

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



---

## MACHINE LEARNING

### LAB 2: CLASSIFICATION

---

*Lớp:* Nhập môn học máy 18\_21

*Nhóm thực hiện:*

18120078 – Ngô Phù Hữu Đại Sơn

18120253 – Mai Ngọc Tú

18120201 – Nguyễn Bảo Long

## MỤC LỤC

|   |    |
|---|----|
| A. Thông tin khái quát .....            | 2  |
| I. Thông tin nhóm .....                 | 2  |
| II. Bảng phân công công việc .....      | 2  |
| B. Nội dung .....                       | 3  |
| I. Mục tiêu của đồ án: .....            | 3  |
| II. Yêu cầu: .....                      | 3  |
| III. Triển khai .....                   | 3  |
| 1. Phân tích bài toán.....              | 3  |
| 2. Phân tích tập dữ liệu hình ảnh ..... | 6  |
| 3. Mạng Neuron sử dụng .....            | 8  |
| 4. Cài đặt mạng Neuron .....            | 10 |
| 5. Báo cáo kết quả.....                 | 12 |
| IV. Đánh giá đồ án.....                 | 12 |
| VII. Nguồn tham khảo .....              | 12 |

## A. THÔNG TIN KHÁI QUÁT

### I. Thông tin nhóm

| MSSV     | Họ tên              | Vai trò     |
|----------|---------------------|-------------|
| 18120078 | Ngô Phù Hữu Đại Sơn | Nhóm trưởng |
| 18120253 | Mai Ngọc Tú         | Thành viên  |
| 18120201 | Nguyễn Bảo Long     | Thành viên  |

### II. Bảng phân công công việc

| MSSV     | Công việc phụ trách   | Mức độ hoàn thành |
|----------|---|-------------------|
| 18120078 | - Thiết kế, huấn luyện model<br>- Vẽ các biểu đồ trực quan      | 100/100           |
| 18120253 | - Tổng hợp báo cáo  | 100/100           |
| 18120201 | - Thiết kế, huấn luyện model<br>- Phân tích bài toán và dữ liệu | 100/100           |

## B. NỘI DUNG

### I. Mục tiêu của đồ án:

- Tìm hiểu và vận dụng các kỹ thuật:
  - **Nhận diện bài toán:** tìm hiểu bài toán và dữ liệu được giao nhằm xác định nội dung và ý nghĩa bài toán thực tế cần giải quyết. Thông qua đó, sinh viên có khả năng ánh xạ vấn đề thực tế sang bài toán lập trình.
  - **Giải quyết vấn đề:** đưa ra các giải pháp và hướng tiếp cận nhằm giải quyết được yêu cầu bài toán thực tế.
  - **Xử lý và phân tích dữ liệu:** xử lý các công cụ phân tích dữ liệu tự động nhằm tìm ra các thông tin hữu ích, các đặc trưng tiềm ẩn ảnh hưởng đến mục tiêu bài toán.
  - **Thiết kế và cài đặt các thuật toán đã học:** đề xuất, triển khai và giải thích các thuật toán mạng Noron đã học nhằm giải quyết bài toán được giao.

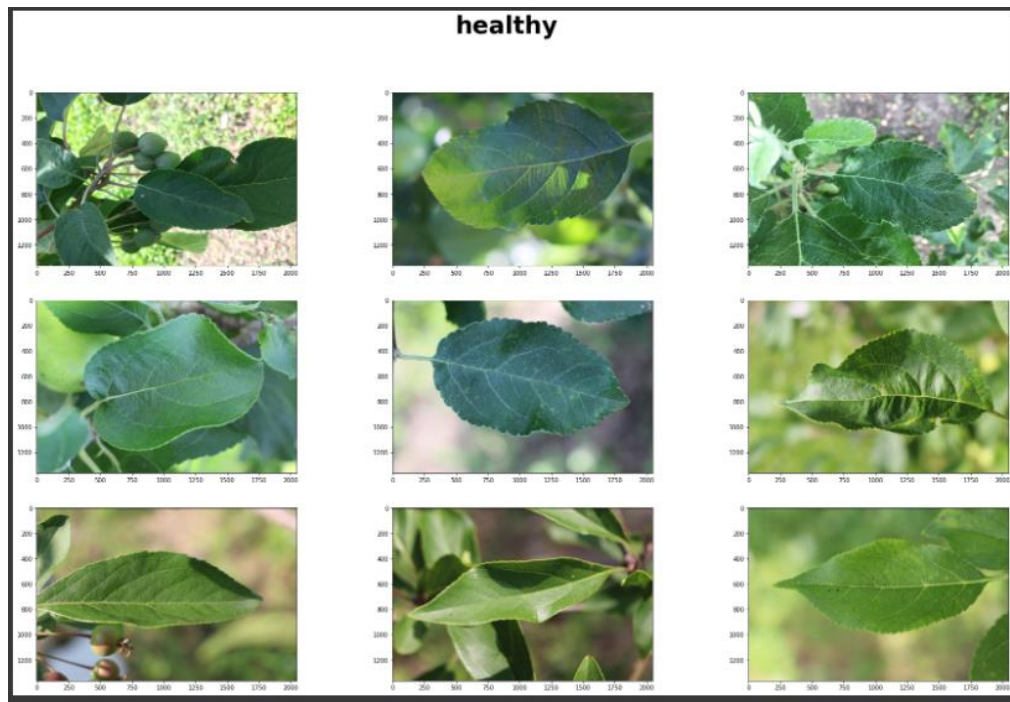
### II. Yêu cầu:

- Phân tích kỹ bài toán và tập dữ liệu hình ảnh được cung cấp. Chọn lựa và trình bày kiểu mạng nơron để giải quyết bài toán.
- Cài đặt mạng Neuron.
- Báo cáo kết quả đạt được sau quá trình phân tích và cài đặt.

### III. Triển khai

#### 1. Phân tích bài toán

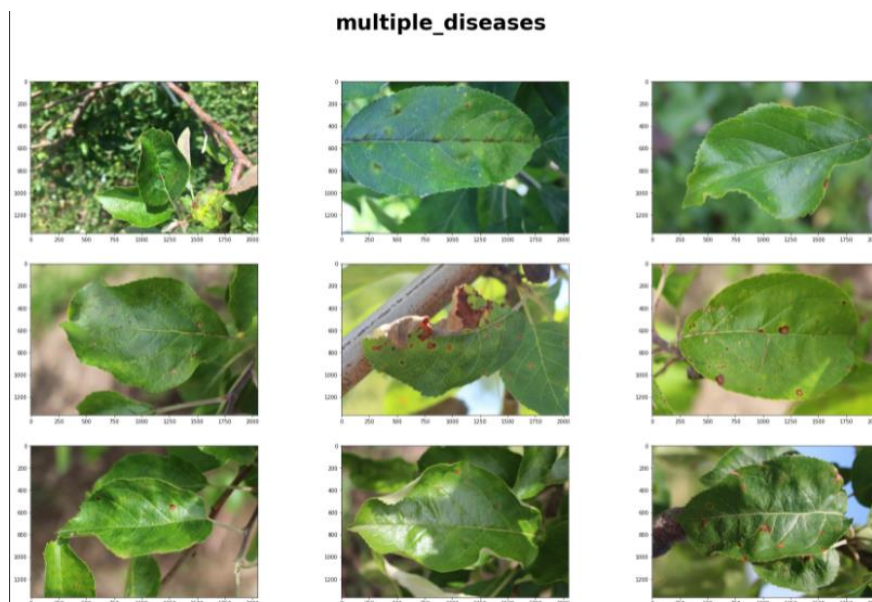
- Bài toán đưa ra 4 loại trạng thái bệnh trên lá cây đang mắc phải để phân lớp:
  - + **Healthy:**
    - Lá cây xanh đều, không chứa các đốm màu khác thường.
    - Bề mặt nhẵn, không chứa biến dạng.
    - Xương lá không bị biến sắc hay biến dạng.
    - Hình dạng lá nguyên vẹn, không bị mất mát, biến dạng, biến sắc.



*Hình ảnh trực quan của Healthy*

**+ Multiple\_diseases:**

- Bề mặt lá chứa các đốm bệnh có màu sắc bất thường (đen, nâu,...), có thể bị biến dạng (cong, co lại bất thường...).
- Có thể tồn tại các đốm bệnh chạy dọc xương lá.
- Hình dáng lá bị biến dạng (bị cắt góc, cháy xém,...).

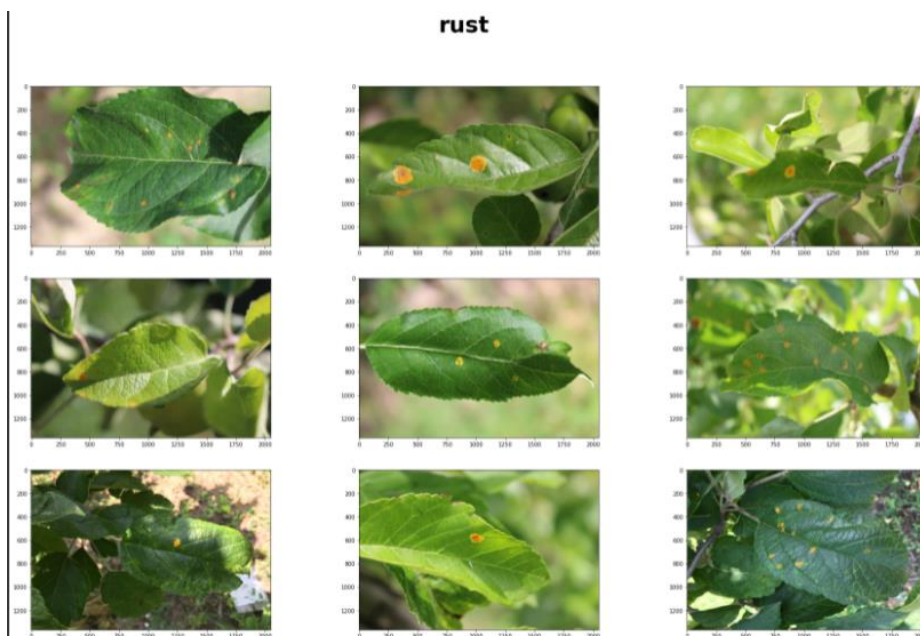


*Hình ảnh trực quan của Multiple\_diseases*



### + Rust:

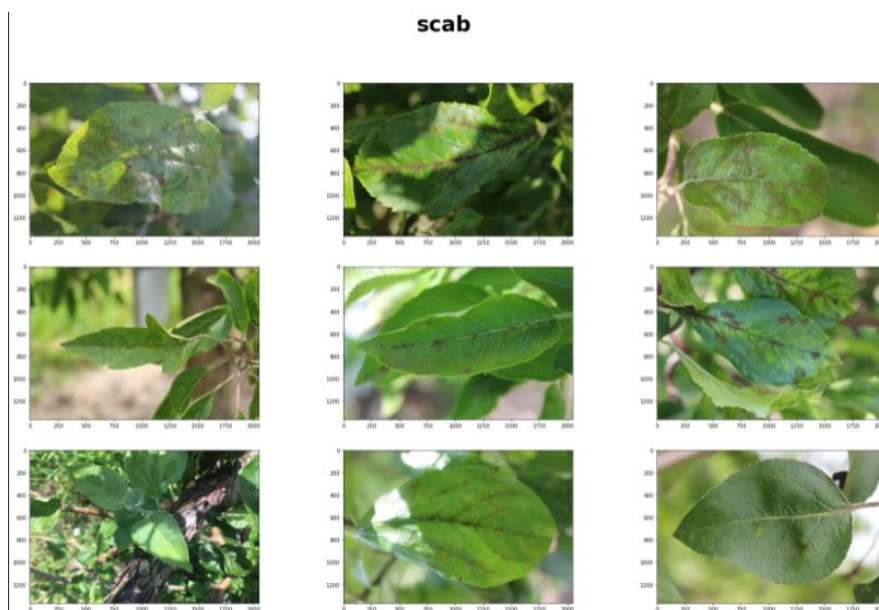
- Bề mặt lá xuất hiện các đốm màu nâu, vàng.
- Các đốm này xuất hiện thành từng cụm, lan rộng về mật độ và diện tích từng đốm.
- Các đốm xuất hiện ở cả mặt trước và mặt sau của lá.



*Hình ảnh trực quan cho Rust*

### + Scab:

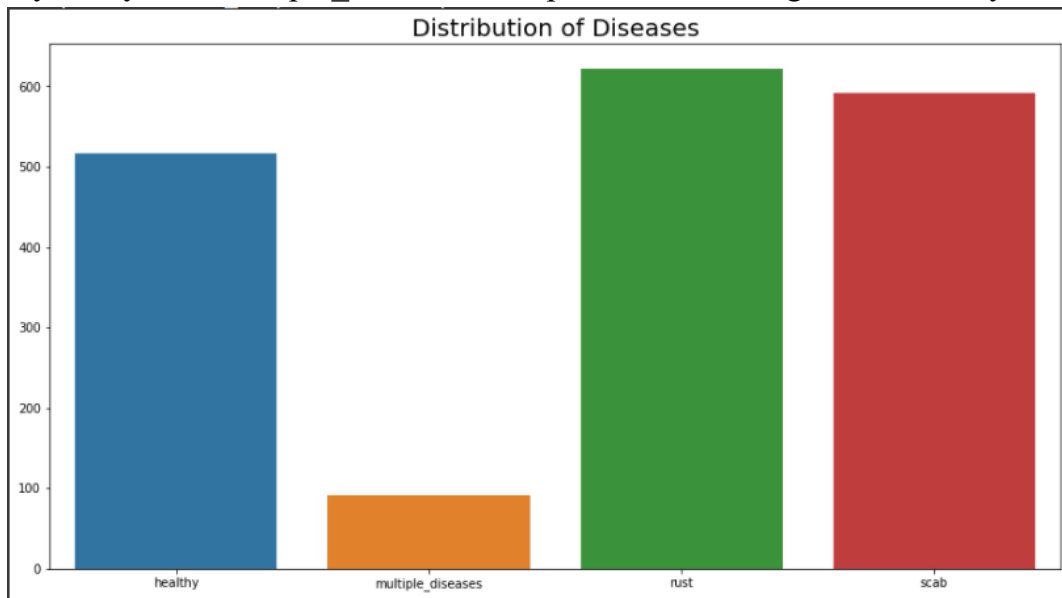
- Có các vết bệnh sẫm màu, xanh ô liu, tím hoặc đen trên lá, thường có hình dạng bất thường và trở nên đen khi bệnh quá nặng.
- Có các vết đốm bệnh chạy dọc xương lá. Khi lan rộng thì tỏa khắp mặt lá.



*Hình ảnh trực quan cho Scab*

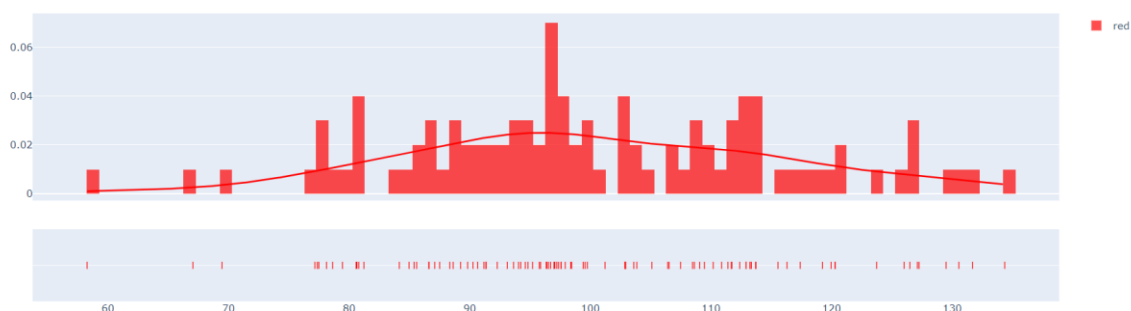
## 2. Phân tích tập dữ liệu hình ảnh

- Nhận xét phân phối số lượng loại bệnh trong tập Train:
  - + Tỷ lệ cây khỏe mạnh (healthy), cây bị bệnh rust, cây bị bệnh scab là tương đương nhau. Trong đó, cây bị bệnh rust nhiều hơn một chút về mặt số lượng.
  - + Tỷ lệ cây mắc multiple\_diseases là thấp nhất, chỉ khoảng 1/5 so với cây healthy.



*Phân phối số lượng các loại bệnh trong tập Train*

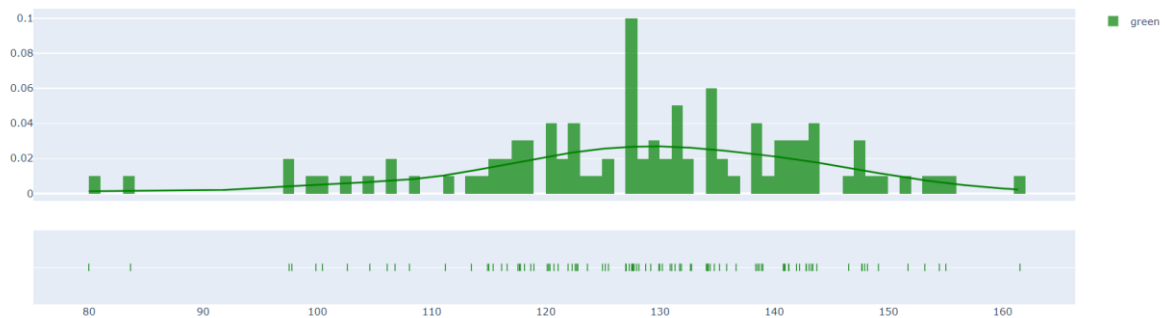
- Phân tích tập dữ liệu Train theo thang đo RGB:
  - + **Kênh đỏ (Red):**
    - Các giá trị của kênh màu đỏ gần như phân phối chuẩn với tâm phân phối vào khoảng 97.
    - Điều này cho thấy rằng kênh màu đỏ có xu hướng tập trung nhiều hơn ở các giá trị xung quanh 97. Chúng ta sẽ thấy rằng độ đỏ trong hình ảnh được ghi nhận ở mức độ thấp.



*Phân bố kênh màu đỏ của các hình ảnh dữ liệu*

#### + Kênh xanh lá (Green):

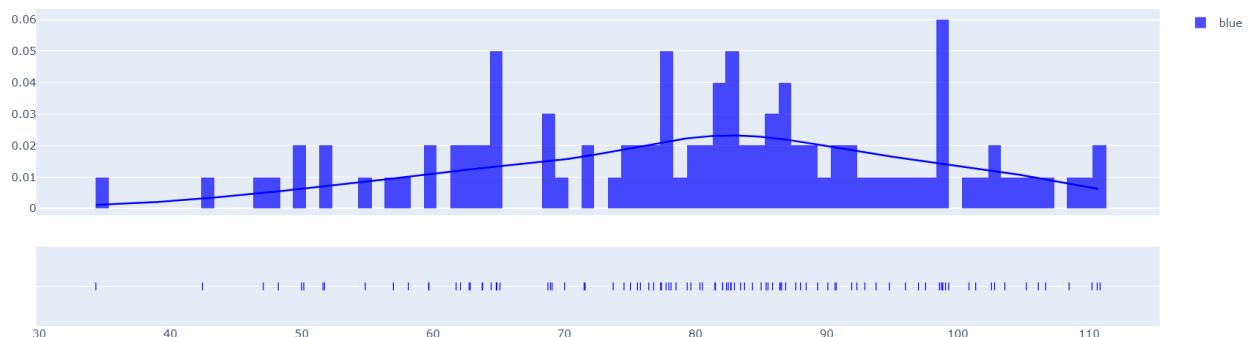
- Giá trị kênh màu xanh lá có hình dáng của phân phối chuẩn và phân bố đồng đều hơn giá trị kênh màu đỏ.
- Các giá trị tập trung quanh tâm phân phối là 140, lớn hơn giá trị tâm của phân phối kênh màu đỏ.
- Điều này hoàn toàn hợp lý khi chúng ta đang khảo sát hình ảnh lá cây.



*Phân phối kênh màu xanh lá của hình ảnh dữ liệu*

#### + Kênh xanh dương (Blue):

- Kênh màu xanh dương có sự phân bố đồng đều nhất trong số 3 kênh màu.



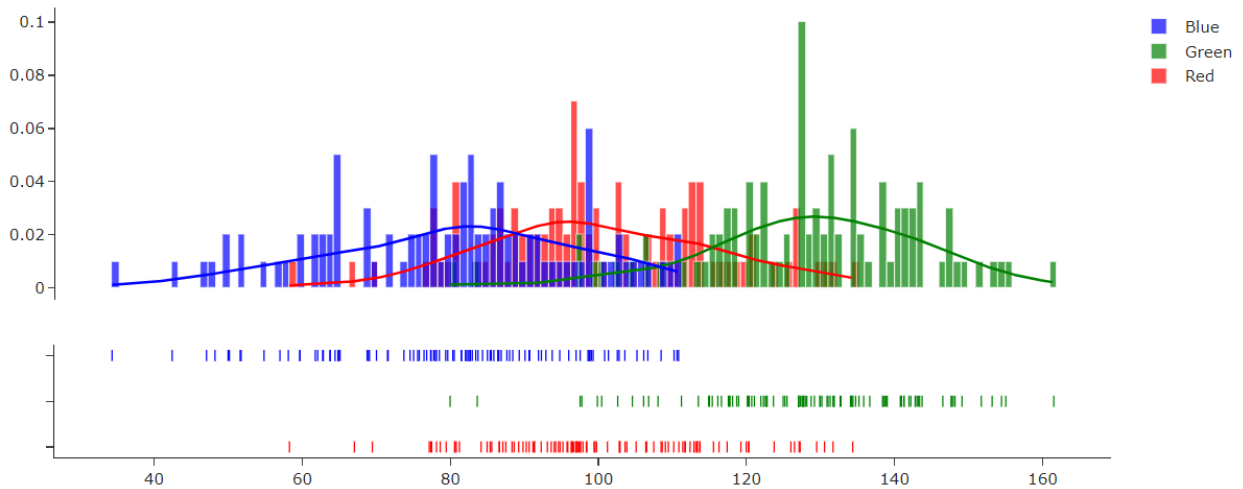
*Phân bố kênh màu xanh dương của hình ảnh dữ liệu*

#### + Quan sát tổng thể phân phối giữa 3 kênh màu, ta có thể nhận thấy:

- Kênh xanh lá có sắc độ cao nhất do các mẫu khảo sát là lá cây.
- Kênh xanh dương có sắc độ thấp nhất trong số 3 kênh. Quan sát thực tế các mẫu cây bệnh scab, có thể thấy sắc tối rất phổ biến trong các đốm bệnh. Nguyên nhân của việc này chính là sự pha trộn màu xanh dương vào các pixel khiến cho màu tối lại. Vậy có thể kết luận rằng, các mẫu ảnh có kênh xanh dương cao sẽ có khả năng bị mắc bệnh scab cao.
- Kênh đỏ nghi ngờ có liên quan đến bệnh rust vì quan sát thực hiện trên mẫu bệnh này chỉ ra rằng, có sự xuất hiện của các đốm vàng và cam trên lá cây



## Distribution of all channel values



*Phân phối tổng hợp thang đo RGB của tập dữ liệu Train*

### 3. Mạng Neuron sử dụng

- Nhóm sử dụng mạng Neuron **CNN (Convolutional Neural Network)**
- Khi khởi tạo một mạng học sâu, chúng ta thường khởi tạo các tạo số có giá trị bằng 0 hoặc gần bằng 0 (Gieo ngẫu nhiên từ phân phối chuẩn tắc) hoặc đôi khi ta còn thực hiện các kỹ thuật chuẩn hóa tham số ( $L_2$ ) khiến cho quá trình học ban đầu rất chậm và thậm chí còn chậm hơn nếu mạng có nhiều lớp.
- Để giải quyết vấn đề này, nhóm sử dụng thuật toán **ResNet152V2**.
- Ý tưởng của ResNet rất đơn giản, giả sử ta có một mô hình mạng học sâu gồm 2 lớp ẩn, đầu vào ( $x$ ) ngoài đi theo còn luồng chính của mạng qua 2 lớp ẩn, nó sẽ được truyền theo một con đường khác và được cộng vào đầu ra của lớp thứ hai trước khi đưa vào hàm kích hoạt (ReLU).
- Có thể thấy khi  $F(x) \approx 0$  thì  $H(x) = F(x) + x \approx x$ . Nhờ vậy mà bảo toàn được các giá trị của  $x$  đến các lớp sau của mạng. Một tổ hợp mạng như vậy được gọi là *Residual Block*.

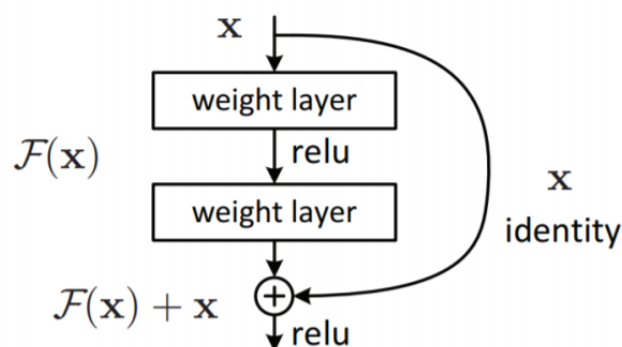


Figure 36: Residual Block

- Bằng cách nối các *Residual block* lại với nhau, ta được một mạng *Residual Network*.

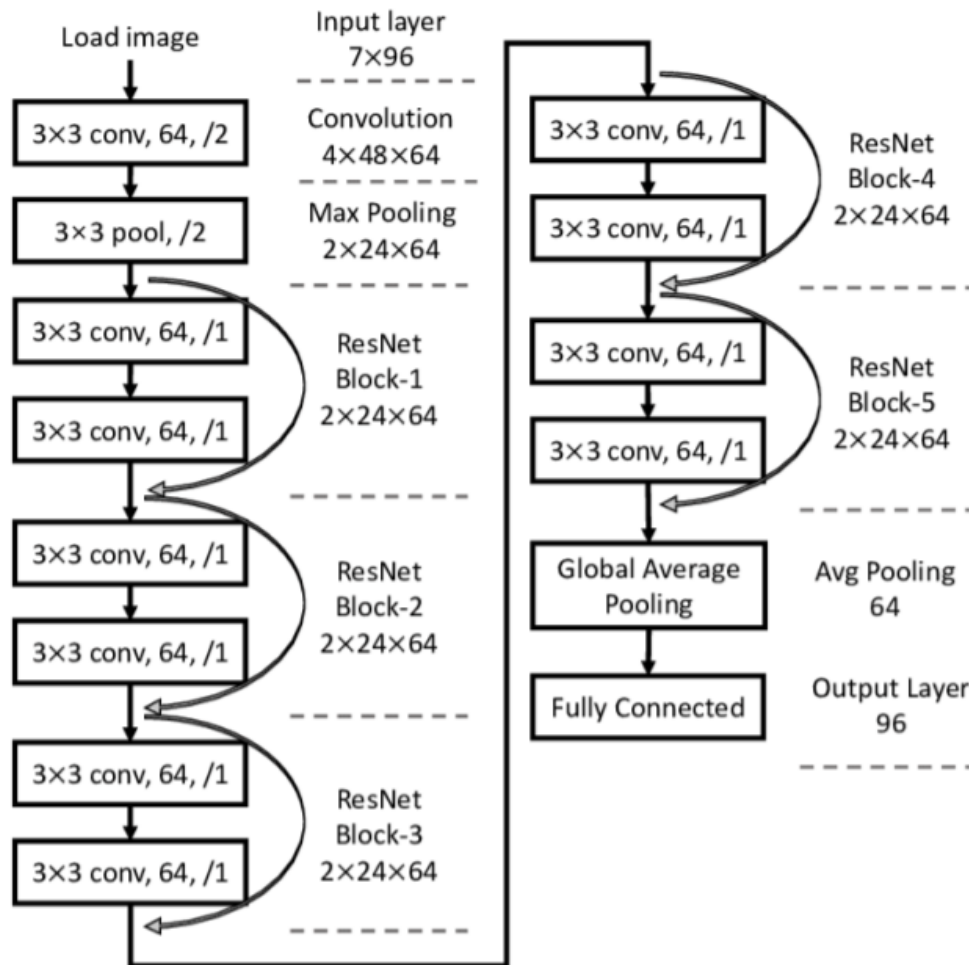


Figure 37: Residual Network

- Để có thể xây dựng các mạng ResNet có độ sâu lên đến hàng trăm lớp, ta cần có 1 cấu trúc tiết kiệm tài nguyên hơn so với các *Residual block*. Mỗi *BottBottle block* được hình thành từ 3 lớp CONV thay vì 2 như *Residual block*. Các lớp CONV lần lượt là  $1 \times 1$ ,  $3 \times 3$  và  $1 \times 1$ . Lớp CONV  $1 \times 1$  đầu tiên có chức năng giảm chiều của ảnh, từ đó mà lớp CONV  $3 \times 3$  có thể thực hiện phép tích chập với dữ liệu có chiều nhỏ hơn, giảm số lượng các phép tính toán. Và sau đó lớp CONV  $1 \times 1$  cuối cùng khôi phục lại chiều ban đầu của của ảnh.

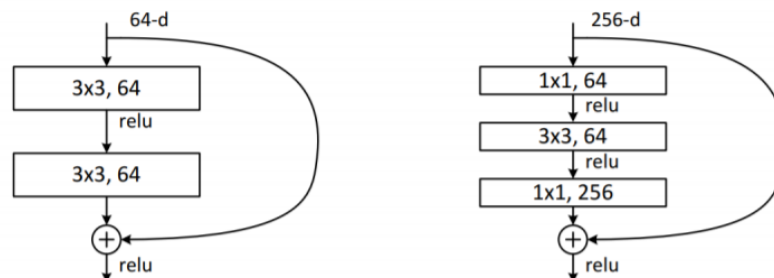
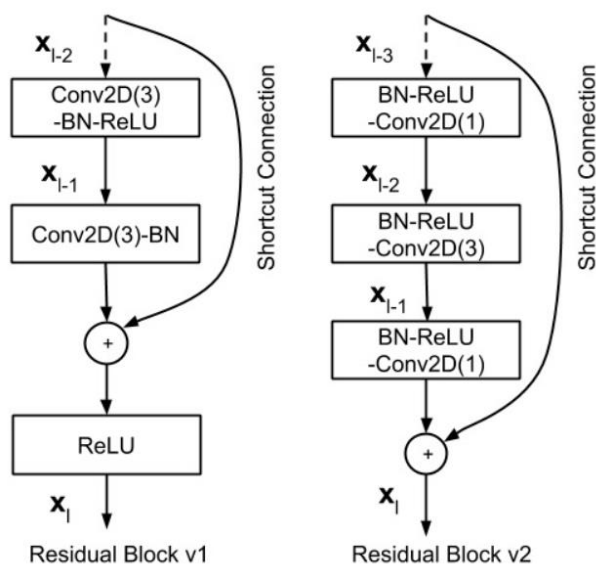


Figure 39: Residual block ( $56 \times 56$  feature maps) dùng trong ResNet-34 (trái). BottBottleneck block dùng trong mạng ResNet-50/101/152.

- Thuật toán ResNetV2 sẽ dùng *Bottleneck block* để giảm bớt tham số, vì vậy có thể huấn luyện sâu hơn, trong khi thuật toán ResNetV1 dùng *Residual block*.



Các cấu trúc mạng Residual Block

## 4. Cài đặt mạng Nơ ron

### a. Chuẩn bị dữ liệu:

Dữ liệu trong tập huấn luyện được chia thành 2 tập với tỉ lệ: (train : valid) = (9:1)

Table 1 Bảng phân bố dữ liệu trên tập huấn luyện

|                            |                | Loại bệnh |                   |      |      | Tổng số phần tử mỗi tập |
|----------------------------|----------------|-----------|-------------------|------|------|-------------------------|
|                            |                | Healthy   | Multiple diseases | Rust | Scab |                         |
| Tập huấn luyện             | Tập Train      | 464       | 81                | 559  | 532  | 1636                    |
|                            | Tập Validation | 52        | 10                | 63   | 60   | 185                     |
| Số ảnh thuộc các loại bệnh |                | 516       | 91                | 622  | 592  | 1821                    |

- Tăng cường dữ liệu:

+ Tập train:

- `rotation_range = 40,`
- `width_shift_range = 0.2,`
- `height_shift_range = 0.2,`
- `shear_range = 0.2,`
- `zoom_range = 0.2,`
- `horizontal_flip`
- `fill_mode: nearest`

+ Tập valid:

- `rotation_range = 40,`
- `width_shift_range = 0.2,`
- `height_shift_range = 0.2,`
- `shear_range = 0.2,`
- `zoom_range = 0.2,`
- `horizontal_flip`
- `fill_mode: nearest`

## b. Khởi tạo mạng

- Cấu trúc mạng học sâu gồm các thành phần được nối tiếp với nhau theo thứ tự như sau:
  - Backbone (*ResNet101V2*, *ResNet152V2*, *InceptionResNetV2*,...) với tham số khởi tạo của “ImageNet”.
  - Lớp *Average Pooling 2D*, với các filter có kích thước (5 x 5).
  - Lớp *Global Average Pooling* có chức năng đưa dữ liệu về 1 vector với mỗi phần tử tương ứng với giá trị trung bình của một feature map. Sau đó có thể input vào mạng *Fully Connected*.
  - Lớp Dense có 64 nút, hàm kích hoạt *ReLU*.
  - Lớp Dense có 32 nút, hàm kích hoạt *ReLU*.
  - Lớp Dense có 4 nút, hàm kích hoạt *Softmax*.

## c. Huấn luyện

- Mạng sau khi khởi tạo sẽ được huấn luyện với Max Accuracy là 0.97 và Epoch bằng 30.

## 5. Báo cáo kết quả

- Kết quả kiểm định trên tập Test và submit [tại đây](#):

| Model             | Private Score | Public Score |
|-------------------|---------------|--------------|
| ResNet152V2       | 0.94080       | 0.93392      |
| ResNet101V2       | 0.93794       | 0.92410      |
| MobileNetV2       | 0.92132       | 0.92592      |
| InceptionNet      | 0.91775       | 0.93327      |
| InceptionResNetV2 | 0.80610       | 0.81000      |

## IV. Đánh giá đồ án

| MSSV     | Mức độ hoàn thành công việc | Đóng góp |
|----------|-----------------------------|----------|
| 18120078 | 100/100                     | 33/100   |
| 18120253 | 100/100                     | 33/100   |
| 18120201 | 100/100                     | 33/100   |

## VII. Nguồn tham khảo

- <https://www.kaggle.com/tarunpapuraju/plant-pathology-2020-eda-models>
- [https://www.youtube.com/watch?v=tDaGT4N4aCA&list=PLZbbT5o\\_s2xrwRnXk\\_yCPtnqgo4\\_u2YGL](https://www.youtube.com/watch?v=tDaGT4N4aCA&list=PLZbbT5o_s2xrwRnXk_yCPtnqgo4_u2YGL)
- <https://keras.io/api/applications/>