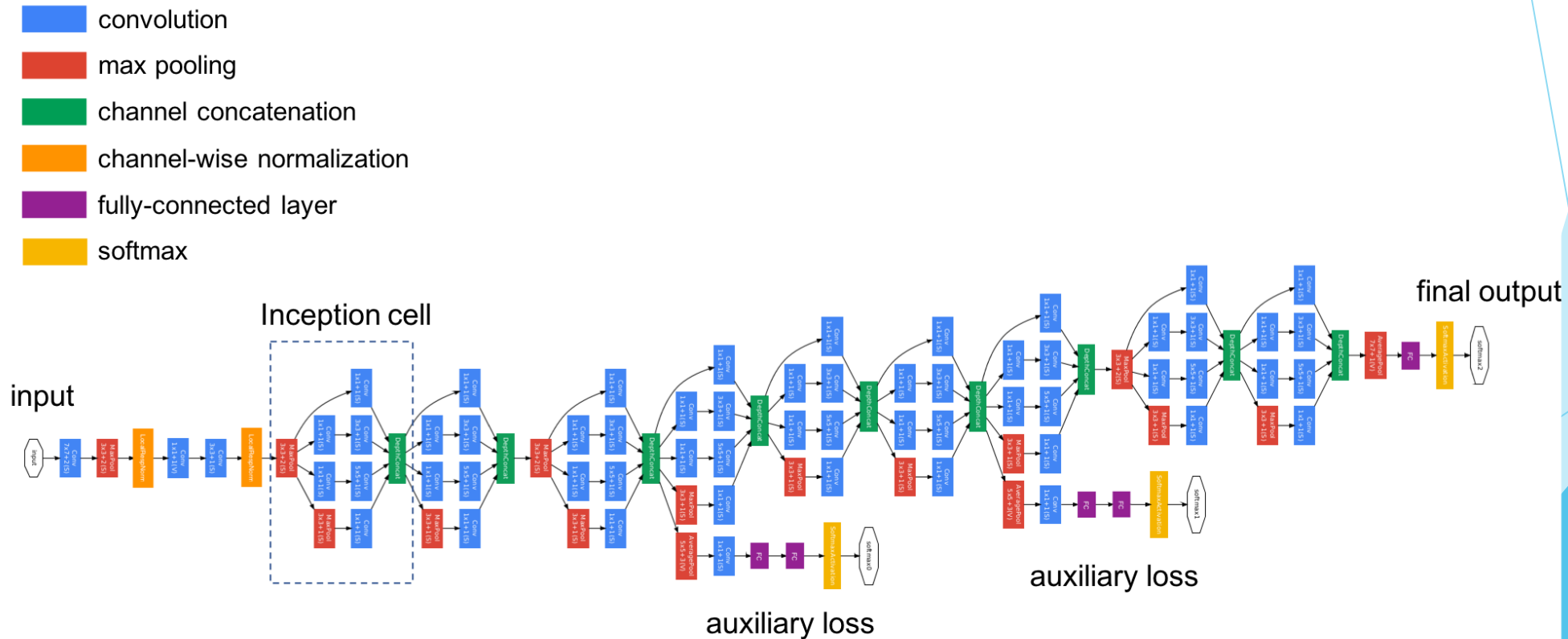
(a) Inception module, naïve version



(b) Inception module with dimension reductions

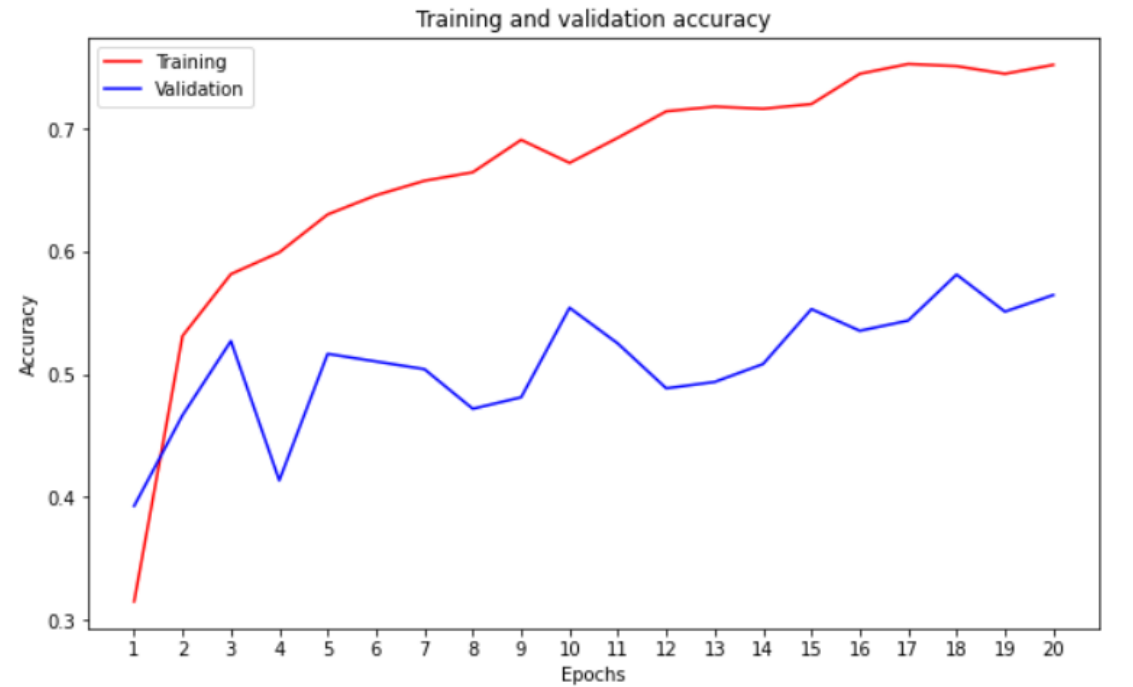
Figure 2: Inception module

- Kiểu thiết kế thứ 2: Thêm các **bottleneck** (kernel size 1x1) trước khi đi qua các kernel size khác để giảm chiều sâu \rightarrow giảm số lượng phép nhân xuống rất nhiều.
- Lớp cuối là **Classifier output** không sử dụng lớp FC. Có 3 output trên cùng 1 network để giảm hiện tượng **gradients vanishing** (Mất mát đạo hàm).



b. Ý tưởng và kết quả

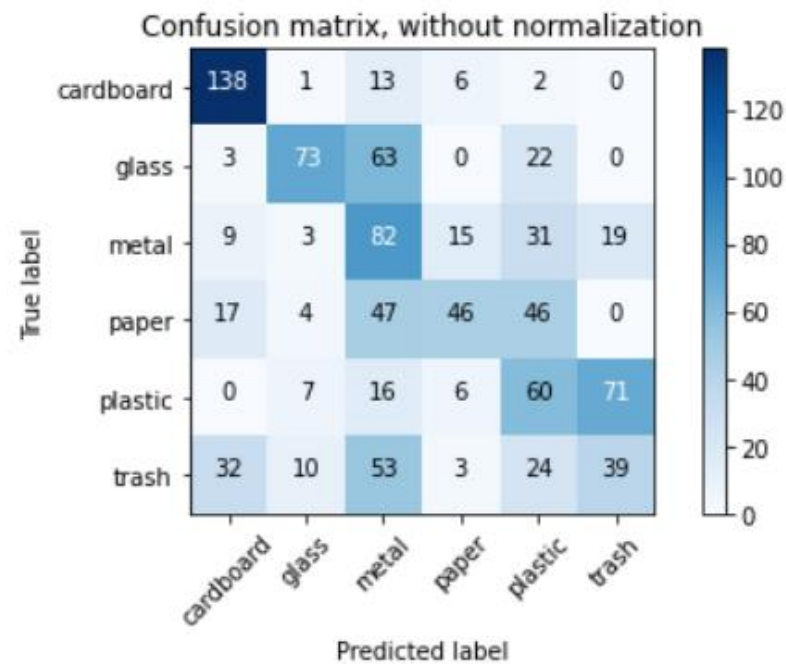
- Hướng 1: Build model và sử dụng layer có activation **softmax** cho việc phân loại.



Visualize accuracy và loss của 2 tập train và test

Classification Report

	precision	recall	f1-score	support
cardboard	0.69	0.86	0.77	160
glass	0.74	0.45	0.56	161
metal	0.30	0.52	0.38	159
paper	0.61	0.29	0.39	160
plastic	0.32	0.38	0.35	160
trash	0.30	0.24	0.27	161
accuracy			0.46	961
macro avg	0.49	0.46	0.45	961
weighted avg	0.50	0.46	0.45	961

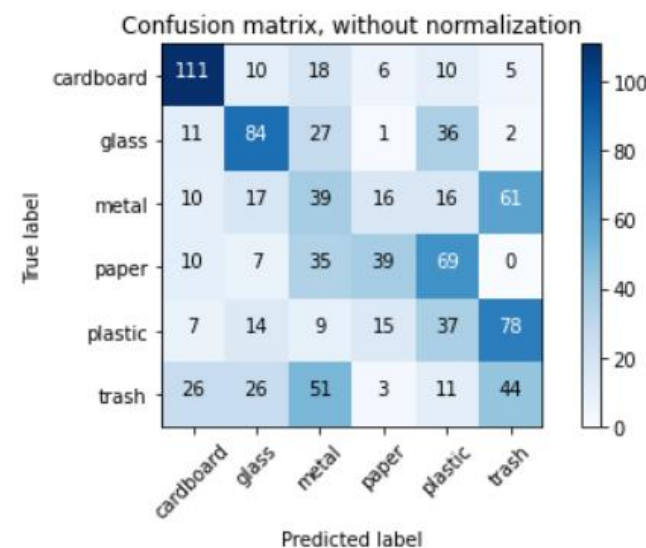
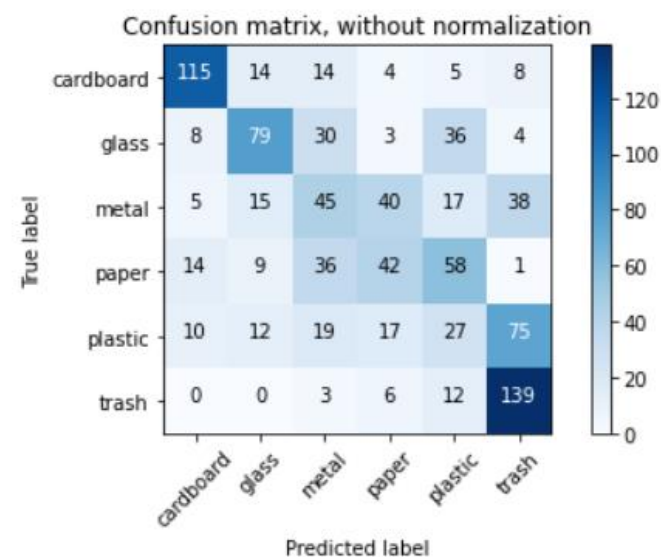


Kết quả đánh giá trên tập test

- Hướng 2: Trích xuất đặc trưng ở lớp Average Pooling cuối và đưa vào SVM để huấn luyện.

----- Validation -----				
	precision	recall	f1-score	support
cardboard	0.76	0.72	0.74	160
glass	0.61	0.49	0.55	160
metal	0.31	0.28	0.29	160
paper	0.38	0.26	0.31	160
plastic	0.17	0.17	0.17	160
trash	0.52	0.87	0.65	160
accuracy			0.47	960
macro avg	0.46	0.47	0.45	960
weighted avg	0.46	0.47	0.45	960

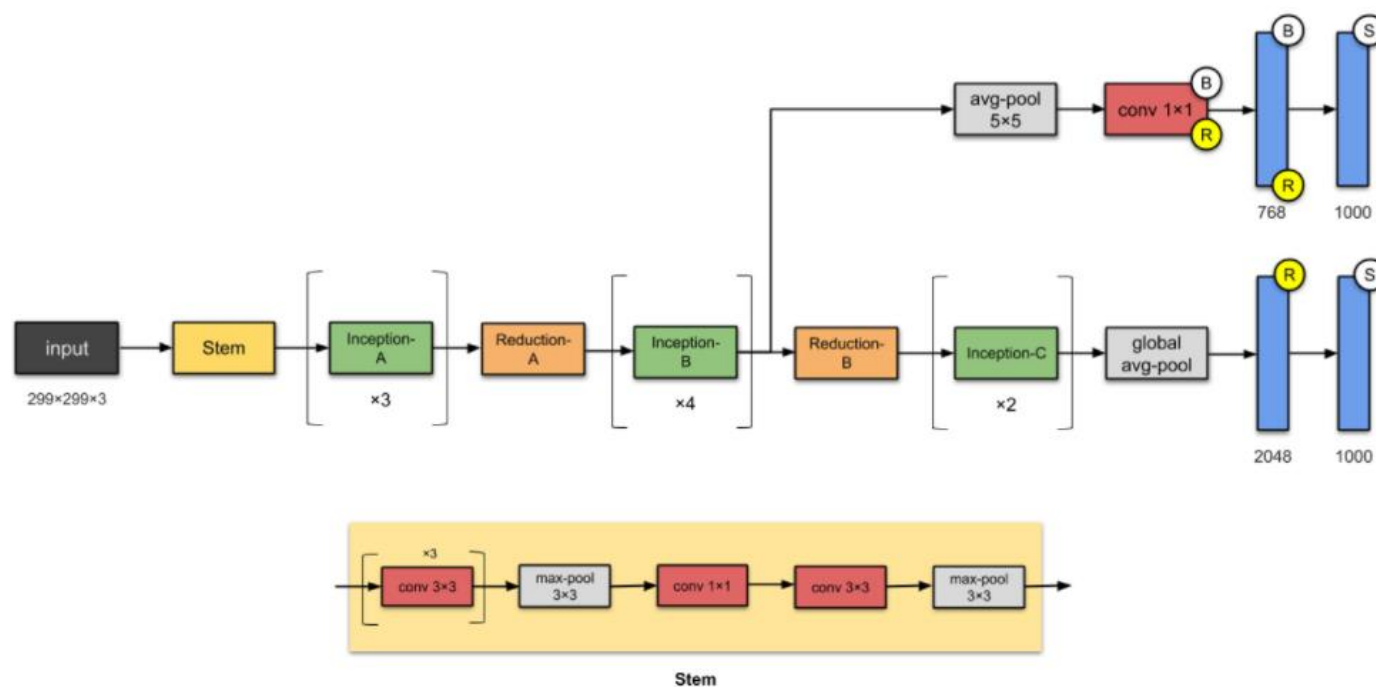
----- Test -----				
	precision	recall	f1-score	support
cardboard	0.63	0.69	0.66	160
glass	0.53	0.52	0.53	161
metal	0.22	0.25	0.23	159
paper	0.49	0.24	0.33	160
plastic	0.21	0.23	0.22	160
trash	0.23	0.27	0.25	161
accuracy			0.37	961
macro avg	0.38	0.37	0.37	961
weighted avg	0.39	0.37	0.37	961



3. MÔ HÌNH MẠNG INCEPTION-V3

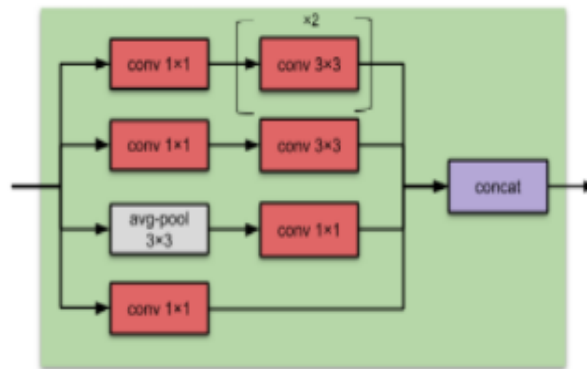
a. Giới thiệu về mô hình

- **Inception-V3** là kế thừa của GoogleNet(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.
- **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.

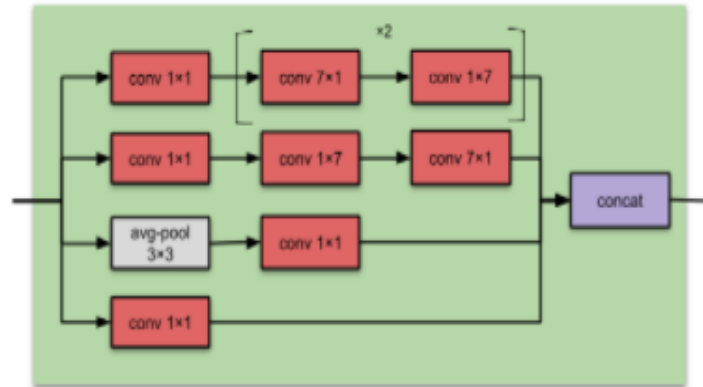


a. Giới thiệu về mô hình

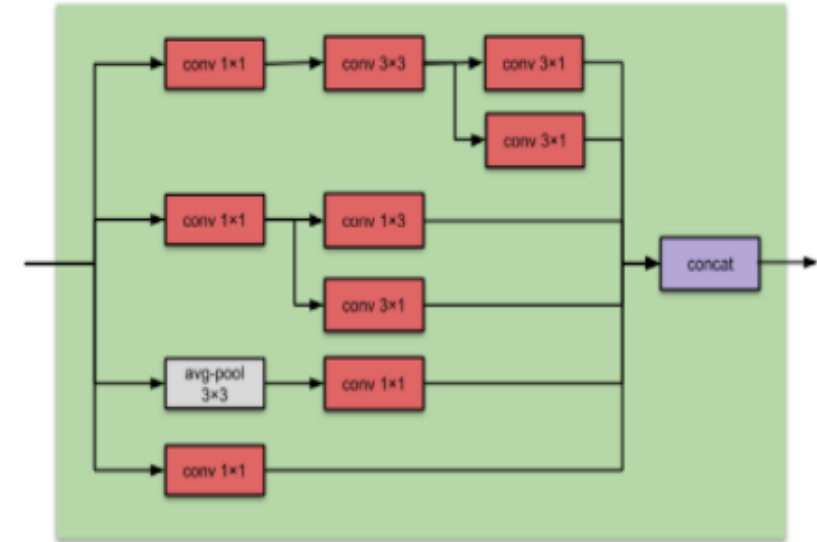
- Hiện tại **Inception module** bao gồm 3 version: Inception-A, Inception-B, Inception-C.
- Ngoài ra ở **Inception-V3** chúng ta còn sử dụng 2 kiến trúc giảm chiều dữ liệu là Reduction-A và Reduction-B.



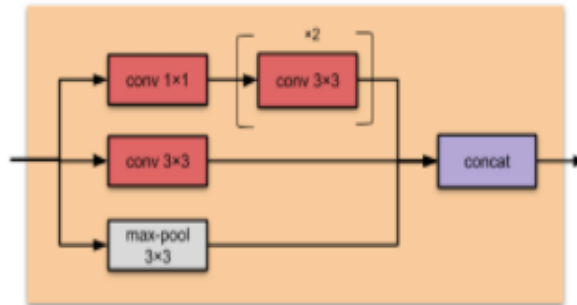
Inception-A



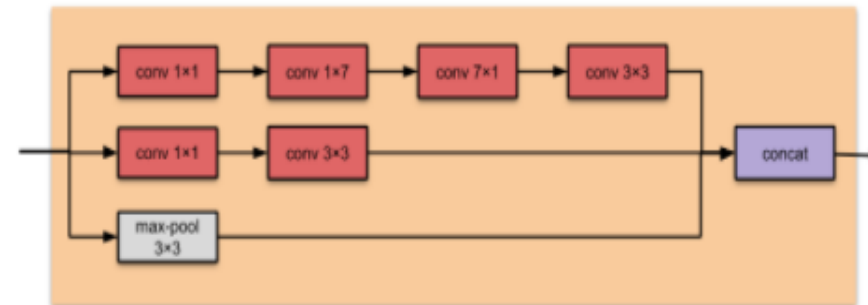
Inception-B



Inception-C



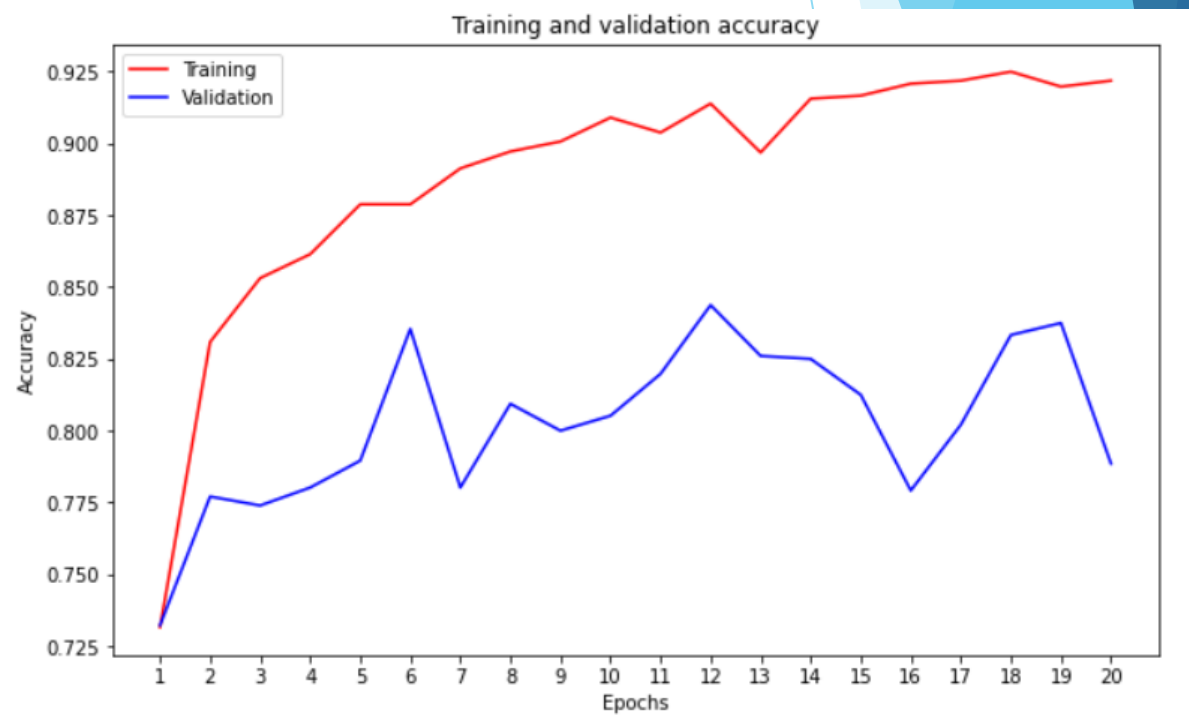
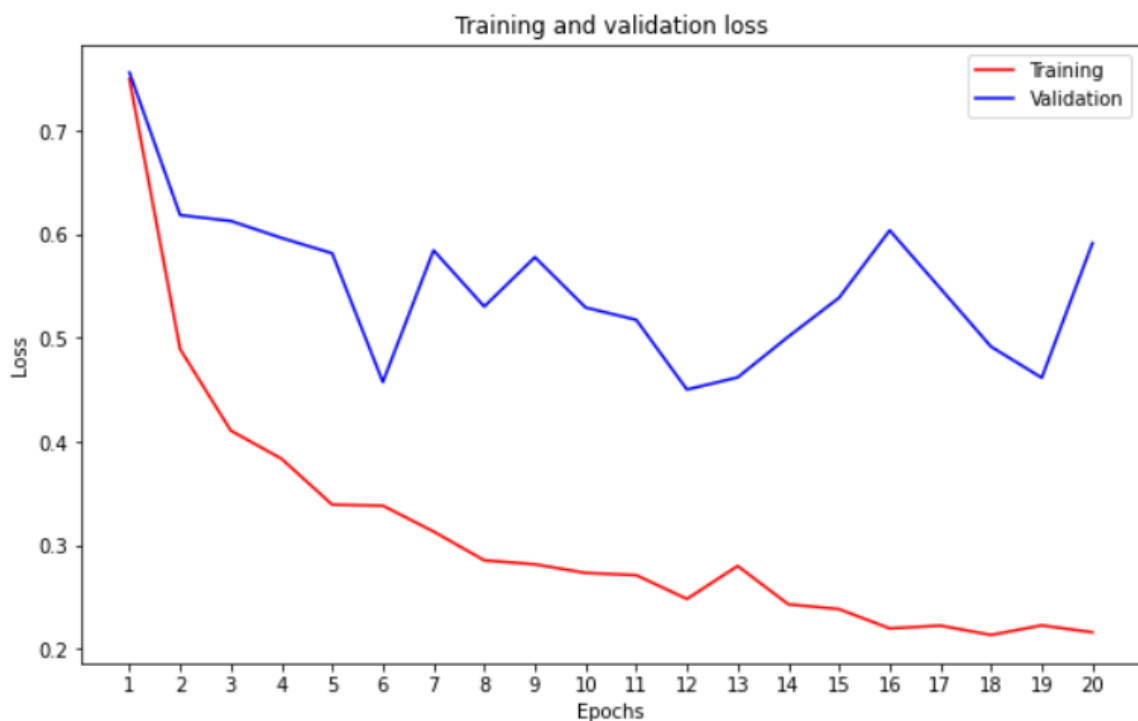
Reduction-A



Reduction-B

b. Ý tưởng và kết quả

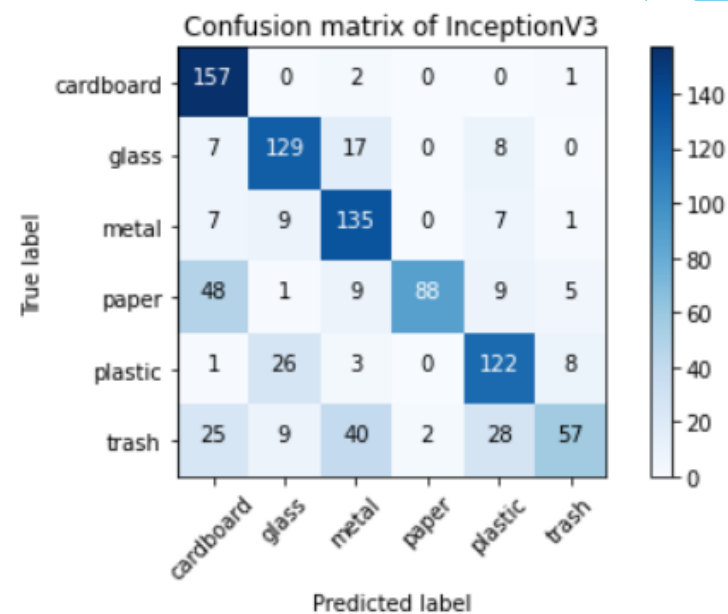
- Hướng 1: Finetune
 - Bỏ các lớp FC ban đầu và thay thế bằng các lớp FC phù hợp với bài toán và sử dụng activation softmax cho việc phân loại.
 - Đóng băng các lớp trước khi thêm lớp FC vào.



Visualize accuracy và loss của 2 tập train và test

Classification Report

	precision	recall	f1-score	support
cardboard	0.64	0.98	0.78	160
glass	0.74	0.80	0.77	161
metal	0.66	0.85	0.74	159
paper	0.98	0.55	0.70	160
plastic	0.70	0.76	0.73	160
trash	0.79	0.35	0.49	161
accuracy			0.72	961
macro avg	0.75	0.72	0.70	961
weighted avg	0.75	0.72	0.70	961

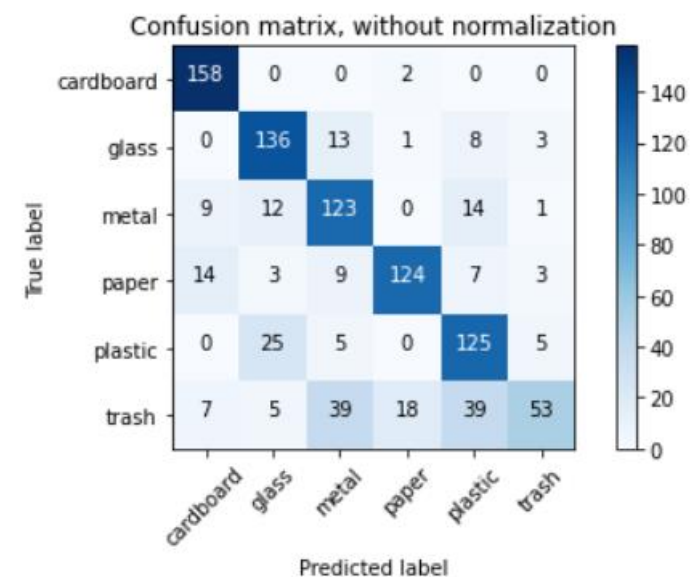
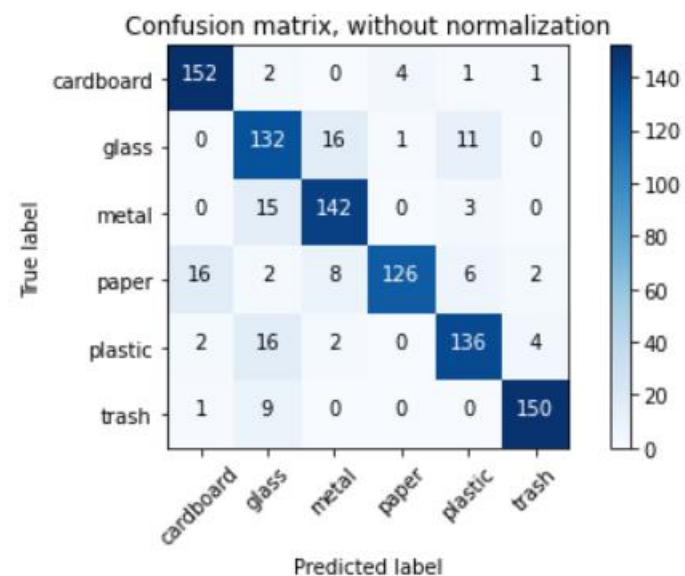


Kết quả đánh giá trên tập test

- Hướng 2: Trích xuất đặc trưng ở lớp GlobalAveragePooling cuối và đưa vào SVM để huấn luyện.

----- Validation -----				
	precision	recall	f1-score	support
cardboard	0.89	0.95	0.92	160
glass	0.75	0.82	0.79	160
metal	0.85	0.89	0.87	160
paper	0.96	0.79	0.87	160
plastic	0.87	0.85	0.86	160
trash	0.96	0.94	0.95	160
accuracy			0.87	960
macro avg	0.88	0.87	0.87	960
weighted avg	0.88	0.87	0.87	960

----- Test -----				
	precision	recall	f1-score	support
cardboard	0.84	0.99	0.91	160
glass	0.75	0.84	0.80	161
metal	0.65	0.77	0.71	159
paper	0.86	0.78	0.81	160
plastic	0.65	0.78	0.71	160
trash	0.82	0.33	0.47	161
accuracy			0.75	961
macro avg	0.76	0.75	0.73	961
weighted avg	0.76	0.75	0.73	961



III. TỔNG HỢP KẾT QUẢ

	Precision	Recall	F1-score	Accuracy
VGG16	0.64	0.63	0.63	0.63
GoogleNet	0.49	0.46	0.45	0.46
InceptionV3	0.75	0.72	0.7	0.72
VGG16 + SVM (linear)	0.58	0.59	0.59	0.59
VGG16 + SVM (rbf)	0.55	0.53	0.52	0.53
VGG16 + SVM (poly)	0.66	0.66	0.65	0.66
GoogleNet + SVM (linear)	0.28	0.24	0.25	0.24
GoogleNet + SVM (rbf)	0.35	0.33	0.33	0.33
GoogleNet + SVM (poly)	0.38	0.37	0.37	0.37
InceptionV3 + SVM (linear)	0.74	0.72	0.71	0.72
InceptionV3 + SVM (rbf)	0.74	0.72	0.71	0.72
InceptionV3 + SVM (poly)	0.76	0.75	0.73	0.75

IV. KẾT LUẬN

- Mô hình mạng Inception-V3 cho kết quả tốt nhất (Accuracy: 72%).
- Với hướng làm trích xuất đặc trưng đưa vào mô hình SVM, mô hình mạng Inception-V3 cũng cho kết quả tốt nhất với kernel polynormal (Accuracy: 75%).
- Lớp “**trash**” trong tập test cho kết quả khá thấp. Điều này là do trong lúc thu thập data, lớp “**trash**” bị thiếu nên nhóm bổ sung thêm data cho lớp này, trong quá trình bổ sung có thể bị trùng lặp data.
- Hướng cải thiện kết quả cho bài toán:
 - Data của nhóm thu thập trên Kaggle là khá ít nên sẽ tự thu thập thêm data hoặc sử dụng thêm những phương pháp augment data.
 - Tiếp cận thêm nhiều mô hình mạng hơn nữa như **ResNet**, **EfficientNet**,...
- Link source code: <https://github.com/trong-khanh-1109/CS231.M13.KHCL>

CẢM ƠN THẦY
VÀ CÁC BẠN ĐÃ
THEO DÕI!