

Nhận diện cây trồng và cỏ dại

Lê Văn Trí

University of Information Technology
19521043
19521043@gm.uit.edu.vn

Đặng Hữu Nam

University of Information Technology
19520748
19520748@gm.uit.edu.vn

Phùng Thế Thiên

University of Information Technology
19520284
19520284@gm.uit.edu.vn

Tóm tắt nội dung—Nông nghiệp luôn đóng một vai trò quan trọng đối với an ninh lương thực trên toàn thế giới. Tuy nhiên, trồng trọt và chăm sóc cây trồng từ hạt giống đến khi thu hoạch luôn đối mặt với nhiều thách thức như tưới tiêu, nhiệt độ, chất lượng đất, cỏ dại, sâu hại,... Nhắc đến cỏ dại, biện pháp được nông dân ưu dùng trong thế kỉ qua đó là sử dụng thuốc diệt cỏ. Đây là phương pháp hiệu quả cao, tuy nhiên nó cũng để lại nhiều tác động xấu đến đất đai, nguồn nước và chính sản phẩm từ cây trồng được chăm sóc. Do đó, một thiết bị hay máy móc nào đó có khả năng diệt cỏ mà không gây những tác động xấu là cần thiết.

Index Terms—cây trồng, cỏ, cỏ dại

I. GIỚI THIỆU

Nông nghiệp luôn đóng vai trò là nền kinh tế chủ đạo của Việt Nam. Cho đến hiện nay thì Việt Nam vẫn là một nước nông nghiệp. Tuy nhiên, có nhiều bất cập, hạn chế như sâu bọ, thiên tai bão lũ, cỏ dại. Và ở đây, nhóm chúng tôi hướng đến vấn đề diệt cỏ dại.

Và vì thế giới ngày càng phát triển nên việc áp dụng công nghệ cao vào nông nghiệp cũng không còn xa lạ. Ý tưởng chúng tôi đưa ra dựa vào phát minh Autonomous Weeder của công ty Carbon Robotics. Là một robot nông nghiệp thông minh có tích hợp một máy tính trên bo mạch hỗ trợ bởi công nghệ trí tuệ nhân tạo (AI) giúp cỗ máy có khả năng tự nhận biết các loại cỏ dại và dùng tia laser diệt chúng một cách chính xác mà không làm ảnh hưởng tới cây trồng, đất, nước và các loại tài nguyên khác.

Đồ án này sẽ là bước đầu để thiết kế hệ thống máy móc có khả năng diệt cỏ mà không gây ô nhiễm môi trường với hiệu suất cao. Trong tương lai, nếu Việt Nam có thể sản xuất ra loại robot này thì đây sẽ là một bước tiến bộ vượt bậc trong nền nông nghiệp nước ta.

II. DATASET

A. Mô tả bài toán

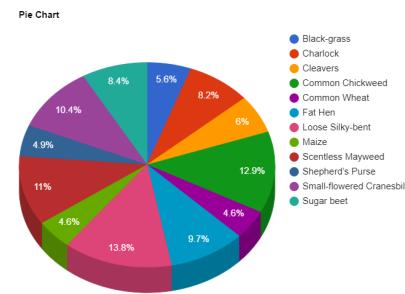
Trong phần này, chúng tôi sẽ tóm tắt về đồ án nhận biết cây trồng, cỏ dại của chúng tôi. Tác vụ này mục đích là để phân biệt các loại cây trồng, cỏ dại thông qua ảnh mầm của cây. Về mặt hình thức, tác vụ được mô tả như sau:

- **Input:** Một ảnh chụp mầm cây trồng / cỏ dại
- **Output:** 1 trong 12 nhãn cây trồng / cỏ dại

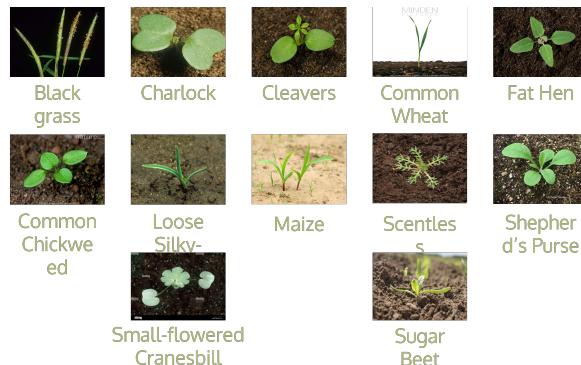
B. Dataset

Chúng tôi sử dụng bộ dữ liệu V2 Plant Seedlings Dataset [1] do Marsh và các cộng sự xây dựng và đã được đăng tải

công khai lên Kaggle. Mục đích của bộ dữ liệu này là để xây dựng mô hình phân loại giống cây trồng, cỏ dại và triển khai mô hình ứng dụng web cho những người nông dân có thể sử dụng.



Hình 1. Thống kê tổng quan về bộ dữ liệu V2 Plant Seedlings Dataset [1]



Hình 2. 12 nhãn cây trồng, cỏ dại

Bộ dữ liệu của chúng tôi bao gồm 5539 hình ảnh tương ứng với 12 nhãn cây trồng, cỏ dại. Mỗi lớp chứa các hình ảnh rgb cho thấy thực vật ở các giai đoạn phát triển khác nhau. Các hình ảnh có nhiều kích cỡ khác nhau và ở định dạng png. Các nhãn cây trồng này đại diện cho các loài thực vật phổ biến ở Đan Mạch:

Tên nhãn	Số lượng
Black-grass	309
Charlock	452
Cleavers	335
Common Chickweed	713
Common Wheat	253
Fat Hen	538
Loose Silky-bent	762
Maize	257
Scentless Mayweed	607
Shepherd's Purse	274
Small-flowered Cranesbill	576
Sugar Beet	463

Bảng I
NHÂN VÀ SỐ LƯỢNG TỔNG QUÁT

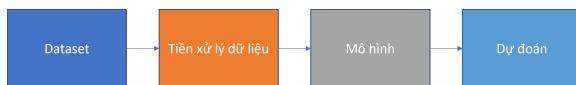


Hình 3. Một số hình ảnh của dataset

III. PHƯƠNG PHÁP TIẾP CẬN

A. Hệ thống đề xuất

Trong phần này, chúng tôi chỉ tập trung vào mô hình CNN và các mạng của CNN để tạo ra một mô hình hiệu suất tốt nhất bằng các kỹ thuật tinh chỉnh. Hình ảnh dưới đây cho thấy tổng quan về cách tiếp cận bài toán, đó là kỹ thuật tiền xử lý và phương pháp cốt lõi CNN, cùng với một số model mà chúng tôi sử dụng để huấn luyện.



Hình 4. Phương pháp tiếp cận của chúng tôi

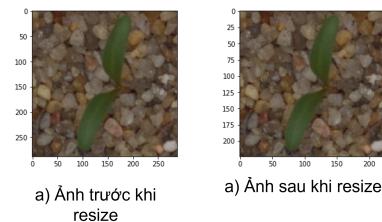
B. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là một bước quan trọng trong hầu hết các dự án Machine Learning hoặc Deep Learning. Qua quá trình tham khảo tập dữ liệu, chúng tôi nhận thấy kích thước hình ảnh trong dữ liệu không đồng đều. Do đó chúng tôi quyết định tiến hành điều chỉnh kích thước, đưa các hình ảnh về cùng một kích thước.



Hình 5. Bước tiền xử lý dữ liệu

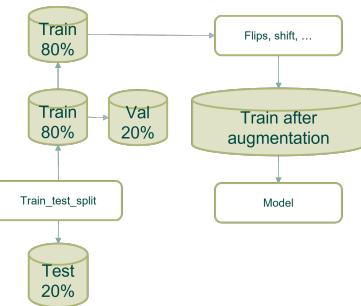
- Resize image:** Chúng tôi sẽ resize ảnh về cùng 1 kích thước là 224x224 (sử dụng tiền xử lý của thư viện tensorflow [5])



Hình 6. Ảnh trước và sau khi resize

C. Tăng cường dữ liệu

Trong Deep Learning, dữ liệu là một vấn đề đóng vai trò rất quan trọng. Vì vậy nên việc có ít dữ liệu cho việc huấn luyện mô hình thì rất khó để tạo ra kết quả tốt trong việc đưa ra dự đoán chính xác. Do nhận thấy tập dữ liệu của chúng tôi chưa đa dạng, nên chúng tôi quyết định sử dụng kỹ thuật tăng cường dữ liệu (data augmentation).



Hình 7. Tăng cường dữ liệu

- train_test_split:** Một hàm để chia mảng dữ liệu thành 2 tập con: dành cho dữ liệu huấn luyện và dữ liệu thử nghiệm.

Một số kỹ thuật tăng cường dữ liệu chúng tôi đã áp dụng thông qua trợ giúp của thư viện Tensorflow [5]:

- Flip:** hàm trong thư viện cho phép sinh thêm ảnh bằng cách lật theo chiều ngang.
- Shift:** Là một hàm trong thư viện cho phép chuyển vị trí sang các phần khác của ảnh.
- Rotation:** Là một hàm cho phép xoay ảnh theo một góc ngẫu nhiên trong phạm vi cho trước.
- Zoom:** Là một hàm cho phép thu phóng ảnh trong một phạm vi cho trước.
- Shear:** Là một hàm cho phép cắt xén góc bức ảnh.

D. CNN

Chúng tôi xây dựng mô hình bằng cách thêm lần lượt các layer và tinh chỉnh các thông số của chúng. So sánh kết quả của mỗi lần thêm layer và tinh chỉnh thông số. Sau đó, chọn ra mô hình có hiệu suất cao nhất.

IV. THỬ NGHIỆM VÀ THẢO LUẬN

A. Thông số đánh giá

- F1-Score:** Trước khi đi qua các kết quả thử nghiệm, trước tiên chúng ta thảo luận về các thước đo đánh giá được

sử dụng trong bài báo cáo này. Đầu tiên chúng tôi chọn sử dụng độ đo F1-Score để đánh giá hiệu suất của các mô hình:

$$1) \text{ accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$2) \text{ Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$3) \text{ Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$4) \text{ F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Vì F1-Score là trung bình điều hòa giữa Precision và Recall. Nên đối với những trường hợp mà precision và recall quá chênh lệch thì F1-Score sẽ cân bằng được cả hai độ lớn này và đưa ra đánh giá khách quan hơn.

2) Accuracy: Ngoài ra chúng tôi còn sử dụng độ đo chính xác accuracy. Tính tỉ lệ chính xác khi đoán đúng trên tổng số lần thử nghiệm.

B. Cài đặt thử nghiệm

Chúng tôi thử nghiệm trên tập dữ liệu đã cho với hai cách tiếp cận: Deep Learning và kết hợp Deep Learning với Machine Learning (Kết hợp CNN và SVM)

Như đã đề cập từ trước, quá trình cài đặt, huấn luyện và thử nghiệm chúng tôi triển khai thông qua thư viện Tensorflow [5] trên ngôn ngữ Python.

Trong quá trình cài đặt, do lượng dữ liệu huấn luyện không lớn và hạn chế của phần cứng chúng tôi có nên kỹ thuật transfer learning đã được sử dụng. Transfer learning là kỹ thuật giúp chuyển giao tri thức đã học được giữa các mô hình từ quá trình huấn luyện trước đó trên một bộ dữ liệu khác, từ đó giúp tăng hiệu quả và giảm chi phí huấn luyện. Ở đây, bộ trọng số của hai kiến trúc Resnet50 và Densenet121 đã huấn luyện từ trước trên tập dữ liệu ImageNet sẽ được sử dụng lại. Thông thường, khi sử dụng kỹ thuật này, toàn bộ các lớp nhận chuyển giao trọng số sẽ bị đóng băng, không được học trong quá trình huấn luyện. Tuy nhiên, chúng tôi nhận thấy điều đó khiến mô hình trở nên rất tệ. Do đó, chúng tôi quyết định, "rã băng" một số lượng lớp tích chập sau cùng trong mỗi mô hình, cho phép chúng học các đặc trưng cấp cao một cách tốt hơn, rút trích được từ các lớp tích chập trước đó.

C. Deep Learning

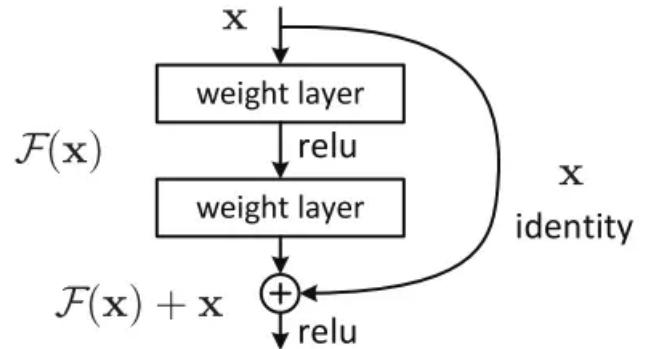
1) **CNN: (Convolutional Neural Network)** Mạng nơ-ron tích chập được sử dụng trong nhận dạng và xử lý hình ảnh với thiết kế đặc biệt để xử lý dữ liệu pixel.

2) **Resnet50 [3]:** Mạng ResNet (R) là một mạng CNN được thiết kế để làm việc với hàng trăm hoặc hàng nghìn lớp chập. Một vấn đề xảy ra khi xây dựng mạng CNN với nhiều lớp chập sẽ xảy ra hiện tượng Vanishing Gradient dẫn tới quá trình học tập không tốt.

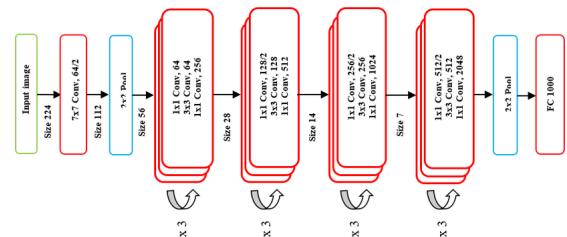
ResNet sử dụng kết nối "tắt" (Skip Connection) đồng nhất để xuyên qua một hay nhiều lớp. Một khối như vậy được gọi là một Residual Block, được miêu tả qua hình

Kiến trúc mạng Resnet50 [3]:

- Input image: Ảnh dữ liệu đầu vào



Hình 8. Resnet Skip Connection

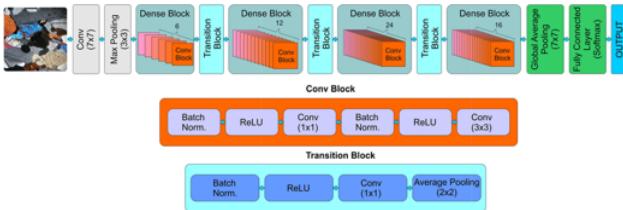


Hình 9. Kiến trúc mạng Resnet50

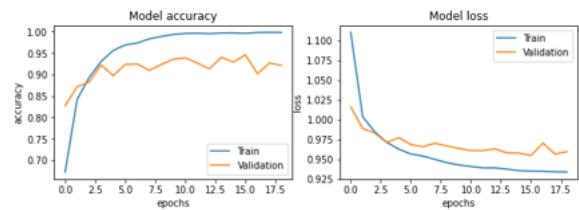
- Tích chập (Conv1) với 64 filter, sử dụng stride 2/2
- Max pooling (2, 2)
- x3 có nghĩa là có 3 khối ID block chồng lên nhau, sử dụng từ giai đoạn 2 đến giai đoạn 5
- Giai đoạn 2: Convolutional sử dụng 3 filter với size 64x64x256.
- Giai đoạn 3: Convolutional sử dụng 3 filter với size 128x128x512.
- Giai đoạn 4: Convolutional sử dụng 3 filter với size 256x256x1024.
- Giai đoạn 5: Convolutional sử dụng 3 filter với size 512x512x2048.
- Max pooling (2,2)
- FC1000: Fully Connected (Dense): Sử dụng softmax activation

3) **Densenet121 [4]:** (Dense connected convolutional network) là một trong những network mới nhất cho visual object recognition. Nó cũng gần giống Resnet nhưng có một vài điểm khác biệt. Densenet có cấu trúc gồm các dense block và các transition layers. Được stack dense block- transition layers- dense block- transition layers như hình 10.

- Dense Block: Ở mỗi Dense block sẽ có normalization, nonlinearity và dropout.
- Để giảm size và depth của feature thì transition layer được đặt giữa các dense block, nó gồm Conv kernel size =1, average pooling (2x2) với stride(2,2)
- Batch Norm: kỹ thuật để đào tạo mạng nơ ron sâu, chuẩn hóa các đầu vào thành một layer cho mỗi mini-batch. Để



Hình 10. Kiến trúc mạng Densenet121



Hình 11. Biểu đồ accuracy và loss của mô hình Densenet121 + SVM

ổn định quá trình học tập và giảm đáng kể số lượng epoch đào tạo cần thiết để đào tạo mạng sâu.

- ReLU(Rectified Linear Unit): một trong những hàm kích hoạt phổ biến nhất trong các mô hình máy học tập, được định nghĩa đơn giản như sau: $relu(x) = max(0, x)$. Nếu đầu vào nhỏ hơn hoặc bằng 0, thì relu sẽ trả về 0. Nếu đầu vào lớn hơn 0, thì relu sẽ chỉ trả về đầu vào đã cho.

D. CNN kết hợp SVM [2]

1) **SVM: (Support Vector Machine)** Đây là một thuật toán học máy được giám sát có thể được sử dụng cho cả các vấn đề phân loại hoặc hồi quy. Tuy nhiên, nó chủ yếu được sử dụng trong các bài toán phân loại.

2) **Kết hợp CNN và SVM [2]:** Chúng tôi đã quyết định thử kết hợp CNN và SVM để xem kết quả thử nghiệm sẽ như thế nào, và cuối cùng đã thu được kết quả khá tốt. Việc kết hợp thông thường được thực hiện bằng cách dùng một kiến trúc CNN như Resnet50 làm features extractor, sau đó dùng các đặc trưng rút trích được làm dữ liệu đầu vào cho mô hình SVM (thông qua thư viện scikit-learn). Tuy nhiên cách làm này có hạn chế là kiến trúc CNN phía trước sẽ không học được gì. Ở đây, chúng tôi đã sử dụng phương pháp khác, bằng cách thêm lớp Dense

E. Kết quả thử nghiệm

Thử nghiệm của chúng tôi trên mô hình Densenet121 [4] + SVM thu được kết quả tốt hơn so với những mô hình khác.

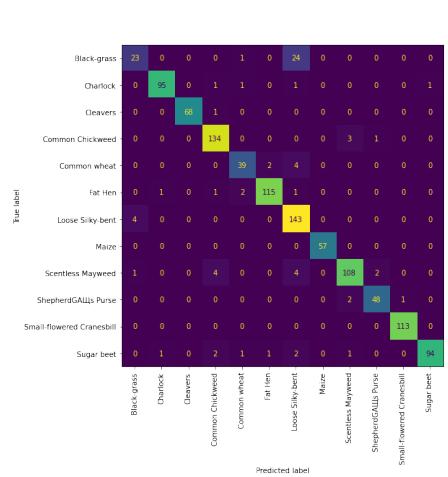
Model	Evaluation Metrics			
	Accuracy	Precision	Recall	F1-Score
Resnet50	92.78%	0.93	0.91	0.93
MobileNet	90.16%	0.89	0.88	0.90
Densenet121	89.71%	0.91	0.88	0.89
Densenet121 + SVM	93.59%	0.94	0.91	0.92

Bảng II
KẾT QUẢ THỬ NGHIỆM

F. Phân tích lỗi

Ở phần này, chúng tôi phân tích các lỗi mà chúng tôi thường gặp trong quá trình huấn luyện mô hình bằng phân tích định lượng sử dụng ma trận nhầm lẫn của mô hình hoạt động tốt nhất của chúng tôi.

Vẫn còn một số sai lầm do nhãn Blackgrass hay bị phân lớp nhầm qua Loose Silky-bent, nguyên nhân vì foreground nhỏ và background đa số là nền đất, sỏi, nên model có thể học nhầm background là đặc trưng chính dẫn đến nhầm lẫn.



Hình 12. Ma trận nhầm lẫn trong mô hình của chúng tôi

Tuy nhiên Blackgrass hay Loose Silky-bent đều là giống cỏ dại nên khi xây dựng mô hình áp dụng cho hệ thống máy diệt cỏ dại tự động chúng ta vẫn có thể chấp nhận sai sót này vì sau cùng hệ thống vẫn đạt được mục đích diệt cỏ dà đê ra.

Ngoài ra, sự chênh lệch kích thước của dữ liệu đầu vào sau khi resize ảnh có thể đã ảnh hưởng đến một phần tính chất ban đầu của ảnh, làm việc học trở nên sai sót.

Giải pháp mà chúng tôi đưa ra chính là xây dựng, thu thập bộ dữ liệu lớn hơn phù hợp với các mô hình đề xuất với độ phân giải cao, chụp từ nhiều góc độ khác nhau và trên các nền đất khác nhau để máy có thể học nhiều hơn các đặc trưng chính. Hoặc sử dụng các phương pháp object detection, segmentation... để tách mầm cỏ, cây trồng ra khỏi background trước khi làm đầu vào cho mô hình học máy.

V. KẾT LUẬT VÀ HƯỚNG PHÁT TRIỂN

Mô hình mà chúng tôi phát triển đã cho ra kết quả tốt hơn khi Accuracy đạt tới 93.59% và F1-Score đạt 0.92 bằng cách sử dụng thêm kỹ thuật tăng cường dữ liệu. Việc tăng cường dữ liệu giúp mô hình được học nhiều hơn, thay vì chỉ sử dụng một phần dữ liệu gốc để huấn luyện. Tuy nhiên, mô hình của chúng tôi vẫn còn nhiều hạn chế về việc tiền xử lý dữ liệu, mức độ đa dạng của dữ liệu huấn luyện. Nhưng nhìn chung, kết quả đạt được rất khả quan.

Trong tương lai, chúng tôi có kế hoạch sẽ sử dụng thêm các phương pháp tiền xử lý dữ liệu dành riêng cho ảnh mầm cây trồng, cỏ dại. Nhằm tối ưu hóa dữ liệu đầu vào của việc

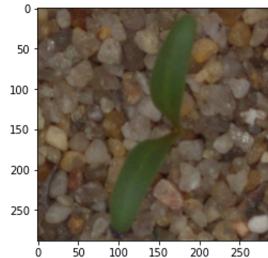


Black-grass

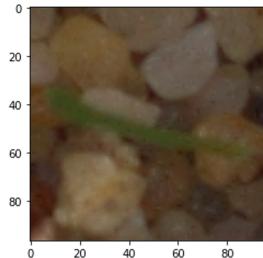


Loose Silky-bent

Hình 13. Black-grass và Loose Silky-bent



a) Size lớn



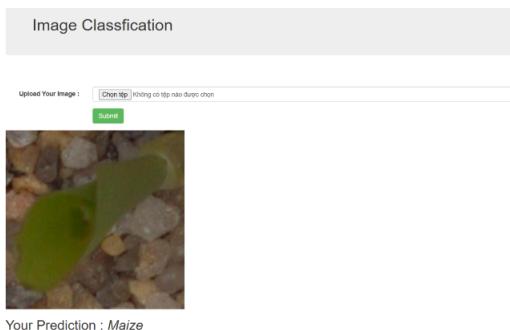
b) Size nhỏ

Hình 14. Chênh lệch size ban đầu

học mô hình CNN, góp phần dự đoán chính xác cây trồng, cỏ dại, là bước đầu, tiền đề xây dựng một robot diệt cỏ dại thông minh cho Việt Nam nhằm đưa nền nông nghiệp phát triển vượt bậc và những người nông dân đỡ vất vả hơn trong việc xử lý cỏ dại.

VI. DEMO

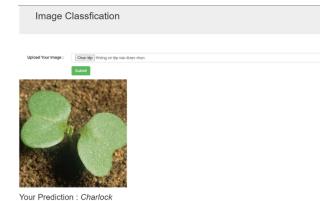
Chúng tôi đã xây dựng demo và đưa mô hình dự đoán, nhận biết cỏ dại và cây trồng lên localhost, dựa trên Flask [6] và Tensorflow [5], dưới đây là một số hình ảnh trong demo của chúng tôi:



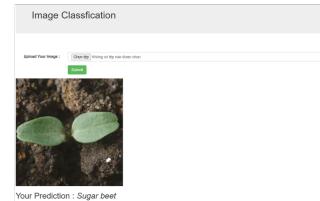
Hình 15. Dự đoán trong dataset

TÀI LIỆU

- [1] Marsh, V2 Plant Seedlings Dataset, 2018. <https://www.kaggle.com/vbookshelf/v2-plant-seedlings-dataset>



Dự đoán đúng



Dự đoán sai

Hình 16. Dự đoán hình ảnh trên google

- [2] Chúng tôi dựa trên bài báo Malaria Cell-Image Classification using InceptionV3 and SVM của Marada Amrutha Reddy, Ganti Sai Siva Rama Krishna và Teki Tanoj Kumar
- [3] Giới thiệu mạng Resnet, To Duc Thang, 2020
- [4] Quá trình phát triển của CNN từ LeNet đến DenseNet, dl_ap
- [5] Thư viện phần mềm mã nguồn mở: <https://www.tensorflow.org/>
- [6] là một micro web framework được viết bằng Python: <https://flask-palletsprojects.com/en/2.0.x/>