TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KỲ

MÔN DEEP LEARNING

Người hướng dẫn: **THẦY LÊ ANH CƯỜNG**

Người thực hiện: **TRẦN ANH KIỆT – 52300124**

TRẦN THẢO MY – 52300129

Nhóm môn học**:** **02**

Năm học**:** **2025-2026**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KỲ

MÔN DEEP LEARNING

Người hướng dẫn: **THẦY LÊ ANH CƯỜNG**

Người thực hiện: **TRẦN ANH KIỆT – 52300124**

TRẦN THẢO MY – 52300129

Nhóm môn học**:** **02**

Năm học**:** **2025-2026**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Lời đầu tiên, chúng em muốn dành những lời tri ân chân thành đến Trường Đại học Tôn Đức Thắng vì đã đưa môn học "Nhập môn học sâu" vào chương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Lê Anh Cường - người đã không chỉ là giảng viên mà còn là người hướng dẫn tận tâm, truyền đạt những kiến thức quý báu cho chúng em suốt thời gian học tập vừa qua.

Trong những buổi học của Thầy, chúng em đã được trải nghiệm sự chuyên nghiệp và tận tâm của người hướng dẫn. Nhờ sự giảng dạy và hướng dẫn kỹ lưỡng của thầy, mà chúng em đã nắm vững những khái niệm, thuật toán, cũng như cách tiếp cận bài toán một cách sáng tạo. Thầy không chỉ truyền đạt kiến thức mà còn khích lệ chúng em phát triển tư duy logic và kỹ năng giải quyết vấn đề.

Môn học "Nhập môn học sâu" không chỉ thú vị mà còn rất bổ ích và thực tế. Chúng em tin rằng những kiến thức chúng em đang học sẽ là hành trang quan trọng, giúp chúng em tự tin bước vào lĩnh vực Công nghệ thông tin sau này. Dù chúng em đã nỗ lực hết mình, nhưng do vẫn còn những hạn chế và khó khăn, bài tiểu luận của chúng em không tránh khỏi những thiếu sót. Chúng em kính mong thầy xem xét và góp ý để bài tiểu luận trở nên hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!”

TP. Hồ Chí Minh, ngày 24 tháng 11 năm 2025

Tác giả

Trần Anh Kiệt

Trần Thảo My

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH

TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của Thầy Lê Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 24 tháng 11 năm 2025

Tác giả

Trần Anh Kiệt

Trần Thảo My

MỤC LỤC

DANH MỤC HÌNH ẢNH

Phần 1: Giới thiệu tổ chức Drivendata

1.1 Tổng quan

DrivenData là một nền tảng thi đấu khoa học dữ liệu (Data Science Competition Platform) trực tuyến, tương tự như Kaggle, nhưng có một sứ mệnh đặc biệt và khác biệt.

1.2. Sứ mệnh: "Data Science for Social Good"

- Khác với các nền tảng thương mại tập trung vào bài toán kinh doanh cho các tập đoàn lớn, slogan và kim chỉ nam của DrivenData là "Khoa học dữ liệu vì lợi ích xã hội".
- Họ chuyên tổ chức các cuộc thi nhằm giải quyết những thách thức khó khăn nhất của thế giới trong các lĩnh vực: Phát triển quốc tế, Y tế, Giáo dục, Nghiên cứu, Bảo tồn thiên nhiên và Dịch vụ công.
- Họ đóng vai trò là cầu nối giữa các tổ chức phi lợi nhuận, các tổ chức NGO (như NASA, World Bank, The Nature Conservancy) – những nơi có dữ liệu nhưng thiếu nhân lực kỹ thuật – với cộng đồng các nhà khoa học dữ liệu tài năng trên toàn thế giới.

1.3. Cách thức hoạt động

- DrivenData nhận dữ liệu thô từ các tổ chức đối tác.
- Họ chuẩn hóa dữ liệu, xác định bài toán và tổ chức thành cuộc thi.
- Cộng đồng lập trình viên/nhà khoa học dữ liệu tham gia giải quyết.
- Mô hình tốt nhất sẽ được chuyển giao lại cho tổ chức đối tác để ứng dụng vào thực tế (Open Source).

1.4 Các cuộc thi nổi bật

DrivenData đã tổ chức thành công nhiều cuộc thi có tiếng vang lớn, giải quyết đa dạng các vấn đề toàn cầu. Một số cuộc thi tiêu biểu của tổ chức:

- Lĩnh vực Cơ sở hạ tầng & Nhân đạo

- Pump it Up: Data Mining the Water Table (Dự án kinh điển nhất của DrivenData)
  - Đối tác: Bộ Nước Cộng hòa Tanzania & Taarifa.
  - Vấn đề: Tanzania có hàng ngàn điểm cung cấp nước sạch, nhưng rất nhiều trong số đó bị hỏng mà không ai biết, gây lãng phí và thiếu nước sinh hoạt.
  - Mục tiêu: Sử dụng dữ liệu về loại máy bơm, vị trí địa lý, độ sâu giếng, đơn vị thi công... để dự đoán xem máy bơm nào đang hoạt động, máy nào cần sửa chữa, và máy nào đã hỏng hoàn toàn.
  - Ý nghĩa: Giúp chính phủ tối ưu hóa chi phí bảo trì và đảm bảo nguồn nước sạch cho người dân.

- Richter's Predictor: Modeling Earthquake Damage
  - Đối tác: Kathmandu Living Labs (Nepal).
  - Bối cảnh: Sau trận động đất kinh hoàng năm 2015 tại Nepal (Gorkha earthquake).
  - Mục tiêu: Dựa trên dữ liệu khảo sát hàng triệu tòa nhà, mô hình cần dự đoán mức độ thiệt hại (nhẹ, trung bình, phá hủy hoàn toàn) của từng tòa nhà dựa trên cấu trúc, vật liệu và vị trí.
  - Ý nghĩa: Hỗ trợ chính phủ và các tổ chức cứu trợ phân bổ nguồn lực tái thiết nhanh chóng và hiệu quả hơn sau thảm họa.
- Lĩnh vực Y tế & Sức khỏe cộng đồng: DengAI: Predicting Disease Spread
  - Đối tác: Các cơ quan y tế công cộng.
  - Vấn đề: Sốt xuất huyết là căn bệnh lan truyền do muỗi, bùng phát mạnh ở các vùng nhiệt đới.
  - Mục tiêu: Sử dụng dữ liệu về thời tiết, khí hậu, lượng mưa và lịch sử dịch bệnh để dự đoán số lượng ca nhiễm sốt xuất huyết tại các thành phố cụ thể (như San Juan và Iquitos) trong tương lai gần.
  - Ý nghĩa: Cung cấp hệ thống cảnh báo sớm, giúp các bệnh viện và cơ quan y tế chuẩn bị thuốc men và nhân lực trước khi dịch bùng phát.
- Lĩnh vực Công nghệ & Mạng xã hội: The Hateful Memes Challenge

- Đối tác: Facebook AI (nay là Meta AI)
- Vấn đề: Các hệ thống AI truyền thống rất giỏi nhận diện văn bản thù ghét hoặc hình ảnh bạo lực riêng lẻ. Tuy nhiên, "Meme" (ảnh chế) kết hợp cả ảnh và chữ, đôi khi ảnh bình thường + chữ bình thường nhưng ghép lại lại mang ý nghĩa thù địch rất khó phát hiện.
- Mục tiêu: Xây dựng mô hình đa phương thức (Multimodal) để phát hiện các meme chứa nội dung thù ghét.
- Ý nghĩa: Với giải thưởng lên tới 100.000 USD, cuộc thi này đã thúc đẩy giới hạn của AI trong việc hiểu ngữ cảnh, góp phần làm sạch môi trường mạng xã hội.

- Lĩnh vực Khí hậu & Vũ trụ: NASA Airathon: Predict Air Quality
  - Đối tác: NASA (Cơ quan Hàng không và Vũ trụ Hoa Kỳ).
  - Mục tiêu: Sử dụng dữ liệu vệ tinh vệ tinh quan sát Trái Đất của NASA để dự đoán nồng độ khí thải NO2 và các hạt bụi mịn (PM2.5) với độ phân giải cao.
  - Ý nghĩa: Giúp theo dõi ô nhiễm không khí chính xác hơn, phục vụ cho các chính sách bảo vệ môi trường toàn cầu.

Phần 2: Giới thiệu cuộc thi.

2.1 Thông Tin Chung (General Information)

- Tên cuộc thi: Conservision Practice Area: Image Classification
- Nền tảng tổ chức: DrivenData (Nền tảng thi đấu Khoa học dữ liệu uy tín, chuyên về các bài toán có tác động xã hội).
- Mã cuộc thi (Competition ID): 87
- Lĩnh vực: Computer Vision (Thị giác máy tính), Ecology (Sinh thái học), Wildlife Conservation (Bảo tồn động vật hoang dã).
- Trạng thái: Cuộc thi dạng "Practice" (Thực hành) - Mở dài hạn để cộng đồng học tập và rèn luyện kỹ năng, không có giải thưởng tiền mặt nhưng có bảng xếp hạng (Leaderboard) để so sánh hiệu quả mô hình.

2.2 Quy mô và Phạm vi tổ chức

Quy mô: Toàn cầu (Global). Bất kỳ ai có kết nối internet đều có thể tham gia.
Số lượng người tham gia: Tính đến thời điểm hiện tại, cuộc thi đã thu hút hàng nghìn lượt nộp bài (submissions) từ các Data Scientist, sinh viên và nghiên cứu sinh trên khắp thế giới. Đây là một trong những bài thi nhập môn phổ biến nhất trên nền tảng DrivenData.
Dữ liệu: Mặc dù là bản "Practice" (đã được thu gọn), tập dữ liệu vẫn đảm bảo tính đại diện với hàng chục nghìn bức ảnh, đủ để huấn luyện các mô hình Deep Learning phức tạp.

2.3 Hình thức tổ chức thi đấu

Trực tuyến 100%: Mọi quy trình từ đăng ký, tải dữ liệu đến nộp kết quả đều diễn ra trên website DrivenData.

Cơ chế Leaderboard (Bảng xếp hạng):
Người chơi tải training set (có nhãn) về để huấn luyện mô hình.
Người chơi chạy mô hình trên test set (không có nhãn) để ra kết quả dự đoán.
Người chơi upload file .csv kết quả lên hệ thống.
Hệ thống tự động chấm điểm (Scoring) dựa trên thuật toán Log Loss và cập nhật thứ hạng ngay lập tức.
Tính chất: Vì là "Practice Area", cuộc thi không có giải thưởng tiền mặt (Prize Money). Mục đích chính là:
Giúp người mới bắt đầu làm quen với quy trình xử lý ảnh thực tế.
Cho phép các chuyên gia thử nghiệm các kiến trúc mô hình mới (SOTA models).
Xây dựng cộng đồng chia sẻ kiến thức (thông qua diễn đàn thảo luận của cuộc thi).

2.4 Bối Cảnh & Ý Nghĩa Thực Tiễn (Background & Context)

- Địa điểm thu thập dữ liệu: Vườn quốc gia Taï (Tai National Park) tại Côte d'Ivoire (Bờ Biển Ngà). Đây là một trong những khu rừng mưa nhiệt đới nguyên sinh cuối cùng ở Tây Phi.
- Vấn đề: Các nhà bảo tồn sử dụng hệ thống "bẫy ảnh" (camera traps) được kích hoạt bằng cảm biến chuyển động hoặc nhiệt để giám sát động vật mà không làm phiền chúng. Tuy nhiên, hệ thống này tạo ra hàng triệu bức ảnh.
- Nhu cầu: Việc phân loại thủ công tốn quá nhiều thời gian và nhân lực. Một mô hình học máy tự động có thể lọc bỏ các ảnh "rỗng" (không có động vật) và phân loại nhanh các loài vật, giúp các nhà nghiên cứu tập trung vào việc phân tích chuyên sâu và bảo vệ hệ sinh thái.
- Kết quả và các giải pháp từ cộng đồng đóng góp trực tiếp vào việc hoàn thiện công cụ Conservision. Khi các thuật toán này được tích hợp vào phần mềm quản lý, nó giúp các nhân viên kiểm lâm và nhà sinh thái học tại Châu Phi giám sát hiệu quả hơn các loài động vật quý hiếm (như Báo hoa mai hay các loài linh dương) trước nguy cơ săn bắt trộm và biến đổi khí hậu.

2.5 Chi Tiết Về Dữ Liệu (Dataset Specification)

Đây là phần quan trọng nhất cho việc xây dựng mô hình. Dữ liệu được cung cấp dưới dạng các tệp hình ảnh và file nhãn (labels).

- Đặc điểm hình ảnh (Input)
  - Định dạng: Ảnh màu (RGB), thường là đuôi .jpg.
  - Nguồn gốc: Ảnh chụp thực tế từ bẫy ảnh trong rừng.
  - Thách thức về chất lượng ảnh:
  - Ánh sáng: Thay đổi mạnh giữa ngày (ảnh màu) và đêm (ảnh hồng ngoại đen trắng).
  - Môi trường: Nền rừng rậm rạp, nhiều cây cối che khuất chủ thể (occlusion).
  - Chuyển động: Ảnh có thể bị mờ (motion blur) do động vật di chuyển nhanh.

- Kích thước chủ thể: Động vật có thể ở rất xa (nhỏ) hoặc quá gần (chỉ thấy một phần lông/cơ thể).

- Các nhãn phân loại (Classes/Labels): Mô hình cần phân loại hình ảnh vào một trong 8 lớp (classes) sau đây. Lưu ý tên lớp (class names) phải chính xác tuyệt đối để khớp với định dạng nộp bài:
  - antelope_duiker: Các loài linh dương nhỏ (duikers).
  - bird: Các loài chim.
  - blank: Ảnh trống (không có động vật, thường do gió làm rung cây kích hoạt bẫy ảnh).
  - civet_genet: Các loài cầy hương, cầy genets.
  - hog: Lợn rừng.
  - leopard: Báo hoa mai (loài thú ăn thịt quan trọng trong hệ sinh thái này).
  - monkey_prosimian: Khỉ và các loài bán hầu (prosimians).
  - rodent: Các loài gặm nhấm.

2.6 Yêu cầu bài toán và phương pháp đánh giá

- Nhiệm vụ: Bài toán thuộc loại Multi-class Image Classification (Phân loại ảnh đa lớp).
  - Input: Một hình ảnh X.
  - Output: Một vector xác suất P = [p_1, p_2,…, p_8] tương ứng với 8 lớp, sao cho tổng xác suất $\sum p_i = 1$.
- Chỉ số đánh giá (Evaluation Metric)
  - Cuộc thi sử dụng chỉ số Multi-class Logarithmic Loss (Log Loss).
  - Công thức:

$$\text{Log Loss} = -\frac{1}{N}\sum_{i-1}^{N}\sum_{j-1}^{M} y_{ij}\log(p_{ij})$$

- Trong đó:
  - N: Số lượng ảnh trong tập kiểm tra.

- M: Số lượng lớp (8 lớp).
- yij: Biến nhị phân (1 nếu ảnh I thuộc lớp j, ngược lại là 0).
- pij: Xác suất mô hình dự đoán ảnh I thuộc lớp j.
- Ý nghĩa: Log Loss trừng phạt rất nặng các dự đoán sai nhưng lại có độ tự tin cao. Do đó, mô hình cần phải được cân chỉnh (calibrated) tốt về mặt xác suất, không chỉ đơn thuần là đoán đúng nhãn.

Phần 3: Phân tích chi tiết mô hình

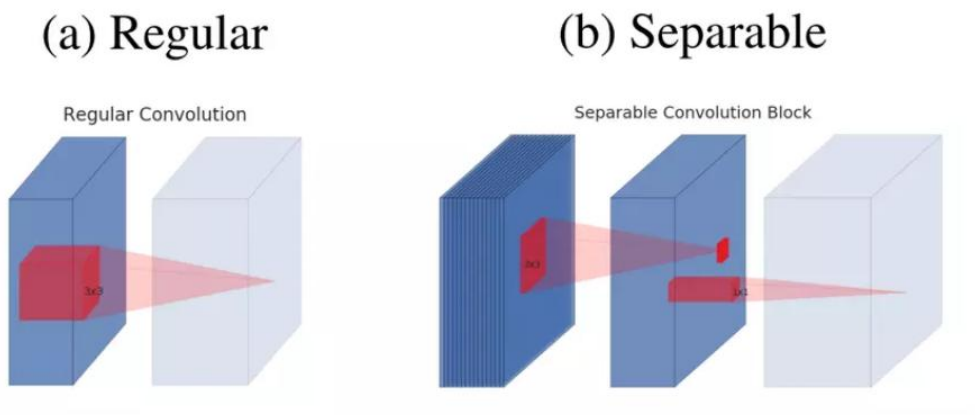
3.1 Giới thiệu tổng quát

- Là một mô hình mạng CNN (Convulutional Neutral Network).
- Ra đời năm 2017. Đến nay đã có 4 version và verison V4 mới nhất được ra đời vào năm 2024.
- Ra đời khi sự thế giới đòi hỏi mô hình CNN càng sâu, càng nhiều tham số, càng phức tạp để việc huấn luyện càng tốt. Nhưng để làm được điều đó đòi hỏi sự khắc nghiệt về phần cứng. Thực tế cũng có nhiều mô hình có hàng trăm triệu đến hàng tỷ tham số.

→ Đây là một mô hình nhỏ, nhẹ, nhanh nhưng có độ chính xác có thể chấp nhận được với gần 6 triệu tham số, chuyên xử lý dữ liệu hình ảnh.

3.2 Đặc điểm MobileNets

Làm thế nào MobileNets có thể rút gọn về vài triệu tham số nhưng vẫn giữ được độ chính xác → câu trả lời là sử dụng một cơ chế là **Depthwise Separable**.



Hình 1: Cơ chế hoạt động của Regular Convolution so với Separeble Convolution Block

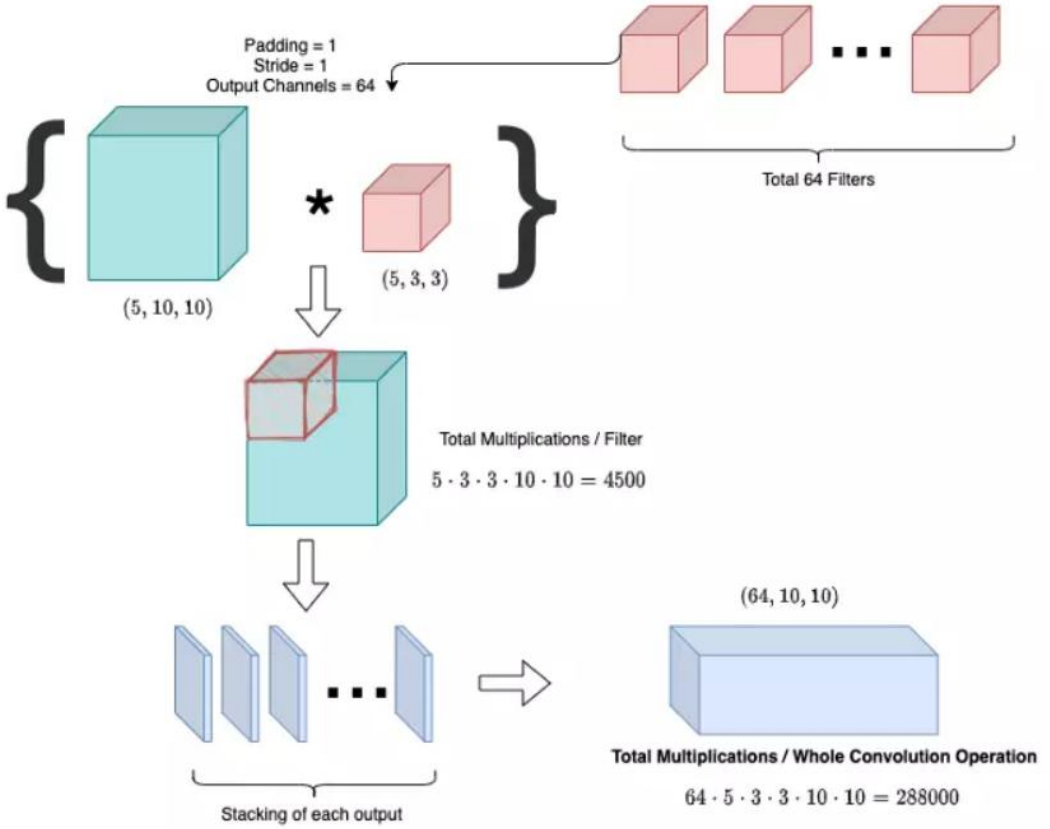
Với kiến trúc truyền thống, Regular Convoution sẽ thực hiện phép tính trên toàn bộ không gian và toàn bộ độ sâu của dữ liệu đầu vào cùng một lúc. Điều này tạo ra khối lượng tính toán lớn và số lượng tham số nhiều, gây nặng nề cho mô hình.

Để khắc phục được điều đó, cơ chế Depthwise Convolution chỉ sử dụng các kernel áp dụng riêng lẻ lên từng kênh đầu vào. Với mục đích chỉ học các đặc trưng không gian của từng kênh mà không quan tâm đến sự liên kết của các kênh. Sau đó chỉ cần Pointwise Convolution để trộn thông tin từ các kênh lại với nhau.

Tóm lại, cơ chế Depthwise Separeble có ý nghĩa:

- Giảm khối lượng tính toán: Về mặt toán học, phương phapps Separeble giúp giảm só lượng phép nhân và cộng từ 8-9 lần so với Regular Convolution.
- Giản số lượng tham số: Giúp mô hình nhẹ hơn, giảm nguy cơ quá hớp (overfitting) khi dữ liệu huấn luyện ít.

3.2.1 CNN thông thường



Hình 2: Minh họa số lượng phép tính nhân trong tích chập thông thường với 65 bộ lọc

- Giả sử, đầu vào của chúng ta là 1 feature map 5x10x10. Bộ lọc của chúng ta có kích thước 5x3x3. Như vậy, sau khi thực hiện phép nhân tích chập trên toàn bộ feature map (padding = 1, stride=1) ta thu được kết quả 1x10x10.
- Với mỗi điểm trên feature map, vì filter là 3x3, nên cần thực hiện 3x3 = 9 phép tính nhân.
- Số tính toán đã thực hiện là 5 x (3 x 3 x 10 x 10) = 4500.
- Sau khi thực hiện phép nhân, ta cộng nó lại.
- Trong một lớp convolutional, ta sẽ sử dụng nhiều bộ lọc. Ở đây ta có 64 bộ lọc, vậy tổng số tính toán cần thực hiện là: 64 x 4500 = 288000 phép tính nhân.
- Như vậy, với:

- M, N là số input, output channels.
- D_f là chiều của input feature map ($M * D_f * D_f$).
- D_k là chiều của kernel.

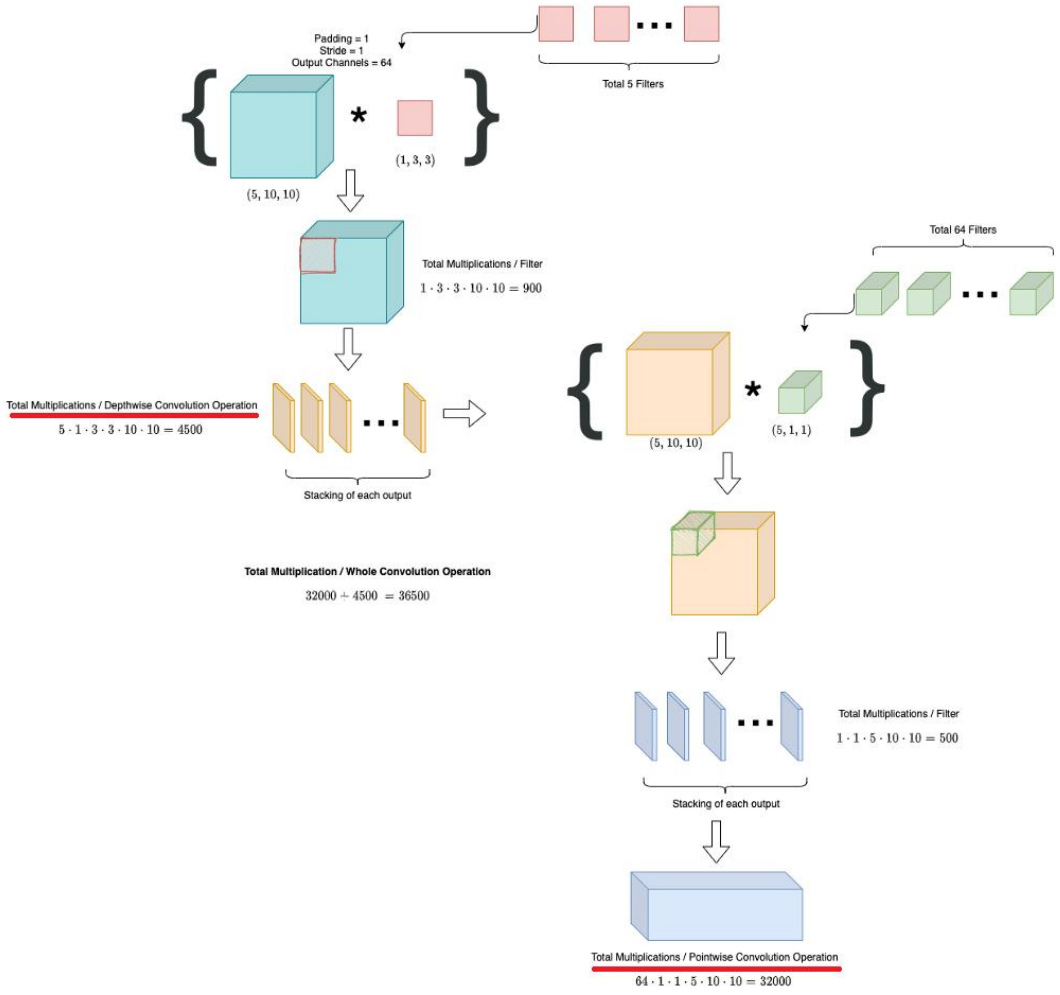
- Thì tổng số phép nhân cần thực hiện là $M*N*D_f^2* D_k^2$

→Với ảnh độ phân giải lớn hiện nay 3x1024x1024, số lượng tính toán là cực kỳ lớn.

3.2.2 Cơ chế Depthwise Separable Convolutions

Hiện nay có khá nhiều kiến trúc CNN với mục đích gọn nhẹ áp dụng cơ chế này, có thể kể đến **MobileNets, ShuffleNet, EffNet...**

Depthwise Separable Convolutions chia CNN cơ bản ra làm hai phần: **Deepwise Convolution** và **Pointwise Convolution**.



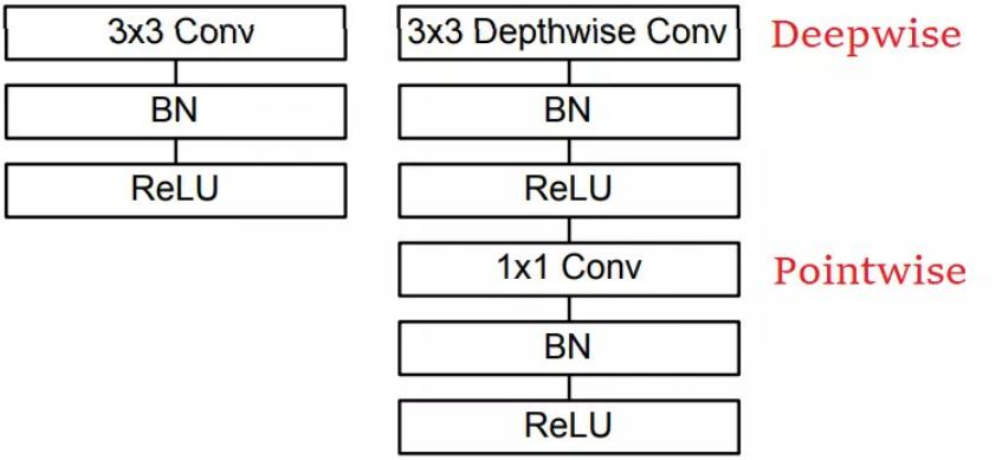
Hình 3: Quy trình của Depthwise Separeble Convolution

- Cơ chế của phần Deepwise Convolution
  - Thay vì nhân một mớ tất cả ($M*N*D_f^2* D_k^2$) thì ta tách ra một chút.
  - Đầu tiên, ta vẫn thực hiện như standard CNN, thực hiện nhân tích chập 5x10x10 với bộ filter giờ chỉ còn là 1x3x3, tương tự với 5 filters như thế, stack nó lại, kết quả thu được output là 5x10x10.
  - Số phép nhân đã thực hiện vẫn là $M*N*D_f^2* D_k^2$ = 5x10x10x3x3 = 4500 phép tính.
  - Tuy nhiên, có thể thấy khác với standard CNN, ở deepwise convolutions, số lượng channel của chúng ta vẫn giữ nguyên, nghĩa là vẫn 5x10x10. (thực hiện phép tích chập một cách rời rạc (separable) trên từng channel)
- Cơ chế của phần Pointwise Convolution
  - Tiếp theo chúng ta sử dụng kết quả từ bước deepwise convolution. Ở bước pointwise này, ta chỉ sử dụng bộ có kích thước là 1x1. Đồng thời số lượng bộ lọc bằng số channel mà ta muốn thu được. Ta muốn tăng lên 64 channel, vậy hãy sử dụng 64 bộ filters.
  - Kích thước ko đổi, chỉ số channels đổi.
  - Như vậy, số phép nhân cần tính chỉ là 5 x 64 x 10 x 10 = 32000 phép nhân.
  - Vậy tổng số phép nhân cần tính là: 4500 + 32000 = 36500 phép nhân.
  - Như vậy, số lượng tính toán đã giảm 8 lần so với standard CNN.

→ Vậy tổng kết, số lượng tính toán đã giảm được là:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

3.2.3 Sự khác biệt giữa CNN thông thường và MobileNets

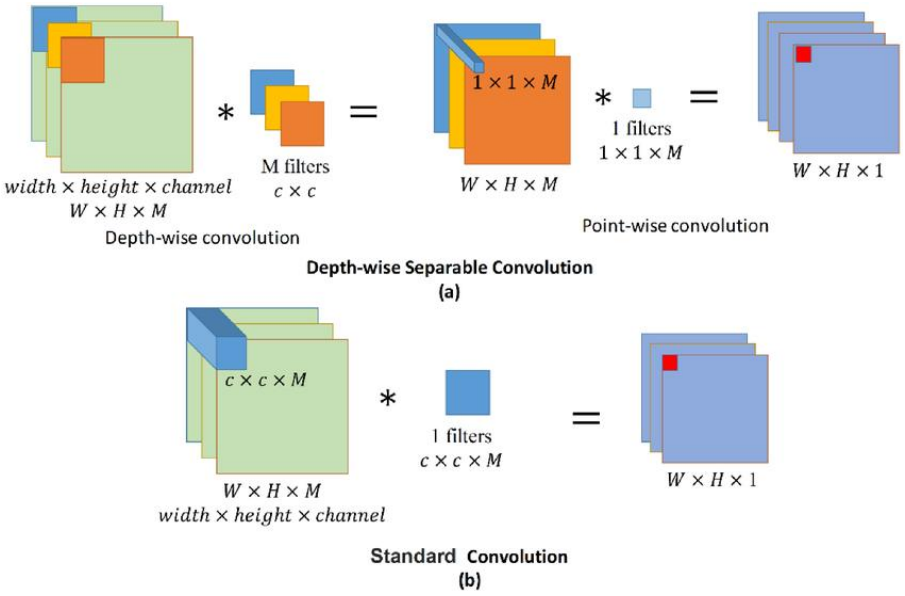


Hình 4: So sánh cấu trúc khối Tích chập tiêu chuẩn và khối Tích chập tách biệt theo chiều sâu trong MobileNet.

- Với phương pháp CNN thông thường, sẽ bao gồm 3 khối chính:
  - 3 x 3 Conv: Thực hiện tích chập (vừa không gian vừa chiều sâu).
  - BN (Batch Normalization): Chuẩn hóa dữ liệu để giúp mạng học ổn định hơn.
  - ReLU: Hàm kích hoạt phi tuyến tính.

→ Nhận xét: Chỉ là một chuỗi duy nhất: Tích chập → Chuẩn hóa → Kích hoạt.

- Đối với cấu trúc của MobileNets, chia khối chuẩn thành 2 giai đoạn riêng biệt, mỗi giai đoạn đều có quy trình xử lý đầy đủ.
  - Giai đoạn Depthwise:
    - 3 x 3 Depthwise Conv: Chỉ lọc các đặc trưng không gian.
    - BN + ReLU: Ngay lập tức chuẩn hóa và kchs hoạt phi tuyến tính sau bước này.
  - Giai đoạn Pointwise:
    - 1 x 1 Conv: Trộn các kênh lại với nhau.
    - BN + ReLU: Tiếp tục chuẩn hóa và kích hoạt lần thứ hai.



Hình 5: Cơ chế xử lý khối dữ liệu đầu vào của hai phương pháp tích chập

Dựa vào hình minh họa, ta thấy mỗi phương pháp xử lý một khối dữ liệu đầu vào kích thước W x H x M để tạo ra đầu ra.

- CNN thông thường:
  - Cơ chế "Một bước": Đây là cách làm truyền thống.
  - Bộ lọc (Filter): là khối đặc xanh dương nhỏ trong hình, có kích thước c x c x M. Bộ lọc này phải xử lý đồng thời cả 3 chiều.
    - Hệ quả: Do bộ lọc quá "dày" (do phải bao trùm toàn bộ M kênh đầu vào), số lượng tham số cần học rất lớn.
- Depthwise Separeble Convolution: Phương pháp này "chia để trị", tách khối dày thành 2 bước mỏng hơn:
  - Depthwise convolution:
    - ✧ Sử dụng M bộ lọc mỏng (kích thước c x c), mỗi bộ lọc chỉ áp dụng lên 1 kênh duy nhất.
    - ✧ Mục đích: Chỉ lọc đặc trưng không gian, chưa quan tâm đến mối liên hệ giữa các kênh.
  - Pointwise convolution:
    - ✧ Sử dung bộ lọc 1 x 1 x M
    - ✧ Mục đích: Chiếu xuyên qua độ sâu để tổng hợp thông tin từ các kênh lại với nhau

3.2.4 Mở rộng của cơ chế MobileNets

- **Mở rộng Width Mutiplier: Thinner Models**
  - Width Multiplier α được thêm vào 1 layer, giờ đây số input channels và output channels là αM, alphaN, thay vì M và N.
  - Như vậy, chi phí tính toán của một phép tích chập là:

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F$$

(Với α ∈ (0,1], có nghĩa là α càng nhỏ thì chi phí tính toán càng giảm)

  - Các α thường dùng bao gồm: 0.25, 0.5, 0.75, 1.

- **Mở rộng Resolution Multiplier: Reduced Representation**
  - Tương tự như Width Multiplier α, Resolution Multiplier ρ cũng được thêm vào các layer để giảm resolution của ảnh đầu vào và các biểu diễn bên trong giữa các layers.

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F$$

(Với ρ ∈ (0,1], có nghĩa là ρ càng nhỏ thì chi phí tính toán càng giảm)

  - Các ρ được cung cấp để ảnh đầu vào có các resolutions bao gồm: 224, 192, 160, 128.

3.3 Mô hình MobileNetsV3

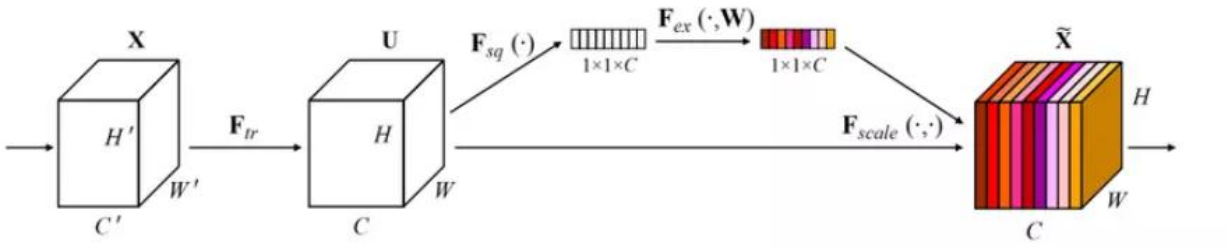
Đây là mô hình nhóm em sử dụng.
Điểm cải tiến chính là bổ sung Squeeze and Excite.

3.3.1 Giới thiệu về Squeeze and Excite

- Các hàm tích chập trích xuất thông tin dựa trên kết hợp cả thông tin không gian (HxW) và thông tin giữa các kênh. Do đó một ý tưởng đã nảy ra "Chúng ta tăng cường thông tin theo không gian rồi. Tại sao chúng ta lại

không tăng cường thông tin giữa chính các kênh? ". **Và đó là lý do mà mô hình Squeeze and Excitation sinh ra.**

- SE là một mạng khá đơn giản chỉ gồm vài lớp nhằm tăng cường thông tin giữa các kênh qua đó tăng chất lượng biểu diễn của mô hình CNN.
- SE làm được điều đó bằng cách sử dụng toàn bộ thông tin sau đó nhấn mạnh có chọn lọc vào từng kênh có đặc trưng quan trọng và ít chú ý vào những kênh ít quan trọng hơn.
- Khái niệm này khá giống với ý tưởng Self attention rất được hay dùng trong các bài toán xử lý ngôn ngữ tự nhiên cũng dùng đầu vào là chính nó để chú ý những thông tin quan trọng của chính nó.



Hình 6 : Kiến trúc mạnh Squeeze and Excitation.

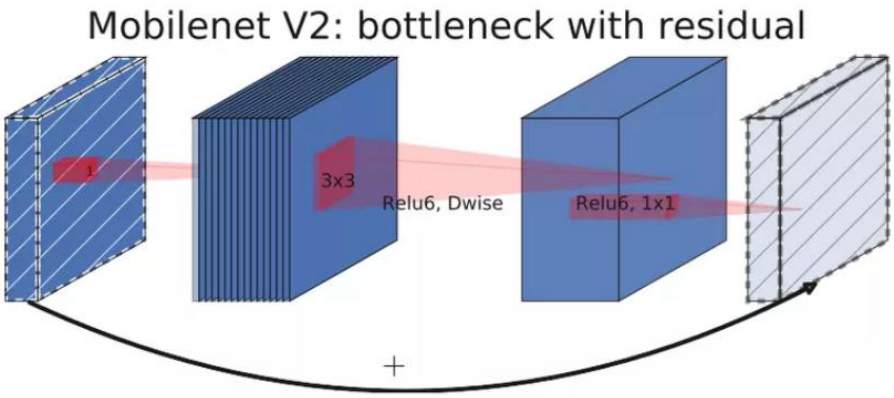
- Giải thích một số ký hiệu:
  - X: ảnh đầu vào có kích thước H' x W' x C'
  - F_{tr}: tập hợp các phép biến đổi: một vài lớp convolution, hoặc 1 stage của VGG, 1 block trong ResNet, ...
  - U: feature map hay đặc trưng được trính xuất từ ảnh đầu vào bởi các phép biển đổi F_{tr}.U có kích thước H x W x C.
- Cơ chế hoạt động của Squeeze and Excitation.

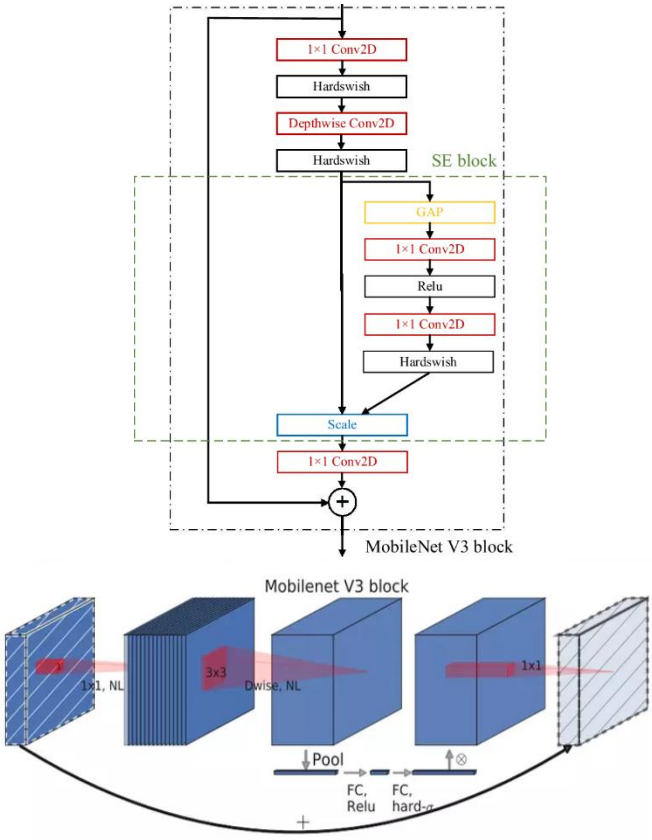
  Bước 1: Ảnh đầu vào X đi qua một tập hợp các phép biển đổi F_{tr} trích xuất ra bản đồ đặc trưng (features map) U.

  Bước 2: Feature map U (H x W x C) được đi qua hàm squeeze sinh ra một ma trận miêu tả đặc trưng của từng kênh (1 x 1 x C) bằng cách tổng hợp features map U theo chiều H và W. Ví dụ hàm squeeze ở đây có thể là global average pooling.

Bước 3: Theo sau hàm squeeze là hàm excitation. Hàm excitation đóng vai trò là cơ chế miêu tả sự phụ thuộc giữa các kênh với nhau. Hàm lấy đầu vào là ma trận tổng hợp đặc trưng của từng kênh được tính toán từ bước 2 qua một vài lớp biến đổi như convolution, hàm activation, ... và cuối cùng qua hàm gate sản sinh ra trọng số chú ý cho từng kênh. Những trọng số này sau đó được nhân với feature map U để tính ra output của khối SE. Output lúc này của khối SE chỉ còn chứa những thông tin thực sự quan trọng cho bài toán. **Hàm gate ở đây thường là hàm sigmoid.**

3.3.2 Cấu trúc mạng





Phần 4: Thí nghiệm mô hình

4.1 Khám phá dữ liệu

- Cấu trúc dữ liệu được cung cấp.

```
├── benchmark.ipynb
├── submission_format.csv
├── test_features
│   ├── ZJ000000.jpg
│   ├── ZJ000001.jpg
│   └── ...
├── test_features.csv
├── train_features
│   ├── ZJ016488.jpg
│   ├── ZJ016489.jpg
│   └── ...
├── train_features.csv
└── train_labels.csv
```

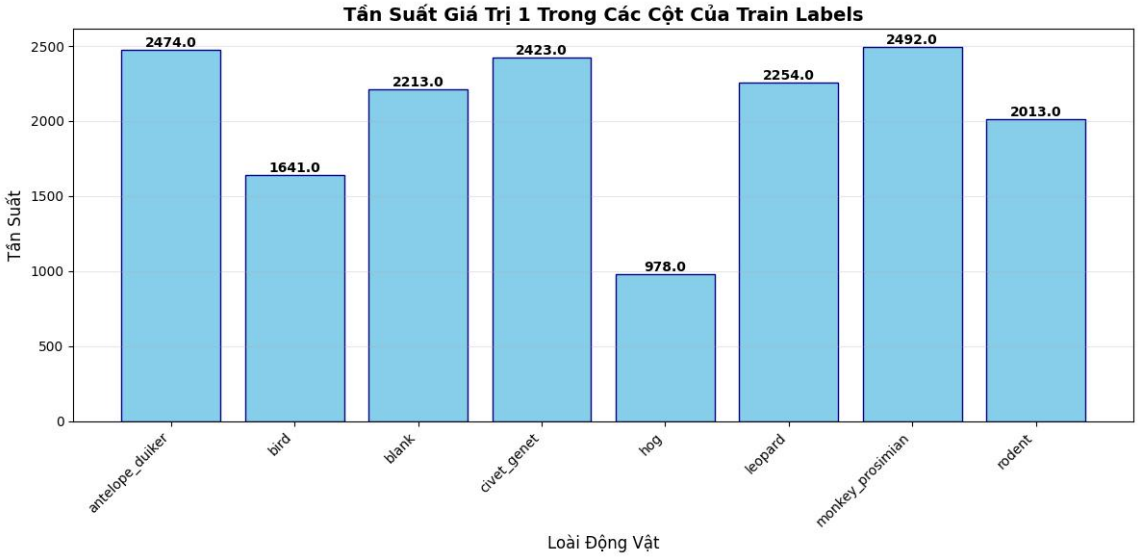
- Cấu trúc thư mục features.csv

id	filepath	site
ZJ000000	train_features/ZJ000000.jpg	S0120
ZJ000001	train_features/ZJ000001.jpg	S0069
ZJ000002	train_features/ZJ000002.jpg	S0009
ZJ000003	train_features/ZJ000003.jpg	S0008
ZJ000004	train_features/ZJ000004.jpg	S0036

- Cấu trúc thư mục labels.csv:

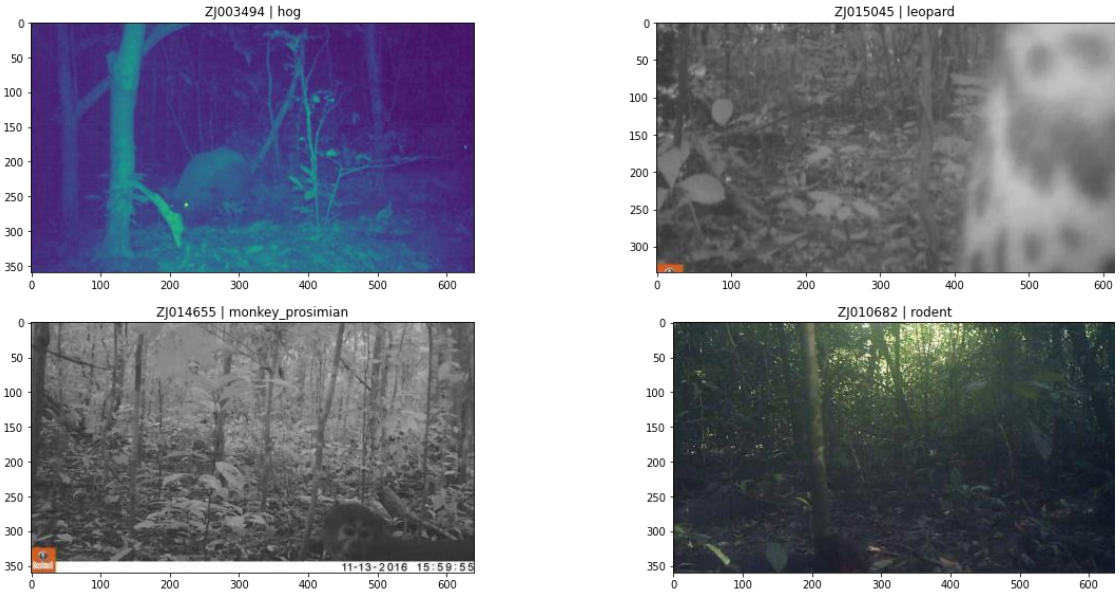
id	antelope_duiker	bird	blank	civet_genet	hog	leopard	monkey_prosimian	rodent
ZJ000000	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
ZJ000001	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
ZJ000002	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
ZJ000003	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
ZJ000004	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

- Số nhãn của loài và mức độ phân bố dữ liệu:



Hình 7: Biểu đồ phân bố tần suất giá trị của các nhãn

- Một số hình ảnh trong tập train:

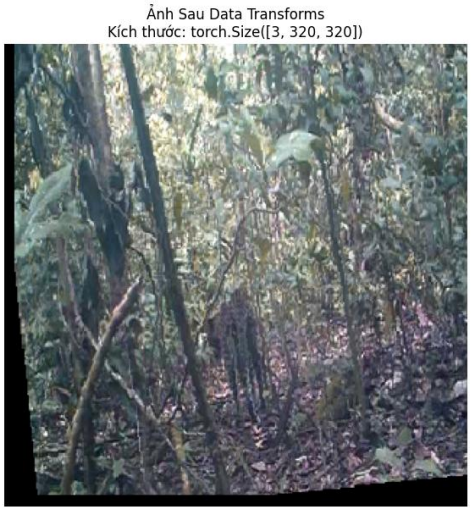


4.2 Tiền xử lý dữ liệu

4.2.1 Xử lý ảnh (Data Transform)

- Các bước biến đổi ảnh của tập train:

- So sánh ảnh ban đầu và kết quả



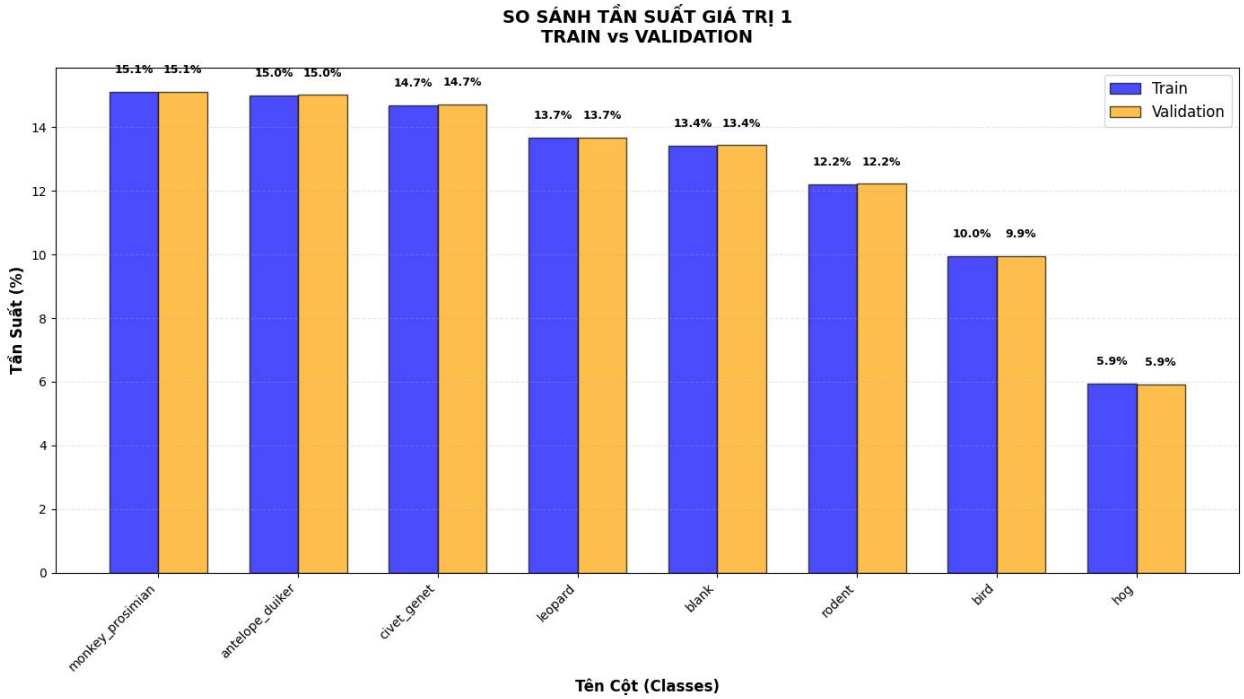
- Các bước biến đổi ảnh của tập val:

-

- Kết quả biến đổi của tập val:

4.2.2 Chia tập train, validation

Phân bố dữ liệu của tập train, validation:



Hình 8: Biểu đồ so sánh tần suất giá trị của các nhã trong tập train và validation

Nhận xét: Nhìn vào biểu đồ, có thể thấy chiều cao của các cột màu xanh (Train) và màu vàng (Validation) gần như bằng nhau ở tất cả các nhãn (classes). Chênh lệch phần trăm giữa hai tập là cực nhỏ (thường là 0% hoặc 0.1%).

Lợi ích: Việc phân phối đồng nhất giúp kết quả đánh giá trên tập Validation trở nên đáng tin cậy. Nếu mô hình hoạt động tốt trên tập Validation, ta có cơ sở vững chắc để tin rằng nó sẽ hoạt động tốt trên tập Test (với giả định tập Test cũng có cùng phân phối).

4.3 Mô hình huấn luyện

4.3.1 Các phương pháp chống overfitting

- Data Augmentation (Tăng cường dữ liệu)

```
train_transform = transforms.Compose([
    transforms.Resize((320, 320)),
    transforms.RandomHorizontalFlip(p=0.5),          # Lật ngang ngẫu nhiên
    transforms.RandomRotation(10),                   # Xoay ảnh ±10 độ
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1), # Thay đổi màu sắ
c
    transforms.RandomAffine(degrees=0, translate=(0.1, 0.1), scale=(0.9, 1.1)), # Biến đổi affine
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

→ Tác dụng: Tạo ra các biến thể của ảnh gốc, giúp mô hình học các đặc trưng tổng quát hơn.

- Transfer Learning với Fine-tuning có chọn lọc

```
# Freeze early layers (optional)
for param in list(self.backbone.parameters())[:-20]:  # Freeze first layers
    param.requires_grad = False
```

→ Tác dụng: Đóng băng các layer đầu của MobileNetV3 (pretrained trên ImageNet) và chỉ cho phép các layer cuối học tập → giảm capacity mô hình

- Weight Decay (L2 Regularization):

```
optimizer = optim.Adam(model.parameters(), lr=0.001, weight_decay=1e-4)
```

→ Tác dụng: Penalize trọng số lớn, ngăn chặn overfitting bằng cách thêm regularization term vào loss function.

- Learning Rate Scheduling:

```
scheduler = optim.lr_scheduler.StepLR(optimizer, step_size=10, gamma=0.1)
```

→ Tác dụng: Giảm learning rate sau mỗi 10 epochs, giúp hội tụ tốt hơn và tránh overshooting.

- Lưu check_point model qua mỗi N epochs đặt cài đặt:

→ Tác dụng: Dừng và điều chỉnh tham số khi xuất hiện overfit, tiếp tục huấn luyện.

4.3.2 Các tham số mô hình

- Kiến trúc mô hình:

- **Backbone**: MobileNetV3-Large (pretrained trên ImageNet)
- **Input size**: 320×320×3 pixels
- **Output**: 8 classes (phân loại 8 loài động vật)
- **Classifier**: Linear layer thay thế classifier gốc

- Tham số huấn luyện

  - **Batch size**: 32 (phù hợp với GPU T4)
  - **Learning rate**: 0.001
  - **Optimizer**: Adam với weight_decay=1e-4
  - **Loss function**: CrossEntropyLoss
  - **Epochs**: 10
  - **Validation split**: 20%

4.4 Kết quả huấn luyện

- Kết quả các tham số đánh giá:

Training Metrics Summary:

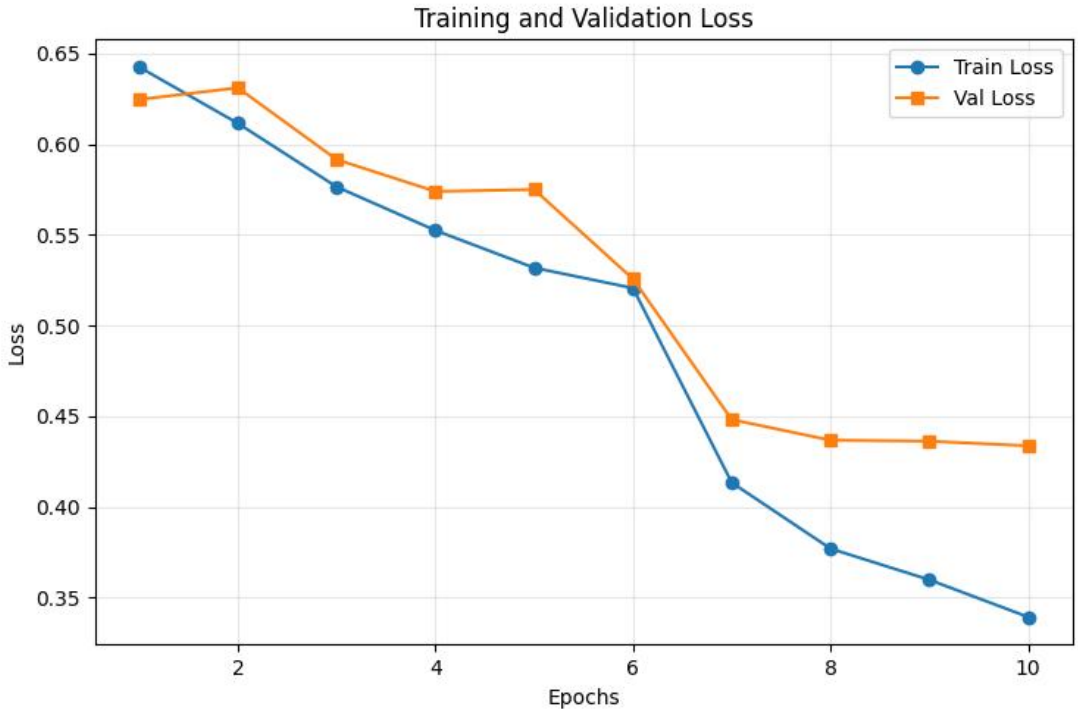
	epoch	train_loss	train_acc	val_loss	val_acc	lr
0	1	0.6426	77.0205	0.6247	78.1686	0.0010
1	2	0.6115	77.3616	0.6311	78.2292	0.0010
2	3	0.5765	79.1509	0.5916	78.7750	0.0010
3	4	0.5525	79.7346	0.5740	80.2001	0.0010
4	5	0.5319	80.4701	0.5751	80.5943	0.0010
5	6	0.5207	80.8795	0.5258	81.9588	0.0010
6	7	0.4135	85.2464	0.4482	84.2025	0.0001
7	8	0.3771	86.0121	0.4369	84.3845	0.0001
8	9	0.3599	87.0129	0.4363	84.5664	0.0001
9	10	0.3394	87.5436	0.4338	85.2638	0.0001

Hình : Kết quả chỉ số Mectrix qua 10 epochs huấn luyện

Nhận xét:

- Mô hình có sự hội tu tốt: Xu hướng chúng là cả train_loss và val_loss đều có xu hướng giảm đều đặn, trong khi train_acc và val_acc tăng dần. Chứng tỏ mô hình thực sự "học" được các đặc trưng từ dữ liệu chứ không phải đoán ngẫu nhiên.

- Tác động thần kỳ của việc giảm Learning rate (tại epochs 7): Từ epoch 1 đến 6, tốc độ học giữ nguyên ở mức 0.0010. Tại epoch 7 trở đi giảm xuống còn 0.0001. Ngay lập tức, train_los giảm mạnh (từ 0.5207 còn 0.4135), độ chính xác val_acc tăng vọt từ 81.9% lên 84.2%. Áp dụng kỹ thuật Learning Rate Scheduler khi mô hình bắt đầu bão hòa ở mức LR lớ, việc giảm LR giúp mô hình tinh chỉn (Fine-tune) trọng số kỹ hơn để thoát khỏi dao động cục bộ và tìm được cục tiểu tốt hơn.
- Kiểm soát được hiện tượng overfitting: tại epoch cuối cùng (10), độ chính xác của tập train và val có sự chênh lệch lớn hơn một chút so với các epoch trước (khoảng 2.3%). Đây là một khoảng cách an toàn, mô hình không bị overfitting nặng.



Hình 9: Biểu đồ biểu diễn giá trị loss trên tập test và valid qua 10 epoch

Nhận xét:

- Xu hướng hội tụ: Giá trị loss trên cả hai tập train và val đều có xu hướng đi xuống và khá đồng đều. Khoảng cách giữa hai đường khá nhỏ.

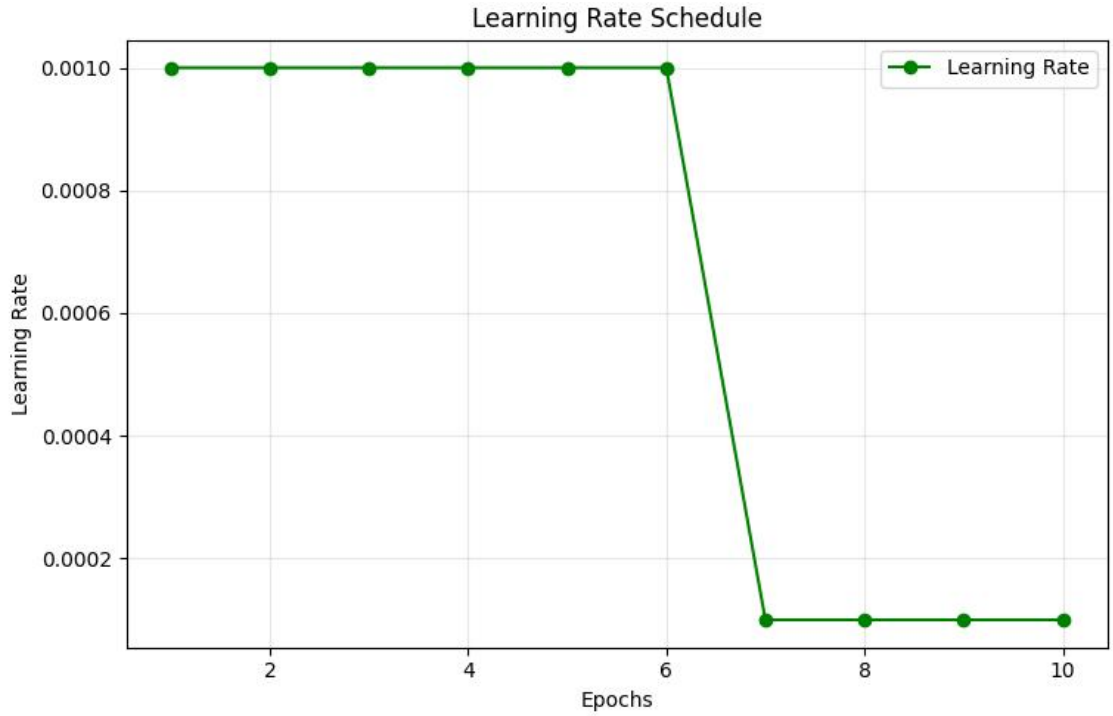
→ Ý nghĩa: Mô hình đang học rất tốt, không có hiện tượng Underfitting (Mô hình quá đơn giản) và cũng chưa xuất hiện Overfitting (học vẹt).

- Điểm gãy khúc quan trọng (Epoch 6-7): Tại khoảng giữa Epoch 6 -7, cả hai đường đều có một độ dóc đứng (giảm mạnh đột ngột). Nguyên nhân chính là thời điểm kỹ thuật Learning Rate Scheduler được kích hoạt (giảm LR từ 0.0100 xuống 0.0001).
- Dấu hiệu chớm Overfitting ở giai đoạn cuối:
  - ✧ Từ sau epoch 8 trở đi, đường màu xanh (train_loss) tiếp túc lao dóc mạnh, đường màu cam (val_loss) bắt đầu đi ngang (bão hòa quanh mức 0.43 - 0.44) và tách xa dần đường màu xanh. Điều này cho thấy mô hình đang bắt đầu học quá kỹ các chi tiết cụ thể của tập train mà không còn cải thiện them được trên tập validation.
  - ✧ Tuy nhiên, đường val_loss chưa có dấu hiệu đi lên (U-shape) nghĩa là mô hình chưa bị overfitting nặng, nó chỉ đang đạt đến giới hạn khả năng tổng quát hoá của kiến trúc hiện tại.

Hình 10: Biểu đồ diễn biến Độ chính xác (Accuracy) trên tập train và validation

Nhận xét:

- Mô hình đạt độ chính xác gần 85.26% trên tập Validation là một kết quả rất khả quan cho bài toán phân loại động vật hoang dã với kiến trúc mạng nhẹ như MobileNets.
- Đường biểu đồ không quá rung lắc (dao động), cho thấy tính ổn định của thuật toán tối ưu hóa và Batch size phù hợp.

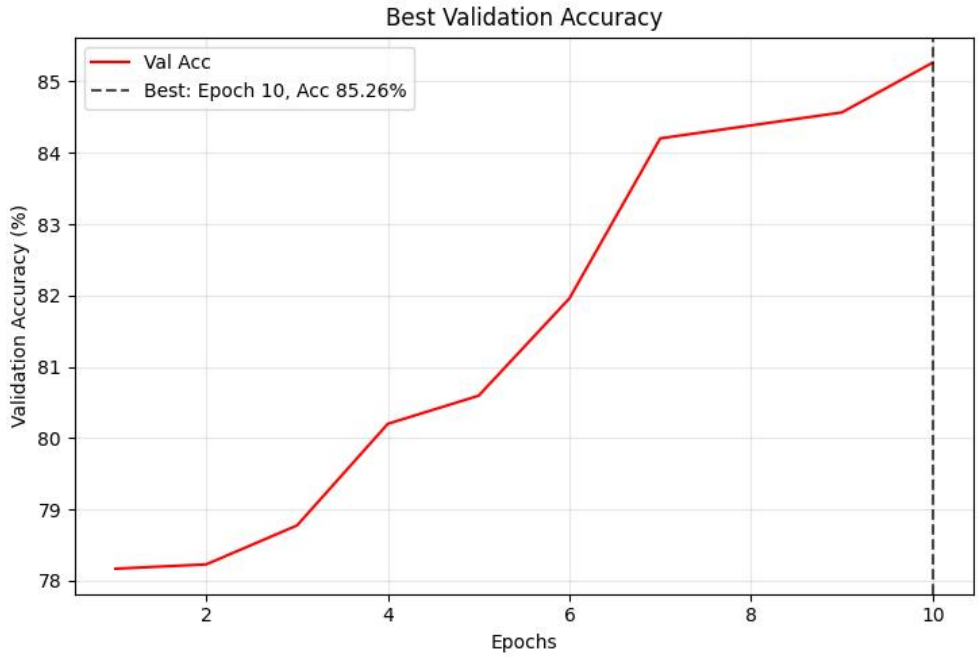


Hình 11: Biểu đồ lịch trình thay đổi tốc độ học (Learning Rate Scheduler)

Nhận xét: Nhìn vào biểu đồ, ta thấy rõ được chiến thuật thay đổi LR trong quá trình huấn luyện mô hình:

Giai đoạn 1 (Epoch 1 - 6): Toocs độ học được giữ cố định ở mức 0.0010. Đây là mức LR khá tiêu chuẩn để bắt đầu huấn luyện, đủ lớn để mô hình học nhanh các đặc trưng cơ bản.

Giai đoạn 2 (Epoch 7 - 10): giảm đột ngột LR 10 lần xuống còn 0.0001. Việc giảm LR chính là "cú hích" cần thiết để mô hình tiếp tục tối ưu hóa.

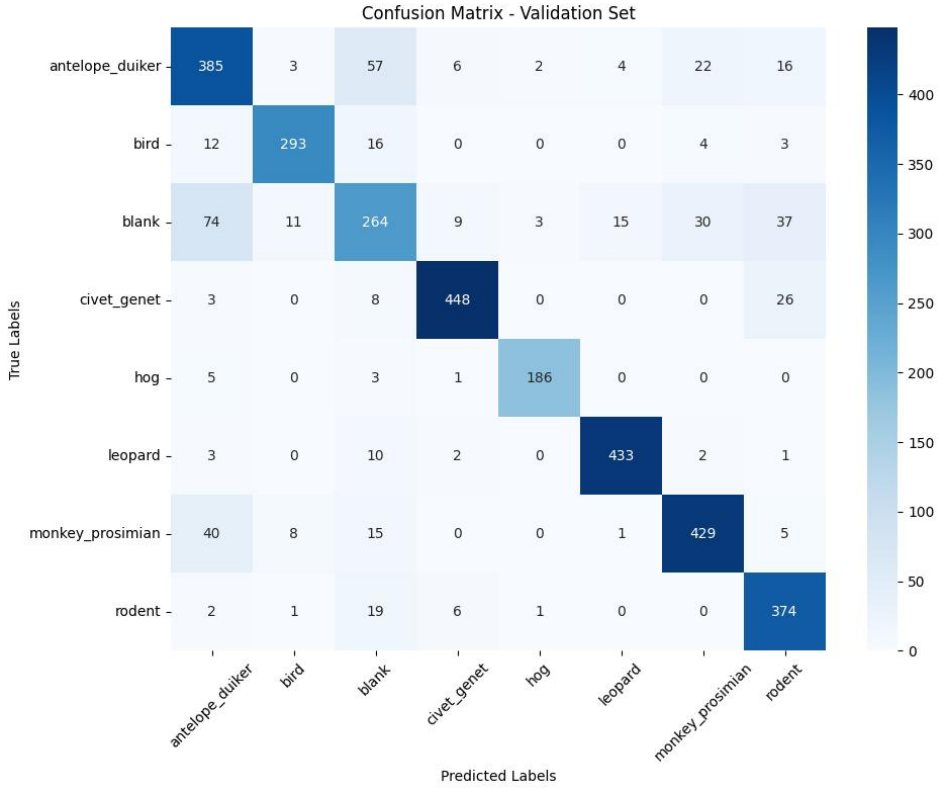


Hình 12: Biểu đồ xác định độ chính xác kiểm thử tốt nhất

Nhận xét:

- Kết quả tối ưu: Độ chính xác trên tập validation liên tục đi lên và đạt đỉnh tại điểm cuối cùng (Epoch 10). Dựa vào kết quả này, chúng ta sẽ lưu lại trọng số của mô hình tại epoch 10 để làm sản phẩm cuối cùng.
- Xu hướng tăng trưởng: Đường Validation Accuracy có xu hướng tăng đơn điệu. Không có đoạn nào bị sụt giảm đáng kể (drop) hay dao động mạnh. Chứng tỏ quá trình huấn luyện mô hình tương đối ổn định.

- **Ma trận nhầm lẫn trên tập dữ liệu:**



Hình 13: Ma trận nhầm lẫn trên tập dữ liệu Validation

Nhận xét:

- Các điểm mạnh: ô trên đường chéo chính đậm màu và số lượng lớn nhất. Điển hình như civet_genet (448 đúng), leopard (433 đúng), monkey_prosimian (429 đúng), đây là những loài có đặc điểm nhận dạng rất rõ ràng nên mô hình dễ dàng học được. Lớp thiểu số vẫn tốt như lợn rừng (hog) dù có ít dữ liệu huấn luyện nhất nhưng vẫn dự đoán đúng 186 trường hợp và rất ít khi bị nhầm qua con khác
- Vấn đề lớn nhất: Sự nhầm lẫn với lớp "Blank" (Ảnh trống). Đây là điểm yếu của hầu hết các mô hình bẫy ảnh và mô hình này cũng tương tự
  - ✧ Blank bị đoán thành Antelope (74 ảnh): Có 74 ảnh thực tế là trống (blank) nhưng mô hình lại nhìn gà hóa cuốc, tưởng là linh dương (antelope). Nguyên nhân: Có thể do gió làm rung lá cây, tạo ra các chuyển động hoặc hình khối mờ ảo giống màu lông của linh dương.
  - ✧ Antelope bị đoán thành Blank (57 ảnh): Có 57 ảnh có linh dương nhưng mô hình lại bảo là trống. Nguyên nhân: Linh dương thường có

màu lông tiệp với màu môi trường (ngụy trang), hoặc chúng đứng quá xa/bị che khuất, khiến mô hình không nhận ra.

- Sự nhầm lẫn giữa các loài động vật (Inter - class Confusion)
  - ✧ Monkey vs. Antelope: Có 40 ảnh monkey bị đoán nhầm thành antelope. Có 22 ảnh antelope bị đoán nhầm thành monkey. Lý do: Cả hai loài này đều có kích thước trung bình, lông màu nâu/xám và thường xuất hiện trong tư thế 4 chân hoặc di chuyển thấp trong rừng rậm, gây khó khăn cho việc phân biệt hình dáng.
  - ✧ Rodent (Gặm nhấm) vs. Civet (Cầy): Có 26 ảnh civet bị nhầm thành rodent. Vì cả hai đều là thú nhỏ, đi sát đất và hay hoạt động về đêm (ảnh đen trắng), nên dễ bị nhầm lẫn.

4.5 Trực quan hóa kết quả huấn luyện bằng hình ảnh

- Trực quan hóa một số kết quả ảnh đúng:

- Trực quan hóa một số kết quả ảnh sai:

4.6 Phong cách nộp bài và kết quả dự thi

Phong cách nộp bài được yêu cầu là gồm 9 cột id và 8 cột dự đoán xác suất của 8 lớp của từng ảnh trong tập test_features:

id	antelope_ duiker	bird	blank	civet_ genet	hog	leopard	monkey_ prosimian	rodent
ZJ016488	0.048233	0.189185	0.044914	0.199588	0.106118	0.132915	0.166410	0.112637
ZJ016489	0.097078	0.061400	0.026409	0.241530	0.144344	0.051780	0.287811	0.089648
ZJ016490	0.124658	0.089101	0.189225	0.174494	0.180540	0.079995	0.085672	0.076314
ZJ016491	0.109966	0.048397	0.055598	0.323600	0.322356	0.063252	0.008160	0.068671
ZJ016492	0.165742	0.184610	0.005431	0.136806	0.000389	0.122078	0.151521	0.233423
...								
ZJ020947	0.143675	0.185103	0.109074	0.158833	0.083497	0.010513	0.155293	0.154011

- Cách tính điểm:

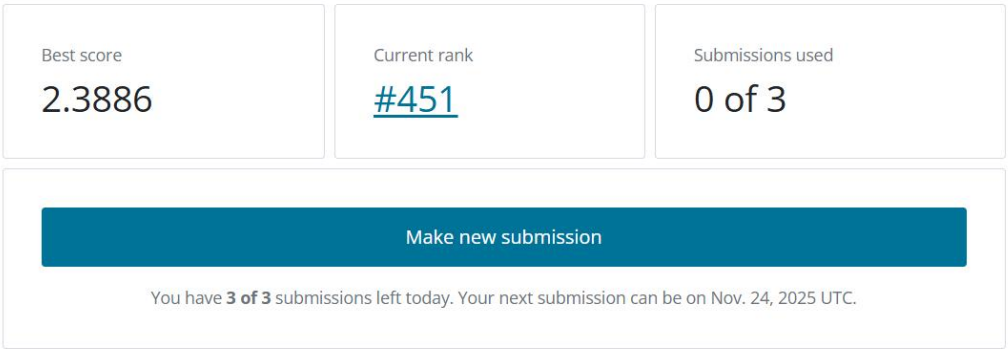
Điểm được tính bằng 1 hàm mất mát (giá trị càng thấp càng tốt:

$$loss = -\frac{1}{N} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log p_{ij}$$

Trong đó:

- N là số mẫu quan sát (N=4465)
- M là số lớp (M=8)
- y_{ij} là biến nhị phân cho biết có dự đoán đúng hay không
- p_{ij} là xác suất dự đoán nhãn i cho quan sát j

- Kết quả dự thi:

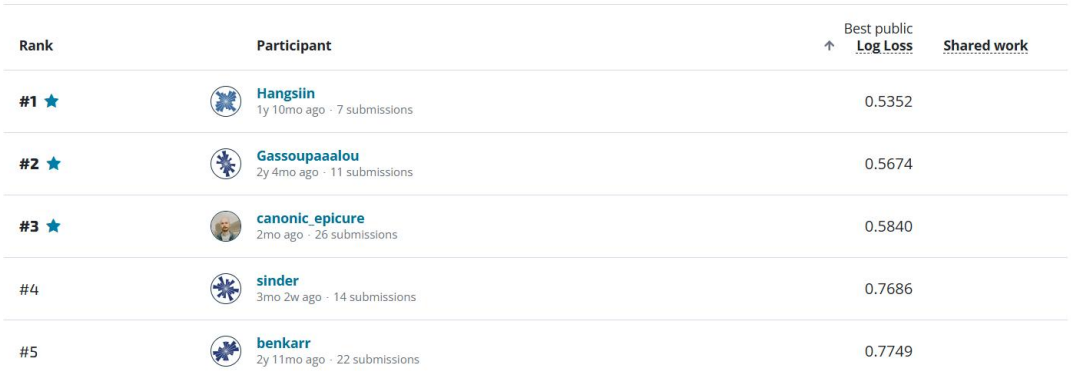


Hình 14: Số điểnm kết quả dự thi và thứ tự xếp hạng

Nhận xét:

- Mặc dù độ chính xác của mô hình rất cao 85.26% ở mô hình tốt nhất và Log loss là: 0.436242 tuy nhiên thực tế độ độ mất mát của mô hình rất cao.
- Kết quả mô hình có độ mất mát là 2.3886 và đứng ở hạng 451/1937.

- Kết quả xếp hạng hiện tại:



Hình 15: Kết quả xếp hạng trong cuộc thi

TÀI LIỆU THAM KHẢO

[1] DrivenData. (2025). *Conservision Practice Area: Image Classification - Competition Details*. Retrieved from https://www.drivendata.org/competitions/87/competition-image-classification-wildlife-conservation/

[2] Howard, A. G., et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv preprint arXiv:1704.04861.

[3] Chollet, F. (2017). *Xception: Deep Learning with Depthwise Separable Convolutions*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[4] Norouzzadeh, M. S., et al. (2018). *Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning*. Proceedings of the National Academy of Sciences (PNAS), 115(25).