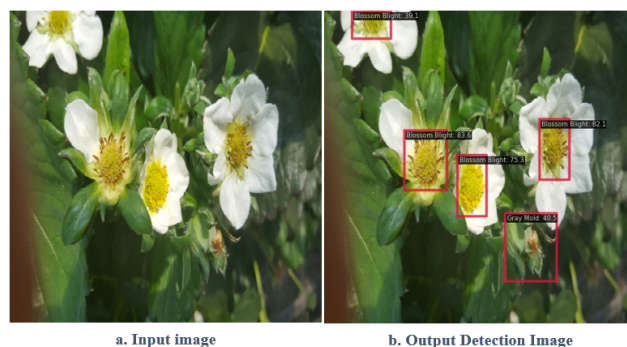


# <STRAWBERRY DISEASE DETECTION - RTMDet - PAPER01>

Trần Quang Anh Kiệt, Tăng Thanh Thiện,  
UIT-Together Research Group  
{22520725, 22521393}@gm.uit.edu.vn

**Tóm tắt**—Bài toán phát hiện sâu bệnh ở cây dâu tây trong ảnh là cơ sở cho các nghiên cứu đọc hiểu tài liệu, được cộng đồng nghiên cứu quan tâm. Trong bài báo này, chúng tôi đã tiến hành đánh giá thực nghiệm Real-Time Models for Object Detection (RTMDet) trên bộ dữ liệu Strawberry Disease Detection Dataset. Đây là một bộ dữ liệu lớn, chứa nhiều thách thức, bộ dữ liệu bao gồm 3243 ảnh và 3243 tệp chú thích phân đoạn tương ứng cho bảy loại bệnh khác nhau trên cây dâu tây. Cuối cùng chúng tôi đưa ra những nhận xét về phương pháp này được sử dụng và hướng phát triển. <source github>

**Từ khóa**—Phát hiện đối tượng, Real-Time Models for Object Detection, 7 loại sâu bệnh.



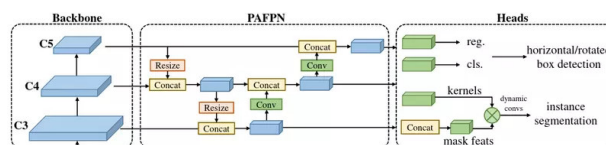
Hình 1. Bài toán phát hiện sâu bệnh trên cây dâu tây. Đầu vào (trái) là bức ảnh chụp bộ phận cây dâu tây. Đầu ra (phải) bao gồm các khung phát hiện vị trí của bệnh bạc hoa (đỏ) và tên bệnh.

## I. BÀI TOÁN

Cùng với sự phát triển của công nghệ thông tin, nhu cầu cho một ứng dụng phát hiện sâu bệnh gây hại ở cây trồng ngày một trở nên cần thiết. Trong bài toán này, chúng tôi sẽ lấy cây dâu tây làm mẫu cụ thể. Việc phát hiện sớm sâu bệnh ở cây dâu tây là rất quan trọng giúp con người có thể tìm ra phương pháp chăm sóc và loại bỏ sâu bệnh trên cây. Tuy nhiên, việc phát hiện sâu bệnh hại trên cây dâu tây một cách thủ công là một phương pháp tốn thời gian và chi phí thuê nhân công để phân loại từng cây, ngoài ra kết quả nhận được cũng không quá đáng tin cậy. Vì vậy, chúng tôi đã phát triển một mô hình phát hiện sâu bệnh hại trên cây dâu tây thông qua việc phân loại các ảnh chụp. Đương nhiên, để làm được việc đó chúng ta cần phải phát hiện các loại sâu bệnh trên cây dâu tây trong tài liệu dạng ảnh. Về vấn đề này, đầu vào cụ thể sẽ là ảnh chụp các bộ phận của cây dâu tây nghi ngờ có bệnh; đầu ra sẽ là vị trí nhiễm bệnh và tên loại bệnh (nếu có) trên cây dâu tây (Hình 1). Đây là một nhiệm vụ vô cùng quan trọng nhận được nhiều sự quan tâm với rất nhiều phương pháp đã được công bố [1] [2] [3] [4]. Tuy nhiên đó cũng là một thách thức lớn bởi ảnh có cấu trúc vô cùng đa dạng, biến hóa.

## II. REAL-TIME MODELS FOR OBJECT DETECTION (RTMDet)

### A. Giới thiệu phương pháp



Hình 2. Cấu trúc RTMDet [5].

RTMDet là một framework nhận diện đối tượng mạnh mẽ và tiên tiến. Nó sử dụng một kiến trúc tổng quát và cân bằng tính toán giữa các phần quan trọng của mô hình. Một đặc điểm nổi bật của RTMDet là việc sử dụng Depth-wise Convolution [6], một kỹ thuật tăng hiệu suất tính toán bằng cách giảm số lượng tham số. Điều này giúp tăng tốc độ và hiệu suất của mô hình.

RTMDet cũng có tính năng thân thiện với việc Quantization [7], cho phép nén mô hình và giảm đáng kể kích thước lưu trữ và tài nguyên tính toán yêu cầu. Điều này làm cho RTMDet trở thành một lựa chọn hợp lý cho các ứng dụng với tài nguyên hạn chế.

Ngoài ra, RTMDet sử dụng AdamW [8] làm optimizer, một biến thể của Adam với trọng số giúp quản lý tốt hơn quá trình học. Điều này cải thiện khả năng hội tụ và tăng cường hiệu suất của mô hình.

### B. Kiến trúc backbone

Kiến trúc backbone của RTMDet bao gồm một stem và bốn stage được đặt liên tiếp nhau. Stem là phần đầu tiên của mạng, trong khi stage sử dụng CSPLayer và Transition Layer để học các đặc trưng cần thiết và giảm chiều dữ liệu. Kiến trúc này bao gồm các khối cơ bản được gọi là khối chuyển đổi, được thiết kế để giảm kích thước của biểu đồ đầu vào và tăng độ sâu của mạng. Các khối chuyển đổi được kết nối với nhau bằng các đường dẫn con, cho phép mạng học các đặc trưng ở các cấp độ khác nhau của biểu đồ đầu vào.

Kiến trúc backbone của RTMDet đã được chứng minh là có hiệu quả trong việc phát hiện đối tượng trong thời gian thực, đồng thời đạt được độ chính xác cao. Kiến trúc này đã được sử dụng trong một số mô hình phát hiện đối tượng thời gian thực, bao gồm RTMDet-tiny, RTMDet-small và RTMDet-medium.

HRNet [9] là một mạng tích chập để giữ đặc trưng các đối tượng có độ phân giải cao với các trạng thái đi từ cao đến thấp.

Bắt đầu ở trạng thái thứ hai HRNet [9] sẽ tạo ra thêm một luồng song song với đầu vào là tích chập của các đầu ra của các luồng song song trước đó và cứ lặp lại ở các trạng thái tiếp theo. Với trạng thái thứ  $n$  sẽ có  $n$  luồng. Ở bước cuối cùng, các đầu ra có độ phân giải thấp sẽ được upsample lên độ phân giải cao hơn sau đó tạo thành một kim tự tháp với các độ phân giải đã được tạo ra.

## III. THỰC NGHIỆM

### A. Bộ dữ liệu

Chúng tôi sử dụng bộ dữ liệu có tên là Strawberry Disease Detection Dataset, được công bố năm 2021, bởi các thành viên của phòng thí nghiệm Trí tuệ nhân tạo, Khoa Khoa học và Kỹ thuật Máy tính, JBNU (Jeonbuk National University) [10]. Bộ dữ liệu bao gồm tổng cộng 6487 hình ảnh cùng các tệp chú thích phân đoạn tương ứng cho bảy loại bệnh khác nhau của cây dâu tây. Trong bộ dữ liệu này [10] (Hình 3), để phục vụ cho nhiệm vụ huấn luyện và kiểm thử trong quá trình huấn luyện bao gồm 2 thư mục train và valid với tổng cộng 3514 tài liệu bao gồm ảnh và các nhãn dạng JSON, ứng với một ảnh sẽ có một nhãn. Để phục vụ cho nhiệm vụ kiểm thử sau khi huấn luyện, bộ dữ liệu bao gồm 2 thư mục test và Test Disease Severity Level (được chia nhỏ thành 2 thư mục với độ khó khác nhau) với tổng cộng 1486 ảnh.

Ngoài ra, bộ dữ liệu còn có tài liệu Word ghi chép lại các nguồn thu thập ảnh. Để tải bộ dữ liệu vui lòng bấm vào [đường dẫn sau](#).

### Data Explorer

Version 1 (617.68 MB)

▼	Test Disease Severity Lev
▶	Level 1
▶	Level 2
▶	test
▶	train
▶	val
	Dataset.txt

### Summary

▼	6487 files	
	.jpg	3243
	.json	3243
	.txt	1

Hình 3. Bộ dữ liệu Strawberry Disease Detection [10].

### B. Cấu hình thực nghiệm

Thông số thực nghiệm:

```
_base_ = 'rtmdet_tiny_8xb32-300e_coco.py'
train_batch_size_per_gpu = 10
train_num_workers = 2
max_epochs = 10
stage2_num_epochs = 1
base_lr = 0.00008
```

Chúng tôi đã chạy thực nghiệm thành công phương pháp RTMDet trên bộ dữ liệu dâu tây với cấu hình máy như sau:

```
sys.platform: linux
Python: 3.10.6 (main, May 29 2023, 11:10:38) [GCC 11.3.0]
CUDA available: True
numpy_random_seed: 2147483648
GPU 0: Tesla T4
```

CUDA\_HOME: /usr/local/cuda  
 NVCC: Cuda compilation tools, release 11.8, V11.8.89  
 GCC: x86\_64-linux-gnu-gcc (Ubuntu 11.3.0-1ubuntu1 22.04.1) 11.3.0  
 PyTorch: 2.0.1+cu118  
 PyTorch compiling details: PyTorch built with:  
 - GCC 9.3  
 - C++ Version: 201703  
 - Intel(R) oneAPI Math Kernel Library Version 2022.2-Product Build 20220804 for Intel(R) 64 architecture applications  
 - Intel(R) MKL-DNN v2.7.3 (Git Hash 6dbeffbae1f23cbbcae17adb7b5b13f1f37c080e)  
 - OpenMP 201511 (a.k.a. OpenMP 4.5)  
 - LAPACK is enabled (usually provided by MKL)  
 - NNPACK is enabled  
 - CPU capability usage: AVX2  
 - CUDA Runtime 11.8  
 - NVCC architecture flags:  
 -gencode;arch=compute\_37,code=sm\_37;  
 -gencode;arch=compute\_50,code=sm\_50;  
 -gencode;arch=compute\_60,code=sm\_60;  
 -gencode;arch=compute\_70,code=sm\_70;  
 -gencode;arch=compute\_75,code=sm\_75;  
 -gencode;arch=compute\_80,code=sm\_80;  
 -gencode;arch=compute\_86,code=sm\_86;  
 -gencode;arch=compute\_90,code=sm\_90  
 - CuDNN 8.7  
 - Magma 2.6.1  
 TorchVision: 0.15.2+cu118  
 OpenCV: 4.7.0  
 MMEEngine: 0.8.2  
 MMDetection: 3.1.0+f78af77

### C. Quá trình thực nghiệm

Chúng tôi đã chạy thực nghiệm thành công phương pháp RTMDet trên bộ dữ liệu Strawberry Disease Detection trong môi trường Google Colab

### D. Bảng kết quả thực nghiệm

Model	AP	AP_50	AP_75
RTMDet	0.522	0.661	0.564

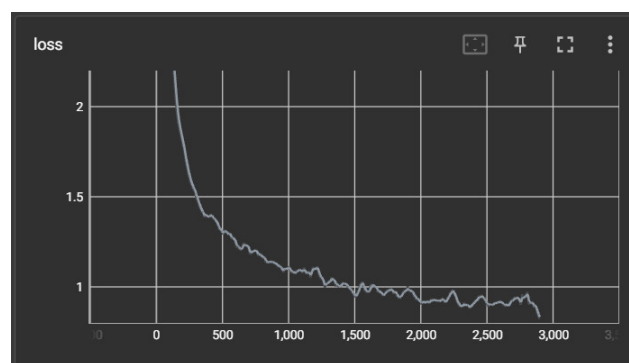
Bảng I

KẾT QUẢ PHƯƠNG PHÁP RTMDet TRÊN BỘ DỮ LIỆU DẦU TÂY.

Từ bảng kết quả trên, ta thấy phương pháp RTMDet chạy trên bộ dataset Strawberry Disease cho độ chính xác cao. Cụ thể như sau: AP là 0.522, AP\_50 là 0.661,

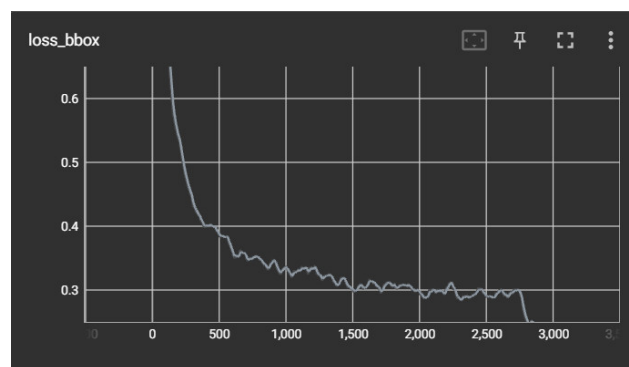
AP\_75 là 0.564. Từ Bảng II ta rút ra nhận xét như sau: độ đo của các lớp Angular Leafspot (0.756), Blossom Blight (0.905), Gray Mold (0.525), Leaf Spot (0.837), Powdery Mildew Leaf (0.735) có chỉ số khá cao nên độ nhận diện và phân loại của các lớp trên là khá tốt. Còn các lớp còn lại có độ đo: Anthracnose Fruit Rot (0.082), Powdery Mildew Fruit (0.235) có chỉ số thấp nên độ nhận diện và phân loại của các lớp thấp.

Đồ thị hàm loss (Hình 4) có trục hoành thể hiện số lần học và trục tung thể hiện độ sai lệch so với kết quả. Dựa vào đồ thị ta có thể thấy rằng số lần học càng nhiều thì độ sai lệch sẽ càng giảm dần.



Hình 4. Đồ thị hàm loss

Đồ thị hàm loss\_bbox (Hình 5) có trục hoành thể hiện số lần học và trục tung thể hiện sự sai khác giữa bbox với kết quả. Dựa vào đồ thị ta thấy số lần học càng nhiều thì tỉ lệ sai sót sẽ giảm xuống đáng kể.



Hình 5. Đồ thị hàm loss\_bbox

### E. Thảo luận

Từ kết quả đánh giá trên, ta thấy mô hình RTMDet cho một mức hiệu suất tổng quan AP (Average Precision) tương đối khá với giá trị trung bình 0.522 (với AP\_50

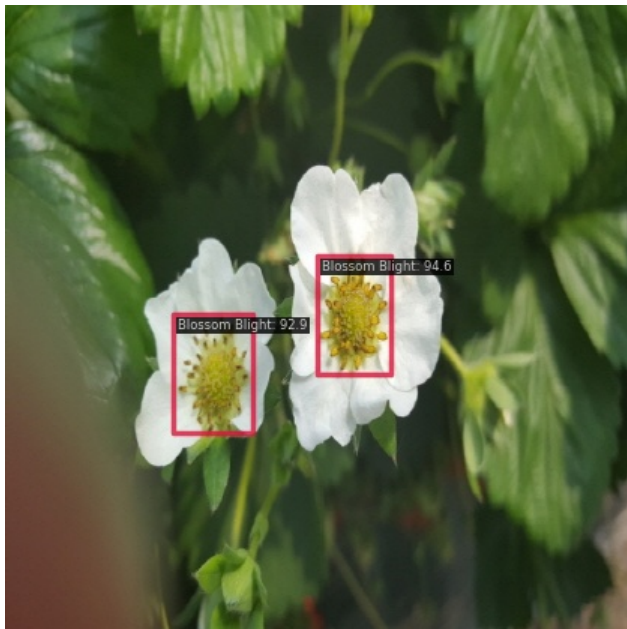
Model	Angular Leafspot	Anthrachnose Fruit Rot	Blossom Blight	Gray Mold	Leaf Spot	Powdery Mildew Fruit	Powdery Mildew Leaf	AP	AP_50	AP_75
RTMDet	0.756	0.082	0.905	0.525	0.837	0.235	0.735	0.522	0.661	0.564

Bảng II  
BẢNG KẾT QUẢ THỰC NGHIỆM TRÊN TỪNG LỚP.

là 0.661 và AP\_75 là 0.564), điều này cho thấy phương pháp này có khả năng phát hiện đối tượng ở mức độ chính xác trung bình cao. Tuy nhiên việc phân bố không đồng đều, có một số loại bệnh có thể được phát hiện với độ chính xác rất cao như Blossom Blight - với độ chính xác lên đến hơn 90% (Hình 6). Tuy nhiên lại có những bệnh gần như không thể phát hiện với độ chính xác chưa đến 10% (Hình 7) (Anthrachnose Fruit Rot). Từ đó, cần tiếp tục nghiên cứu và cải tiến mô hình nhằm mục tiêu tăng độ chính xác đối với 2 loại bệnh (Anthrachnose Fruit Rot và Powdery Mildew Fruit) và cải thiện các bệnh khác.



Hình 7. Trường hợp không phát hiện được bệnh Anthracnose Fruit Rot.



Hình 6. Trường hợp phát hiện tốt bệnh Blossom Blight với độ chính xác hơn 90%.

#### IV. KẾT LUẬN

Mô hình đã phần nào đáp ứng được mục tiêu ban đầu đề ra là phát hiện 7 loại bệnh trên cây dâu tây. Tuy nhiên có một số loại bệnh gần như không thể phát hiện hoặc phát hiện sai, từ đó cần nghiên cứu thêm để cải thiện độ chính xác của mô hình.

#### GHI CHÚ

Bản báo cáo vắn tắt bài tập 01 của sinh viên nhóm  
**UIT-Together Research Group**

#### TÀI LIỆU THAM KHẢO

- [1] L. Ma, X. Guo, S. Zhao, D. Yin, Y. Fu, P. Duan, B. Wang, and L. Zhang, "Algorithm of strawberry disease recognition based on deep convolutional neural network," *Complexity*, vol. 2021, pp. 1–10, 2021.
- [2] S. Lee, A. S. Arora, and C. M. Yun, "Detecting strawberry diseases and pest infections in the very early stage with an ensemble deep-learning model," *Frontiers in Plant Science*, vol. 13, p. 991134, 2022.
- [3] Y. Sonata, "Deteksi dini penyakit pada daun stroberi berbasis pengolahan citra," *Journal of Agricultural and Biosystem Engineering Research*, vol. 1, no. 2, pp. 29–40, 2020.

- [4] U. Afzaal, B. Bhattarai, Y. R. Pandeya, and J. Lee, "An instance segmentation model for strawberry diseases based on mask r-cnn," *Sensors*, vol. 21, no. 19, p. 6565, 2021.
- [5] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang, and K. Chen, "Rtmdet: An empirical study of designing real-time object detectors," *arXiv:2212.07784*, 2022.
- [6] X. Yang, R. B. Bist, S. Subedi, and L. Chai, "A computer vision-based automatic system for egg grading and defect detection," *Animals*, vol. 13, no. 14, p. 2354, 2023.
- [7] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE transactions on information theory*, vol. 44, no. 6, pp. 2325–2383, 1998.
- [8] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," *arXiv preprint arXiv:1904.00962*, 2019.
- [9] J. Huang, Z. Zhu, and G. Huang, "Multi-stage hrnet: Multiple stage high-resolution network for human pose estimation," *arXiv preprint arXiv:1910.05901*, 2019.
- [10] U. Afzaal, B. Bhattarai, Y. R. Pandeya, and J. Lee, "An instance segmentation model for strawberry diseases based on mask r-cnn," *Sensors*, vol. 21, no. 19, p. 6565, 2021.