

IDENTIFICATION OF SOME TYPES OF LONGAN (THROUGH LEAVES) USING IMAGE AND DEEP LEARNING TECHNOLOGY

Le Thi Diem¹, Nguyen Duc Thien¹, Truong Quoc Bao^{2*}

¹Soc Trang Vocational College, ²Can Tho University

ARTICLE INFO	ABSTRACT
Received: 17/11/2022 Revised: 11/01/2023 Published: 11/01/2023	In the Mekong Delta region, besides rice being the most important tree, longan has been planted in a large area with various families, bringing high income to the farmers. Each of them has different properties in harvesting time, flowering, and supervising. In fact, the farmers must regularly monitor, take care and rely on their experience to choose the right time to intervene in each type of longan for increasing productivity. From longan pictures taken in the gardens, it is not easy to determine the type of longan excluding knowledgeable people and scientists. Therefore, it is necessary to help apply technology to assess the kind of longan-by-leaf photographs taken of longan. The study used data set of 3 types of longan leaves, namely Ido, Thach Kiet, and Dimocarpus Longan Lour in the total of 2182 collected pictures. Using deep learning techniques with the VGG16 model to train the obtained data, the accuracy result was 98.3%. Thus, the research results can help agronomists and researchers have special measures to support and cooperate with the farmers in classifying and orienting to plant suitable longan with the highest efficiency.
KEYWORDS	
Convolutional Neural Network	
VGG16	
Deep learning	
Computer vision	
Transfer learning	

NHẬN DẠNG MỘT SỐ LOẠI NHẪN (THÔNG QUA LÁ NHẪN) DÙNG CÔNG NGHỆ ẢNH VÀ KỸ THUẬT HỌC SÂU

Lê Thị Diễm¹, Nguyễn Đức Thiện¹, Trương Quốc Bảo^{2*}

¹Trường Cao đẳng Nghề Sóc Trăng, ²Trường Đại học Cần Thơ

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 17/11/2022 Ngày hoàn thiện: 11/01/2023 Ngày đăng: 11/01/2023	Khu vực đồng bằng sông Cửu Long, bên cạnh cây lúa, nhãn là loại cây có diện tích trồng khá lớn, với nhiều loại nhãn khác nhau, đem lại thu nhập cao cho người nông dân. Mỗi loại nhãn sẽ có đặc điểm khác nhau về thời gian thu hoạch, cách xử lý ra hoa, cách chăm sóc. Trên thực tế, người trồng nhãn phải thường xuyên theo dõi, chăm sóc, dựa vào kinh nghiệm để xác định thời điểm cần thiết nhằm can thiệp vào từng loại nhãn giúp tăng năng suất. Từ hình ảnh chụp lá nhãn tại vườn rất khó phân biệt được loại nhãn ngoại trừ những người có kinh nghiệm, nhà khoa học. Vì vậy, việc sử dụng công nghệ để phân loại cây nhãn thông qua hình ảnh chụp của lá nhãn là rất cần thiết. Nghiên cứu sử dụng tập dữ liệu của 3 loại lá nhãn: Ido, Thạch Kiệt, xoòng cơm vàng, trong tổng số hình ảnh thu thập được là 2182 hình. Sử dụng kỹ thuật học sâu với mô hình VGG16 để huấn luyện dữ liệu thu được, kết quả độ chính xác đạt 98,3%. Như vậy, kết quả nghiên cứu có thể giúp các nhà nông học, nghiên cứu có các biện pháp cụ thể hỗ trợ, phối hợp với người nông dân trong việc phân loại, có định hướng trồng cây nhãn phù hợp đạt hiệu quả cao.
TỪ KHÓA	
Mạng nơ-ron tích chập	
Mạng học sâu VGG16	
Học sâu	
Thị giác máy tính	
Học chuyển giao	

DOI: <https://doi.org/10.34238/tnu-jst.6946>

* Corresponding author. Email: tqbao@ctu.edu.vn

1. Giới thiệu

Trong sự phát triển của khoa học kỹ thuật, thị giác máy tính được xem là tập con của trí tuệ nhân tạo sử dụng sức mạnh của máy tính để trích xuất thông tin có ý nghĩa từ các tập dữ liệu được cung cấp, các tập dữ liệu đó có thể là hình ảnh, video... Hiện nay, việc sử dụng thị giác máy tính vào việc giải quyết các bài toán liên quan đến nông nghiệp đang được áp dụng rộng rãi. Phát hiện, nhận dạng các loại lá cây là ứng dụng cơ bản trong nông nghiệp, lâm nghiệp, y học nông thôn và nhiều ứng dụng thương mại khác [1].

Một số nghiên cứu về phân loại các loài thực vật đã được đề xuất và thực hiện. Ứng dụng thị giác máy tính hỗ trợ phân loại cỏ, sử dụng cho máy diệt cỏ có chọn lọc, trình bày nền tảng phân loại có hiệu quả trong thời gian thực với AdaBoost kết hợp Naïve Bayes làm bộ phân loại cơ sở, khai thác các tính năng trực quan của việc tăng cường dữ liệu hình ảnh có chứa cỏ dại, đạt độ chính xác 94,72% với tập dữ liệu mở rộng [2].

Một kiến trúc mạng nơ-ron tích chập (CNN) mới được đề xuất và áp dụng vào nhiệm vụ phân loại lá, đưa ra so sánh giữa việc sử dụng đơn vị tuyến tính hàm mũ (ELU) được giới thiệu gần đây thay vì đơn vị tuyến tính chỉnh lưu (ReLU) là hàm phi tuyến tính của CNN; mô hình được kiểm tra trên bộ dữ liệu lá Flavia và Swedish cho thấy hiệu quả trong phân loại lá [3].

Chuẩn đoán thiếu hụt dinh dưỡng trên cây cọ dầu, dựa trên các xét nghiệm sinh hóa trên lá cọ thu thập được, mối quan hệ giữa các đặc điểm của lá cọ dầu và lượng chất dinh dưỡng của nó đã được nghiên cứu, sử dụng mạng CNN để chẩn đoán sự thiếu hụt chất dinh dưỡng trong cây cọ dầu từ hình ảnh của lá, bao gồm 3 loại: thiếu, bình thường và dư thừa [4]. Kiến trúc CSBio2020 và BettaNet tích hợp phân tách được cho là đủ lý tưởng cho bộ dữ liệu được thu thập gồm 682 hình ảnh từ 25 tán cây trong một trang trại; Độ chính xác trung bình của mô hình là 77,2% đối với CSBio2020 và 80,4% đối với BettaNet [4].

Nhận dạng thực vật đã trở thành một trọng tâm liên ngành trong cả phân loại thực vật và thị giác máy tính [5]. Mô hình học sâu được đề xuất gồm 26 lớp thuộc 8 khối được thiết kế cho việc phân loại thực vật trên quy mô lớn, dữ liệu được thu thập bằng thiết bị di động bao gồm 10.000 hình ảnh thuộc 100 loại cây cảnh trong khuôn viên Đại học Nông lâm Bắc Kinh; mô hình đề xuất đạt được tỷ lệ nhận dạng 91,78% [5].

Thảo luận về phân loại các loại lá ở Ấn Độ dựa trên mạng CNN, với dữ liệu là hình ảnh lá trên nền trắng sử dụng điện thoại thông minh [1]. Các biến thể của mô hình CNN cho thấy tính hiệu quả trong việc phân loại lá, dựa trên các đặc điểm như hình dạng, kết cấu, màu sắc và đường vân chính, ngoài ra còn có các đặc điểm thu nhỏ khác về tính đồng nhất của các mẫu cạnh, đầu lá [1].

Trong kỹ thuật học sâu, mạng CNN là một loại mạng thần kinh nhân tạo với tập hợp các lớp ẩn cố định hoặc tùy chỉnh chủ yếu được sử dụng cho các nghiên cứu liên quan đến thị giác máy tính, ứng dụng hiệu quả trong bài toán phân loại hình ảnh. Có rất nhiều mạng CNN khác nhau, từ LeNet là mạng CNN đầu tiên được phát triển và các kiến trúc khác bao gồm AlexNet, Efficient Net, NasNet, DenseNet, MobileNet, VGG16...[4]. Trong đó, mạng CNN VGG16 đã được sử dụng hiệu quả trong nhiều nghiên cứu khác nhau. Nghiên cứu trình bày một phương pháp nhận diện các loại cá dựa trên mạng CNN VGG16 đã được huấn luyện trên tập dữ liệu ImageNet thông qua kỹ thuật học chuyển giao. Kết quả cho thấy mô hình được huấn luyện trên tập dữ liệu hình ảnh mới đạt kết quả nhận diện trong thực tế khá tốt, với tỷ lệ là 96,4% [6].

Nghiên cứu này trình bày một mô hình học tập chuyển giao dựa trên kiến trúc mô hình VGG16 đóng vai trò là bộ trích xuất đặc trưng, kết hợp với các bộ phân loại khác bao gồm: RBF Support Vector Machine (SVM), Logistic Regression (LR), Poly SVM, K-Nearest neighbors (KNN), and Neural Network (NN). Mô hình đạt độ chính xác cao nhất 89,83% trong thử nghiệm phân loại nhiều lớp với bộ phân loại RBF Support Vector Machine (SVM), cho thấy hiệu quả tốt hơn so với các mô hình máy học hiện tại [7].

Nghiên cứu đề xuất các kỹ thuật học chuyển giao (Transfer Learning) sử dụng mô hình VGG16 trong phân loại độ chín của các loại trái cây dựa trên màu sắc của trái [8]. Kết quả nghiên

cứ cho thấy rằng hiệu suất của mô hình học sâu sử dụng kỹ thuật học chuyển giao luôn tốt hơn, hiệu suất hơn so với việc sử dụng máy học với tính năng trích xuất truyền thống để xác định độ chín của trái cây.

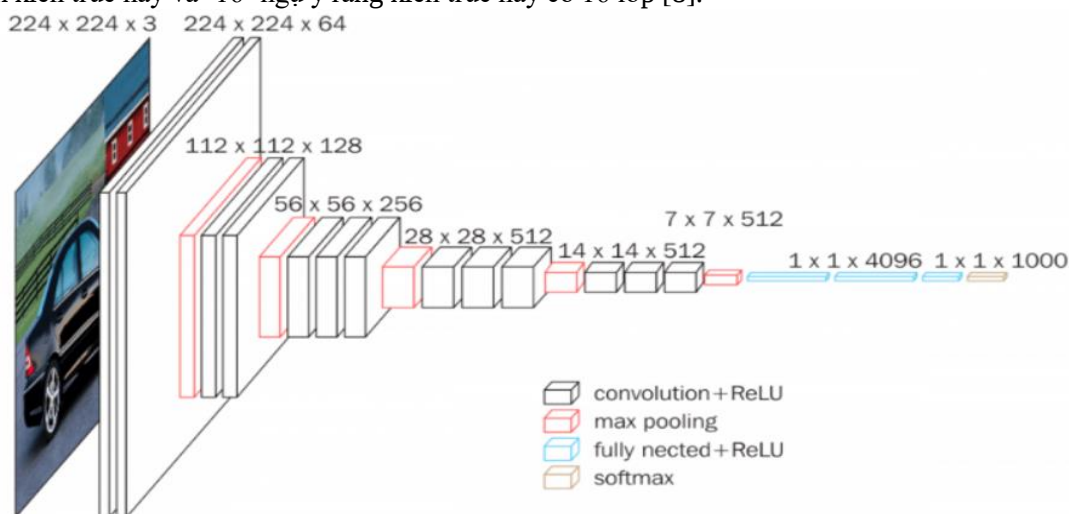
Nghiên cứu này trình bày phương pháp sử dụng mạng VGG16 trong bài toán phân loại lá nhãn, dữ liệu đầu vào là hình ảnh của ba loại nhãn được trồng phổ biến ở các nhà vườn Vĩnh Long đã được gán nhãn phân loại dữ liệu (data labeling). Môi trường huấn luyện và thử nghiệm mô hình là Google Colab [9]. Với hình ảnh đầu vào là lá nhãn, mô hình VGG16 được huấn luyện với độ chính xác tương đối cao, đồng thời dự đoán một cách hiệu quả được trên tập dữ liệu kiểm tra.

Trong các phần tiếp theo, bài báo trình bày các vấn đề liên quan đến mô hình VGG16, cách thu thập dữ liệu, huấn luyện mô hình cũng như trình bày kết quả đã đạt được và hướng phát triển nghiên cứu trong thời gian tiếp theo.

2. Phương pháp nghiên cứu

2.1. Mô hình học sâu VGG16

VGG16 là một kiến trúc mạng nơ-ron tích chập (CNN) (Hình 1) đơn giản và được sử dụng rộng rãi cho ImageNet [10], một dự án cơ sở dữ liệu trực quan lớn được sử dụng trong nghiên cứu phần mềm nhận dạng đối tượng trực quan. Kiến trúc VGG16 được phát triển và giới thiệu bởi Karen Simonyan và Andrew Zisserman từ Đại học Oxford [11], vào năm 2014, thông qua bài báo của họ “Các mạng kết hợp rất sâu để nhận dạng hình ảnh quy mô lớn”. Trong đó ‘VGG’ là tên viết tắt của Visual Geometry Group, là một nhóm các nhà nghiên cứu tại Đại học Oxford, những người đã phát triển kiến trúc này và ‘16’ ngụ ý rằng kiến trúc này có 16 lớp [8].



Hình 1. Kiến trúc mô hình VGG16

Hình 1 cho thấy mô hình VGG16 tiêu chuẩn nhận hình ảnh màu có kích thước [224x224]. Về kiến trúc thì VGG-16 vẫn giữ các đặc điểm của AlexNet nhưng có những cải tiến [11]:

Kiến trúc VGG-16 sâu hơn, bao gồm 13 lớp tích chập 2 chiều (thay vì 5 so với AlexNet) và 3 lớp fully connected.

Kiến trúc VGG-16 trình bày khái niệm về khối tích chập (block). Đây là những kiến trúc gồm một tập hợp các lớp CNN được lặp lại giống nhau. Kiến trúc khối đã khởi nguồn cho một dạng kiến trúc hình mẫu rất thường gặp ở các mạng CNN tiếp theo.

VGG-16 cũng kế thừa lại hàm kích hoạt ReLU ở AlexNet.

VGG-16 cũng là kiến trúc đầu tiên thay đổi thứ tự của các block khi xếp nhiều lớp CNN + max pooling thay vì xen kẽ chỉ một layer CNN + max pooling. Các lớp CNN sâu hơn có thể trích lọc đặc trưng tốt hơn so với chỉ 1 lớp CNN.

VGG-16 chỉ sử dụng các bộ lọc kích thước nhỏ [3x3] thay vì nhiều kích thước bộ lọc như AlexNet. Kích thước bộ lọc nhỏ sẽ giúp giảm số lượng tham số cho mô hình và mang lại hiệu quả tính toán hơn. VD: Nếu sử dụng 2 bộ lọc kích thước [3x3] trên một bản đồ đặc trưng (feature map) là output của một layer CNN có độ sâu là 3 thì ta sẽ cần $n_filters \times kernel_size \times kernel_size \times n_channels = 2 \times 3 \times 3 \times 3 = 54$ tham số. Nhưng nếu sử dụng 1 bộ lọc kích thước 5×5 sẽ cần $5 \times 5 \times 3 = 75$ tham số. 2 bộ lọc [3x3] vẫn mang lại hiệu quả hơn so với 1 bộ lọc [5x5].

Mạng VGG-16 sâu hơn so với AlexNet và số lượng tham số của nó lên tới 138 triệu tham số. Đây là một trong những mạng mà có số lượng tham số lớn nhất. Kết quả của nó hiện đang xếp thứ 2 trên bộ dữ liệu ImageNet validation ở thời điểm được công bố [11].

2.2. Thu thập dữ liệu cho huấn luyện mô hình

Tập dữ liệu lá nhãn được sử dụng bao gồm 2182 hình ảnh, các ảnh lá cây nhãn được chụp thủ công bằng camera của nhiều hãng điện thoại khác nhau như: samsung j7 prim, samsung A22 5G, samsung Note 20, Asus, Iphone 13. Lá nhãn nằm ở trung tâm ảnh, không bị mờ, mỗi loại chụp nhiều góc độ (như mặt trên, dưới của lá, lá đứng, lá ngang), theo nhiều độ sinh trưởng của lá (từ non đến già), mỗi loại lá nhãn để trên nền trắng, kích thước hình ảnh bao gồm [8000x6000] và [4160x3120] pixels. Dữ liệu sau khi thu thập sẽ tiếp tục xử lý gán nhãn dữ liệu bằng cách đưa hình mỗi loại lá cây nhãn vào một thư mục đại diện cho một loại lá nhãn (Bảng 1). Sau đó dữ liệu tiếp tục được chia làm 3 phần độc lập, không trùng lặp với nhau, trong đó bao gồm:

Tập dữ liệu huấn luyện (Training Dataset): 1954 hình ảnh, sử dụng trong quá trình huấn luyện mô hình, chiếm tỷ lệ 89,55%.

Tập dữ liệu kiểm định (Validation Dataset): 152 hình ảnh, sử dụng trong quá trình kiểm tra lại tính hiệu quả của mô hình qua từng vòng huấn luyện, chiếm tỷ lệ 6,97%.

Tập dữ liệu kiểm tra (Testing Dataset): 76 hình ảnh, sử dụng trong quá trình kiểm tra độ chính xác của mô hình sau khi trải qua quá trình huấn luyện, chiếm tỷ lệ 3,48%.

Bảng 1. Mẫu dữ liệu hình ảnh lá nhãn

Hình ảnh thực tế	Tên lá nhãn
	Ido
	Thạch Kiệt
	Xuồng com vàng

Chi tiết số lượng ảnh của từng loại được thể hiện trong Bảng 2, số lượng hình ảnh trong tập dữ liệu có số lượng tương đối cân bằng giữa các loại nhãn dữ liệu (label), tạo điều kiện thuận lợi cho việc huấn luyện mô hình.

Bảng 2. Chi tiết phân chia dữ liệu hình ảnh của các loại lá nhãn

Loại nhãn	Tập dữ liệu huấn luyện	Tập dữ liệu kiểm định	Tập dữ liệu kiểm tra
Ido	658	48	20
Thạch Kiệt	674	50	24
Xuồng com vàng	622	54	32

2.3. Huấn luyện mô hình

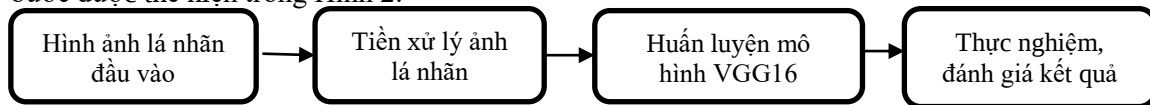
Mô hình VGG16 cần được huấn luyện lại với tập dữ liệu về lá nhãn đã được thu thập. Trước khi hình ảnh được vào mô hình cần phải qua bước tiền xử lý. Bước tiền xử lý đầu tiên là chuyển đổi kích thước hình ảnh (Resize) thuộc cả 3 tập dữ liệu đã phân chia về kích thước chuẩn của mô hình là [224x224].

Đối với tập dữ liệu huấn luyện ngoài áp dụng thay đổi kích thước hình ảnh, tập dữ liệu này còn được áp dụng thêm các phương pháp khác để tăng độ đa dạng cho hình ảnh huấn luyện, giúp mô hình được học tốt hơn, giúp gia tăng độ chính xác bao gồm:

RandomResizedCrop: Cắt kết hợp chuyển đổi kích thước ảnh một cách ngẫu nhiên.

RandomHorizontalFlip: Lật hình ảnh theo chiều dọc một cách ngẫu nhiên.

Sau khi hình ảnh đã được tiền xử lý, tiến hành đưa vào mô hình để huấn luyện, tóm tắt các bước được thể hiện trong Hình 2.



Hình 2. Sơ đồ huấn luyện mô hình VGG16

```

VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=3, bias=True)
  )
)
  
```

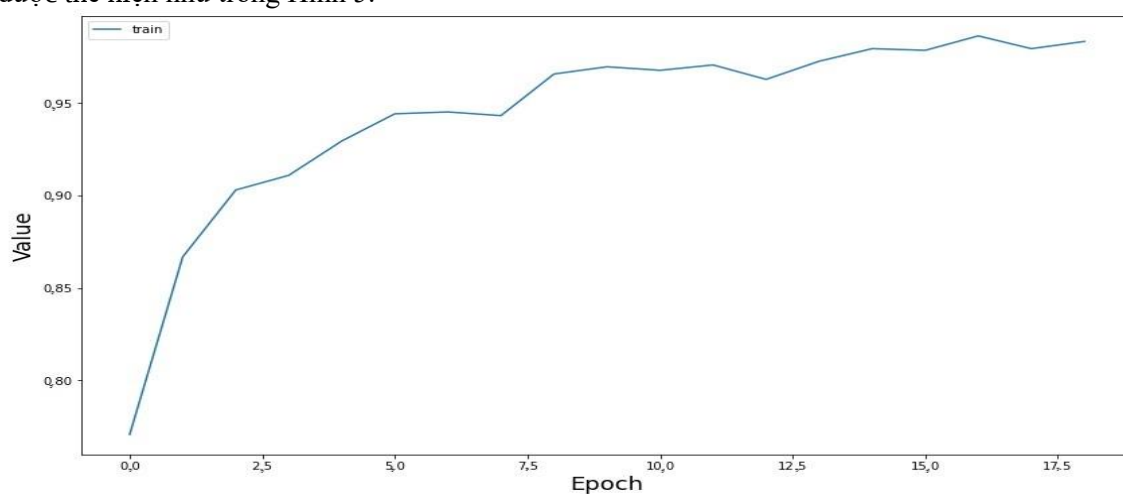
Hình 3. Chi tiết mô hình VGG16 dùng trong huấn luyện

Nghiên cứu sử dụng Google Colab để làm môi trường huấn luyện và kiểm tra đánh giá mô hình. Google Colab hỗ trợ việc lập trình với ngôn ngữ lập trình Python, hỗ trợ sử dụng các nền tảng học sâu phổ biến như Tensorflow, Keras, Pytorch. Ngoài ra, việc huấn luyện với sự hỗ trợ của GPU (Graphics Processing Unit) giúp quá trình này được thực hiện nhanh hơn nhờ khả năng tính toán song song.

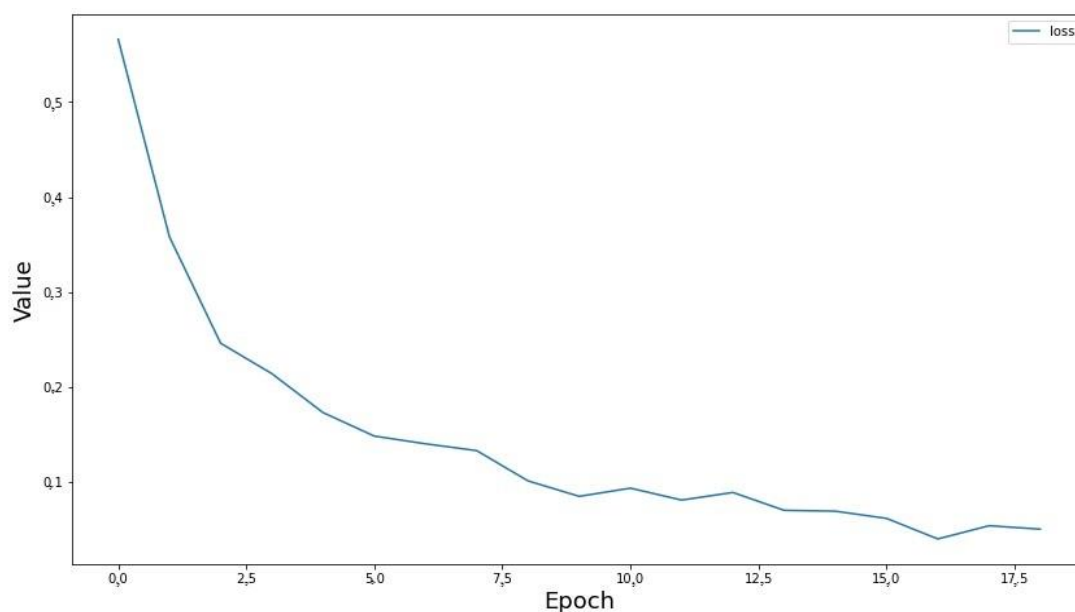
Một điểm quan trọng là nghiên cứu sử dụng mô hình VGG16 đã được huấn luyện sẵn với tập dữ liệu ImageNet với khả năng phân loại 1000 đối tượng khác nhau [12], vì vậy để phù hợp với số lượng đối tượng phân loại đang thực hiện là 3, thay đổi kích thước của lớp đầu ra từ ***Linear(in_features=4096, out_features=1000, bias=True)*** thành ***Linear(in_features=4096, out_features=3, bias=True)***. Chi tiết mô hình VGG16 sau khi đã thay đổi lớp cuối được xây dựng trên nền tảng Pytorch [13] lớp cuối (6) của thành phần (classifier) đã được thay đổi `out_features=3` như Hình 3.

3. Kết quả và bàn luận

Sau quá trình huấn luyện trên Google colab với tập dữ liệu huấn luyện là 1954 hình ảnh của lá nhãn được chia thành ba nhãn dữ liệu khác nhau như trong Bảng 2. Hình 4 cho thấy, mô hình VGG16 hội tụ và đạt độ chính xác khá cao trên tập dữ liệu huấn luyện được cài đặt thực hiện 20 vòng lặp huấn luyện (epoch). Đồng thời, độ lỗi của mô hình cũng giảm dần qua các vòng lặp được thể hiện như trong Hình 5.



Hình 4. Độ chính xác của mô hình sau 20 vòng lặp huấn luyện trên tập dữ liệu huấn luyện



Hình 5. Độ lỗi (loss) của mô hình sau 20 vòng lặp huấn luyện trên tập dữ liệu huấn luyện

Bảng 3. Kết quả độ chính xác trong quá trình huấn luyện qua từng vòng lặp

Vòng huấn luyện	Kết quả trên tập huấn luyện (%)	Kết quả trên tập kiểm định (%)
1	0,00	30,00
2	78,40	63,00
3	90,30	68,40
4	90,30	65,79
5	91,09	68,42
6	92,95	69,74
7	94,42	68,42
8	94,52	66,45
9	94,32	70,39
10	96,57	68,47
11	96,69	67,76
12	96,77	66,45
13	97,06	69,74
14	96,28	66,45
15	97,26	66,45
16	97,94	70,39
17	97,85	67,76
18	98,63	67,11
19	97,94	65,13
20	98,33	64,47

Từ Bảng 3 có thể thấy độ chính xác của mô hình trên tập huấn luyện được cải thiện qua từng vòng lặp, cho thấy việc lựa chọn mô hình VGG16 cho bài toán phân loại lá nhãn đạt hiệu quả tốt.

Bảng 4. Kết quả dự đoán của mô hình trên tập kiểm tra

	Ido	Thạch Kiệt	Xuồng
Thực tế	20	24	32
Dự đoán	19	24	32
Tỷ lệ (%)	95	100	100

Sử dụng mô hình đã huấn luyện xây dựng ứng dụng nhận dạng lá nhãn trong thực tế, đánh giá độ chính xác của mô hình bằng cách dự đoán nhãn của tập dữ liệu kiểm tra, kết quả được thể hiện trong Bảng 4, kết quả dự đoán trung bình đạt 98,30%.

Từ Bảng 4 nghiên cứu thể hiện kết quả trung bình của việc kiểm tra nhận dạng là 98,30%, kết quả này khá cao tuy có lệch so với kết quả tập huấn luyện nhưng không nhiều. Tuy nhiên dự đoán loại Ido còn sai 1 nhãn, việc nhận dạng sai này có thể bị ảnh hưởng của nhiều yếu tố như: thiết bị chụp, môi trường chụp, tư thế chụp,... từ đó dẫn đến sai sót 01 nhãn trên. Điều này khẳng định phương pháp nghiên cứu đã chọn mạng VGG16 với quá trình huấn luyện trên Google Colab kết hợp ngôn ngữ lập trình Python với framework chuyên dùng cho học sâu là Pytorch đạt hiệu quả trong bài toán phân loại lá nhãn.

4. Kết luận

Nghiên cứu đã xây dựng được ứng dụng áp dụng kỹ thuật học sâu trong bài toán phân loại ảnh, thông qua phương pháp học chuyển giao kết hợp với tinh chỉnh (fine-tuning) với mô hình VGG16 đã cho kết quả phân loại lá nhãn với độ chính xác cao 98,30%. Vì vậy, thông qua hình ảnh lá nhãn sử dụng công nghệ xử lý ảnh và kỹ thuật học sâu, nghiên cứu đã đáp ứng đủ yêu cầu về phân loại cây nhãn ngay từ giai đoạn cây con, cây giống.

Đề xuất nghiên cứu phát triển thêm ứng dụng trên thiết bị di động, giúp hỗ trợ tốt hơn cho việc phân loại lá nhãn trong thực tế. Xây dựng bộ dữ liệu huấn luyện lớn hơn, không chỉ dữ liệu của ba loại nhãn đã thực hiện trong nghiên cứu mà các loại nhãn phổ biến ở nước ta. Tiếp tục

phát triển nghiên cứu dựa vào hình thái, màu sắc của lá để xác định được thời điểm bón phân, xử lý ra hoa nhằm tăng hiệu quả, năng suất của việc trồng nhãn.

Lời cảm ơn

Nghiên cứu xin chân thành cảm ơn gia đình và cảm ơn đến các bác, các cô có vườn nhãn tại Cù Lao An Bình, xã An Bình, huyện Long Hồ, tỉnh Vĩnh Long đã hỗ trợ nghiên cứu trong quá trình thu thập dữ liệu.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] M. Vilasini and P. Ramamoorthy, "CNN approaches for classification of Indian leaf species using smartphones," *Comput. Mater. Contin.*, vol. 62, no. 3, pp. 1445–1472, 2020, doi: 10.32604/cmc.2020.08857.
- [2] J. Ahmad, K. Muhammad, I. Ahmad, W. Ahmad, M. L. Smith, L. N. Smith, D. K. Jain, H. Wang, and I. Mehmood, "Visual features based boosted classification of weeds for real-time selective herbicide sprayer systems," *Comput. Ind.*, vol. 98, pp. 23–33, 2018, doi: 10.1016/j.compind.2018.02.005.
- [3] H. A. Atabay, "Article a Convolutional Neural Network With a New," *Iioab J.*, vol. 7, no. October 2016, pp. 226–231, 2017.
- [4] N. Srisook, O. Tuntoolavest, P. Danphitsanuparn, V. Pattana-anake, and F. J. J. Joseph, "Convolutional Neural Network Based Nutrient Deficiency Classification in Leaves of *Elaeis guineensis* Jacq," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 14, pp. 19–27, 2022.
- [5] Y. Sun, Y. Liu, G. Wang, and H. Zhang, "Deep Learning for Plant Identification in Natural Environment," *Comput. Intell. Neurosci.*, vol. 2017, 2017, doi: 10.1155/2017/7361042.
- [6] P. Hridayami, I. K. G. D. Putra, and K. S. Wibawa, "Fish species recognition using VGG16 deep convolutional neural network," *J. Comput. Sci. Eng.*, vol. 13, no. 3, pp. 124–130, 2019, doi: 10.5626/JCSE.2019.13.3.124.
- [7] D. Albashish, R. Al-Sayyed, A. Abdullah, M. H. Ryalat, and N. Ahmad Almansour, "Deep CNN Model based on VGG16 for Breast Cancer Classification," in *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, 2021, pp. 805–810. doi: 10.1109/ICIT52682.2021.9491631.
- [8] J. Pardede, B. Sitohang, S. Akbar, and M. L. Khodra, "Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection," *Int. J. Intell. Syst. Appl.*, vol. 13, no. 2, pp. 52–61, 2021, doi: 10.5815/ijisa.2021.02.04.
- [9] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, E. Bisong, Ed. Berkeley, CA: Apress, 2019, pp. 59–64. doi: 10.1007/978-1-4842-4470-8_7.
- [10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
- [11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.
- [13] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala, "PyTorch Distributed: Experiences on Accelerating Data Parallel Training," *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 3005–3018, 2020, doi: 10.14778/3415478.3415530.