

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
BỘ MÔN CƠ ĐIỆN TỬ



HCMUTE

BÁO CÁO CUỐI KỲ
TRÍ TUỆ NHÂN TẠO

GVHD: PGS. TS. Nguyễn Trường Thịnh

SVTH: Lê Nguyễn Trung MSSV:19146024

LỚP: ARIN337629_21_2_05CLC

Thủ Đức, thứ hai ngày 20 tháng 6 năm 2022

BÁO CÁO

HỌC KỲ II NĂM HỌC 2021-2022

**ĐỀ TÀI : PHÂN LOẠI RÁC THẢI THEO THỜI GIAN THỰC SỬ DỤNG
THUẬT TOÁN CONVOLUTIONAL NEURAL NETWORK**

Ghi chú:

- Tỷ lệ = 100%.
 - Sinh viên: Lê Nguyễn Trung 19146024
 - Thông tin liên lạc: 0989804092
-

Nhận xét của giảng viên

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày 17 tháng 6 năm 2022

MỤC LỤC

| | | |
|-------|--|----|
| 1 | CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI..... | 1 |
| 1.1 | Lý do chọn đề tài..... | 1 |
| 1.2 | Mục tiêu nghiên cứu | 2 |
| 2 | CHƯƠNG 2. CƠ SỞ LÝ THUYẾT..... | 3 |
| 2.1 | Giới thiệu về trí tuệ nhân tạo (AI)..... | 3 |
| 2.2 | Giới thiệu về Deep learning | 4 |
| 2.3 | Dữ liệu mất cân bằng – Imbalanced data..... | 5 |
| 2.3.1 | Định nghĩa | 5 |
| 2.3.2 | Các phương pháp xử lý: | 5 |
| 2.4 | Thuật toán Convolutional Neural Network (CNN) | 6 |
| 2.4.1 | Các lớp cơ bản của CNN..... | 6 |
| 2.4.2 | Cấu trúc mạng CNN..... | 8 |
| 2.5 | Các phương pháp đánh giá mô hình – Performance metrics | 9 |
| 2.5.1 | Training curve | 9 |
| 2.5.2 | Confusion matrix..... | 11 |
| 2.6 | Các thư viện sử dụng trong đề tài | 12 |
| 2.6.1 | Thư viện TensorFlow | 12 |
| 2.6.2 | Các Component của TensorFlow | 13 |
| 2.6.3 | Thư viện Keras | 14 |
| 2.6.4 | Thư viện OpenCV (CV2)..... | 15 |
| 2.7 | Real-time..... | 15 |
| 2.7.1 | Tensorflow lite | 15 |
| 3 | CHƯƠNG 3. NỘI DUNG ĐỀ TÀI..... | 16 |

| | | |
|-------|---|----|
| 3.1 | Mô tả đề tài | 16 |
| 3.2 | Dữ liệu (dataset)..... | 17 |
| 3.2.1 | Tải dữ liệu | 17 |
| 3.2.2 | “Garbage Classification (12 classes)” | 17 |
| 3.2.3 | “Waste Classification data “ | 18 |
| 3.3 | Xử lý Imbalanced data | 19 |
| 3.3.1 | Tăng cường dữ liệu (Data Augmentation) | 20 |
| 3.4 | <i>Mạng Convolutional neural network</i> | 21 |
| 3.5 | Tiến hành train model | 23 |
| 3.6 | Các phương pháp đánh giá mô hình | 25 |
| 3.7 | Triển khai mô hình lên app adroid mobile chạy thực nghiệm. | 29 |
| | Tài liệu tham khảo | 32 |

DANH MỤC CÁC KÝ HIỆU VÀ TỪ VIẾT TẮT

| Viết tắt | Tiếng Anh | Tiếng Việt |
|----------|------------------------------|----------------------|
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| ML | Machine Learning | Học máy |
| DL | Deep Learning | Học sâu |
| CNN | Convolutional Neurol Network | Mạng Noron tích chập |
| GPU | Graphics Processing Unit | Bộ xử lý |

PHỤ LỤC HÌNH ẢNH

| | |
|---|----|
| Hình 1 Những đóng góp của AI trong các lĩnh vực | 3 |
| Hình 2 Phân biệt AI, ML, DL..... | 4 |
| Hình 3 Oversampling and Undersampling | 5 |
| Hình 4 Cấu trúc của CNN..... | 6 |
| Hình 5 Max Pooling | 7 |
| Hình 6 Cấu trúc mạng CNN | 8 |
| Hình 7 Underfitting, Fit, Overfitting | 10 |
| Hình 8. Các ô màu đậm thể hiện các giá trị cao | 11 |
| Hình 9 Mật độ sử dụng keras so với phần còn lại | 14 |
| Hình 10 Biểu đồ thể hiện số lượng ảnh trong mỗi lớp | 18 |
| Hình 11 Biểu đồ thể hiện số lượng ảnh trong mỗi lớp | 18 |
| Hình 12 Biểu đồ thể hiện số lượng ảnh trong từng lớp sau khi drop bớt ảnh | 19 |
| Hình 13 Confusion matrix của model A..... | 25 |
| Hình 14 Các chỉ số đánh giá mô hình..... | 25 |
| Hình 15 Learning cure của model A | 26 |
| Hình 16 Learning curve của model B..... | 26 |
| Hình 17 Test model phân loại 12 loại rác..... | 27 |

| | |
|--|----|
| Hình 18 Test model phân loại rác tái chế (R) và rác hữu cơ (O) | 28 |
| Hình 19 Giao diện phần mềm..... | 29 |
| Hình 20 Chạy real time nhận diện 12 loại rác | 30 |
| Hình 21 Chạy real time nhận diện rác tái chế được và không tái chế được..... | 31 |

A. MỞ ĐẦU

Hiện nay, rác thải đang trở thành vấn đề nan giải đối với các quốc gia trên toàn cầu trong đó có Việt Nam. Môi trường sống ngày càng bị ảnh hưởng và trở nên nóng hơn bao giờ hết khi tốc độ đô thị hóa ngày càng cao. Số liệu thống kê lượng rác thải ở Việt Nam và trên thế giới cho thấy thực trạng ô nhiễm đang rất báo động.. Hiện nay, công nghệ chôn lấp vẫn là phương pháp được sử dụng phổ biến nhất để xử lý rác thải. Tuy nhiên, với lượng rác thải khổng lồ như hiện nay thì các công nghệ xử lý cũng như khả năng vận chuyển vẫn chưa đáp ứng được yêu cầu. ^[1]

Theo Business Insider, các robot hỗ trợ xử lý rác thải có hiệu quả và năng suất cao đã được chế tạo để giải quyết bài toán rác thải. Một số hãng sản xuất robot lớn trên thế giới như AMP Robotics, Bulk Handling Systems of Eugene ở Oregon (Mỹ), ZenRobotics ở Helsinki (Phần Lan)... đã đẩy mạnh nghiên cứu, phát triển các robot phân loại rác thải, lần lượt cho ra mắt nhiều robot hiện đại, tích hợp công nghệ AI để giúp con người phân loại rác thải, tăng năng suất lao động.

Forbes cho biết, robot của Công ty AMP Robotics là giải pháp tối ưu cho vấn đề nhân công, hiện có mặt tại 600 cơ sở tái chế rác tại Mỹ. Nhà sáng lập, đồng thời là Tổng Giám đốc điều hành (CEO) của AMP Robotics, ông Matanya Horowitz cho biết, trước khi đại dịch Covid-19 bùng phát, sản phẩm của AMP Robotics đã được nhiều khách hàng quan tâm và tình hình kinh doanh của hãng thật sự khấm khá khi các nhà máy tái chế bị quá tải vì không tuyển đủ lao động. Theo chia sẻ của Horowitz, AMP Robotics đã bán và cho thuê 100 dây chuyền robot tích hợp AI phục vụ 40 nhà máy tái chế quy mô lớn trên khắp Bắc Mỹ, châu Âu và Nhật Bản. Giá của mỗi bộ robot này khoảng 300.000 USD, khá đắt, song nhiều trung tâm tái chế tại Mỹ vẫn đầu tư lắp đặt hệ thống này với hy vọng có thể cắt giảm lượng nhân công và nâng cao năng suất làm việc. ^[2]

Tóm tắt nội dung văn bản gồm 4 chương.

Chương 1: Tổng quan. Lý do chọn đề tài và mục tiêu nghiên cứu

Chương 2: Cơ sở lý thuyết : AI, Deep Learning, Convolutional Neural Network,...

Chương 3: Xây dựng mô hình. Trình bày về mô hình CNN

Chương 4: Triển khai mô hình chạy trên ứng dụng

B. NỘI DUNG

1 CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Theo số liệu thống kê, lượng rác thải mỗi ngày tại Việt Nam là vào khoảng 50.000 tấn. Trong đó, các đô thị có lượng rác thải là vào khoảng 35.000 tấn/ngày chiếm 70%, còn lại là ở các vùng nông thôn. Các thành phố lớn như Hà Nội, TP. Hồ Chí Minh đang hằng ngày thải ra lượng rác từ 7.000 đến 8.000 tấn. Tuy nhiên, phương pháp xử lý bằng công nghệ chôn lấp chiếm tới 80% lại gây ô nhiễm rất lớn^[1]. Việc phân loại rác thải sinh hoạt thành các loại như không tái chế được (Non-recyclable) và rác tái chế được (Recyclable) là một trong những biện pháp đơn giản và đóng một vai trò quan trọng trong hệ thống quản lý chất thải để bảo vệ môi trường.

Ngoài hai loại chính là rác không tái chế và rác tái chế, rác thải còn được chia nhỏ ra thành nhiều loại như: kim loại, giấy các loại (giấy và bìa các tông), nhựa (chai nhựa, túi nilong,...), thủy tinh các loại (thủy tinh trắng, xanh và nâu) thuộc nhóm rác tái chế; rác thải sinh học (các loại thực phẩm, trái cây), các loại pin, các sản phẩm từ vải (quần áo và giày), tã trẻ em và khẩu trang là những loại rác không tái chế. Xử lý rác thải bằng cách phân rác thải thành các loại như trên thay vì việc chôn xuống đất hay đốt sẽ có lợi cho sức khỏe con người, nó còn giúp đảm bảo các loại rác còn giá trị được tái chế và sử dụng, tiết kiệm tài nguyên và hạn chế ô nhiễm. Việc phân loại rác thông thường sẽ được thực hiện thủ công. Điều này không thật sự hiệu quả do nó tốn thời gian và dễ nhầm lẫn do thiếu kiến thức về phân loại rác.

Rất may, những nghiên cứu ứng dụng trí tuệ nhân tạo vào phân loại rác từ hình ảnh ngày càng phát triển đặc biệt có hiệu quả cao với thuật toán Convolutional Neural Network gọi tắt là CNN. Đề tài cho dự án này được xác định để góp phần giải quyết những vấn đề

nêu trên đồng thời là cơ hội để áp dụng kiến thức môn học vào giải quyết một vấn đề mang tính xã hội.

1.2 Mục tiêu nghiên cứu

Xây dựng và huấn luyện hai mô hình sử dụng hai tập dữ liệu từ Kaggle¹ bằng mạng CNN² được xây dựng với năm lớp. Tập dữ liệu thứ nhất chứa 25,077 hình ảnh được phân ra 13,996 ảnh về rác tái chế được và 11,111 ảnh rác không tái chế được dùng để xây dựng mô hình có thể phân loại được hai loại rác trên và những vật không phải là rác với độ chính xác đạt được là 89% . Tập dữ liệu thứ hai gồm 15,150 ảnh chia thành 12 loại rác khác nhau dùng để huấn luyện mô hình phân loại được 12 loại rác thuộc tập dữ liệu với độ chính xác 71% . Cả hai mô hình sẽ được triển khai lên ứng dụng điện thoại android và chạy theo thời gian thực³. Ứng dụng sẽ sử dụng camera của điện thoại để nhận dạng một vật thể và hiển thị đó có phải rác không và là loại rác thải gì, người dùng có thể xem thư viện trong ứng dụng để tra cứu các thông tin liên qua về loại rác đó để dễ dàng phân loại và xử lý.

Đề tài hướng đến việc cung cấp giải pháp để giải quyết vấn đề nhận diện rác thải thông qua các kiến thức đã học cụ thể là mạng CCN- Convolutional neural network.

¹ <https://www.kaggle.com>

² Convolutional neural network

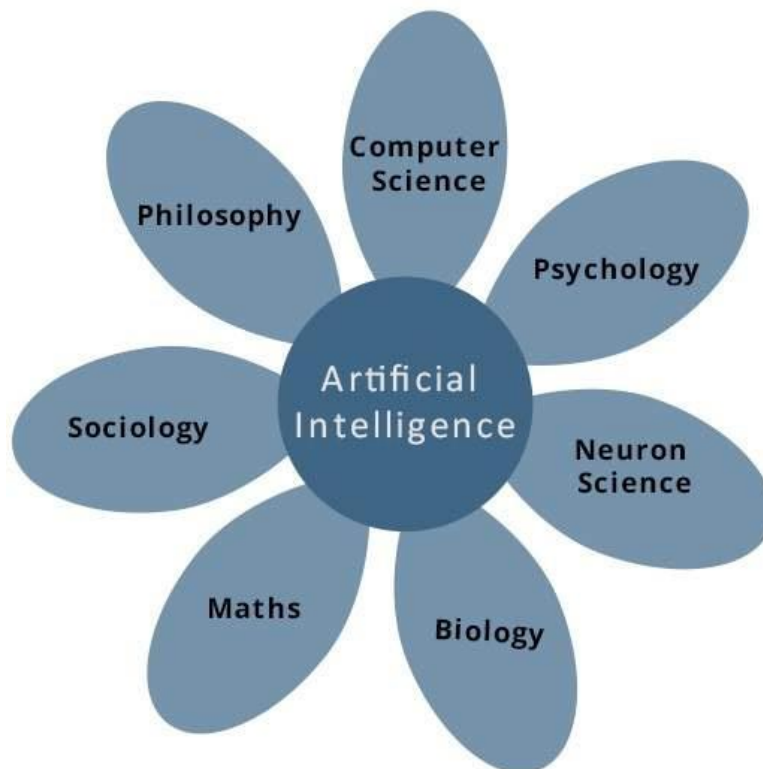
³ Real-time

2 CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về trí tuệ nhân tạo (AI)

AI - Artificial Intelligence hay còn gọi là *Trí tuệ nhân tạo* là một ngành khoa học, kỹ thuật chế tạo máy móc thông minh, đặc biệt là các chương trình máy tính thông minh.

AI được thực hiện bằng cách nghiên cứu cách suy nghĩ của con người, cách con người học hỏi, quyết định và làm việc trong khi giải quyết một vấn đề nào đó, và sử dụng những kết quả nghiên cứu này như một nền tảng để phát triển các phần mềm và hệ thống thông minh, từ đó áp dụng vào các mục đích khác nhau trong cuộc sống. Nói một cách dễ hiểu thì AI là việc sử dụng, phân tích các dữ liệu đầu vào nhằm đưa ra sự dự đoán rồi đi đến quyết định cuối cùng.^[3]

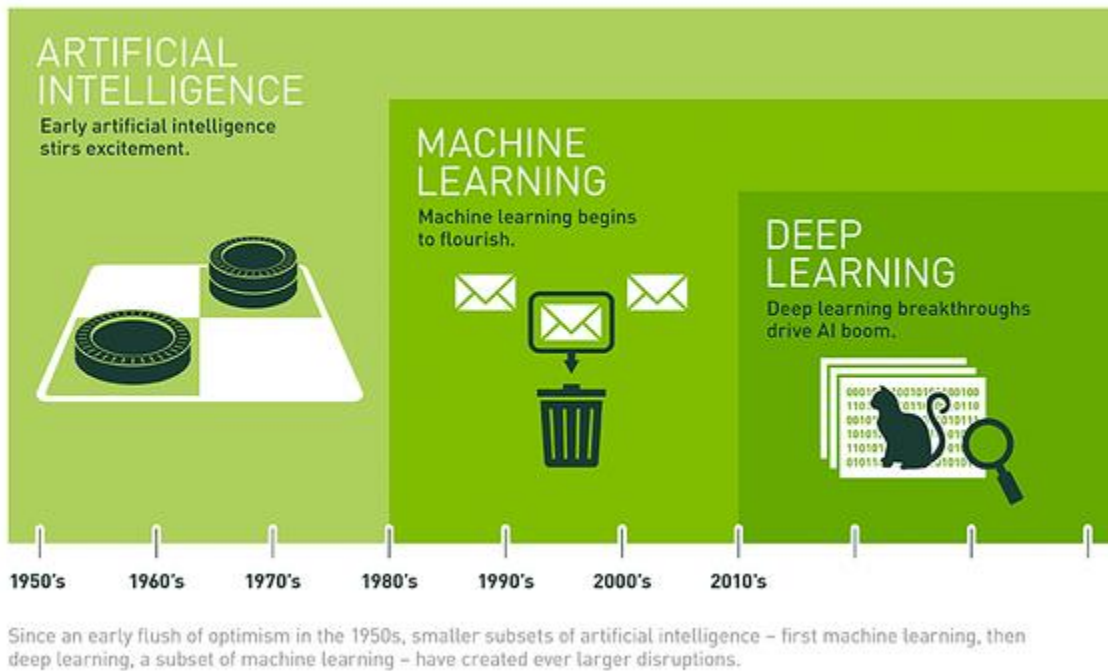


Hình 1 Những đóng góp của AI trong các lĩnh vực

2.2 Giới thiệu về Deep learning

Deep Learning được bắt nguồn từ thuật toán Neural network của AI, là một ngành nhỏ của Machine Learning. Deep learning tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên.

Trí tuệ nhân tạo có thể được hiểu đơn giản là được cấu thành từ các lớp xếp chồng lên nhau, trong đó mạng thần kinh nhân tạo nằm ở dưới đáy, Machine learning nằm ở tầng tiếp theo và Deep learning nằm ở tầng trên cùng.



Hình 2 Phân biệt AI, ML, DL

2.3 Dữ liệu mất cân bằng – Imbalanced data

2.3.1 Định nghĩa

Mất cân bằng dữ liệu (Imbalanced dataset) là tập dữ liệu có tỷ lệ categories khác nhau, thường là chênh nhau khá xa. Ví dụ dữ liệu y tế, chắc hẳn dữ liệu về một số bệnh sẽ có nhiều mẫu âm tính nhiều hơn nhiều so với mẫu dương tính, có thể lên tới tỷ lệ 98%-2%. Điều này ảnh hưởng rất lớn tới mô hình dự đoán, khi mà sự mất cân bằng như vậy sẽ khiến mô hình dự báo kém chính xác trên nhóm thiểu số, bởi đa phần kết quả dự báo thường thiên về 1 nhóm là nhóm đa số.

2.3.2 Các phương pháp xử lý:

- Random Oversampling and Undersampling : kỹ thuật Oversampling ở đây đơn giản là tăng số lượng mẫu ở Lớp thiểu số, giả sử class A có 100 mẫu, class B có 1000 mẫu ta sẽ tăng số lượng mẫu của class A bằng cách Random vị trí từ tập dữ liệu có trong mẫu và đưa vào training set. Đi cùng với oversampling thì là Undersampling. Ý tưởng tương tự như Over nhưng thay vì tăng thì sẽ là giảm data ở Lớp đa số cũng bằng cách Random như trên và delete dữ liệu đó đi.



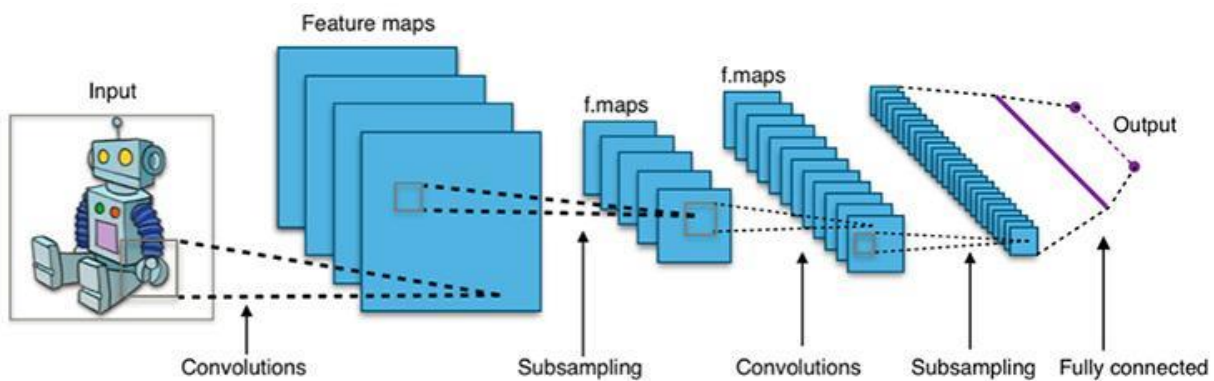
Hình 3 Oversampling and Undersampling

- SMOTEL: Ý tưởng chung của SMOTE là tạo ra dữ liệu tổng hợp giữa mỗi mẫu của lớp thiểu số và “k” hàng xóm gần nhất của nó. Có nghĩa là, đối với mỗi một trong các mẫu của lớp thiểu số, “k” các láng giềng gần nhất của nó được chọn (theo mặc định là $k = 5$). Sau đó giữa các cặp điểm được tạo bởi

mẫu và từng láng giềng của nó thì ta sẽ có được data mới. Ví dụ như hình bên dưới, với P1 là dữ liệu mẫu với $k=3 \rightarrow P2, 3, 4$ ta sẽ random giá trị giữa từng khoảng cách từ P2, 3, 4 đến P. Và cũng từ P2, 3, 4 làm tương tự như thế. [4]

2.4 Thuật toán Convolutional Neural Network (CNN)

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.



Hình 4 Cấu trúc của CNN

2.4.1 Các lớp cơ bản của CNN

a. Convolutional layer

Đây là lớp quan trọng nhất của CNN, lớp này có nhiệm vụ thực hiện mọi tính toán. Những yếu tố quan trọng của một convolutional layer là: stride, padding, filter map, feature map.

- CNN sử dụng các filter để áp dụng vào vùng của hình ảnh. Những filter map này được gọi là ma trận 3 chiều, mà bên trong nó là các con số và chúng là parameter.
- Stride có nghĩa là khi bạn dịch chuyển filter map theo pixel dựa vào giá trị trừ trái sang phải. Và sự chuyển dịch này chính là Stride.
- Padding: Là các giá trị 0 được thêm vào với lớp input.

- Feature map: Nó thể hiện kết quả của mỗi lần filter map quét qua input. Sau mỗi lần quét sẽ xảy ra quá trình tính toán.

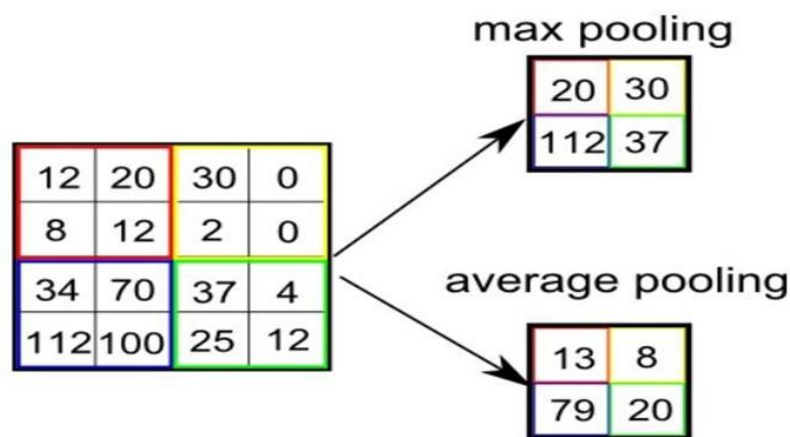
b. Relu layer

Relu layer là hàm kích hoạt trong neural network và hàm này còn được gọi là activation function. Hàm kích hoạt có tác dụng mô phỏng các neuron có tỷ lệ truyền xung qua axon. Trong activation function thì nó còn có hàm nghĩa là: Relu, Leaky, Tanh, Sigmoid, Maxout,...Hiện nay, hàm relu được dùng phổ biến và vô cùng thông dụng.

Nó được sử dụng nhiều cho các nhu cầu huấn luyện mạng neuron thì relu mang lại rất nhiều ưu điểm nổi bật như: việc tính toán sẽ trở nên nhanh hơn,... Quá trình sử dụng relu, chúng ta cần lưu ý đến vấn đề tùy chỉnh các learning rate và theo dõi dead unit. Những lớp relu layer đã được sử dụng sau khi filter map được tính ra và áp dụng hàm relu lên những giá trị của filter map.

c. Pooling layer

Khi đầu vào quá lớn, những lớp pooling layer sẽ được xếp vào giữa giữa những lớp Convolutional layer để làm giảm parameter. Hiện nay, pooling layer có 2 loại chủ yếu là: max pooling và average pooling.



Hình 5 Max Pooling

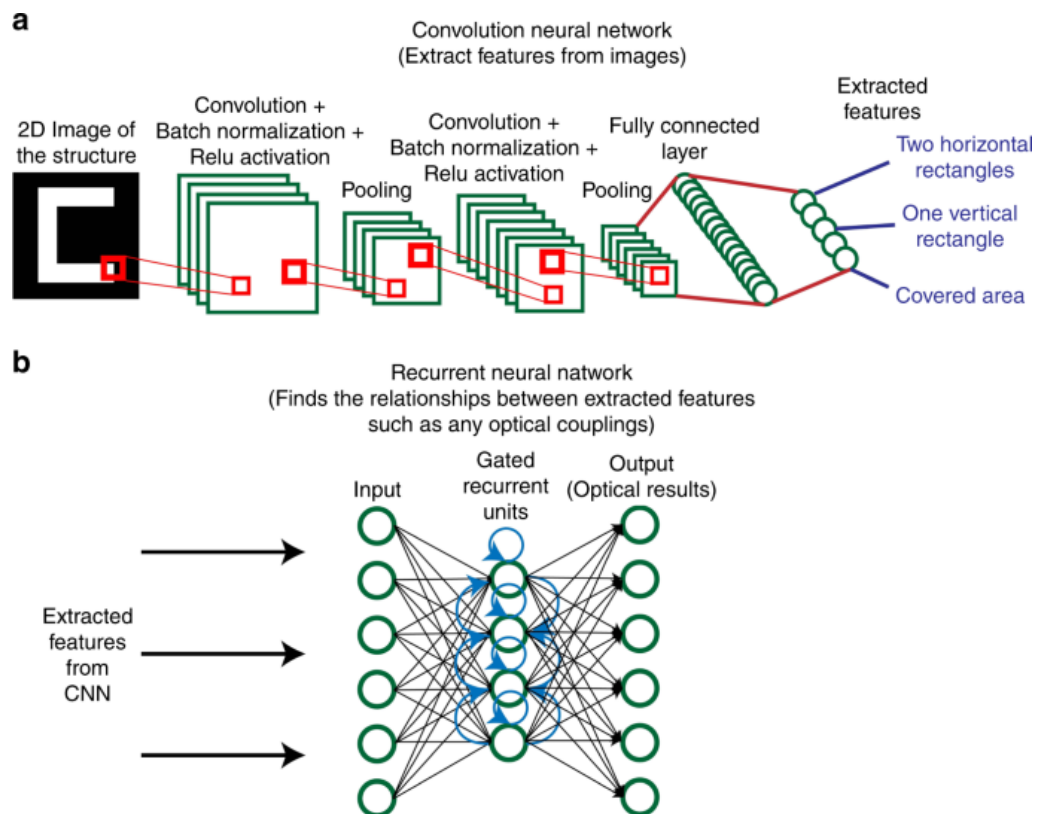
d. Fully connected layer

Lớp này có nhiệm vụ đưa ra kết quả sau khi lớp convolutional layer và pooling layer đã nhận được ảnh truyền. Lúc này, ta thu được kết quả là model đã đọc được thông tin của ảnh và để liên kết chúng cũng như cho ra nhiều output hơn thì ta sử dụng fully connected layer.

Ngoài ra, nếu như fully connected layer có được giữ liệu hình ảnh thì chúng sẽ chuyển nó thành mục chưa được phân chia chất lượng. Cái này khá giống với phiếu bầu rồi chúng sẽ đánh giá để bầu chọn ra hình ảnh có chất lượng cao nhất.

2.4.2 Cấu trúc mạng CNN

Mạng CNN là một trong những tập hợp của lớp Convolution bị chồng lên nhau cũng như sử dụng hàm nonlinear activation như ReLU và tanh để kích hoạt trọng số trong node. Lớp này sau khi thông qua hàm thì sẽ được trọng số trong các node. Những lớp này sau khi đã thông qua hàm kích hoạt thì có thể tạo ra những thông tin trừu tượng hơn cho những lớp tiếp theo.



Hình 6 Cấu trúc mạng CNN

Trong đó, cấu trúc cơ bản của CNN thường bao gồm 3 phần chính là:

- **Local receptive field** (trường cục bộ): Lớp này có nhiệm vụ tách lọc dữ liệu, thông tin ảnh và lựa chọn các vùng ảnh có giá trị sử dụng cao nhất.
- **Shared weights and bias** (trọng số chia sẻ): Lớp này giúp làm giảm tối đa lượng tham số có tác dụng chính của yếu tố này trong mạng CNN. Trong mỗi convolution sẽ có các feature map khác nhau và mỗi feature lại có khả năng giúp detect một vài feature trong ảnh.
- **Pooling layer** (lớp tổng hợp): Pooling layer là lớp cuối cùng và có tác dụng làm đơn giản các thông tin đầu ra. Có nghĩa là, sau khi đã hoàn tất tính toán và quét qua các lớp thì đến pooling layer để lược bớt các thông tin không cần thiết. Từ đó, cho ra kết quả theo như ý mà người dùng mong muốn.^[5]

2.5 Các phương pháp đánh giá mô hình – Performance metrics

2.5.1 *Training curve*

Một đường cong học tập cho thấy điểm xác nhận và đào tạo của một ước tính cho số lượng mẫu đào tạo khác nhau. Đây là một công cụ để tìm hiểu xem người dùng sẽ được hưởng lợi bao nhiêu từ việc thêm nhiều dữ liệu để huấn luyện hơn và liệu ước tính có bị lỗi phương sai hay lỗi thiên vị hay không.

Có ba hình dạng có thể quan sát được ở các đường cong học tập đó là:

- Underfitting (chưa khớp).
- Overfitting (quá khớp).
- Good fitting (vừa khớp).

a. Underfitting

Underfitting là hiện tượng khi mô hình xây dựng chưa có độ chính xác cao trong tập dữ liệu huấn luyện cũng như tổng quát hóa với tổng thể dữ liệu.

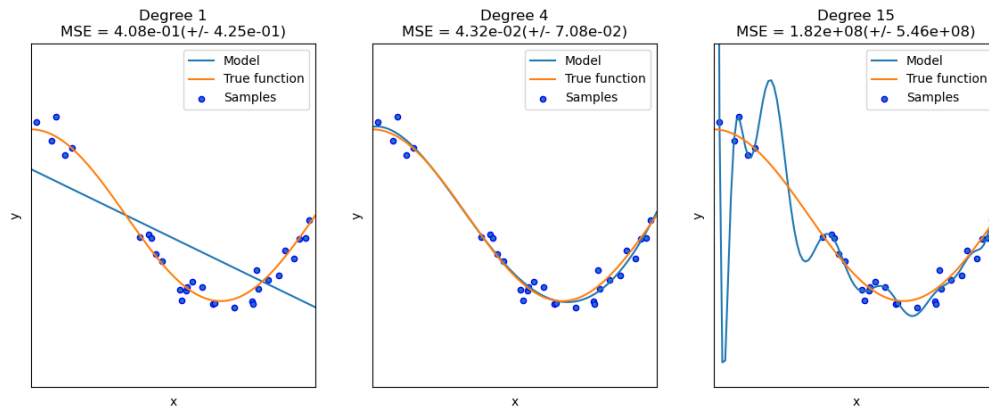
b. Overfitting

Ovefitting là một lỗi mô hình hóa xảy ra khi một hàm quá phù hợp với một tập hợp giới hạn các điểm dữ liệu. Mô hình đã học bộ dữ liệu huấn luyện quá tốt, bao gồm cả những dữ liệu nhiễu hoặc những biến động ngẫu nhiên trong bộ dữ liệu.

c. Good fitting

Good Fitting (vừa khớp) là nằm giữa Underfitting và Overfitting. Mô hình cho ra kết quả hợp lý với cả tập dữ liệu huấn luyện và các tập dữ liệu mới. Đây là mô hình lý tưởng mang tính tổng quát và khớp được với nhiều dữ liệu mẫu và cả các dữ liệu mới.

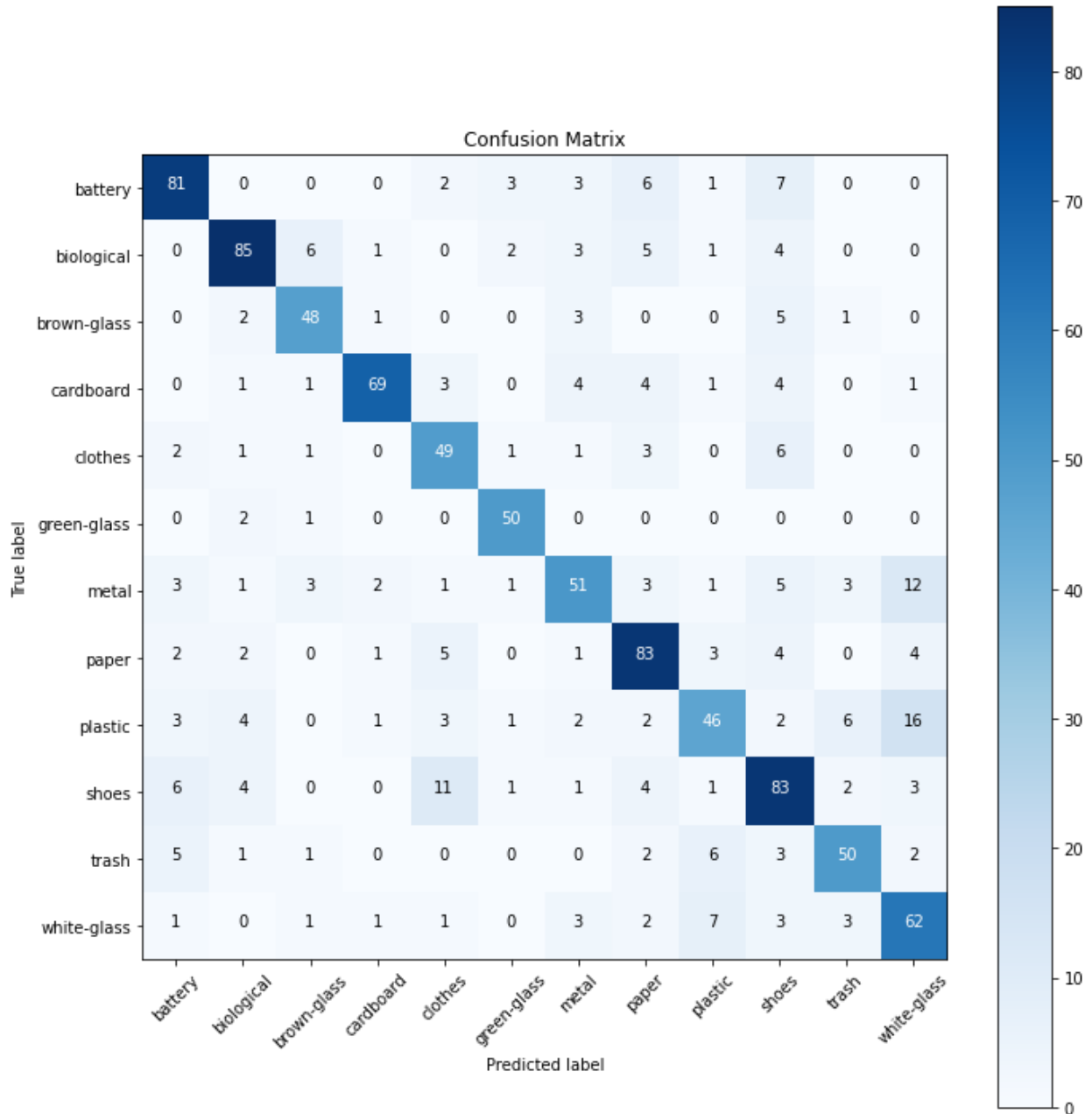
Good Fitting là mục tiêu của mỗi bài toán. Tuy nhiên, trên thực tế, vấn đề này rất khó thực hiện. Để tìm được điểm Good Fitting, ta phải theo dõi hiệu suất của thuật toán học máy theo thời gian khi thuật toán thực hiện việc học trên bộ dữ liệu huấn luyện. Ta có thể mô tả và thể hiện các thông số mô hình, độ chính xác của mô hình trên cả hai tập dữ liệu huấn luyện và đào tạo.



Hình 7 Underfitting, Fit, Overfitting

2.5.2 Confusion matrix

Confusion matrix là một ma trận vuông với kích thước mỗi chiều bằng số lượng lớp dữ liệu. Giá trị tại hàng thứ i , cột thứ j là số lượng điểm lẽ ra thuộc vào class i nhưng lại được dự đoán là thuộc vào class j .



Hình 8. Các ô màu đậm thể hiện các giá trị cao

Với các bài toán với nhiều lớp dữ liệu, cách biểu diễn bằng màu này rất hữu ích. Các ô màu đậm thể hiện các giá trị cao. Một mô hình tốt sẽ cho một confusion matrix có các phần tử trên đường chéo chính có giá trị lớn, các phần tử còn lại có giá trị nhỏ. Nói cách khác, khi biểu diễn bằng màu sắc, đường chéo có màu càng đậm so với phần còn lại sẽ càng tốt. Từ hai hình trên ta thấy rằng confusion matrix đã chuẩn hoá mang nhiều thông tin hơn. Sự khác nhau được thấy ở ô trên cùng bên trái. Lớp dữ liệu 0 được phân loại không thực sự tốt nhưng trong *unnormalized confusion matrix*, nó vẫn có màu đậm như hai ô còn lại trên đường chéo chính.

2.6 Các thư viện sử dụng trong đề tài

2.6.1 Thư viện TensorFlow

a. Giới thiệu về TensorFlow

TensorFlow cơ bản là một thư viện dạng nguồn mở được sử dụng rất phổ biến trong lĩnh vực học máy giúp gia tăng tốc độ nhanh chóng và dễ dàng hơn. Một khi trí tuệ nhân tạo được đưa vào trong giảng dạy tại các trường đại học thì sẽ trở nên nổi tiếng và được dùng phổ biến trong giáo dục.

TensorFlow được tạo ra và phát triển bởi đội ngũ chuyên viên của Google mà cụ thể là Google Brain. Nó được tạo ra với mục đích chính là sử dụng cho các nhu cầu nghiên cứu và áp dụng trong sản xuất sao cho hiệu quả nhất. TensorFlow được cấp phép hoạt động vào tháng 11 năm 2015.^[7]

b. Kiến trúc của TensorFlow hoạt động được chia thành 3 phần:

- Tiền xử lý dữ liệu (Data Preprocess).
- Dựng model.
- Train và ước tính model.

c. Ưu điểm của TensorFlow:

Abstraction: người dùng chỉ cần tập trung vào phần logic tổng thể thay vì phải đối phó với việc triển khai thuật toán rườm rà. Chế độ [eager](#)

[execution](#) cho phép bạn đánh giá và sửa đổi từng operation của biểu đồ 1 cách riêng biệt và minh bạch, thay vì phải dựng toàn bộ biểu đồ dưới dạng 1 đối tượng độc lập vốn khá mơ hồ hay phải đánh giá chung tổng thể.

d. TensorBoard cho phép bạn quan sát 1 cách trực quan những gì TensorFlow đang làm.

2.6.2 Các Component của TensorFlow

a. Tensor

Tensor là framework cốt lõi của TensorFlow và tất cả các tính toán đều liên quan đến framework này. Một Tensor là một vector hoặc là một ma trận n-chiều không gian đại diện cho tất cả các loại dữ liệu. Tất cả các giá trị trong cùng một Tensor chứa đựng loại dữ liệu giống hệt nhau với một shape đã biết (hoặc đã biết một phần). Shape của dữ liệu chính là chiều của ma trận hoặc của mảng.

Một tensor có thể được bắt nguồn từ dữ liệu input hay kết quả của một tính toán. Trong TensorFlow, tất cả các hoạt động được tiến hành bên trong một graph – biểu đồ. Biểu đồ là một tập hợp tính toán được diễn ra liên tiếp. Mỗi operation được gọi là một op node (operation node) và được kết nối với nhau.

b. Graph

TensorFlow sử dụng framework dạng biểu đồ, tập hợp và mô tả tất cả các chuỗi tính toán được thực hiện trong quá trình training. Biểu đồ cũng mang rất nhiều lợi thế:

- Nó được làm ra để chạy trên nhiều CPU hay GPU, ngay cả các hệ điều hành trên thiết bị điện thoại.
- Tính di động của biểu đồ cho phép bảo toàn các tính toán để bạn sử dụng ngay hay sau đó. Biểu đồ có thể được lưu lại để thực thi trong tương lai.
- Tất cả tính toán trong biểu đồ được thực hiện bằng cách kết nối các tensor lại với nhau, một tensor có một node và một edge. Node mang operation toán học và sản xuất các output ở đầu cuối. Các edge giải thích mối quan hệ input/output giữa các node.^[8]

2.6.3 Thư viện Keras

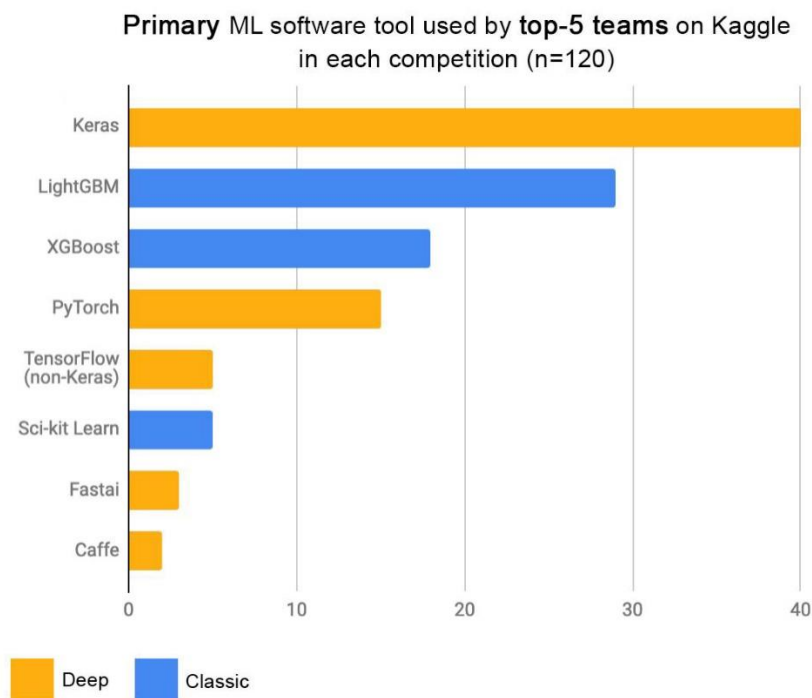
2.6.3.1 Giới thiệu về Keras

Keras là một open source cho mạng Noron, được viết bằng Python. Đây là một thư viện được phát triển bởi Francois Chollet, là một kỹ sư nghiên cứu Deep Learning.

Keras chạy trên nền tảng học máy TensorFlow, và được phát triển với trọng tâm là cho phép thử nghiệm nhanh. Có thể đi từ ý tưởng đến kết quả càng nhanh càng tốt và là chìa khóa quan trọng để thực hiện nghiên cứu tốt.

2.6.3.2 Ưu điểm của Keras

- Giảm tải nhận thức của nhà phát triển để giải phóng, tập trung vào các phần vấn đề quan trọng.
- Áp dụng nguyên tắc bộ lộ dần mức độ phức tạp, quy trình công việc đơn giản phải nhanh chóng và dễ dàng, trong khi quy trình công việc nâng cao tùy ý có thể thực hiện được thông qua một con đường rõ ràng dựa trên những gì đã học.
- Cung cấp khả năng mở rộng và hiệu suất toàn ngành.



Hình 9 Mật độ sử dụng keras so với phần còn lại

2.6.4 Thư viện OpenCV (CV2)

2.6.4.1 Giới thiệu về OpenCV

OpenCV là một thư viện mã nguồn mở hàng đầu cho lĩnh vực thị giác máy tính (Computer Vision), lĩnh vực học máy (Machine learning), xử lý ảnh và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

OpenCV có thể được sử dụng ở hầu hết các ngôn ngữ như C/C++, Python, Java... và các hệ điều hành khác nhau như Windows, Linux, OS, Android, iOS... Ngoài ra có thể sử dụng card đồ họa GPU để xử lý nhằm tăng tốc độ xử lý.

OpenCV là dự án bắt đầu bởi hãng Intel vào năm 1999 bởi Gary Bradsky và ra mắt lần đầu tiên vào năm 2000.

2.6.4.2 Các ứng dụng của OpenCV

OpenCV được sử dụng trong các ứng dụng:

- Hình ảnh Street Views.
- Kiểm tra và giám sát tự động Robot và xe tự động.
- Phân tích hình ảnh y tế.
- Tìm kiếm và phục hồi hình ảnh/ video phim – cấu trúc 3D từ chuyển động.
- Nghệ thuật sắp đặt tương tác.

2.7 Real-time

2.7.1 Tensorflow lite

Để triển khai mô hình sử dụng thuật toán CNN cần chuyển đổi file h5 sang file tflite để có thể sử dụng kết quả huấn luyện mô hình trên TensorFlow Lite. Sau khi chuyển đổi file xong thì nhúng file đó vào trong Android Studio. Ở đây vì cả hai đều thuộc sở hữu của Google nên trên Android Studio có hỗ trợ trực tiếp việc nhúng file tflite vào ứng dụng.

3 CHƯƠNG 3. NỘI DUNG ĐỀ TÀI

3.1 Mô tả đề tài

Project cuối kỳ môn học trí tuệ nhân tạo với đề tài: “Phân loại rác thải theo thời gian thực sử dụng thuật toán Convolutional neural network”. Bài báo cáo này sẽ trình bày quá trình xây dựng và huấn luyện hai mô hình sử dụng hai tập dữ liệu từ Kaggle⁴ bằng mạng CNN⁵ được xây dựng với năm lớp. Tập dữ liệu thứ nhất chứa 25,077 hình ảnh được phân ra 13,996 ảnh về rác tái chế được và 11,111 ảnh rác không tái chế được dùng để xây dựng mô hình có thể phân loại được hai loại rác trên và những vật không phải là rác với độ chính xác đạt được là 89% . Tập dữ liệu thứ hai gồm 15,150 ảnh chia thành 12 loại rác khác nhau dùng để huấn luyện mô hình phân loại được 12 loại rác thuộc tập dữ liệu với độ chính xác 71% . Cả hai mô hình sẽ được triển khai lên ứng dụng điện thoại android và chạy theo thời gian thực⁶. Ứng dụng sẽ sử dụng camera của điện thoại để nhận dạng một vật thể và hiển thị đó có phải rác không và là loại rác thải gì, người dùng có thể xem thư viện trong ứng dụng để tra cứu các thông tin liên qua về loại rác đó để dễ dàng phân loại và xử lý.

⁴ <https://www.kaggle.com>

⁵ Convolutional neural network

⁶ Real-time

3.2 Dữ liệu (dataset)

3.2.1 Tải dữ liệu

Tải dữ liệu từ KAGGLE bằng cách lấy API token để tạo đường dẫn sau đó tải data về thông qua API command rồi giải nén vào thư mục đã tạo sẵn

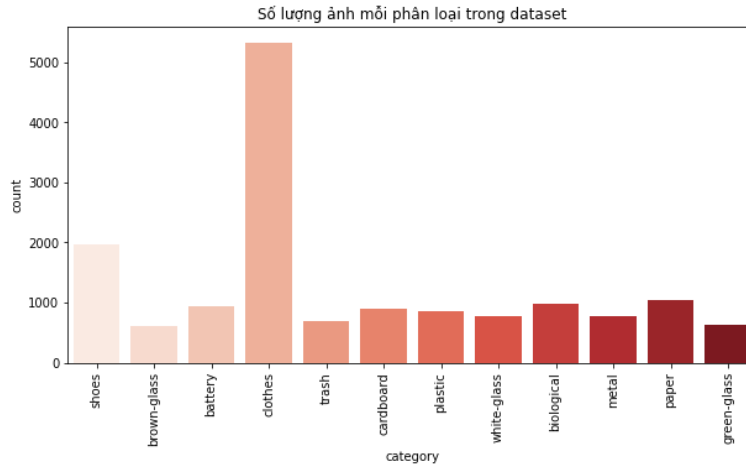
```
import os
import json
kaggleAPIToken = {"username":"lenguyentrung", "key":"53097035ab313b0e4ac545a7637f04f9"}
with open('/content/kaggle/kaggle.json', 'w') as file:
    json.dump(kaggleAPIToken, file)
!chmod 600 /content/kaggle/kaggle.json
!sudo mkdir ~/.kaggle
!cp /content/kaggle/kaggle.json ~/.kaggle/kaggle.json
!mkdir dataset
!kaggle datasets download -d lenguyentrung/garbageonr
```

3.2.2 “Garbage Classification (12 classes)”⁷

Tập dữ liệu này bao gồm hơn 15,000 ảnh phân ra 12 lớp khác nhau: paper, carboard, biological, metal, plastic, green-class, brown-glass, clothes, shoes, battery and trash. Theo thông tin từ tác giả của dataset thì tập dữ liệu này có 22% dữ liệu trong lớp “shoes” được lấy từ “Clothing dataset”. Quan sát đồ thị dưới đây ta dễ dàng thấy dữ liệu đang phân bố không đều hay đang gặp tình trạng imbalanced⁸.

⁷ <https://www.kaggle.com/datasets/mostafaabla/garbage-classification>

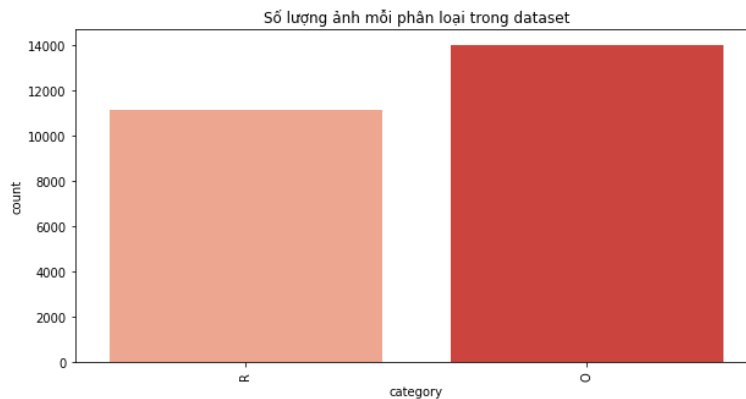
⁸ Dữ liệu có độ lệch lớn về số lượng mẫu trong từng lớp.



Hình 10 Biểu đồ thể hiện số lượng ảnh trong mỗi lớp

3.2.3 “Waste Classification data9 “

Đây là tập dữ liệu chứa hình ảnh về các loại rác thải rắn (solid waste) được đăng tải bởi Sashaank Sekar. Dataset bao gồm 25,077 hình ảnh được phân ra 13,996 ảnh về rác tái chế được (Recyclable) và 11,111 ảnh rác không tái chế được (Organic) hay rác thải hữu cơ. Mỗi class có hơn 10,00 ảnh và khá cân bằng.



Hình 11 Biểu đồ thể hiện số lượng ảnh trong mỗi lớp

***Hai mô hình sẽ được huấn luyện và đánh giá là mô hình phân loại rác thải tái chế và không tái chế (model B) và mô hình phân loại 12 loại rác (model A).**

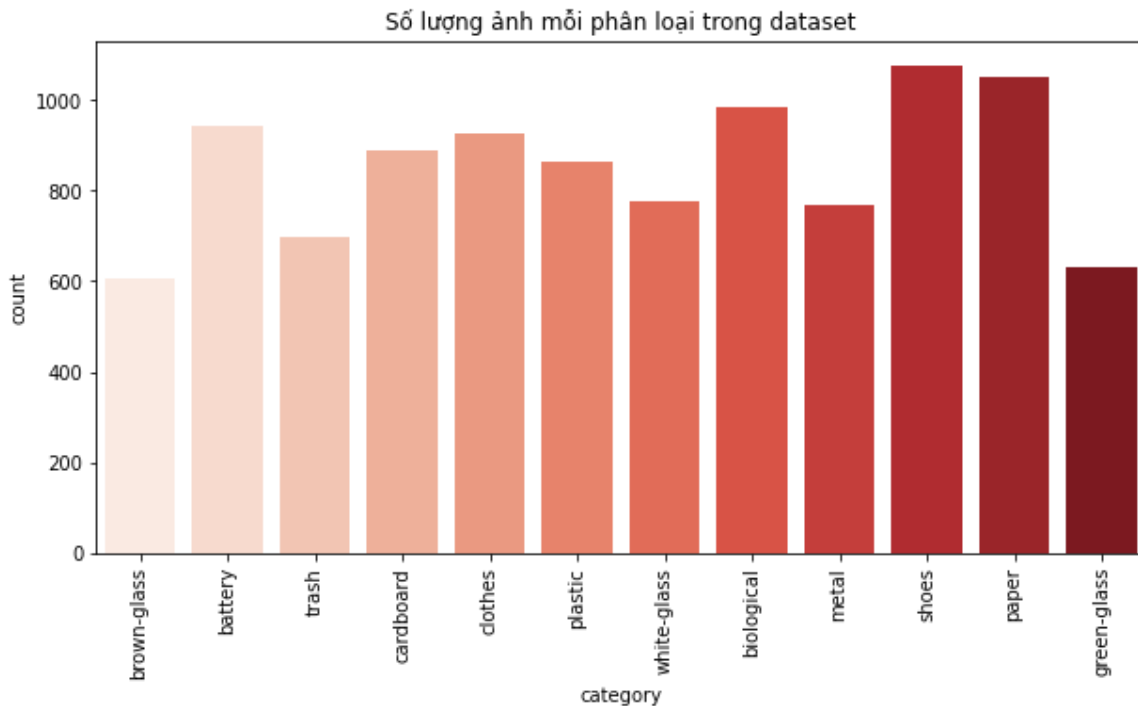
⁹ <https://www.kaggle.com/datasets/techsash/waste-classification-data>

3.3 Xử lý Imbalanced data

Như đã đề cập ở trên đối với tập dữ liệu 12 loại rác bị mất cân bằng do hai số lượng ảnh vượt trội ở lớp “clothes” và ‘shoes’ do đó cần phải xử lý trước khi sử dụng. Có nhiều phương pháp để xử lý mất cân bằng dữ liệu (handling imbalanced) như over/under sampling hay smote,...Do số lượng lớp gây imbalanced là 2/12 lớp và cả hai đều có số lượng sample cao hơn so với phần còn lại nên áp dụng under sampling sẽ hiệu quả và đơn giản nhất. Cụ thể ở đây ta sẽ drop bớt sample của hai class này.

```
clothesDrop=df[df['category']=='clothes'].sample(n=4400)
shoesDrop=df[df['category']=='shoes'].sample(n=900)
df.drop(labels=clothesDrop.index.values, inplace=True)
df.drop(labels=shoesDrop.index.values, inplace=True)
```

Sau khi drop 4400 ảnh từ class ‘clothes’ và 900 ảnh từ class ‘shoes’ :



Hình 12 Biểu đồ thể hiện số lượng ảnh trong từng lớp sau khi drop bớt ảnh

3.3.1 Tăng cường dữ liệu (Data Augmentation)

Tăng cường dữ liệu giúp khắc phục vấn đề “không đủ dữ liệu”, khắc phục các điểm overfitting và giúp cho mô hình hoạt động tốt hơn với đa dạng mẫu dữ liệu. Data Augmentation (Tăng cường dữ liệu) là một kỹ thuật được sử dụng để mở rộng kích thước của tập huấn luyện bằng cách tạo thêm dữ liệu đã được sửa đổi từ dữ liệu ban đầu. Trong project này, các phương thức được sử dụng bao gồm:

```
horizontal_flip,    vertical_flip,    rescale=1./255,    validation_split=0.1,  
rotation_range=40,    brightness_range= [1.1, 1.5],    zoom_range=0.2,  
width_shift_range = 0.2,    height_shift_range = 0.2,    shear_range= 0.2,  
fill_mode='nearest'
```

Sử dụng `flow_from_directory()`:

```
train_generator = train_generator.flow_from_dataframe(  
    dataframe=train,  
    directory=data_path,  
    x_col='filename',  
    y_col='category',  
    target_size=im_shape,  
    class_mode='categorical',  
    batch_size=batch_size,  
    seed=seed)
```

Trong đó:

- **directory**: phải đặt đường dẫn có các classes của folder.
- **target_size**: là size của các ảnh input đầu vào, mỗi ảnh sẽ được resized theo kích thước này.
- **color_mode**: Nếu hình ảnh là màu đen và màu trắng hoặc là grayscale thì set "grayscale" hoặc nếu nó gồm 3 channels thì set "rgb"
- **batch_size** : Số lượng ảnh được yielded từ generator cho mỗi lô batch.
- **class_mode** : set "binary" nếu bạn có 2 classes để dự đoán, nếu không thì bạn set "categorical". trong trường hợp nếu bạn đang lập trình một hệ thống tự động

Autoencoder, thì cả input và output đều là ảnh, trong trường hợp này thì bạn set là input

- **shuffle**: set True nếu bạn muốn đổi thứ tự hình ảnh, ngược lại set False.
- **seed** : Random seed để áp dụng tăng hình ảnh ngẫu nhiên và xáo trộn thứ tự của hình ảnh

→ Đối với mô hình phân loại 12 loại ra ta sẽ làm giàu cả ba tập train, test và validation còn đối với mô hình phân loại rác tái chế ta chỉ làm giàu tập train.

3.4 Mạng Convolutional neural network

Thuật toán Convolutional neural network (CNN) là một trong những thuật toán huấn luyện mô hình deep learning phổ biến nhất. CNN được dùng trong trong nhiều bài toán như nhận dạng ảnh, phân tích video, ảnh MRI, hoặc cho bài các bài của lĩnh vực xử lý ngôn ngữ tự nhiên, và hầu hết đều giải quyết tốt các bài toán này^[2]. Trong đề tài này, mô hình CNN được xây dựng với năm lớp được thiết lập với ảnh input có kích thước 256x256.

```

model = Sequential()

model.add(Conv2D(16, kernel_size=(3,3), activation='relu', input_shape=(im
_shape[0], im_shape[1], 3)))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(32, kernel_size=(3,3), activation='relu', input_shape=(im
_shape[0], im_shape[1], 3)))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(128, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(12, activation='softmax')) # Sử dụng hàm softmax đối với
                                           # model nào có 12 lớp model
                                           # có 2 lớp dùng hàm sigmoid

model.summary()
# compile model
model.compile(loss='categorical_crossentropy',
              optimizer=Adam(),
              metrics=['accuracy'])

```

| Layer (type) | Output Shape | Param # |
|---------------------------------|----------------------|---------|
| conv2d_10 (Conv2D) | (None, 254, 254, 16) | 448 |
| max_pooling2d_10 (MaxPooling2D) | (None, 127, 127, 16) | 0 |
| conv2d_11 (Conv2D) | (None, 125, 125, 32) | 4640 |
| max_pooling2d_11 (MaxPooling2D) | (None, 62, 62, 32) | 0 |
| conv2d_12 (Conv2D) | (None, 60, 60, 64) | 18496 |
| max_pooling2d_12 (MaxPooling2D) | (None, 30, 30, 64) | 0 |
| conv2d_13 (Conv2D) | (None, 28, 28, 128) | 73856 |
| max_pooling2d_13 (MaxPooling2D) | (None, 14, 14, 128) | 0 |
| conv2d_14 (Conv2D) | (None, 12, 12, 256) | 295168 |
| max_pooling2d_14 (MaxPooling2D) | (None, 6, 6, 256) | 0 |
| flatten_2 (Flatten) | (None, 9216) | 0 |
| dense_6 (Dense) | (None, 512) | 4719104 |
| dropout_4 (Dropout) | (None, 512) | 0 |
| dense_7 (Dense) | (None, 512) | 262656 |
| dropout_5 (Dropout) | (None, 512) | 0 |
| dense_8 (Dense) | (None, 12) | 6156 |
| Total params: 5,380,524 | | |
| Trainable params: 5,380,524 | | |
| Non-trainable params: 0 | | |

3.5 Tiến hành train model

```
## Đặt giá trị epoch để train
epochs = 100
# Hàm callback để lưu lại model nào tốt nhất sử dụng EarlyStopping để dừng
train nếu sau 10 epoch mà val_loss ko giảm
callbacks_list = [ModelCheckpoint(filepath='model_19_6.h5',monitor='val_loss',
save_best_only=True, verbose=1),EarlyStopping(monitor='val_loss', patience=10,verbose=1)]
```

```
#Train model
history = model.fit(
    train_generator,
    steps_per_epoch=nb_train_samples // batch_size,
    epochs=epochs,
    callbacks = callbacks_list,
    validation_data=val_generator,
    verbose = 1,
    validation_steps=nb_validation_samples // batch_size)
```

Epoch=100 đối với model A

Epoch =50 đối với model B

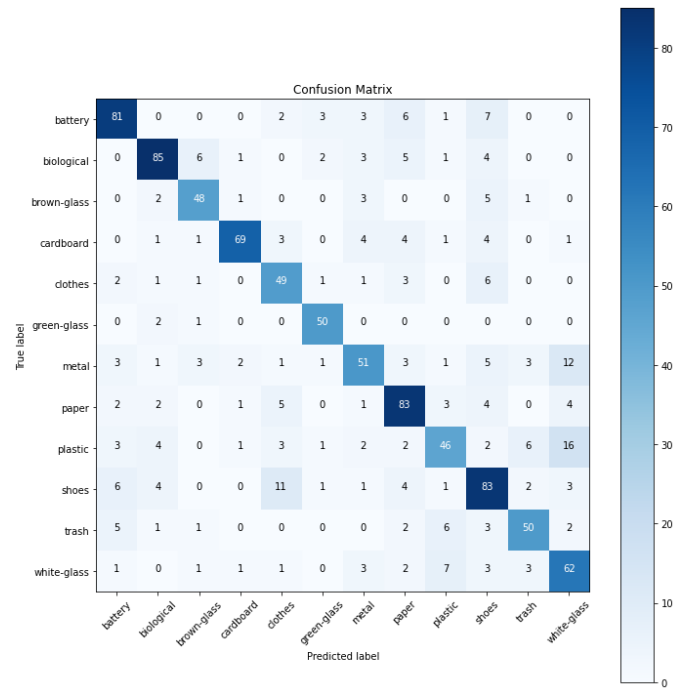
Kết quả thu được sau khi train model A:

```
16/16 [=====] - 16s 1s/step - loss: 0.9041 -
accuracy: 0.7150
Val loss: 0.9040566682815552
Val accuracy: 0.7149853110313416
```

```
1022/1022 [=====] - 17s 17ms/step - loss: 0.7799
- accuracy: 0.7524
Test loss: 0.7799473404884338
Test accuracy: 0.752446174621582
```


3.6 Các phương pháp đánh giá mô hình

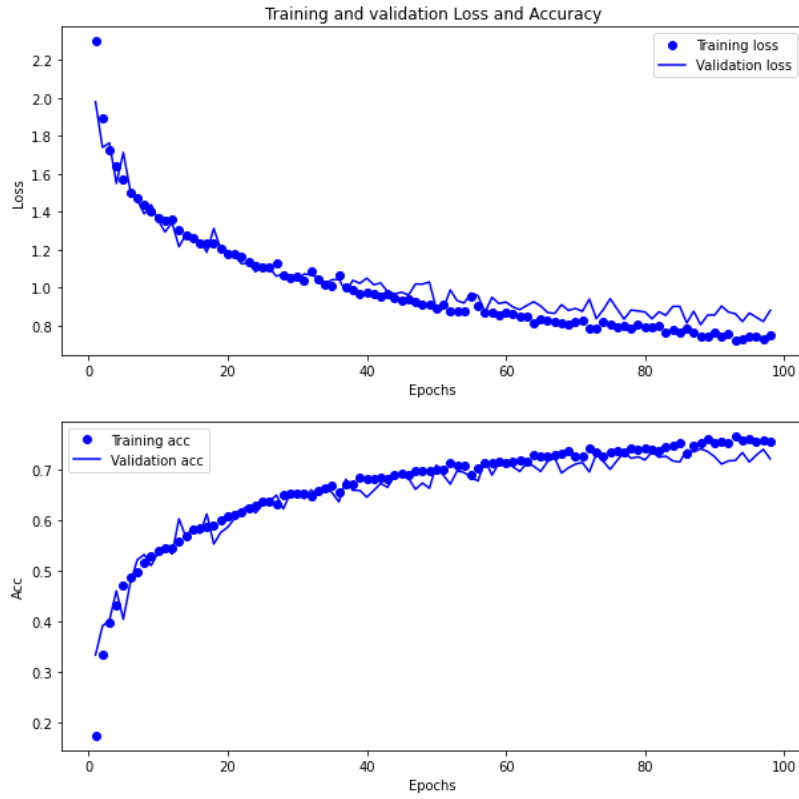
Ma trận nhầm lẫn (confusion matrix), training curve, precision, recall, f1 score, support vector machine được dùng để đánh giá mô hình. Tuy nhiên thì độ chính xác “accuracy” là công cụ đánh giá chuẩn xác và nhanh nhất.



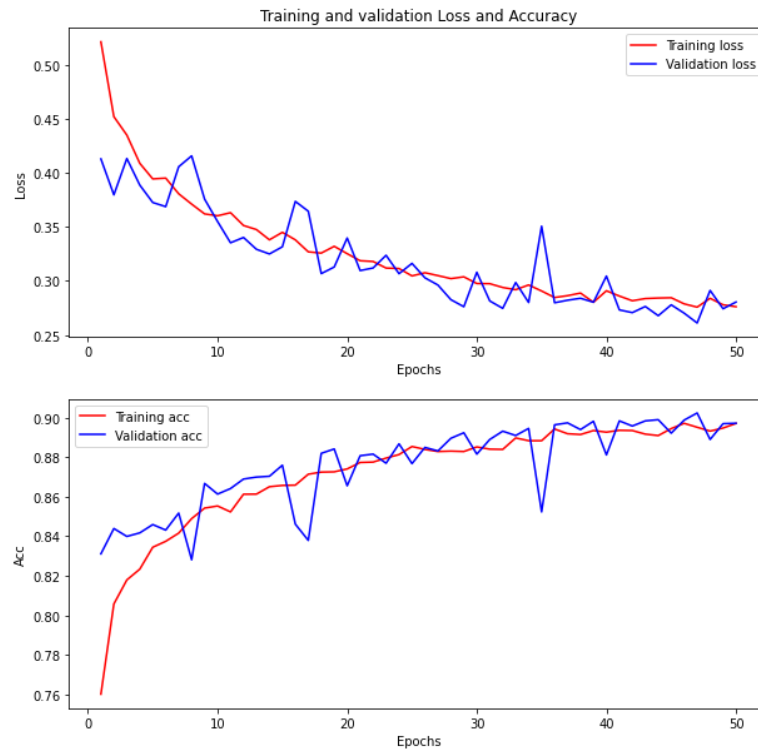
Hình 13 Confusion matrix của model A

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| battery | 0.79 | 0.79 | 0.79 | 103 |
| biological | 0.83 | 0.79 | 0.81 | 107 |
| brown-glass | 0.77 | 0.80 | 0.79 | 60 |
| cardboard | 0.91 | 0.78 | 0.84 | 88 |
| clothes | 0.65 | 0.77 | 0.71 | 64 |
| green-glass | 0.85 | 0.94 | 0.89 | 53 |
| metal | 0.71 | 0.59 | 0.65 | 86 |
| paper | 0.73 | 0.79 | 0.76 | 105 |
| plastic | 0.69 | 0.53 | 0.60 | 86 |
| shoes | 0.66 | 0.72 | 0.69 | 116 |
| trash | 0.77 | 0.71 | 0.74 | 70 |
| white-glass | 0.62 | 0.74 | 0.67 | 84 |
| accuracy | | | 0.74 | 1022 |
| macro avg | 0.75 | 0.75 | 0.74 | 1022 |
| weighted avg | 0.74 | 0.74 | 0.74 | 1022 |

Hình 14 Các chỉ số đánh giá mô hình

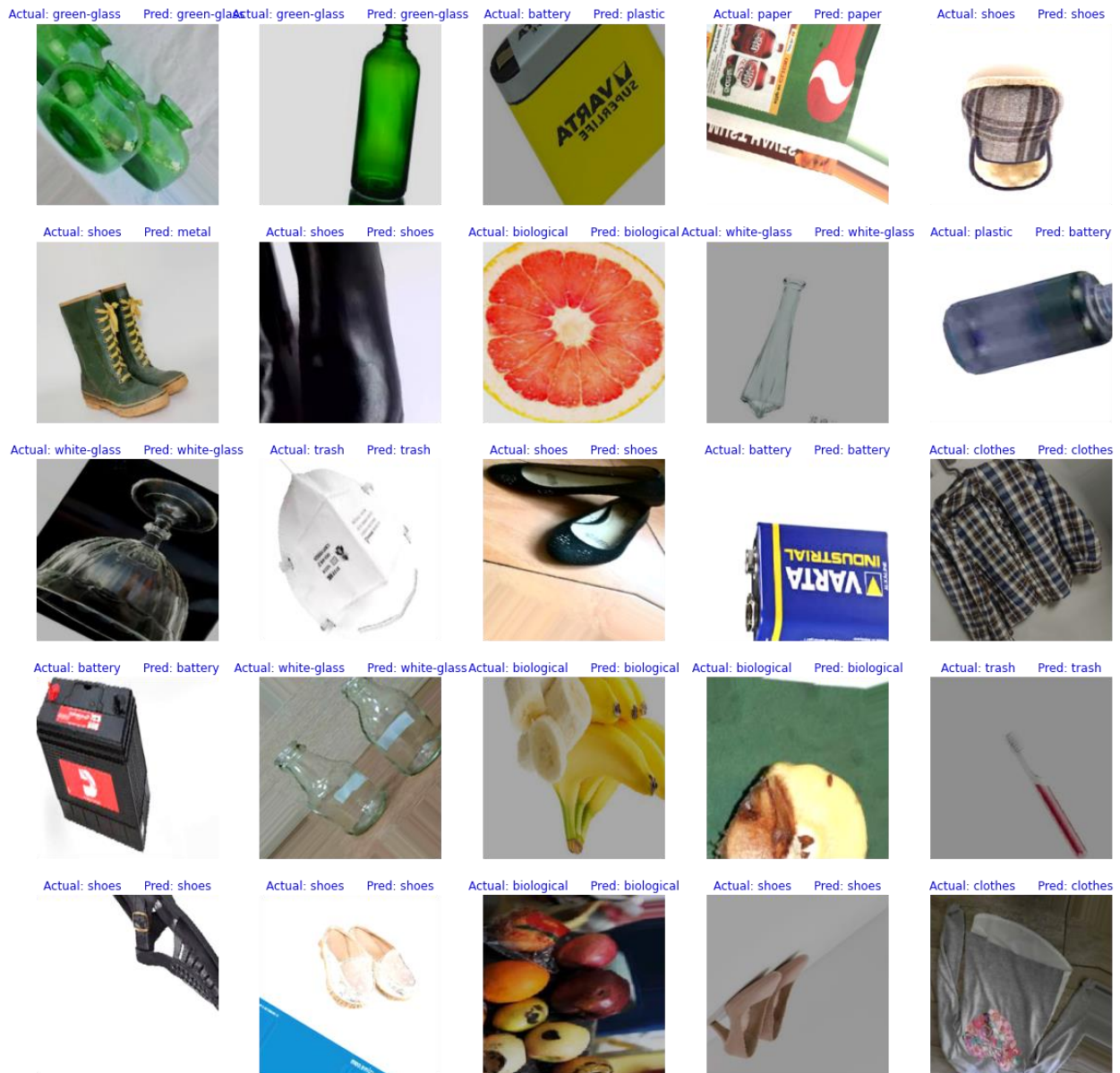


Hình 15 Learning cure của model A



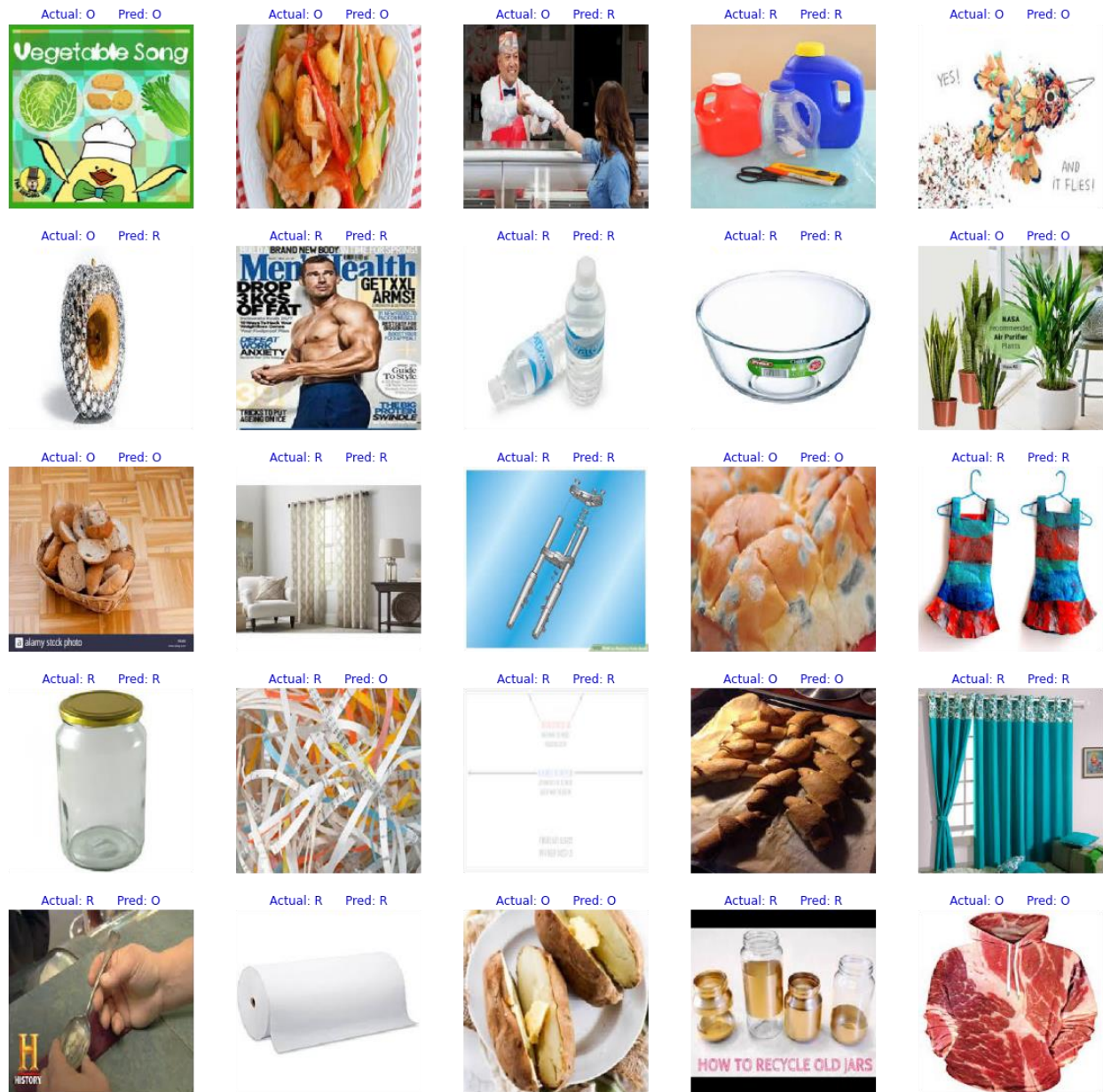
Hình 16 Learning curve của model B

Test model phân loại 12 loại rác với ảnh từ tập test:



Hình 17 Test model phân loại 12 loại rác

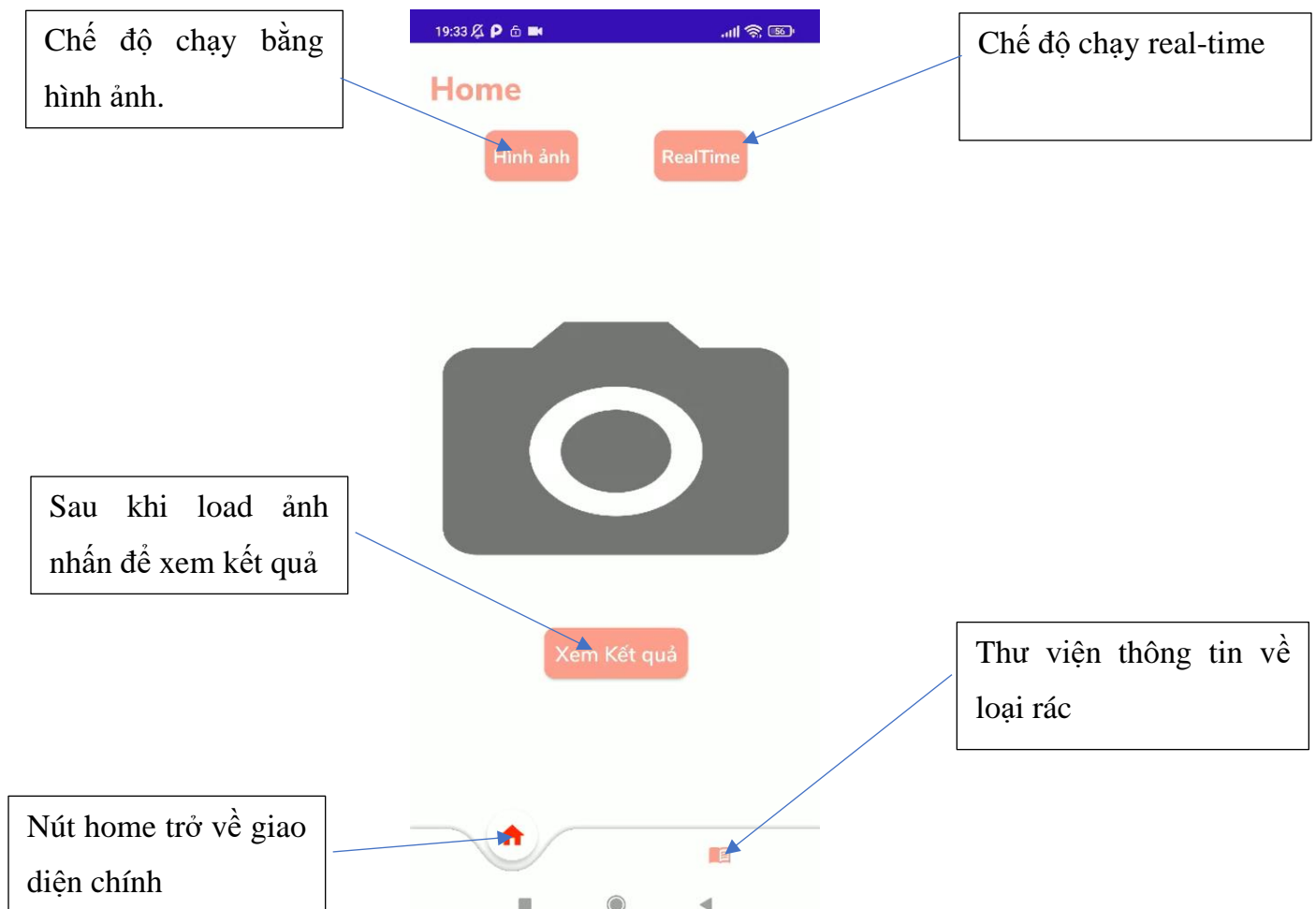
Test model phân loại 12 loại rác với ảnh từ tập test:



Hình 18 Test model phân loại rác tái chế (R) và rác hữu cơ (O)

3.7 Triển khai mô hình lên app android mobile chạy thực nghiệm.

Thiết kế ứng dụng android bằng Android studio, chuyển đổi model sang file tfLite và triển khai lên phần mềm. Phần mềm sử dụng camera sau của điện thoại để nhận dạng rác thải, có thể chuyển chế độ từ nhận diện rác thải tái chế và không tái chế được sang chế độ phân loại rác thải. Có hai chế độ lấy dữ liệu đầu vào là sử dụng camera chạy real time, lấy ảnh bằng cách chụp ảnh hoặc từ thư viện trên điện thoại để phân loại rác.



Hình 19 Giao diện phần mềm



Hình 20 Chạy real time nhận diện 12 loại rác



Hình 21 Chạy real time nhận diện rác tái chế được và không tái chế được

Tài liệu tham khảo

1. “Số liệu thống kê lượng rác thải ở Việt Nam và thực trạng đáng báo động “, Cfmobi, [https://cfmobi.vn/so-lieu-thong-ke-luong-rac-thai-o-viet nam/](https://cfmobi.vn/so-lieu-thong-ke-luong-rac-thai-o-viet-nam/)
2. “ Công nghệ robot phân loại rác thải”, Sứ giả môi trường, <https://sugiamoitruong.com.vn/cong-nghe-robot-phan-loai-rac-thai>
3. “ Tổng quan về AI- Artifical intelligence”, Charlie, <https://insights.magestore.com/posts/tong-quan-ai-artificial-intelligence>
4. “Imbalanced Multiclass Datasets”, Tô Đức Thắng, <https://viblo.asia/p/imbalanced-multiclass-datasets-Do754dmQ5M6>
5. “ Convolutional Neural Network là gì vậy? phương pháp chọn tham số cho Convolutional Neural Network chuẩn chỉnh”, Hỏi đáp, <https://hocdauthau.com/convolutional-neural-network-la-gi-cach-chon-tham-so-cho-convolutional-neural-network-chuan-chinh/>