

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



BÁO CÁO ĐỒ ÁN MÔN MÁY HỌC
Đề tài:
ỨNG DỤNG NHẬN DIỆN LOÀI HOA

GVHD: PGS.TS Lê Đình Duy
ThS. Võ Ngọc Tân

Nhóm sinh viên thực hiện:

- | | |
|---------------------|----------------|
| 1. Nguyễn Quốc Quân | MSSV: 19520871 |
| 2. Nguyễn Như Long | MSSV: 19521790 |
| 3. Trần Đình Đồng | MSSV: 17520352 |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

....., ngày.....tháng.....năm 2021

Người nhận xét

(Ký tên và ghi rõ họ tên)

MỤC LỤC

Chương 1: GIẢI TRÌNH CHỈNH SỬA SAU VẤN ĐÁP.....	5
1.1Đánh giá đơn giản, không phân biệt được hoa nào sai nhiều nhất.....	5
1.2Hoa vẫn còn khá ít	5
1.3Dẫn nguồn tên tác giả CNN, ResNet-50, Keras và source code mà nhóm tham khảo.....	5
Chương 2: BÀI TOÁN	6
2.1Giới thiệu	6
2.2Mục tiêu bài toán	Error! Bookmark not defined.
2.3Input và Output bài toán	Error! Bookmark not defined.
Chương 3: DATASET	Error! Bookmark not defined.
3.1Dữ liệu bài toán.....	6
Chương 4: CÔNG CỤ SỬ DỤNG.....	10
4.1CNN Algorithm	10
4.2Model ResNet-50	11
4.3Thư viện Keras.....	11
Chương 5: TRAIN MODEL VÀ ĐÁNH GIÁ	13
5.1Train model lần đầu	13
5.2Xử lý Overfitting.....	14
5.2.1 Data Augmentation	14
5.2.2 Drop Layer.....	14
5.3Train model sau khi xử lý Overfitting	15
5.4Train model ResNet-50.....	16
5.5Đánh giá	17
5.5.1 Accuracy	17
5.5.2 Confusion Matrix	17
Chương 6: KẾT LUẬN	20
6.1Thuận lợi	20
6.2Khó khăn	20
6.3Định hướng phát triển ứng dụng.....	21
TÀI LIỆU THAM KHẢO	21

Chương 1: GIẢI TRÌNH CHÍNH SỬA SAU VẤN ĐÁP

1.1 Đánh giá đơn giản, không phân biệt được hoa nào sai nhiều nhất.

Nhóm đã tham khảo thêm Confusion Matrix để có thể đánh giá được tốt hơn, phân biệt được hoa nào sai nhiều nhất giúp dễ dàng đánh giá model hơn. [*Truy cập*](#)

1.2 Hoa vẫn còn khá ít

Hiện tại nhóm em đã bổ sung thêm 26 loại hoa khác nâng tổng số loại hoa lên 50 loại, phân biệt được thêm nhiều loại hoa mới như: hoa Cúc Gai Tím, hoa Mao Lương, hoa Kim Cương,... [*Truy cập*](#)

1.3 Dẫn nguồn tên tác giả CNN, ResNet-50, Keras và source code mà nhóm tham khảo.

Nhóm em đã tìm hiểu và bổ sung tên tác giả, dẫn nguồn tài liệu để tôn trọng công sức của họ.

Truy cập: [*CNN*](#), [*ResNet-50*](#), [*Keras*](#), [*Tham khảo*](#)

Chương 2: BÀI TOÁN

2.1 Giới thiệu

Trong cuộc sống hằng ngày ta gặp rất nhiều loại hoa khác nhau ở trên đường, và nhiều lúc ta gặp một bông hoa đẹp nào đó thường muốn biết tên của chúng để có thể tìm kiếm thêm thông tin. Tuy nhiên không ai có thể biết hết được tên mọi loại hoa trong tự nhiên, và khi ta tự hỏi bông hoa đó tên là gì thì sẽ gặp khó khăn bởi không phải lúc nào cũng có người ở đó để ta hỏi. Việc có thể nhận diện, biết được tên của loài hoa chỉ qua một bức ảnh chụp điện thoại sẽ giúp cho người dùng dễ dàng tìm hiểu hơn rất nhiều.

Nhận thấy nhu cầu quan trọng của việc tìm hiểu, nhận diện các loại hoa, nhóm em đã chọn đề tài “Nhận diện loài hoa” với mục đích nghiên cứu, hỗ trợ người dùng có thể phân biệt được các loài hoa một cách thuận tiện chỉ cần qua một bức ảnh chụp của điện thoại.

2.2 Xác định yêu cầu đề tài

Mục đích bài toán: Mục tiêu của bài toán là giúp người dùng nhận biết được loại hoa trong chỉ qua một bức ảnh chụp từ điện thoại.

Phân loại: Đây thuộc loại bài toán Classification.

Input: Một hình ảnh bông hoa chụp từ điện thoại.

Output: Tên của loại hoa.

Hình ảnh Input nên được chụp gần, rõ nét chủ thể để model có thể dự đoán được tốt nhất.

2.3 Dữ liệu bài toán

Dữ liệu bao gồm hơn 6170 hình ảnh chia thành 50 loại hoa.

Tên hoa	Số lượng	Tên hoa	Số lượng
Anh Thảo	40	Anh Túc	101
Bìm Bịp	204	Bỉ Ngạn	91
Bồ Câu	144	Bồ Công Anh	38
Bướm	85	Cẩm Chướng	74
Cẩm Tú Cầu	125	Chi Vân Môn	201

Cúc	119	Cúc Cánh Mối	60
Cúc Gai Tím	85	Cúc Mắt Bò	236
Dâm Bụt	319	Đào	102
Dã Quỳ	143	Diên Vĩ	49
Dừa Cạn	140	Giấy	183
Hồng	222	Hồng Bụt	51
Hồng Môn	202	Hồng Sa Mạc	62
Huệ Chuối	80	Hướng Dương	192
Kế Sờ	45	Kim Cương	46
Kim Liên	441	Lạc Tiên	251
Loa Kèn Vần	38	Ly	178
Ly Lửa	40	Mắc Cỡ	178
Mai	138	Mao Lương	71
Mộc Lan	63	Ngũ Sắc	148
Phong Lữ	114	Quạt Ba Tiêu	34
Riềng Tía	132	Sen Cạn	183
Sứ	171	Súng	200
Thiên Điều	154	Thược Dược	109
Thủy Tiên	82	Tiên Khách Lai	152
Trạng Nguyên	173	Vạn Thọ	81

Dữ liệu được lấy từ nhiều nguồn khác nhau như Google, Bing, Flickr và hình ảnh chụp thực tế. Các hình ảnh trong dataset sẽ được lọc kỹ, hình ảnh cần phải rõ nét chủ thể, chụp cự ly đủ gần để nhận biết và không bị lẫn vào background. Nhóm em đã sử dụng công cụ [Download Images From Search Engine](#) để có thể tìm kiếm hình ảnh một cách nhanh nhất.

Một số hình ảnh minh họa từ dataset:



Hình 2.1 Hoa Ngũ Sắc



Hình 2.2 Hoa Hồng



Hình 2.3 Hoa Mắc Cỡ

Chương 3: CÔNG CỤ SỬ DỤNG

3.1 CNN Algorithm

Convolutional Neural Networks, hay còn được gọi là ConvNets lần đầu giới thiệu vào những năm 1980s bởi Yann Lecun, một nhà khoa học máy tính người Pháp làm việc trong lĩnh vực máy học, thị giác máy tính, robot di động và khoa học thần kinh tính toán.



Hình 3.1 Chân dung Yann Lecun

Lecun đã xây dựng dựa trên công trình của Kunihiko Fukushima, một nhà khoa học người Nhật đã sáng chế ra Neocognitron, một mạng neurol nhận diện hình ảnh.



3.2 Model ResNet-50

ResNet-50 là một deep neural network được sử dụng làm nền tảng cho nhiều ứng dụng thị giác máy tính như phát hiện vật thể, phân loại hình ảnh,... ResNet được tạo ra bởi 4 nhà nghiên cứu là Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun trong một bài báo khoa học *Deep Residual Learning for Image Recognition* và nó cũng chiến thắng trong cuộc *ImageNet Challenge* (ILSVRC) năm 2015 với tỉ lệ lỗi chỉ 3.57%.

```
tf.keras.applications.ResNet50(  
    include_top=True,  
    weights="imagenet",  
    input_tensor=None,  
    input_shape=None,  
    pooling=None,  
    classes=1000,  
    **kwargs  
)
```

Hình 3.3 Function ResNet50 trong thư viện Keras

Hiện tại 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.

Đột phá của ResNet vào lúc đó chính là khả năng huấn luyện deep neural networks với hơn 150 lớp. Trước khi ResNet xuất hiện, việc huấn luyện deep neural networks sẽ gặp tình trạng Vanishing Gradient dẫn tới quá trình học không được tốt.

3.3 Thư viện Keras



Hình 3.4 Logo Keras

Keras là một thư viện được phát triển vào năm 2015 bởi Francois Chollet, là một kỹ sư nghiên cứu deep learning tại Google.



Hình 3.5 Chân dung Francois Chollet

Keras là một open source cho neural network được viết bởi ngôn ngữ Python. Keras là một API bậc cao có thể sử dụng chung với các thư viện deep learning nổi tiếng như Tensorflow (được phát triển bởi Google), CNTK (được phát triển bởi Microsoft), Theano (người phát triển chính Yoshua Bengio).

Keras có nhiều ưu điểm cho người dùng như:

- Tối giản, dễ sử dụng. Keras giảm tải công việc cho lập trình viên, giúp cho họ tập trung hơn vào công việc của họ.
- Keras đã được sử dụng rộng rãi trong doanh nghiệp và cộng đồng nghiên cứu, bao gồm cả NASA, Youtube và Waymo.
- Keras được hỗ trợ huấn luyện trên nhiều GPU phân tán.
- Keras hỗ trợ đa backend engines và không giới hạn bạn vào một hệ sinh thái, bao gồm Tensorflow, CNTK và Theano.
- Có thể chạy trên cả CPU và GPU.
- Hỗ trợ xây dựng CNN, RNN và có thể kết hợp cả 2.

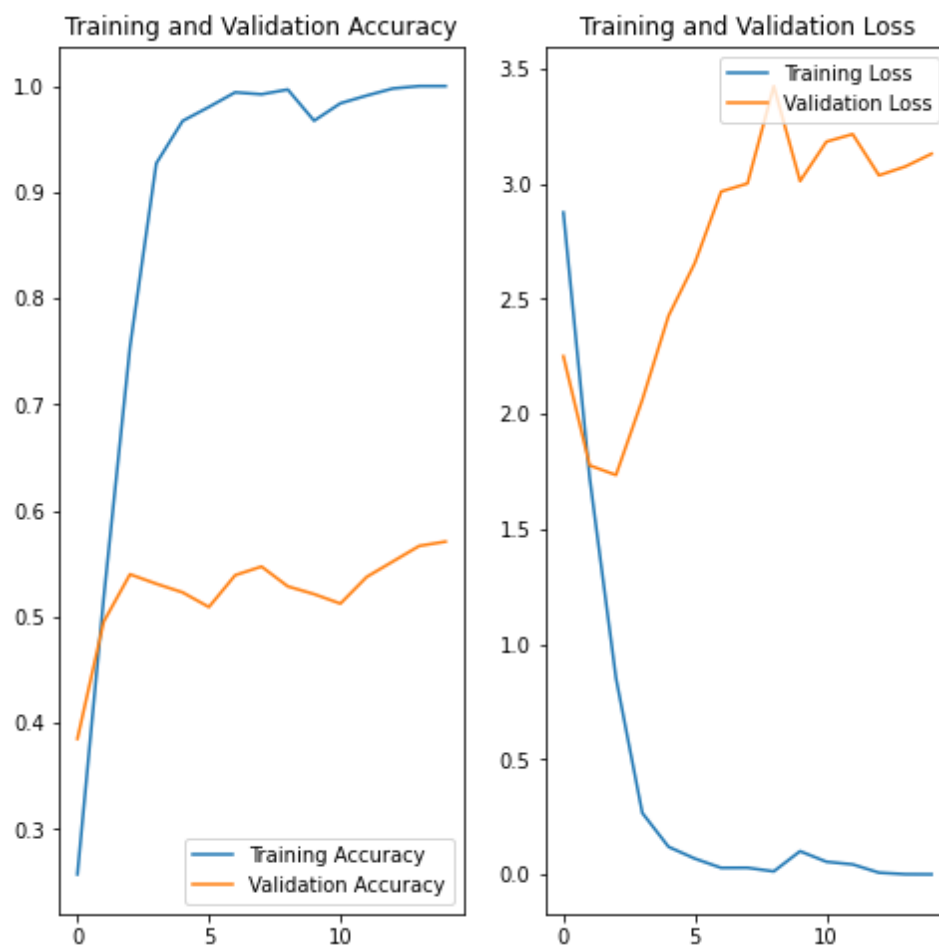
Chương 4: TRAIN MODEL VÀ ĐÁNH GIÁ

4.1 Train model lần đầu

Đầu tiên ta chỉnh size của hình ảnh về kích thước 256 x 256. Đây là kích thước phù hợp nhất để hình ảnh thu được nhiều chi tiết nhất nhưng kích thước không quá lớn, phù hợp với deep learning.

Tiếp theo ta chuẩn hóa dữ liệu, chuyển giá trị RGB từng pixel trong ảnh từ [0, 255] về [0, 1]. Giá trị như vậy phù hợp với deep learning hơn.

Sau đó ta tạo model CNN và train model đó bằng Keras. Ta được biểu đồ accuracy và loss của 2 tập training và validation như sau.



Hình 4.1 Biểu đồ model đầu tiên

Nhìn vào biểu đồ ta có thể thấy model đã gặp hiện tượng Overfitting, khi accuracy của tập train đạt giá trị rất lớn nhưng accuracy của tập validation thì lại rất thấp. Để tránh hiện tượng này ta cần phải xử lý Overfitting.

4.2 Xử lý Overfitting

Overfitting là hiện tượng model không học tốt nhất, khi accuracy của data train đạt tới tuyệt đối nhưng accuracy của data validate rất thấp. Biểu hiện rõ nhất là giá trị loss function của tập validate rất cao. Nhóm em sử dụng 2 phương pháp sau để xử lý Overfitting:

- Data Augmentation
- Drop Layer

4.2.1 Data Augmentation

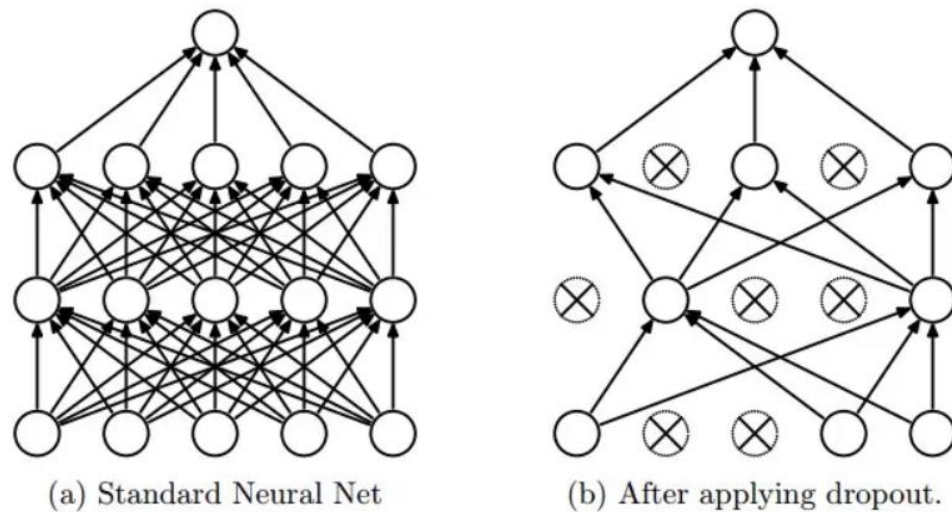
Data Augmentation là phương pháp tạo thêm ảnh từ các ảnh dữ liệu ban đầu bằng cách Zoom, Rotation và Flip để tạo được thêm ảnh mới.



Hình 4.2 Hình ảnh sau khi xử lý Data Augmentation

4.2.2 Drop Layer

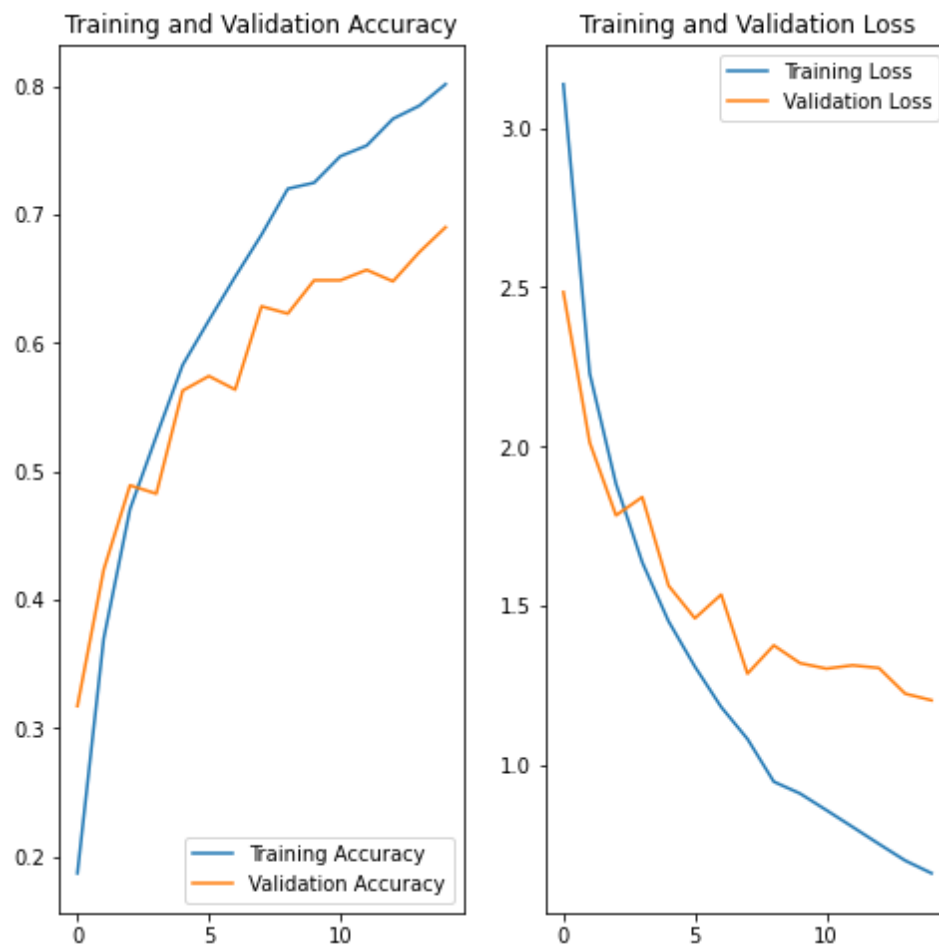
Đây là kỹ thuật cho phép ta bỏ qua vài uni trong quá trình train model một cách ngẫu nhiên để giảm việc các nút trong mạng phụ thuộc lẫn nhau.



Hình 4.3 Phương pháp hoạt động của Drop Layer

4.3 Train model sau khi xử lý Overfitting

Sau khi xử lý Overfitting nhóm đã train lại model và được kết quả khả quan hơn khá nhiều.

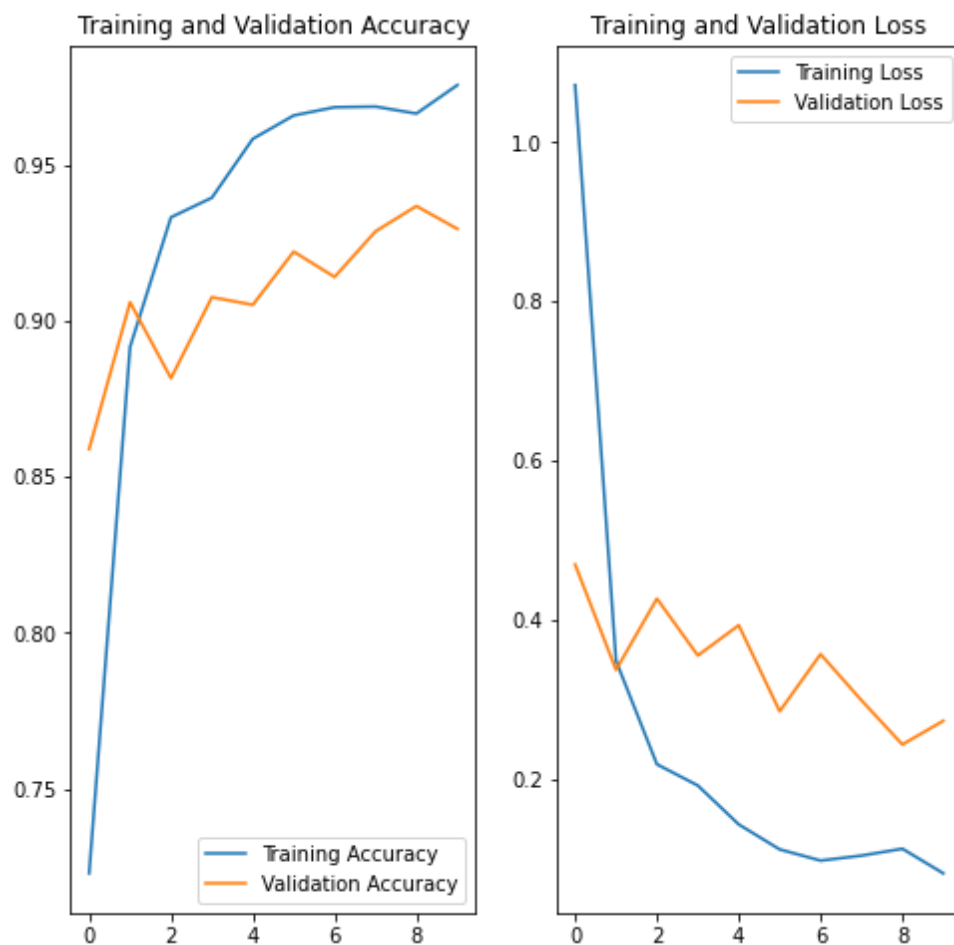


Hình 4.4 Biểu đồ của model sau khi xử lý Overfitting

Model đã không còn gặp tình trạng Overfitting và Accuracy cũng đã được cải thiện. Tuy nhiên Accuracy vẫn chưa thực sự cao với 0.69 và giá trị Loss Function vẫn còn khá cao tới tận 1.203.

4.4 Train model ResNet-50

Để có thể tăng độ chính xác cho model thêm nữa, nhóm đã sử dụng ResNet-50 kết hợp với Data Augmentation và Drop Layer.



Hình 4.4 Biểu đồ của model ResNet-50

Model đã đạt được Accuracy cao nhất trong cả 3 model với giá trị cao nhất lên tới 0.937, giá trị Loss Function cũng rất thấp với giá trị thấp nhất chỉ là 0.273 cho thấy khả năng học của model rất tốt.

4.5 Đánh giá

4.5.1 Accuracy

Đầu tiên ta đánh giá Accuracy của model, đây là cách đơn giản và hay sử dụng nhất. Cách đánh giá này đơn giản là tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu. Việc tính Accuracy của model đã được tích hợp sẵn trong thư viện Keras.

Model	Accuracy	Top 5 Accuracy	Min Loss
Tự build	0.571	0.552	1.735
Data Augmentation + Drop Layer	0.69	0.663	1.203
Data Augmentation + Drop Layer + ResNet-50	0.937	0.946	0.273

4.5.2 Confusion Matrix

Việc đánh giá với Accuracy mới chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác. Để có thể đánh giá được giá trị này nhóm đã sử dụng Confusion Matrix.

Dựa vào Confusion Matrix của tập Validation, ta có thể thấy rằng:

- Sau khi xem hình ảnh của hoa thì nhóm em thấy rằng các hoa này thường có màu và đặc điểm giống nhau. Ví dụ như hoa Anh Túc và hoa Mao Lương đều có màu vàng, chúng chỉ khác nhau ở nhụy, với hoa Anh Túc là nhụy vàng còn hoa Mao Lương là nhụy xanh.



Hình 4.7 Hoa Anh Túc và hoa Mao Lương

Chương 5: KẾT LUẬN

5.1 Thuận lợi

Qua quá trình làm đồ án và qua môn học, nhóm chúng em có những thuận lợi sau:

- Biết được cách khảo sát ứng dụng, từ đó lên ý tưởng và thiết kế một giao diện khá hoàn chỉnh đảm bảo chức năng cơ bản của ứng dụng đọc truyện đáp ứng được nhu cầu của người dùng.
- Học hỏi thêm được các kỹ năng làm việc nhóm.
- Học hỏi thêm được các thư viện công nghệ mới như CNN, model Keras-50, các kỹ năng Data Augmentation.
- Hiểu và áp dụng được các quy tắc, quy trình để thiết kế được một ứng dụng hoàn chỉnh bao gồm Front-end, Back-end và Database.

5.2 Khó khăn

Bên cạnh các thuận lợi đã kể trên, nhóm vẫn còn một số khó khăn trong quá trình thực hiện đồ án như:

- Vì lần đầu học máy học nên vẫn còn nhiều thứ chưa quen.
- Việc hoạt động nhóm trên nền tảng online có một số khó khăn cho việc thảo luận nhóm.

5.3 Định hướng phát triển ứng dụng

Sau đây là một số định hướng phát triển mà nhóm dự định sẽ làm trong tương lai gần:

- Tạo một ứng dụng di động sử dụng giúp người dùng nhận diện loài hoa qua việc chụp ảnh hoặc lấy hình ảnh từ điện thoại.
- Thu thập thêm dữ liệu các loại hoa khác.
- Hỗ trợ thêm khả năng phân biệt được hình ảnh có hoa và hình ảnh không có hoa.

TÀI LIỆU THAM KHẢO

TÀI LIỆU THAM KHẢO:

1. “*What are convolutional neural networks (CNN)?*”, Techtalk [Trực tuyến]. Địa chỉ: <https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets/#:~:text=Convolutional%20neural%20networks%2C%20also%20called,a%20postdoctoral%20computer%20science%20researcher> [Truy cập lần cuối 8/2/2021]
2. “*About Keras*”, Keras [Trực tuyến]. Địa chỉ: <https://keras.io/about/> [Truy cập lần cuối 8/2/2022]
3. “*Deep Residual Learning for Image Recognition*”, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jiansun [Trực tuyến]. Địa chỉ: <https://arxiv.org/abs/1512.03385> [Truy cập lần cuối 2/8/2022]
4. “Giới thiệu mạng ResNet”, To Duc Thang [Trực tuyến]. Địa chỉ: <https://viblo.asia/p/gioi-thieu-mang-resnet-vyDZOa7R5wj> [Truy cập lần cuối 8/2/2022]
5. “*Các phương pháp đánh giá một hệ thống phân lớp*”, Machine Learning cơ bản [Trực tuyến]. Địa chỉ: <https://machinelearningcoban.com/2017/08/31/evaluation/> [Truy cập lần cuối 8/2/2022]

SOURCE CODE, CÔNG CỤ SỬ DỤNG:

6. “*102 Flowers Classification*”, Muhammad MujtabaSaeed [Trực tuyến]. Địa chỉ: <https://github.com/Muhammad-MujtabaSaeed/102-Flowers-Classification>

7. “*Kaggle Flowers Classification Keras*”, RockyXu66 [Trực tuyến]. Địa chỉ:
https://github.com/RockyXu66/Kaggle_Flowers_Classification_Keras
8. “*Download Images From Search Engine*”, 18520339 [Trực tuyến]. Địa chỉ:
<https://github.com/18520339/image-search-downloader>