

TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI: PHÂN LOẠI NGUY CƠ MƯA CỤC BỘ
TỪ ẢNH VỆ TINH SỬ DỤNG MÔ HÌNH CNN

GVHD: ThS. Nguyễn Thị Thùy Trang

Nhóm: 15

Thành viên:

- | | |
|-------------------------|------------|
| 1. Nguyễn Hưng Khánh Tú | 2001230843 |
| 2. Huỳnh Tiến Trung | 2001231008 |
| 3. Nguyễn Duy Phát | 2001230644 |

Thành phố Hồ Chí Minh, tháng 12 năm 2025

BẢNG PHÂN CHIA CÔNG VIỆC

THÀNH VIÊN	NHIỆM VỤ	MỨC ĐỘ HOÀN THÀNH
Nguyễn Hưng Khánh Tú	Xử lý dữ liệu ảnh vệ tinh, tiền xử lý ảnh, Data Augmentation	100%
Huỳnh Tiến Trung	Xây dựng model ResNet34, thiết kế hàm Loss, huấn luyện model	100%
Nguyễn Duy Phát	Xây dựng chức năng dự đoán, đánh giá model, phát triển Web Application	100%

MỤC LỤC

DANH MỤC HÌNH ẢNH	ii
DANH MỤC BẢNG – BIỂU ĐỒ	iii
CHƯƠNG 1. GIỚI THIỆU.....	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu của đồ án	1
1.3 Phạm vi nghiên cứu.....	1
1.4 Cấu trúc báo cáo.....	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	3
2.1 Tổng quan Machine Learning & Deep Learning	3
2.2 Kiến trúc mô hình Resnet.....	3
2.3 Lý thuyết về Ảnh vệ tinh và Khí tượng (Nowcasting).....	4
2.4 Các kỹ thuật đánh giá và tối ưu.....	5
2.5 Mạng Nơ-ron Tích chập (CNN - Convolutional Neural Networks)	5
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	9
3.1 Thu thập và xử lý dữ liệu	9
3.2 Thiết kế mô hình / thuật toán	9
3.3 Thiết kế kiến trúc hệ thống	10
CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI	12
4.1 Môi trường phát triển	12
4.2 Các chức năng chính	12
4.2.1 Tiếp nhận và Xử lý Ảnh Đầu vào (Input Handling).....	12
4.2.2 Cơ chế Suy luận Mô hình.....	12
4.2.3 Trả về Kết quả.....	13
4.3 Hướng dẫn sử dụng	13

CHƯƠNG 5. KẾT QUẢ VÀ ĐÁNH GIÁ	15
5.1 Môi trường thử nghiệm	15
5.2 Kịch bản thử nghiệm.....	15
5.3 Kết quả	15
5.4 Đánh giá kết quả	19
KẾT LUẬN	22
TÀI LIỆU THAM KHẢO	23

DANH MỤC HÌNH ẢNH

<i>Hình 1.1: Giới hạn vùng dự báo</i>	<i>2</i>
<i>Hình 2.1: Chi tiết các giai đoạn xử lý.....</i>	<i>4</i>
<i>Hình 2.2: Mạng CNN được thiết kế chuyên biệt để xử lý dữ liệu hình ảnh và thị giác máy tính</i>	<i>5</i>
<i>Hình 2.3: Convolutional layer làm nhiệm vụ trích xuất các đặc trưng từ dữ liệu đầu vào</i>	<i>6</i>
<i>Hình 2.4: Fully connected layer tổng hợp tất cả các đặc trưng đã trích xuất và thực hiện nhiệm vụ phân loại hình ảnh</i>	<i>7</i>
<i>Hình 3.1: Các thành phần của chương trình.....</i>	<i>10</i>
<i>Hình 5.1: Confusion Matrix của mô hình có độ chính xác 82%</i>	<i>16</i>
<i>Hình 5.2: Biểu đồ so sánh các chỉ số Precision, Recall và F1-score cho từng lớp phân loại (Không mưa, Mưa vừa, Mưa to) cho mô hình có độ chính xác 82%</i>	<i>16</i>
<i>Hình 5.3: Confusion Matrix của mô hình có độ chính xác 76%</i>	<i>17</i>
<i>Hình 5.4: Biểu đồ so sánh các chỉ số Precision, Recall và F1-score cho từng lớp phân loại (Không mưa, Mưa vừa, Mưa to) cho mô hình có độ chính xác 76%</i>	<i>17</i>
<i>Hình 5.5: Giao diện của ứng dụng</i>	<i>18</i>
<i>Hình 5.6: Màn hình gửi Mail cảnh báo đến người dùng.....</i>	<i>19</i>
<i>Bảng 5.1: Bảng đánh giá kết quả kì vọng.....</i>	<i>19</i>

CHƯƠNG 1. GIỚI THIỆU

1.1 Đặt vấn đề

Các hiện tượng thời tiết cực đoan như mưa lớn cục bộ, giông lốc ngày càng gia tăng, gây thiệt hại nặng nề nhưng lại rất khó dự báo chính xác trong thời gian ngắn bằng các phương pháp truyền thống. Việc cảnh báo sớm các hiện tượng này là một thách thức lớn, đòi hỏi những phương pháp tiếp cận mới và hiệu quả hơn.

Trong bối cảnh đó, ảnh mây vệ tinh nổi lên như một nguồn dữ liệu quan trọng, cung cấp thông tin trực quan và liên tục về sự phát triển của các hệ thống mây. Cùng với sự phát triển của trí tuệ nhân tạo, kỹ thuật học sâu, đặc biệt là Mạng Nơ-ron Tích chập (CNN), đã mở ra hướng tiếp cận đột phá, cho phép tự động phân tích và nhận dạng các mẫu phức tạp từ hình ảnh. Vì vậy, đề tài “Phân loại nguy cơ mưa cục bộ từ ảnh vệ tinh sử dụng mô hình CNN” được chúng em nghiên cứu và thực hiện.

1.2 Mục tiêu của đề án

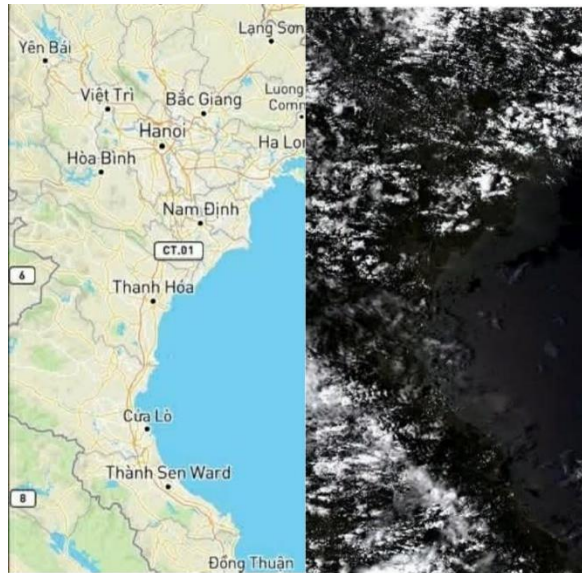
Mục tiêu của đề án là xây dựng một mô hình phân loại dựa trên CNN để tự động nhận dạng và đánh giá nguy cơ xảy ra mưa lớn/giông lốc trong vòng 1-2 giờ tới. Đề tài sẽ tập trung vào việc áp dụng các kỹ thuật xử lý ảnh vệ tinh và gán nhãn dữ liệu thực tế nhằm xây dựng một công cụ cảnh báo sớm, hiệu quả và mục tiêu là hạn chế bỏ sót lớp hiểm

1.3 Phạm vi nghiên cứu

Dữ liệu: Xây dựng và xử lý một tập dữ liệu nhỏ (khoảng 4000 ảnh) từ nguồn ảnh vệ tinh công khai của NOAA/NASA. Dữ liệu sẽ được gán nhãn thủ công thành ba lớp dựa trên việc đối chiếu với dữ liệu mưa lịch sử.

Mô hình: Ứng dụng các kiến trúc Mạng Nơ-ron Tích chập (CNN) cơ bản (như VGG, ResNet) để giải quyết bài toán đa lớp, xác định ảnh có nguy cơ gây mưa lớn, mưa vừa hay không.

Sản phẩm: Xây dựng một chương trình ứng dụng đơn giản có giao diện, cho phép người dùng tải lên ảnh vệ tinh và nhận lại kết quả phân loại là "Nguy cơ Mưa/Giông Lớn" hay "Nguy cơ mưa vừa, nhỏ" hoặc "Thời tiết ổn định".



Hình 1.1: Giới hạn vùng dự báo

1.4 Cấu trúc báo cáo

- + Cơ sở lý thuyết
- + Phân tích và thiết kế hệ thống
- + Cài đặt và triển khai
- + Kết quả và đánh giá
- + Kết luận

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Tổng quan Machine Learning & Deep Learning

Machine Learning: Là một nhánh của AI giúp máy tính học hỏi từ dữ liệu mà không cần lập trình rõ ràng cho từng tác vụ.

Học giám sát (Supervised Learning): Đây là phương pháp nhóm sử dụng. Máy học từ các cặp dữ liệu (ảnh vệ tinh) và nhãn (có mưa/không mưa).

Deep Learning: Là tập hợp con của Học máy, sử dụng các mạng nơ-ron nhiều lớp (Neural Networks) để mô phỏng cách não bộ xử lý thông tin.

Tại sao dùng Deep Learning cho ảnh vệ tinh? Vì ảnh vệ tinh có cấu trúc phức tạp, dữ liệu lớn, các thuật toán cổ điển khó trích xuất đặc trưng hiệu quả.

Bài toán Phân loại ảnh (Image Classification): Là quá trình gán một nhãn (label) cho một bức ảnh đầu vào.

Trong đề án này: Phân loại nhiều lớp (Multi-class Classification).

2.2 Kiến trúc mô hình Resnet

Ở bài nghiên cứu này chúng em dùng mô hình Resnet-34

ResNet-34 bao gồm tổng cộng 34 lớp có trọng số (learnable weights), bao gồm các lớp tích chập (Convolutional layers) và lớp kết nối đầy đủ (Fully Connected layer). Khác với các phiên bản sâu hơn như ResNet-50 hay ResNet-101 sử dụng khối "Bottleneck", ResNet-34 được xây dựng dựa trên chuỗi các Khối cơ bản (Basic Blocks).

Tên lớp	Kích thước đầu ra	Cấu trúc chi tiết (ResNet-34)
Conv1	112×112	$7 \times 7, 64, stride2$
Conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
Conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$
Conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$
Conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$
Output	1×1	Average Pool, FC, Softmax

Hình 2.1: Chi tiết các giai đoạn xử lý

Với tập dữ liệu ảnh vệ tinh tự gán nhãn có quy mô nhỏ, ResNet-34 là sự lựa chọn tối ưu hơn so với ResNet-50 hay ResNet-101 vì:

1. **Tránh Overfitting:** Số lượng tham số của ResNet-34 ít hơn (khoảng 21 triệu tham số so với 25 triệu của ResNet-50), giúp giảm nguy cơ mô hình "học vẹt" trên tập dữ liệu nhỏ.
2. **Hiệu quả tính toán:** Tốc độ huấn luyện và suy diễn (inference) nhanh hơn, phù hợp để triển khai trên các máy tính cá nhân không có GPU quá mạnh.

2.3 Lý thuyết về Ảnh vệ tinh và Khí tượng (Nowcasting)

Dự báo cực ngắn (Nowcasting): Là dự báo thời tiết trong phạm vi thời gian ngắn (0-6 giờ), dựa chủ yếu vào quan sát hiện tại.

Đặc điểm ảnh vệ tinh (Kênh RGB): Ảnh hồng ngoại đo nhiệt độ đỉnh mây.

Quy luật: Mây càng trắng sáng (hoặc đỏ/đen tùy bảng màu) → Nhiệt độ càng thấp → Đỉnh mây càng cao → Mây đối lưu mạnh → Nguy cơ mưa lớn/giông.

2.4 Các kỹ thuật đánh giá và tối ưu

Hàm mất mát (Loss Function): Focal Loss

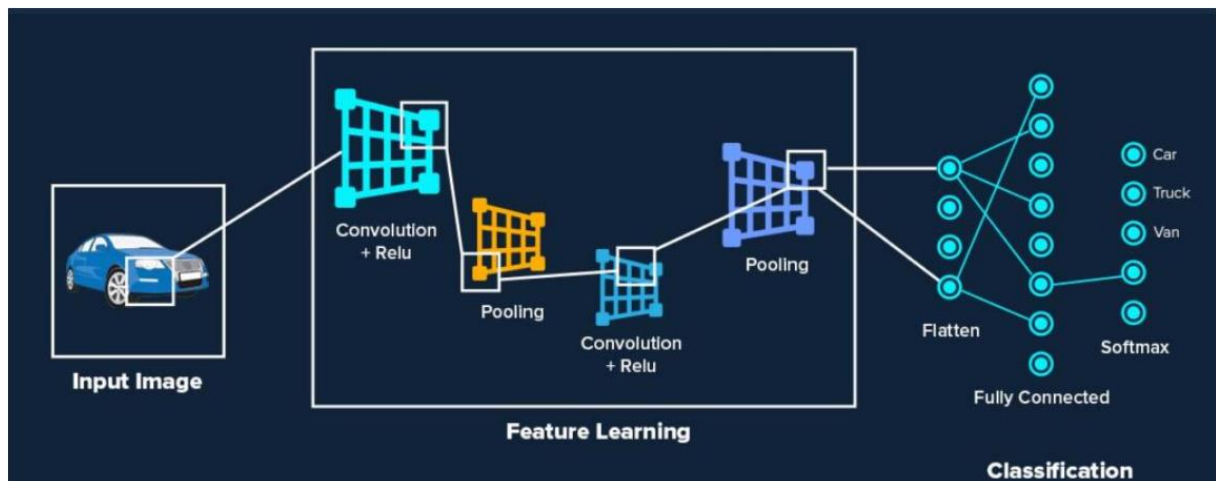
Thuật toán tối ưu (Optimizer): Adam

Các chỉ số đánh giá (Metrics):

- Accuracy: Tỷ lệ dự đoán đúng trên tổng số.
- Precision (Độ chính xác): Trong các lần báo mưa, bao nhiêu lần mưa thật? (Quan trọng để tránh báo động giả).
- Recall (Độ nhạy): Trong thực tế có mưa, mô hình phát hiện được bao nhiêu? (Quan trọng để không bỏ sót nguy hiểm).
- F1-score: F1-Score là trung bình điều hòa giữa precision và recall. Do đó nó đại diện hơn trong việc đánh giá độ chính xác trên đồng thời precision và recall.[1]

2.5 Mạng Nơ-ron Tích chập (CNN - Convolutional Neural Networks)

Convolutional Neural Networks (viết tắt là CNN) là một mô hình học sâu (Deep Learning) được thiết kế chuyên biệt để xử lý dữ liệu hình ảnh và thị giác máy tính. CNN hoạt động dựa trên nguyên lý của mạng nơ-ron truyền thống, nhưng điểm khác biệt chính là khả năng tự động trích xuất đặc trưng mà không cần sự can thiệp thủ công từ con người. Nhờ đó, CNN trở thành công cụ có khả năng nhận diện vật thể, phân loại hình ảnh và xử lý video rất hiệu quả, mạnh mẽ. [2]



Hình 2.2: Mạng CNN được thiết kế chuyên biệt để xử lý dữ liệu hình ảnh và thị giác máy tính

Các thành phần chính của CNN - Convolutional Neural Networks

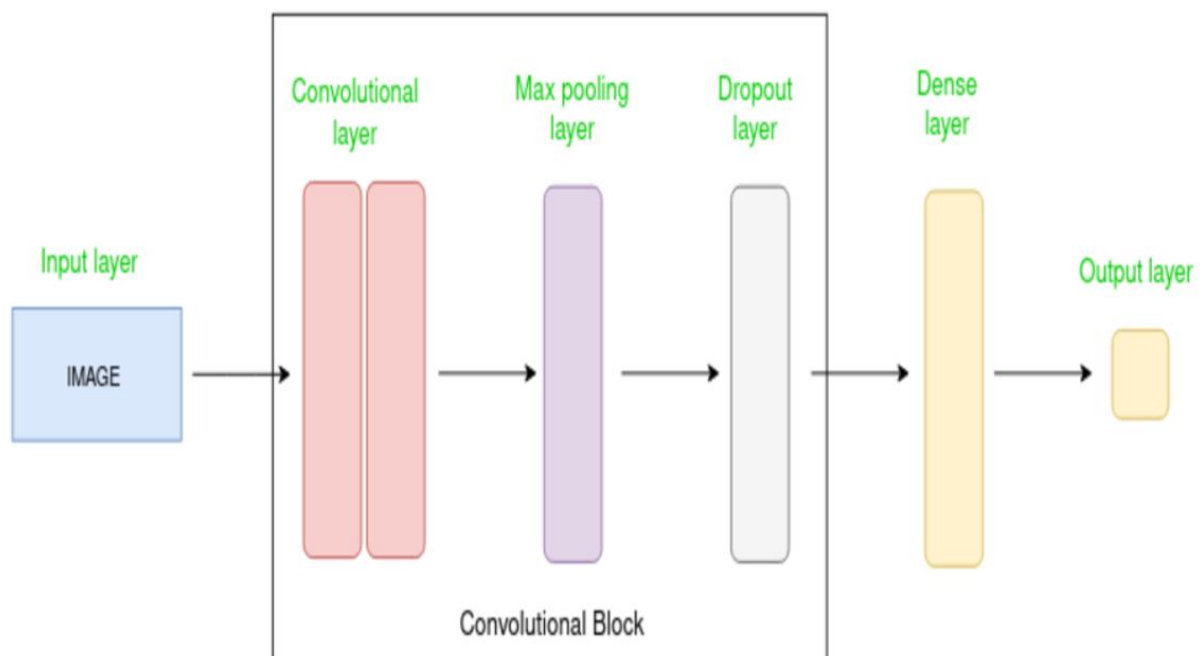
Convolutional Layer

Convolutional layer là lớp tích chập là thành phần quan trọng nhất của CNN, chịu trách nhiệm trích xuất các đặc trưng từ dữ liệu đầu vào. Lớp này sử dụng một bộ lọc (kernel) - một ma trận nhỏ có kích thước phổ biến như 3x3 hoặc 5x5 - quét qua từng vùng nhỏ của hình ảnh và thực hiện phép nhân tích chập (convolution) giữa các giá trị pixel với trọng số của bộ lọc. Kết quả của quá trình này tạo thành bản đồ đặc trưng (feature map), giúp mô hình phát hiện các đặc điểm như cạnh, góc, màu sắc hoặc kết cấu trong ảnh.

Các tham số quan trọng của lớp tích chập bao gồm: Số lượng bộ lọc, Stride (bước di chuyển của bộ lọc) và Padding (giữ kích thước ảnh). Trong đó:

- Stride xác định khoảng cách di chuyển của kernel trên ảnh đầu vào theo cả chiều ngang (trái sang phải) và chiều dọc (trên xuống dưới).
- Padding là quá trình thêm giá trị vào viền ảnh để kiểm soát kích thước feature map, bảo vệ thông tin viền ảnh khi thực hiện tích chập.

Sau mỗi phép tích chập, Convolutional Neural Networks thường áp dụng hàm kích hoạt ReLU (Rectified Linear Unit) để loại bỏ giá trị âm, tăng tính phi tuyến và giúp mô hình học hiệu quả hơn.



Hình 2.3: Convolutional layer làm nhiệm vụ trích xuất các đặc trưng từ dữ liệu đầu vào

Pooling layer

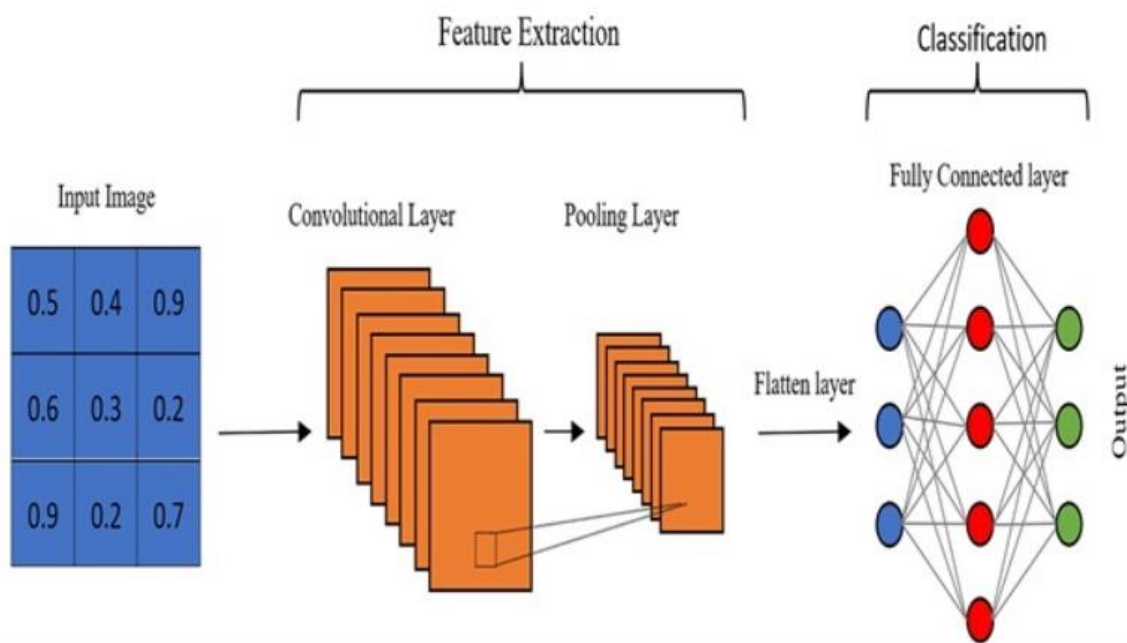
Sau khi trích xuất đặc trưng qua lớp tích chập, Convolutional Neural Networks sử dụng Pooling Layer để giảm kích thước feature map, từ đó giảm số lượng tham số, tăng hiệu suất tính toán và tránh hiện tượng overfitting (mô hình học quá kỹ vào dữ liệu huấn luyện, nhưng lại hoạt động kém khi gặp dữ liệu mới). Pooling hoạt động bằng cách áp dụng một bộ lọc nhỏ (thường là 2x2 hoặc 3x3) để lấy giá trị đại diện cho mỗi vùng quét, giúp giữ lại những thông tin quan trọng nhất.

Mặc dù pooling làm mất đi một số thông tin, nhưng đổi lại, nó giúp mô hình hoạt động hiệu quả hơn, giảm thiểu độ phức tạp và cải thiện khả năng tổng quát hóa đối với dữ liệu mới.

Fully connected layer

Fully connected layer là lớp kết nối đầy đủ nằm ở cuối mạng Convolutional Neural Networks, đóng vai trò tổng hợp tất cả các đặc trưng đã trích xuất và thực hiện nhiệm vụ phân loại hình ảnh. Ở lớp này, mỗi nơ-ron được kết nối với toàn bộ nơ-ron ở lớp trước, tạo nên một mạng lưới liên kết chặt chẽ. Các giá trị từ feature map trước đó sẽ được chuyển thành một vector một chiều, một chuỗi dài duy nhất và đưa vào lớp fully connected để xử lý. Quá trình này được gọi là làm phẳng **Flattening**.

Tiếp đó, CNN sử dụng các hàm kích hoạt phi tuyến như Softmax hoặc Sigmoid để tính toán xác suất cho từng lớp đầu ra. Điều này giúp cho mô hình đưa ra quyết định cuối cùng, chẳng hạn như phân loại hình ảnh thành các nhóm khác nhau.



Hình 2.4: Fully connected layer tổng hợp tất cả các đặc trưng đã trích xuất và thực hiện nhiệm vụ phân loại hình ảnh

Additional layers

Bên cạnh ba lớp chính, CNN có thể bao gồm một số lớp bổ sung để tối ưu hiệu suất và độ chính xác của mô hình.

- Lớp kích hoạt (Activation Layer) giúp tăng khả năng học và tạo tính phi tuyến cho mô hình bằng cách áp dụng các hàm phi tuyến như ReLU, Sigmoid hoặc Tanh.
- Lớp dropout (Dropout Layer) là một kỹ thuật quan trọng để giảm overfitting, bằng cách tạm thời loại bỏ ngẫu nhiên một số nơ-ron trong quá trình huấn luyện, giúp mô hình trở nên linh hoạt và tổng quát hơn.
- Ngoài ra, lớp chuẩn hóa (Batch Normalization Layer) cũng thường được sử dụng để tăng tốc độ huấn luyện và giúp mô hình ổn định hơn bằng cách chuẩn hóa dữ liệu giữa các lớp.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Thu thập và xử lý dữ liệu

Nguồn dữ liệu:

- Ảnh vệ tinh từ: NASA MODIS (Aqua/Terra).
- Dữ liệu có nhiều kênh: RGB.
- Ảnh định dạng: GeoTIFF (.tif).

Kích thước và tiền xử lý:

- Chuẩn hoá ảnh về 224x224 px để tương thích với kiến trúc ResNet34
- Loại bỏ giá trị lỗi (NaN \rightarrow 0).
- Chuẩn hóa: Áp dụng transforms.Normalize (sử dụng thông số ImageNet) để đưa ảnh vào cùng thang đo mà mô hình đã được huấn luyện.

Gán nhãn dữ liệu:

- Gán nhãn dữ liệu: Tập dữ liệu khoảng 1300 ảnh được gán nhãn tương ứng với dữ liệu mưa lịch sử.
- Phân loại:
 - Nguy cơ mưa / giông lớn (Cao)
 - Mưa vừa
 - Thời tiết ổn định (Thấp)

3.2 Thiết kế mô hình - thuật toán

Transfer learning:

Nhóm không huấn luyện mô hình từ đầu. Thay vào đó nhóm sử dụng mô hình đã được huấn luyện trên tập ImageNet với 1000 lớp. Bằng cách tận dụng tri thức đã học về nhận dạng cạnh, hình dạng, nhóm chỉ cần tinh chỉnh (fine-tune) mô hình. Điều này giúp dự án có tính khả thi cao và hội tụ nhanh chóng, mặc dù nhóm chỉ có khoảng hơn 1300 ảnh. Sau đó, tầng phân loại cuối cùng (Fully Connected Layer) được thay thế và tinh chỉnh lại để phù hợp với 3 lớp phân loại rủi ro mưa của bài toán.

Data augmentation:

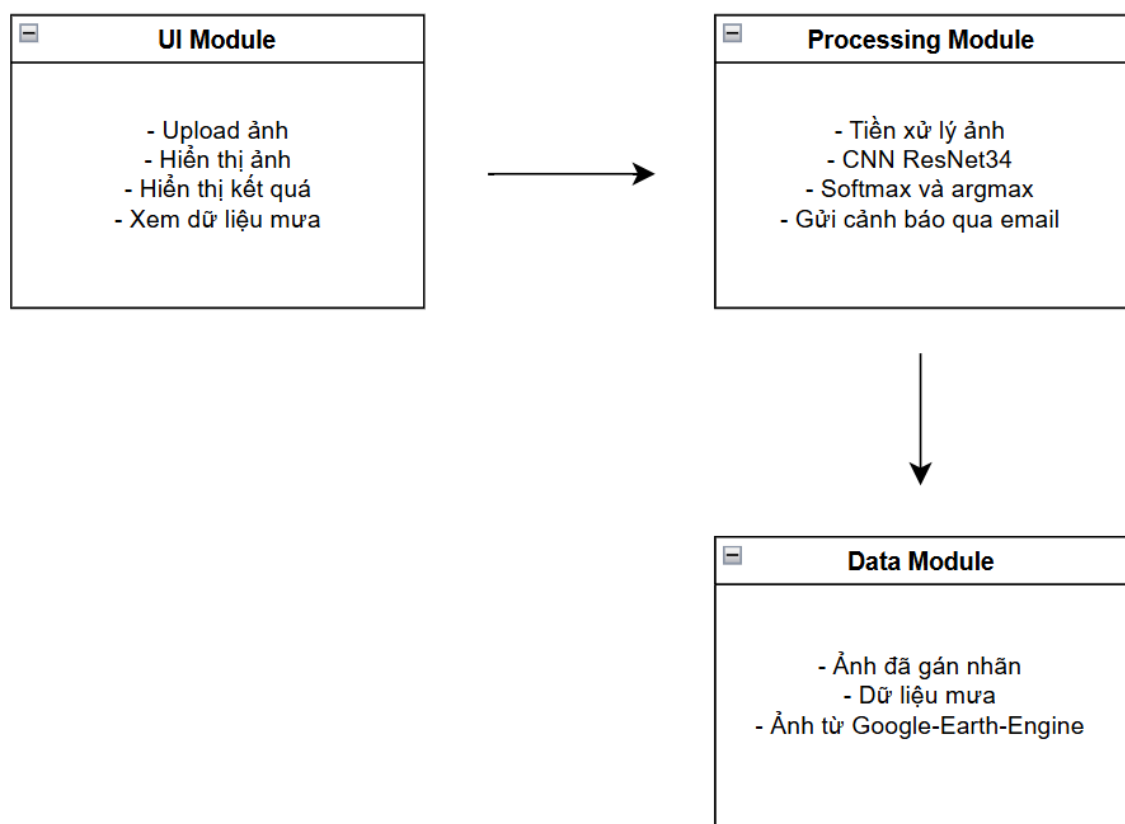
Vì dữ liệu ảnh của nhóm có số lượng hạn chế (khoảng 1300 ảnh), mô hình có nguy cơ cao sẽ "học thuộc lòng" (overfitting) ngay sau vài epoch đầu tiên. Để phòng chống rủi ro này và buộc mô hình phải tổng quát hóa tốt hơn, nhóm áp dụng chiến lược Data Augmentation (Tăng cường Dữ liệu). Các kỹ thuật như xoay ảnh (Rotation), lật

(Flipping), cắt ảnh ngẫu nhiên (Random Cropping) và xóa ngẫu nhiên một vùng (Random Erasing/Cutout) được sử dụng để tạo ra hàng ngàn phiên bản "biến thể" của các sự kiện mưa, giúp mô hình thấy được nhiều ngữ cảnh hơn và học được các đặc trưng thật sự có ý nghĩa thay vì chỉ ghi nhớ các pixel cụ thể.

Fine tune learning rate:

Nhóm điều chỉnh learning rate theo từng lớp của mô hình. Với những lớp đầu learning rate thấp nhằm giữ lại các tri thức giá trị đã được học từ ImageNet về nhận dạng cạnh và góc, tránh phá vỡ tri thức nền tảng. Lớp cuối cùng là lớp phân loại cần learning rate cao nhất vì đây là phần hoàn toàn mới, cần LR cao để các trọng số được cập nhật nhanh chóng và học cách ánh xạ các đặc trưng từ tầng trước vào 3 lớp phân loại mưa. Lớp giữa áp dụng learning rate trung bình để cho phép các lớp này tinh chỉnh các đặc trưng đã học, thích nghi với sự khác biệt giữa ảnh vật thể thông thường (ImageNet) và ảnh vệ tinh (MODIS).

3.3 Thiết kế kiến trúc hệ thống



Hình 3.1: Các thành phần của chương trình

Sơ đồ:

1. Người dùng tải ảnh vệ tinh từ giao diện.
2. Ảnh được chuyển đến module xử lý, thực hiện tiền xử lý và dự đoán.
3. Module xử lý truy cập đến module dữ liệu để tham chiếu dữ liệu mưa lịch sử, lưu ảnh gán nhãn.
4. Kết quả dự đoán trả về giao diện, nếu mưa lớn sẽ gửi cảnh báo (qua email).

CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI

4.1 Môi trường phát triển

Ngôn ngữ lập trình Python phiên bản 3.10.9.

Thư viện sử dụng là pytorch để huấn luyện mô hình, tifffile để xử lý ảnh tiff, flask để triển khai lên môi trường web, pandas để lọc dữ liệu từ file excel, google-earth-engine để lấy ảnh vệ tinh cùng với dữ liệu mưa theo toạ độ.

Phần cứng: GPU Nvidia (có hỗ trợ CUDA) để cho quá trình huấn luyện mô hình và tính toán song song.

4.2 Các chức năng chính

4.2.1 Tiếp nhận và Xử lý Ảnh Đầu vào (Input Handling)

Chức năng đầu tiên là đảm bảo hệ thống có thể nhận và chuyển đổi ảnh GeoTIFF (MODIS) của người dùng sang định dạng mà mô hình CNN đã được huấn luyện.

- Tải lên (Upload): Hệ thống sử dụng phương thức HTTP POST để nhận file ảnh (GeoTIFF/TIF) từ trình duyệt của người dùng.
- Xử lý GeoTIFF: Sử dụng thư viện tifffile hoặc rasterio để đọc trực tiếp dữ liệu pixel đa kênh từ tệp GeoTIFF (là định dạng ảnh vệ tinh của bạn).
- Tiền xử lý (Preprocessing):
 - + Loại bỏ Dữ liệu lỗi: Áp dụng np.nan_to_num để chuyển các giá trị 'No Data' (NaN, thường là màu đen) thành 0, đảm bảo tính ổn định.
 - + Đồng nhất Kích thước: Áp dụng transforms.Resize((224, 224)) để đưa ảnh về kích thước chuẩn mà ResNet34 yêu cầu.
 - + Chuẩn hóa (Normalization): Áp dụng transforms.Normalize (sử dụng thông số ImageNet) để đưa ảnh vào cùng thang đo (scale) mà mô hình đã được huấn luyện.

4.2.2 Cơ chế Suy luận Mô hình

Đây là chức năng cốt lõi. Sau khi ảnh được xử lý, nó được đưa qua kiến trúc Deep Learning đã được huấn luyện.

- Kiến trúc: Sử dụng mô hình ResNet34 (đã được fine-tune) làm xương sống (backbone) cho hệ thống.

- Chạy Dự đoán (Inference): Dữ liệu ảnh (tensor) được chuyển đến GPU (hoặc CPU) và được đưa qua mô hình ở chế độ torch.no_grad (để tiết kiệm tài nguyên và thời gian).
- Xác suất (Softmax): Đầu ra của mô hình được chuyển qua hàm Softmax để tính xác suất cho từng lớp (Not Rain, Medium Rain, Heavy Rain).
- Kết quả: Xác định lớp có xác suất cao nhất (dùng argmax) làm kết quả dự đoán cuối cùng.

4.2.3 Trả về Kết quả

Chức năng này đóng gói kết quả dự đoán và trình bày lại cho người dùng theo định dạng dễ hiểu.

- Gán Nhãn: Chuyển đổi chỉ số lớp số (0, 1, 2) thành tên lớp tương ứng (not_rain, medium_rain, heavy_rain).
- Hiển thị Ảnh: Ảnh được tải lên được chuyển đổi thành định dạng Base64 và hiển thị lại trên giao diện web, cho phép người dùng đối chiếu hình ảnh họ đã tải lên với kết quả dự báo.

4.3 Hướng dẫn sử dụng

Khởi động Ứng dụng (Local Host)

Trước tiên, bạn cần đảm bảo máy chủ (server) Flask đang chạy trên máy cục bộ của bạn.

- 1 Cài đặt Môi trường: Đảm bảo tất cả các thư viện (PyTorch, Flask, tiffle, v.v.) đã được cài đặt. Có thể sử dụng lệnh pip install -r requirement.txt để cài đặt nhanh chóng.
- 2 Chạy Server: Chạy file Python chính (chứa code Flask) để khởi động server.
- 3 Truy cập: Mở trình duyệt web của bạn và nhập địa chỉ sau:
 - + Giao diện Dự đoán: <http://127.0.0.1:5000/> (hoặc cổng mà Flask đang chạy).
 - + Giao diện Tra cứu: <http://127.0.0.1:5000/rainfall>

Dự đoán Mưa (Tương tác Chính)

Chức năng này cho phép người dùng dự đoán mức độ mưa dựa trên dữ liệu ảnh vệ tinh MODIS.

- 1 Đầu vào: Trên giao diện chính, chọn nút "Choose File".
- 2 Nhập Dữ liệu: Chọn tệp ảnh GeoTIFF có định dạng .tif hoặc .tiff mà bạn muốn mô hình phân tích.
- 3 Gửi Yêu cầu: Nhấn nút "Upload".
- 4 Kết quả trả về: Chương trình sẽ hiển thị:
 - + Dự đoán Lớp: Tên lớp dự đoán (not_rain, medium_rain, hoặc heavy_rain).
 - + Hiển thị Ảnh: Ảnh đã tải lên sẽ được hiển thị lại trên trang để xác nhận.

Tra cứu Dữ liệu Mưa Lịch sử

Chức năng này cho phép người dùng tương tác với dữ liệu mưa lịch sử

- 1 Truy cập: Mở địa chỉ: <http://127.0.0.1:5000/rainfall>
- 2 Nhập Yêu cầu: Giao diện sẽ hiển thị danh sách các năm có dữ liệu (ví dụ: 2021, 2022, ...).
- 3 Lựa chọn Năm: Chọn một năm cụ thể từ danh sách thả xuống (dropdown menu) và nhấn "Submit" (hoặc nút tìm kiếm).
- 4 Kết quả trả về: Hệ thống sẽ truy vấn và trả về bảng dữ liệu chi tiết về lượng mưa cho từng ngày trong năm đã chọn.

CHƯƠNG 5. KẾT QUẢ VÀ ĐÁNH GIÁ

5.1 Môi trường thử nghiệm

Phần cứng: GPU Nvidia có hỗ trợ CUDA nhằm huấn luyện.

Ngôn ngữ & Framework: Python, Pytorch.

Thư viện xử lý dữ liệu: pandas, tiffle, google-earth-engine.

5.2 Kịch bản thử nghiệm

Nhóm đã huấn luyện hai mô hình trên cùng một tập dữ liệu để chứng minh việc sử dụng hàm Focal loss hiệu quả hơn hàm thông thường.

Bộ dữ liệu: Tập ảnh test từ vệ tinh Aqua gắn máy cảm biến MODIS đã lọc bỏ ảnh kém chất lượng, được chia thành 3 lớp:

- + Kịch bản 1: Mô hình độ chính xác 81.5%

Mục tiêu: Cho thấy việc không dùng kỹ thuật xử lý mất cân bằng nhãn sẽ khiến mô hình không tập trung vào lớp hiếm (mưa lớn)

Hàm loss: Hàm Cross-Entropy Loss thông thường

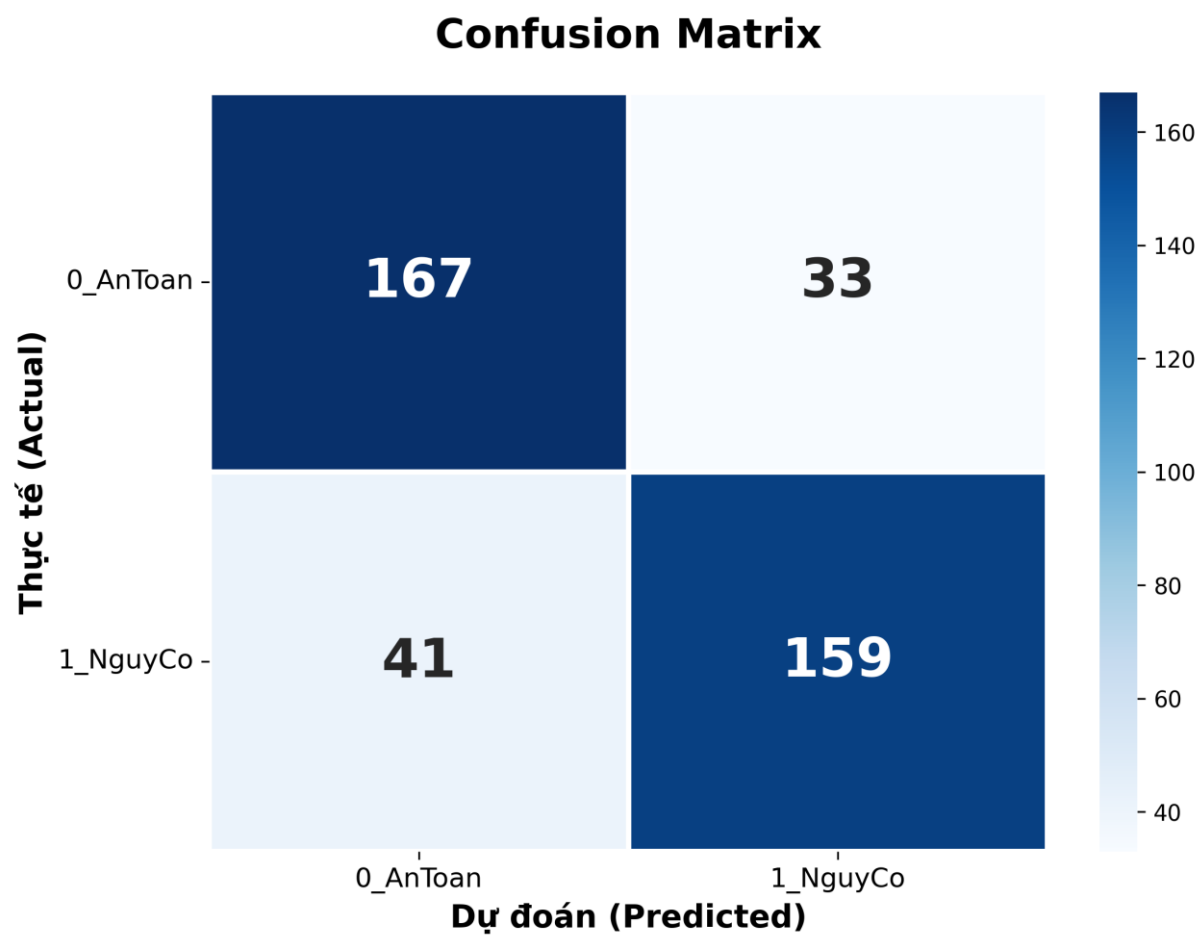
- + Kịch bản 2: Mô hình độ chính xác 76%

Mục tiêu: Cho thấy mô hình dự đoán đúng và ít bỏ sót lớp hiếm (mưa lớn)

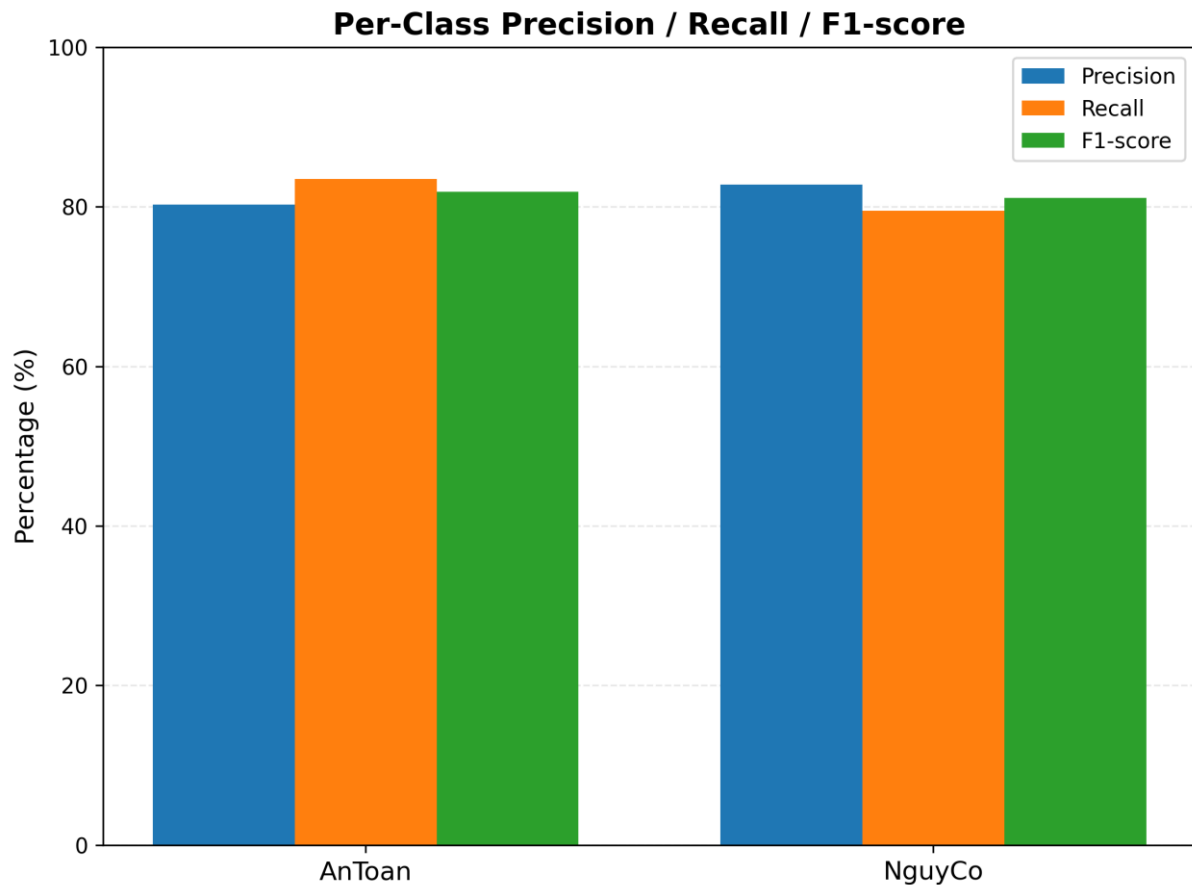
Hàm loss: Focal loss với α = trọng số của lớp với lớp hiếm được tăng trọng số và $\gamma = 2.5$

5.3 Kết quả

Mô hình độ chính xác 81.5%:



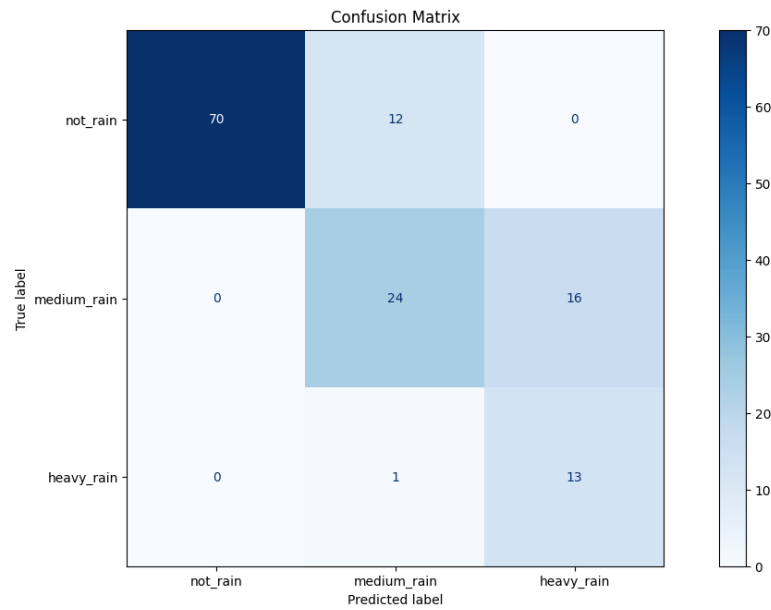
Hình 5.1: Confusion Matrix của mô hình có độ chính xác 82%



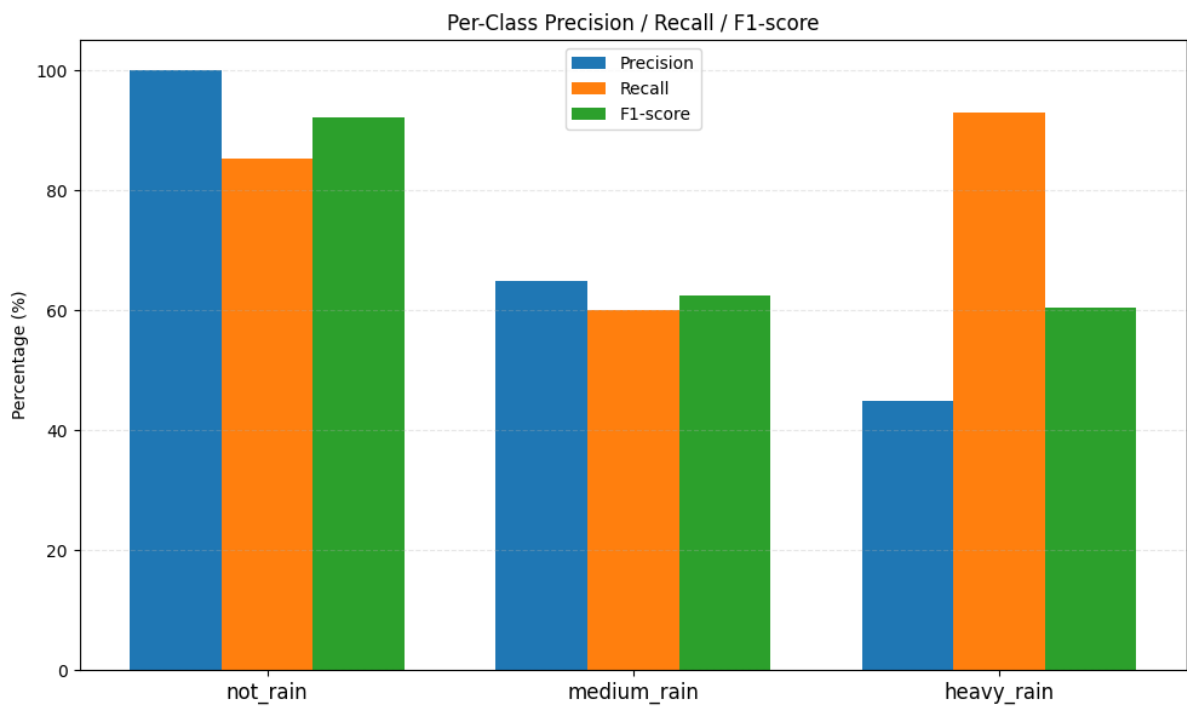
Hình 5.2: Biểu đồ so sánh các chỉ số Precision, Recall và F1-score cho từng lớp phân loại (Không mưa, Mưa vừa, Mưa to) cho mô hình có độ chính xác 81.5%

Tuy mô hình có chỉ số precision (độ chính xác) cao khi dự đoán ở cả ba lớp không mưa, mưa vừa và mưa lớn. Nhưng độ nhạy ở lớp mưa lớn lại chỉ đạt xấp xỉ 57% dù precision đạt tới 72%. Qua đó có thể thấy mô hình chỉ dự đoán lớp mưa lớn khi nó cực kỳ chắc chắn. Sự thận trọng quá mức này khiến Recall (Độ nhạy) cho lớp Mưa lớn chỉ đạt xấp xỉ 57%. Điều này có nghĩa là mô hình đã bỏ sót 6 trên 14 ảnh mưa lớn thực tế, một tỉ lệ bỏ sót lên tới 43%. Nếu áp dụng mô hình này vào thực tế, khả năng bỏ sót hơn 40% các cơn bão lớn sẽ gây ra hậu quả thảm họa và không thể chấp nhận được trong một ứng dụng dự báo an toàn.

Mô hình độ chính xác 76%:



Hình 5.3: Confusion Matrix của mô hình có độ chính xác 76%

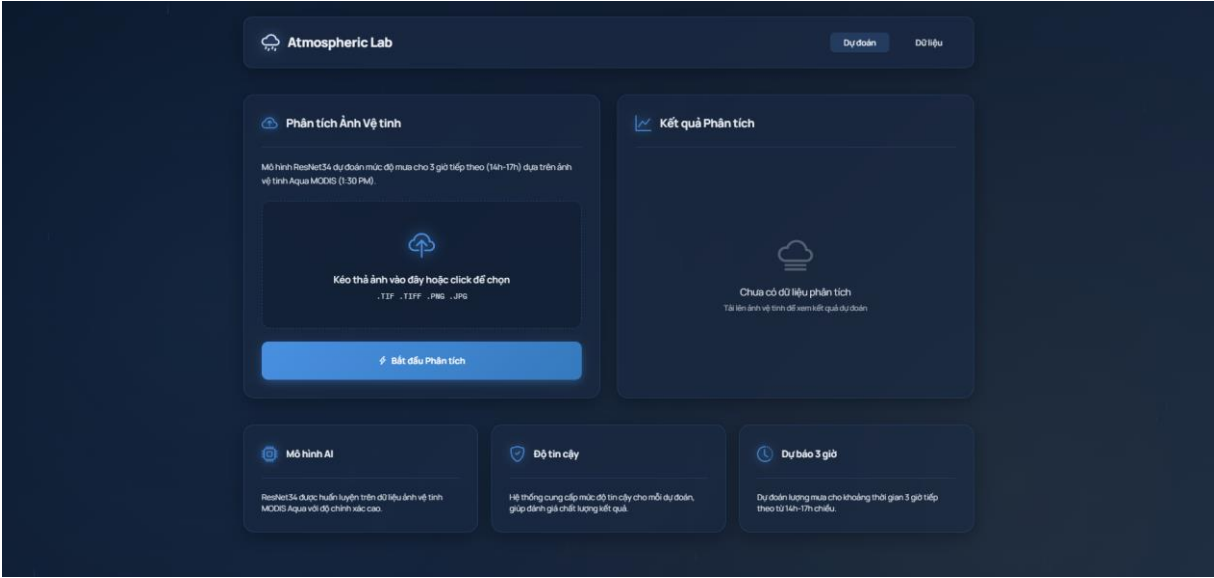


Hình 5.4: Biểu đồ so sánh các chỉ số Precision, Recall và F1-score cho từng lớp phân loại (Không mưa, Mưa vừa, Mưa to) cho mô hình có độ chính xác 76%

Ngược lại, mô hình 76% đã thực hiện thành công sự đánh đổi chiến lược: mặc dù Precision giảm xuống 50%, mô hình đã đẩy Recall cho lớp Mưa lớn lên mức 92.86%. Điều này chứng minh rằng việc sử dụng hàm Focal Loss đã thành công trong việc buộc

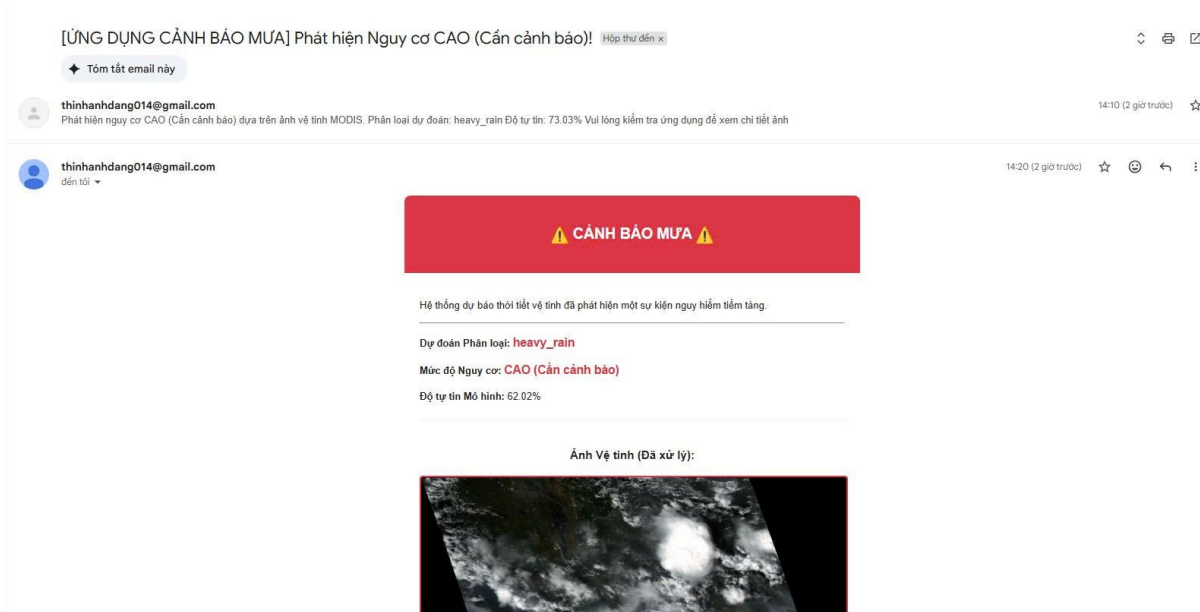
mô hình ưu tiên an toàn, giảm thiểu lỗi bỏ sót thảm họa xuống mức gần như bằng không. Việc hy sinh một phần nhỏ Precision là cái giá cần thiết và hoàn toàn hợp lý để bảo vệ mục tiêu an toàn chính của toàn bộ hệ thống dự báo.

Giao diện ứng dụng



Hình 5.5: Giao diện của ứng dụng

Nếu hình ảnh được dự báo là mưa lớn thì ứng dụng sẽ gửi Mail cảnh báo đến cho người dùng



Hình 5.6: Màn hình gửi Mail cảnh báo đến người dùng

5.4 Đánh giá kết quả

Đánh giá Kết quả Kỳ vọng (Achievement Assessment)

Đã đạt được "Kết quả kỳ vọng"

Nhóm khẳng định dự án đã đạt được và vượt mong đợi so với yêu cầu ban đầu

Yêu cầu kì vọng	Kết quả đạt được	Đánh giá
Giao diện tra về phân loại	Giao diện Flask trả về phân loại rủi ro (rất thấp, trung bình, cao)	Đạt chức năng hoạt động cơ bản
Độ chính xác nhận dạng	Recall (độ nhạy) cho lớp mưa lớn đạt 93%	Vượt mong đợi. Đã đạt được độ tin cậy an toàn gần như tuyệt đối (mục tiêu chính)

Bảng 5.1: Bảng đánh giá kết quả kì vọng

Phân tích Chuyển đổi và Thành công

Thành công này là kết quả trực tiếp của việc chuyển đổi từ bài toán **mất cân bằng** sang một **bài toán ưu tiên an toàn** thông qua Focal Loss:

- **Chỉ số an toàn (Recall):** Đã được tối ưu hóa. Khả năng bỏ sót một cơn mưa lớn (lỗi thảm họa) đã giảm từ 43% (Mô hình 82%: Recall xấp xỉ 57%) xuống chỉ còn 7%) (Mô hình Focal Loss: Recall đạt 93%).
- **Luận điểm Chứng minh:** Sự chênh lệch lớn này là bằng chứng không thể chối cãi cho thấy sử dụng Focal Loss là bắt buộc để mô hình vượt qua

sự áp đảo của lớp "Không mưa" (73% dữ liệu) và học được các đặc trưng quan trọng của Lớp Mưa lớn.

Phân Tích Các Kết Quả

Tại sao focal loss lại tốt hơn hàm mất mát thông thường?

Các hàm mất mát thông thường như Cross-Entropy Loss thì đánh giá các lớp ngang nhau, việc thưởng và phạt cho việc dự đoán các lớp đều như nhau. Điều này đã khiến mô hình trở nên lười biếng trong việc học lớp hiếm gặp như Mưa lớn mà chỉ tập trung vào lớp dễ. Còn hàm focal loss khiến mô hình tập trung vào lớp hiếm và bỏ qua lớp dễ. Với gamma giúp mô hình bỏ qua các mẫu Không mưa dễ đoán và chỉ tập trung vào mẫu khó. Sự "tập trung" này giúp mô hình không bị lệch hướng và đạt được khả năng phân loại tốt nhất ở Lớp Mưa lớn. Nhưng chính điều này làm mô hình nhút nhát hơn với các mẫu dễ và buộc nó phải dành toàn bộ tài nguyên cho mẫu khó, hiếm gặp như Mưa lớn. Cùng với đó, alpha giúp mô hình dũng cảm hơn trong việc đưa ra quyết định. Bằng cách gán trọng số cao hơn cho lớp Mưa lớn, alpha trực tiếp tăng cường mức độ nghiêm trọng của hình phạt đối với lỗi False Negative (bỏ sót mưa lớn). Chính alpha đã biến mô hình thành một mô hình dũng cảm, buộc nó phải hạ thấp ngưỡng tự tin dự đoán, thà báo động giả còn hơn là bỏ qua nguy cơ thực sự.

Giải thích sự đánh đổi

Nhóm chúng em chấp nhận Precision cho lớp Mưa lớn chỉ đạt xấp xỉ 50% (nghĩa là một nửa số lần mô hình báo động là báo động giả/phóng đại) để đổi lấy Recall đạt 93% (không bỏ sót thảm họa). Đây là một sự đánh đổi hợp lý và cần thiết vì chi phí của một lỗi False Negative (bỏ sót thảm họa) cao hơn nhiều lần so với chi phí của một lỗi False Positive (báo động giả).

Hạn chế và khó khăn gặp phải

Chất lượng Dữ liệu:

- **Thách thức:** Tập MODIS có hiện tượng "mất góc" (no data/pixel lỗi) do quỹ đạo vệ tinh.
- **Giải pháp:** Phải triển khai một bộ lọc phức tạp trong GEE để loại bỏ các ảnh có độ che phủ dữ liệu dưới 65% trước khi huấn luyện.

Khó khăn về Mất cân bằng:

- **Thách thức:** Số lượng mẫu Mura lớn ban đầu quá ít (40 mẫu), dẫn đến rủi ro Recall không ổn định trên tập Test (chỉ 4-7 mẫu).
- **Giải pháp:** Phải nới lỏng ngưỡng phân vị (từ Top 10% xuống Top 25%) để tăng số lượng mẫu.

CHƯƠNG 6. KẾT LUẬN

6.1 Tổng kết công việc và kết quả chính

Dự án này đã chứng minh tính khả thi của việc sử dụng Deep Learning để giải quyết bài toán dự báo mưa mất cân bằng và safety-critical (ưu tiên an toàn).

Thu thập và xử lý 2,101 cặp ảnh vệ tinh MODIS (Terra 10:30 + Aqua 13:30) từ vùng Bắc Trung Bộ, Việt Nam (2020-2024)

Gán nhãn tự động bằng dữ liệu mưa GPM với ngưỡng 12mm/h, thu được 1,349 mẫu nguy cơ thấp và 752 mẫu nguy cơ cao

Xây dựng mô hình dual-input CNN sử dụng ResNet50, tận dụng thông tin từ 2 thời điểm (10:30 và 13:30) để dự đoán mưa lúc 15:00-18:00

Chia dữ liệu theo tỷ lệ 70-15-15 cho train/validation/test với stratified sampling đảm bảo phân bố cân bằng

Đạt recall 100% trên tập test (bắt được toàn bộ trường hợp mưa lớn), phù hợp cho ứng dụng cảnh báo thiên tai

Nhưng vẫn còn những hạn chế:

- Độ chính xác tổng thể thấp (36.08%) do mô hình thiên về dự đoán nguy cơ cao
- Precision chỉ đạt 35.87%, dẫn đến nhiều cảnh báo giả (202/315 dự đoán High Risk là sai)
- Sử dụng single snapshot tại 2 thời điểm chưa đủ để nắm bắt đầy đủ diễn biến thời tiết
- Dữ liệu vẫn còn class imbalance (64% Low Risk vs 36% High Risk)
- Độ phân giải không gian MODIS (500m) và GPM (10km) không tương đồng, gây nhiễu trong quá trình gán nhãn
- Chỉ sử dụng 6 kênh phổ MODIS, thiếu thông tin về nhiệt độ đỉnh mây và độ ẩm khí quyển

6.2 Hướng phát triển trong tương lai

Nâng cấp kiến trúc mô hình

Đây là bước phát triển tiếp theo lý tưởng để thử thách khả năng của mô hình:

+ Mục tiêu: Vượt qua giới hạn của ResNet (CNN) trong việc tìm kiếm các mối quan hệ toàn cục (global dependencies) trong ảnh mây.

+ Hành động: Thu thập chuỗi ảnh liên tục (temporal sequences) mỗi 15-30 phút từ vệ tinh địa tĩnh (GOES, Himawari-8) lúc, có tiềm năng mang lại hiệu suất cao hơn ResNet trên các đặc trưng mây phức tạp. Và bổ sung thêm kênh IR (infrared) và Water Vapor để phát hiện mây đối lưu

Cải thiện mô hình

- + Thử nghiệm mô hình LSTM/GRU để xử lý temporal sequences
- + Điều chỉnh ngưỡng phân loại (threshold tuning) để cân bằng precision-recall
- + Áp dụng ensemble learning kết hợp nhiều models
- + Fine-tune focal loss parameters để xử lý class imbalance tốt hơn

Ứng dụng thực tiễn

+ Cải tiến: Chuyển đổi kết quả từ phân loại toàn vùng thành một bản đồ rủi ro cục bộ. Phát triển giao diện web cho phép upload ảnh vệ tinh và nhận kết quả dự đoán real-time

+ Hành động: Xây dựng module đầu ra để hiển thị Bản đồ Rủi ro (Risk Map) chi tiết, nơi các khu vực nhỏ (ví dụ: 10x10 km có nguy cơ cao được tô màu nổi bật trên bản đồ địa lý). Tích hợp với hệ thống cảnh báo thiên tai quốc gia, cũng như kết hợp với mô hình số trị thời tiết (NWP) để tăng độ tin cậy

TÀI LIỆU THAM KHẢO

- [1] P. Đ. Khánh, “Đánh giá mô hình phân loại trong ML,” *KhanhBlog*, 13 Aug. 2020. [Online]. Available: <https://phamdinhkhanh.github.io/2020/08/13/ModelMetric.html>. [Accessed: 22 Nov. 2025].
- [2] N. M. Hải, “Convolutional Neural Networks là gì? Khám phá cấu trúc của CNN,” *VNPT AI Blog*, 17 Mar. 2025. [Online]. Available: <https://vnptai.io/vi/blog/detail/convolutional-neural-networks-la-gi>. [Accessed: 22 Nov. 2025].