

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN

HỒ ĐĂNG CAO ĐỖ ĐỨC DUY

PHÂN LỚP ĐA ĐỐI TƯỢNG DỰA TRÊN  
MÔ HÌNH HỌC SÂU

KHÓA LUẬN TỐT NGHIỆP CỦ NHÂN CNTT  
CHƯƠNG TRÌNH CHẤT LƯỢNG CAO

Tp. Hồ Chí Minh, tháng 08/2024

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN

Hồ Đăng Cao - 20127452

Đỗ Đức Duy - 20127476

PHÂN LỐP ĐA ĐỐI TƯỢNG DỰA TRÊN  
MÔ HÌNH HỌC SÂU

KHÓA LUẬN TỐT NGHIỆP CỦ NHÂN CNTT  
CHƯƠNG TRÌNH CHẤT LƯỢNG CAO

GIẢNG VIÊN HƯỚNG DẪN  
TS. BÙI TIẾN LÊN

Tp. Hồ Chí Minh, tháng 08/2024

# Lời cảm ơn

Trước hết, chúng tôi xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Ts. Bùi Tiến Lên – người đã tận tình hướng dẫn, chỉ bảo nhóm trong suốt quá trình thực hiện khóa luận này. Nhờ sự truyền đạt kiến thức và những lời khuyên quý báu của thầy, nhóm chúng tôi đã có thể hoàn thành khóa luận này một cách tốt nhất.

Chúng tôi cũng xin bày tỏ lòng biết ơn đến các thầy/cô trong Khoa Công Nghệ Thông Tin, những người đã truyền đạt kiến thức và tạo điều kiện thuận lợi để chúng tôi có thể học tập và nghiên cứu trong suốt những năm học qua.

# Mục lục

<b>Lời cảm ơn</b>	i
<b>Mục lục</b>	ii
<b>Danh sách hình</b>	v
<b>Danh sách bảng</b>	viii
<b>Bảng ký hiệu</b>	ix
<b>Tóm tắt</b>	xi
<b>1 Giới thiệu tổng quan</b>	1
1.1 Động lực nghiên cứu . . . . .	1
1.2 Vấn đề nghiên cứu . . . . .	2
1.3 Đóng góp chính . . . . .	2
<b>2 Các công trình liên quan</b>	3
2.1 Học sâu dư thừa cho nhận dạng hình ảnh (ResNet) . . . . .	3
2.2 MobileNet V2 . . . . .	4
2.3 EfficientNet . . . . .	5
2.4 Ứng dụng Transformer cho nhận dạng hình ảnh . . . . .	7
2.5 Nhận dạng hình ảnh đa nhãn bằng mạng nơ-ron tích chập đồ thị . . . . .	9
2.6 Học Đa Nhãn Chỉ Từ Một Nhãn Dương . . . . .	11

2.6.1	Các biến thể tập dữ liệu . . . . .	12
2.6.2	Các hàm mất mát . . . . .	12
2.6.3	Kiến trúc mô hình . . . . .	18
2.6.4	Các chế độ học . . . . .	19
2.7	Phân loại ảnh đa nhãn bằng Transformer (C-Tran) . . . . .	20
2.7.1	Mô hình . . . . .	20
2.7.2	Các thành phần chính của mô hình . . . . .	21
2.7.3	Hàm mất mát . . . . .	22
<b>3</b>	<b>Các mô hình, phương pháp đề xuất</b>	<b>24</b>
3.1	Kỹ thuật cải tiến dữ liệu . . . . .	24
3.2	Mô hình Đơn nhãn dương . . . . .	28
3.2.1	Tóm tắt các đề xuất . . . . .	28
3.2.2	Các hàm mất mát . . . . .	29
3.2.3	Hàm kích hoạt và bộ ước lượng nhãn . . . . .	31
3.2.4	Bộ phân lớp . . . . .	33
3.2.5	Giá trị trung bình và độ lệch chuẩn . . . . .	34
3.3	Phân loại ảnh đa nhãn bằng Transformer (C-Tran) . . . . .	35
3.3.1	Thay đổi mạng trích xuất đặc trưng cho quá trình trích xuất đặc trưng từ hình ảnh . . . . .	35
3.3.2	Một số thay đổi khác . . . . .	35
3.4	Mô hình kết hợp . . . . .	36
<b>4</b>	<b>Thực nghiệm và đánh giá</b>	<b>38</b>
4.1	Giới thiệu tập dữ liệu . . . . .	38
4.1.1	Nguồn gốc tập dữ liệu . . . . .	38
4.1.2	Phân tích tập dữ liệu . . . . .	38
4.2	Cài đặt thí nghiệm . . . . .	43
4.2.1	Sự phân chia dữ liệu . . . . .	43
4.2.2	Dộ đo . . . . .	44
4.2.3	Các tham số của các mô hình . . . . .	45
4.3	Kết quả mô hình Đơn nhãn dương . . . . .	46

4.3.1	Kết quả . . . . .	46
4.3.2	Bàn luận và giải thích . . . . .	50
4.4	Kết quả mô hình C-Tran . . . . .	61
4.4.1	Các tham số và kết quả tương ứng . . . . .	61
4.4.2	Dồ thị của các kết quả tốt nhất . . . . .	62
4.5	Kết quả mô hình kết hợp . . . . .	65
<b>5</b>	<b>Kết luận và hướng phát triển</b>	<b>67</b>
<b>Danh mục công trình của tác giả</b>		<b>69</b>
<b>Tài liệu tham khảo</b>		<b>69</b>

# Danh sách hình

2.1	Kiến trúc của Khối học dư thừa [6] . . . . .	4
2.2	Kiến trúc của Khối học dư thừa với nút thắt cổ chai [6]. . . . .	4
2.3	Kiến trúc tổng quan của MobileNetV2 . . . . .	5
2.4	Mở rộng mô hình trong EfficientNet [16] . . . . .	6
2.5	Tổng quan về mô hình Transformer thị giác [4]. . . . .	8
2.6	Một số biểu diễn trực quan khi cho ảnh đi qua mô hình ViT.	8
2.7	Biểu diễn về sự chú ý từ token đầu ra đến không gian đầu vào [4]. . . . .	9
2.8	Đồ thị có hướng trên các nhãn đối tượng để mô hình hóa sự phụ thuộc nhãn trong nhận diện hình ảnh đa nhãn [2]. . . . .	10
2.9	Minh họa xác suất có điều kiện giữa hai nhãn [2]. . . . .	10
2.10	Tổng quan về mô hình Da nhãn-GCN [2]. . . . .	11
2.11	Mô hình học đa nhãn với Đơn nhãn dương. . . . .	18
2.12	Mô hình phân loại ảnh đa nhãn bằng Transformer [9]. . . . .	20
3.1	Mô phỏng cách cắt hình . . . . .	25
3.2	Cắt ảnh bằng khung chứa . . . . .	26
3.3	Cắt ảnh bằng phân vùng . . . . .	26
3.4	Ví dụ cắt ảnh của U-Net . . . . .	27
3.5	Mô hình học đa nhãn với đơn nhãn dương. . . . .	28
3.6	Mô hình kết hợp . . . . .	37
4.1	Top 30 nhãn có tần suất xuất hiện cao trong tập dữ liệu huấn luyện. . . . .	39

4.2	Top 30 nhãn có tần suất xuất hiện thấp trong tập dữ liệu huấn luyện. . . . .	40
4.3	Phân bố kích thước hình ảnh trong tập dữ liệu huấn luyện.	40
4.4	Top 30 nhãn có tần suất xuất hiện cao trong tập dữ liệu đánh giá. . . . .	41
4.5	Top 30 nhãn có tần suất xuất hiện thấp trong tập dữ liệu đánh giá. . . . .	42
4.6	Phân bố kích thước hình ảnh trong tập dữ liệu đánh giá. .	42
4.7	Một số hình ảnh chất lượng thấp . . . . .	43
4.8	Dồ thị học ROLE - đầu cuối . . . . .	49
4.9	Dồ thị học AN-LS - đầu cuối . . . . .	49
4.10	Dồ thị học AN-LS - tuyến tính . . . . .	50
4.11	Số lượng nhãn được phân lớp vào các ảnh. . . . .	51
4.12	Phân tích lỗi ở ngưỡng 0.5. . . . .	52
4.13	Ảnh mẫu nhóm nước . . . . .	54
4.14	Ảnh mẫu nhóm cà phê . . . . .	54
4.15	Phân tích lỗi ở trường hợp mọi bức ảnh đều được dự đoán vào một lớp. . . . .	55
4.16	Ma trận nhầm lẫn ở ngưỡng thỏa 1 nhãn lớn nhất. . . . .	57
4.17	Ảnh mẫu rice - risotto . . . . .	58
4.18	Ảnh mẫu nhóm bánh mì . . . . .	58
4.19	Ảnh mẫu nhóm salad . . . . .	59
4.20	Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có. . . . .	63
4.21	Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có. . . . .	63
4.22	Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có. . . . .	64

4.23 Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 243 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có. . . . .	64
4.24 Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 2 - Che nhãn huấn luyện: không. . . . .	65

# Danh sách bảng

4.1	Phân bố nhãn trong tập dữ liệu huấn luyện. . . . .	39
4.2	Phân bố nhãn trong tập dữ liệu đánh giá. . . . .	41
4.3	Bảng các tham số của mô hình Đơn nhãn dương. . . . .	45
4.4	Bảng các tham số của mô hình C-Tran. . . . .	46
4.5	Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng ResNet 50. . . . .	46
4.5	Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng ResNet 50. . . . .	47
4.6	Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng EfficientNet B7. . . . .	47
4.7	Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng EfficientNet B0. . . . .	47
4.7	Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng EfficientNet B0. . . . .	48
4.8	Các cặp (nhãn đúng, nhãn đoán) ở ngưỡng 0.5 . . . . .	53
4.9	Bảng các nhóm nhãn thường xuất hiện cùng với nhau trong tập huấn luyện và đánh giá. . . . .	59
4.10	Bảng số lần xuất hiện riêng lẻ của các nhãn thường xuất hiện cùng với nhau trong bảng 4.9. . . . .	60
4.11	Bảng các tham số và kết quả tương ứng của C-Tran . . . . .	61
4.12	Bảng các tham số chung của mô hình kết hợp. . . . .	65
4.13	Bảng các tham số và kết quả tương ứng của mô hình kết hợp khi sử dụng các tham số ở bảng 4.12. . . . .	66

# Thuật ngữ chuyên môn

Tiếng Anh	Tiếng Việt
AN	Kỹ thuật giả sử nhãn không biết là âm
AN-LS	Kỹ thuật giả sử nhãn không biết là âm kết hợp làm mịn nhãn
BiFPN	Mạng đặc trưng kim tự tháp 2 chiều
BCE	Độ chêch chéo nhị phân
bottom-right	Điểm dưới cùng bên phải của khung cắt
CAM	Bản đồ kích hoạt lớp
CNN	Mạng nơ-ron tích chập
CT	Chuyển tải có điều kiện
EPR	Kỹ thuật điều chuẩn dương kì vọng
FFN	Mạng chuyển tiếp độc lập
FLOPS	Số Phép Tính Dấu Phẩy Động Mỗi Giây
IoU	Do lường mức độ chồng lấn giữa vùng dự đoán và vùng thực tế của đối tượng trong ảnh
IUN	Kỹ thuật bỏ qua nhãn âm không biết
IU	Kỹ thuật bỏ qua nhãn không biết
KL Divergence	Phân kỳ Kullback-Leibler
logit	Các giá trị thu được sau khi qua mạng FFN
MAE	Sai số tuyệt đối trung bình
MSE	Sai số bình phương trung bình

Tiếng Anh	Tiếng Việt
Perceptron	Mô hình mạng nơ-ron đơn giản được sử dụng để phân loại dữ liệu
RGB	Hệ màu đỏ, lục, lam
ROLE	Kỹ thuật ước lượng nhãn trực tuyến điều chuẩn
Token	Đơn vị cơ bản của văn bản, thường là một từ, chữ hoặc dấu câu
top-left	Điểm trên cùng bên trái của khung cắt
Transformer Encoder	Bộ mã hóa của mô hình Transformer
WAN	Kỹ thuật giả sử nhãn không biết là âm được đánh trọng số

# Tóm tắt

Nghiên cứu này trong lĩnh vực thị giác máy tính tập trung vào việc cải thiện chất lượng của tập dữ liệu trong bài toán phân lớp đa nhãn. Các vấn đề cần giải quyết bao gồm độ phân giải kém, nhiễu, sự chênh lệch số lượng nhãn và ảnh, cũng như nhãn đồng nghĩa hoặc bao hàm lẫn nhau. Để đạt được mục tiêu, nghiên cứu áp dụng các kỹ thuật xử lý hình ảnh để loại bỏ cảnh nền không cần thiết, cải tiến mô hình bằng cách thay đổi hàm kích hoạt, thử nghiệm hàm mất mát mới, thay đổi kiến trúc và kết hợp các mô hình. Dù hạn chế về tài nguyên, nghiên cứu vẫn đạt được kết quả khả quan, với độ chính xác cao và giảm số lượng tham số của mô hình gốc.

# Chương 1

## Giới thiệu tổng quan

*Ở chương này, chúng tôi mô tả chung các vấn đề thường gặp phải khi phát triển một mô hình phân lớp đa nhãn và những hạn chế của chúng. Đồng thời đề xuất hướng giải quyết mới của chúng tôi trong bài báo cáo này.*

### 1.1 Động lực nghiên cứu

Trong lĩnh vực thị giác máy tính nói chung, chất lượng tập dữ liệu ảnh đóng vai trò rất quan trọng trong bài toán phân lớp đa nhãn. Tập dữ liệu chất lượng cao, cân bằng và đầy đủ sẽ giúp mô hình tránh được hiện tượng quá khớp, tức là mô hình chỉ hoạt động tốt trên dữ liệu huấn luyện nhưng kém hiệu quả trên dữ liệu mới. Ngoài ra, tập dữ liệu có chất lượng cao giúp dễ dàng phát hiện và xử lý các vấn đề kỹ thuật liên quan đến hình ảnh, như độ phân giải kém, nhiễu. Điều này giúp mô hình có thể tập trung vào việc học các đặc trưng chính xác hơn, từ đó tăng hiệu quả huấn luyện và giảm thời gian cần thiết để đạt được độ chính xác mong muốn. Dữ liệu tốt làm cho quá trình huấn luyện trở nên mượt mà và ít gặp phải các vấn đề như nhiễu hay thông tin không cần thiết.

## 1.2 Vấn đề nghiên cứu

Các tập dữ liệu ảnh trong thực tế lại dễ dàng xuất hiện các vấn đề kỹ thuật được nêu trước đó. Điều này đối với các tập dữ liệu lớn thì có thể không là vấn đề lớn, do qua quá trình huấn luyện lượng lớn, đa dạng các ảnh khác nhau, mô hình có thể học được chính xác các đặc trưng của các đối tượng chính và bỏ qua các đặc trưng gây nhiễu. Tuy nhiên, đối với các tập dữ liệu nhỏ hoặc trung bình, chúng tôi cho rằng chất lượng tập dữ liệu sẽ ảnh hưởng lớn, trực tiếp đến hiệu suất và độ chính xác của mô hình học máy. Kích thước của tập dữ liệu cần được cải thiện bằng chất lượng.

Ngoài ra, chất lượng dữ liệu ảnh tốt giúp mô hình học máy có khả năng khai quát hóa tốt hơn, tức là có thể áp dụng chính xác vào các tập dữ liệu mới, chưa được huấn luyện. Điều này đặc biệt quan trọng trong bài toán phân lớp đa nhãn, nơi mà một ảnh có thể có nhiều nhãn khác nhau. Tuy nhiên, trong phân lớp đa nhãn số lượng nhãn ở mỗi bức ảnh có sự chênh lệch lớn, tức là số lượng ảnh có 1 nhãn lớn hơn nhiều số lượng ảnh có 10 nhãn. Và số lượng ảnh ở mỗi nhãn cũng có sự chênh lệch lớn, tức số lượng ảnh chứa nhãn A lớn hơn nhiều số lượng ảnh chứa nhãn B. Ngoài ra, tập nhãn càng lớn càng dễ dẫn đến trường hợp các nhãn đồng nghĩa hoặc nghĩa của nhãn này bao hàm luôn nghĩa của nhãn kia, ví dụ: tập nhãn vừa chứa nhãn cà phê vừa chứa nhãn cappuccino - 1 loại cà phê.

## 1.3 Đóng góp chính

Chúng tôi cho rằng đầu tư vào việc thu thập và xử lý dữ liệu chất lượng cao là một bước quan trọng và cần thiết. Chúng tôi tiến hành đề xuất những phương pháp để xử lý dữ liệu hình ảnh đầu vào. Đồng thời cũng đề xuất thêm những phương pháp cải tiến, thay đổi, cũng như kết hợp đối với mô hình đã được nghiên cứu trước đây. Với những đề xuất này, chúng tôi hy vọng sẽ mang lại những kết quả tốt trong nhiệm vụ phân lớp đa nhãn.

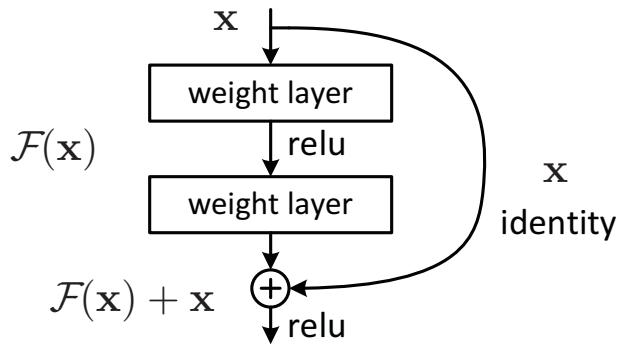
## Chương 2

# Các công trình liên quan

*Trong chương này sẽ trình bày các công trình và phương pháp được chúng tôi tìm hiểu cho bài toán phân lớp đa nhãn.*

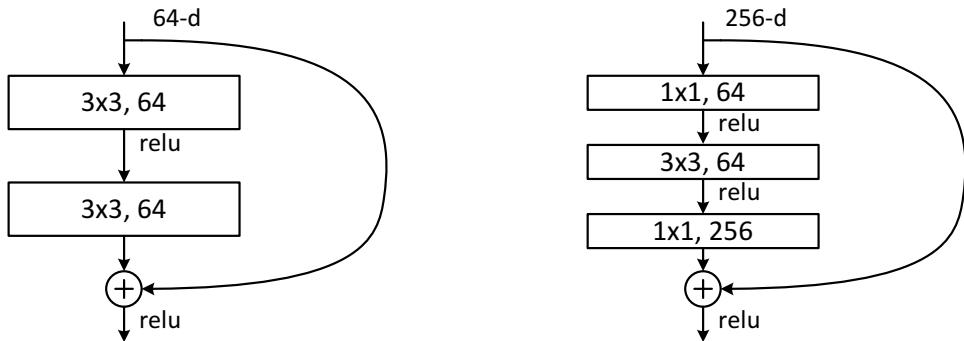
### 2.1 Học sâu dư thừa cho nhận dạng hình ảnh (ResNet)

Phương pháp học phần dư được đề xuất để giải quyết vấn đề khó khăn trong việc huấn luyện tạo các mạng nơ-ron sâu, vốn gặp phải hiện tượng tăng độ lỗi huấn luyện khi tăng số lớp của mạng. Thay vì xây dựng các lớp tích chập xếp chồng như mạng CNN thông thường, ResNet sẽ bổ sung đầu vào của lớp hiện tại bằng cách cộng đầu vào của lớp hiện tại với đầu vào của vài lớp trước nó, điều giúp cho việc tối ưu hóa dễ dàng hơn nhằm cải thiện độ chính xác của mô hình.



**Hình 2.1:** Kiến trúc của Khối học dư thừa [6].

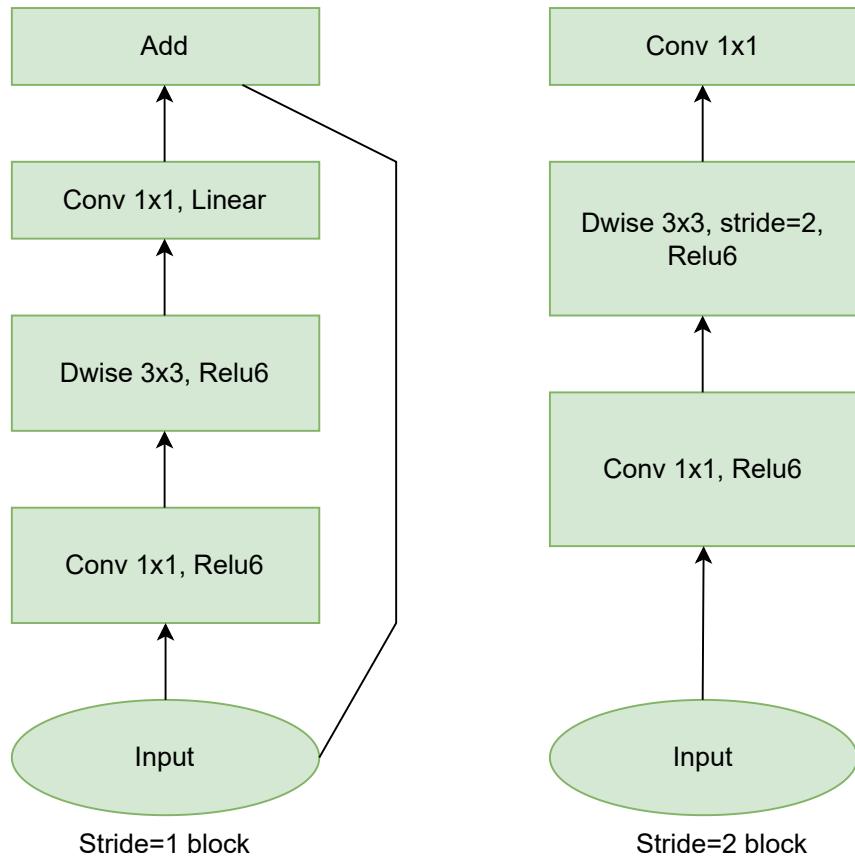
Ngoài ra, ResNet còn đề xuất kiến trúc nút thắt cổ chai giúp giảm tham số cho mô hình và chi phí tính toán để áp dụng cho các mạng sâu hơn.



**Hình 2.2:** Trái: Kiến trúc Khối học dư thừa thông thường. Phải: Kiến trúc Khối nút thắt cổ chai [6].

## 2.2 MobileNet V2

MobileNet V2 giới thiệu một kiến trúc mạng nơ-ron mới, tối ưu cho môi trường di động và hạn chế tài nguyên, giúp giảm đáng kể lượng tính toán và bộ nhớ cần thiết trong khi vẫn duy trì độ chính xác cao. Đóng góp chính của MobileNet V2 là mô-đun: khối phần dư đảo ngược kết hợp với nút thắt cổ chai tuyến tính. Mô-đun này giúp nắm bắt nhiều thông tin cần thiết hơn cũng như giảm sự mất mát thông tin mà các lớp phi tuyến gây ra.



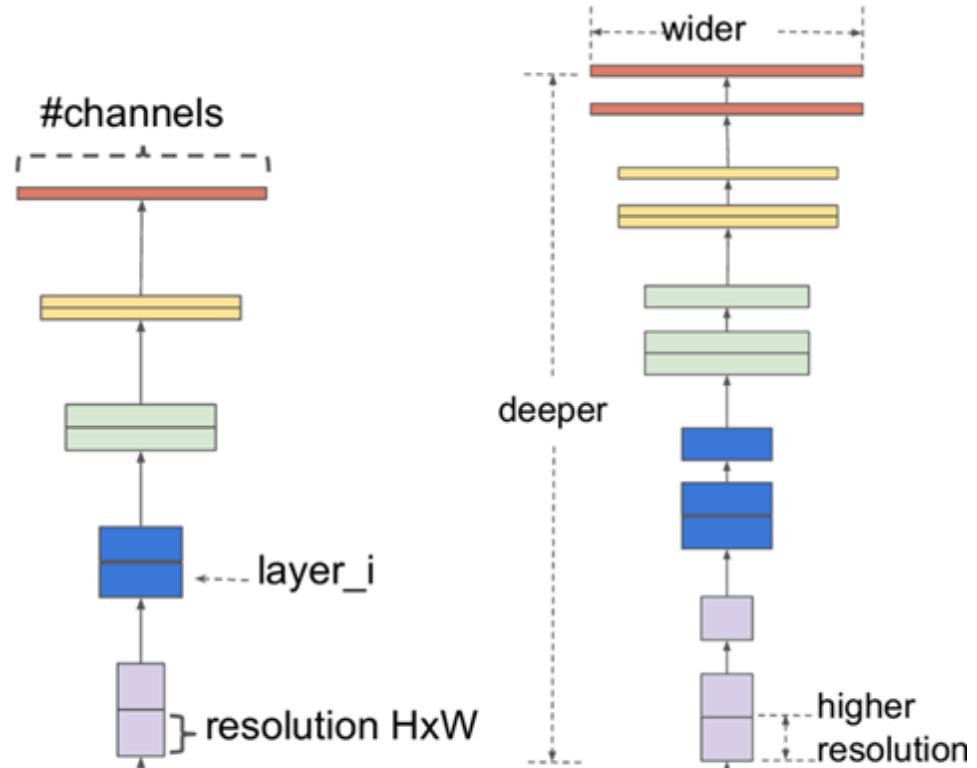
**Hình 2.3:** Kiến trúc tổng quan của MobileNet V2.

## 2.3 EfficientNet

EfficientNet là một họ kiến trúc mạng nơ-ron sâu sử dụng phương pháp mở rộng kết hợp, cân bằng cản thận độ sâu của mạng (lớp), chiều rộng (kênh), và độ phân giải với tỷ lệ không đổi. Mô hình không chỉ làm tăng độ chính xác mà còn cải thiện hiệu suất bằng cách giảm số lượng tham số và FLOPS (Floating Point Operations Per Second).

Bắt đầu là EfficientNet B0. Mạng có kiến trúc tương tự như MnasNet [17] với khối xây dựng chính là tích chập của MobileNet (MBConv) [14] và tối ưu hóa squeeze-and-excitation (SE) [7], tập trung vào việc mô hình hóa sự phụ thuộc lẫn nhau giữa các kênh khác nhau. Sau đó, áp dụng phương

pháp mở rộng kết hợp để có được các phiên bản mô hình tốt hơn.



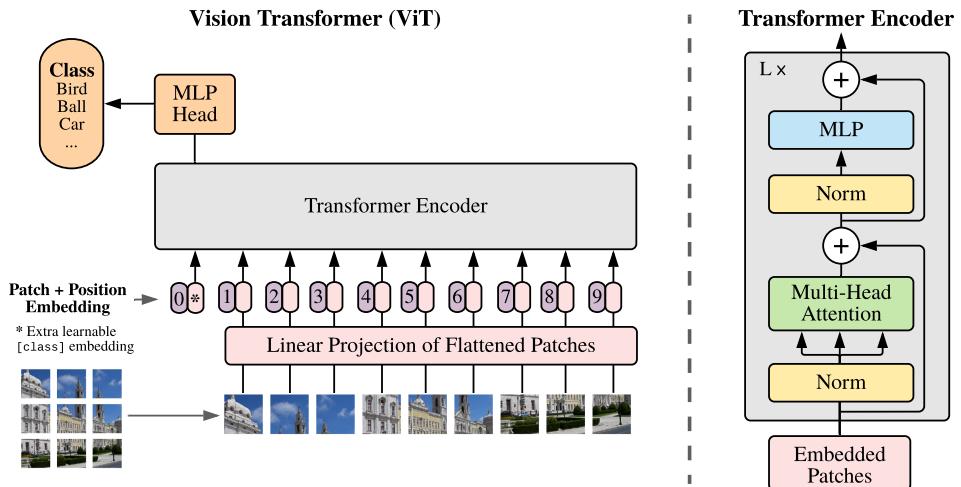
**Hình 2.4:** Tinh chỉnh mô hình trong EfficientNet [16] Bên trái là ví dụ về mạng cơ sở; Bên phải là phương pháp mở rộng kết hợp, mở rộng đều cả ba chiều theo một tỷ lệ cố định.

Phương pháp tuân theo nguyên tắc: độ sâu  $d = \alpha^\phi$ , chiều rộng  $w = \beta^\phi$  và độ phân giải  $r = \gamma^\phi$ . Trong đó,  $\phi$  là hệ số mở rộng toàn bộ, kiểm soát lượng tài nguyên có sẵn;  $\alpha, \beta, \gamma$  xác định cách phân bố tài nguyên cho độ sâu, độ rộng và độ phân giải của mạng. FLOPS của phép tính tích chập tỉ lệ với  $d, w^2, r^2$ . Do đó mở rộng mạng cơ sở lên  $\phi$  sẽ tăng tổng số FLOPS lên  $(\alpha \cdot \beta^2 \cdot \gamma^2)^\phi$ . Vì vậy để đảm bảo FLOPS không vượt quá  $2^\phi$  thì cần rằng buộc  $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$  với  $\alpha \geq 1, \beta \geq 1, \gamma \geq 1$ . Điều này có nghĩa rằng nếu có gấp đôi tài nguyên, chúng ta đơn giản có thể sử dụng hệ số kết hợp  $\phi = 1$  để mở rộng FLOPS lên  $2^1$ . Các tham số  $\alpha, \beta$  và  $\gamma$  được xác định bằng phương pháp tìm kiếm lưỡng. Cố định  $\phi = 1$  và tìm bộ tham số để mô hình đạt được độ chính xác cao nhất. Sau đó, chúng ta cố định  $\alpha, \beta, \gamma$  rồi tăng giá trị  $\phi$  để thu được các mô hình EfficientNet B1 đến B7. Chúng lớn

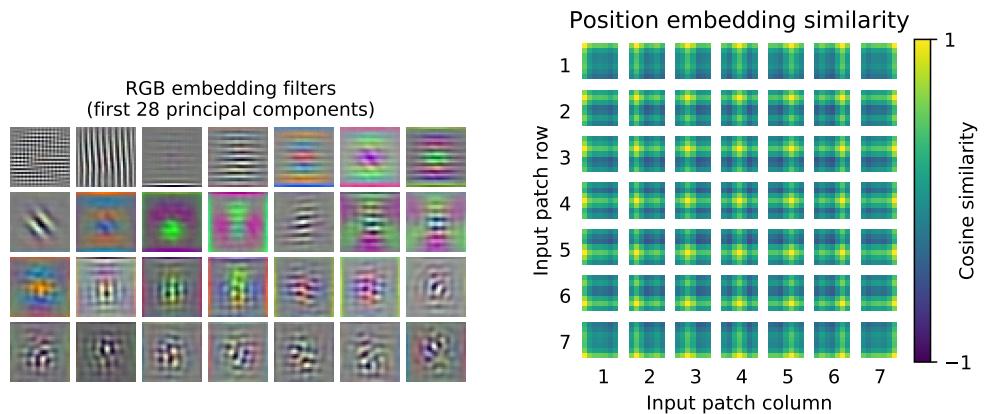
hơn nhưng chính xác hơn so với mạng EfficientNet B0 đầu tiên.

## 2.4 Ứng dụng Transformer cho nhận dạng hình ảnh

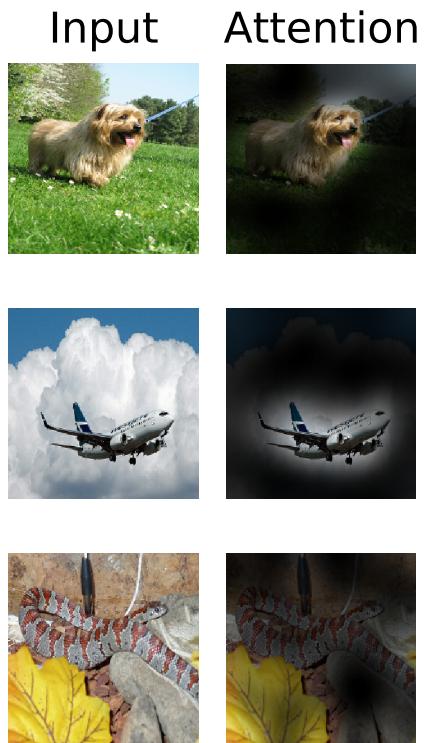
Kiến trúc Transformer đã trở thành tiêu chuẩn cho các nhiệm vụ xử lý ngôn ngữ tự nhiên, tuy nhiên ứng dụng của nó vào các nhiệm vụ thị giác máy tính vẫn còn hạn chế. Trong thị giác máy tính, cơ chế chú ý thường được kết hợp cùng với các mạng nơ-ron tích chập (CNN), hoặc được sử dụng để thay thế một số thành phần của mạng CNN trong khi vẫn giữ nguyên cấu trúc tổng thể của chúng. Tuy nhiên, kiến trúc Transformer thuận tiện khi được áp dụng trực tiếp lên các chuỗi mảng ghép hình ảnh, có thể thực hiện rất tốt các nhiệm vụ phân loại hình ảnh. Transformer thị giác (ViT) sử dụng các nguyên tắc cơ bản của kiến trúc Transformer, chia hình ảnh thành các mảng nhỏ và xử lý chúng như chuỗi các tokens trong xử lí ngôn ngữ tự nhiên. Các mảng hình ảnh được nhúng vào không gian một chiều, đồng thời sử dụng nhúng vị trí để giữ thông tin về vị trí của các mảng hình ảnh. Sau đó sẽ được đưa vào khối Transformer Encoder bao gồm các lớp tự - chú ý đa đầu, cuối cùng sẽ sử dụng lớp perceptron để phân loại.



**Hình 2.5:** Tổng quan về mô hình Transformer thị giác. Hình ảnh được chia thành các mảng có kích thước cố định, nhúng tuyến tính từng mảng, thêm thông tin nhúng vị trí vào các mảng, và đưa chuỗi các véc-tơ này vào bộ mã hóa Transformer tiêu chuẩn [4].



**Hình 2.6: Bên trái:** Thể hiện bộ lọc nhúng tuyến tính các giá trị RGB sau khi đã giảm chiều. **Bên phải:** Biểu diễn sự tương đồng cosin của các mảng hình ảnh sau khi được nhúng vị trí [4].

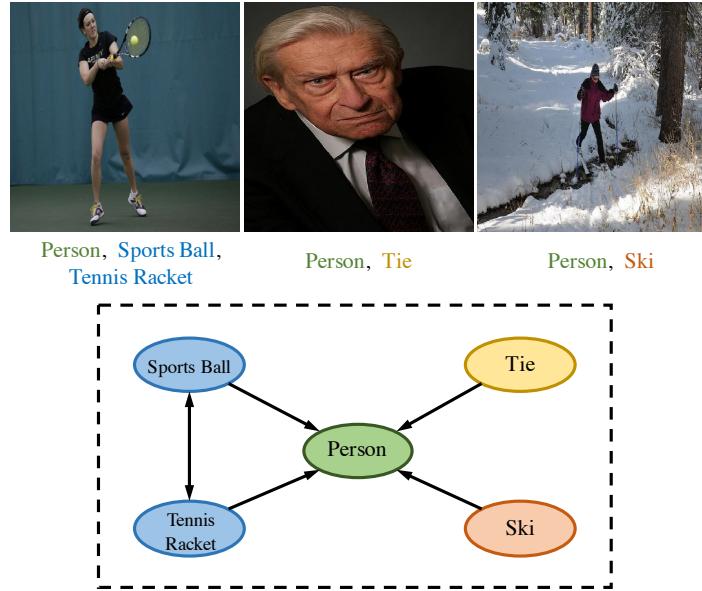


**Hình 2.7:** Biểu diễn về sự chú ý từ token đầu ra đến không gian đầu vào [4].

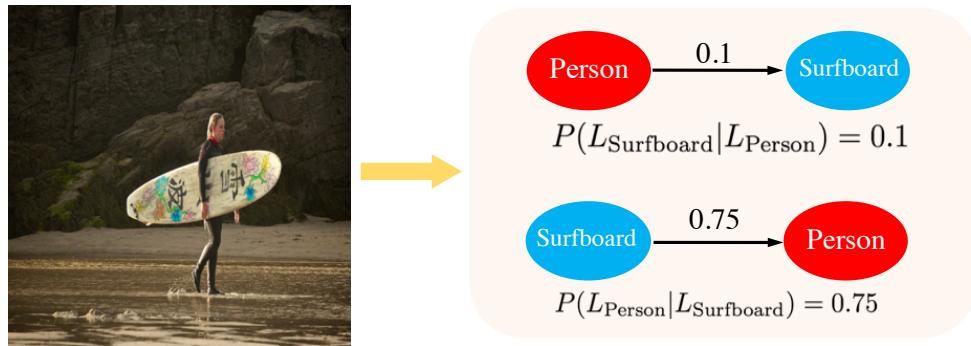
## 2.5 Nhận dạng hình ảnh đa nhãn bằng mạng nơ-ron tích chập đồ thị

Nhiệm vụ nhận dạng đa nhãn là dự đoán sự xuất hiện của các đối tượng trong hình ảnh. Vì các đối tượng thường xuất hiện cùng nhau, nên việc mô hình hóa sự phụ thuộc của các nhãn sẽ giúp ích cho việc cải thiện hiệu suất nhận dạng của mô hình. Dựa trên ý tưởng đó, một mô hình hoàn chỉnh về nhận diện hình ảnh đa nhãn dựa trên mạng Nơ-ron tích chập đồ thị (GCN) đã được nghiên cứu và phát triển. Mô hình này xây dựng một đồ thị có hướng trên các nhãn đối tượng, trong đó mỗi nút (nhãn) được biểu diễn bằng các véc-tơ nhúng từ của một nhãn, và GCN được học để ánh xạ đồ thị nhãn này thành một tập hợp các bộ phân loại đối tượng phụ thuộc lẫn nhau. Các bộ phân loại này được áp dụng cho các mô tả hình

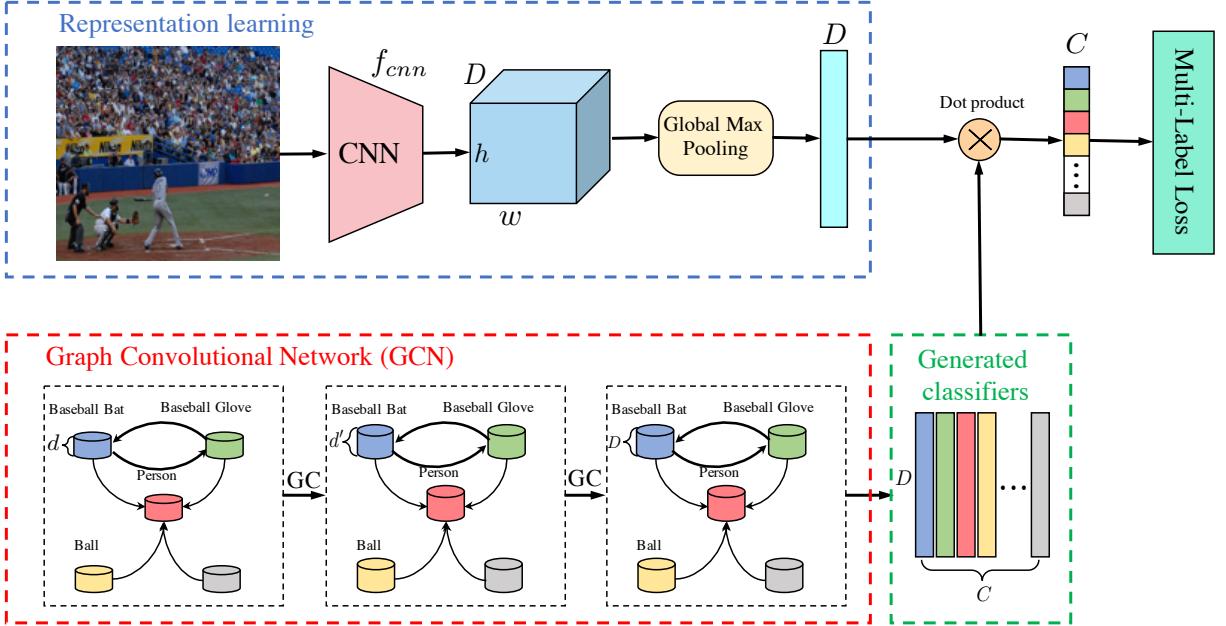
ảnh được trích xuất bởi một mạng con khác, cho phép toàn bộ mạng có thể được huấn luyện đầu-cuối.



**Hình 2.8:** Đồ thị có hướng trên các nhãn đối tượng để mô hình hóa sự phụ thuộc nhãn trong nhận diện hình ảnh đa nhãn. Trong hình này,  $\text{Label}_A \rightarrow \text{Label}_B$  có nghĩa là khi  $\text{Label}_A$  xuất hiện thì  $\text{Label}_B$  có khả năng xuất hiện, nhưng điều ngược lại có thể không đúng [2].



**Hình 2.9:** Minh họa xác suất có điều kiện giữa hai nhãn. Khi "ván lướt sóng" xuất hiện trong hình ảnh, "người" cũng sẽ xuất hiện với xác suất cao. Tuy nhiên, trong điều kiện "người" xuất hiện, "ván lướt sóng" không nhất thiết sẽ xuất hiện. [2].



**Hình 2.10:** Mô hình Đa nhãn-GCN cho nhận diện hình ảnh đa nhãn. Các nhãn đối tượng được biểu diễn bằng các nhúng từ  $Z \in \mathbb{R}^{C \times d}$  (trong đó  $C$  là số lượng nhãn và  $d$  là kích thước của véc-tơ nhúng từ). Một đồ thị có hướng được xây dựng dựa trên các biểu diễn nhãn này, mỗi nút biểu diễn một nhãn. Các GCN xếp chồng được huấn luyện trên đồ thị nhãn để ánh xạ các biểu diễn nhãn này thành một tập hợp các bộ phân loại đối tượng liên thuộc, tức là  $W \in \mathbb{R}^{C \times D}$ , được áp dụng cho biểu diễn hình ảnh được trích xuất từ hình ảnh đầu vào thông qua mảng tích chập để nhận diện hình ảnh đa nhãn [2].

## 2.6 Học Đa Nhãn Chỉ Từ Một Nhãn Dương

Ở phương pháp học PU trước đây [11], dữ liệu chỉ có hai loại gồm các mẫu dương và các mẫu không được gán nhãn. Mục tiêu là xây dựng mô hình để phân loại các mẫu không được gán nhãn thành nhãn dương hoặc không phải nhãn dương. Học Đa Nhãn Chỉ Từ Một Nhãn Dương (gọi tắt là mô hình Đơn nhãn dương) [3] có sự tương đồng với học PU về các tập huấn luyện sẽ chỉ có một nhãn dương cho mỗi hình ảnh và không có nhãn âm nào được xác nhận. Tuy nhiên, sẽ mở rộng các hàm mất mát đa nhãn hiện có cho trường hợp này với nhiều chế độ học khác nhau. Chỉ học trên bộ phân loại tuyến tính và học trên toàn bộ mạng học sâu được tinh chỉnh

từ đầu đến cuối. Đây là mô hình cùng với C-Tran 2.7 sẽ được sử dụng chủ yếu trong khóa luận này.

### 2.6.1 Các biến thể tập dữ liệu

Với  $L$  là số lượng nhãn,  $N$  là số lượng ảnh, tập dữ liệu được chia làm ba loại gồm tập được quan sát đầy đủ, tập được quan sát một phần và tập Đơn nhãn dương.

Tập dữ liệu được quan sát đầy đủ được ký hiệu như sau  $\{(x_n, y_n)\}_{n=1}^N$ , trong đó mỗi  $x_n \in$  không gian đầu vào  $\mathcal{X}$  và có liên kết với một véc-tơ nhãn  $y_n \in \mathcal{Y} = \{0, 1\}^L$ . Ở đây,  $y_{ni} = 1$  chỉ ra rằng nhãn thứ  $i$  được phân cho  $x_n$ , còn  $y_{ni} = 0$  thì ngược lại.

Trong tập dữ liệu được quan sát một phần  $\{(x_n, z_n)\}_{n=1}^N$ , mỗi  $z_n \in \mathcal{Z} = \{0, 1, \emptyset\}^L$ . Tại đây,  $z_{ni} = \emptyset$  cho biết nhãn thứ  $i$  không được quan sát (không biết) đối với  $x_n$ , và  $y_{ni}$  có thể là 0 hoặc 1.

Trường hợp Đơn nhãn dương đặt ra rằng chỉ có một nhãn dương duy nhất được quan sát trên mỗi mẫu huấn luyện và các nhãn khác là không biết, tức  $z_{ni} \in \{1, \emptyset\}$  và  $\sum_{i=1}^L \mathbb{I}_{[z_{ni}=1]} = 1, \forall n \in \{1, \dots, N\}$ . Ở đây,  $\mathbb{I}_{[\cdot]}$  là hàm chỉ báo, trong đó  $\mathbb{I}_{[z_{ni}=1]} = 1$  nếu  $z_{ni} = 1$ , và 0 trong trường hợp ngược lại.

### 2.6.2 Các hàm măt mát

Để xây dựng các hàm măt mát chúng ta cần véc-tơ dự đoán  $\mathbf{f}_n = f(x_n; \theta) \in [0, 1]^L$  cung cấp xác suất dự đoán cho mỗi nhãn của  $x_n$ . Và  $f_{ni}$  biểu thị phần tử thứ  $i$  của  $\mathbf{f}_n$ . Chú ý rằng, tổng của  $f_{ni}$  qua tất cả các nhãn  $i$  không nhất thiết phải bằng 1.

## 1) Toàn bộ các nhãn được quan sát:

Hàm mất mát Binary Cross-Entropy (BCE):

$$\mathcal{L}_{BCE}(\mathbf{f}_n, \mathbf{y}_n) = -\frac{1}{L} \sum_{i=1}^L [\mathbb{I}_{[y_{ni}=1]} \log(f_{ni}) + \mathbb{I}_{[y_{ni}=0]} \log(1-f_{ni})] \quad (2.1)$$

Trong đó,  $\mathbb{I}_{[y_{ni}=1]}$  thay cho  $P(y_i = 1|x_n)$  và  $\mathbb{I}_{[y_{ni}=0]}$  thay cho  $P(y_i = 0|x_n)$ .

## 2) Các nhãn được quan sát một phần

Hàm mất mát bỏ qua các nhãn không được quan sát: giả sử các nhãn không được quan sát được dự đoán chính xác một cách hoàn hảo, tức  $f_{ni} = P(y_i = 1|x_n)$  nếu  $z_{ni} = \emptyset$

$$\mathcal{L}_{IU}(\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L [\mathbb{I}_{[z_{ni}=1]} \log(f_{ni}) + \mathbb{I}_{[z_{ni}=0]} \log(1-f_{ni})] \quad (2.2)$$

Trong đó,  $\mathbb{I}_{[z_{ni}=1]}$  thay cho  $P(y_i = 1|x_n)$  và  $\mathbb{I}_{[z_{ni}=0]}$  thay cho  $P(y_i = 0|x_n)$ . Tương tự như hàm  $\mathcal{L}_{BCE}$  trong trường hợp toàn bộ nhãn được quan sát, nhưng bỏ qua các nhãn  $z_{ni} = \emptyset$ .

Hàm mất mát bỏ qua các nhãn âm không được quan sát:

$$\mathcal{L}_{IUN}(\mathbf{f}_n, \mathbf{z}_n, \mathbf{y}_n) = -\frac{1}{L} \sum_{i=1}^L [\mathbb{I}_{[z_{ni}=1]} \log(f_{ni}) + \mathbb{I}_{[y_{ni}=0]} \log(1-f_{ni})] \quad (2.3)$$

Đây là một cách tiếp cận không thực tế khi sử dụng trực tiếp các nhãn âm đúng trên véc-tơ quan sát đầy đủ  $y_n$

## 3) Đơn nhãn dương

Trong trường hợp này, nhãn duy nhất được quan sát là nhãn dương i.e.  $z_{ni} \neq \emptyset \rightarrow z_{ni} = 1$ . Khi chỉ còn nhãn dương, hàm mất mát BCE sẽ trở

thành:

$$\mathcal{L}_{BCE}^+ (\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L \mathbb{I}_{[z_{ni}=1]} \log (f_{ni}) \quad (2.4)$$

Độ lỗi lúc này sẽ giảm vì không có sự phạt cho việc dự đoán sai nhãm âm. Điều này dễ dàng dẫn đến vấn đề rằng mô hình sẽ dự đoán tất cả các lớp là dương vì không có nhãm âm để so sánh. Vấn đề này có thể được giải quyết theo 2 hướng. Hướng thứ nhất là bổ sung nhãm âm từ các nhãm không được quan sát. Hướng thứ hai là bổ sung mức phạt khi mô hình dự đoán quá nhiều nhãm dương.

Ở hướng thứ nhất, các nhãm không được quan sát được giả sử là âm, i.e.  $P(y_{ni} = 1|x_n) = 0$  nếu  $z_{ni} = \emptyset$ . Chúng ta được hàm mất mát giả sử nhãm âm:

$$\mathcal{L}_{AN} (\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L [\mathbb{I}_{[z_{ni}=1]} \log (f_{ni}) + \mathbb{I}_{[z_{ni}\neq 1]} \log (1 - f_{ni})] \quad (2.5)$$

Hạn chế của hướng đi này là việc tạo ra các nhãm âm sai (nhiều nhãm). Điều này làm giảm đáng kể độ chính xác. Để hạn chế ảnh hưởng của nhiều nhãm, độ lỗi của các nhãm âm giả sẽ được thêm trọng số. Lúc này, chúng ta được hàm mất mát giả sử nhãm âm yếu:

$$\mathcal{L}_{WAN} (\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L [\mathbb{I}_{[z_{ni}=1]} \log (f_{ni}) + \mathbb{I}_{[z_{ni}\neq 1]} \gamma \log (1 - f_{ni})] \quad (2.6)$$

Với  $\gamma = \frac{1}{L-1} \in [0; 1]$ , nhãm dương duy nhất sẽ có ảnh hưởng lên hàm mất mát giống với  $L-1$  nhãm được giả sử là âm còn lại. Điều này làm giảm sự đóng góp của các nhãm âm giả vào hàm mất mát.

Ngoài ra, để giảm tác động của nhiều nhãm, chúng ta có thể kết hợp hàm mất mát giả sử nhãm âm  $\mathcal{L}_{AN}$  với kỹ thuật làm mịn nhãm. Trong bài toán phân loại đa lớp, mỗi ảnh chỉ có một nhãm dương, việc làm mịn nhãm thay véc-tơ  $y_n$  bằng  $(1-\epsilon)y_n + \epsilon u$ , với  $u = [\frac{1}{L}, \dots, \frac{1}{L}]$  và  $\epsilon \in (0, 1)$ , làm cho các giá trị trong  $y_n$  trở nên mềm hơn. Nhãm dương giảm nhẹ từ 1 xuống

$1 - \epsilon(1 - \frac{1}{L})$ , và các nhãn khác từ 0 trở thành  $\frac{\epsilon}{L}$ .

Trong bối cảnh nhị phân, mỗi lớp được xem như một bài toán phân loại nhị phân riêng biệt ( $L = 2$ ). Do đó, làm mịn nhãn có thể được áp dụng độc lập cho từng mục tiêu nhị phân ( $\mathbb{I}_{[z_{ni}=1]}, \mathbb{I}_{[z_{ni}\neq 1]}$ ). Hàm chỉ báo trong trường hợp này được làm mịn thành  $\mathbb{I}_{[Q]}^{\frac{\epsilon}{2}} = (1 - \frac{\epsilon}{2})\mathbb{I}_{[Q]} + \frac{\epsilon}{2}\mathbb{I}_{[-Q]}$ , với mọi mệnh đề  $Q$ . Áp dụng hàm chỉ báo mới vào hàm mất mát giả sử nhãn âm  $\mathcal{L}_{AN}$ , ta thu được hàm mất mát:

$$\mathcal{L}_{AN-LS}(\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L \left[ \mathbb{I}_{[z_{ni}=1]}^{\frac{\epsilon}{2}} \log(f_{ni}) + \mathbb{I}_{[z_{ni}\neq 1]}^{\frac{\epsilon}{2}} \log(1 - f_{ni}) \right] \quad (2.7)$$

Trong thực nghiệm,  $\epsilon = 0.1$  được sử dụng theo công trình [3].

Ở hướng thứ hai, chúng ta cần quan tâm đến số nhãn dương trung bình mỗi bức ảnh:

$$k = \frac{\sum_{n=1}^N \sum_{i=1}^L \mathbb{I}_{[y_{ni}=1]}}{N} \quad (2.8)$$

Đây là số nhãn dương mà mỗi bức ảnh kỳ vọng đạt được và được xem là siêu tham số trong quá trình huấn luyện.

Bên cạnh đó, giả sử chúng ta lấy một lô  $B$  có  $|B|$  ảnh. Gọi  $F_B = [f_{ni}]_{n \in B, i \in \{1, \dots, L\}}$  là ma trận các dự đoán  $f_{ni} \in [0, 1]$  cho mỗi ảnh trong  $B$  và mỗi nhãn trong tập dữ liệu. Chúng ta tính trung bình các dự đoán  $f_{ni}$  trên tất cả các nhãn  $i$  của từng ảnh trong lô  $B$ . Nó được xem như là số nhãn dương dự đoán trung bình mỗi bức ảnh:

$$\hat{k}(F_B) = \frac{\sum_{n \in B} \sum_{i=1}^L f_{ni}}{|B|} \quad (2.9)$$

Với mong muốn  $\hat{k}(F_B)$  gần với  $k$ , điều chuẩn  $R_k(F_B)$  được thêm vào để khuyến khích mô hình học. Vì mỗi tập dữ liệu có số lượng nhãn  $L$  khác nhau, nên cần chuẩn hóa sự sai lệch  $\hat{k}(F_B)$  so với  $k$  để phù hợp giữa các tập dữ liệu.

$$\frac{|\hat{k}(F_B) - k|}{L} \in [0, 1] \quad (2.10)$$

Vì cùng lệch 1 nhän, những lệch 1 nhän trong 10 nhän ( $L = 10$ ) sẽ nghiêm trọng hơn trong 100 nhän ( $L = 100$ ), tương ứng  $\frac{|\hat{k}(F_B) - k|}{10} > \frac{|\hat{k}(F_B) - k|}{100}$ . Ngoài ra, để đảm bảo hàm măt măt luôn khả vi tại mọi điểm và mô hình có thể phạt càng nặng khi độ lệch càng cao, công thức điều chuẩn cuối cùng là bình phương sai số:

$$R_k(\mathbf{F}_B) = \left( \frac{\hat{k}(F_B) - k}{L} \right)^2 \in [0; 1] \quad (2.11)$$

Kết hợp điều chuẩn  $R_k(F_B)$  với  $\mathcal{L}_{BCE}^+(\mathbf{f}_n, \mathbf{z}_n)$ , chúng ta được hàm măt măt điều chuẩn dương kỳ vọng (EPR):

$$\mathcal{L}_{EPR}(\mathbf{F}_B, \mathbf{Z}_B) = \frac{1}{|B|} \sum_{n \in B} \mathcal{L}_{BCE}^+(\mathbf{f}_n, \mathbf{z}_n) + \lambda R_k(\mathbf{F}_B) \quad (2.12)$$

Trong đó  $\lambda$  là một siêu tham số cho điều chuẩn.  $Z_B = [z_{ni}]_{n \in B, i \in \{1, \dots, L\}}$  là ma trận các quan sát  $z_{ni} \in \{\emptyset, 1\}$  cho mỗi ảnh trong lô  $B$ . Hàm măt măt EPR thực hiện điều chuẩn ở mức lô (thay vì mức ảnh) vì thực tế rằng một số ảnh sẽ có nhiều hơn hoặc ít hơn  $k$  nhän dương. Hàm măt măt này ngầm phạt các nhän âm, tránh tình huống mô hình luôn dự đoán dương và vấn đề nhiễu nhän của hướng giả sử nhän âm.

Tuy nhiên, EPR lại phụ thuộc vào số lượng nhän dương kỳ vọng  $k$  ban đầu và không thể thích nghi với các thay đổi trong dữ liệu, e.g. khi tập dữ liệu biết thêm thuộc tính thật của các nhän không biết ban đầu. Để cải thiện điều này,  $\mathcal{L}_{EPR}$  được kết hợp với một mô-đun thứ hai dùng để duy trì và cập nhật các ước lượng nhän trong suốt quá trình huấn luyện. Mô-đun này được gọi là bộ phận ước lượng nhän. Mô hình lúc này sẽ kết hợp việc huấn luyện bộ phận phân loại ảnh và mô-đun mới. Kỹ thuật này được gọi là ước lượng nhän trực tuyến điều chuẩn (ROLE).

Gọi ma trận các nhän ước lượng là  $\tilde{Y} = [\tilde{y}_{ni}]_{n \in \{1, \dots, N\}, i \in \{1, \dots, L\}} \in [0, 1]^{N \times L}$ , ma trận nhän thực  $Y \in [0, 1]^{N \times L}$  và ma trận dự đoán  $F \in [0, 1]^{N \times L}$ . Với mỗi lô  $B$ , ma trận các nhän ước lượng là  $\tilde{Y}_B = [\tilde{y}_{ni}]_{n \in B, i \in \{1, \dots, L\}} \in [0, 1]^{|B| \times L}$ .

Cuối cùng, chúng ta xây dựng bộ ước lượng nhãn  $g(x_n; \phi)$  sao cho  $\tilde{y}_n = g(x_n; \phi)$ . Hoàn hảo nhất là khi  $\tilde{Y} = Y$ , nên chúng ta sử dụng hàm kích hoạt cuối của bộ phân lớp làm hàm  $g$  và khởi tạo tham số  $\phi$  bằng cách áp dụng hàm ngược  $g^{-1}$  lên ma trận  $Y$ :  $\phi = g^{-1}(Y)$ .

Để khuyến khích các dự đoán của bộ phân loại ảnh  $F_B$  khớp với các nhãn ước lượng  $\tilde{Y}_B$ , chúng ta cần tối ưu hàm lỗi:

$$\frac{1}{|B|} \sum_{n \in B} \mathcal{L}_{BCE}(\mathbf{f}_n, sg(\tilde{y}_n)) \quad (2.13)$$

trong đó  $sg$  là hàm dừng gradient [5], ngăn chặn gradient lan truyền ngược lại. Điều này có nghĩa là khi thực hiện lan truyền ngược, các gradient sẽ không được tính toán và cập nhật cho các tham số  $\phi$ .

Kết hợp hàm lỗi này với  $\mathcal{L}_{EPR}(\mathbf{F}_B, \mathbf{Z}_B)$  - hàm dùng để đẩy  $F_B$  dự đoán chính xác các nhãn dương đã biết và đảm bảo số lượng nhãn dương kỳ vọng  $k$  cho mỗi ảnh, ta được hàm mất mát trung gian:

$$\mathcal{L}'(F_B | \tilde{Y}_B) = \frac{1}{|B|} \sum_{n \in B} \mathcal{L}_{BCE}(\mathbf{f}_n, sg(\tilde{y}_n)) + \mathcal{L}_{EPR}(\mathbf{F}_B, \mathbf{Z}_B) \quad (2.14)$$

Vì  $Z$  cố định trong suốt quá trình huấn luyện, nên về trái của hàm đã bỏ qua sự phụ thuộc vào  $Z_B$ .

Với mục tiêu là huấn luyện đồng thời bộ phận ước lượng nhãn  $g(\cdot; \phi)$  và bộ phận phân loại ảnh  $f(\cdot; \theta)$ . Đầu tiên, hàm mất mát trên dùng để cập nhật  $\theta$  trong khi cố định  $\phi$  bằng hàm  $sg$ . Sau đó đổi các đối số trong phương trình trên thành  $\mathcal{L}'(\tilde{Y}_B | F_B)$ . Đây là hàm mất mát tương tự cho phép cập nhật  $\phi$  trong khi cố định  $\theta$ . Hàm mất mát cuối cùng là:

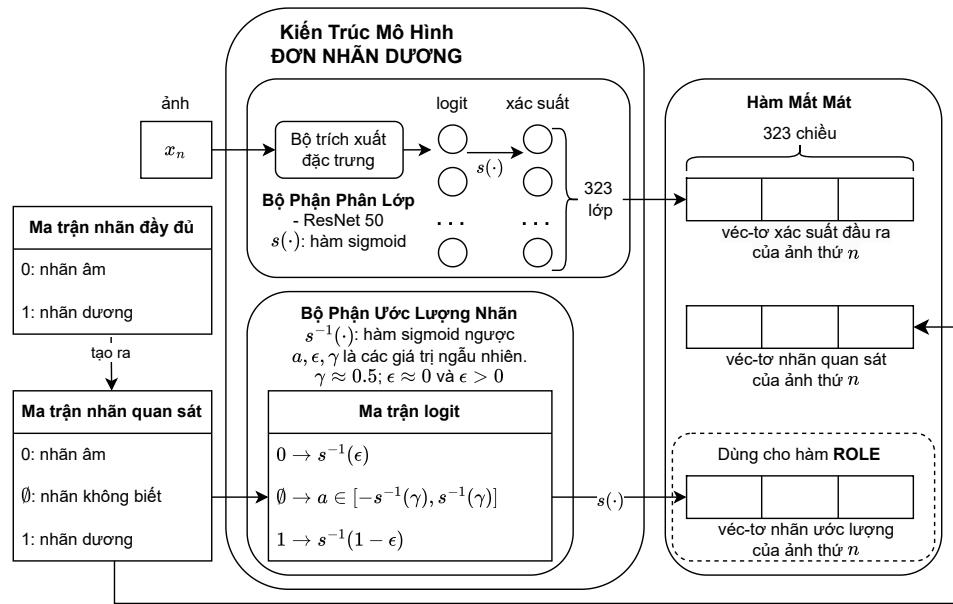
$$\mathcal{L}_{ROLE}(F_B, \tilde{Y}_B) = \frac{\mathcal{L}'(F_B | \tilde{Y}_B) + \mathcal{L}'(\tilde{Y}_B | F_B)}{2} \quad (2.15)$$

qua đó  $F_B$  và  $\tilde{Y}_B$  được cập nhật đồng thời.

Trong thực nghiệm, ma trận nhãn đầy đủ  $F_B$  được thay bằng ma trận

nhãn quan sát  $Z_B$ , bảo đảm rằng hàm măt mát này được xây dựng cho trường hợp Đơn nhãn dương.

### 2.6.3 Kiến trúc mô hình



**Hình 2.11:** Mô hình học đa nhãn với Đơn nhãn dương.

Mô hình Đơn nhãn dương gồm 2 bộ phận là bộ phận phân lớp và bộ phận ước lượng nhãn (dùng cho hàm măt măt ROLE).

Bộ phận phân lớp các tác giả đã sử dụng là ResNet 50, gồm bộ trích xuất đặc trưng và bộ phân loại tuyến tính. Bộ phân loại tuyến tính dùng hàm sigmoid  $s(x) = \frac{1}{1+e^{-x}}$  làm kích hoạt ở lớp cuối. Các giá trị đầu ra của hàm  $s(x_{ni})$  thuộc  $(0; 1)$ , được xem là xác suất để ảnh  $x_n$  chứa nhãn thứ  $i$ . Ta mong muốn  $s(x_{ni}) \approx 1$  nếu  $x_{ni}$  chứa nhãn thứ  $i$ ,  $s(x_{ni}) \approx 0$  nếu  $x_{ni}$  không chứa nhãn thứ  $i$  và  $s(x_{ni}) \approx 0.5$  nếu không biết.

Dùng các giá trị xác suất 0, 1 và 0.5 để làm ma trận nhãn ước lượng ban đầu tương ứng với trạng thái các nhãn âm, dương và không biết. Sau đó, cho các giá trị này đi qua hàm kích hoạt sigmoid ngược  $s^{-1}(y) = \ln\left(\frac{y}{1-y}\right)$  để thu được các logit. Ma trận logit này được dùng làm tham số khởi tạo cho bộ phận ước lượng nhãn. Để ý rằng  $\lim_{y \rightarrow 1} s^{-1}(y) = +\infty$  và

$\lim_{y \rightarrow 0} s^{-1}(y) = -\infty$ . Để tránh các trường hợp này, chúng ta sẽ giảm nhẹ xác xuất nhãn dương từ 1 thành  $1 - \epsilon$  và tăng nhẹ xác suất các nhãn âm từ 0 thành  $\epsilon$  trước khi qua hàm kích hoạt ngược ( $\epsilon$  là hệ số làm mịn nhãn). Ngoài ra, 0.5 cũng đổi thành một số thực  $\gamma \approx 0.5$ . Các giá trị của các logit tương ứng với nhãn không biết sẽ dao động trong  $[-s^{-1}(\gamma), s^{-1}(\gamma)]$ , tức dao động quanh số 0. Quá trình tìm hàm  $s^{-1}$  được trình bày ở [3.2.3](#).

## 2.6.4 Các chế độ học

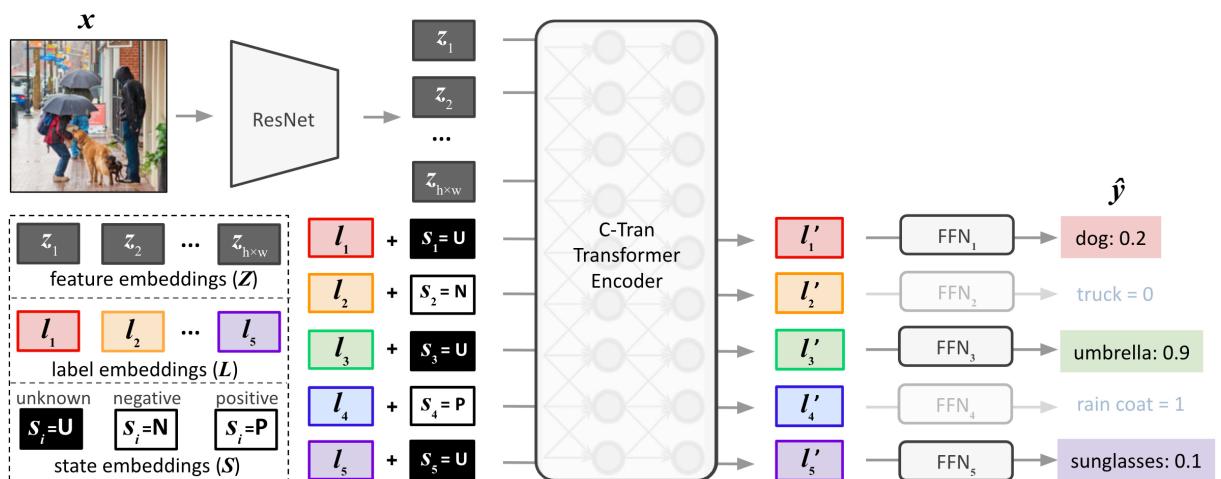
Các tác giả đã thử nghiệm trên 3 chế độ học khác nhau bao gồm: *tuyến tính*, *dầu cuối* và *chuyển giao*. Học tuyến tính là ở bộ phận phân lớp, mô hình chỉ đào tạo bộ phận phân lớp tuyến tính trên bộ trích xuất đặc trưng cố định. Đầu cuối là mô hình đào tạo cả bộ trích xuất đặc trưng và bộ phân loại tuyến tính của bộ phận phân lớp. Cuối cùng, học chuyển giao là mô hình sẽ đóng băng trọng số của bộ trích xuất đặc trưng cho các giai đoạn đào tạo ban đầu và sau đó tinh chỉnh toàn bộ mạng từ đầu đến cuối cho các giai đoạn còn lại. Hai chế độ học tuyến tính và học chuyển giao có bộ trích xuất đặc trưng bị đóng băng được khởi tạo ban đầu là mô hình được huấn luyện sẵn trên tập dữ liệu ImageNet.

Học chuyển giao thường sẽ cho kết quả tốt hơn đối với tập dữ liệu nhỏ và trung bình do sử dụng mô hình đã được huấn luyện trên nhiều loại dữ liệu khác nhau từ tập dữ liệu ImageNet. Đây là tập dữ liệu lớn hơn nhiều so với tập thử nghiệm. Điều này giúp mô hình có khả năng tổng quát hóa tốt hơn, từ đó hoạt động hiệu quả trên các tập dữ liệu mới mà không bị quá khớp với dữ liệu huấn luyện ban đầu. Từ đó, cũng tiết kiệm thời gian và tài nguyên huấn luyện.

## 2.7 Phân loại ảnh đa nhãn bằng Transformer (C-Tran)

Mô hình C-Tran được đề xuất để thúc đẩy việc khai thác sự phụ thuộc phức tạp giữa các đặc trưng và nhãn của hình ảnh. Cách tiếp cận chính của C-Tran bao gồm việc đào tạo lớp Encoder của mô hình Transformer để dự đoán một tập hợp nhãn mục tiêu từ đầu vào bao gồm: các nhãn bị che và các đặc trưng hình ảnh được trích xuất từ mạng nơ-ron tích chập. Thành phần quan trọng của C-Tran là sử dụng phương pháp che nhãn trong quá trình đào tạo bằng sơ đồ mã hóa bậc ba thể hiện các trạng thái của nhãn là dương, âm hoặc không xác định. Và những phương pháp được đề xuất trong mô hình C-Tran cho thấy tính hiệu quả thông qua kết quả về hiệu suất của mô hình được cải thiện trên các tập dữ liệu đầy thách thức.

### 2.7.1 Mô hình



**Hình 2.12:** Mô hình phân loại ảnh đa nhãn bằng Transformer [9].

## 2.7.2 Các thành phần chính của mô hình

### 1) Nhúng đặc trưng $Z$

Cho hình ảnh đầu vào  $x \in \mathbb{R}^{H \times W \times 3}$ , đặc trưng được trích xuất ở đầu ra là một ma trận  $Z \in \mathbb{R}^{h \times w \times d}$ , trong đó,  $h$ ,  $w$  và  $d$  là chiều cao, chiều rộng và số kênh tương ứng. Xét mỗi véc-tơ  $z_i \in \mathbb{R}^d$  từ  $Z$ , với  $i$  trong khoảng 1 đến  $P$  (trong đó  $P = h \times w$ ), là đại diện cho một vùng ánh xạ từ các mảng trong không gian gốc.

### 2) Nhúng nhãn $L$

Với mỗi hình ảnh, chúng ta thu được một tập hợp các nhãn được nhúng là  $L = \{l_1, l_2, \dots, l_l\}$ ,  $l_i \in \mathbb{R}^d$ , đại diện cho các nhãn  $l$  có thể có trong  $y$ . Việc nhúng nhãn được thực hiện thông qua lớp nhúng với kích thước  $d \times l$ .

### 3) Thêm kiến thức về nhãn thông qua nhúng trạng thái $S$

Mỗi nhãn nhúng  $l_i$  được cộng thêm một véc-tơ nhúng “trạng thái”,  $s_i \in \mathbb{R}^d$ :

$$\tilde{l}_i = l_i + s_i \quad (2.16)$$

trong đó  $s_i$  có một trong ba trạng thái có thể xảy ra: không xác định (U), âm (N) hoặc dương (P).

### 4) Mô hình hóa sự tương tác giữa đặc trưng và nhãn bằng Transformer Encoder

Đặt  $H = \{z_1, \dots, z_{h \times w}, \tilde{l}_1, \dots, \tilde{l}_l\}$  là tập hợp các phần nhúng được đưa vào Transformer Encoder. Trong Transformer, độ quan trọng hoặc trọng số của véc-tơ nhúng  $h_j \in H$  với véc-tơ nhúng  $h_i \in H$  được học thông qua cơ chế “tự chú ý”. Trọng số “chú ý”,  $\alpha_{ij}^t$  giữa hai véc-tơ nhúng  $i$  và  $j$  được tính theo cách sau. Đầu tiên, ta tính “hệ số chú ý vô hướng chuẩn hóa”  $\alpha_{ij}$  giữa cặp véc-tơ nhúng  $i$  và  $j$ . Sau khi tính giá trị  $\alpha_{ij}$  cho tất cả các cặp  $i$

và  $j$ , ta tiến hành cập nhật  $h_i$  thành  $h'_i$  bằng cách sử dụng tổng trọng số của tất cả các phần nhúng theo sau là một lớp phi tuyến ReLU:

$$\alpha_{ij} = \text{softmax} \left( \frac{(W^q h_i)^T (W^k h_j)}{\sqrt{d}} \right) \quad (2.17)$$

$$\bar{h}_i = \sum_{j=1}^M \alpha_{ij} W^v h_j \quad (2.18)$$

$$h'_i = \text{ReLU}(\bar{h}_i W^r + b_1) W^o + b_2 \quad (2.19)$$

trong đó  $W^k$  là ma trận trọng số khóa,  $W^q$  là ma trận trọng số truy vấn,  $W^v$  là ma trận trọng số giá trị,  $W^r$  và  $W^o$  là các ma trận biến đổi, và  $b_1$ ,  $b_2$  là các véc-tơ bias. Quy trình cập nhật này được lặp lại cho  $L$  lớp, trong đó các phần nhúng được cập nhật  $h'_i$  được cung cấp để làm đầu vào cho các lớp Transformer Encoder tiếp theo. Các ma trận trọng số được học  $\{W^k, W^q, W^v, W^r, W^o\} \in \mathbb{R}^{d \times d}$  khác nhau giữa các lớp. Đầu ra cuối cùng của bộ mã hóa Transformer sau  $L$  lớp là  $H' = \{z'_1, \dots, z'_{h \times w}, l'_1, \dots, l'_l\}$ .

### 5) Quá trình suy luận để phân loại nhãn

Cuối cùng, sau khi các phụ thuộc đặc trưng và nhãn được mô hình hóa thông các lớp Transformer Encoder, bộ phân loại sẽ đưa ra dự đoán cuối cùng. Mạng chuyển tiếp độc lập ( $\text{FFN}_i$ ) cho lớp nhúng ở cuối là  $l'_i$ .  $\text{FFN}_i$  bao gồm một lớp tuyến tính, trong đó trọng số  $w_i^c$  cho nhãn  $i$  là véc-tơ có kích thước  $1 \times d$ , và  $\sigma$  là hàm số sigmoid:

$$\hat{y}_i = \text{FFN}_i(l'_i) = \sigma((w_i^c \cdot l'_i) + b_i) \quad (2.20)$$

#### 2.7.3 Hàm mất mát

$$L = \sum_{n=1}^{N_{tr}} \mathbb{E}_{p(\mathbf{y}_k)} \left\{ \text{CE} \left( \hat{\mathbf{y}}_u^{(n)}, \mathbf{y}_u^{(n)} \right) \mid \mathbf{y}_k \right\} \quad (2.21)$$

Trong đó,  $y_u$  nhãn chưa xác định,  $y_k$  là nhãn đã biết,  $\mathbb{E}_{p(y_k)}(\cdot | y_k)$  thể hiện kỳ vọng về phân bố xác suất với nhãn đã biết  $y_k$ , và CE là hàm đánh giá độ lỗi Cross Entropy.

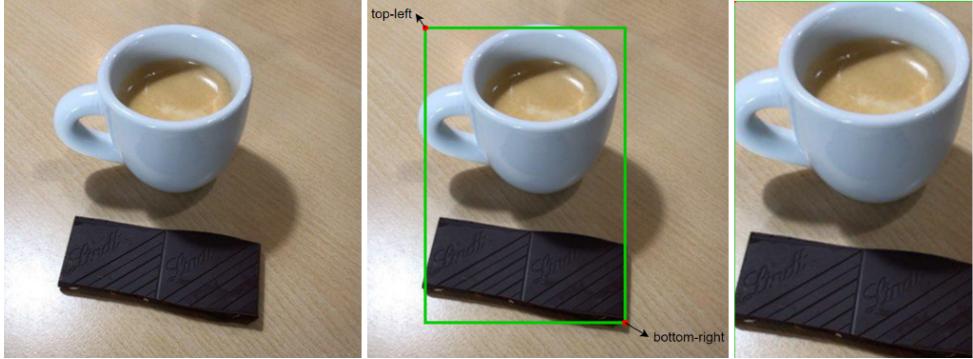
# Chương 3

## Các mô hình, phương pháp để xuất

*Trong chương này sẽ trình bày các cách phát triển, cải tiến, đồng thời thử nghiệm kết hợp các mô hình học sâu từ các công trình trước đó, chủ yếu là C-Tran, Học Da Nhẫn Chỉ Từ 1 Nhẫn Dương (mô hình Đơn nhẫn dương).*

### 3.1 Kỹ thuật cải tiến dữ liệu

Để giải quyết các vấn đề kỹ thuật liên quan đến hình ảnh, chúng tôi để xuất cắt ảnh nhằm bỏ bớt nhiều cảnh nền nhất có thể và chỉ giữ lại các đối tượng chính trong mỗi bức ảnh. Để cắt ảnh cần xác định vùng cắt. Vùng cắt được xác định bởi cặp điểm trên cùng bên trái (đặt là top-left) và điểm dưới cùng bên phải (đặt là bottom-right). Các ảnh sau khi cắt sẽ được dùng làm dữ liệu cho quá trình huấn luyện của mô hình.



**Hình 3.1:** Từ trái qua phải gồm: ảnh gốc, xác định vùng cắt, ảnh sau khi cắt.

Cặp điểm top-left và bottom-right được tạo bởi các điểm tạo thành phân vùng hoặc các điểm trên cùng bên trái và điểm dưới cùng bên phải của khung chứa của tất cả đối tượng chính trong ảnh. Chia tọa độ của tất cả điểm này thành 2 tập  $X_s$  và  $Y_s$ . Trong đó,  $X_s$  chứa các giá trị hoành độ,  $Y_s$  chứa các giá trị tung độ. Lúc này, hoành độ và tung độ của điểm top-left là các giá trị nhỏ nhất trong tập  $X_s$  và  $Y_s$ ; hoành độ và tung độ của điểm bottom-right là các giá trị lớn nhất trong tập  $X_s$  và  $Y_s$  (hoành độ, tung độ có thể là của 2 điểm khác nhau). Do phân vùng bao sát với đối tượng hơn khung chứa nên các ảnh thu được sau khi cắt bằng phân vùng sẽ có chất lượng cao hơn (đầy đủ đối tượng và hạn chế ảnh thừa). Thuật toán xác định tọa độ của điểm top-left và bottom-right cụ thể như sau:

---

#### Algorithm 1 Tìm top-left bottom-right

---

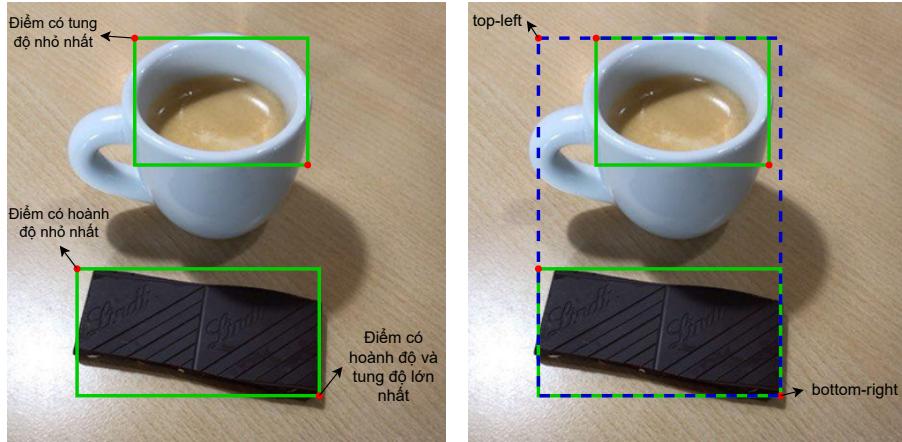
```

1: procedure FINDCOORDINATE( $X_s, Y_s$ )
2:    $minX \leftarrow$  Minimum value of  $X_s$ 
3:    $minY \leftarrow$  Minimum value of  $Y_s$ 
4:    $maxX \leftarrow$  Maximum value of  $X_s$ 
5:    $maxY \leftarrow$  Maximum value of  $Y_s$ 
6:   return ( $minX, minY, maxX, maxY$ )

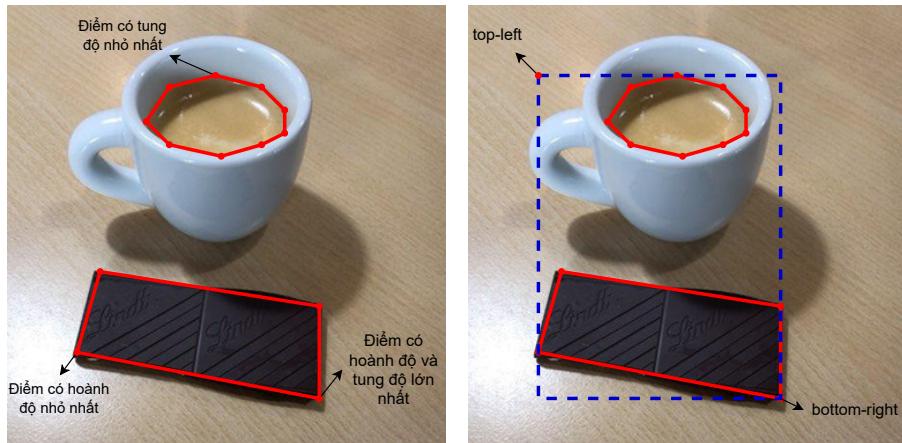
```

---

Trong đó,  $(minX, minY)$  là tọa độ của top-left và  $(maxX, maxY)$  là tọa độ của bottom-right.



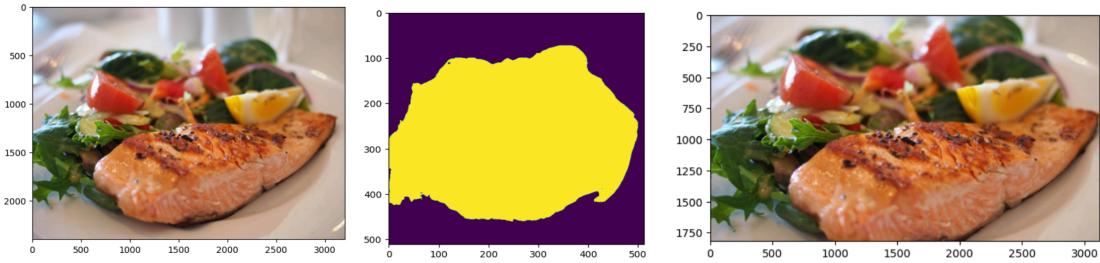
**Hình 3.2:** Cắt ảnh bằng khung chứa



**Hình 3.3:** Cắt ảnh bằng phân vùng

Đối với các tập dữ liệu lớn thường có sẵn thông tin phân vùng và khung chứa của các vật thể trong mỗi bức ảnh. Đối với các tập dữ liệu không có các thông tin này, chúng tôi đề xuất sử dụng các mô hình được huấn luyện sẵn để lấy hai thông tin này. Ví dụ sử dụng mô hình U-Net để lấy các phân vùng của mỗi bức ảnh. Cụ thể cách lấy các thông tin về phân vùng của U-Net.

Sau khi cho hình ảnh đi qua mạng U-Net, ta sẽ có được một đầu ra chứa giá trị đánh dấu vật thể trong hình. Từ đó, ta sẽ tiến hành xử lý đầu ra này để có thể lấy được các tọa độ [top-left, bottom-right] để phục vụ cho việc cắt ảnh.



**Hình 3.4:** Bên Trái: Ảnh gốc, Ở giữa: Ảnh gốc sau khi qua mạng U-Net, Bên phải: Kết quả sau khi cắt ảnh.

Các bước xử lý để lấy thông tin các điểm [top-left, bottom-right] sẽ được mô tả trong thuật toán bên dưới.

---

### Algorithm 2 Tìm top-left bottom-right trong mặt nạ ảnh

---

```

procedure FINDCOORDINATEINMASK(mask, width, height)
    array2d  $\leftarrow$  Transpose and reshape mask
    (Xs, Ys)  $\leftarrow$  Find indices where value is 1 in array2d
    Xs  $\leftarrow$  Xs  $\times$  width/512
    Ys  $\leftarrow$  Ys  $\times$  height/512
    Call FINDCOORDINATE procedure (Algorithm 1)
    result  $\leftarrow$  FINDCOORDINATE(Xs, Ys)
    return result

```

---

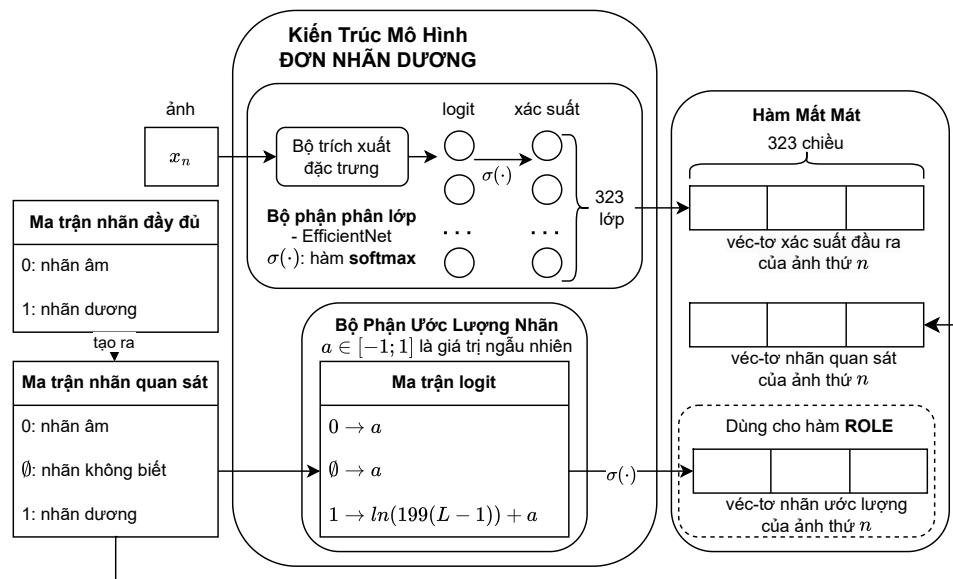
Trong đó, đầu vào: *mask* là ma trận 3D đầu ra của hình ảnh sau khi đi qua mạng U-Net; *width*, *height* là các kích thước của hình ảnh lúc ban đầu. Đầu ra: (*minX*, *minY*) là tọa độ của điểm top-left, (*maxX*, *maxY*) là tọa độ của điểm bottom-right.

Để đánh giá hiệu quả việc lấy thông tin phân vùng của vật thể trong hình ảnh khi sử dụng mô hình U-Net được huấn luyện sẵn, chúng tôi tiến hành tính IoU cho tập dữ liệu huấn luyện [4.1.2](#) và đánh giá [4.1.2](#), các kết quả thu được lần lượt là 0.419 và 0.506. Những kết quả này cho thấy rằng, U-Net phù hợp ở mức tạm ổn so với tập dữ liệu đánh giá nhưng ở mức khá thấp so với tập huấn luyện. Do đó, việc sử dụng mô hình U-Net được huấn luyện sẵn chưa thật sự tốt trên tập dữ liệu của chúng tôi. Trong điều

kiện cho phép, chúng tôi đề xuất việc huấn luyện U-Net trên tập dữ liệu có nhiều nét tương đồng với tập dữ liệu của chúng tôi để sử dụng, điều này sẽ giúp cho U-Net hoạt động hiệu quả hơn.

## 3.2 Mô hình Đơn nhãn dương

### 3.2.1 Tóm tắt các đề xuất



**Hình 3.5:** Mô hình học đa nhãn với đơn nhãn dương.

Bên cạnh ResNet 50, mạng được dùng cho bộ phận lớp được thử nghiệm trong khóa luận được thay bởi EfficientNet 3.2.4. Với tính phù hợp của cài đặt đơn nhãn dương trong bối cảnh giả sử âm, chúng tôi thử nghiệm hàm kích hoạt khác là softmax. Đồng thời sử dụng hàm softmax ngược để khởi ma trận logit. Cách khởi tạo ở sơ đồ trên là kết quả nghiên cứu ở 3.2.3. Ngoài ra, chúng tôi còn thử nghiệm thêm các hàm mất mát khác phù hợp với cài đặt của đơn nhãn dương 3.2.2.

### 3.2.2 Các hàm mất mát

Chúng tôi bổ sung thêm các hàm mất mát thường được dùng cho các bài toán phân lớp khác gồm:

Hàm mất mát Squared Hinge [10] với  $m$  là biên an toàn:

$$\mathcal{L}_{HI}(\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L [\max(0, m - \mathbf{f}_n \cdot \mathbf{z}_n)]^2 \quad (3.1)$$

Hàm mất mát này phạt nhiều hơn cho các dự đoán sai lầm lớn so với hàm mất mát Hinge truyền thống vì giá trị mất mát tăng theo bình phương. Tuy nhiên, nó vẫn giữ nguyên các đặc điểm của hàm mất mát Hinge trong việc khuyến khích phân tách tuyến tính lớn giữa các lớp.

Hàm mất mát Huber [15]:

$$\mathcal{L}_{HU}(\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L \begin{cases} \frac{1}{2} (\mathbf{f}_n - \mathbf{z}_n)^2, & \text{nếu } \mathbf{f}_n - \mathbf{z}_n \leq \delta \\ \delta |\mathbf{f}_n - \mathbf{z}_n| - \frac{1}{2}\delta, & \text{ngược lại} \end{cases} \quad (3.2)$$

Chúng tôi thí nghiệm với  $\delta = 1$ .

Hàm mất mát này kết hợp giữa sai số bình phương trung bình (MSE) và sai số tuyệt đối trung bình (MAE) để tận dụng những ưu điểm của cả hai, làm cho nó trở nên ít nhạy cảm hơn với các giá trị ngoại lai so với MSE, nhưng vẫn tròn trịa hơn so với MAE.

Để tính sự khác biệt giữa hai phân phối xác suất  $p$  (mong muốn) và  $q$  (dự đoán), bên cạnh BCE thi phân kỳ Kullback-Leibler (KL Divergence) [8] cũng có sự tương đồng.

KL Divergence = Cross Entropy – Entropy

$$\begin{aligned}
D_{KL}(p \parallel q) &= H(p, q) - H(p) \\
&= - \sum_i p_i \log q_i - \left( - \sum_i p_i \log p_i \right) \\
&= \sum_i p_i \log p_i - \sum_i p_i \log q_i \\
&= \sum_i p_i \log \frac{p_i}{q_i}
\end{aligned} \tag{3.3}$$

Dùng  $D_{KL}(p \parallel q)$  như hàm mất mát cho bài toán phân lớp đa nhãn, với  $p$  mong muốn  $\in \{0, 1\}$ :

$$L_{KL}(f_n, z_n) = \frac{1}{L} \sum_{i=1}^L \left[ \mathbb{I}_{[z_{ni}=1]} \log \left( \frac{1}{f_{ni}} \right) + \mathbb{I}_{[z_{ni}=0]} \log \left( \frac{1}{1-f_{ni}} \right) \right] \tag{3.4}$$

Lúc này, dễ dàng nhận ra  $L_{KL} = L_{BCE}$ .

Hàm mất mát Focal [12]:

$$\mathcal{L}_{FO}(\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L \left[ \alpha_i (1 - f_{ni})^\gamma \mathbb{I}_{[z_{ni}=1]} \log(f_{ni}) + \alpha_i f_{ni}^\gamma \mathbb{I}_{[z_{ni}=0]} \log(1 - f_{ni}) \right] \tag{3.5}$$

Trong đó,  $\alpha_i = \frac{1}{fr_i + \epsilon}$ ,  $fr_i$  là tần suất xuất hiện của nhãn thứ  $i$ ,  $\epsilon$  là một số rất nhỏ để tránh mẫu bằng 0. Để thuận tiện,  $\alpha_i$  được xem như siêu tham số  $\alpha$  cho nhãn dương và  $1 - \alpha$  cho nhãn âm. Trong thực nghiệm, chúng tôi sử dụng bộ tham số tốt nhất trong công trình [12] là  $\gamma = 2$  và  $\alpha = 0.25$ .

Hàm mất mát Focal có khả năng giải quyết vấn đề mất cân bằng nhãn, bằng cách tập trung hơn vào việc học từ các nhãn hiếm hơn hoặc quan trọng hơn (các nhãn bị phân loại sai nghiêm trọng). Tham số  $\alpha_i$  cho phép các nhãn xuất hiện ít hơn thì có giá trị tác động tới hàm mất mát lớn hơn. Vì  $\gamma$  càng lớn thì giá trị lỗi ở các nhãn dễ phân biệt sẽ càng nhỏ và giá trị của các nhãn khó phân biệt sẽ càng đóng góp nhiều hơn vào tổng độ lỗi của mô hình.

Hàm mất mát Asymmetric [13]:

$$L_{\text{ASL}}(f_n, z_n) = -\frac{1}{L} \sum_{i=1}^L \left[ (1 - f_{ni})^{\gamma^+} \mathbb{I}_{[z_{ni}=1]} \log(f_{ni}) + f_{ni\_m}^{\gamma^-} \mathbb{I}_{[z_{ni}=0]} \log(1 - f_{ni\_m}) \right] \quad (3.6)$$

Biết  $f_{ni\_m} = \max(f_{ni} - m, 0)$ . Ở thực nghiệm, chúng tôi sử dụng lại bộ tham số tốt nhất đã được thử nghiệm trong công trình [13] là  $\gamma^+ = 0; \gamma^- = 4; m = 0.05$ .

Hàm mất mát Asymmetric là sự cải tiến của hàm Focal. Hàm Asymmetric sử dụng các tham số  $\gamma$  khác nhau cho các mẫu dương và mẫu âm. Thông qua đó, mô hình có thể kiểm soát tốt hơn đối với sự đóng góp của các mẫu dương và âm vào hàm mất mát. Điều này giúp mạng tìm hiểu các tính năng có ý nghĩa từ các mẫu chiếm thiểu số. Và nó cũng giảm thiểu tác động của các mẫu dễ dàng phân loại. Tuy nhiên, vì mức độ mất cân bằng trong phân loại đa nhãn rất cao. Vì thế, Asymmetric loại bỏ hoàn toàn các mẫu âm khi xác suất đầu ra của chúng thấp (xem chúng đã dự đoán đúng) bằng cơ chế nếu xác suất dự đoán  $f_{ni}$  bé hơn ngưỡng  $m$  thì xem như  $f_{ni} = 0$ .

Các công thức hàm mất trên được thay đổi để phù hợp với bài toán phân lớp đa nhãn và véc-tơ đầu ra tương ứng. Chúng tôi sẽ thử nghiệm các hàm mất mát này cho trường hợp Đơn nhãn dương. Điều này có nghĩa là ma trận nhãn quan sát chỉ có một nhãn dương, còn lại là nhãn không được quan sát. Chúng tôi sẽ xem các nhãn không biết như nhãn âm. Đây chính là cài đặt của hàm mất mát giả sử nhãn âm nhưng được dùng cho các hàm mất mới thêm vào.

### 3.2.3 Hàm kích hoạt và bộ ước lượng nhãn

Mô hình ban đầu sử dụng hàm sigmoid  $s(x)$  làm hàm kích hoạt cho đầu ra của bộ phân lớp. Để khởi tạo ma trận tham số cho bộ ước lượng

nhãn, cần sử dụng hàm ngược  $s^{-1}(x)$  tương ứng:

$$\begin{aligned} s(x) = y = \frac{1}{1 + e^{-x}} &\Leftrightarrow \frac{1}{y} = 1 + e^{-x} \Leftrightarrow e^{-x} = \frac{1}{y} - 1 \\ &\Leftrightarrow s^{-1}(y) = x = -\ln\left(\frac{1}{y} - 1\right) = -\ln\left(\frac{1-y}{y}\right) = \ln\left(\frac{y}{1-y}\right) \end{aligned} \quad (3.7)$$

Trong các mô hình học đa nhãn thường không sử dụng hàm kích hoạt softmax vì hàm softmax đảm bảo tổng xác suất của tất cả các nhãn bằng 1, nên kích thước tập nhãn càng lớn thì xác suất càng bị chia nhỏ ra. Điều này dẫn đến việc một bức ảnh chứa nhiều nhãn dương nhưng do xác suất bị chia nhỏ mà không đạt ngưỡng yêu cầu để được phân lớp. Tuy nhiên, trong tập quan sát của mô hình Đơn nhãn dương trong cài đặt **giả sử nhãn âm** lại chỉ có 1 nhãn dương các nhãn còn lại xem như nhãn âm giống với bài toán phân lớp đơn nhãn, nên hàm kích hoạt softmax phù hợp với mô hình này. Chúng tôi sẽ xây dựng hàm softmax ngược  $\sigma^{-1}(y)$  để khởi tạo tham số cho bộ phận ước lượng nhãn của mô hình. Hàm softmax gốc  $\sigma(x)$  phụ thuộc vào giá trị đầu vào  $x_i, \forall i \in [1, L]$  của tất cả các nhãn ( $L$  là số lượng lớp).

$$y_i = \sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^L e^{x_j}} \quad (3.8)$$

Do đó để xây dựng được hàm softmax ngược  $x_i$  theo  $y_i$  một cách tổng quát là tương đối khó. Để dễ dàng hơn, chúng tôi sẽ cụ thể hóa hàm softmax theo bối cảnh của bài toán Đơn nhãn dương, tức mỗi bức ảnh chỉ có 1 nhãn dương, còn lại là nhãn không biết. Cho 1 bức ảnh có véc-tơ nhãn quan sát  $z$  có  $L$  chiều như sau:

$$(1 \ \emptyset \ \emptyset \ \emptyset \ \dots \ \emptyset)$$

Chuyển tập giá trị  $[1; 0; \emptyset]$  lần lượt ứng với nhãn dương, nhãn âm và không biết sang tập giá trị xác suất tương ứng  $[1; 0; 0.5]$  phù hợp với đầu ra của hàm softmax. Lúc này, nhãn không biết cũng được xem như nhãn

âm để đảm bảo  $\sum_{i=1}^L y_i = 1$ .

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Với  $L - 1$  là số nhãn âm, áp dụng kỹ thuật làm mượt nhãn cho véc-tơ xác suất trên để tránh các giá trị  $x_i$  không xác định, ta được véc-tơ nhãn mới:

$$\begin{pmatrix} 1 - \epsilon & \frac{\epsilon}{L-1} & \frac{\epsilon}{L-1} & \frac{\epsilon}{L-1} & \dots & \frac{\epsilon}{L-1} \end{pmatrix}$$

Gọi  $x^+$  và  $x^-$  lần lượt là logit thỏa  $\sigma(x^+) = 1 - \epsilon$  và  $\sigma(x^-) = \frac{\epsilon}{L-1}$ . Thực nghiệm với  $\epsilon = 0.005$ . Ta có:

$$\begin{cases} \frac{e^{x^+}}{e^{x^+} + (L-1)e^{x^-}} = 0.995 \\ \frac{e^{x^-}}{e^{x^+} + (L-1)e^{x^-}} = \frac{0.005}{L-1} \end{cases} \Leftrightarrow 0.005e^{x^+} = 0.995(L-1)e^{x^-} \quad (3.9)$$

$$\Leftrightarrow e^{x^+} = 199(L-1)e^{x^-} \Leftrightarrow x^+ = \ln(199(L-1)) + x^-$$

Vậy logit  $x^+$  phụ thuộc vào logit  $x^-$ . Trong thực tế, chúng tôi sẽ lấy ngẫu nhiên giá trị logit  $x^- \in [-1, 1]$  cho các nhãn không biết và nhãn âm để tính  $x^+$  cho nhãn dương.

### 3.2.4 Bộ phân lớp

Chúng tôi sử dụng các mạng trích xuất đặc trưng mới EfficientNet 2.3 để thay thế cho mạng trích xuất đặc trưng cũ là ResNet 2.1. Bởi vì, các mô hình EfficientNet là sự tối ưu về lượng tham số mà vẫn cải thiện tốt về độ chính xác của mô hình so với ResNet 50. Ngoài ra, lớp được kết nối đầy đủ của bộ phân lớp tuyến tính có số nút đầu ra được thay đổi bằng với số lượng nhãn  $L$  cần phân lớp. Với thay đổi này, chúng tôi mong muốn mô hình sẽ đạt được hiệu suất cao trong quá trình học và dự đoán.

```
1 Linear(in_features=2560, out_features=L, bias=True)
```

### 3.2.5 Giá trị trung bình và độ lệch chuẩn

Các mô hình ở các bài báo khoa học thường được thực nghiệm trên tập dữ liệu nổi tiếng ImageNet. Vì thế, tập đặc trưng ảnh đầu vào được chuẩn hóa bằng 2 véc-tơ trung bình và véc-tơ độ lệch chuẩn tương ứng:

```
1 imangenet_mean = [0.5957, 0.5094, 0.4278]
2 imangenet_std = [0.2104, 0.2192, 0.2290]
```

Mỗi giá trị trong mỗi véc-tơ tương ứng với mỗi kênh màu RGB của các bức ảnh. Chúng tôi tính toán lại các giá trị trung bình và độ lệch chuẩn của các đặc trưng ảnh đầu vào của tập dữ liệu mới cho việc chuẩn hóa. Đầu tiên, chúng tôi chuyển tất cả ảnh về cùng kích thước. Cụ thể là  $448 \times 448$  như tập dữ liệu ImageNet. Tiếp theo, chia tập dữ liệu thành từng lô. Dối với mỗi lô, định hình lại các hình ảnh thành (kích thước lô, kênh màu, chiều rộng  $\times$  chiều cao). Tính giá trị trung bình và độ lệch chuẩn của từng bức ảnh rồi cộng dồn chúng theo mỗi kênh màu. Chia giá trị trung bình tổng và độ lệch chuẩn cho tổng số hình ảnh để có giá trị trung bình và độ lệch chuẩn cuối cùng.

---

**Algorithm 3** Tính toán lại véc-tơ trung bình và véc-tơ độ lệch chuẩn

---

```
procedure CALCULATEMEANSTD(data_loader)
    mean ← 0
    std ← 0
    total_images_count ← 0
    for all batch ∈ data_loader do
        (images, batch_size) ← batch
        reshape(images, (batch_size, channels, width × height))
        mean_batch ← mean(images, 2).sum(0)
        std_batch ← std(images, 2).sum(0)
        mean ← mean + mean_batch
        std ← std + std_batch
        total_images_count ← total_images_count + batch_size
    mean ← mean/total_images_count
    std ← std/total_images_count
    return (mean, std)
```

---

### 3.3 Phân loại ảnh đa nhãn bằng Transformer (C-Tran)

Mô hình gốc của C-Tran đạt kết quả tương đối cao đối với tập dữ liệu mà chúng thử nghiệm. Do đó, mục tiêu của những thay đổi mà chúng tôi muốn hướng đến là giảm đi lượng tham số của mô hình, nhưng vẫn giữ được những kết quả cao ban đầu.

#### 3.3.1 Thay đổi mạng trích xuất đặc trưng cho quá trình trích xuất đặc trưng từ hình ảnh

Chúng tôi sử dụng các mạng trích xuất đặc trưng MobileNet V2 [2.2](#) và EfficientNet B0 [2.3](#) để thay thế cho mạng trích xuất đặc trưng gốc là ResNet 101 [2.1](#). Bởi vì, lượng tham số của ResNet 101 lớn gấp nhiều lần so với 2 mạng trích xuất đặc trưng mà chúng tôi thay thế, và đặc biệt là kiến trúc của 2 mạng trích xuất đặc trưng MobileNet V2 và EfficientNet B0 đều được cải tiến hơn so với ResNet 101. Do đó, đối với thay đổi này, chúng tôi mong muốn là mô hình sẽ có thể giảm đi một lượng lớn tham số nhưng vẫn đạt được hiệu suất cao trong quá trình dự đoán.

#### 3.3.2 Một số thay đổi khác

**Thay đổi phương thức thêm đặc trưng nhúng:** Thay vì cộng véc-tơ nhúng nhãn với véc-tơ nhúng trạng thái, chúng tôi tiến hành nhân véc-tơ nhúng nhãn với véc-tơ nhúng trạng thái.

Với nhãn nhúng  $l_i$ , chúng ta nhân thêm một véc-tơ nhúng “trạng thái”,  $s_i \in \mathbb{R}^d$ :

$$\tilde{l}_i = l_i \times s_i \quad (3.10)$$

trong đó  $s_i$  có một trong ba trạng thái có thể xảy ra: không xác định (U), âm (N) hoặc dương (P).

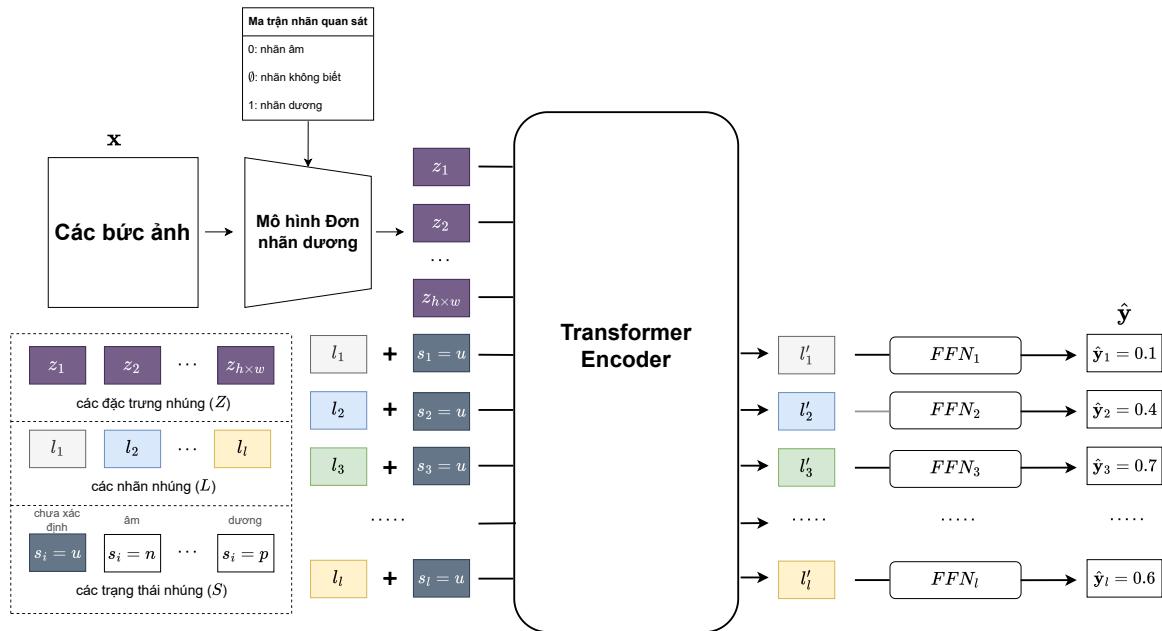
**Thay đổi số lượng lớp encoder:** Chúng tôi thay đổi số lượng lớp encoder của mô hình thành 2 hoặc 4 so với 3 như của mô hình gốc với mong muốn mô hình vẫn có thể học được tốt được sự tương tác giữa nhãn và đặc trưng.

**Thay đổi hàm kích hoạt:** Để tính xác suất của đầu ra thì mô hình gốc dùng hàm kích hoạt sigmoid, chúng tôi tiến hành thay đổi hàm sigmoid thành hàm softmax.

**Không sử dụng phương pháp thêm thông tin trạng thái cho nhãn:** Chúng tôi tiến hành giữ nguyên véc-tơ nhúng nhãn  $l_i$  để đưa vào lớp Encoder chứ không hề thêm bất kì thông tin về "trạng thái" nào. Điều này nhằm đánh giá khả năng liệu mô hình có còn phân loại tốt hay không khi mất đi thông tin về "trạng thái" của nhãn.

### 3.4 Mô hình kết hợp

Chúng tôi thử nghiệm kết hợp giữa 2 mô hình là Đơn nhãn dương và C-Tran với mong muốn kết hợp giữa sự đơn giản hóa dữ liệu nhãn đầu vào và khả năng học mỗi liên hệ giữa các đặc trưng, cũng như giữa các nhãn của C-Tran. Cụ thể, chúng tôi áp dụng cấu hình Đơn nhãn dương, tức là ở mỗi véc-tơ nhãn chỉ giữ lại một nhãn dương, các nhãn còn lại là không biết trong tập nhãn vào C-Tran. Vì ma trận nhãn đã cố định trạng thái của các nhãn, nên ma trận trạng thái nhúng  $S$  trong kiến trúc C-Tran khi kết hợp chúng tôi đều đặt là trạng thái không xác định. Lúc này, mô hình kết hợp sẽ tính độ lỗi cho toàn bộ nhãn thay vì chỉ tính độ lỗi trên các nhãn được nhúng là chưa xác định.Thêm vào đó, chúng tôi thay đổi mạng trích xuất đặc trưng của C-Tran (vốn là ResNet 101) sang kiến trúc của mô hình Đơn nhãn dương và thử nghiệm các hàm mất mát có kết quả tốt nhất trên mô hình Đơn nhãn dương như là ROLE, AN-LS và Huber.



**Hình 3.6:** Mô hình kết hợp giữa Đơn nhãn dương và C-Tran.

# Chương 4

## Thực nghiệm và đánh giá

*Ở chương này, chúng tôi sẽ trình bày về các kết quả thực nghiệm cho các đề suất ở Chương 3, nhận xét và bàn luận ý nghĩa của các kết quả này. Đồng thời mô tả về tập dữ liệu 4.1, độ đo đã sử dụng 4.2.2.*

### 4.1 Giới thiệu tập dữ liệu

#### 4.1.1 Nguồn gốc tập dữ liệu

Tập dữ liệu mà chúng tôi thử nghiệm được lấy từ cuộc thi [Food Recognition Benchmark 2022](#). Nguồn hình ảnh trong tập dữ liệu bao gồm hình ảnh về những bữa ăn hằng ngày được những tình nguyện viên ở Thụy Sĩ chụp lại và được tổng hợp lại bởi tổ chức Food & You. Sau đó chúng sẽ được phân loại và gán nhãn.

#### 4.1.2 Phân tích tập dữ liệu

Tập dữ liệu chính bao gồm 3 tập dữ liệu con là các tập (huấn luyện, đánh giá và kiểm tra) và số lượng nhãn là 323.

Giá trị trung bình, độ lệch chuẩn của hình ảnh trên toàn bộ tập dữ liệu huấn luyện và đánh giá lần lượt được tính toán theo 3 là [0.5957, 0.5094, 0.4278] và [0.2123, 0.2210, 0.2308]. Trên tập dữ liệu ImageNet cũng có giá

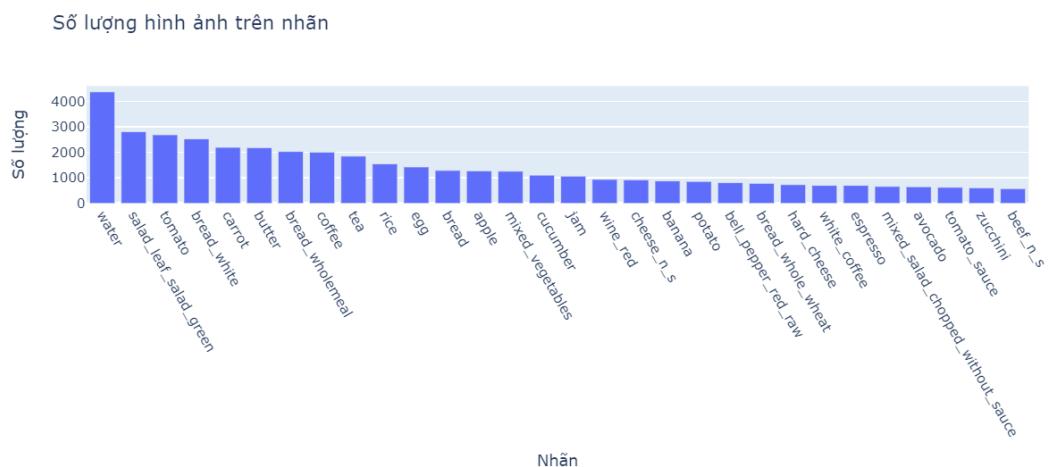
trị trung bình là [0.5957, 0.5094, 0.4278] nhưng khác đôi chút về độ lệch chuẩn [0.2104, 0.2192, 0.2290]. Có thể thấy hai tập dữ liệu rất tương đồng về mặt thống kê. Điều này cho thấy các hình ảnh trong cả hai tập dữ liệu có thể có chất lượng và điều kiện ánh sáng tương tự nhau. Trong bối cảnh học máy, điều này có thể giúp mô hình hoạt động nhất quán và hiệu quả trên cả hai tập dữ liệu mà không cần điều chỉnh nhiều.

**Tập huấn luyện** bao gồm 54392 hình ảnh. Số lượng nhãn trung bình, tối đa và tối thiểu trên một hình ảnh lần lượt là: 1.72, 13 và 1. Phần lớn ảnh trong tập dữ liệu có nhãn có lượng nhãn nằm trong khoảng 1 đến 3 (91.34%). Điều này cho thấy số lượng nhãn trên hình ảnh trong tập dữ liệu phân bố không đều.

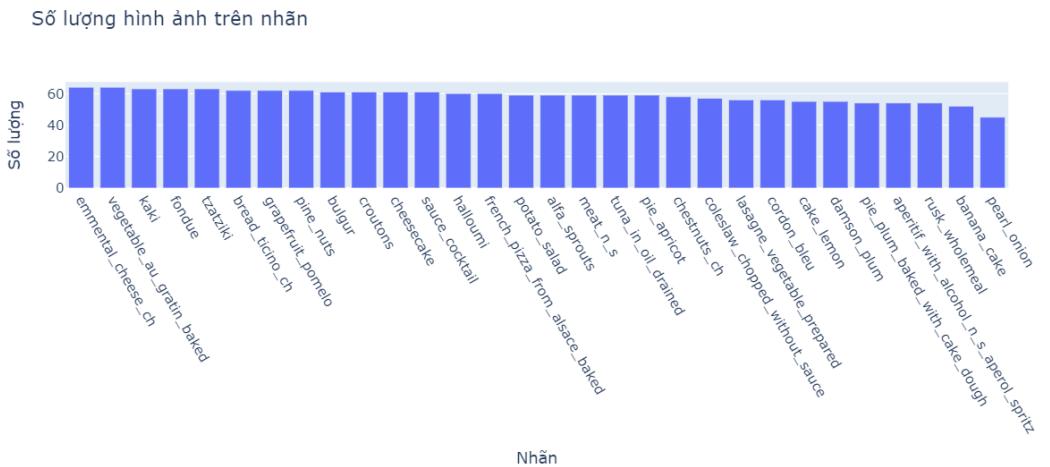
**Bảng 4.1:** Phân bố nhãn trong tập dữ liệu huấn luyện.

Số lượng nhãn trên hình ảnh	1	2	3	4	5	6	7	8	9	10	11	13
Tỉ lệ % trong tập dữ liệu	62	18.68	10.66	5.01	2.12	0.9	0.38	0.15	0.06	0.03	0.01	0

Tần suất xuất hiện của các nhãn trong tập dữ liệu không đều, có những nhãn có tần suất xuất hiện cao, cũng có những nhãn có tần suất xuất hiện rất thấp. Điều này cho thấy, trong tập dữ liệu có sự mất cân bằng về nhãn.

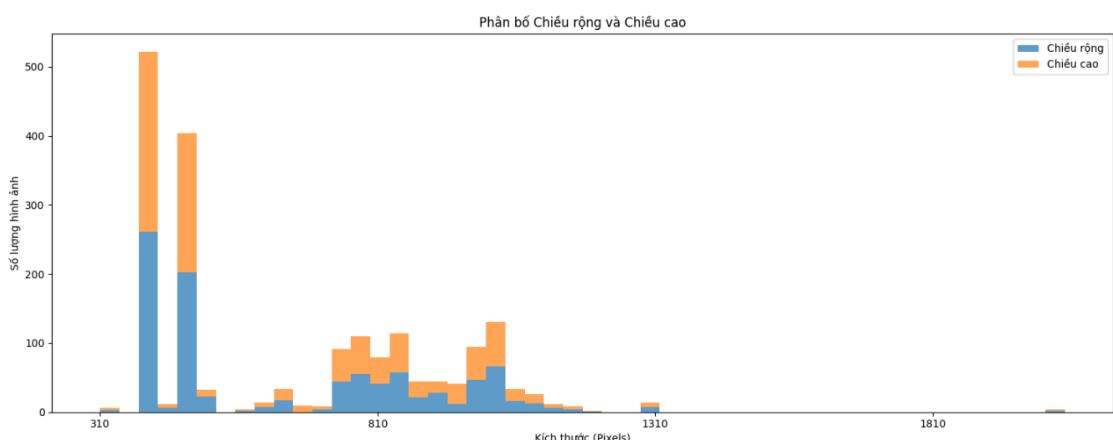


**Hình 4.1:** Top 30 nhãn có tần suất xuất hiện cao trong tập dữ liệu huấn luyện.



**Hình 4.2:** Top 30 nhãn có tần suất xuất hiện thấp trong tập dữ liệu huấn luyện.

Hình ảnh trong tập dữ liệu có kích thước chênh lệch với nhau, có một số hình ảnh có kích thước rất lớn và ngược lại.



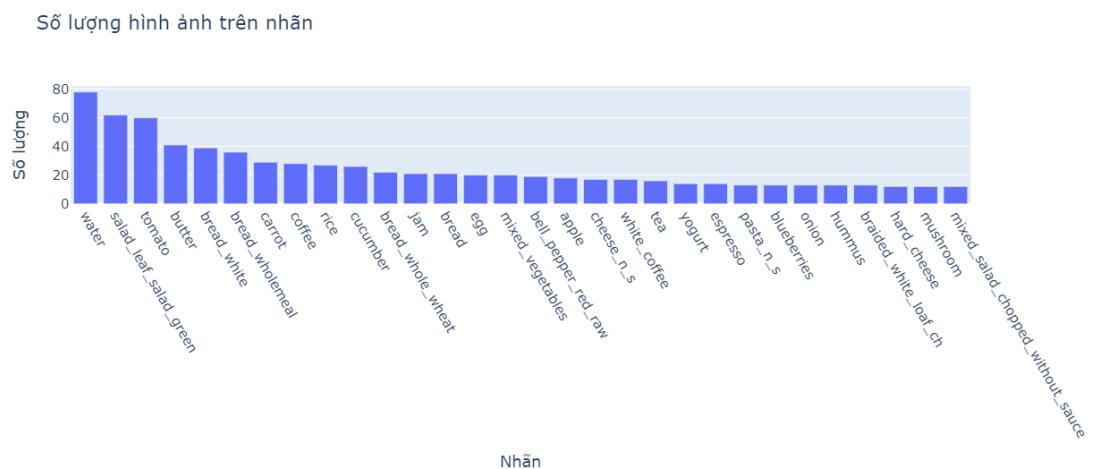
**Hình 4.3:** Phân bố kích thước hình ảnh trong tập dữ liệu huấn luyện.

**Tập đánh giá** bao gồm 946 hình ảnh. Số lượng nhãn trung bình, tối đa và tối thiểu trên một hình ảnh lần lượt là: 1.78, 9 và 1. Phần lớn ảnh trong tập dữ liệu có nhãn có lượng nhãn nằm trong khoảng 1 đến 3 (90.28%). Điều này cho thấy số lượng nhãn trên hình ảnh trong tập dữ liệu phân bố không đều.

**Bảng 4.2:** Phân bố nhãn trong tập dữ liệu đánh giá.

Số lượng nhãn trên hình ảnh	1	2	3	4	5	6	7	8	9
Tỉ lệ % trong tập dữ liệu	58.88	21.04	10.36	6.03	1.9	0.63	0.95	0.11	0.11

Tần suất xuất hiện của các nhãn trong tập dữ liệu không đều, có những nhãn có tần suất xuất hiện cao, cũng có những nhãn có tần suất xuất hiện rất thấp, thậm chí có những nhãn không xuất hiện. Điều này cho thấy, trong tập dữ liệu có sự mất cân bằng về nhãn.

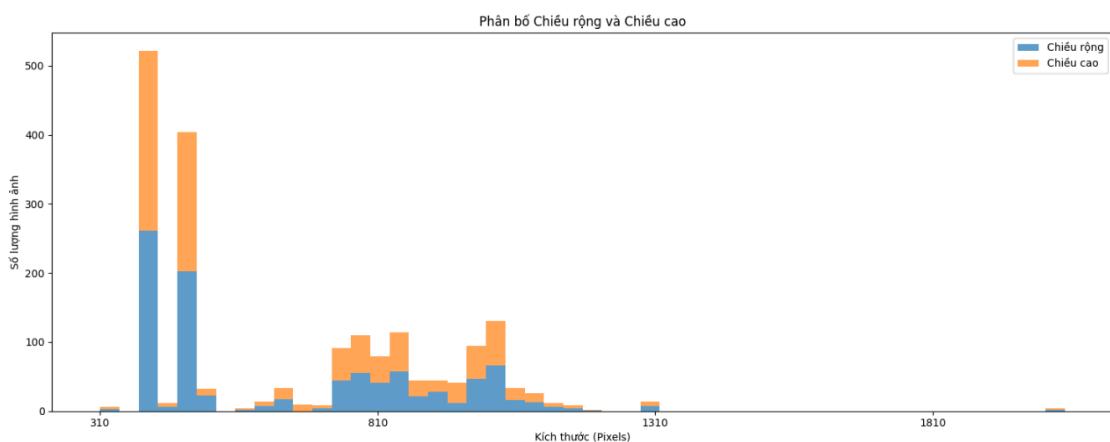


**Hình 4.4:** Top 30 nhãn có tần suất xuất hiện cao trong tập dữ liệu đánh giá.



**Hình 4.5:** Top 30 nhãn có tần suất xuất hiện thấp trong tập dữ liệu đánh giá.

Hình ảnh trong tập dữ liệu có kích thước chênh lệch với nhau, có một số hình ảnh có kích thước rất lớn và ngược lại.



**Hình 4.6:** Phân bố kích thước hình ảnh trong tập dữ liệu đánh giá.

**Tập kiểm tra:** Vì đây là cuộc thi nên các hình ảnh trong tập kiểm tra sẽ không được đánh nhãn. Do đó, trong luận văn này, chúng tôi chỉ sử dụng tập huấn luyện và tập đánh giá.

Ngoài ra, trong tập nhãn còn có hiện tượng nhãn này bao hàm nhãn khác. Ví dụ: nuts với peanut, coffee với espresso ...

Và do các ảnh trong tập dữ liệu được chụp bởi các người dùng khác nhau nên không có sự đồng nhất về cách chụp. Điều đó dẫn đến việc trong tập dữ liệu tồn tại nhiều bức ảnh chụp lệch, không rõ ràng hoặc chứa nhiều vật thể khác không phải thực phẩm.



**Hình 4.7:** Một số hình ảnh chất lượng thấp trong tập dữ liệu.

**Nhận xét:** Tập dữ liệu gặp những vấn đề như sau: số lượng nhãn phân bố không đều, mất cân bằng nhãn, nghĩa của nhãn bị bao hàm. Những điều này sẽ dẫn tới việc mô hình sẽ gặp khó khăn trong quá trình huấn luyện. Do đó, tập dữ liệu này là phù hợp để minh họa cho những vấn đề được nêu ra ở Chương 1 và phù hợp với các mô hình, phương pháp được đề xuất. Chúng tôi hy vọng có thể giải quyết được các vấn đề gãy phai ở tập dữ liệu này bởi các phương pháp, mô hình trên.

## 4.2 Cài đặt thí nghiệm

Các thí nghiệm này được thực hiện trên máy chủ của khoa công nghệ thông tin, trường đại học Khoa học Tự nhiên.

### 4.2.1 Sự phân chia dữ liệu

Để thử nghiệm mô hình Đơn nhãn dương, chúng tôi chia tập dữ liệu thành 3 phần: tập huấn luyện gốc được chia thành tập huấn luyện và tập

đánh giá, tập đánh giá gốc được dùng làm tập kiểm tra. Với cách phân chia này, tập kiểm tra hoàn toàn không được học bởi mô hình nên độ chính xác đo được ở tập kiểm tra là đáng tin cậy. Mô hình được huấn luyện bởi các tham số như bảng 4.3. Trong khi đó, C-Tran chia tập dữ liệu thành 2 phần: tập dữ liệu huấn luyện gốc và tập đánh giá được dùng làm tập kiểm tra. Cách chia này giúp cho mô hình có thể học một cách khái quát hơn trên tập dữ liệu. Và mô hình được huấn luyện theo các tham số như bảng 4.4.

#### 4.2.2 Độ đo

Tất cả mô hình thí nghiệm đều được đánh giá bằng độ đo *Mean Average Precision* (mAP) thường dùng cho các bài toán phân lớp đa nhãn.

Trước khi đi sâu vào mAP, chúng ta cần hiểu về *Average Precision* (AP). AP là diện tích dưới đường cong Precision-Recall (Độ Chính Xác - Độ Gọi). Công thức để tính AP là:

$$AP = \sum_n (R(n) - R(n-1)) P(n) \quad (4.1)$$

Ở đây:

- $n \in (0, 1)$  là giá trị ngưỡng phân lớp.
- $P(n)$  là giá trị độ chính xác tại điểm thứ  $n$ .
- $R(n)$  là giá trị độ gọi tại điểm thứ  $n$ .
- $R(n) - R(n-1)$  là sự thay đổi độ gọi giữa hai điểm liên tiếp.

Để tính mAP, chúng ta cần tính AP cho từng nhãn và sau đó lấy trung bình của tất cả các AP. Công thức tính mAP như sau:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4.2)$$

Trong đó:

- $N$  là tổng số nhãn.

-  $AP_i$  là giá trị AP cho nhãn thứ  $i$ .

### 4.2.3 Các tham số của các mô hình

#### 1) Mô hình Đơn nhãn dương

**Bảng 4.3:** Bảng các tham số của mô hình Đơn nhãn dương.

Các tham số phụ thuộc		
Tham số bậc cao	<ul style="list-style-type: none"> <li>- hàm mất mát</li> <li>- chế độ huấn luyện</li> <li>- biến thể tập đánh giá</li> </ul>	quan sát hoặc đầy đủ
Tham số học chuyển giao	<ul style="list-style-type: none"> <li>- tốc độ học khởi động</li> <li>- kích thước lô khởi động</li> </ul>	
Tham số tối ưu	<ul style="list-style-type: none"> <li>- kích thước lô</li> <li>- tốc độ học</li> </ul>	
Tham số bổ sung	<ul style="list-style-type: none"> <li>- sử dụng mô hình huấn luyện sẵn?</li> </ul>	
Tham số phụ thuộc chế độ học	<ul style="list-style-type: none"> <li>- biến thể tập huấn luyện</li> <li>- đóng băng bộ trích xuất đặc trưng?</li> <li>- sử dụng đặc trưng có sẵn?</li> <li>- bộ phận phân lớp được sử dụng</li> </ul>	quan sát hoặc đầy đủ
Các tham số cố định		
Tham số tối ưu	<ul style="list-style-type: none"> <li>- hệ số cho tốc độ học bộ ước lượng nhãn</li> <li>- số epoch</li> <li>- độ đo</li> </ul>	$10 \times$ tốc độ học 10 hoặc 25 map
Tham số bổ sung	<ul style="list-style-type: none"> <li>- hệ số làm mịn nhãn</li> </ul>	0.1
Tham số tập dữ liệu	<ul style="list-style-type: none"> <li>- tỉ lệ chia tập đánh giá</li> <li>- số lượng lớp</li> <li>- số nhãn dương kỳ vọng</li> </ul>	0.2 323 1.7

## 2) Mô hình C-Tran

**Bảng 4.4:** Bảng các tham số của mô hình C-Tran.

Tham số dữ liệu	- số lượng nhãn	323
Tham số xử lí dữ liệu đầu vào	- kích thước thu/ phóng ảnh - kích thước cắt ảnh	640 576
Tham số cài đặt mô hình	- số lớp Encoder - số đầu của cơ chế tự chú ý - sử dụng chiến lược che nhãn huấn luyện - không cập nhật tham số trên mạng trích xuất đặc trưng	3 4 có không
Tham số tối ưu	- số lượng ảnh đầu vào - tốc độ học - thuật toán tối ưu - số vòng lặp để cập nhật tham số - tỉ lệ tham số không được cập nhật trong mạng	8 $10^{-5}$ adam 2 0.1

## 4.3 Kết quả mô hình Đơn nhãn dương

### 4.3.1 Kết quả

#### 1) Các tham số và kết quả tương ứng

**Bảng 4.5:** Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng ResNet 50.

Hàm mất mát	Chế độ huấn luyện	Hàm kích hoạt	mAP tập đánh giá	mAP tập kiểm tra
FO	đầu cuối	sigmoid	0.3724	0.7938
ROLE	tuyên tính	sigmoid	0.4040	1.1570
BCE	đầu cuối	sigmoid	1.9319	5.8275
KL	đầu cuối	sigmoid	2.0014	5.9030
HI	đầu cuối	sigmoid	2.4091	7.0870
IUN	đầu cuối	sigmoid	3.8977	8.0127
ROLE	đầu cuối	sigmoid	19.0979	26.5441

**Bảng 4.5:** Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng ResNet 50.

Hàm măt mát	Chế độ huấn luyện	Hàm kích hoạt	mAP tập đánh giá	mAP tập kiểm tra
HU	đầu cuối	sigmoid	21.0043	32.6947
AN-LS	đầu cuối	sigmoid	24.2951	34.6173
AN-LS	chuyển giao	sigmoid	24.4427	34.8327
HU	chuyển giao	softmax	0.4554	1.5206

**Bảng 4.6:** Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng EfficientNet B7.

Hàm măt mát	Chế độ huấn luyện	Hàm kích hoạt	mAP tập đánh giá	mAP tập kiểm tra
FO	đầu cuối	sigmoid	0.4113	0.9288
AN-LS	chuyển giao	sigmoid	0.4538	1.1082
ROLE	chuyển giao	sigmoid	1.0426	2.2215
HU	chuyển giao	sigmoid	1.1985	2.7821
HU	đầu cuối	sigmoid	5.3637	9.0930
AN-LS	chuyển giao	softmax	0.4960	1.6186

**Bảng 4.7:** Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng EfficientNet B0.

Hàm măt mát	Chế độ huấn luyện	Hàm kích hoạt	mAP tập đánh giá	mAP tập kiểm tra
AN-LS	chuyển giao	softmax	0.4709	1.0321

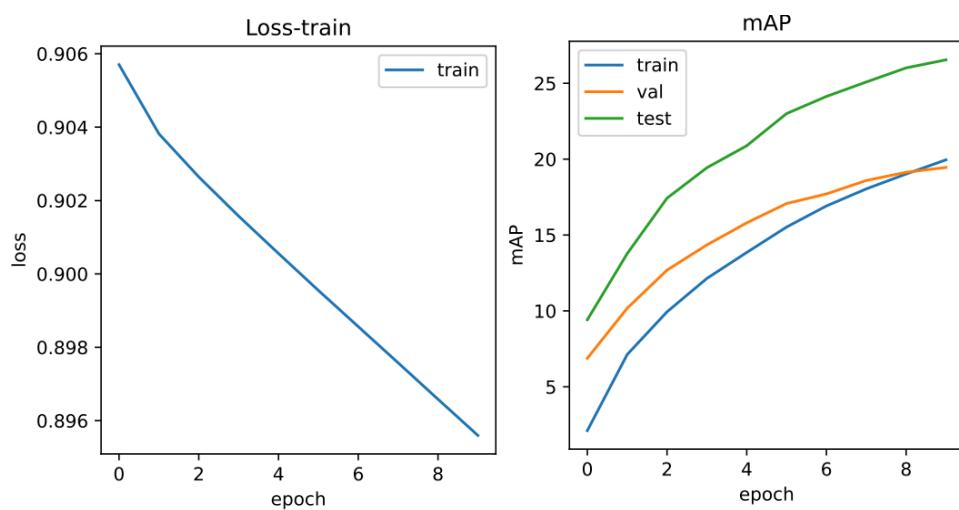
**Bảng 4.7:** Bảng các tham số và kết quả tương ứng của đa nhãn dương sử dụng EfficientNet B0.

Hàm măt mát	Chế độ huấn luyện	Hàm kích hoạt	mAP tập đánh giá	mAP tập kiểm tra
HU	chuyển giao	softmax	0.4193	1.2309

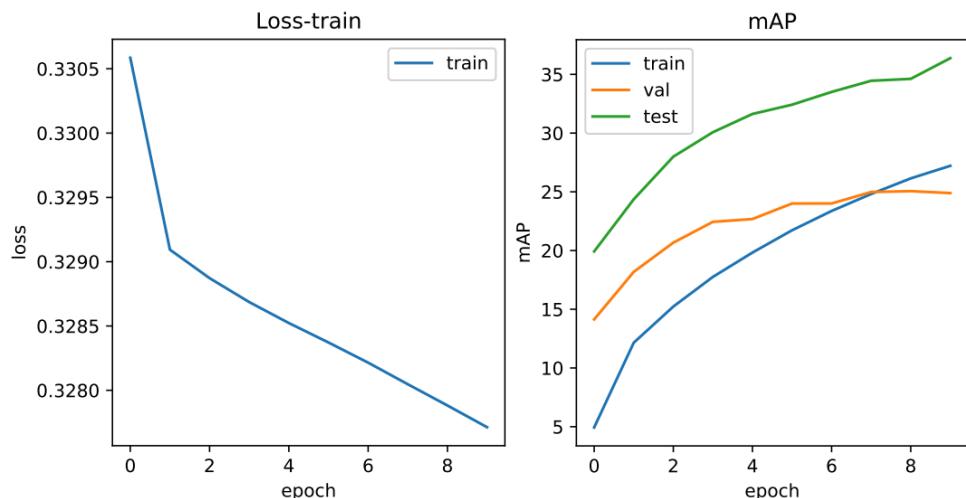
**Chú ý:** FO, HU, HI, KL lần lượt là các hàm măt mát Focal, Huber, Squared Hinge và phân kỳ KL. Mỗi kết quả trên đều là kết quả tốt nhất thu được khi thí nghiệm trên các tốc độ học khác nhau gồm  $10^{-3}, 10^{-4}, 10^{-5}$  và các kích thước lô khác nhau gồm 8, 16, 32. Tất cả kết quả đều được thí nghiệm trên tập dữ liệu gốc (chưa cắt). Chúng tôi chưa thực nghiệm trên tập dữ liệu đã xử lý (đã cắt), cũng như thử nghiệm với bộ phận ước lượng nhãn mới (sử dụng hàm softmax) vì giới hạn về tài nguyên sử dụng.

**Nhận xét:** Ở mọi cài đặt, EfficientNet B7 cho kết quả tệ hơn ResNet 50. Hàm kích hoạt sigmoid cho kết quả tốt hơn softmax (bộ phận ước lượng nhãn vẫn sử dụng hàm sigmoid). Các hàm măt mát cho các kết quả tốt nhất là ROLE, Huber và AN-LS. Trong đó, hàm măt mát AN-LS với chế độ chuyển giao cho kết quả tốt nhất. Và hàm măt mát mới Huber cho kết quả tốt hơn hàm ROLE, hàm được cho là tốt nhất trong công trình Đơn nhãn dương ban đầu. Tuy nhiên nhìn chung, cài đặt của mô hình Đơn nhãn dương cho kết quả tệ đối với tập dữ liệu thức ăn [4.1](#). Vấn đề này sẽ được phân tích kĩ hơn ở phần phân tích lõi cơ bản [4.3.2](#).

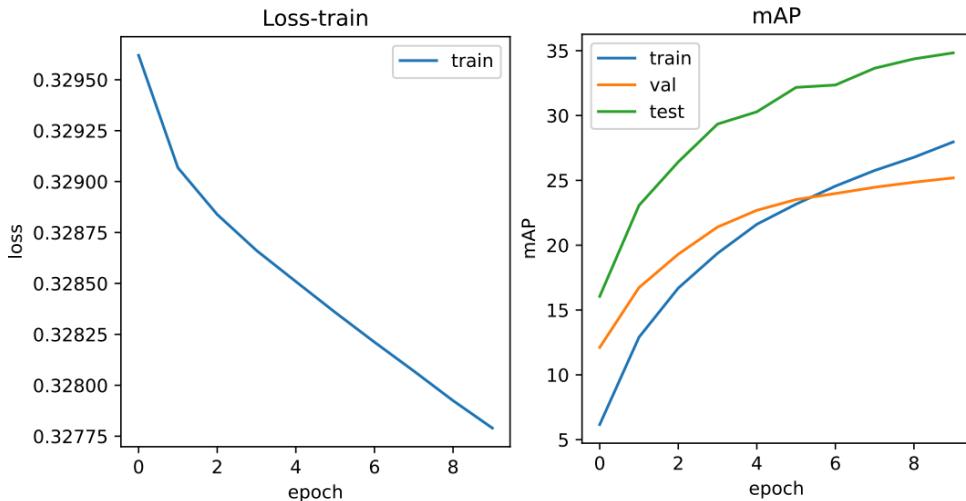
## 2) Đồ thị học của một số kết quả tốt nhất



**Hình 4.8:** Hàm mất mát: ROLE - Chế độ học: đầu cuối - Bộ phân lớp: ResNet 50.



**Hình 4.9:** Hàm mất mát: AN-LS - Chế độ học: đầu cuối - Bộ phân lớp: ResNet 50.



**Hình 4.10:** Hàm mất mát: AN-LS - Chế độ học: chuyển giao - Bộ phân lớp: ResNet 50.

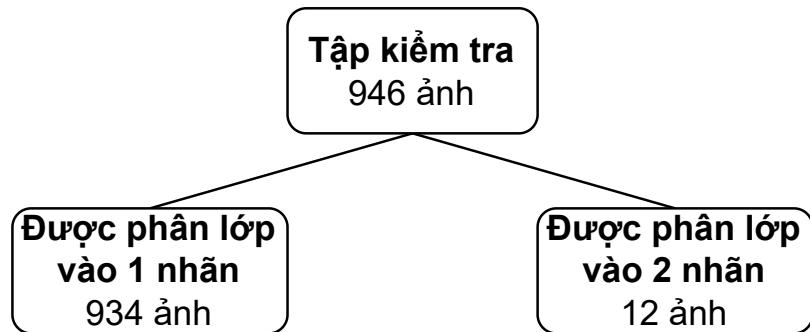
**Chú ý:** loss là độ lỗi; các đồ thị bên trái biểu thị độ lỗi trên tập huấn luyện; các đồ thị bên phải biểu thị độ chính xác của mô hình trên cả 3 tập huấn luyện (train), tập đánh giá (val) và tập kiểm tra (test) qua các lần học (epoch).

**Nhận xét:** Độ lỗi trên tập huấn luyện đã rất thấp. Tuy nhiên, độ chính xác lại không cao như mong muốn. Điều này xảy ra có thể do mô hình quá khớp khi học với các nhãn chiếm số lượng lớn hơn nhiều so với các nhãn còn lại. Hoặc do độ lỗi bị chia quá nhỏ trong các hàm mất mát. Vấn đề này sẽ được tìm hiểu kĩ hơn ở phần phân tích lỗi cơ bản 4.3.2.

### 4.3.2 Bàn luận và giải thích

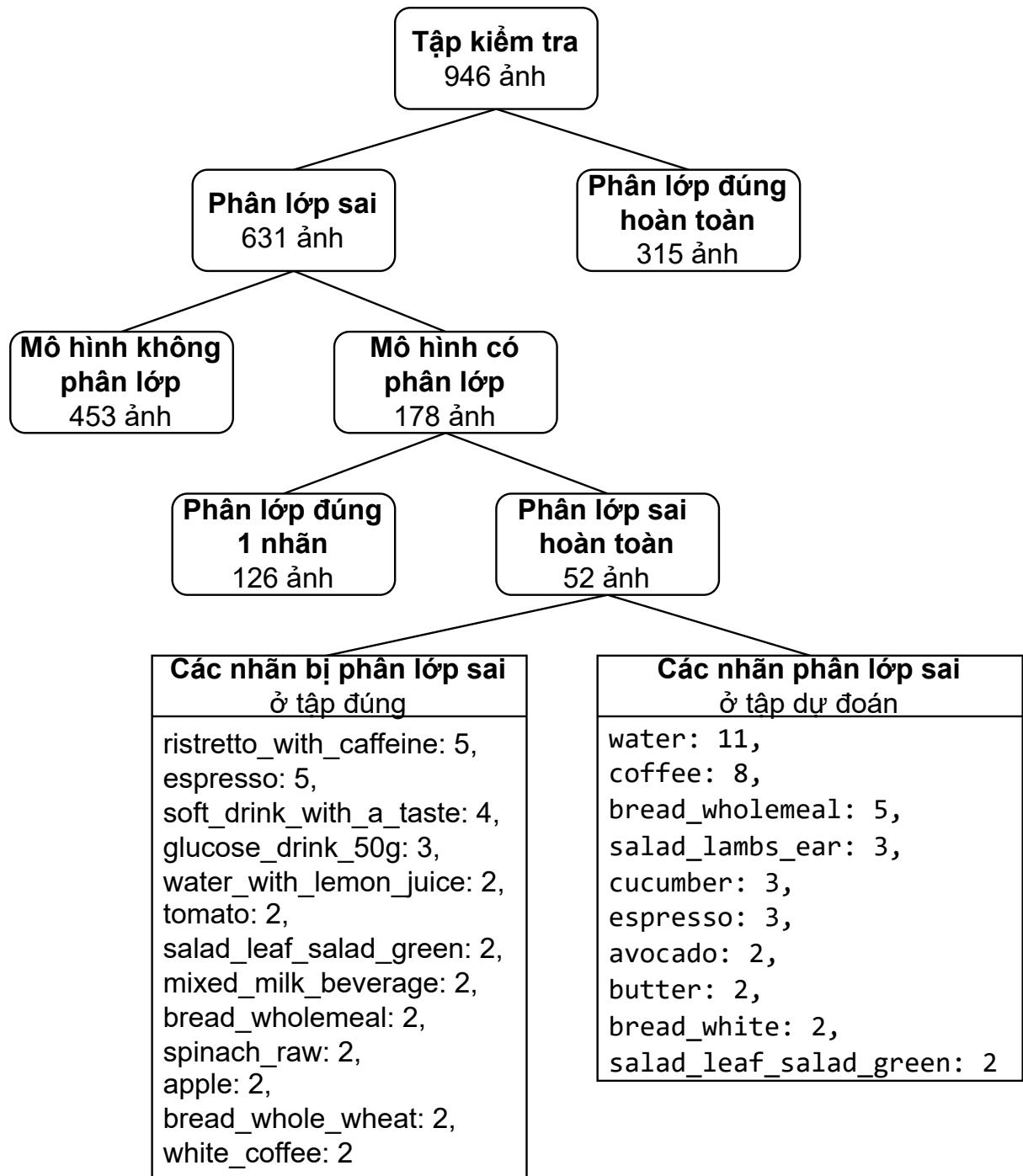
Mô hình Đơn nhãn dương cho ra kết quả thấp trên tập dữ liệu thí nghiệm. Để có thể tìm ra lý do cho hiện tượng này và đưa ra hướng cải tiến phù hợp, chúng tôi thực hiện phân tích lỗi cơ bản. Ở các bức ảnh bị gán nhãn sai trên tập kiểm tra, chúng tôi sẽ quan sát tập nhãn đúng và tập nhãn dự đoán trên các ngưỡng phân lớp khác nhau. Từ đó rút ra các đối tượng mà mô hình dễ nhầm lẫn với nhau. Mô hình được dùng để phân tích là mô hình cho kết quả tốt nhất ở các bảng 4.5, 4.6 và 4.7.

## 1) Ở ngưỡng phân lớp 0.5



**Hình 4.11:** Số lượng nhãn được phân lớp vào các ảnh.

Do tính chất của mô hình Đơn nhãn dương (chỉ giữ lại 1 nhãn dương mỗi véc-tơ nhãn) và tính chất của tập dữ liệu (hơn 80% số lượng ảnh có 1 đến 2 nhãn), nên số lượng ảnh được dự đoán chỉ chứa 1 nhãn lên đến 98.7%.



**Hình 4.12:** Phân tích lỗi ở ngưỡng 0.5.

Ở ngưỡng 0.5, tỉ lệ lỗi của mô hình là  $\frac{631}{946} = 66.7\%$ . Trong các ảnh phân lớp sai này, số lượng ảnh mà mô hình không phân lớp (tập nhãn dự đoán là rỗng) chiếm đến  $\frac{453}{631} = 71.8\%$ . Có thể thấy rằng, 0.5 là ngưỡng phân lớp cao đối với mô hình chưa tự tin phân biệt tốt được đặc trưng của các

đối tượng thực phẩm. Ở phần sau, chúng tôi sẽ thay đổi ngưỡng phân lớp để đảm bảo tất cả ảnh đều được phân lớp 2). Từ đó, có sự phân tích sâu hơn về hiệu suất của mô hình trên tập dữ liệu thực phẩm. Trong 178 ảnh bị phân lớp sai còn lại, có đến  $\frac{126}{178} = 70.8\%$  ảnh được mô hình dự đoán đúng 1 nhãn. Mà có đến 98.7% ảnh được dự đoán chỉ có một nhãn. Điều này chỉ ra rằng, tỉ lệ mô hình dự đoán đúng nhưng không đầy đủ cũng khá cao. Nếu xem các dự đoán không đầy đủ này cũng là một dự đoán chính xác thì độ chính xác của mô hình lúc này lên đến  $\frac{315+126}{946} = 46.6\%$  (tăng 13.3% so với độ chính xác 33.3% ban đầu).

Quan sát tập các nhãn bị phân lớp sai và tập các nhãn phân lớp sai nhiều nhất lần lượt ở các tập nhãn đúng và các tập nhãn dự đoán trong 52 ảnh phân lớp sai hoàn toàn, dễ dàng nhận ra giữa chúng có sự tương đồng về nghĩa. Ví dụ chúng ta có espresso, white\_coffee với coffee hay giữa bread\_wholemeal, bread\_whole\_wheat và bread\_white. Đồng nghĩa với việc nếu một ảnh có nhãn là espresso được đoán nhãn là coffee sẽ bị xem là sai mặc dù thực tế mô hình đã gán nhãn chính xác. Điều này ảnh hưởng đến độ tin cậy về kết quả độ chính xác của mô hình.

Để có thể đánh giá chính xác mức độ ảnh hưởng của vấn đề về nhãn tương đồng nghĩa, chúng tôi sẽ khai thác các cặp nhãn dễ bị phân loại nhầm nhất. Bắt đầu, chúng tôi sẽ tách tập nhãn đúng và tập dự đoán ở mỗi bức ảnh trong 52 ảnh trên thành các cặp (nhãn đúng, nhãn đoán). Sau đó đếm các cặp xuất hiện nhiều nhất.

**Bảng 4.8:** Các cặp (nhãn đúng, nhãn đoán) với số lần xuất hiện đã được xếp giảm dần.

Số lượng	Các cặp (nhãn đúng, nhãn đoán)
4	(water, soft_drink_with_a_taste), (espresso, coffee)
3	(water, glucose_drink_50g), (espresso, ristretto_with_caffeine)
2	(water, water_with_lemon_juice), (bread_wholemeal, bread_whole_wheat), (mixed_salad_chopped_without_sauce, salad_leaf_salad_green), (coffee, white_coffee), (coffee, ristretto_with_caffeine)

Dễ quan sát thấy các nhóm nhãn mà mô hình Đơn nhãn dương dễ nhầm lẫn với nhau gồm: nhóm nước (water, soft\_drink\_with\_a\_taste, glucose\_drink\_50g, water\_with\_lemon\_juice), nhóm cà phê (espresso, coffee, white\_coffee, ristretto\_with\_caffeine), nhóm bánh mì (bread\_wholemeal, bread\_whole\_wheat) và nhóm salad (mixed\_salad\_chopped\_without\_sauce, salad\_leaf\_salad\_green). Số lượng ảnh bị phân loại sai rơi vào các nhóm này chiếm đến 46.2% trong các ảnh bị phân loại sai hoàn toàn. Ở mức con người cũng khó có thể phân loại được các bức ảnh thuộc cùng một nhóm trong các nhóm trên. Ví dụ:



**Hình 4.13:** Từ trái qua lần lượt là ảnh chứa các nhãn: water, soft\_drink\_with\_a\_taste, glucose\_drink\_50g, water\_with\_lemon\_juice.

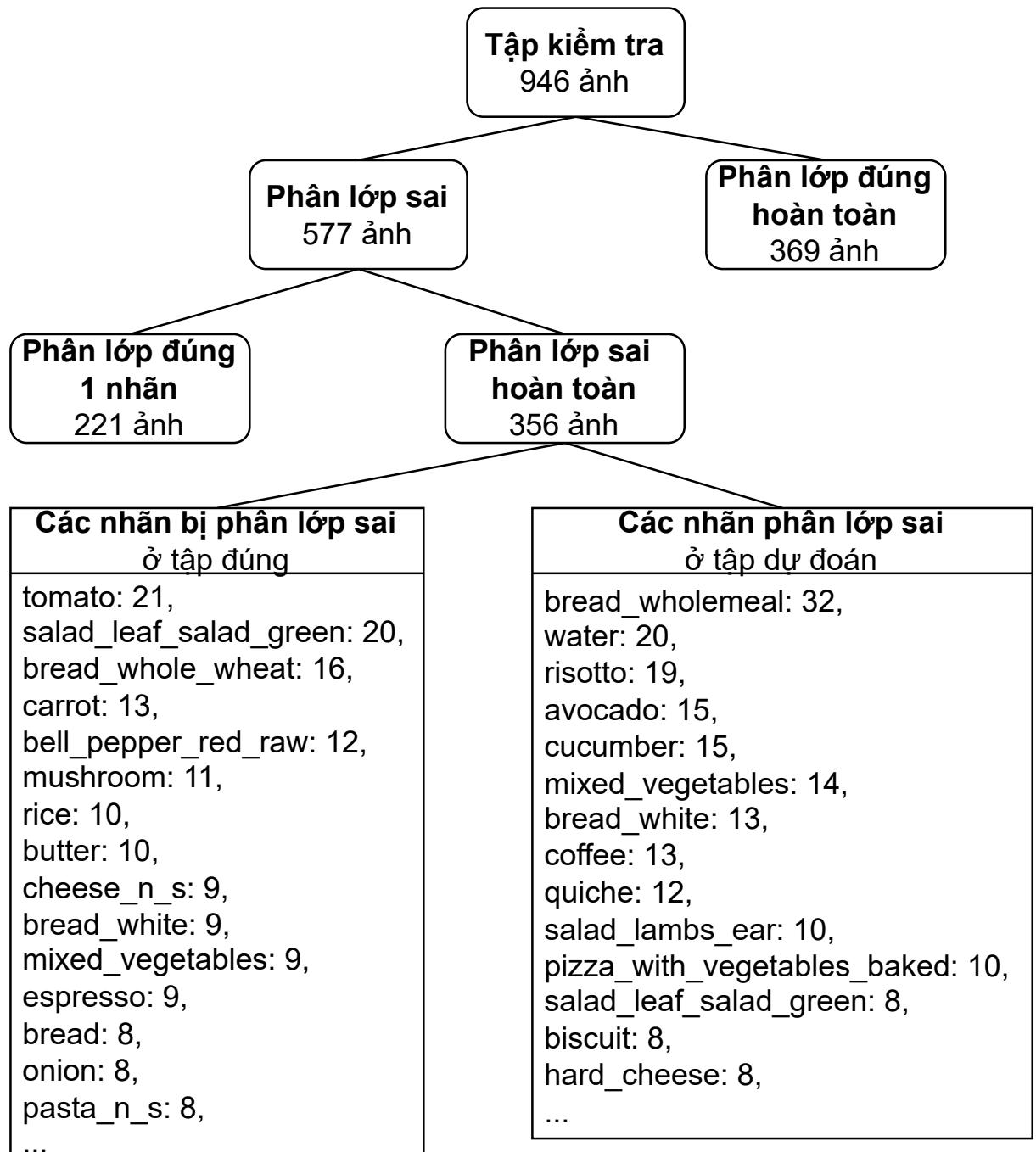


**Hình 4.14:** Từ trái qua lần lượt là ảnh chứa các nhãn: coffee, espresso, white\_coffee, ristretto\_with\_caffeine.

## 2) Mọi bức ảnh đều được dự đoán vào một lớp

Ngoài 0.5 khiến  $\frac{453}{946} = 47.9\%$  số lượng ảnh không được phân lớp 1). Điều này ảnh hưởng đến độ chính xác của mô hình. Chúng tôi sẽ phân tích trường hợp mọi bức ảnh đều được dự đoán, bằng cách gán mỗi bức ảnh

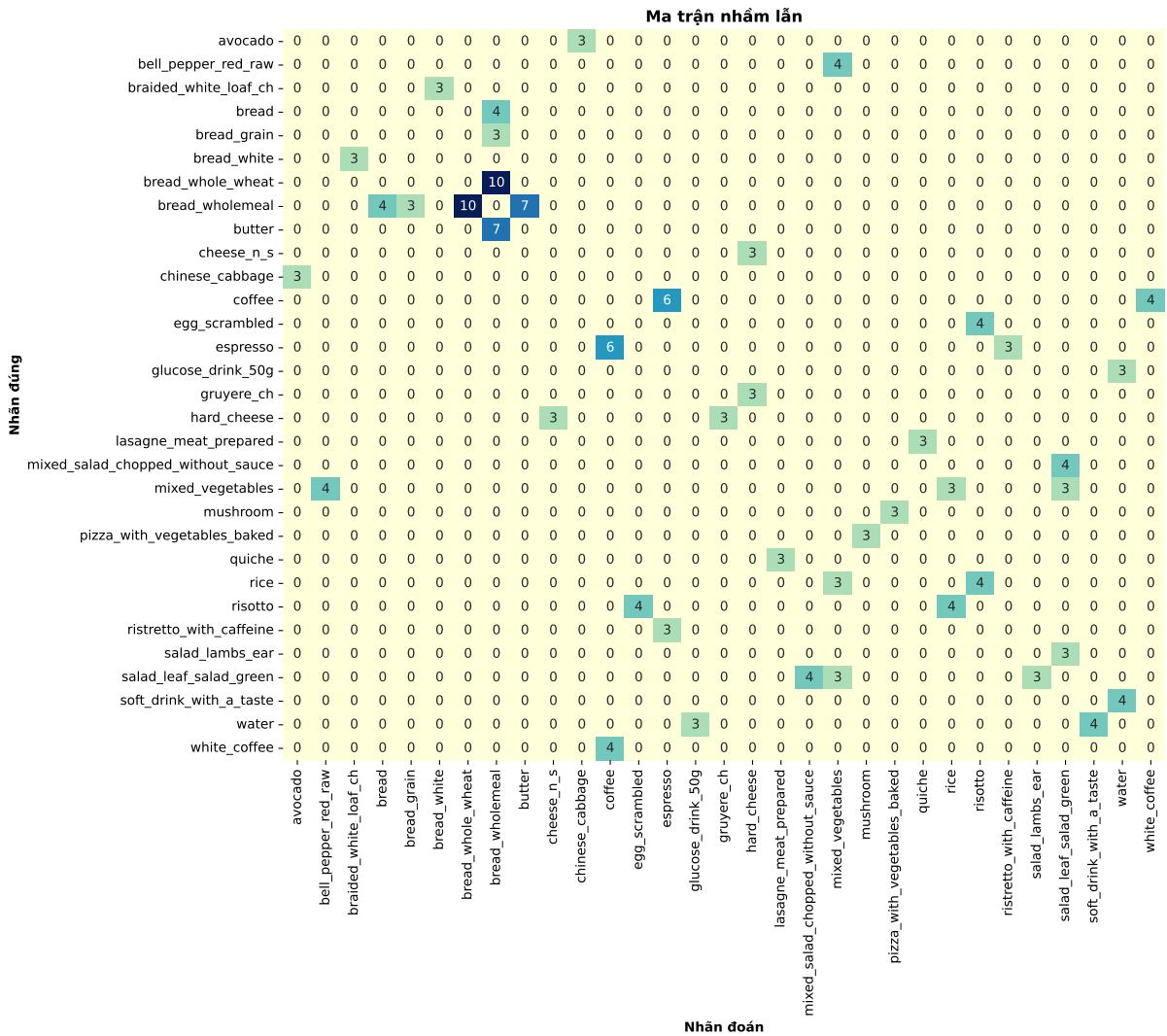
vào lớp có xác suất đầu ra lớn nhất và không phụ thuộc vào một ngưỡng cụ thể. Tập nhãn dự đoán ở mỗi bức ảnh lúc này đều chỉ có 1 phần tử. Kết quả thu được như sau:



**Hình 4.15:** Phân tích lỗi ở trường hợp mọi bức ảnh đều được dự đoán vào một lớp.

Dáng ngạc nhiên tỉ lệ phân lớp đúng hoàn toàn lúc này là  $\frac{369}{946} = 39\%$  (tăng 5.7% so với độ chính xác 33.3% ban đầu). Nếu tính thêm số ảnh phân lớp đúng 1 nhãn (đoán đúng nhưng không đủ tất cả nhãn) thì độ chính xác lúc này lên đến  $\frac{221+369}{946} = 62.4\%$  (tăng 23.4% so với trường hợp chỉ tính các trường hợp phân lớp đúng hoàn toàn). Tuy nhiên, số ảnh phân lớp sai hoàn toàn cũng tăng lên đáng kể (tăng 304 ảnh - tăng 584.6%), chiếm 37.6% số ảnh tập kiểm tra.

Tập các nhãn bị phân lớp sai và tập các nhãn phân lớp sai nhiều nhất cũng có tính chất tương tự ở ngưỡng 0.5. Ngoài ra còn có thêm tính chất nhãn này là nguyên liệu của nhãn khác. Ví dụ như rice là nguyên liệu chính của risotto (một món cơm kiểu Ý); tomato, carrot, avocado, ... là các nguyên liệu dễ thấy trong các món nhóm salad. Trong đó, các cặp nhãn dễ bị phân loại nhầm nhất là:



**Hình 4.16:** Ma trận nhầm lẫn ở ngưỡng thỏa 1 nhãn lớn nhất.

Quan sát một số mẫu thuộc cùng nhóm dễ bị nhầm lẫn ngay cả ở mức con người:



**Hình 4.17:** Hàng trên là các ảnh chứa rice - Hàng dưới là các ảnh chứa risotto.



**Hình 4.18:** Từ trái qua lần lượt là ảnh chứa các nhãn: bread, bread\_-white, bread\_grain, bread\_wholemeal, bread\_whole\_wheat.



**Hình 4.19:** Từ trái qua lần lượt là ảnh chứa các nhãn: mixed\_salad\_-chopped\_without\_sauce, salad\_leaf\_salad\_green.

Chúng tôi nhận ra rằng các ảnh chứa nhãn thuộc nhóm bánh mì trên cũng đều chứa nhãn butter. Điều này xuất phát từ thói quen ăn bánh mì với bơ của các tình nguyện viên cung cấp dữ liệu. Và các bức ảnh thuộc nhóm salad cũng chứa các nhãn nguyên liệu như tomato. Trường hợp các đối tượng phân lớp thường xuyên xuất hiện cùng nhau, nhưng tập dữ liệu có ít các ảnh riêng lẻ của từng đối tượng sẽ ảnh hưởng đến khả năng học để phân biệt chúng của mô hình. Sau khi khai thác các tập phổ biến (thuật toán FP-Growth [1]) với ngưỡng  $minsupport = 500$  trong ma trận nhãn huấn luyện và đánh giá (biết số lượng ảnh trung bình ở mỗi nhãn là 290), chúng tôi thu được kết quả sau:

**Bảng 4.9:** Bảng các nhóm nhãn thường xuất hiện cùng với nhau trong tập huấn luyện và đánh giá.

Số lượng	Tập phổ biến
757	(salad_leaf_salad_green, tomato)
616	(butter, bread_white)
512	(butter, bread_wholemeal)

**Bảng 4.10:** Bảng số lần xuất hiện riêng lẻ của các nhãn thường xuất hiện cùng với nhau trong bảng 4.9.

Số lượng	Nhãn phổ biến
2807	salad_leaf_salad_green
2686	tomato
2526	bread_white
2180	butter
2030	bread_wholemeal

Số lần nhãn butter xuất hiện cùng 2 nhãn thuộc nhóm bánh mì gồm bread\_white và bread\_wholemeal là rất cao. Không những thế, với số lần nhãn bread\_white hoặc bread\_wholemeal xuất hiện mỗi lần có nhãn butter xuất hiện là 1123, có thể thấy xác xuất để 2 nhãn bánh mì xuất hiện khi có nhãn butter  $P(\text{butter} \rightarrow \text{bread\_white} \text{ hoặc } \text{bread\_wholemeal}) = \frac{1123}{2180}$  lên đến 51.5%. Nếu mở rộng ra cho trường hợp tất cả nhãn bánh mì (bread\_white hoặc bread\_wholemeal hoặc bread hoặc bread\_grain hoặc bread\_whole\_wheat) xuất hiện cùng nhãn butter thì ta có  $P(\text{butter} \rightarrow \text{nhóm bánh mì}) = 69.9\%$ . Đây là tỉ lệ rất cao. Cặp nhãn (salad\_leaf\_salad\_green, tomato) tuy cùng xuất hiện nhiều nhưng tỉ lệ để xuất hiện nhãn tomato trong món salad\_leaf\_salad\_green chỉ có 27% nên mô hình vẫn phân biệt tốt 2 nhãn.

Qua quá trình phân tích, chúng tôi nhận thấy rằng mô hình Đơn nhãn dương đã làm tốt trong vấn đề trích xuất và nhận diện các đặc trưng ảnh với số lượng nhãn cần đánh thấp, giúp giảm gánh nặng cho yêu cầu cao về số lượng nhãn cần đánh cho các tập dữ liệu sau này. Tuy nhiên, Với các vấn đề về ý nghĩa nhãn làm ảnh hưởng lớn đến kết quả phân lớp, chúng ta cần một mô hình có thể học được mối liên hệ giữa các nhãn. Đó là lý do chúng tôi chọn mô hình C-Tran 2.7 cho vấn đề này. Kết quả ở phần tiếp theo sẽ cho thấy C-Tran đã làm rất tốt điều đó.

## 4.4 Kết quả mô hình C-Tran

Các kết quả được đề cập trong phần này được chúng tôi thực hiện với nhiều cài đặt và thay đổi khác nhau trên mô hình C-Tran để có thể tìm ra những tham số tối ưu của mô hình đối với tập dữ liệu thực nghiệm.

### 4.4.1 Các tham số và kết quả tương ứng

**Bảng 4.11:** Bảng các tham số và kết quả tương ứng của C-Tran khi sử dụng hàm mất mát binary cross-entropy.

Mạng trích xuất đặc trưng	Hàm kích hoạt	Số lớp En- coder	Che nhān huân biết luyện	Lượng nhān trước	Trạng thái nhān	Kết qua kiểm tra
ResNet 101	sigmoid	3	có	0	tổng	91.3
ResNet 101	softmax	3	có	0	tổng	90.1
ResNet 101	sigmoid	3	có	243	tổng	91.3
EfficientNet B0	sigmoid	3	có	0	tổng	91.3
MobileNet V2	sigmoid	3	có	0	tổng	91.3
MobileNet V2	sigmoid	3	có	243	tổng	91.3
MobileNet V2	sigmoid	3	có	0	tích	90.6
MobileNet V2	softmax	3	có	0	tổng	89.8
MobileNet V2	sigmoid	4	có	0	tổng	91.3
MobileNet V2	sigmoid	2	không	0	tổng	91.3
MobileNet V2	sigmoid	2	có	0	tổng	91.3

**Chú ý:** Tất cả kết quả đều được thí nghiệm trên tập dữ liệu gốc (chưa cắt). Chúng tôi chưa thực nghiệm trên tập dữ liệu đã xử lý (đã cắt) vì giới hạn về tài nguyên.

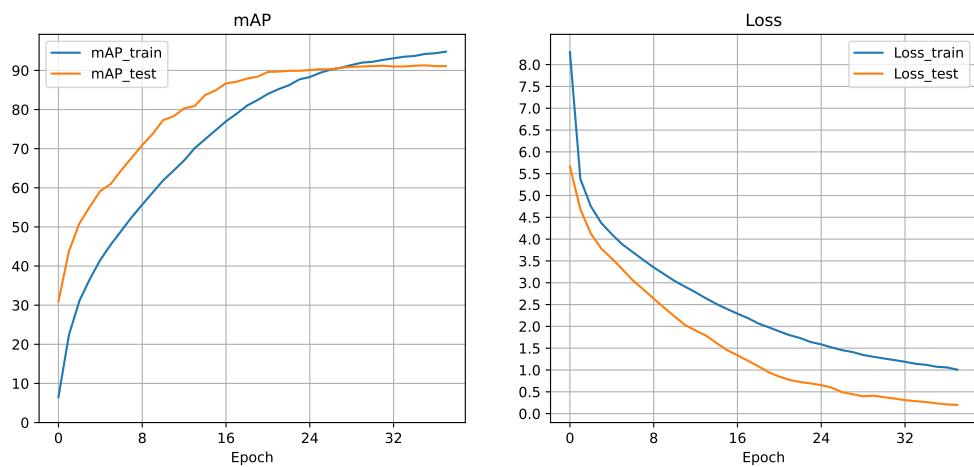
**Nhận xét:**

Đối với các cài đặt thay đổi so với mô hình gốc: Trong hầu hết các cài đặt, các mạng trích xuất đặc trưng như EfficientNet B0, MobileNet V2, và ResNet 101 đều cho ra kết quả tương đương. Tuy nhiên, MobileNet V2 là lựa chọn tối ưu nhất về số lượng tham số và độ chính xác. Khi sử dụng hàm sigmoid thay vì hàm softmax làm hàm kích hoạt, hiệu quả đầu ra sẽ cao hơn. Cơ chế tổng trạng thái cho kết quả tốt hơn so với tích trạng thái. Việc có sử dụng nhãn biết trước trong quá trình huấn luyện hay không cũng không ảnh hưởng đến kết quả đầu ra. Số lượng lớp Encoder là 2, 3, hoặc 4 đều cho ra kết quả tương tự nhau nhưng 2 là lựa chọn tối ưu về mặt tham số cũng như kết quả, và phương pháp che nhãn huấn luyện không tác động đáng kể đến kết quả cuối cùng của mô hình. Tóm lại, các cài đặt đều cho kết quả tốt và các kết quả này không quá chênh lệch với nhau.

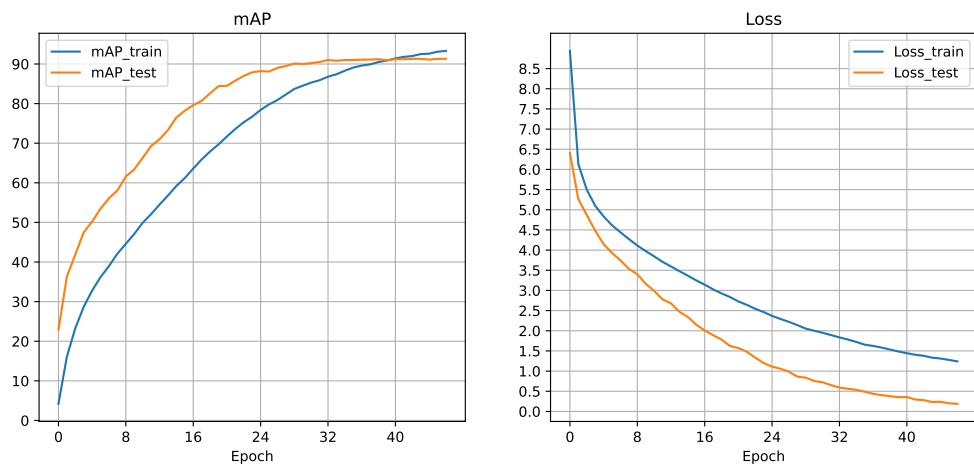
So sánh với mô hình Đơn nhãn dương: C-Tran cho ra những kết quả vượt trội hoàn toàn so với mô hình Đơn nhãn dương mà chúng tôi đã đề cập đến trước đó. Điều này thể hiện qua việc C-Tran được chúng tôi thay đổi phương thức huấn luyện, chúng tôi không chia nhỏ tập huấn luyện ra mà trực tiếp lấy toàn bộ dữ liệu để huấn luyện cho C-Tran nên C-Tran có thể học và khai quát hóa nhiều đặc trưng hơn đối với tập dữ liệu. Đồng thời, kiến trúc C-Tran cho thấy sự mạnh mẽ trong việc khai thác những tương quan về mặt ngữ nghĩa giữa các nhãn với nhau, điều này là vô cùng quan trọng đối với nhiệm vụ phân loại đa nhãn nói chung và tập dữ liệu mà chúng tôi thực nghiệm nói riêng.

#### 4.4.2 Đồ thị của các kết quả tốt nhất

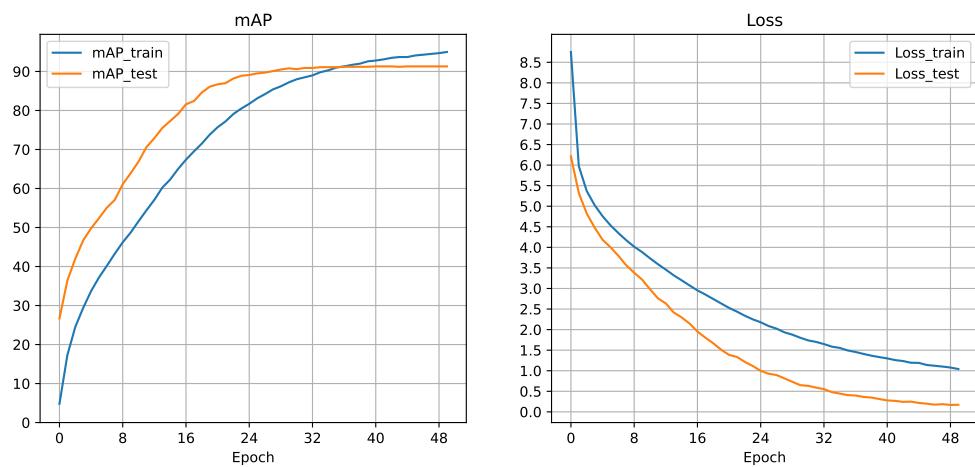
Những đồ thị kết quả bên dưới được chúng tôi trích xuất từ những mô hình đạt kết quả tốt nhất trong quá trình thực nghiệm của chúng tôi.



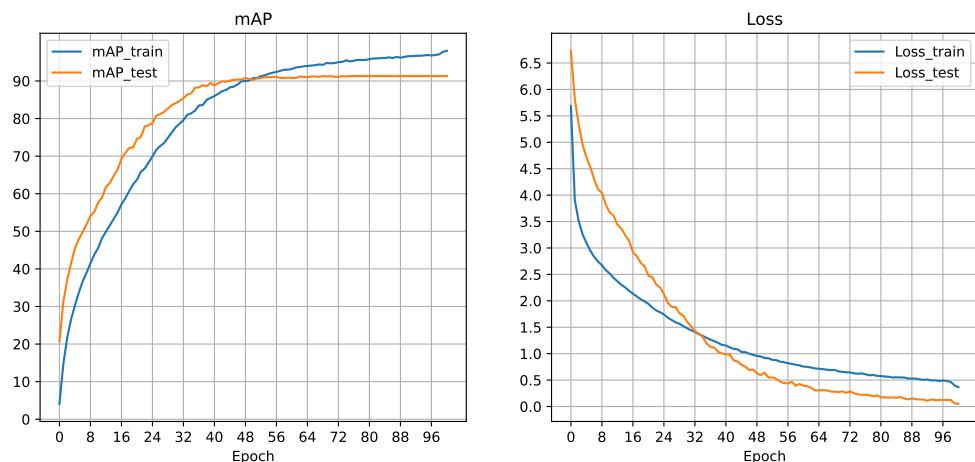
**Hình 4.20:** Mạng trích xuất đặc trưng: ResNet 101 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có.



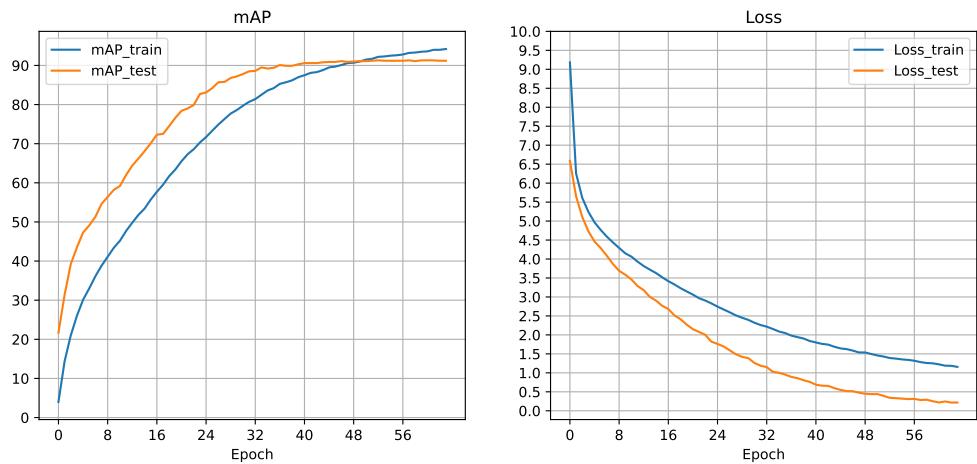
**Hình 4.21:** Mạng trích xuất đặc trưng: EfficientNet B0 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có.



**Hình 4.22:** Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có.



**Hình 4.23:** Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 243 - Số lớp Encoder: 3 - Che nhãn huấn luyện: có.



**Hình 4.24:** Mạng trích xuất đặc trưng: MobileNet V2 - Hàm kích hoạt: sigmoid - Lượng nhãn biết trước: 0 - Số lớp Encoder: 2 - Che nhãn huấn luyện: không.

**Nhận xét:** Nhìn chung, các mô hình càng phức tạp thì hội tụ nhanh hơn so với các mô hình ít phức tạp hơn, nhưng giá trị mAP của chúng vẫn ở ngưỡng 91.3 trên tập đánh giá dù có hội tụ nhanh hơn hay chậm hơn.

## 4.5 Kết quả mô hình kết hợp

### 1) Bảng tham số và các kết quả tương ứng

**Bảng 4.12:** Bảng các tham số chung của mô hình kết hợp.

Hàm kích hoạt	Số lớp Encoder	Che nhãn huấn luyện	Lượng nhãn biết trước	Trạng thái nhãn
sigmoid	3	có	0	tổng

**Bảng 4.13:** Bảng các tham số và kết quả tương ứng của mô hình kết hợp khi sử dụng các tham số ở bảng 4.12.

mạng trích xuất đặc trưng	Hàm mất mát	Kết quả kiểm tra
EfficientNet B0	ROLE	54.9
MobileNet V2	AN-LS	51
MobileNet V2	HU	47

**Nhận xét:** Ở các cài đặt trên mô hình kết hợp cho thấy rằng khi sử dụng hàm mất mát ROLE cho kết quả tốt nhất so với các hàm mất mát AN-LS và Huber. Nhưng kết quả tốt nhất này lại kém hơn nhiều so với những kết quả trên mô hình C-Tran khi chưa kết hợp.

## Chương 5

# Kết luận và hướng phát triển

Tóm lại, các vấn đề thường gặp đối với dữ liệu trong bài toán phân lớp đa nhãn gồm: chênh lệch nhãn trên các bức ảnh khác nhau, chênh lệch số lượng ảnh ở mỗi loại nhãn và sự bao hàm lẫn nhau về nghĩa ở các nhãn khác nhau trong tập nhãn. Để giải quyết vấn đề này, chúng tôi đã tìm hiểu và tìm cách cải thiện các mô hình gồm: Đơn nhãn dương và C-Tran.

Ý tưởng chính của mô hình đơn nhãn dương là đưa tập nhãn của mỗi bức ảnh (vốn có nhiều nhãn dương và nhãn âm) về bối cảnh chỉ có một nhãn dương duy nhất. Từ đó xây dựng các hàm mất mát giúp mô hình có sẵn học tốt hơn trên bối cảnh mới này. Quá trình huấn luyện được thực hiện trên nhiều chế độ khác nhau gồm: học tuyến tính, học đầu cuối và học chuyển giao. Chúng tôi cải thiện mô hình theo 3 hướng chính gồm: thay đổi mạng trích xuất đặc trưng, thay đổi hàm kích hoạt và bộ phận ước lượng nhãn tương ứng, bổ sung và thử nghiệm các hàm mất mát mới nhằm giải quyết các vấn đề liên quan đến dữ liệu.

Còn đối với C-Tran, ý tưởng chính của mô hình là tận dụng sức mạnh của khối Encoder từ mô hình Transformer để tìm ra sự tương quan, phụ thuộc giữa các đặc trưng từ hình ảnh và nhãn của chúng. Đồng thời, mô hình còn đề xuất phương pháp che nhãn trong quá trình huấn luyện bằng

cách sử dụng một sơ đồ mã hóa để đánh dấu trạng thái, phương pháp này giúp cho việc phân biệt và làm nổi bật sự tương quan nhau một cách rõ ràng hơn. Chúng tôi cũng tiến hành cải thiện mô hình theo các hướng như: thay đổi mạng trích xuất đặc trưng, thay đổi hàm kích hoạt, thay đổi số lớp Encoder của mô hình.

Tiếp đó, chúng tôi cũng đề xuất đến việc kết hợp cả 2 mô hình Học Đơn nhau dương và C-Tran lại với nhau với mục tiêu là kết hợp lại những điểm mạnh của 2 mô hình. Cuối cùng, chúng tôi đề xuất thêm những cải tiến trong việc nâng cao chất lượng của dữ liệu đầu vào bằng việc sử dụng thông tin phân vùng của đối tượng trong hình ảnh ở bước xử lý dữ liệu đầu vào để loại bỏ những thành phần dư thừa trong hình ảnh, giúp cho mô hình của chúng tôi có thể tập trung chú trọng vào đối tượng chính trong hình ảnh nhằm cải thiện chất lượng cho quá trình học của mô hình. Và cũng do sự giới hạn về tài nguyên mà chúng tôi đã không thể thử nghiệm cho và ra các kết quả cuối cùng của tất cả các đề xuất. Tuy nhiên, chúng tôi hy vọng điều này sẽ đóng góp cho những nền tảng mô hình khác sau này.

Chúng tôi chưa giải quyết tối ưu vấn đề ngữ nghĩa nhau ở mô hình Đơn nhau dương, cũng như giải quyết vấn đề cải hiện độ phân giải của ảnh đầu vào. Về hướng phát triển tiếp theo, chúng tôi có thể cân bằng lại dữ liệu bằng cách sử dụng các kỹ thuật như tăng số lượng mẫu của lớp thiểu số, giảm số lượng mẫu của lớp đa số. Ngoài ra, có thể sử dụng các phương pháp khác điều chỉnh trọng số trong hàm mất mát và các độ đo để giảm thiểu ảnh hưởng của sự mất cân bằng dữ liệu cho bài toán phân lớp đa nhau.

# Tài liệu tham khảo

## Tiếng Anh

- [1] Borgelt, Christian. “An Implementation of the FP-growth Algorithm”. In: *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*. 2005, pp. 1–5.
- [2] Chen, Zhao-Min et al. “Multi-label image recognition with graph convolutional networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5177–5186.
- [3] Cole, Elijah et al. “Multi-label learning from single positive labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 933–942.
- [4] Dosovitskiy, Alexey et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).
- [5] Grill, Jean-Bastien et al. “Bootstrap your own latent-a new approach to self-supervised learning”. In: *Advances in neural information processing systems* 33 (2020), pp. 21271–21284.
- [6] He, Kaiming et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

- [7] Hu, Jie, Shen, Li, and Sun, Gang. “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.
- [8] Kim, Taehyeon et al. “Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation”. In: *arXiv preprint arXiv:2105.08919* (2021).
- [9] Lanchantin, Jack et al. “General multi-label image classification with transformers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16478–16488.
- [10] Lee, Ching-Pei and Lin, Chih-Jen. “A study on L2-loss (squared hinge-loss) multiclass SVM”. In: *Neural computation* 25.5 (2013), pp. 1302–1323.
- [11] Li, Xiaoli and Liu, Bing. “Learning to classify texts using positive and unlabeled data”. In: *IJCAI*. Vol. 3. 2003. Citeseer. 2003, pp. 587–592.
- [12] Lin, Tsung-Yi et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.
- [13] Ridnik, Tal et al. “Asymmetric loss for multi-label classification”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 82–91.
- [14] Sandler, Mark et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.
- [15] Taggart, Robert J. “Point forecasting and forecast evaluation with generalized Huber loss”. In: *Electronic Journal of Statistics* 16.1 (2022), pp. 201–231.
- [16] Tan, Mingxing and Le, Quoc. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

- [17] Tan, Mingxing et al. “Mnasnet: Platform-aware neural architecture search for mobile”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2820–2828.