

Báo Cáo Thuyết Minh Olympiad in Artificial Intelligence

Nhóm: BFC

Phan Trọng Đài – HCMUS
Email: phantrongdaimath@gmail.com

Ngày 19 tháng 4 năm 2025

Mục lục

1	Cách chạy mô hình	2
1.1	Cấu trúc file hiện tại	2
1.2	Cách chạy chi tiết	2
2	Mô tả file main.py	2
3	Ý tưởng chính	3
3.1	Transform	3
3.2	Mixup class	3
3.3	Spdresnet	3
3.4	LRNet	3
4	Một số nhắn gửi tới Ban Tổ Chức	4

1 Cách chạy mô hình

1.1 Cấu trúc file hiện tại

```
data/  
output/  
main.py  
requirements.txt
```

1.2 Cách chạy chi tiết

1. Bỏ các ảnh của test và train vào thư mục `data/`.
2. Chỉnh lại seed trong file `main.py` (nếu cần).
3. Cài đặt các thư viện cần thiết:

```
pip install -r requirements.txt
```

4. Di chuyển vào thư mục chứa file `main.py` bằng terminal hoặc command prompt.
5. Chạy mô hình:

```
python main.py
```

6. Kết quả cuối cùng nằm trong file `output/results.csv`.

2 Mô tả file `main.py`

File `main.py` được chia thành các phần chính sau:

1. **Global variables và Imports:** Thiết lập các biến toàn cục và import các thư viện cần thiết.
2. **Split CV:** Chia dữ liệu theo Cross-Validation. Sau khi chạy xong sẽ có 1 file tên là `split_cv` (Lưu ý: đã set seed lại theo BTC).
3. **Dataset:** Định nghĩa lớp Dataset cho việc tải dữ liệu.
4. **Transforms:** Các hàm biến đổi ảnh để hỗ trợ Augmentation trong quá trình training.
5. **Utils:** Các hàm tiện ích phụ trợ.
6. **PolyLoss:** Cài đặt lại hàm loss PolyLoss.
7. **Models:** Tập hợp các mô hình đã thử nghiệm (bao gồm cả các model không được dùng do không hiệu quả).
8. **Model utils:** Các hàm tiện ích liên quan đến mô hình.

9. **Training Functions:** Các hàm phục vụ cho quá trình training.
10. **Inference Functions:** Các hàm phục vụ cho quá trình dự đoán (inference).
11. **Main Execution:** Phần chính, thực hiện việc training và inference.
12. **Final Result Aggregation:** Lấy trung bình kết quả từ các lần chạy (folds) và xuất ra kết quả cuối cùng.

3 Ý tưởng chính

3.1 Transform

Qua vài thử nghiệm, nhóm em khám phá ra rằng strong spatial augmentation (biến đổi không gian mạnh) sẽ khiến model khó học được các đặc trưng (feature) do ảnh có độ phân giải thấp (low resolution).

Kiểm tra tập train và test, em nhận ra rằng hầu hết cả 4 loại nấm được chụp trong các loại môi trường khác nhau và dưới các góc chụp khác nhau. Nhóm em nhận thấy có 2 loại spatial transform đóng góp lớn cho model là: *RandomCropAndZoom* và *MultiScaleTransform*.

- **RandomCropAndZoom:** Phóng to vào một vị trí ngẫu nhiên nào đó của bức ảnh (mô phỏng trường hợp góc chụp không bao phủ hết cây nấm).
- **MultiScaleTransform:** Phóng to hoặc thu nhỏ bức ảnh (mô phỏng góc quay rộng hơn hoặc hẹp hơn).

3.2 Mixup class

Đây là ý tưởng chính giúp model của nhóm em cải thiện hiệu năng đáng kể (ví dụ: từ 0.9 lên 0.92). Ý tưởng này lấy cảm hứng từ các kỹ thuật mosaic và mixup augmentation.

Bạn em chọn ra 4 tấm ảnh ngẫu nhiên từ 4 class (4 loại nấm), sau đó lấy trung bình (mean) của 4 tấm ảnh này để tạo ra một loại class mới được gọi là *mixup*. Sau đó, model được train với bộ dữ liệu đã bao gồm cả class mới này.

3.3 Spdresnet

Lấy cảm hứng từ SPD-Conv Building Block trong bài báo [1], em đã tự cài đặt lại một model tên là *SPDResnet* sử dụng block này và mang lại hiệu quả cao.

Sơ đồ minh họa có thể xem tại đây: SPDRESNET on mermaid.

3.4 LRNet

Mô hình thứ hai em sử dụng là *LRNet* từ bài báo [2].

Sơ đồ minh họa có thể xem tại đây: LRNET on Mermaid

Lưu ý: Trong mã nguồn (code), do có một số sai sót ban đầu, mô hình này đã được đặt tên là *rlnet*. Em đã giữ nguyên tên này trong code để tránh thay đổi cấu trúc quá nhiều.

4 Một số nhắn gửi tới Ban Tổ Chức

- **Thư viện:** Hiện tại em chỉ liệt kê các thư viện chính sử dụng thêm. Có thể em sẽ liệt kê thiếu các thư viện phụ thuộc nhỏ khác. Hi vọng sẽ được BTC liên hệ để khắc phục nếu có thiếu sót.
- **SEED:** Phần `split_cv` của em có sử dụng SEED và em đã set GLOBAL SEED. Ngoài ra, em có một hàm là `set_seed(SEED)` được gọi ở mỗi lần chạy model. Do trước đó em sử dụng một SEED mặc định nên có thể vài chỗ em đã ghi cứng (hardcode) SEED và quên đổi. Nếu có, mong Ban Tổ Chức liên hệ em để khắc phục.
- **Phần cứng:** Mặc dù em đã cấu hình để code có thể chạy được trên cả GPU và CPU, nhưng em nghĩ bài làm của em cần sử dụng GPU để tối ưu thời gian chạy (hiện tại em chạy trên Kaggle mất khoảng 5 giờ đồng hồ).
- **Chuyển đổi code:** File code của em được chuyển từ định dạng notebook (`.ipynb`) trên Kaggle về file `.py`. Do đó, có khả năng phát sinh lỗi trong quá trình training và inference. Mặc dù em đã cố gắng chạy thử trên máy cá nhân với số epochs nhỏ và lượng dữ liệu giới hạn, nhưng có thể vẫn còn các lỗi không lường trước được. Hi vọng Ban Tổ Chức sẽ liên hệ với em để khắc phục nếu điều này xảy ra.

Tài liệu

- [1] Lee, K., Narayanan, H., Le, T., Le, H., Pham, T., Nakashima, Y., ... & Rawat, A. S. (2022). *No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects*. arXiv preprint arXiv:2208.03641. <https://arxiv.org/abs/2208.03641>
- [2] Do, T. V., Nguyen, T. H., Tran, M. T., Nguyen, H. P., Nguyen, C. V., Tjiputra, E. (2022). *LR-Net: A Block-based Convolutional Neural Network for Low-Resolution Image Classification*. arXiv preprint arXiv:2207.09531. <https://arxiv.org/abs/2207.09531>