BIRDS SPECIES CLASSIFICATION

Hồ Mỹ Hạnh, Đinh Hoàng Linh Đan, Trần Nguyễn Quỳnh Anh
Trường đại học Công Nghệ Thông Tin - ĐHQG TP.HCM

***Tóm tắt nội dung*—Phân loại các loài chim ngày càng nhận được nhiều sự chú ý trong lĩnh vực thị giác máy tính, vì những ứng dụng đầy hứa hẹn của nó trong nghiên cứu sinh học và môi trường. Nhận dạng các loài chim là một trong những nhiệm vụ đầy thách thức với người quan sát chim do sự giống nhau về hình dáng, kích thước, màu sắc. Vì vậy cần một hệ thống dựa trên hình ảnh để giúp con người xác định được loài chim tốt hơn. Đồ án này nhằm mục đích áp dụng một số mạng học sâu và so sánh để đưa ra mô hình tốt nhất. Đề tài được thực hiện trên dữ liệu gồm 70626 ảnh chứa 450 loài chim được thu thập từ Kaggle. Nhóm sử dụng accuracy và macro average matrix để đánh giá các mô hình, từ đó chọn ra mô hình tốt nhất để phát triển hệ thống sau này.**

***Index Terms*—classification, image**

NHỮNG VẤN ĐỀ ĐÃ GIẢI QUYẾT

A. Ưu và nhược điểm của các thuật toán

1) Inception V3: Mô hình Inception V3 có các hiệu chỉnh giúp nó có tỉ lệ lỗi thấp hơn, độ chính xác cao hơn và thời gian training thấp hơn các phiên bản tiền nhiệm như factorization thành các lớp convolution nhỏ hơn, các lớp convolution bất đối xứng, tận dụng Auxiliary Classifiers và sử dụng hiệu quả giảm kích thước grid.

2) EfficientNet B0: Mô hình EfficientNet được tiếp cận theo hướng mới so với các mô hình khác. Mô hình tập trung vào việc mở rộng tham số theo cả ba chiều bao gồm độ sâu, độ rộng và độ phân giải của mạng. Mô hình cho phép giảm chi phí tính toán mà vẫn đảm bảo tính hiệu quả.

3) MobileNet V3: MobileNet sử dụng Depthwise Separable Convolutions để giảm số lượng tính toán, giảm số lượng params, đồng thời có thể thực hiện trích xuất đặc trưng một cách tách biệt trên các channel khác nhau.

B. Số lượng tham số của các thuật toán

	InceptionV3	EfficientNetB0	MobileNetV3
Params	22 724 834	4 049 571	7 486 554

I. PHÁT BIỂU BÀI TOÁN

A. Image classification

Phân loại hình ảnh (Image classification) là một trong những tác vụ của thị giác máy tính, ở đó thuật toán xem xét và dán nhãn cho hình ảnh từ một tập danh mục được xác định và đào tạo trước.

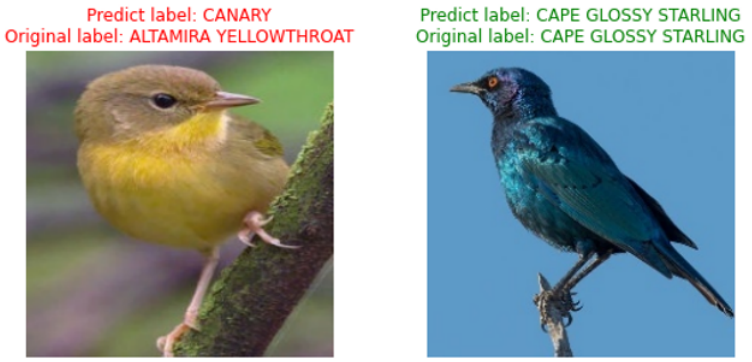
Ví dụ, với một tập các hình ảnh, mỗi hình ảnh mô tả một con mèo hoặc một con chó, thuật toán sẽ "quan sát" toàn bộ dữ liệu và dựa trên hình dạng, màu sắc để hình thành giả thuyết liên quan đến nội dung của ảnh. Kết quả thu được là từ tập dữ liệu ban đầu, các hình ảnh chó/mèo đã được phân loại một cách tự động.

Thực tế, thị giác góp phần tạo nên 80-85 % nhận thức của con người về thế giới. Hàng ngày, mỗi người phải thực hiện phân loại trên bất kỳ dữ liệu hình ảnh nào mà chúng ta bắt gặp. Do đó, mô phỏng nhiệm vụ phân loại với sự trợ giúp của mạng nơ-ron là một trong những ứng dụng đầu tiên của thị giác máy tính mà các nhà nghiên cứu nghĩ đến.

B. Birds classification

Thế giới loài chim ngày càng phong phú về chủng loại, hình dáng, màu sắc... Điều này dẫn đến việc nhận biết bằng mắt thường là vô cùng khó khăn. Vì vậy bài toán phân loại loài chim giúp con người nhận dạng các loài chim dễ dàng hơn chỉ bằng hình ảnh. Từ đó phục vụ tốt cho quá trình bảo tồn các loài chim, ngoài ra còn giúp ích cho việc giảng dạy giúp học sinh có thể nhận biết về một số loài chim.

- Input: ảnh chứa 1 loài chim
- Output: dự đoán tên của loài chim và tên loại chim đó. Nếu dự đoán đúng loài chim chữ sẽ hiện màu xanh lá, và nếu dự đoán sai thì chữ sẽ hiện màu đỏ.



Hình 1. Hình ảnh output

II. BỘ DỮ LIỆU

Bộ dữ liệu sử dụng là BIRDS 450 SPECIES - IMAGE CLASSIFICATION thuộc Kaggle do tác giả Gerry Piosenka tổng hợp và chỉnh sửa lại từ ảnh trên internet về 450 loài chim, bao gồm 70,626 training images, 2250 test images (5 ảnh mỗi loài) và 2250 validation images (5 ảnh mỗi loài).

Bộ dữ liệu này chứa các hình ảnh màu với độ phân giải cao (224x224) ở định dạng jpg, với mỗi ảnh chỉ chứa một con chim và chiếm hơn 50% diện tích bức ảnh.

Hình 2. Ví dụ một số loài chim trong bộ dữ liệu

III. Mô hình

Trong đề tài này nhóm sử dụng ba mô hình học sâu: Inception V3, EfficientNet B0, MobileNet V3. Để chọn tham số tốt nhất cho mỗi mô hình, nhóm đã thử nghiệm với nhiều bộ tham số khác nhau và chọn bộ cho ra độ chính xác cao nhất để tiến hành huấn luyện.

Nhóm sử dụng protocol như sau:

- Learning rate: 0.1, 0.01, 0.001
- Batch size: 32, 64
- Epoch: 5
- Optimizer: Adam, SGD, RMSprop
- Loss: categorical_crossentropy
- Với SGD có sử dụng momentum (0.9, 0.95) và không sử dụng momentum

A. Inception V3

Mô hình Inception V3 dựa trên bài báo gốc: *"Rethinking the Inception Architecture for Computer Vision"* của Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna được đề xuất vào tháng 12/2015. Inception V3 là kế thừa của Inception V1 – mô hình đã dành chiến thắng ở cuộc thi ImageNet năm 2015 bao gồm 24 triệu tham số.

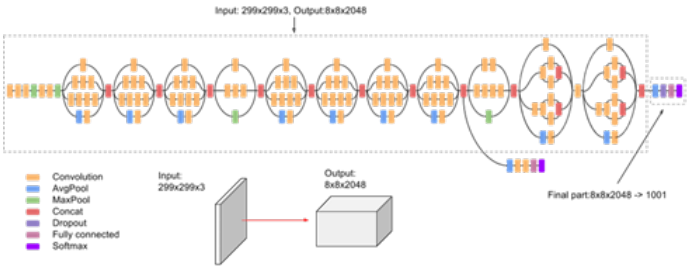
Bằng cách giảm số lượng parameters liên quan đến mạng, factorized các lớp convolutions hay thay thế những tích chập lớn bằng những tích chập nhỏ hơn làm giúp giảm computational costs, và quá trình training sẽ diễn ra nhanh hơn.

Khi các lớp convolution lớn được factorized thành các convolution nhỏ hơn, điều gì sẽ xảy ra nếu ta factorize nó nhỏ hơn nữa, một giải pháp làm cho mô hình hiệu quả hơn là Asymmetric convolutions - các tích chập bất đối xứng.

Mô hình sử dụng Auxiliary classifier - bộ phân loại phụ trợ một cách hiệu quả. Đây là một CNN nhỏ được chèn vào giữa các layer trong quá trình training, loss phát sinh sẽ được thêm vào loss của network chính) như một regularizer - bộ hiệu chỉnh. Mục tiêu của việc sử dụng Auxiliary classifier là để cải thiện sự hội tụ của các neural networks rất sâu. Các Auxiliary classifier không dẫn đến bất kỳ sự cải thiện nào trong giai đoạn đầu của training. Nhưng về cuối của quá trình training, mạng có Auxiliary classifier cho thấy độ chính xác cao hơn so với mạng không có Auxiliary classifier.

Giảm grid size của các feature map thường được giải quyết bởi các lớp pooling, như max pooling và average pooling, tuy nhiên, để giải quyết được vấn đề thắt cổ chai (representational bottlenecks) về computational cost, Inceptionet V3 đã sử dụng một phương pháp hiệu quả hơn là activation dimension của bộ lọc network được mở rộng. Và điều này được thực hiện bằng cách sử dụng hai khối convolution song song và pooling sau đó được nối với nhau.

Sau cùng, tất cả các khái niệm trên đã được hợp nhất tạo thành mô hình hoàn chỉnh dưới đây.



Hình 3. Kiến trúc mô hình Inception V3

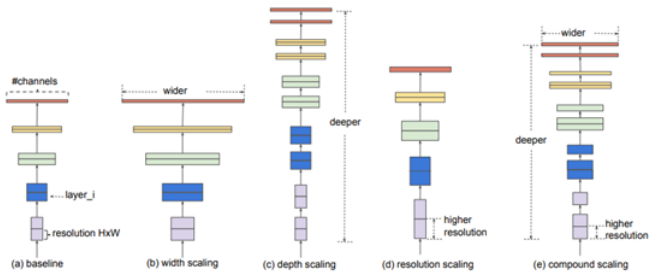
B. EfficientNet B0

Mạng tích chập (ConvNet) đã thực sự trở thành một trong những mô hình được sử dụng nhiều nhất trong Computer Vision. Đa số các mô hình ConvNet đều đạt được một kết quả khả quan và nhất định với lượng resources cố định. Do đó nhằm tăng độ chính xác mô hình thì chúng ta thường có 3 hướng sau:

- Tăng độ sâu của mô hình.
- Tăng độ rộng của từng layer trong mô hình.
- Cải thiện chất lượng của đầu vào (tăng chất lượng, kích thước ảnh).

Tuy nhiên các cách để scale mô hình bây giờ đa số là chọn một trong ba hướng trên để tinh chỉnh hoặc thử nghiệm với các số random hay thậm chí là tìm kiếm gridsearch trong một không gian lớn và việc này có thể sẽ khó khăn khi ta cần tối ưu đối với một mạng ConvNet lớn.

Năm 2019, nhóm nghiên cứu Google gồm Mingxing Tan và Quoc V.Le đã phát hành bài báo liên quan tới họ CNN mới: *"EfficienNet Rethinking Model Scaling for Convolutional Neural Networks"*. Ý tưởng của nhóm tác giả hướng đến là việc phối hợp và cân bằng các thông số một cách có hệ thống để mang đến hiệu suất tốt hơn.



Hình 4. Các cách scale mô hình ConvNet

Hình ảnh trên mô tả 3 cách để scale mô hình ConvNet trong đó gồm (b) tăng độ rộng trên từng layer của mô hình , (c) tăng độ sâu của mô hình hay (d) tăng chất lượng ảnh đầu vào nhằm mô hình học được những trích xuất rõ ràng nhất của hình ảnh (hình ảnh rõ nét sẽ dễ dàng đưa ra được đặc trưng hay tính chất của 1 vật). Cách cuối cùng là kết hợp của 3 cách trên với bộ tham số tùy chỉnh.

Thực tế, EfficientNet B0 - phiên bản đơn giản nhất - đã trả về một kết quả ấn tượng với độ chính xác lên đến 77% khi xét trên bộ dữ liệu ImageNet.

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Hình 5. Kiến trúc EfficientNet B0

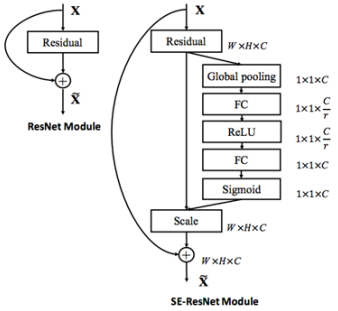
C. MobileNet V3

MobileNet V3 được phát triển bởi đội ngũ Google, được giới thiệu lần đầu năm 2017.

Mô hình sử dụng các tính chập tích tên DSC (Depthwise Separable Convolution) nhằm giảm kích thước mô hình và giảm độ phức tạp tính toán.

Ý tưởng của Depthwise Separable Convolution là chia phép convolution làm 2 phần: Depthwise convolution và Pointwise convolution.

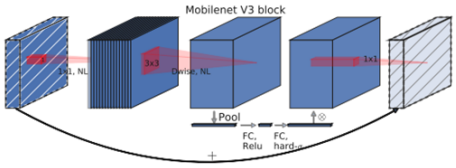
Mô hình MobileNet V3 thêm Squeeze and Excitation (SE) vào block Residual để tạo thành một kiến trúc có độ chính xác cao hơn.



Hình 6. Residual Block

SE-ResNet áp dụng thêm một nhánh Global pooling có tác dụng ghi nhận bối cảnh của toàn bộ layer trước đó. Kết quả sau cùng ở nhánh này ta thu được một véc tơ global context được dùng để scale đầu vào X.

Tương tự như vậy SE được tích hợp vào kiến trúc của một residual block trong mobilenetV3 như sau:



Hình 7. MobileNetV3 block

Tại layer thứ 3 có một nhánh Squeeze and Excitation có kích thước bằng 1 x 1 có tác dụng tổng hợp global context. Nhánh này lần lượt đi qua các biến đổi FC -> Relu -> FC -> hard sigmoid . Cuối cùng được nhân trực tiếp vào nhánh input để scale input theo global context. Các kiến trúc còn lại hoàn toàn giữ nguyên như MobileNet V2.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Hình 8. Kiến trúc mô hình MobileNetV3

IV. ĐÁNH GIÁ

A. Độ đo đánh giá

Nhóm sử dụng **Accuracy** và **Macro average matrix** để đánh giá các mô hình

- Accuracy: tỷ lệ phần trăm dự đoán đúng trong tổng số lượng mẫu.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Hình 9. Công thức của accuracy

- Macro average matrix

$$Precision_{macro} = \frac{1}{Class_{num}} \sum_{class} \frac{TruePositive_{class}}{TruePositive_{class} + FalsePositive_{class}}$$

$$Recall_{macro} = \frac{1}{Class_{num}} \sum_{class} \frac{TruePositive_{class}}{TruePositive_{class} + FalseNegative_{class}}$$

$$F1 - score_{macro} = \frac{1}{1/Precision_{macro} + 1/Recall_{macro}}$$

where $Class_{num}$ is the number of classes.

Hình 10. Công thức của macro avarage matrix

B. Kết quả thử nghiệm

	InceptionV3	EfficientNetB0	MobileNetV3
Accuracy	0.9458	0.9689	0.9258
F1 - score	0.9506	0.9716	0.9335
Precision	0.9556	0.9743	0.9414
Recall	0.9458	0.9689	0.9258
Params	22 724 834	4 049 571	7 486 554

V. KẾT LUẬN

Sau khi thử nghiệm bộ dữ liệu trên ba mô hình với từng bộ tham số tốt nhất, nhóm nhận thấy cả ba mô hình đều cho ra kết quả khá tốt (độ chính xác > 92%). Trong đó mô hình EfficientNet B0 cho ra kết quả vượt trội hơn với số lượng tham số ít nhất.

Do bộ dữ liệu mà nhóm thử nghiệm có chất lượng cao khi mỗi hình chỉ chứa một loại chim với diện tích gần như chiếm trọn khung hình, nên khi sử dụng các mô hình học sâu để huấn luyện dữ liệu ta sẽ nhận được kết quả tốt. Vì vậy nhóm có thể sẽ tiếp tục thử nghiệm với những bộ dữ liệu khác (có nhiều hơn một loại trong hình, vị trí của loài chim xa gần khác nhau...) để có được đánh giá khách quan hơn.

TÀI LIỆU

[1] Gerry. "BIRDS 400 - SPECIES IMAGE CLASSIFICATION." Www.kaggle.com, www.kaggle.com/datasets/gpiosenka/100-bird-species.

[2] "Advanced Guide to Inception v3 on Cloud TPU | Cloud TPU | Google Cloud." Google Cloud, 28 Jan. 2019, cloud.google.com/tpu/docs/inception-v3-advanced.

[3] "Papers with Code - Inception-v3 Explained." Paperswithcode.com, paperswithcode.com/method/inception-v3.

[4] Raj, Bharath. "A Simple Guide to the Versions of the Inception Network." Towards Data Science, Towards Data Science, 29 May 2018, towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202.

[5] "Inception v3 Model Architecture." OpenGenus IQ: Computing Expertise Legacy, 5 Sept. 2021, iq.opengenus.org/inception-v3-model-architecture/.

[6] Mingxing Tan, Quoc V. Le , "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019.

[7] "OpenGenus," [Online]. Available: https://iq.opengenus.org/efficientnet/.

[8] H. N. Thanh, "Viblo," 16 September 2021. [Online]. Available: https://viblo.asia/p/cnn-architecture-series-1-mobilenets-mo-hinh-gon-nhe-cho-mobile-applications-1VgZvJV1ZAw. [Accessed 16 September 2021].

[9] P. D. Khanh, "Khoa hoc du lieu," 19 September 2020. [Online]. Available: https://phamdinhkhanh.github.io/2020/09/19/MobileNet.html.