

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO CUỐI KỲ
XỬ LÝ ẢNH VÀ ỨNG DỤNG - CS406.O11.KHCL

ĐỀ TÀI
LOCATION RECOGNITION WITH FEW SHOT DATASET

Giảng viên hướng dẫn: TS. Mai Tiến Dũng

Nhóm sinh viên thực hiện:

Họ và tên	MSSV
Nguyễn Thị Ngọc Hà	21520217
Huỳnh Võ Ngọc Thanh	21520449

TP.HCM, tháng 01 năm 2024

Mục lục

1 PHẦN 1. GIỚI THIỆU BÀI TOÁN	2
2 PHẦN 2. CÔNG TRÌNH LIÊN QUAN	3
2.1 Tổng quan về Few-shot learning	3
2.1.1 Tiếp cận theo cấp độ tham số (Parameter-level approach)	3
2.1.2 Tiếp cận theo cấp độ dữ liệu (Data-level approach)	4
2.2 Tổng quan về tăng cường dữ liệu	5
3 PHẦN 3. PHƯƠNG PHÁP	5
3.1 Tăng cường dữ liệu	5
3.2 Mô hình phân loại	7
3.2.1 ResNet-18	7
3.2.2 MobileNetV2	8
3.2.3 VGG-16	9
4 PHẦN 4. THỰC NGHIỆM	9
4.1 Giới thiệu bộ dữ liệu	9
4.1.1 Dữ liệu huấn luyện DVU-TRECVID	10
4.1.2 Dữ liệu kiểm tra DVU-TRECVID	10
4.2 Chuẩn bị dữ liệu	11
4.2.1 Chuẩn bị dữ liệu huấn luyện	11
4.2.2 Chuẩn bị dữ liệu kiểm tra	11
4.3 Tăng cường dữ liệu huấn luyện	13
4.3.1 Các kỹ thuật tăng cường dữ liệu đã áp dụng	13
4.3.2 Các phiên bản thử nghiệm	14
4.4 Huấn luyện mô hình	14
4.5 Dánh giá kết quả thực nghiệm	14
4.5.1 Độ đo Accuracy	14
4.5.2 Kết quả đánh giá trên tập kiểm tra	15
4.5.3 Kết quả thực nghiệm	15
5 PHẦN 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	18
5.1 Kết luận	18
5.2 Hướng phát triển	19
6 TÀI LIỆU THAM KHẢO	19

PHẦN 1. GIỚI THIỆU BÀI TOÁN

Mỗi ngày, hàng tỷ hình ảnh và video được tải lên trên các nền tảng trực tuyến, tạo thành một nguồn tài nguyên vô cùng đồ sộ và đa dạng. Điều này không chỉ làm tăng cường trải nghiệm giải trí của người xem mà còn đánh thức trí tò mò về các cảnh quay đẹp, độc đáo xuất hiện trong video. Điều này đặt ra một thách thức lớn và làm cho bài toán nhận diện địa điểm trở nên quan trọng hơn bao giờ hết.

Bài toán nhận diện địa điểm không chỉ đóng vai trò quan trọng trong việc cải thiện trải nghiệm người dùng mà còn mở ra nhiều ứng dụng tiềm năng trong lĩnh vực du lịch, quảng cáo địa điểm, và quản lý nguồn lực đô thị. Với sự phát triển nhanh chóng của công nghệ, việc nhận diện chính xác địa điểm trong bức ảnh hay tổng quát hơn là trong một đoạn hình ảnh liên tục (shot) của video ngày càng trở nên cần thiết trong các lĩnh vực của cuộc sống. Công nghệ nhận diện địa điểm trong video có thể hỗ trợ trong lĩnh vực bảo mật và giám sát, giúp theo dõi và phát hiện các sự kiện quan trọng tại một địa điểm cụ thể. Hay có thể sử dụng nhận diện địa điểm để cung cấp hướng dẫn điều hướng, chỉ đường trên thời gian thực cho người dùng.

Trong đồ án này, chúng tôi thực hiện giải quyết bài toán nhận diện địa điểm ghi hình trong các địa điểm cho trước của các shot dựa vào dữ liệu huấn luyện ít ỏi.

- Input:

- Tập dữ liệu huấn luyện gồm N lớp (địa điểm), mỗi lớp có M ảnh (4-8 ảnh).
- Một đoạn shot video cần xác định địa điểm.

- Output: địa điểm quay shot video đó.



Hình 1: Ảnh minh họa

PHẦN 2. CÔNG TRÌNH LIÊN QUAN

2.1 Tổng quan về Few-shot learning

Few-shot learning là một lĩnh vực quan trọng trong Machine learning, Mô hình Few-shot learning được huấn luyện để phân loại dữ liệu mới dựa số lượng nhỏ dữ liệu ở mỗi lớp.

Mục tiêu của Few-shot learning là phát triển các mô hình có khả năng đổi mới với tình huống thiếu dữ liệu lớn. Điều quan trọng là mô hình cần có khả năng tổng quát hóa, tức nó có thể áp dụng kiến thức đã học từ một số ít dữ liệu huấn luyện lên các dữ liệu mới mà nó chưa thấy trước đó. Điều này làm Few-shot learning trở nên quan trọng trong những tình huống thực tế, nơi việc thu thập dữ liệu vô cùng khó khăn và đòi hỏi chi phí cao.

Tầm quan trọng của Few-shot learning

- **Học từ các điểm dị thường (anomalies):** Few-shot learning cho phép học các dữ liệu hiếm gặp. Ví dụ, khi phân loại ảnh các bệnh hiếm như COVID, mô hình sử dụng Few-shot Learning có thể phân loại chính xác ảnh X-quang phổi từ một số ít hình ảnh cho trước.
- **Giảm chi phí thu thập dữ liệu:** Few-shot learning giảm lượng dữ liệu cần thiết để huấn luyện mô hình, loại bỏ các chi phí đắt đỏ để thu thập và gán nhãn dữ liệu.
- **Học giống như con người:** con người có khả năng nhận biết sự khác nhau giữa các đối tượng dựa trên chỉ vài dữ liệu cho trước. Ngược lại, máy tính thường cần lượng lớn dữ liệu để hiểu và phân biệt. Few-shot learning là một bước tiến lớn giúp máy tính học từ lượng ít dữ liệu tương tự như cách con người học.

2.1.1 Tiếp cận theo cấp độ tham số (Parameter-level approach)

Việc overfitting trên các mẫu few-shot learning là khá dễ xảy ra, do few-shot learning thường đối mặt với thách thức của việc học từ một lượng nhỏ các mẫu trong khi không gian tham số có thể rất lớn. Để vượt qua vấn đề này, chúng ta nên giới hạn không gian tham số và sử dụng các kỹ thuật regularization với hàm mất mát phù hợp. Mô hình sẽ tổng quát hóa số lượng hạn chế các mẫu huấn luyện.

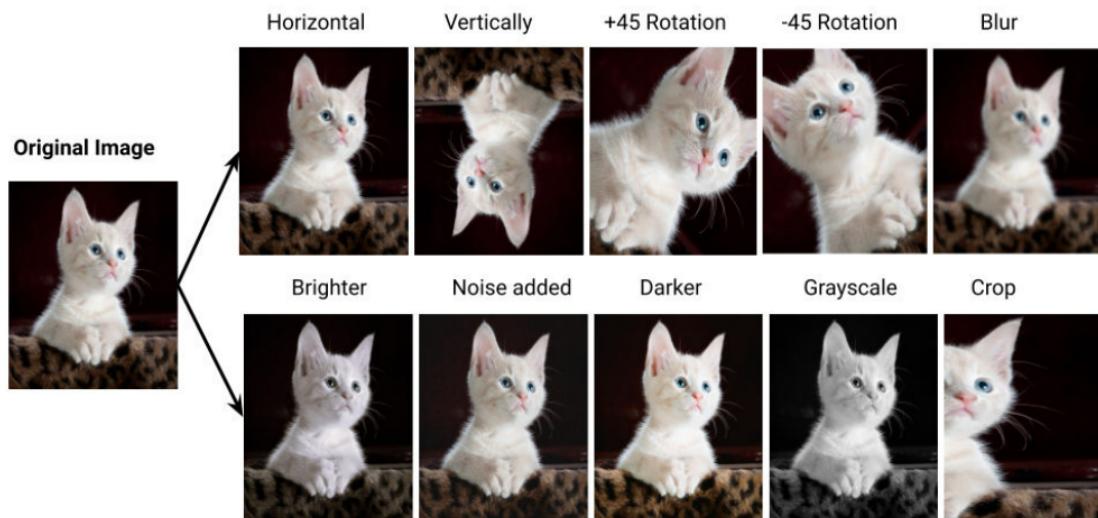


Hình 2: Few-Shot Image Classification with Meta-Learning

2.1.2 Tiếp cận theo cấp độ dữ liệu (Data-level approach)

Cách tiếp cận này dựa trên ý tưởng nếu không có đủ dữ liệu để xây dựng một mô hình đáng tin cậy và tránh overfitting hay underfitting, ta nên tăng thêm dữ liệu. Đó là lý do tại sao nhiều vấn đề Few-Shot learning được giải quyết bằng cách sử dụng thông tin bổ sung từ một tập dữ liệu lớn gốc. Đặc điểm chính của tập dữ liệu gốc là nó không có các lớp mà chúng ta có trong tập hỗ trợ cho nhiệm vụ few-shot. Ví dụ, nếu chúng ta muốn phân loại một loài chim cụ thể, tập dữ liệu gốc có thể chứa hình ảnh của nhiều loài chim khác.

Chúng ta cũng có thể tạo thêm dữ liệu bằng cách sử dụng kỹ thuật tăng cường dữ liệu hoặc thậm chí sử dụng mạng GANs để sinh thêm dữ liệu.



Hình 3: Ảnh minh họa kỹ thuật tăng cường dữ liệu

2.2 Tổng quan về tăng cường dữ liệu

Tăng cường dữ liệu là một kỹ thuật trong machine learning và computer vision được sử dụng để tạo ra sự đa dạng và phong phú trong tập dữ liệu huấn luyện bằng cách thêm vào các phiên bản biến đổi của dữ liệu gốc. Mục tiêu chính của tăng cường dữ liệu là cải thiện khả năng tổng quát hóa của mô hình và giảm nguy cơ overfitting.

Một vài mô hình nâng cao để tăng cường dữ liệu:

- **Mạng GAN** bao gồm hai mô hình chính: một mô hình sinh (Generator) và một mô hình phân biệt (Discriminator).

Generator tạo ra dữ liệu giả mạo để lừa mô hình phân biệt và Discriminator phân biệt giữa dữ liệu thật từ tập dữ liệu và dữ liệu được tạo ra bởi Generator. Hai mô hình này cạnh tranh với nhau trong quá trình đào tạo. Generator học cách tạo dữ liệu giả mạo sao cho nó khó phân biệt với dữ liệu thật, trong khi Discriminator học cách phân biệt giữa dữ liệu thật và giả mạo. GAN có thể sử dụng để tạo ra dữ liệu mới, phong phú và đa dạng từ dữ liệu huấn luyện ban đầu thông qua quá trình đào tạo.

- **Chuyển kiểu neural (Neural Style Transfer)** giúp kết hợp nội dung của một hình ảnh với phong cách của một hình ảnh khác.

Bằng cách áp dụng chuyển kiểu neural, ta có thể tạo ra các phiên bản mới của hình ảnh với các phong cách khác nhau, tăng cường sự đa dạng của dữ liệu.

- **Học Tăng Cường (Reinforcement Learning)** liên quan đến việc các mô hình đào tạo các tác nhân để hoàn thành mục tiêu và đưa ra quyết định trong môi trường ảo. Tác nhân nhận được phản hồi trong hình thức phần thưởng hoặc hình phạt dựa trên hành động của mình.

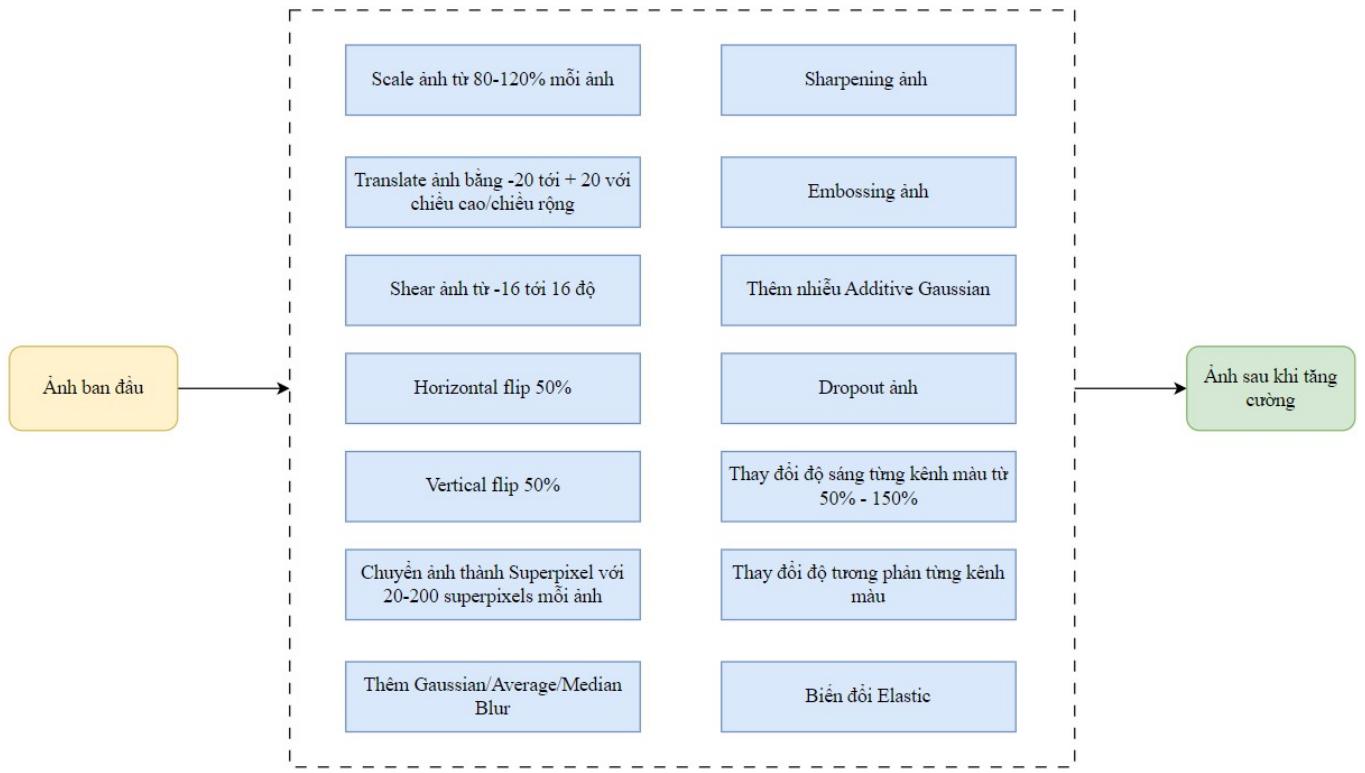
Trong ngữ cảnh tăng cường dữ liệu, mô hình học tăng cường có thể được sử dụng để tạo ra các tình huống mới và đa dạng, mà sau đó có thể được sử dụng làm dữ liệu huấn luyện.

PHẦN 3. PHƯƠNG PHÁP

3.1 Tăng cường dữ liệu

Pipeline tăng cường dữ liệu mà nhóm sử dụng có thể thấy ở hình 4

Tăng cường dữ liệu



Hình 4: Pipeline tăng cường dữ liệu mà nhóm sử dụng

Flip ảnh là quá trình đảo ngược hình ảnh theo một hoặc nhiều trục, tạo ra hình ảnh mới có hướng khác nhau. Có hai loại:

- **Horizontal Flip** (Flip theo chiều ngang): ảnh được đảo ngược theo chiều ngang, nghĩa là phần trái trở thành phải và ngược lại.
- **Vertical Flip** (Flip theo chiều dọc): ảnh được đảo ngược theo chiều dọc, nghĩa là phần trên trở thành dưới và ngược lại.

Thay đổi kích thước ảnh (Scaling image) là quá trình điều chỉnh kích thước của một hình ảnh, thường là để làm cho nó phù hợp với yêu cầu cụ thể hoặc để tối ưu hóa hiển thị trong một ngữ cảnh nhất định.

Dịch chuyển ảnh (translate images) là một phép biến đổi affine cơ bản. Mỗi điểm ảnh trong hình ảnh ban đầu được di chuyển một khoảng cố định theo chiều ngang hoặc chiều dọc.

Làm nghiêng ảnh (shearing) là một phép biến đổi affine thường được sử dụng để làm thay đổi hình dạng của một đối tượng bằng cách nghiêng nó theo một hoặc cả hai trục.

Gaussian Blur: một kỹ thuật xử lý ảnh được sử dụng để làm mịn hình ảnh bằng cách áp dụng một bộ lọc Gaussian. Cụ thể, mỗi điểm ảnh mới trong hình ảnh sau khi được xử lý được tính toán bằng cách lấy trung bình của các giá trị pixel trong một vùng xung quanh nó, với trọng số dựa trên hàm phân phối Gaussian.

Average Blur: là một kỹ thuật xử lý ảnh sử dụng một bộ lọc trung bình để làm mịn hình ảnh. Mỗi pixel mới trong hình ảnh được tính toán bằng cách lấy trung bình của các giá trị pixel trong một vùng xung quanh nó.

Median Blur: là một phương pháp xử lý ảnh dựa trên việc thực hiện bộ lọc trung vị trên một vùng xung quanh của mỗi điểm ảnh trong hình ảnh. Bộ lọc trung vị sắp xếp các giá trị pixel trong vùng xung quanh thành một dãy tăng dần và chọn giá trị ở giữa làm giá trị mới cho điểm ảnh.

Phép biến đổi Superpixels được sử dụng để chia hình ảnh thành các vùng (superpixels). Mỗi superpixel là một nhóm các pixel liền kề được nhóm lại để biểu diễn một vùng đồng nhất của hình ảnh. Phép biến đổi Superpixels thường được sử dụng để giảm kích thước của hình ảnh và làm cho mô hình tập trung vào các đặc trưng quan trọng.

Phép biến đổi Sharpen là một kỹ thuật xử lý ảnh được sử dụng để làm tăng độ rõ nét và tăng sắc nét của đối tượng trong hình ảnh. Mỗi pixel trong hình ảnh được cập nhật bằng cách thêm một lượng trọng số có hướng từ các pixel lân cận, tăng cường sự chênh lệch giữa các giá trị pixel.

Hiệu ứng Emboss thường được sử dụng để làm nổi bật các chi tiết và biên giữa các vùng trong hình ảnh. Trong phép biến đổi Emboss, mỗi pixel trong ảnh được cập nhật dựa trên giá trị của pixel lân cận để tạo ra sự tương phản giữa các vùng. Thông thường, phương pháp này sử dụng một kernel hoặc một bộ lọc để tính toán giá trị mới cho mỗi pixel. Kết quả là tạo ra các điểm sáng và tối tạo cảm giác chiều sâu và 3D cho hình ảnh.

Thêm nhiều Gaussian là một kỹ thuật trong xử lý ảnh, một lượng nhỏ nhiễu theo phân phối Gaussian (normal) được thêm vào mỗi pixel trong hình ảnh. Nuisance này thường được sử dụng để mô phỏng các biến động ngẫu nhiên trong dữ liệu hình ảnh và có thể được áp dụng để làm cho hình ảnh trở nên thực tế hơn hoặc để kiểm thử hiệu suất của các thuật toán xử lý ảnh.

Dropout ảnh là một kỹ thuật được sử dụng trong quá trình xử lý ảnh để ngăn chặn hiện tượng overfitting. Một số pixel trong ảnh được chọn ngẫu nhiên và được đặt giá trị về 0. Quá trình này giống như việc tắt ngẫu nhiên một số đơn vị trong mạng nơ-ron.

Thay đổi độ sáng trên từng kênh màu của hình ảnh là ta sẽ thực hiện phép nhân với một giá trị scale cho mỗi kênh màu riêng biệt. Quá trình này giúp điều chỉnh độ sáng của từng kênh màu một cách độc lập, tạo ra hiệu ứng thay đổi mức độ sáng mà không làm thay đổi tổng quan của hình ảnh.

Thay đổi độ tương phản trên từng kênh màu của hình ảnh viết tiếp.

Biến đổi đàn hồi (Elastic deformation) là một kỹ thuật trong xử lý ảnh được sử dụng để tạo ra hiệu ứng biến dạng trên ảnh một cách tự nhiên, nhưng vẫn giữ được cấu trúc và thông tin quan trọng của đối tượng trong ảnh.

3.2 Mô hình phân loại

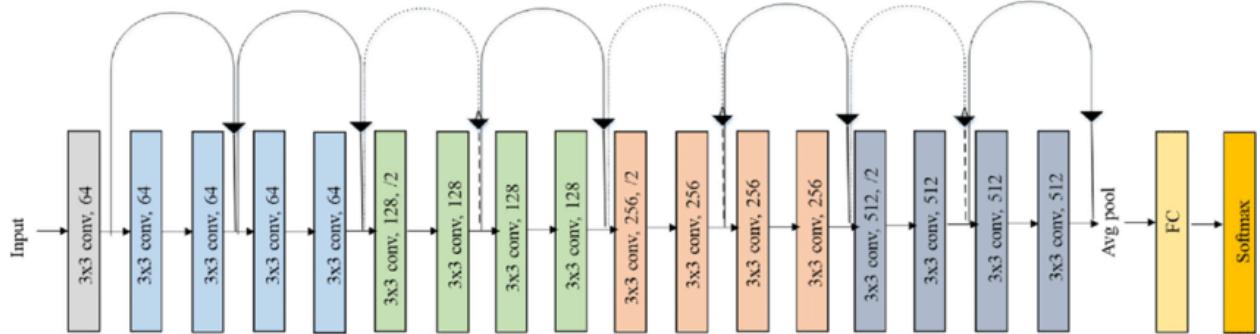
3.2.1 ResNet-18

ResNet-18 là một kiến trúc mạng nơ-ron sâu thuộc dòng ResNet (Residual Network), được thiết kế bởi Microsoft Research. ResNet-18 có kiến trúc rất sâu gồm 18 lớp như Hình 5. Kiến trúc này thường được sử dụng cho bài toán phân loại hình ảnh.

Chúng tôi đã sử dụng ResNet-18 làm mô hình đã được huấn luyện trước trên tập dữ liệu

ImageNet trong phương pháp này. Lớp Fully Connected ở cuối mô hình, kèm theo hàm Softmax, đã được thay thế bằng một lớp Linear mới. Lớp Linear này có số lượng đầu ra tương ứng với số lớp cần phân loại trong tập dữ liệu mới của chúng tôi.

Qua đó, chúng tôi giữ lại kiến thức đã học được từ ImageNet, giúp tối ưu hóa mô hình để phân loại các đối tượng hoặc đặc trưng trong tập dữ liệu mới. Việc này giúp giảm bớt áp lực huấn luyện và giảm thời gian đào tạo, đồng thời cung cấp một mô hình có khả năng tổng quát hóa tốt trên dữ liệu mới.



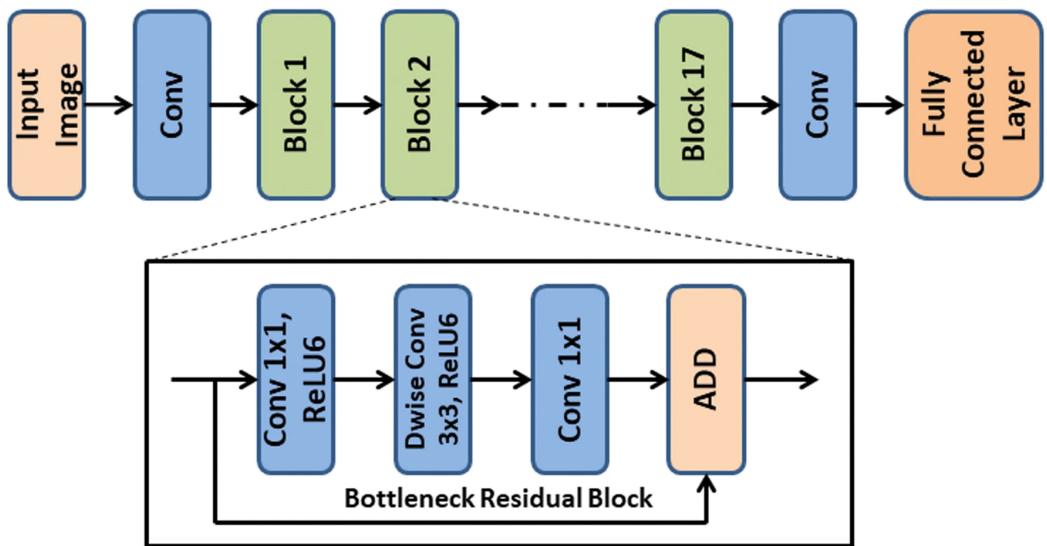
Hình 5: Mô hình mạng ResNet-18 gốc

3.2.2 MobileNetV2

MobileNet v2 là một trong những kiến trúc được ưa chuộng nhất khi phát triển các ứng dụng AI trong computer vision được giới thiệu năm 2018. Rất nhiều các kiến trúc sử dụng backbone là MobileNetV2 như SSDLite trong object detection và DeepLabV3 trong image segmentation. Mô hình này sử dụng các kỹ thuật như bottleneck structures, depthwise separable convolution, linear bottlenecks và inverted residuals để giảm lượng tham số và chi phí tính toán. MobileNetV2 tập trung vào cân bằng giữa hiệu suất và tính di động, làm cho nó trở thành lựa chọn phổ biến cho nhận diện vật thể và phân loại hình ảnh trên thiết bị di động và các ứng dụng nhúng khác.

Chúng tôi đã áp dụng phương pháp Transfer Learning bằng cách sử dụng mô hình MobileNet-v2 đã được huấn luyện trước trên tập dữ liệu ImageNet. Quá trình này bao gồm việc thay đổi lớp phân loại cuối cùng của mô hình để phù hợp với số lượng lớp mới trong tập dữ liệu của chúng tôi. Thay vì sử dụng lớp phân loại cuối cùng của MobileNet-v2, chúng tôi đã thay thế nó bằng một lớp Linear mới. Lớp Linear này có số lượng đầu ra tương ứng với số lớp trong tập dữ liệu mới.

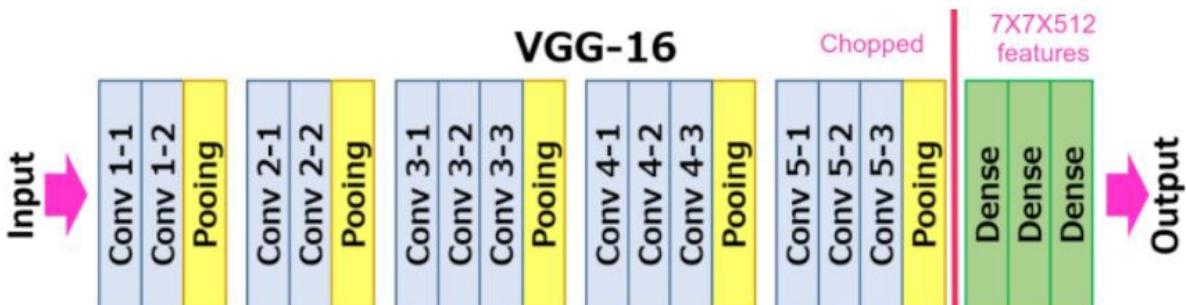
Điều này cho phép chúng ta tận dụng kiến thức đã học được từ tập dữ liệu lớn của ImageNet và tối ưu hóa mô hình cho công việc cụ thể mà chúng ta quan tâm, như phân loại các lớp trong tập dữ liệu mới. Bằng cách này, ta không phải huấn luyện mô hình từ đầu mà vẫn có thể đạt được hiệu suất tốt đối với dữ liệu mới.



Hình 6: Mô hình mạng MobileNet-v2 gốc

3.2.3 VGG-16

VGG-16 là một mạng convolutional neural network được đề xuất bởi K. Simonyan and A. Zisserman, University of Oxford năm 2014. Với VGG-16, quan điểm về mạng nơ ron sâu hơn sẽ giúp ích cho cải thiện độ chính xác của mô hình tốt hơn. Kiến trúc của VGG-16 gồm 16 lớp bao gồm 13 lớp tích chập 2 chiều và 3 lớp fully connected như Hình 7. VGG-16 sử dụng các bộ lọc nhỏ có kích thước 3×3 cho tất cả các lớp Convolution, với stride 1 và padding 1 đi kèm với lớp Max Pooling có kích thước 2×2 và stride là 2.



Hình 7: Mô hình mạng VGG-16 gốc

PHẦN 4. THỰC NGHIỆM

4.1 Giới thiệu bộ dữ liệu

Bộ dữ liệu TRECVID2022¹ gồm 6 bộ phim được thu thập từ các public websites như Vimeo, Internet Archive và Deep Video Understanding task các năm trước.

¹DVU-TRECVID2022: <https://www-nlpir.nist.gov/projects/trecvid/dvu/trecvid.2022.queries.groundTruth/tv22.dvu.queries/images/>

4.1.1 Dữ liệu huấn luyện DVU-TRECVID

Dữ liệu huấn luyện là 6 folder phim với mỗi folder chứa ảnh chụp các thực thể chính (persons, locations, concepts). Qua quá trình chuẩn bị dữ liệu (sẽ nói rõ ở phần sau), ta thu được tập huấn luyện gồm 163 ảnh thuộc thực thể *locations* được gán nhãn vào 31 lớp địa điểm. Lưu ý: trong phạm vi đồ án này chỉ quan tâm đến thực thể *locations* trong bộ dữ liệu.



Baseball field



Statue of liberty



Gas station



Rooftop

Hình 8: Một số mẫu dữ liệu trong tập huấn luyện

4.1.2 Dữ liệu kiểm tra DVU-TRECVID

Dữ liệu kiểm tra gồm 6 folder chứa các cảnh quay (scenes) ở mỗi bộ phim. Mỗi cảnh quay được ghi hình tại nhiều địa điểm khác nhau.

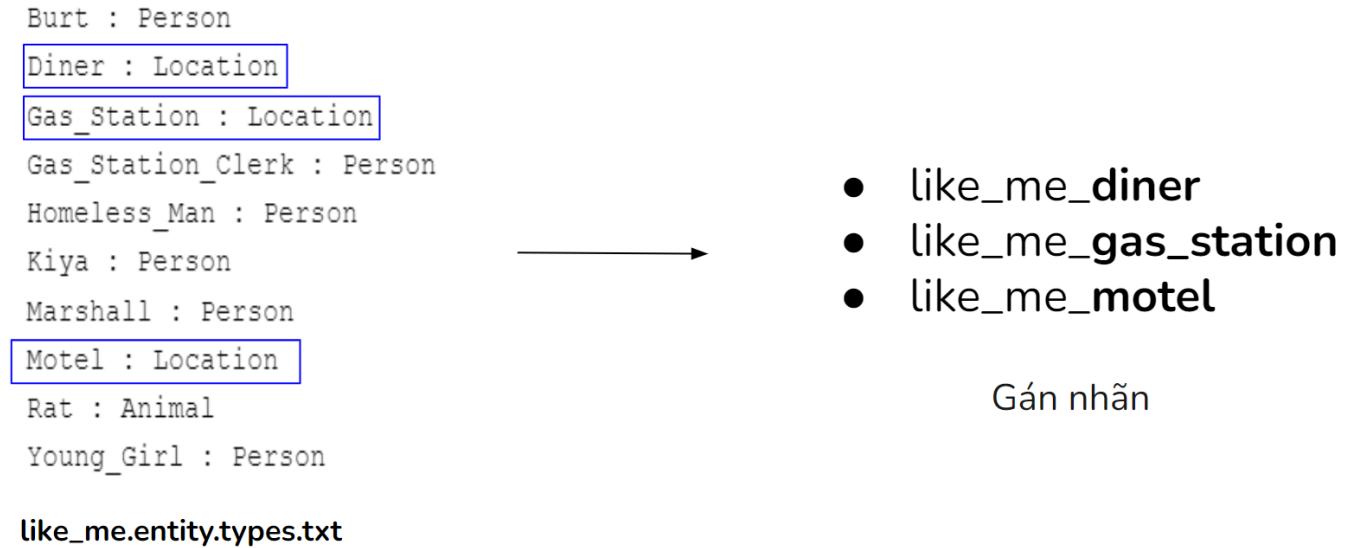


Hình 9: Một số scenes trong tập kiểm tra

4.2 Chuẩn bị dữ liệu

4.2.1 Chuẩn bị dữ liệu huấn luyện

Dựa vào file định dạng thực thể (txt) có sẵn trong TRECVID2022 để lấy ra các thực thể là Location và tiến hành gán nhãn cho các lớp địa điểm theo form *tên phim + tên địa điểm* có trong phim:



Hình 10: Ảnh minh họa

4.2.2 Chuẩn bị dữ liệu kiểm tra

Do một cảnh quay (scene) được ghi hình từ nhiều shot quay, do đó ta sử dụng **PySceneDetect** để tiến hành cắt mỗi cảnh quay thành các shot.

PySceneDetect là một thư viện mã nguồn mở trong ngôn ngữ Python được thiết kế để phát hiện các phân đoạn (scenes) trong video. Phương pháp phát hiện này giúp chia video thành các đoạn nhỏ dựa trên sự thay đổi trong nội dung của video.



Hình 11: Ảnh minh họa

Tiếp theo, tiến hành cắt shot thành từng frame ảnh (5fps) phục vụ cho việc phân loại địa điểm mà shot được quay.



Hình 12: Ảnh minh họa



Hình 13: Một số frames được cắt từ shot 3, scene 1 của phim Little Rock

Sau khi cắt shot, ta tiến hành gán nhãn thủ công và thu được tập kiểm tra gồm 176 shots như sau:

- Calloused hands: 60 shots
- Chained for life: 15 shots
- Like me: 18 shots
- Little rock: 27 shots
- Losing ground: 33 shots
- Liberty kid: 23 shots

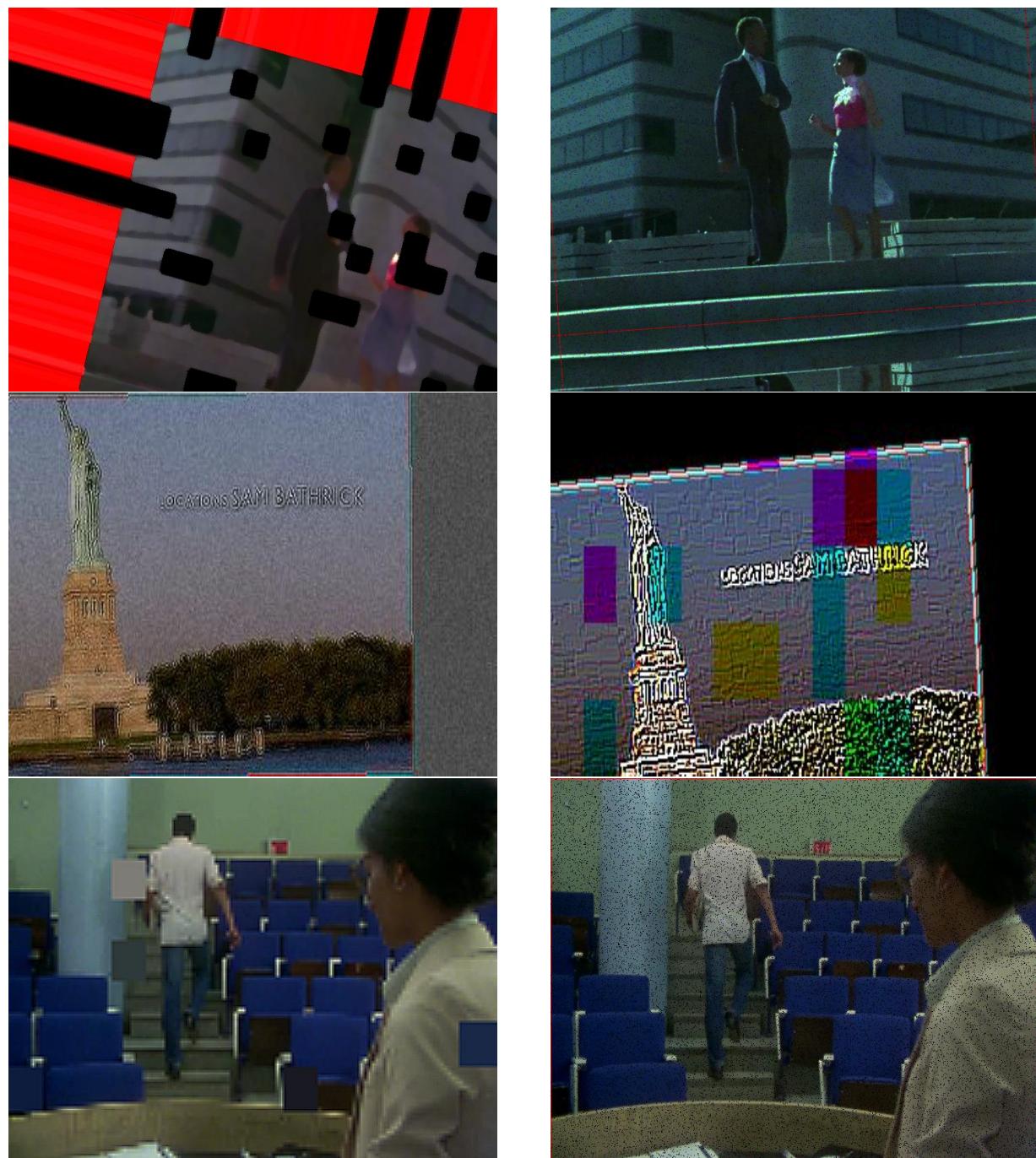
Link bộ dữ liệu kiểm tra: [tại đây](#)

4.3 Tăng cường dữ liệu huấn luyện

4.3.1 Các kỹ thuật tăng cường dữ liệu đã áp dụng

Sử dụng thư viện **imgaug** trong Python để tăng cường dữ liệu hình ảnh bao gồm: lật ảnh theo trục ngang dọc, phóng to, thu, nhở ảnh, dịch ảnh, nghiêng ảnh, blur, chuyển đổi thành biểu diễn superpixel, sharpen ảnh (làm sắc nét), emboss ảnh (tăng giảm độ tương phản), thêm nhiễu Gaussian, dropout (loại bỏ ngẫu nhiên pixel), thay đổi độ sáng, thay đổi độ sáng trên từng kênh màu (R,G,B), biến dạng ảnh Elastic,...

Một số hình ảnh trong tập huấn luyện sau khi tăng cường dữ liệu:



Hình 14: Một số ảnh của bộ dữ liệu tăng cường

4.3.2 Các phiên bản thử nghiệm

Nhóm đã tiến hành tăng cường trên 3 phiên bản 100 ảnh, 200 ảnh và 500 ảnh để có thể so sánh và đánh giá một cách hiệu quả trong quá trình huấn luyện.

Sau khi tăng cường, kích thước tập huấn luyện cụ thể như sau:

- Tăng cường 100 ảnh: từ 163 ảnh thành 16463 ảnh
- Tăng cường 200 ảnh: từ 163 ảnh thành 32763 ảnh
- Tăng cường 500 ảnh: từ 163 ảnh thành 8663 ảnh

Link bộ dữ liệu huấn luyện sau khi tăng cường: [tại đây](#)

4.4 Huấn luyện mô hình

Cả ba mô hình ResNet-18, MobileNet-v2 và VGG-16 đã được huấn luyện trên Google Colab sử dụng card đồ họa T4 Tesla GPU. Mỗi mô hình trải qua 10 epochs trong quá trình huấn luyện, với việc sử dụng thuật toán Stochastic Gradient Descent (SGD). Tham số của SGD được thiết lập như sau: learning rate là 10^{-4} và momentum là 0.9. Để cải thiện hiệu suất, ta đã thực hiện việc giảm learning rate xuống 10^{-5} sau khi huấn luyện được 7 epochs, với hệ số giảm (gamma) là 0.1. Tổng thời gian huấn luyện cho mỗi mô hình là 70 phút với ập dữ liệu được tăng cường 500 ảnh.

Link checkpoints sau khi huấn luyện mô hình: [checkpoints](#)

4.5 Đánh giá kết quả thực nghiệm

4.5.1 Độ đo Accuracy

Dộ đo accuracy cho biết tỷ lệ các trường hợp được dự đoán đúng trên tổng số các trường hợp. Độ chính xác càng cao thì mô hình của chúng ta càng chuẩn xác.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{total sample}}$$

- True Positive (TP) là số lượng các mẫu dữ liệu thực tế thuộc vào lớp positive và mô hình cũng dự đoán chúng thuộc vào lớp positive.
- False Negative (FN) là số lượng các mẫu dữ liệu thực tế thuộc vào lớp positive nhưng mô hình dự đoán chúng thuộc vào lớp negative.
- False Positive (FP) là số lượng các mẫu dữ liệu thực tế thuộc vào lớp negative nhưng mô hình dự đoán chúng thuộc vào lớp positive.
- True Negative (TN) là số lượng các mẫu dữ liệu thực tế thuộc vào lớp negative và mô hình cũng dự đoán chúng thuộc vào lớp negative.

4.5.2 Kết quả đánh giá trên tập kiểm tra

Dể kiểm tra một shot được ghi hình ở địa điểm nào, nhóm quyết định test trên từng frame trong shot đó. Lớp được dự đoán nhiều nhất trên tổng các frame của một shot sẽ là lớp của shot đó.

Bảng 1: Kết quả đánh giá trên bản thử nghiệm 100 ảnh

	Correct/Total (176 shots)	Accuracy
ResNet-18	98	0.557
MobileNetv2	104	0.591
VGG-16	96	0.546

Bảng 2: Kết quả đánh giá trên bản thử nghiệm 200 ảnh

	Correct/Total (176 shots)	Accuracy
ResNet-18	98	0.557
MobileNetv2	103	0.585
VGG-16	97	0.551

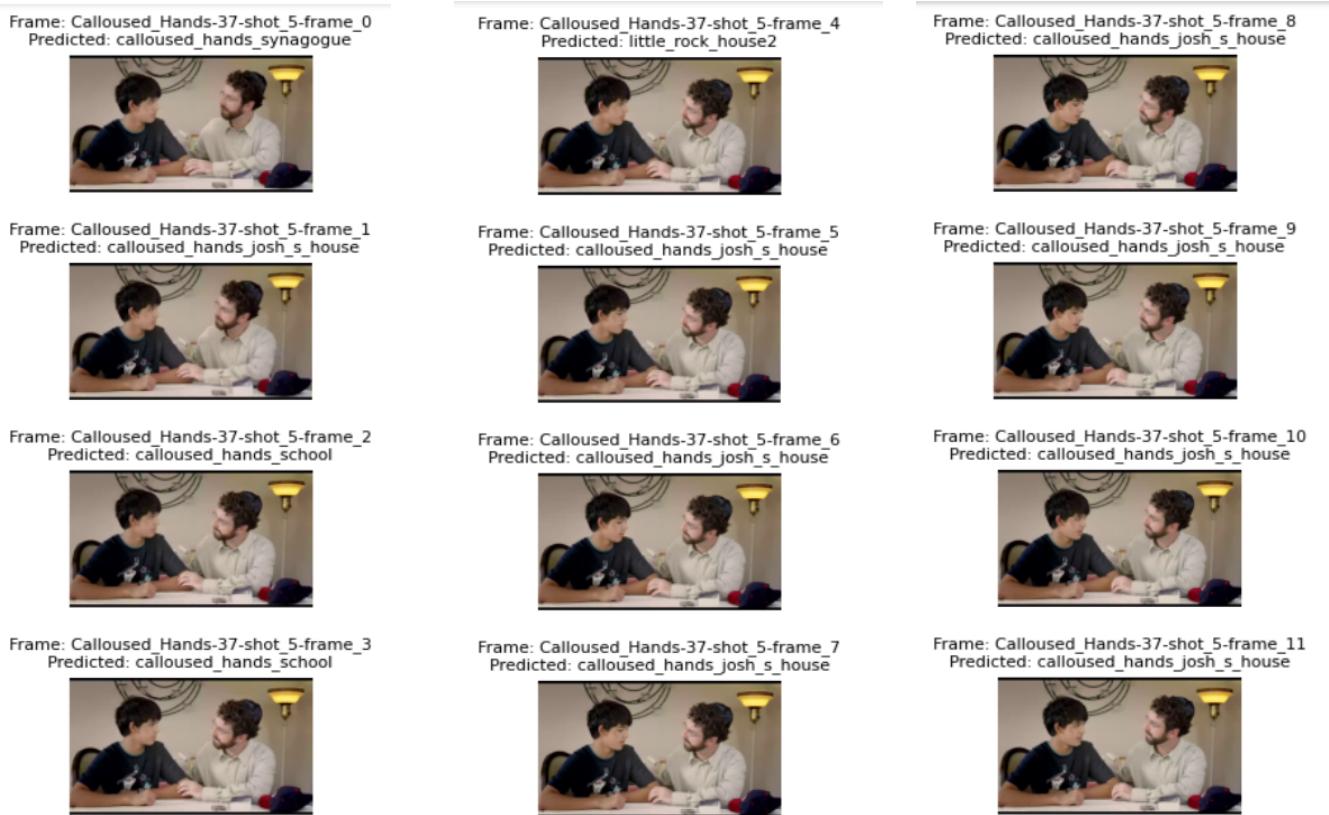
Bảng 3: Kết quả đánh giá trên bản thử nghiệm 500 ảnh

	Correct/Total (176 shots)	Accuracy
ResNet-18	101	0.574
MobileNetv2	107	0.608
VGG-16	99	0.563

4.5.3 Kết quả thực nghiệm

Sau đây là một số kết quả đạt được trong quá trình thực nghiệm:

ResNet18:



Hình 15: Một số ảnh kết quả khi test shot 5, scene 37, Calloused Hands

-----Predictions in a shot-----
 calloused_hands_synagogue: 12 predicted/55 frames
 calloused_hands_josh_s_house: 17 predicted/55 frames
 calloused_hands_school: 2 predicted/55 frames
 little_rock_house2: 2 predicted/55 frames
 calloused_hands_rabbi_s_house: 21 predicted/55 frames
 liberty_kid_house1: 1 predicted/55 frames
 => Calloused_Hands-37-shot_5 is: calloused_hands_rabbi_s_house (21 predictions)

Hình 16: Kết quả test trên từng frame

MobileNetv2:

Frame: Calloused_Hands-2-shot_1-frame_0
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_3
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_6
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_1
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_4
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_7
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_2
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_5
Predicted: calloused_hands_baseball_field



Frame: Calloused_Hands-2-shot_1-frame_8
Predicted: calloused_hands_baseball_field



Hình 17: Một số ảnh kết quả khi test shot 1, scene 2, Calloused Hands

-----Predictions in a shot-----

calloused_hands_baseball_field: 12 predicted/12 frames

=> Calloused_Hands-2-shot_1 is: calloused_hands_baseball_field (12 predictions)

Hình 18: Kết quả test trên từng frame

VGG-16:



Hình 19: Một số ảnh kết quả khi test shot 1, scene 37, Little Rock

```
-----Predictions in a shot-----
calloused_hands_josh_s_house: 1 predicted/42 frames
little_rock_gallery: 41 predicted/42 frames
=> little_rock-37-shot_1 is: little_rock_gallery (41 predictions)
```

Hình 20: Kết quả test trên từng frame

PHẦN 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN²

5.1 Kết luận

Ta thấy khi huấn luyện mô hình phân loại trên tập train với bản thử nghiệm 500 ảnh sẽ cho hiệu quả cao hơn hai bản thử nghiệm còn lại, do tập huấn luyện đủ lớn và đa dạng nên khả năng tổng quát hóa tốt hơn.

Ở cả ba bản thử nghiệm, MobileNetv2 cho hiệu suất cao nhất do được thiết kế để có kiến trúc lightweight với sự kết hợp của các lớp depthwise separable convolution và bottleneck structures giúp tối ưu hóa và giảm lượng tham số và tính toán, tiết kiệm thời gian trong quá trình huấn luyện. Kế tiếp là Resnet18 đứng thứ hai về độ hiệu quả do giữ lại kiến thức đã học được từ ImageNet, giúp tối ưu hóa mô hình để phân loại các đối tượng chính xác trên tập dữ liệu mới. Tuy nhiên, ResNet-18 có kích thước mô hình lớn hơn so với MobileNetV2. Điều này có thể ảnh hưởng đến khả năng triển khai và tính toán, đặc biệt là trên các thiết bị có tài

nguyên hạn chế. Cho hiệu suất thấp nhất trong ba mô hình là VGG-16, quá trình huấn luyện VGG-16 tốn khá nhiều thời gian so với hai mô hình còn lại do lượng tham số khá lớn, điều này có thể làm tăng khả năng overfitting trên tập dữ liệu nhỏ và làm giảm khả năng tổng quát hóa trên dữ liệu mới.

5.2 Hướng phát triển

- Dành thời gian để thử nghiệm hướng còn lại là tiếp cận theo cấp độ tham số, meta-learning và so sánh, đánh giá giữa hai hướng tiếp cận.
- Nghiên cứu và áp dụng thêm nhiều phương pháp tăng cường hình ảnh để phục vụ cho nhiều bài toán ít dữ liệu một cách hiệu quả.
- Cải tiến và sử dụng các mô hình phức tạp hơn cho việc phân loại hình ảnh nhằm nâng cao hiệu suất trong việc nhận diện địa điểm ngay trên dữ liệu mới.
- Tích hợp với hệ thống web giúp người dùng chỉ việc tải lên video hoặc một shot nhỏ của video thì hệ thống sẽ trả về kết quả là tên địa điểm mà shot đó được quay. Mở rộng quy mô lên tất cả địa điểm trong một thành phố,...

TÀI LIỆU THAM KHẢO

- [1] <https://neptune.ai/blog/understanding-few-shot-learning-in-computer-vision>
- [2] <https://paperswithcode.com/task/few-shot-learning>
- [3] https://imgaug.readthedocs.io/en/latest/source/examples_.html
- [4] <https://nttuan8.com/bai-9-transfer-learning-va-data-augmentation/>
- [5] <https://viblo.asia/p/gioi-thieu-mang-resnet-vyDZOa7R5wj>
- [6] <https://www.geeksforgeeks.org/vgg-16-cnn-model/>
- [7] <https://paperswithcode.com/method/mobilenetv2>