

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



Đề tài: **GARBAGE CLASSIFICATION**

ĐỒ ÁN CUỐI KỲ MÔN MÁY HỌC - CS114.N21

GIẢNG VIÊN HƯỚNG DẪN:

PGS.TS. Lê Đình Duy

ThS. Phạm Nguyễn Trường An

Nhóm sinh viên thực hiện: NSFV

Bùi Huỳnh Kim Uyên	21521659
Nguyễn Nguyễn Giáp	21522025
Nguyễn Trọng Ân	21520548
Nguyễn Bùi Thanh Mai	21522320

TP.HCM – Tháng 07 Năm 2023

TÓM TẮT ĐỒ ÁN

Đồ án này nhằm tạo ra một mô hình máy học có khả năng phân loại bảy loại rác có thể tái chế hoặc không thể tái chế tương ứng với 7 classes: cardboard, compost, glass, metal, plastic, paper và trash.

Nhóm đã sử dụng ba mô hình khác nhau để train và đánh giá, gồm SVM, neural network dùng kiến trúc VGG16 và neural network dùng kiến trúc ResNet50. Nhóm đã đánh giá dựa trên ba metric Accuracy, Precision, Recall, trong đó Accuracy sẽ là metric chính quyết định model nào tốt nhất.

Mã nguồn và dataset của nhóm được lưu trữ ở Google Drive [\[NSFW\]FinalProject \(https://drive.google.com/drive/folders/12n0kOq3OJUrkc2QSttbLUGO0n_QHv_n5?usp=drive link\)](https://drive.google.com/drive/folders/12n0kOq3OJUrkc2QSttbLUGO0n_QHv_n5?usp=drive_link)

BỔ SUNG SAU VẤN ĐÁP

Theo gợi ý của Giảng viên, nhóm đã thực hiện bổ sung thêm một số thí nghiệm mới. Nhóm đã thực hiện thêm hai thí nghiệm sử dụng MobileNet, MobileNetV2, và ba thí nghiệm sử dụng SVM với Feature Extraction bằng CNN. Kết quả thí nghiệm được bổ sung tại mục **VI. Bổ sung và đánh giá** (Trang 24).

Mục lục

I.	Tổng quan về đề án.....	3
1.1	Ngữ cảnh ứng dụng.....	3
1.2	Input và output bài toán.....	5
1.3	Hướng đi của nhóm.....	5
1.4	Cách đánh giá performance.....	5
1.5	Bộ dữ liệu	5
1.6	Kết quả phân tích bộ dữ liệu.....	6
II.	Chia tập train/val/test	9
III.	Training và đánh giá model.....	9
3.1	Phân tích hướng đi và kết quả những người đi trước	9
3.2	Tổng quan các model của nhóm	10
3.2.1	<i>Support Vector Machine (SVM)</i>	10
3.2.2	<i>VGG16</i>	14
3.2.3	<i>ResNet50</i>	19
IV.	Kết luận	23
V.	Hướng phát triển trong tương lai.....	23
VI.	Bổ sung và đánh giá.....	24
6.1	Bổ sung kết quả thí nghiệm.....	24
6.2	Đánh giá	25

I. Tổng quan về đề án

1.1 Ngữ cảnh ứng dụng

Đã từ lâu người dân luôn có thói quen vứt bỏ rác thải bừa bãi ra môi trường xung quanh như: sông, rạch, ao, hồ, đường ngõ hay bất kỳ một chỗ đất trống nào đó... làm cho vấn đề ô nhiễm môi trường do rác ngày càng trở nên nghiêm trọng hơn bao giờ hết làm ảnh hưởng trực tiếp đến môi trường sống của người dân tại khu vực gây nên ô nhiễm nguồn nước, không khí, tạo điều kiện cho vi khuẩn và các loại bệnh tật phát triển có thể trở thành nguồn phát sinh dịch bệnh, tác động nghiêm trọng tới môi trường, gây mất mỹ quan và không tận dụng được các nguyên liệu có thể tái sinh.

Phân loại rác thải từ hộ gia đình (tại nguồn) góp phần tận dụng những nguyên liệu có thể tái chế, chuyển đổi công dụng như kim loại, nhựa, giấy, thủy tinh, bìa cứng... từ đó làm giảm khối lượng rác thải sinh hoạt phát sinh phải vận chuyển, xử lý, đồng thời tăng cường tái sử dụng chất thải hữu cơ, tạo nguồn phân bón sạch cho cây trồng, giảm chi phí trong sản xuất nông nghiệp, giảm thiểu ô nhiễm và các dịch bệnh từ rác thải gây ra.

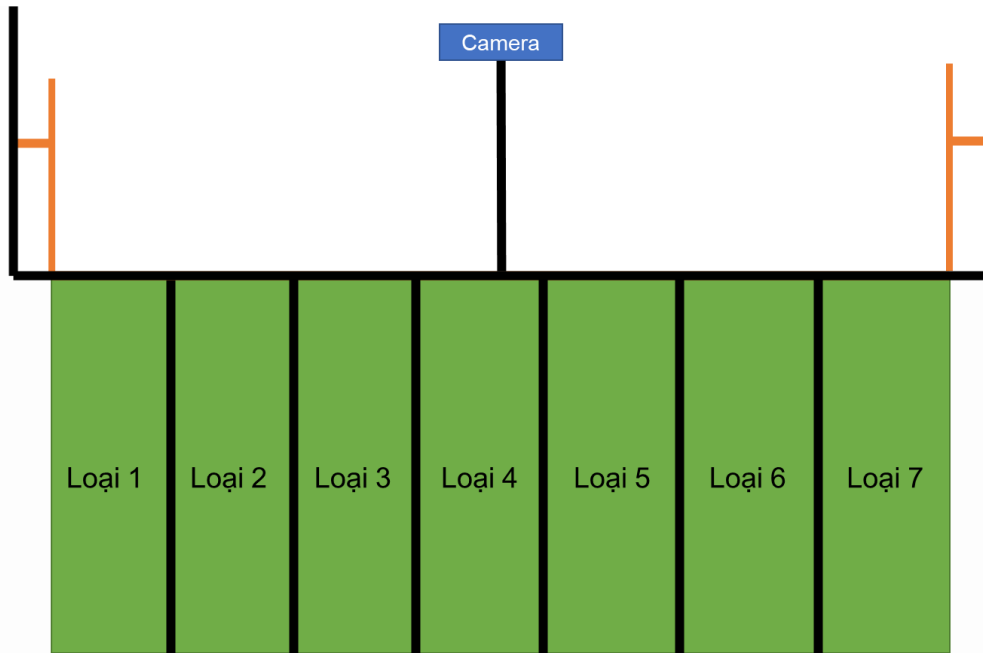
Tóm lại, việc phân loại rác thải sinh hoạt tại nguồn sẽ giúp:

- Giảm lượng rác thải ra môi trường - tiết kiệm chi phí thu gom, vận chuyển, xử lý
- Giảm lượng rác chôn lấp - tiết kiệm tài nguyên đất; giảm ô nhiễm môi trường
- Tăng lượng rác tái chế, tái sử dụng - tận dụng, tiết kiệm tài nguyên; mang lợi ích kinh tế cho gia đình; gây quỹ cho hoạt động cộng đồng;
- Nâng cao nhận thức về bảo vệ môi trường và sử dụng hợp lý tài nguyên;
- Là trách nhiệm của mỗi cá nhân đối với xã hội.

Vốn là một người quan tâm đến vấn đề môi trường, bạn Thanh Mai có mong muốn phân loại rác thải trong nhà bạn để từ đó có thể đem tái chế rác thải và làm phân hữu cơ cho cây của bạn. Vấn đề đặt ra là mọi người trong gia đình bạn rất bận, thế nên bạn cũng không thể bắt mọi người phải tốn thời gian để thực hiện thói quen sống xanh như bạn. Sắp tới cuối năm

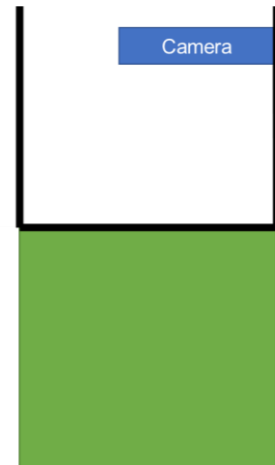
cũng là sinh nhật bạn Thanh Mai, nhóm quyết định chế tạo thùng rác thông minh tặng bạn để bạn có thể cùng gia đình sống “xanh”.

Để làm được điều đó, nhóm cần phải tạo được hệ thống phân loại rác tự động. Do đó, mục tiêu đề tài hướng đến việc xây dựng một mô hình máy học có thể phân loại rác cho chiếc thùng rác thông minh để có thể tự động phân loại rác tương tự như thiết kế bên dưới.



Hình 1: Thiết kế thùng rác bỏ dục chính diện

Chiếc thùng rác này được thiết kế không có nắp, với hai động cơ đẩy ở hai bên để đẩy rác về đúng ô phân loại. Camera được đặt ở giữa thùng và hướng từ trên xuống. Khi có rác được bỏ vào thùng, hệ thống sẽ tiến hành phân loại loại rác, sau đó nắp ô phân loại tương ứng sẽ mở để đợi các động cơ đẩy đẩy rác vào. Dựa trên setup này, nhóm sẽ xây dựng một hệ thống máy học có khả năng phân loại từng loại rác tương ứng với ô phân loại.



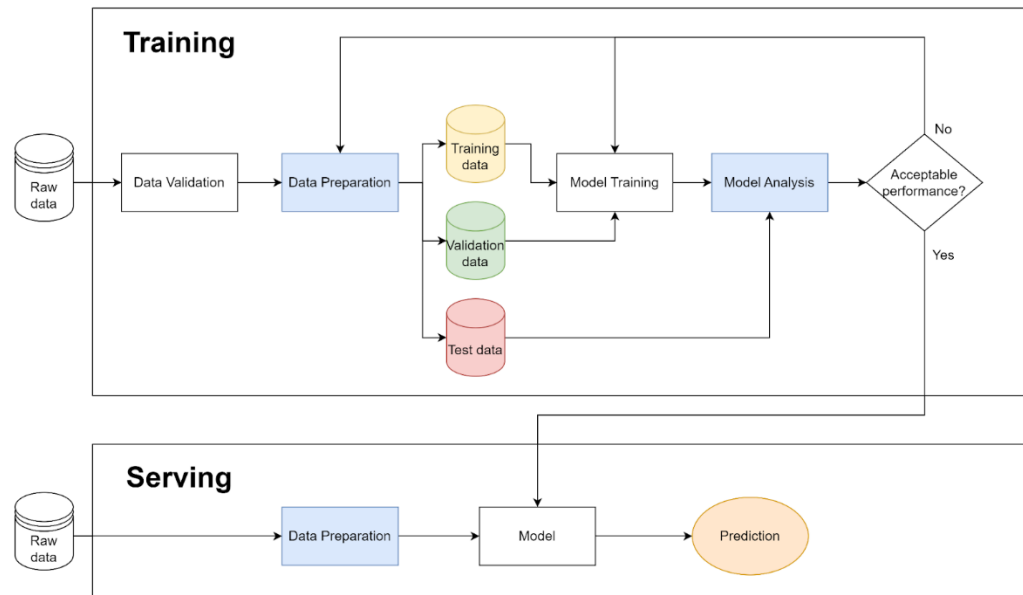
Hình 2: Bỏ dục từ bên qua

1.2 Input và output bài toán

- **Input:** Một ảnh chụp một vật thể rác trên nền trắng
- **Output:** Tên loại rác có trong ảnh

1.3 Hướng đi của nhóm

Để xây dựng nên model máy học như mong muốn, nhóm đã thiết kế và thực hiện theo pipeline như sau:



Hình 3: Machine Learning pipeline

1.4 Cách đánh giá performance

Vì việc phân loại rác cần sự chính xác cao, nếu không thì việc phân loại sẽ vô nghĩa, do đó, nhóm sử dụng metric chính là Accuracy. Với 7 class, kỳ vọng cho Accuracy tối thiểu xấp xỉ 0.15.

Tuy nhiên, việc phân loại chuẩn từng loại rác cũng là một vấn đề quan trọng, do đó, nhóm cũng sẽ quan tâm về Recall và Precision của từng lớp.

1.5 Bộ dữ liệu

Hướng đi của nhóm là kết hợp một số bộ datasets đã có lại với nhau và bổ sung thêm data tự thu thập để tạo nên bộ dataset cho nhóm. Sau khi tham khảo qua một số dataset liên quan đến bài toán, nhóm nhận thấy bộ dữ liệu [Trashnet](#) của tác giả Gary Thung và Mindy Yang kết hợp với một số dữ liệu thêm từ GitHub [Waste-or-Garbage-Classification-Using-Deep-Learning-Dataset · deepak2233 · GitHub](#), cũng được xây dựng dựa trên Trashnet, sẽ rất phù hợp cho thiết kế thùng rác của nhóm vì có background

đã được làm sạch. Bên cạnh hai dataset này, nhóm cũng đã tìm thấy một số dataset khác nhưng tuy nhiên những dataset đó có sử dụng thêm ảnh crawl từ trên mạng (Bing, Google) hoặc những dataset đó phục vụ cho bài toán khác, có thể sẽ không phù hợp với mục tiêu nhóm hướng đến.

Sau khi tìm hiểu, các tác giả của Trashnet xây dựng dataset này như sau: Các tấm ảnh được chụp bằng cách đặt vật thể lên trên một nền màu trắng, sử dụng ánh sáng mặt trời hoặc ánh sáng trong phòng. Các vật thể có thể được chụp từ nhiều góc độ, sau đó các tấm ảnh được resize về cùng một size bằng script do tác giả tạo ra.

Thiết bị họ sử dụng: Apple iPhone 7 Plus, Apple iPhone 5S, và Apple iPhone SE.

Dựa trên đó, nhóm sẽ thu thập và bổ sung vào bộ dataset này.

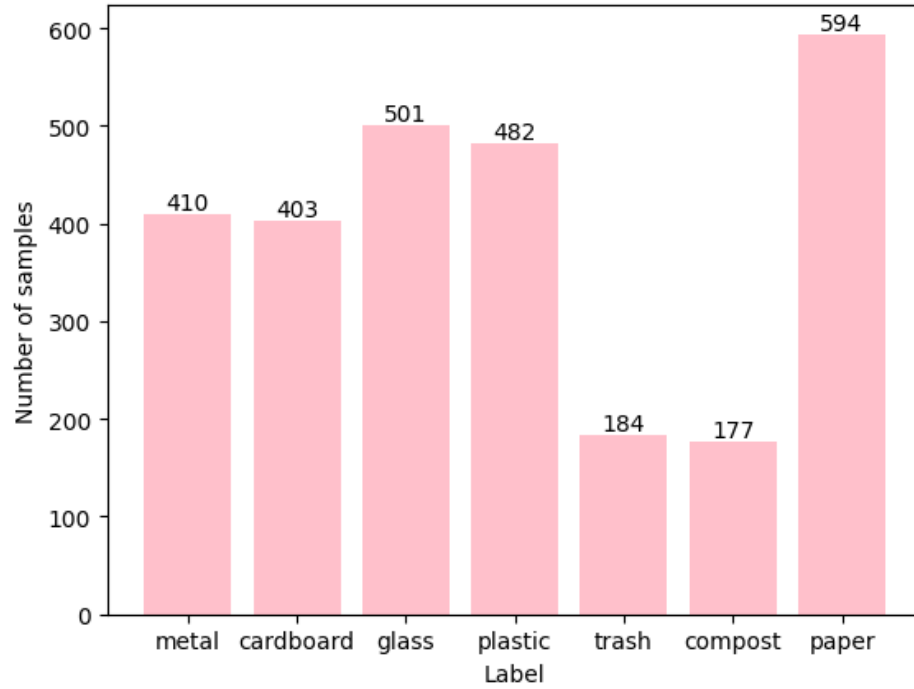
Thiết bị nhóm sử dụng: Redmi Note 9S, Blackberry Evolve, Apple iPhone 6S Plus, Apple iPhone 8S Plus và Apple iPhone XS Max.

Nhóm sẽ tiến hành kết hợp hai bộ dữ liệu lại với nhau và resize tất cả ảnh về 300x400 pixel bằng script resize tác giả cung cấp. Tuy nhiên có một vấn đề là do module sciPy tác giả sử dụng đã quá cũ, một số hàm đã không còn hoạt động, do đó nhóm đã chuyển sang dùng OpenCV và điều chỉnh tham số để giống như tác giả đã sử dụng sciPy.

1.6 Kết quả phân tích bộ dữ liệu

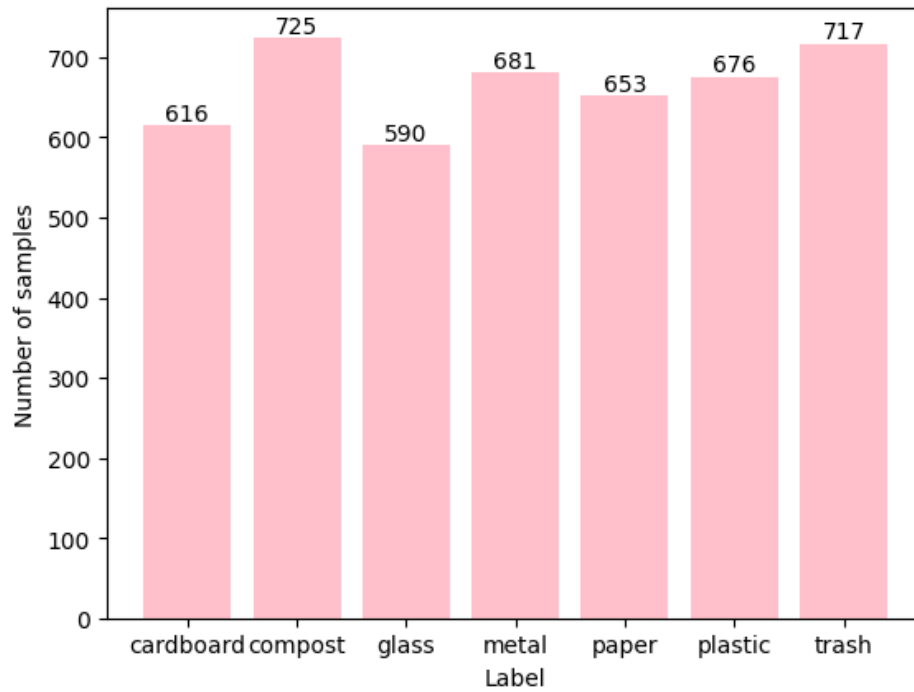
Sau khi thực hiện kết hợp hai bộ dữ liệu từ hai nguồn (chưa tính phần của nhóm), xóa bỏ trùng lặp và resize ảnh, nhóm thu được bộ dữ liệu chứa tất cả 2751 ảnh thuộc 7 class.

- metal: vật thể làm bằng kim loại (thau, lon, inox,...)
- cardboard: vật thể làm bằng bìa giấy cứng (thùng giấy)
- glass: vật thể làm bằng thủy tinh (chai, lọ thủy tinh, ly thủy tinh,...)
- plastic: vật thể làm bằng nhựa có thể tái chế (chai nhựa, thùng nhựa,...)
- compost: vật thể là rác hữu cơ (đồ ăn thừa, bánh, kẹo, rau củ,...)
- paper: vật thể làm bằng giấy (giấy báo, tờ rơi, giấy,...)
- trash: vật thể khác như nilon, toothpaste, kết hợp giữa nhiều loại ở trên...

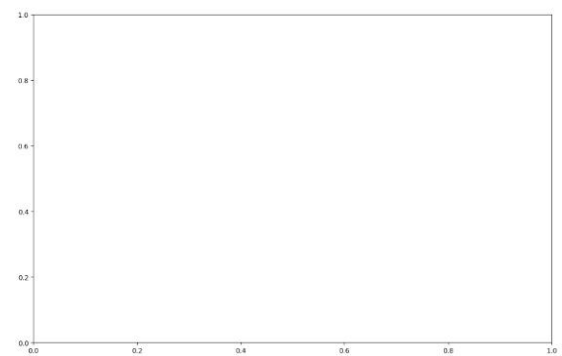
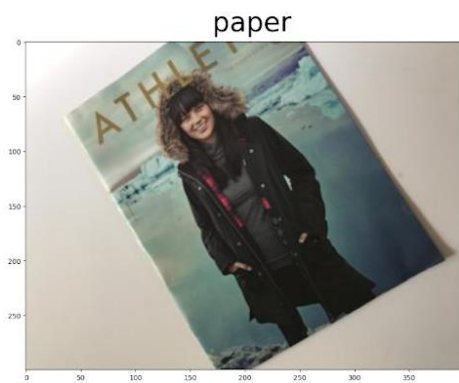
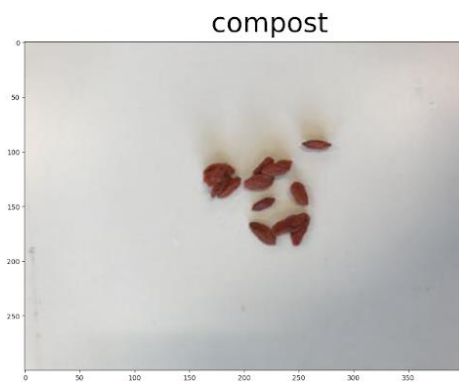
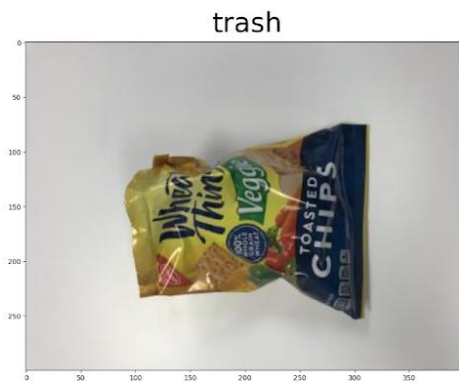
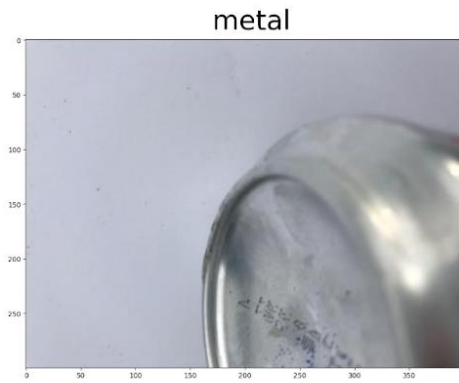


Hình 4: Phân phối bộ dữ liệu của hai bộ dataset

Sau khi nhóm thêm dataset nhóm tự thu thập thêm, tổng số lượng tăng lên 4658 ảnh cho 7 class. Số lượng ảnh mỗi class phân bố như ảnh sau khi nhóm thêm dữ liệu tự thu thập:



Hình 5: Phân phối bộ dữ liệu sau cùng của nhóm



Hình 6: Một số mẫu dữ liệu

II. Chia tập train/val/test

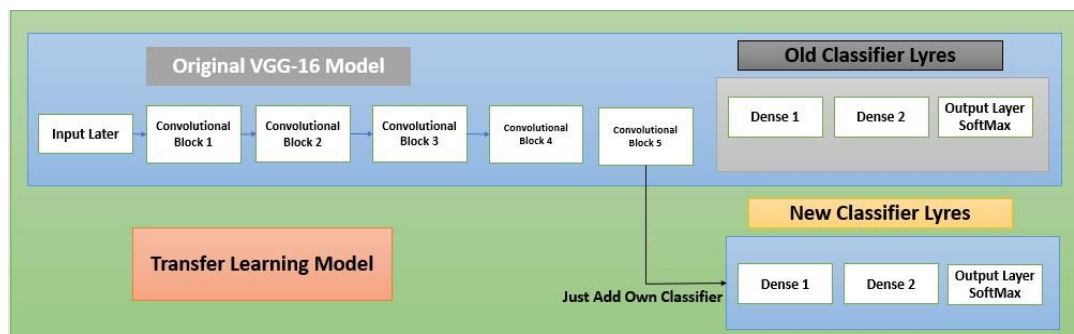
- Thực hiện split dataset thành 3 tập train/val/test với tỉ lệ tương ứng lần lượt là 70%/15%/15%.
- Trong đó:
 - Tập train được dùng để huấn luyện model.
 - Tập validation được dùng để validate model được huấn luyện, cũng từ đó giúp ta thực hiện hyperparameter tuning và tránh overfitting.
 - Tập test sẽ được dùng để test đánh giá performance của model.

III. Training và đánh giá model

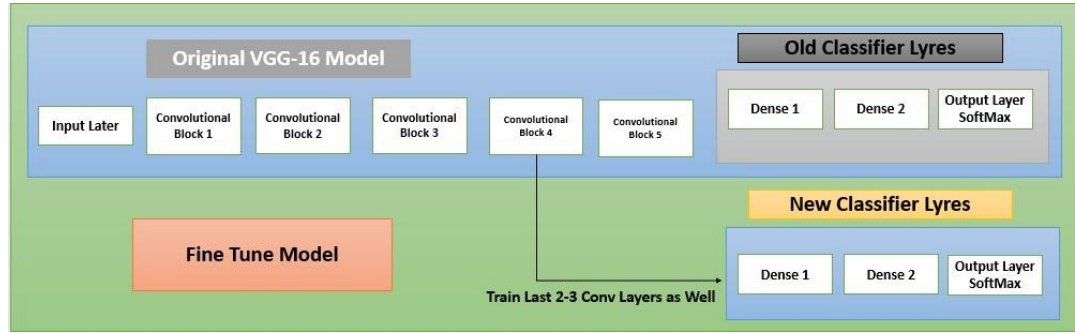
3.1 Phân tích hướng đi và kết quả những người đi trước

Hướng đi của [GitHub - deepak2233/Waste-or-Garbage-Classification-Using-Deep-Learning: This model is created using pre-trained CNN architecture \(VGG16 and RESNET50\) via Transfer Learning that classifies the Waste or Garbage material \(class labels =7\) for recycling.](#):

- Về bước Data Preparation: họ chỉ thực hiện Data Augmentation với bộ dataset đã resize.
- Về bước Training:
 - Họ thực hiện train ResNet50, VGG16 model trên bộ dataset thông qua việc áp dụng kỹ thuật Transfer Learning của CNN cho bài toán Classification.
 - Họ thực hiện cải thiện VGG16 model thông qua việc áp dụng kỹ thuật Transfer Learning và Fine Tuning.
 - Pipeline kỹ thuật Transfer Learning và Fine Tuning của họ:



Hình 7: Kiến trúc VGG16 Transfer Learning



Hình 8: Kiến trúc VGG16 train lại Block 4, 5

- Về Evaluation Metric: họ chỉ sử dụng thông số accuracy để đánh giá performance. Kết quả của họ sau khi train và test:

Bảng 1: Bảng kết quả của Deepak2233

Architecture	ResNet50	VGG16	VGG16	VGG16
Technique	Transfer Learning	Transfer Learning	Transfer Learning	Fine Tuning
Number of epochs	5	5	20	22
Train Accuracy	29.93%	41.3%	93.03%	95.14%
Validation Accuracy	28.37%	42.02%	81.46%	81.69%
Test Accuracy	29.78%	43.09%	72.52%	80.86%

3.2 Tổng quan các model của nhóm

3.2.1 Support Vector Machine (SVM)

3.2.1.1 Tổng quan

Support Vector Machine là một thuật toán máy học có giám sát (supervised learning), được tạo ra bởi Vladimir N. Vapnik và Alexey Ya. Chervonenkis. Ý tưởng chính của SVM là tìm một siêu phẳng (hyperplane) tối đa hoá sự phân tách giữa các lớp dữ liệu khác nhau trong tập huấn luyện. Điều này được thực hiện bằng cách tìm siêu phẳng có độ rộng lớn nhất, được xác định là khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất của mỗi lớp. Khi siêu phẳng đã được xác định, dữ liệu mới có thể được phân loại bằng cách xác định nó thuộc về phía nào của siêu phẳng. SVM

đặc biệt hữu ích khi dữ liệu có nhiều đặc trưng và/hoặc có ranh giới phân tách rõ ràng giữa các lớp.

Nhóm dự định sẽ áp dụng SVM - một traditional machine learning và thực hiện đánh giá performance của nó. Đây cũng là tiền đề để nhóm xem xét có nên sử dụng các model CNN hay không.

3.2.1.2 Các bước train

- Data Preparation: Chỉ thực hiện split train/val/test. Mục đích của việc split đã được thể hiện ở phần II.
- Feature Engineering: Thực hiện 2 phương pháp riêng biệt. Với mục đích chung là có thể extract important features, giảm thời gian training.
 - PCA (Principal Component Analysis): là một phương pháp giảm chiều dữ liệu bằng cách đi tìm một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài tọa độ, bỏ qua những thành phần ít quan trọng nhất. Tuy nhiên, phương pháp này có một số nhược điểm trong quá trình xử lý, được thể hiện rõ trong notebook.
 - SIFT (Scale-invariant feature transform): với đầu vào là một hình ảnh >> SIFT >> các keypoint. Mỗi đối tượng trong hình ảnh sẽ cho ra rất nhiều các keypoint khác nhau, để ta phân biệt được các keypoint này với nhau sẽ thông qua một vector 128 chiều hay còn gọi là descriptor. Các descriptor này sẽ được dùng để nhận dạng đối tượng trong ảnh, hay dùng cho các bài toán classification.
- Hyperparameter Tuning: Dùng GridSearchCV để tìm ra bộ tham số tốt nhất của model. Ở đây, nhóm thực hiện phương pháp Cross-Validation để validate.
- Training: Với bộ tham số tìm được ở bước Hyperparameter Tuning, thực hiện train model dựa trên training set.
- Prediction: dùng model đã được tạo ra ở bước Training để thực hiện dự toán với test set.

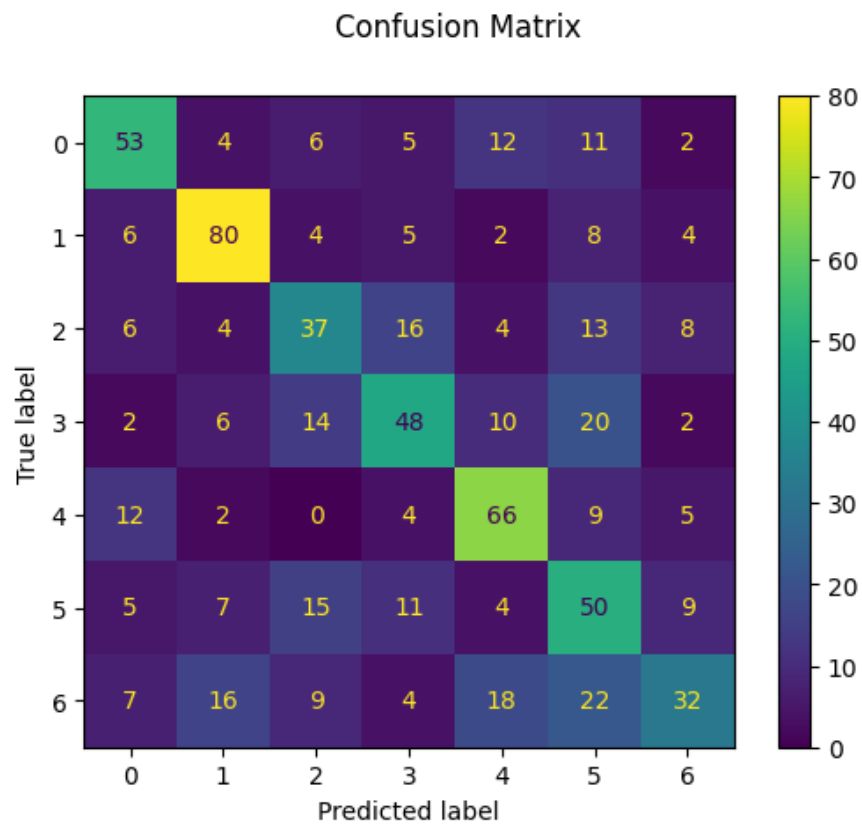
3.2.1.3 Đánh giá performance

Nhóm sử dụng các metric accuracy, precision, recall, F1 score để đánh giá model này. Cách thực hiện và kết quả đánh giá được trình bày chi tiết trong phần model analysis trong notebook.

```
Classification report for classifier SVC(C=10, gamma=0.001):
```

	precision	recall	f1-score	support
0	0.58	0.57	0.58	93
1	0.67	0.73	0.70	109
2	0.44	0.42	0.43	88
3	0.52	0.47	0.49	102
4	0.57	0.67	0.62	98
5	0.38	0.50	0.43	101
6	0.52	0.30	0.38	108
accuracy			0.52	699
macro avg	0.52	0.52	0.52	699
weighted avg	0.53	0.52	0.52	699

Hình 9: Classification report của model SVM sử dụng SIFT technique

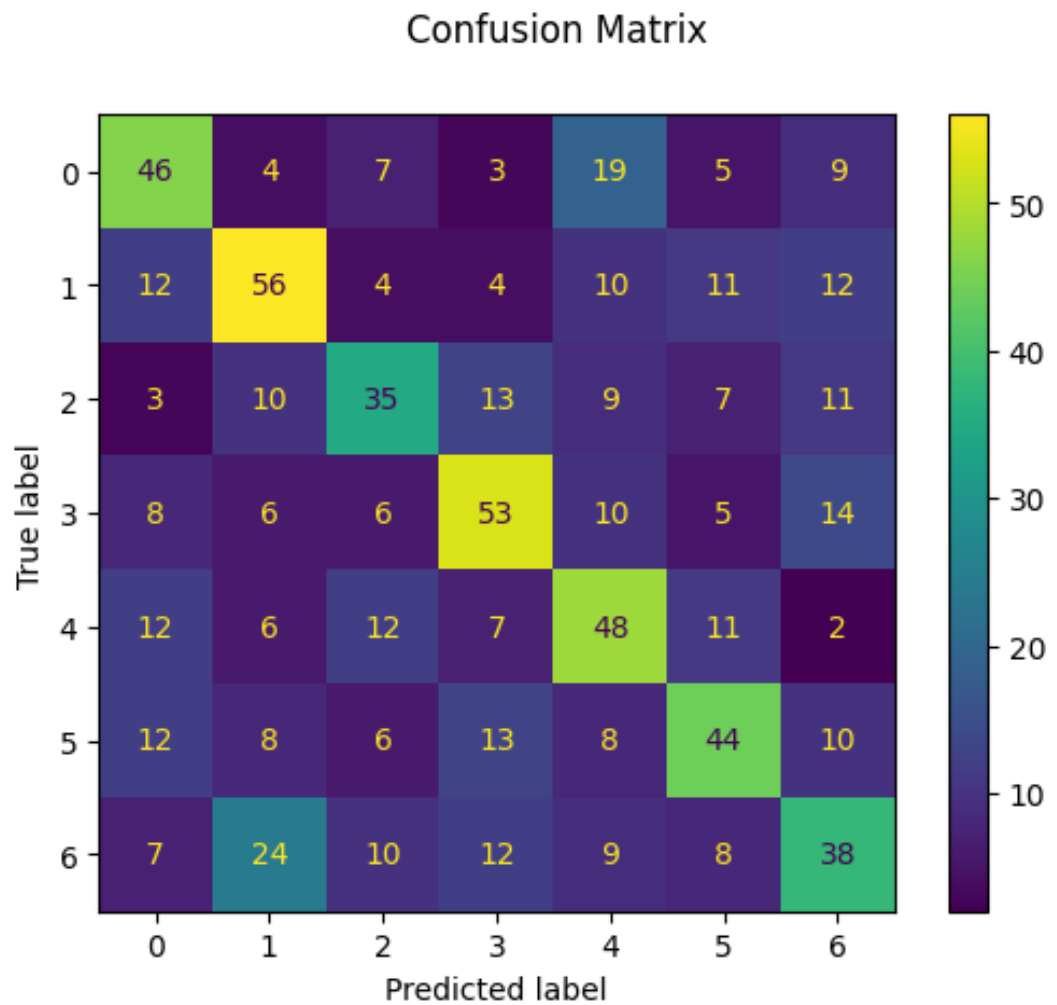


Hình 10: Confusion matrix của model SVM sử dụng SIFT technique

Classification report for classifier SVC(C=10, gamma=1e-08):

	precision	recall	f1-score	support
0	0.46	0.49	0.48	93
1	0.49	0.51	0.50	109
2	0.44	0.40	0.42	88
3	0.50	0.52	0.51	102
4	0.42	0.49	0.45	98
5	0.48	0.44	0.46	101
6	0.40	0.35	0.37	108
accuracy			0.46	699
macro avg	0.46	0.46	0.46	699
weighted avg	0.46	0.46	0.46	699

Hình 11: Classification report của model SVM sử dụng PCA technique



Hình 12: Confusion matrix của model SVM sử dụng PCA technique

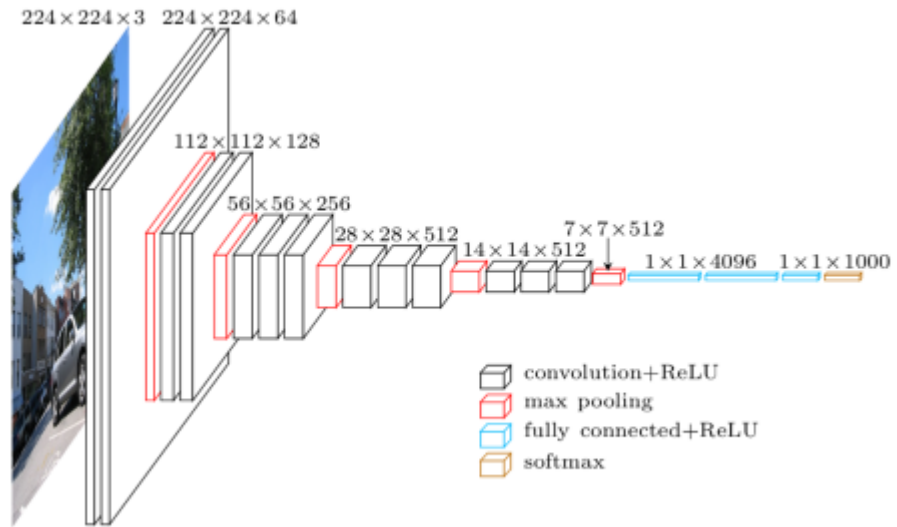
Tóm lại, accuracy của model SVM là không cao bằng so với 2 model cải thiện VGG16 của được đề cập ở 3.1. Nguyên nhân có thể do giai đoạn Feature Extraction với PCA và SIFT của nhóm đã loại bỏ đi quá nhiều dữ liệu. Việc thêm dữ liệu cũng không thể vì nó khiến chi phí tính toán tăng cao, nằm ngoài khả năng tính toán của nhóm. Với kết quả mà model SVM mang lại vẫn chưa thật sự tốt, nhóm tiếp tục thực hiện một số thí nghiệm với CNN.

3.2.2 VGG16

3.2.2.1 Tổng quan

VGG là viết tắt của Visual Geometry Group; đó là một kiến trúc Convolutional Neural Network (CNN) tiêu chuẩn với nhiều lớp. "Deep" liên quan đến số lượng lớp như VGG-16 hoặc VGG-19 có nghĩa là kiến trúc này có 16 và 19 lớp.

Mô hình VGG, hoặc VGGNet, hỗ trợ 16 lớp còn được gọi là VGG16, đây là mô hình CNN được đề xuất bởi A. Zisserman và K. Simonyan từ Đại học Oxford trong bài báo "[Very Deep Convolutional Networks for Large-Scale Image Recognition](#)". Mô hình VGG16 đạt được gần 92,7% độ chính xác thử nghiệm top-5 trên ImageNet. ImageNet là một tập dữ liệu gồm hơn 14 triệu hình ảnh thuộc gần 1000 lớp khác nhau. Nó sử dụng phép tích chập với kernel kích thước 3x3 liên tiếp nhau, same padding và Max Pool 2x2, stride = 2, do đó làm cải thiện đáng kể số lượng siêu tham số cần fine tune so với AlexNet.



Hình 13: Kiến trúc VGG16

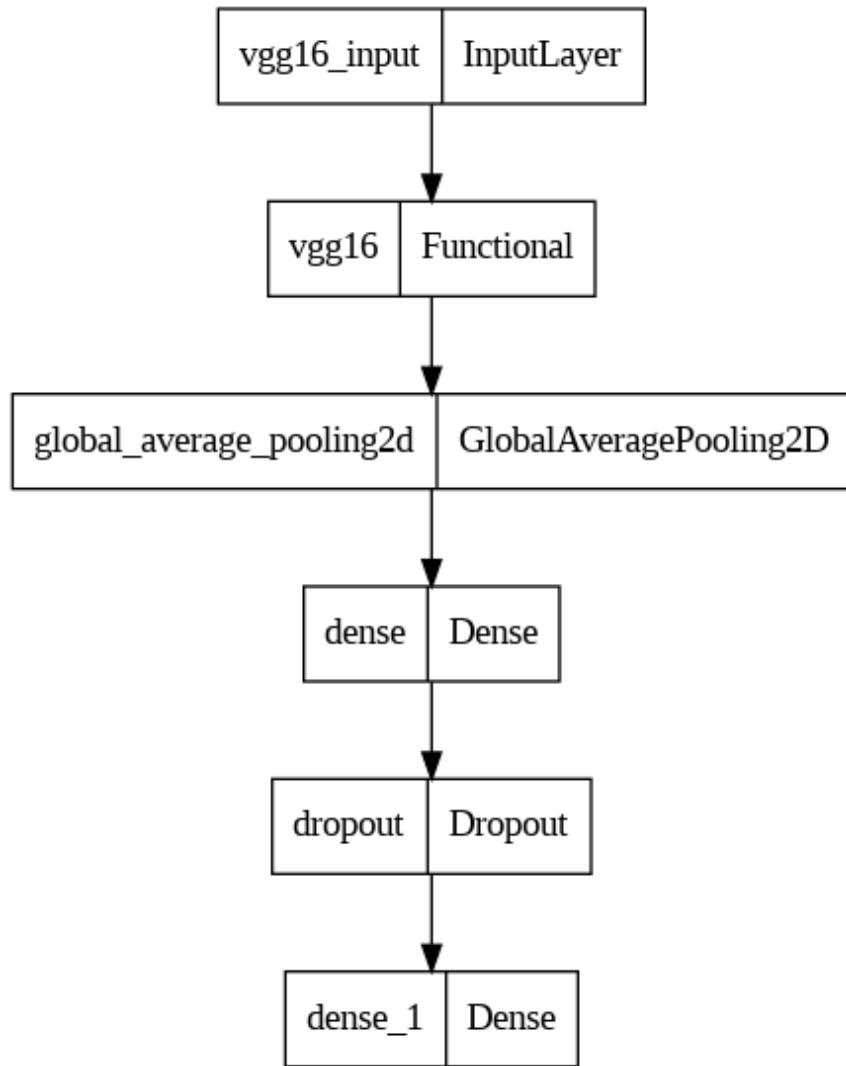
Với mô hình này, đi càng sâu thì kích thước 2D càng giảm, trong khi số lượng kernels tăng lên một cách có hệ thống. Tuy nhiên kiến trúc có thể gặp phải vấn đề gradient vanishing.



Hình 14: Vanishing Gradient

3.2.2.2 Các bước train

- Thiết kế model: thực hiện transfer learning VGG16 của Tensorflow và thêm vào ba lớp để dần giảm chiều về layer output cho 7 lớp. Freeze tất cả các layer của VGG16.



Hình 15: Mô hình VGG16 được thiết kế

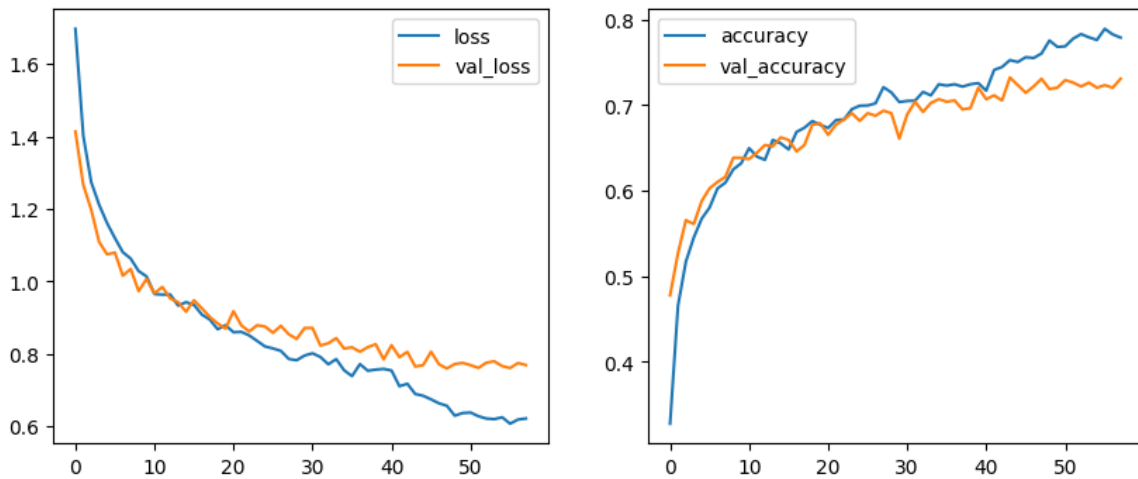
- Training: Thực hiện train 100 epochs với hàm loss Adam cho bài toán phân loại nhiều lớp. Thêm cơ chế dừng sớm dựa trên kết quả tập validation và cơ chế tự động giảm learning rate nếu sắp gặp phải hiện tượng gradient vanishing trên tập validation.
- Prediction: Dùng model đã được tạo ra ở bước Training để thực hiện dự toán với test set.
- Tuning: Thực hiện unfreeze một số hoặc toàn bộ layer và train lại.

3.2.2.3 Đánh giá performance

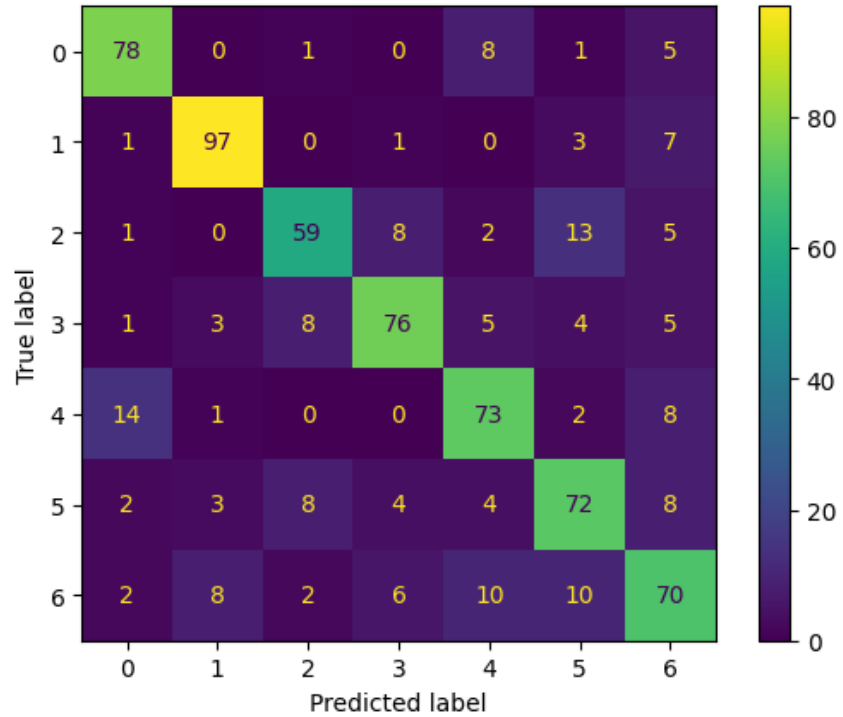
Dựa trên hình 15, 16, có thể thấy sau 58 epochs, model dần trở overfitting với tập train.

```
Epoch 51/100
101/101 [=====] - ETA: 0s - loss: 0.6380 - accuracy: 0.7686
Epoch 51: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
101/101 [=====] - 70s 691ms/step - loss: 0.6380 - accuracy: 0.7686 - val_loss: 0.7689 - val_accuracy: 0.7292 - lr: 2.5000e-04
Epoch 52/100
101/101 [=====] - 70s 690ms/step - loss: 0.6278 - accuracy: 0.7776 - val_loss: 0.7612 - val_accuracy: 0.7262 - lr: 1.2500e-04
Epoch 53/100
101/101 [=====] - 70s 693ms/step - loss: 0.6216 - accuracy: 0.7828 - val_loss: 0.7757 - val_accuracy: 0.7217 - lr: 1.2500e-04
Epoch 54/100
101/101 [=====] - ETA: 0s - loss: 0.6199 - accuracy: 0.7791
Epoch 54: ReduceLROnPlateau reducing learning rate to 6.25000029685907e-05.
101/101 [=====] - 69s 679ms/step - loss: 0.6199 - accuracy: 0.7791 - val_loss: 0.7794 - val_accuracy: 0.7262 - lr: 1.2500e-04
Epoch 55/100
101/101 [=====] - 69s 686ms/step - loss: 0.6250 - accuracy: 0.7760 - val_loss: 0.7663 - val_accuracy: 0.7202 - lr: 6.2500e-05
Epoch 56/100
101/101 [=====] - 69s 682ms/step - loss: 0.6073 - accuracy: 0.7890 - val_loss: 0.7608 - val_accuracy: 0.7232 - lr: 6.2500e-05
Epoch 57/100
101/101 [=====] - ETA: 0s - loss: 0.6188 - accuracy: 0.7825
Epoch 57: ReduceLROnPlateau reducing learning rate to 3.125000148429535e-05.
101/101 [=====] - 69s 679ms/step - loss: 0.6188 - accuracy: 0.7825 - val_loss: 0.7747 - val_accuracy: 0.7202 - lr: 6.2500e-05
Epoch 58/100
101/101 [=====] - 69s 682ms/step - loss: 0.6216 - accuracy: 0.7788 - val_loss: 0.7692 - val_accuracy: 0.7307 - lr: 3.1250e-05
```

Hình 16: Kết quả training 7 epochs cuối



Hình 17: Learning curve



Hình 18: Confusion matrix

	precision	recall	f1-score	support
cardboard	0.79	0.84	0.81	93
compost	0.87	0.89	0.88	109
glass	0.76	0.67	0.71	88
metal	0.80	0.75	0.77	102
paper	0.72	0.74	0.73	98
plastic	0.69	0.71	0.70	101
trash	0.65	0.65	0.65	108
accuracy			0.75	699
macro avg	0.75	0.75	0.75	699
weighted avg	0.75	0.75	0.75	699

Hình 19: Kết quả phân loại

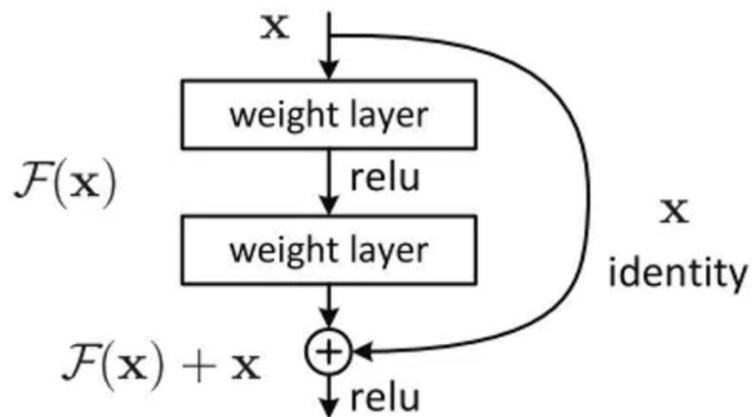
Tuy nhiên, có thể thấy performance được cải thiện rõ rệt khi dùng model VGG16 so với model SVM, mặc dù cả 2 model cùng được build từ chung bộ dataset. Đồng thời, precision, recall trên từng class cũng đã tốt hơn, tuy nhiên có sự chênh lệch khá cao giữa các class.

3.2.3 ResNet50

3.2.3.1 Tổng quan

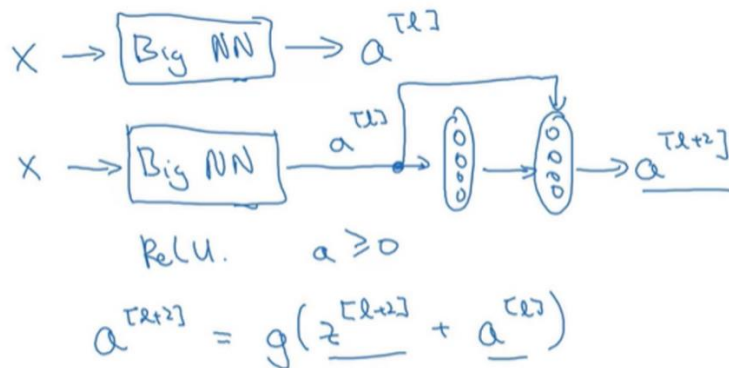
ResNet, viết tắt của Residual Network, là một loại Convolutional neural network đã được giới thiệu vào năm 2015 bởi Kaiming He, Xiangyu Zhang, Shaoqing Ren và Jian Sun trong bài báo "[Deep Residual Learning for Image Recognition](#)" của họ.

Khác với VGG, các kiến trúc ResNet được tạo nên từ những residual blocks. Kiến trúc có thêm những shortcut hay skipped connections để đưa input từ lớp hiện tại tới những layer phía trước mà không cần phải đi qua các bước trung gian.



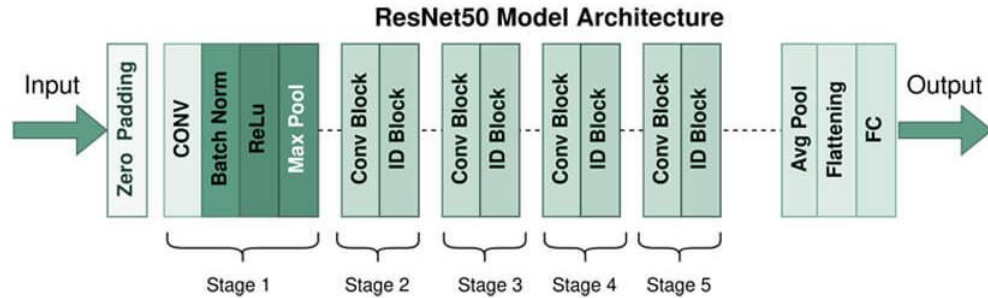
Hình 20: Residual Block

Nhờ cơ chế này, ResNet giúp tăng thêm thông tin cho các lớp phía sau, từ đó không gặp phải các vấn đề như gradient vanishing hay giảm performance.



Hình 21: Mô phỏng shortcut

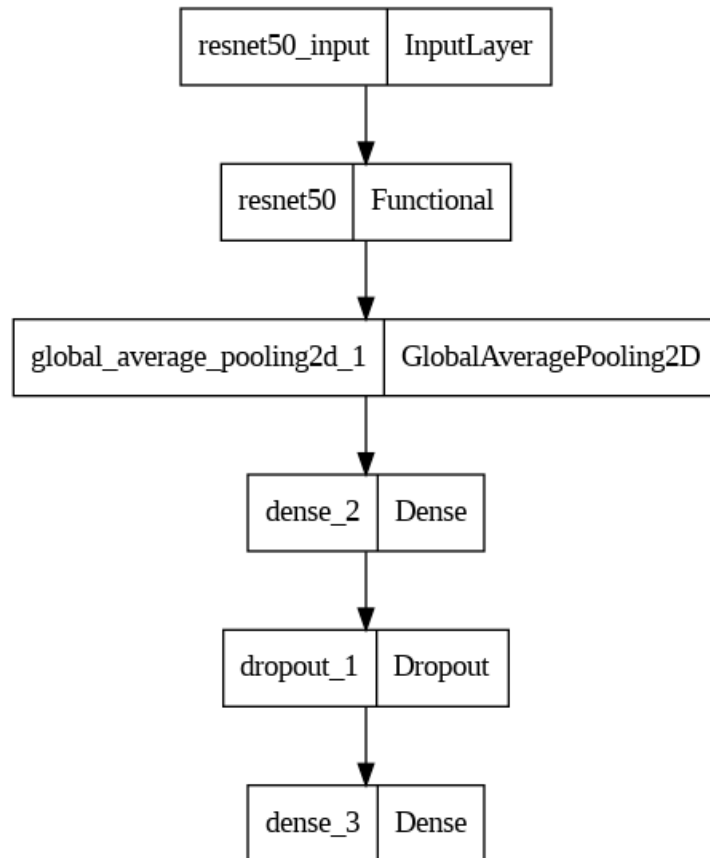
ResNet có nhiều biến thể trên cùng một khái niệm nhưng có số lượng lớp khác nhau. Resnet50 được sử dụng để chỉ phiên bản có thể hoạt động với 50 lớp mạng neural.



Hình 22: Kiến trúc ResNet50

3.2.3.2 Các bước train

- Thiết kế model: thực hiện transfer learning ResNet50 của Tensorflow và thêm vào ba lớp để dần giảm chiều về layer output cho 7 lớp. Freeze tất cả các layer của ResNet50.



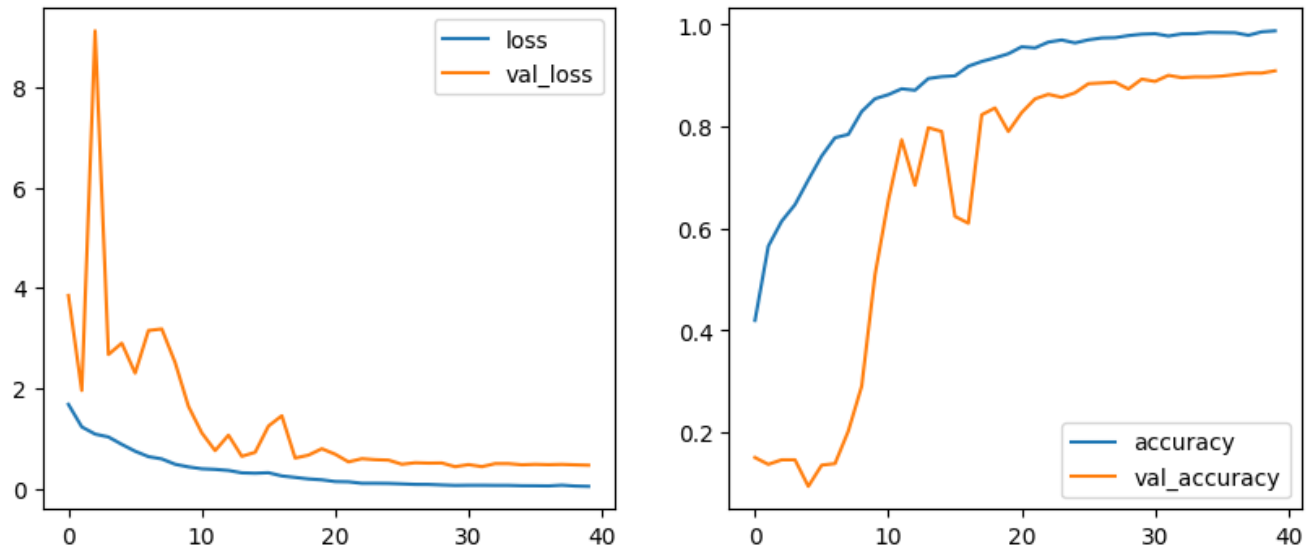
Hình 23: Mô hình ResNet50 được thiết kế

- Training: Thực hiện train 50 epochs với hàm loss Adam cho bài toán phân loại nhiều lớp. Thêm cơ chế dừng sớm dựa trên kết quả tập validation và cơ chế tự động giảm learning rate nếu sắp gặp phải hiện tượng gradient vanishing.
- Prediction: dùng model đã được tạo ra ở bước Training để thực hiện dự toán với test set.
- Tuning: Thực hiện unfreeze một số hoặc tất cả layer của ResNet50 và train lại.

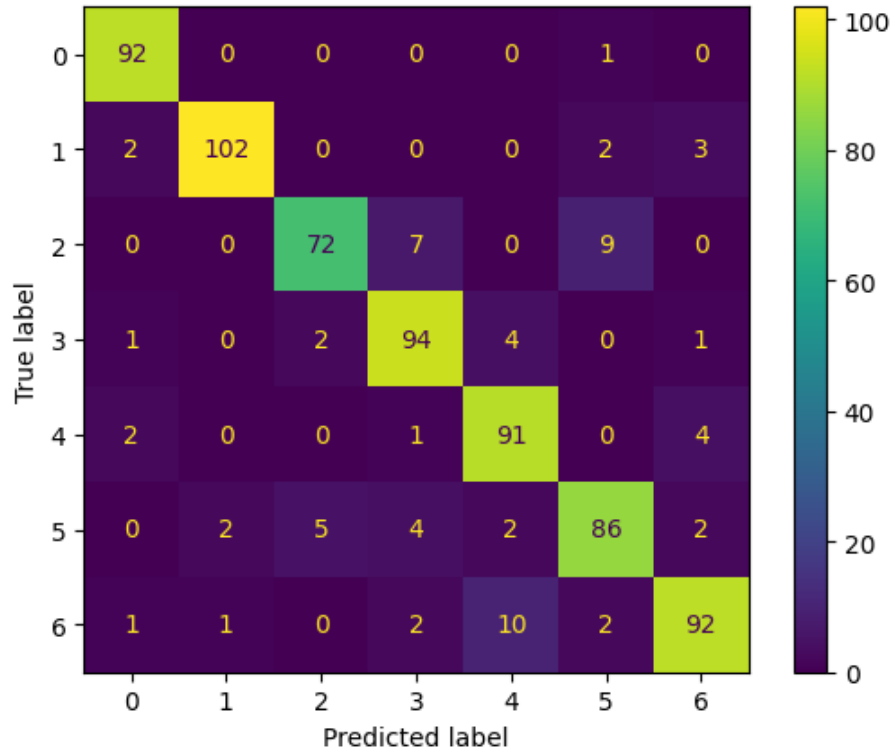
3.2.3.3 Đánh giá performance

```
Epoch 34/50
101/101 [=====] - 75s 745ms/step - loss: 0.0602 - accuracy: 0.9820 - val_loss: 0.4935 - val_accuracy: 0.8973 - lr: 7.8125e-06
Epoch 35/50
101/101 [=====] - 74s 731ms/step - loss: 0.0534 - accuracy: 0.9845 - val_loss: 0.4706 - val_accuracy: 0.8973 - lr: 7.8125e-06
Epoch 36/50
101/101 [=====] - ETA: 0s - loss: 0.0521 - accuracy: 0.9842
Epoch 36: ReduceLROnPlateau reducing learning rate to 3.906250185536919e-06.
101/101 [=====] - 75s 741ms/step - loss: 0.0521 - accuracy: 0.9842 - val_loss: 0.4790 - val_accuracy: 0.8988 - lr: 7.8125e-06
Epoch 37/50
101/101 [=====] - 73s 723ms/step - loss: 0.0494 - accuracy: 0.9839 - val_loss: 0.4726 - val_accuracy: 0.9018 - lr: 3.9063e-06
Epoch 38/50
101/101 [=====] - 74s 730ms/step - loss: 0.0639 - accuracy: 0.9789 - val_loss: 0.4781 - val_accuracy: 0.9048 - lr: 3.9063e-06
Epoch 39/50
101/101 [=====] - ETA: 0s - loss: 0.0466 - accuracy: 0.9857
Epoch 39: ReduceLROnPlateau reducing learning rate to 1.9531250927684596e-06.
101/101 [=====] - 75s 739ms/step - loss: 0.0466 - accuracy: 0.9857 - val_loss: 0.4696 - val_accuracy: 0.9048 - lr: 3.9063e-06
Epoch 40/50
101/101 [=====] - 75s 737ms/step - loss: 0.0403 - accuracy: 0.9876 - val_loss: 0.4626 - val_accuracy: 0.9092 - lr: 1.9531e-06
```

Hình 24: Kết quả train 7 epochs cuối



Hình 25: Learning curve



Hình 26: Confusion matrix

	precision	recall	f1-score	support
cardboard	0.94	0.99	0.96	93
compost	0.97	0.94	0.95	109
glass	0.91	0.82	0.86	88
metal	0.87	0.92	0.90	102
paper	0.85	0.93	0.89	98
plastic	0.86	0.85	0.86	101
trash	0.90	0.85	0.88	108
accuracy			0.90	699
macro avg	0.90	0.90	0.90	699
weighted avg	0.90	0.90	0.90	699

Hình 27: Kết quả phân loại

Có thể thấy rằng performance của ResNet50 được cải thiện đáng kể so với VGG16. Accuracy của ResNet50 lên đến 0.9 so với VGG16 là 0.75. Tuy nhiên, performance giữa các class của ResNet50 cũng có sự chênh lệch nhau tương tự như VGG16 nhưng không đáng kể.

IV. Kết luận

Trong đồ án này, nhóm đã thực hiện các thử nghiệm để đánh giá performance của các phương pháp phân loại ảnh rác thải sử dụng các model machine learning và convolutional deep learning, cụ thể là các model SVM, neural network dựa trên kiến trúc VGG16 và kiến trúc ResNet50. Có thể thấy performance của ResNet50 tốt nhất trong số các model nhóm sử dụng với accuracy 0.9, precision và recall rate đều trên mức 0.8. Trong khi đó, hai model SVM có performance khá tệ với accuracy chỉ khoảng 0.5.

Bảng 2: Bảng kết quả performance của các model

Model	Precision	Recall	Accuracy
SVM with SIFT	0.52	0.52	0.52
SVM with PCA	0.46	0.46	0.46
VGG16	0.75	0.75	0.75
ResNet50	0.90	0.90	0.90

V. Hướng phát triển trong tương lai

Về hướng phát triển trong tương lai, nhóm nhận thấy rằng do nhóm chỉ mới sử dụng ba mô hình phân loại nên chưa thể đánh giá toàn diện hiệu quả của các phương pháp khác nhau. Do đó, trong tương lai, nhóm hi vọng có thể thử nghiệm được nhiều phương pháp nữa.

Bên cạnh model, nhóm cũng sẽ tăng thêm số lượng và sự đa dạng của data để nó trở nên tổng quát hoá hơn với dữ liệu thực tế.

Bên cạnh đó, nhóm cũng sẽ thêm hai metrics đánh giá model nữa là về tốc độ và kích thước model. Vì hai model deep learning trên khá nặng, với VGG16 là khoảng 80MB, ResNet50 là khoảng 250MB, để giải pháp có thể đưa vào thực tế, nhóm sẽ cần một model có tốc độ infer nhanh hơn và kích thước nhẹ hơn so với các model trên để có thể nhúng vào bo mạch.

Tóm lại, đề tài nghiên cứu phân loại rác thải của nhóm dựa trên học máy nói riêng và trí tuệ nhân tạo nói chung đã đạt được những kết quả khả quan và mở ra nhiều hướng phát triển tiếp theo. Nhóm hi vọng những nghiên cứu trong đồ án này sẽ có thể ứng dụng được và hữu dụng trong thực tế trong một tương lai không xa. Ít nhất là có thể giúp bạn Mai thực hiện được mong muốn sống “xanh” cùng với gia đình bạn ấy như một món quà ý nghĩa cho ngày sinh nhật.

VI. Bổ sung và đánh giá

6.1 Bổ sung kết quả thí nghiệm

Để thí nghiệm, đầu tiên, nhóm thực hiện feature extraction với một pretrained model CNN, sau đó cho qua một lớp Global Average Pooling để biến từ 3D về 1D. Đây sẽ chính là kết quả feature extraction. Sau khi tuning SVM và train, có thể thấy performance tốt hơn hẳn so với dùng PCA hay SIFT.

```
Classification report for classifier SVC(C=100, gamma=0.01):
```

	precision	recall	f1-score	support
cardboard	0.84	0.85	0.84	93
compost	0.81	0.85	0.83	109
glass	0.79	0.75	0.77	88
metal	0.77	0.75	0.76	102
paper	0.76	0.79	0.77	98
plastic	0.66	0.69	0.68	101
trash	0.63	0.58	0.61	108
accuracy			0.75	699
macro avg	0.75	0.75	0.75	699
weighted avg	0.75	0.75	0.75	699

Hình 28: Kết quả phân loại khi dùng SVM với MobileNetV2

```
Classification report for classifier SVC(C=10, gamma=0.001):
```

	precision	recall	f1-score	support
cardboard	0.92	0.90	0.91	93
compost	0.95	0.96	0.95	109
glass	0.88	0.78	0.83	88
metal	0.85	0.81	0.83	102
paper	0.71	0.90	0.79	98
plastic	0.85	0.84	0.85	101
trash	0.87	0.78	0.82	108
accuracy			0.86	699
macro avg	0.86	0.85	0.86	699
weighted avg	0.86	0.86	0.86	699

Hình 29: Kết quả phân loại của SVM dùng VGG16

```

Classification report for classifier SVC(C=10, gamma=0.001):
              precision    recall  f1-score   support

 cardboard      0.98      0.95      0.96         93
   compost      0.97      0.99      0.98        109
    glass       0.92      0.93      0.93         88
    metal       0.93      0.90      0.92        102
    paper       0.95      0.90      0.92         98
   plastic      0.93      0.92      0.93        101
    trash       0.85      0.93      0.89        108

 accuracy              0.93         699
 macro avg              0.93         699
weighted avg              0.93         699

```

Hình 30: Kết quả phân loại khi dùng SVM với ResNet50

Có thể thấy, dùng CNN cho giai đoạn Feature Extraction cho kết quả tốt hơn hẳn so với dùng PCA và SIFT. Trong đó, ResNet50 cho kết quả rất tốt.

6.2 Đánh giá

Trong phần bổ sung này, nhóm thực hiện thêm năm thí nghiệm và đánh giá performance giữa chín thí nghiệm mà nhóm thực hiện. Bên cạnh đó, nhóm cũng bổ sung thêm vào một thông tin nữa là tổng kích thước pipeline. Kết quả được tổng hợp ở Bảng 3 cho thấy SVM dùng ResNet50 và MobileNetV2 có performance tốt nhất với accuracy 0.93.

Bảng 3: Bảng so sánh performance mới

Model	Size (MB)	Precision	Recall	Accuracy
SVM with PCA	95.1	0.46	0.46	0.46
SVM with SIFT	7.1	0.52	0.52	0.52
SVM with MobileNetV2	37.4	0.75	0.75	0.75
SVM with VGG16	63.4	0.86	0.85	0.86
SVM with ResNet50	124.2	0.93	0.93	0.93
VGG16	54.6	0.75	0.75	0.75
ResNet50	261.1	0.90	0.90	0.90
MobileNet	39.3	0.91	0.91	0.91
MobileNetV2	30.8	0.93	0.93	0.93