

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



**TOPIC: IMAGE CAPTIONING
FOR SPORT BALLS**

CLASS: CS114.N11.KHCL

Học kì I (NH: 2022 – 2023)

GIẢNG VIÊN HƯỚNG DẪN: PGS.TS. LÊ ĐÌNH DUY
TH.S. PHẠM NGUYỄN TRƯỜNG AN

SINH VIÊN THỰC HIỆN: LÊ TẤN LỘC 20521546
BÙI QUỐC THỊNH 20520934
NGUYỄN HỒNG ANH THƯ 20520313

TP Hồ Chí Minh, tháng 02 năm 2023

1. Tổng quan.....	2
1.1. Giới thiệu bài toán	2
1.2. Ngữ cảnh ứng dụng.....	2
1.3. Phương pháp xây dựng bộ dữ liệu	3
2. Pipeline	4
3. Encoder.....	5
3.1. Giới thiệu	5
3.2. Vấn đề của EfficientNetV1	7
3.3. EfficientNet_V2	9
4. Decoder	13
4.1. Tiền xử lý caption.....	13
4.2. Decoder Transformer	14
4.2.1. Positional Encoding.....	14
4.2.2. Masked Multi-Head Attention.....	16
5. Phương pháp đánh giá	18
5.1. BLEU (Bilingual Evaluation Understudy)	18
5.2. ROUGE (Recall-Oriented Understudy for Gisting Evaluation)	19
5.3. CIDEr (Consensus-based Image Description Evaluation).....	19
6. Thực nghiệm, kết quả và nhận xét	21
6.1. Cài đặt thực nghiệm.....	21
6.2. Kết quả thực nghiệm	21
6.3. So sánh và nhận xét	23
7. Hướng phát triển và cải tiến.....	24
7.1. Về data	24
7.2. Về model	24
Tài liệu tham khảo.....	25

Lời Cảm Ơn

Lời đầu tiên, nhóm chúng em xin được gửi lời cảm ơn chân thành đến thầy!!! Cảm ơn thầy đã giảng dạy, chỉ dẫn và đồng hành cùng chúng em trong học kì vừa qua. Cảm ơn thầy đã luôn nhiệt tình truyền đạt kiến thức cho chúng em trong các buổi học, đã luôn tận tâm trong từng buổi giảng dạy.

Giờ cũng là thời điểm kết thúc môn học, bên cạnh đồ án cuối kỳ mà nhóm thực hiện, chúng em muốn gửi đến thầy lời cảm ơn và lời chúc sức khỏe. Chúc thầy đạt được nhiều thành công trong sự nghiệp giáo dục!!

Tóm tắt nội dung

Phát triển một hệ thống máy tính có thể hiểu được thế giới thị giác và giao tiếp với chúng ta bằng ngôn ngữ là một trong những mục tiêu lớn của trí tuệ nhân tạo. Để hiện thực hóa giấc mơ này, vô số bài toán đã được đặt ra, trong đó có *Image Captioning*. Bài toán này nhận đầu vào là một hình ảnh và cố gắng sinh ra một câu mô tả bằng ngôn ngữ tự nhiên cho ảnh đó.

Hiện nay, phương pháp tiếp cận giải quyết bài toán này là áp dụng khai thác đặc trưng ảnh qua CNN và sử dụng RNN để sinh câu mô tả. Tuy nhiên, phần lớn các nghiên cứu hiện tại chủ yếu tạo chú thích bằng tiếng Anh hoặc tiếng Trung cho ảnh. Trong đồ án này, chúng em tập trung giải quyết bài toán *Image Captioning* cho tiếng Việt – ngôn ngữ đang có gần 100 triệu người sử dụng. Chúng em sẽ kế thừa bộ dữ liệu *UIT-ViIC* - bộ dữ liệu đầu tiên cho bài toán *Image Captioning* cho tiếng Việt và xây dựng mở rộng thêm bộ dataset này theo phương pháp được đề xuất trong bài báo nghiên cứu. Hướng tiếp cận của chúng em cho bài toán trên sẽ sử dụng *EfficientNetV2* để trích xuất đặc trưng ảnh và *Transformer* cho việc hình thành câu mô tả. Chúng em hy vọng kết quả đạt được sẽ tạo động lực cho các nghiên cứu sâu hơn về lĩnh vực *Image Captioning* trên tiếng Việt cũng như đa ngôn ngữ.

1. Tổng quan

1.1. Giới thiệu bài toán

Image Captioning là bài toán tự động tạo ra mô tả văn bản cho một hình ảnh đầu vào. Nhiệm vụ của Image Captioning là phân tích các đặc trưng của hình ảnh và sử dụng chúng để tạo ra một câu hoặc đoạn văn miêu tả nội dung của hình ảnh đó một cách tự động và có ý nghĩa.

Để giải quyết bài toán này, cần kết hợp sử dụng các kỹ thuật học sâu (deep learning) như mạng nơ-ron tích chập (Convolutional Neural Network - CNN) để trích xuất đặc trưng của hình ảnh, và mô hình ngôn ngữ (Language Model) để tạo ra mô tả cho hình ảnh.

Input: Một tấm ảnh có chứa môn thể thao có bóng

Output: Câu mô tả bằng Tiếng Việt cho bức ảnh



Những người đàn ông đang chơi bóng đá trên sân.

1.2. Ngữ cảnh ứng dụng

Một số ứng dụng của Image Captioning trong đời sống hiện nay như:

- Hỗ trợ người khiếm thị: Image Captioning có thể giúp người khiếm thị hiểu được nội dung của các hình ảnh mà họ không thể nhìn thấy. Bằng cách sử dụng các phần mềm đọc mô tả văn bản cho hình ảnh, người khiếm thị có thể sử dụng điện thoại thông minh hoặc máy tính để xem và hiểu được nội dung của các hình ảnh.
- Mạng xã hội: Image Captioning cũng được sử dụng để tạo ra mô tả cho các bức ảnh trên các mạng xã hội như Instagram, Facebook, Twitter, v.v. Điều này giúp cho người dùng có thể tìm kiếm và hiểu được nội dung của các bức ảnh một cách dễ dàng hơn.
- Máy tính xử lý ảnh: Image Captioning có thể được sử dụng trong các ứng dụng máy tính xử lý ảnh để giúp máy tính hiểu được nội dung của các hình ảnh. Các ứng dụng

có thể bao gồm xác định đối tượng trong hình ảnh, phân loại hình ảnh, phát hiện hành động, v.v.

- Giáo dục: Image Captioning cũng có thể được sử dụng trong giáo dục để giúp học sinh và sinh viên hiểu được nội dung của các bức ảnh trong sách giáo khoa hoặc tài liệu học tập.
- Các ứng dụng thương mại điện tử: Image Captioning có thể được sử dụng trong các ứng dụng thương mại điện tử để giúp khách hàng hiểu được nội dung của các sản phẩm và đặt hàng một cách chính xác hơn.

Tóm lại, Image Captioning là một bài toán quan trọng trong lĩnh vực Trí tuệ nhân tạo và có rất nhiều ứng dụng thực tế trong đời sống và công nghiệp.

1.3. Phương pháp xây dựng bộ dữ liệu

Dataset của nhóm sử dụng bao gồm 2 phần: bộ dữ liệu UIT-ViIC và bộ dữ liệu Flickr_sportballs.

Dataset	Train		Test	
	Ảnh	Caption	Ảnh	Caption
UIT-ViIC	3619	18101	231	1155
Flickr_sportballs	100	500	100	500

Bộ dữ liệu UIT-ViIC là bộ dữ liệu được các chuyên gia thu thập từ MS COCO và sử dụng các quy luật annotation để label caption Tiếng Việt cho các hình ảnh.

Bộ dữ liệu Flickr_sportballs là bộ dữ liệu do nhóm tự thu thập dựa trên bộ dữ liệu Flickr30k. Công việc cần làm là mỗi thành viên của nhóm sẽ gán nhãn caption Tiếng Việt cho các hình ảnh môn thể thao có bóng trong bộ dữ liệu Flickr30k dựa trên một số quy luật annotation nhất định. Để tạo nên bộ dữ liệu Flickr_sportballs, nhóm tiến hành theo các bước sau đây:

Bước 1: Lọc hình ảnh môn thể thao có bóng trong bộ dữ liệu Flickr30k

Dựa vào bộ dữ liệu Flickr30k có caption Tiếng Anh cho trước. Nhóm tiến hành tìm kiếm những keyword được cho là có liên quan đến môn thể thao có bóng.

Ví dụ: “bowling”, “tennis”, “baseball”, “basketball”, “football”,.....

Bước 2: Làm sạch nhiễu

Những hình ảnh có chứa keyword trên sẽ được chọn ra và tiến hành bước làm sạch nhiều. Nhiều ở đây được xem là những hình ảnh có chứa bóng nhưng không phải là một hình ảnh mang tính thể thao hoặc những hình ảnh có chứa keyword ta cần nhưng không miêu tả một môn thể thao có bóng (“basketball basket”, “baseball cap”, ...).

Ví dụ: “A man in a red apron wearing a **baseball cap** is sitting on a step.”

Bước làm sạch nhiều sẽ do các thành viên đánh giá khách quan hình ảnh đó có chứa nhiều hay không.

Bước 3: Xây dựng các quy định annotation

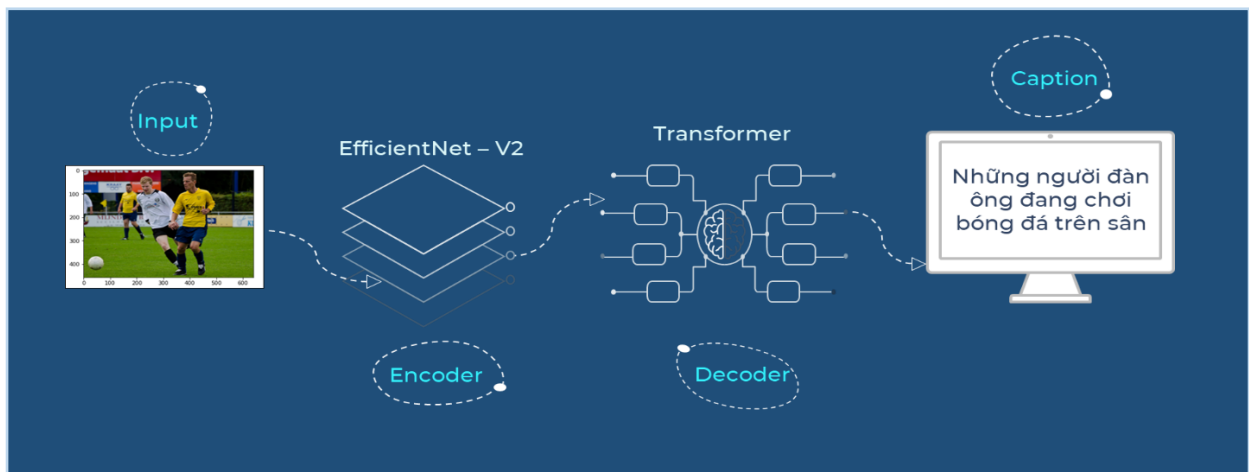
Theo khảo sát một số caption của hình ảnh trên mạng bị đánh giá là không tốt do caption nó đưa ra không trùng khớp với những gì hình ảnh miêu tả như tên người, tên địa điểm, thời gian, ... Bởi vì thế, để tạo ra những caption miêu tả đúng nghĩa chúng ta cần phải đặt ra những quy tắc chặt chẽ để tiến hành annotation. Dựa trên những quy tắc có sẵn do các chuyên gia đề ra để annotation, chúng ta có các quy tắc như sau:

- Mỗi caption chứa ít nhất 6 từ tiếng Việt.
- Chỉ miêu tả những hoạt động và đối tượng hiện hữu trong bức ảnh.
- Không đề cập đến tên của các địa điểm, sự vật (tên thành phố, tên người, tên tòa nhà, ...) và các con số cụ thể (ngày giờ, số phòng,...)
- Các caption phải được viết bằng thì hiện tại tiếp diễn (continuous present tense).
- Những ý kiến và cảm xúc cá nhân không được bao gồm trong caption.
- Những đối tượng không rõ ràng (bị che khuất, không hoàn chỉnh, ...) sẽ bị bỏ qua.
- Đối với những đối tượng có cùng loại hay đặc điểm với số lượng nhiều, annotators không cần phải đề cập chúng trong caption.

Sau khi xây dựng các quy tắc annotation các thành viên của nhóm sẽ tiến hành gán nhãn cho các hình ảnh trong bộ dữ liệu và kết hợp cùng sự giúp đỡ của các thành viên nhóm khác để tạo ra một bộ dữ liệu caption Tiếng Việt khách quan nhất.

2. Pipeline

Hướng tiếp cận phổ biến cho bài toán *Image captioning* là sử dụng một mô hình *encoder* để biểu diễn đặc trưng của ảnh. Đặc trưng này sau đó qua một mô hình *decoder* để tạo câu caption cho ảnh đầu vào. *Hình 1* minh họa pipeline nhóm sẽ sử dụng trong đề án này.

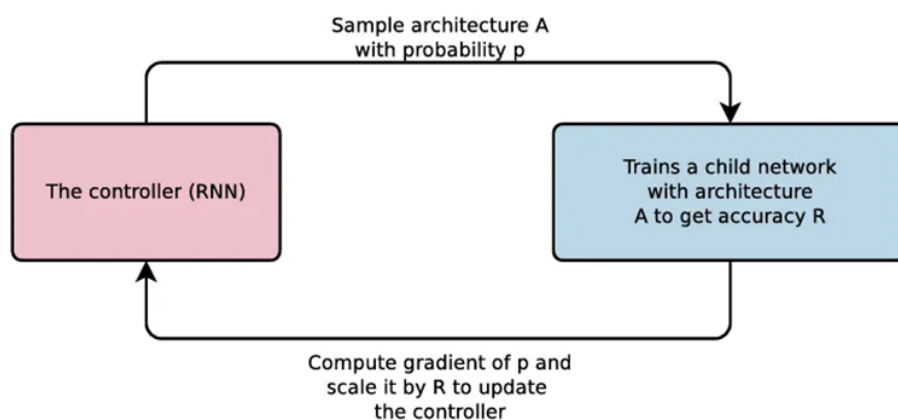


Hình 1 Pipeline bài toán Image Captioning

3. Encoder

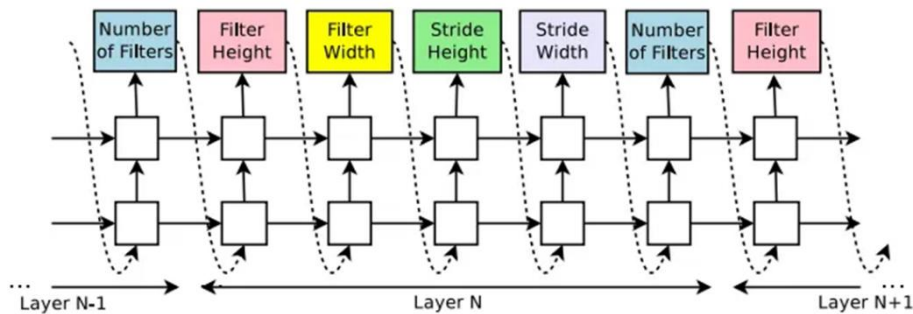
3.1. Giới thiệu

Ý tưởng của EfficientNet là sử dụng bộ điều khiển (mạng chẳng hạn như RNN) và lấy mẫu kiến trúc mạng từ không gian tìm kiếm có xác suất 'p'. Kiến trúc này sau đó được đánh giá bằng cách huấn luyện mạng đầu tiên, sau đó xác thực nó trên một bộ thử nghiệm để có được độ chính xác 'R'. Độ dốc của 'p' được tính toán và chia tỷ lệ theo độ chính xác 'R'. Kết quả (phần thưởng) được đưa đến bộ điều khiển RNN. Bộ điều khiển đóng vai trò là tác nhân, quá trình đào tạo, kiểm tra mạng đóng vai trò là môi trường và kết quả đóng vai trò là phần thưởng. Đây là vòng lặp của Học tăng cường (Reinforcement learning) phổ biến. Vòng lặp này chạy nhiều lần cho đến khi bộ điều khiển tìm thấy kiến trúc mạng mang lại phần thưởng cao (độ chính xác kiểm tra cao).



Bộ điều khiển RNN lấy mẫu các tham số kiến trúc mạng khác nhau — chẳng hạn như số lượng filters, độ cao filter, độ rộng filter, độ cao stride, và độ rộng stride cho mỗi lớp.

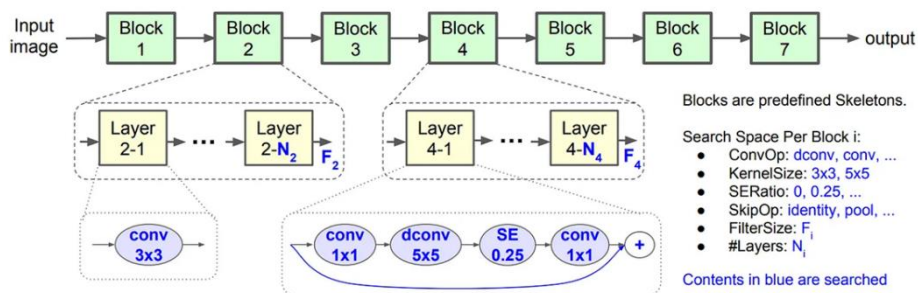
Các tham số này có thể khác nhau đối với từng lớp của mạng. Cuối cùng, mạng có kết quả (phần thưởng) cao nhất được chọn làm kiến trúc mạng cuối cùng.



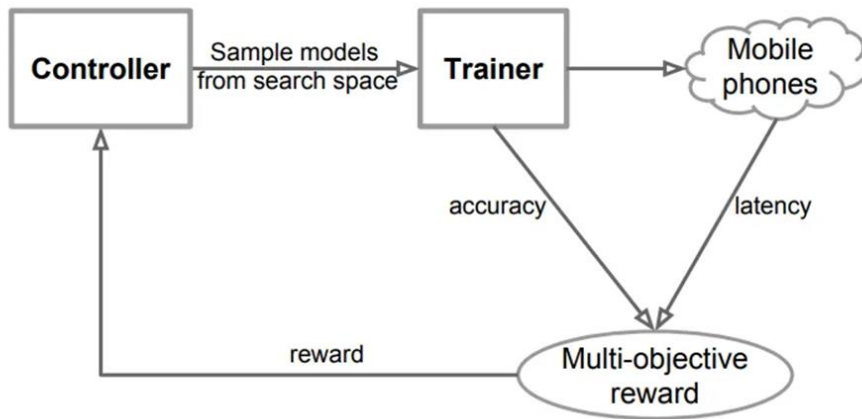
Mặc dù phương pháp này hoạt động tốt, nhưng một trong những vấn đề với phương pháp này là nó đòi hỏi một lượng lớn sức mạnh tính toán cũng như thời gian.

Các kiến trúc này không có các tham số khác nhau trong mỗi lớp, mà có một khối với nhiều lớp tích chập (còn gọi là ConvNet / CNN) và lớp tổng hợp, và trong toàn bộ kiến trúc mạng, các khối này được sử dụng nhiều lần. Các tác giả đã sử dụng ý tưởng này để tìm các khối như vậy bằng bộ điều khiển Reinforcement learning và chỉ cần lặp lại các khối này N lần để tạo kiến trúc NASNet có thể mở rộng.

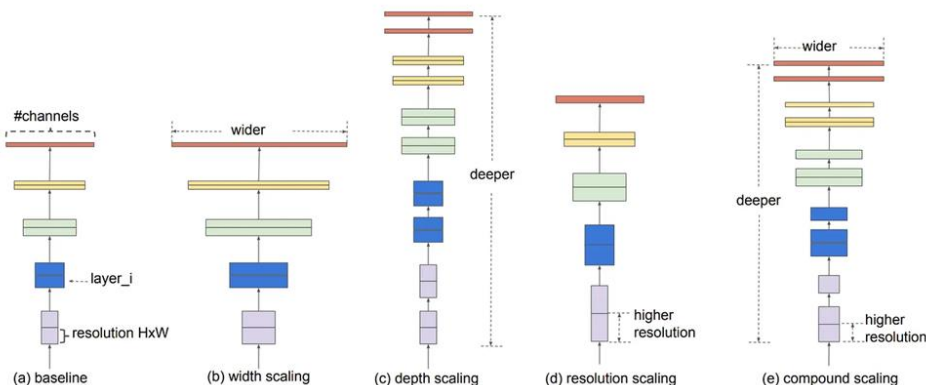
Trong mạng này, các tác giả đã chọn 7 khối và một lớp của khối được lấy mẫu và lặp lại cho mỗi khối.



Ngoài các tham số này, một tham số rất quan trọng khác đã được xem xét khi quyết định phần thưởng, tham số này được đưa vào bộ điều khiển và đó là “độ trễ”. Vì vậy, đối với MnasNet, các tác giả đã xem xét cả độ chính xác và độ trễ để tìm ra kiến trúc mô hình tốt nhất. Điều này được thể hiện trong hình dưới. Điều này làm cho kiến trúc trở nên nhỏ gọn và nó có thể chạy trên thiết bị di động hoặc thiết bị biên.



Quy trình tìm kiếm kiến trúc EfficientNet rất giống với MnasNet, nhưng thay vì coi 'độ trễ' là tham số phần thưởng, 'FLOP (floating point operations per second)' đã được xem xét. Tìm kiếm theo tiêu chí này đã cho ra một mô hình cơ sở được gọi là EfficientNetB0. Tiếp theo, tăng tỷ lệ độ sâu, chiều rộng và độ phân giải hình ảnh của mô hình cơ sở (sử dụng tìm kiếm dạng lưới hay vét cạn) để tạo thêm 6 mô hình, từ EfficientNetB1 đến EfficientNetB7. Tỷ lệ này được hiển thị trong hình dưới.

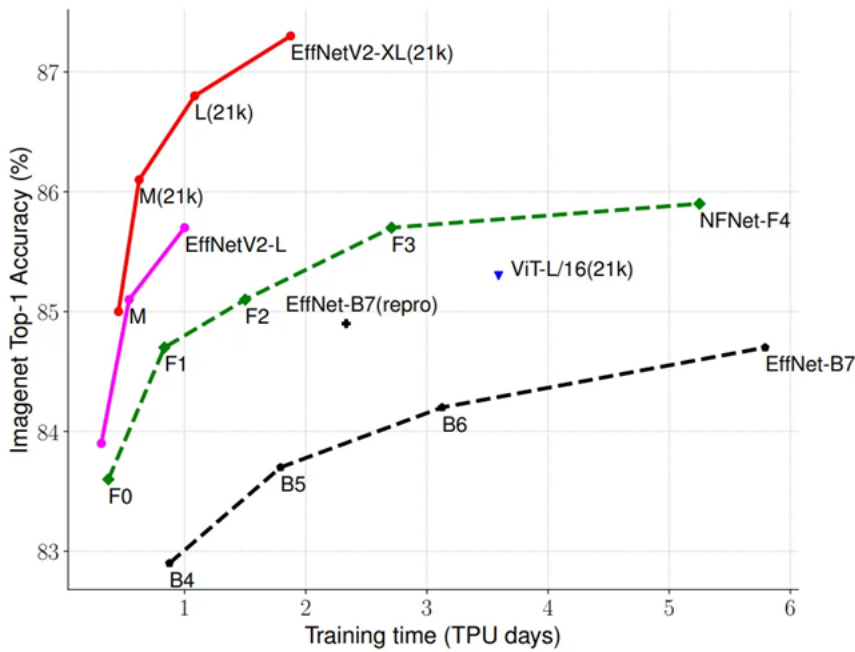


3.2. Vấn đề của EfficientNetV1

EfficientNetV2 tiến xa hơn một bước so với EfficientNet để tăng tốc độ đào tạo và hiệu quả tham số. Mạng này được tạo bằng cách sử dụng kết hợp chia tỷ lệ (chiều rộng, độ sâu, độ phân giải) và tìm kiếm kiến trúc neural. Mục tiêu chính là tối ưu hóa tốc độ đào tạo và hiệu quả tham số. Ngoài ra, lần này không gian tìm kiếm cũng bao gồm các khối tích chập mới như Fused-MBConv. Cuối cùng, các tác giả đã thu được kiến trúc EfficientNetV2 nhanh hơn nhiều so với các mô hình hiện đại trước đây và mới hơn, đồng thời nhỏ hơn nhiều (lên tới 6,8 lần). Điều này được thể hiện trong hình dưới.

Hình Parameter efficiency cho thấy rõ ràng rằng EfficientNetV2 có 24 triệu tham số, trong khi Vision Transformer (ViT) có 86 triệu tham số. Phiên bản V2 cũng có gần một nửa thông số của EfficientNet ban đầu. Mặc dù nó làm giảm đáng kể kích thước tham số, nhưng nó vẫn duy trì độ chính xác tương tự hoặc cao hơn so với các mô hình khác trên bộ dữ liệu ImageNet.

Ngoài ra cũng thực hiện progressive learning, đây là một phương pháp để tăng dần kích thước hình ảnh cùng với các quy định như bỏ học và tăng cường dữ liệu. Phương pháp này tiếp tục tăng tốc đào tạo.



(a) Training efficiency.

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

EfficientNets thường đào tạo nhanh hơn các mô hình CNN lớn khác. Tuy nhiên, khi độ phân giải hình ảnh lớn được sử dụng để huấn luyện các mô hình (mô hình B6 hoặc B7), quá trình huấn luyện diễn ra chậm. Điều này là do các mô hình EfficientNet lớn hơn yêu cầu kích thước hình ảnh lớn hơn để có được kết quả tối ưu và khi sử dụng hình ảnh lớn hơn, kích thước lô cần phải được hạ xuống để phù hợp với những hình ảnh này trong bộ nhớ GPU/TPU, khiến quá trình tổng thể chậm lại.

Trong các lớp đầu tiên của kiến trúc mạng, các lớp tích chập theo chiều sâu (MBConv) hoạt động chậm. Các lớp tích chập theo chiều sâu thường có ít tham số hơn các lớp tích chập thông thường, nhưng vấn đề là chúng không thể tận dụng triệt để các modern accelerator. Để khắc phục vấn đề này, EfficientNetV2 sử dụng kết hợp MBConv và Fused MBConv để đào tạo nhanh hơn mà không cần tăng các tham số.

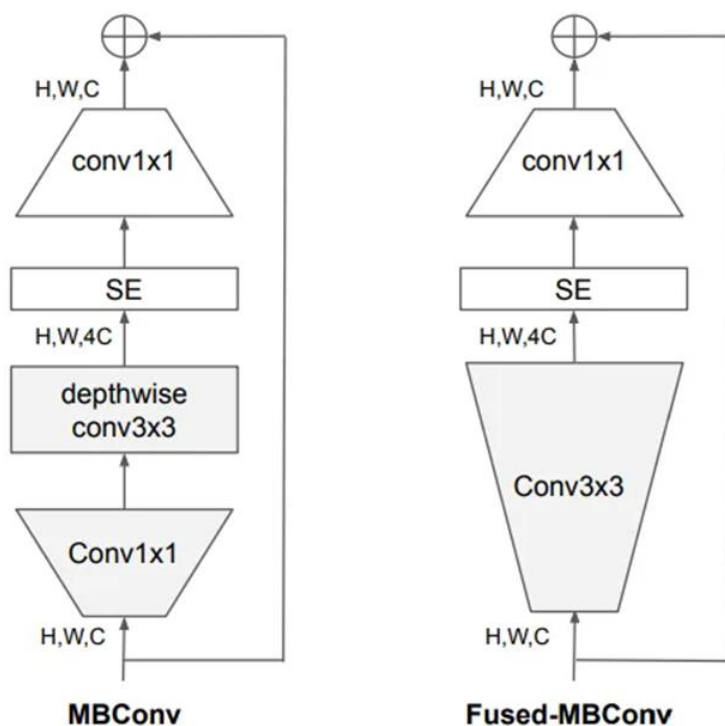
Tỷ lệ bằng nhau được áp dụng cho chiều cao, chiều rộng và độ phân giải hình ảnh để tạo các mô hình EfficientNet khác nhau từ B0 đến B7. Tỷ lệ bằng nhau của tất cả các lớp này không phải là tối ưu. Ví dụ: nếu độ sâu được chia tỷ lệ 2, tất cả các khối trong mạng sẽ được tăng tỷ lệ 2 lần, làm cho mạng trở nên rất lớn/sâu. Có thể tối ưu hơn nếu chia tỷ lệ một khối hai lần và khối kia 1,5 lần (tỷ lệ không đồng đều), để giảm kích thước mô hình trong khi vẫn duy trì độ chính xác tốt.

3.3. EfficientNet_V2

Như đã đề cập ở trên, khối MBConv thường không thể tận dụng triệt để các modern accelerator. Các lớp Fused-MBConv có thể sử dụng tốt hơn các trình tăng tốc máy chủ/thiết bị di động.

Lớp MBConv lần đầu tiên được giới thiệu trong MobileNets. Như đã thấy trong Hình 7, sự khác biệt duy nhất giữa cấu trúc của MBConv và Fused-MBConv là hai khối cuối cùng. Trong khi MBConv sử dụng tích chập theo chiều sâu (3x3) theo sau là lớp tích chập 1x1, thì Fused-MBConv thay thế/kết hợp hai lớp này bằng lớp tích chập 3x3 đơn giản.

Các lớp MBConv được hợp nhất có thể giúp đào tạo nhanh hơn chỉ với một lượng nhỏ tham số tăng lên, nhưng nếu nhiều khối trong số này được sử dụng, nó có thể làm chậm quá trình đào tạo với nhiều tham số được thêm vào. Để khắc phục vấn đề này, các tác giả đã chuyển cả MBConv và Fused-MBConv trong tìm kiếm kiến trúc neural, tự động quyết định sự kết hợp tốt nhất của các khối này để có hiệu suất và tốc độ đào tạo tốt nhất.



Việc tìm kiếm kiến trúc neural được thực hiện để cùng nhau tối ưu hóa độ chính xác, hiệu quả tham số và hiệu quả đào tạo. Mô hình EfficientNet được sử dụng làm xương sống và quá trình tìm kiếm được tiến hành với các lựa chọn thiết kế khác nhau, chẳng hạn như — khối tích chập, số lớp, kích thước bộ lọc, tỷ lệ mở rộng, v.v. Gần 1000 mô hình là mẫu và được đào tạo trong 10 epochs và kết quả của chúng được so sánh. Mô hình được tối ưu hóa tốt nhất về độ chính xác, thời gian bước đào tạo và kích thước tham số được chọn làm mô hình cơ sở cuối cùng cho EfficientNetV2.

Hình dưới cho thấy kiến trúc mô hình cơ sở của mô hình EfficientNetV2 (EfficientNetV2-S). Mô hình chứa các lớp Fused-MBConv lúc đầu nhưng sau đó chuyển sang các lớp MBConv. Để so sánh, chúng tôi cũng đã chỉ ra kiến trúc mô hình cơ sở cho bài báo EfficientNet trước đó trong Hình 9. Phiên bản trước chỉ có các lớp MBConv và không có các lớp Fused-MBConv.

EfficientNetV2-S cũng có tỷ lệ mở rộng nhỏ hơn so với EfficientNet-B0. EfficientNetV2 không sử dụng bộ lọc 5x5 và chỉ sử dụng bộ lọc 3x3.

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

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

Sau khi có được mô hình EfficientNetV2-S, nó sẽ được mở rộng quy mô để có được các mô hình EfficientNetV2-M và EfficientNetV2-L. Một phương pháp chia tỷ lệ hỗn hợp đã được sử dụng, tương tự như EfficientNet, nhưng một số thay đổi khác đã được thực hiện để làm cho các mô hình nhỏ hơn và nhanh hơn —

Đầu tiên, kích thước hình ảnh tối đa được giới hạn ở 480x480 pixel để giảm mức sử dụng bộ nhớ GPU/TPU, do đó tăng tốc độ đào tạo.

Thứ hai, nhiều lớp hơn đã được thêm vào các giai đoạn sau (giai đoạn 5 và 6 trong Hình 8), để tăng dung lượng mạng mà không làm tăng nhiều chi phí thời gian chạy.

Kích thước hình ảnh lớn hơn thường có xu hướng cho kết quả đào tạo tốt hơn nhưng tăng thời gian đào tạo. Một số bài báo trước đây đã đề xuất kích thước hình ảnh thay đổi linh hoạt, nhưng nó thường dẫn đến mất độ chính xác trong đào tạo.

Các tác giả của EfficientNetV2 cho thấy rằng khi kích thước hình ảnh được thay đổi linh hoạt trong khi đào tạo mạng, do đó, việc chuẩn hóa cũng nên được thay đổi tương ứng. Thay đổi kích thước hình ảnh, nhưng vẫn giữ nguyên chuẩn hóa dẫn đến mất độ chính

xác. Hơn nữa, các mô hình lớn hơn đòi hỏi sự chính quy hóa nhiều hơn các mô hình nhỏ hơn.

Các tác giả kiểm tra giả thuyết của họ bằng cách sử dụng các kích thước hình ảnh khác nhau và các phần mở rộng khác nhau. Như đã thấy trong dưới, khi kích thước hình ảnh nhỏ, phần mở rộng yếu hơn sẽ cho kết quả tốt hơn, nhưng khi kích thước hình ảnh lớn, phần tăng cường mạnh hơn sẽ cho kết quả tốt hơn.

	Size=128	Size=192	Size=300
RandAug magnitude=5	78.3 \pm0.16	81.2 \pm 0.06	82.5 \pm 0.05
RandAug magnitude=10	78.0 \pm 0.08	81.6 \pm0.08	82.7 \pm 0.08
RandAug magnitude=15	77.7 \pm 0.15	81.5 \pm 0.05	83.2 \pm0.09

Cân nhắc giả thuyết này, các tác giả của EfficientNetV2 đã sử dụng phương pháp Progressive Learning with Adaptive Regularization. Ý tưởng rất đơn giản. Trong các bước đầu tiên, mạng đã được đào tạo về hình ảnh nhỏ và regularization yếu. Điều này cho phép mạng học các tính năng nhanh chóng. Sau đó, kích thước hình ảnh được tăng dần và các regularizations cũng vậy. Điều này làm cho mạng khó học. Nhìn chung, phương pháp này cho độ chính xác cao hơn, tốc độ đào tạo nhanh hơn và ít trang bị thừa hơn.

Kích thước hình ảnh ban đầu và tham số chuẩn hóa do người dùng xác định. Linear interpolation sau đó được áp dụng để tăng kích thước hình ảnh và chuẩn hóa sau một giai đoạn cụ thể (M), như thể hiện trong hình dưới. Điều này được giải thích trực quan hơn trong cuối cùng. Khi số lượng epochs tăng kích thước hình ảnh và các phần mở rộng cũng tăng dần. EfficientNetV2 sử dụng ba loại regularization khác nhau — Dropout, RandAugment và Mixup.

Algorithm 1 Progressive learning with adaptive regularization.

Input: Initial image size S_0 and regularization $\{\phi_0^k\}$.

Input: Final image size S_e and regularization $\{\phi_e^k\}$.

Input: Number of total training steps N and stages M .

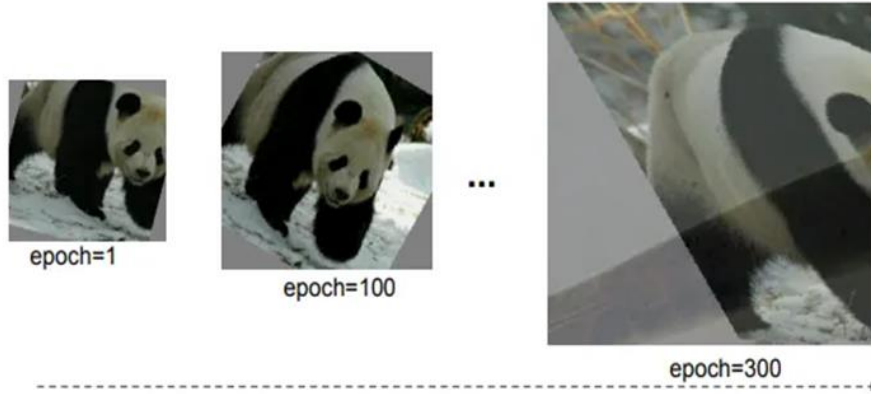
for $i = 0$ **to** $M - 1$ **do**

Image size: $S_i \leftarrow S_0 + (S_e - S_0) \cdot \frac{i}{M-1}$

Regularization: $R_i \leftarrow \{\phi_i^k = \phi_0^k + (\phi_e^k - \phi_0^k) \cdot \frac{i}{M-1}\}$

Train the model for $\frac{N}{M}$ steps with S_i and R_i .

end for



4. Decoder

4.1. Tiền xử lý caption

Những câu caption trong bộ dữ liệu *UIT-ViC* và *Flickr_sportballs* được gán nhãn thủ công nên có những lỗi về ngữ pháp, chính tả hoặc khoảng cách, sai dấu câu, ... Vì vậy, trước khi huấn luyện mô hình, chúng em đã tiến hành loại sửa những lỗi này. *Hình 2* minh họa một ảnh có câu caption bị sai.



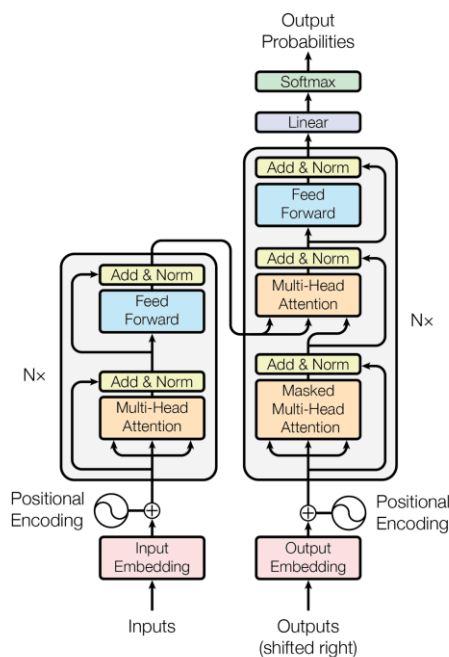
1. Các cầu thủ bóng đá **dang** di chuyển ở trên sân để tranh bóng.
2. Một nam thủ môn đang **truwowc** trên sân để bắt bóng.
3. Các cầu thủ bóng đá đang thi đấu ở trên sân. **/**
4. Một cầu thủ áo xanh đang khụy gối ở cạnh quả bóng.
5. Những cậu bé đang chơi bóng đá ở trên sân.

Hình 2 Ví dụ về lỗi ngữ pháp, dấu câu của caption trong bộ dữ liệu

Trong đồ án này, chúng em sẽ áp dụng các kỹ thuật tiền xử lý văn bản như chuyển chữ in hoa thành chữ thường (*lowercase*), loại bỏ kí tự đặc biệt ($\backslash./!?$), loại bỏ chữ số, ... để tạo ra được một bộ từ vựng đơn giản nhưng có thể mang lại hiệu quả trong quá trình tạo câu caption cho ảnh.

4.2. Decoder Transformer

Transformer là một mô hình học sâu được giới thiệu năm 2017, được dùng chủ yếu ở lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Đây được coi là mô hình mạng học sâu hiện đại và mang lại hiệu quả cao hiện nay (state-of-the-art).

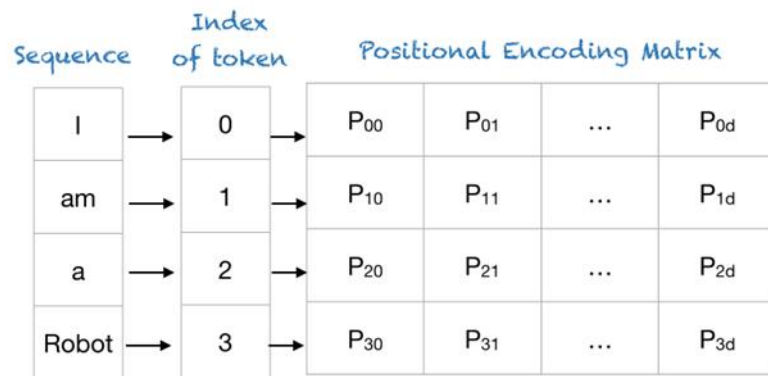


Sau khi thu được những đặc trưng ảnh dưới dạng các ma trận *image embedding* thông qua *encoder*, chúng em sẽ sử dụng kiến trúc *decoder* của mô hình mạng học sâu *Transformer* để tiến hành tạo câu caption cho ảnh.

4.2.1. Positional Encoding

Positional Encoding là một kỹ thuật được sử dụng để đưa thông tin vị trí vào input embedding của mô hình mạng nơ-ron. Mục đích là cung cấp cho mô hình thông tin về thứ tự và vị trí của các từ trong câu. Điều này rất quan trọng vì các neural networks thường hoạt động trên các vectơ có kích thước cố định và chúng không có ý nghĩa vốn có về thứ tự của các phần tử trong chuỗi đầu vào. Do đó, nếu không có *positional*

encoding, mô hình sẽ gặp khó khăn trong việc phân biệt giữa các chuỗi khác nhau chứa các từ giống nhau theo các thứ tự khác nhau.



Positional Encoding Matrix for the sequence 'I am a robot'

Công thức được sử dụng cho *positional encoder*:

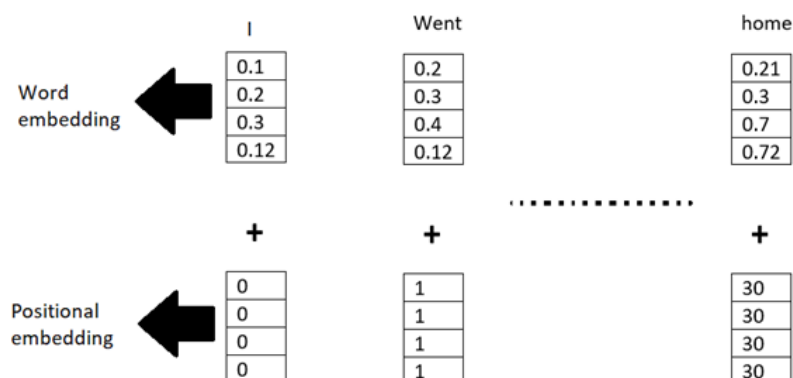
$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

Trong đó:

- pos : vị trí token trong chuỗi
- i : chỉ số của token trong chiều embedding
- d_{model} : tổng số chiều embedding

Bằng việc cộng thêm positional encoding vào input embedding sẽ giúp mô hình hiểu rõ hơn về cấu trúc và ý nghĩa của chuỗi văn bản đầu vào.

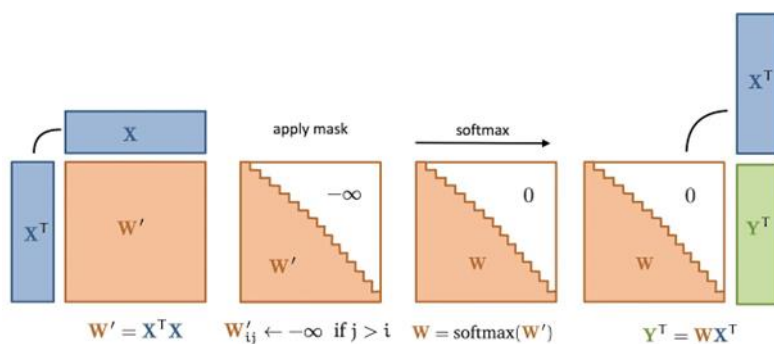


4.2.2. Masked Multi-Head Attention

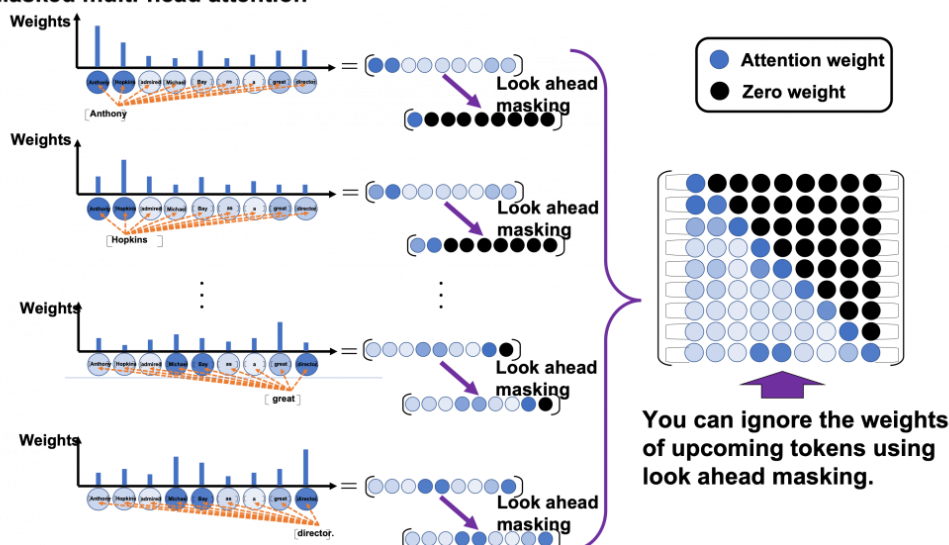
Masked Multi-Head Attention là một cơ chế *attention* được sử dụng trong kiến trúc mạng neural *sequence-to-sequence*. Mục đích là cho phép *decoder* chỉ tập trung vào các mã thông báo đã tạo trước đó thay vì toàn bộ chuỗi đầu vào. Cơ chế *Masked Multi-Head Attention* hoạt động như sau:

1. Đầu vào của cơ chế là một chuỗi các embeddings, biểu thị các mã thông báo đầu ra đã tạo ra trước đó.
2. Các embeddings được chuyển đổi bằng ba phép chiếu tuyến tính riêng biệt, tạo ra các vector truy vấn, vector chia khóa và vector giá trị cho mỗi mã thông báo. Những vector này sau đó được chia thành nhiều head khác nhau, cho phép mô hình tập trung vào các phần khác nhau của chuỗi.
3. Các điểm attention giữa các *vector query* và *key* được tính toán cho mỗi head, sử dụng cơ chế *scaled dot-product attention*. Các điểm attention này sau đó được sử dụng để trọng số hóa các vector giá trị, và các giá trị được trọng số này kết hợp để tạo ra một tập hợp các đầu ra attention.
4. Các đầu ra attention được nối lại và thông qua một phép chiếu tuyến tính khác, tạo ra đầu ra cuối cùng của cơ chế.

Trong quá trình huấn luyện, chuỗi đầu vào được che để ngăn *decoder* “nhìn thấy” các tokens kết quả chưa được tạo ra. Điều này được thực hiện bằng cách đặt các điểm attention cho các mã thông báo trong tương lai thành một giá trị âm vô cực.



Masked multi-head attention

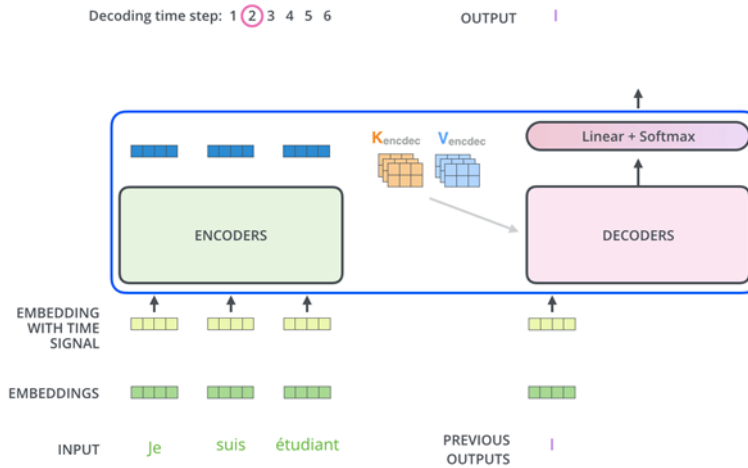


4.2.3. Multi-Head Attention

Multi-Head Attention trong *decoder Transformer* được sử dụng để tạo ra các từ trong câu mô tả ảnh. Cụ thể, *Multi-Head Attention* được sử dụng để tính toán độ quan trọng của các từ trong câu mô tả ảnh đối với các hình ảnh đã được *encoder*. Việc này giúp mô hình tập trung vào các phần khác nhau của hình ảnh để tạo ra các từ mô tả chính xác hơn trong câu caption.

Để thực hiện *Multi-Head Attention*, đầu vào của cơ chế *attention* sẽ bao gồm các vector truy vấn (*query*) và các vector giá trị (*value*). Các vector truy vấn được tạo ra bằng cách lấy đầu ra của lớp xử lý tuyến tính trước đó (*Masked Multi-Head Attention*) và truyền qua một lớp xử lý tuyến tính khác để tạo ra các vector truy vấn (*Add & Norm*). Các vector giá trị được tạo ra bằng cách lấy đầu ra của lớp *encoder* hình ảnh và truyền qua một lớp xử lý tuyến tính khác để tạo ra các vector giá trị (*image embedding*).

Sau đó, các vector query và value được chia thành nhiều head và truyền vào hàm attention. Các điểm attention giữa các vector query và value được tính toán cho mỗi head, và các giá trị được trọng số này kết hợp để tạo ra một tập hợp các đầu ra attention. Các đầu ra attention này sau đó được nối lại và thông qua một lớp xử lý tuyến tính khác để tạo ra các đại diện cho các từ trong câu mô tả ảnh.



5. Phương pháp đánh giá

5.1. BLEU (Bilingual Evaluation Understudy)

Ý tưởng chính của *BLEU* là đếm số *matching n-grams* của *candidate* (câu được mô hình sinh ra) và *reference* (là câu ground truth) hoặc *match* trên bất kỳ *reference* nào nếu như có nhiều *references*. Kết quả sẽ là số match chia cho số từ của candidate. Các match này không phụ thuộc vào vị trí, do vậy *BLEU* không sử dụng *word order*. Càng *match* nhiều tức là càng tốt. Do đó, khi đếm *matching n-grams* cần chú ý cả số lần xuất hiện của từ trong *reference*, một từ trong *reference* khi được *match* rồi thì không nên *match* nữa để tránh hiện tượng một từ *match* với *reference* nhưng được lặp lại nhiều lần trong *candidate*.

BLEU còn được dùng để đánh giá một *corpus* (tập hợp của các *sentence*, hay một đoạn văn). Đầu tiên là tính số *match* với từng câu. Cộng các số này rồi chia cho tổng số *n-gram* từ các câu là ra *modified precision score* cho *test corpus*.

$$BLEU = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')}$$

Ngoài ra, còn có thể sử dụng trọng số (weights) khác nhau cho các n-grams khác nhau để tính điểm BLEU cuối cùng. Dựa vào trọng số này ta chia ra cumulative và individual BLEU với weights là 1 tuple thể hiện trọng số tương ứng với từng i-gram score ở vị trí thứ i. Thông thường, trong các bài báo nghiên cứu, để so sánh các kiến trúc khác nhau trên một benchmark dataset, cumulative BLEU-1, BLEU-2, BLEU-3, BLEU-4 được sử dụng.

Hạn chế: Có nhiều cách để dịch (tốt) một câu. Một bản dịch tốt có thể có điểm BLEU thấp vì nó có ít n-gram trùng với ground truth. Vì vậy, điểm BLEU phụ thuộc nhiều vào chất lượng của ground truth.

5.2. ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) là một bộ các chỉ số đánh giá thường được sử dụng trong xử lý ngôn ngữ tự nhiên (NLP) và tóm tắt văn bản để đo độ tương đồng giữa bản tóm tắt được sinh ra và bản tóm tắt tham chiếu.

Các chỉ số ROUGE dựa trên recall của các n-gram (chuỗi liên tiếp các từ) trong bản tóm tắt được sinh ra so với bản tóm tắt tham chiếu.

- *ROUGE-N*: đo lường recall của các n-gram
- *ROUGE-L*: đo độ dài của chuỗi con chung dài nhất giữa bản tóm tắt được sinh ra và bản tóm tắt tham chiếu (*longest common subsequence*)
- *ROUGE-W*: cho trọng số cao hơn cho các n-gram xuất hiện sớm hơn trong bản tóm tắt.

Độ đo ROUGE được tính theo công thức:

$$ROUGE = \frac{\sum_{S \in \{Reference\}} \sum_{gram_n \in S} Count_{match}(n - gram)}{\sum_{S \in \{Reference\}} \sum_{gram_n \in S} Count(n - gram)}$$

Hạn chế: ROUGE chỉ đánh giá sự tương đồng về mặt từ vựng và cấu trúc câu giữa bản tóm tắt được sinh ra và bản tóm tắt tham chiếu. Nó không đánh giá được tính mạch lạc, đọc được hay đúng ngữ nghĩa của bản tóm tắt.

5.3. CIDEr (Consensus-based Image Description Evaluation)

CIDEr là một metric (đánh giá độ đo) được sử dụng để đánh giá chất lượng của các mô hình image captioning. *CIDEr* là viết tắt của cụm từ "*Consensus-based Image Description Evaluation*" và được ra đời bởi nhóm nghiên cứu tại Đại học Cornell.

CIDEr tính độ tương đồng giữa các captions của một ảnh với nhau và với caption được tạo bởi mô hình. *CIDEr* sử dụng một phương pháp đánh giá độ đa dạng gọi là *n-gram distribution similarity*, trong đó độ đa dạng của các câu mô tả được đánh giá bằng cách tính toán sự tương đồng giữa phân bố các từ trong câu mô tả với phân bố của các từ trong tất cả các câu mô tả.

Điểm *CIDEr* được tính bằng cách so sánh các từ trong caption được tạo bởi mô hình với các từ trong các caption đúng của ảnh. Các từ được đánh giá bằng cách tính xác suất xuất hiện của từ trong các caption đúng và đánh trọng số dựa trên độ quan trọng của từ đó trong caption. *CIDEr* có giá trị từ 0 đến 1, với giá trị càng cao thể hiện mô hình tạo ra các caption có chất lượng cao hơn.

Các bước thực hiện:

Bước 1: các word được đưa về dạng từ gốc, gọi là "stem" hay "root form". Ví dụ: "fishs", "fishing", "fished" được đưa về "fish".

Bước 2: Biểu diễn mỗi câu bằng một tập các n-grams (n thường từ 1 đến 4).

Bước 3: Tính toán tần số xuất hiện TF (*Term Frequency*) của mỗi n-gram trong tập các câu mô tả thực tế và trong tập các câu mô tả được tạo ra bởi mô hình.

Bước 4: Tính toán trọng số đóng góp của mỗi n-gram bằng cách sử dụng IDF (*Inverse Document Frequency*).

Bước 5: Tính toán điểm *CIDEr* bằng cách tính tổng trọng số đóng góp của các n-gram được tính toán ở bước trên.

Số lần một n-grams w_k xuất hiện trong một reference sentence s_{ij} được biểu thị là $h_k(s_{ij})$ và cho candidate sentence c_i là $h_k(c_i)$. Tính toán trọng số TF-IDF cho mỗi n-grams $g_k(s_{ij})$ như sau:

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{w_l \in \Omega} h_l(s_{ij})} \log \left(\frac{|I|}{\sum_{l_p \in I} \min(1, \sum_q h_k(s_{pq}))} \right)$$

Với w là tập từ vựng của tất cả các n -grams và I là tập tất cả ảnh trong bộ dữ liệu. Theo công thức trên, cụm đầu tiên (trước \log) tính TF cho mỗi n -gram w_k , cụm thứ hai tính độ hiếm của w_k sử dụng IDF . TF sẽ đánh trọng số cao hơn vào những n -grams thường xuất hiện trong reference sentence cho một ảnh trong khi IDF giảm trọng số của các n -grams xuất hiện phổ biến trong tất cả ảnh trong bộ dữ liệu. Điều này có nghĩa là, IDF cung cấp một thước đo độ quan trọng của từ bằng cách giảm trọng số các từ phổ biến có khả năng mang ít thông tin trực quan hơn. IDF được tính bằng cách sử dụng *logarit* của số lượng hình ảnh trong tập dữ liệu $|I|$ chia cho số lượng hình ảnh mà w_k xuất hiện trong bất kỳ reference sentence nào của nó.

Điểm $CIDEr$ cho n -grams với chiều dài n được tính bằng trung bình *cosine* giữa candidate sentence và reference sentences.

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \circ g^n(s_{ij})}{\|g^n(c_i)\| \|g^n(s_{ij})\|}$$

Với $g^n(c_i)$ là một vector tạo thành bởi $g_k(c_i)$ ứng với tất cả n -grams có chiều dài n và $\|g^n(c_i)\|$ là độ lớn của vector $g^n(c_i)$. Tương tự cho $g^n(s_{ij})$.

6. Thực nghiệm, kết quả và nhận xét

6.1. Cài đặt thực nghiệm

- Trong bài toán này, chúng em sẽ sử dụng pretrained model *EfficientNetV2* cho phần *encoder* và *Transformer* cho phần *decoder*.
- Tập dữ liệu chúng em sử dụng cho việc huấn luyện và đánh giá là UIT-ViIC và Flickr_sportballs.
- Để thuận tiện cho việc thực nghiệm và kiểm tra, chúng em sẽ tiến hành xây dựng trên môi trường *Google Colab* với ngôn ngữ *Python*.

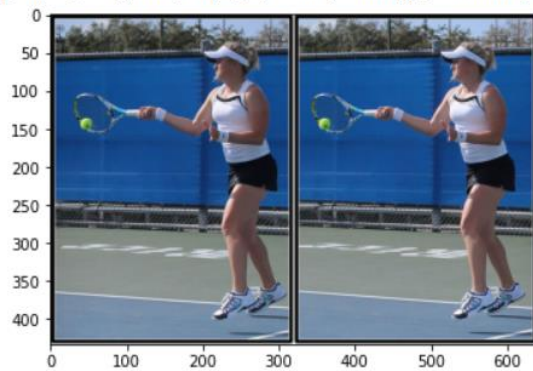
6.2. Kết quả thực nghiệm

Kết quả thu được sau khi training mô hình với bộ dữ liệu UIT-ViIC và Flickr_sportballs được thể hiện qua các độ đo đánh giá như sau:

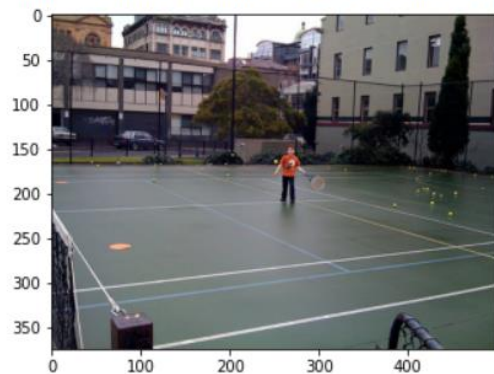
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	CIDEr
<i>UIT-ViIC + Flickr_sportballs</i>	0.6957	0.5562	0.4576	0.3863	0.5802	1.0703

Một số hình ảnh kết quả kiểm tra thực tế:

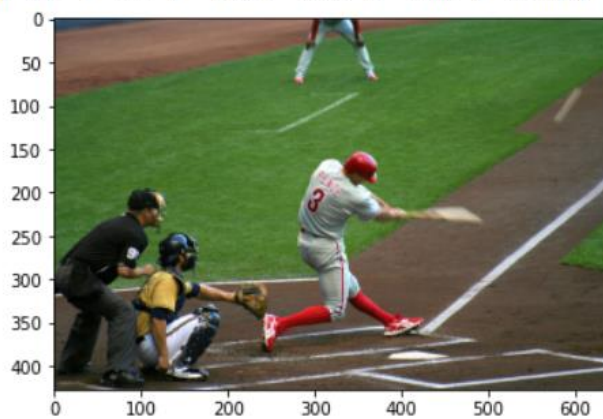
['một', 'người', 'phụ', 'nữ', 'đang', 'chuẩn', 'bị', 'đánh', 'quả', 'bóng', 'tennis', '<end>']



['một', 'người', 'đàn', 'ông', 'đang', 'vung', 'vợt', 'để', 'đánh', 'quả', 'bóng', 'tennis', '<end>']



['một', 'cầu', 'thủ', 'đánh', 'bóng', 'đang', 'vung', 'gậy', 'để', 'đánh', 'bóng', '<end>']



['các', 'cầu', 'thủ', 'bóng', 'đá', 'đang', 'thi', 'đấu', 'trên', 'sân', '<end>']



6.3. So sánh và nhận xét

Sau khi tiến hành thực nghiệm và đánh giá, chúng em sẽ thực hiện so sánh giữa mô hình nhóm xây dựng so với một số nghiên cứu trước đó. Cụ thể, nhóm chúng em sẽ so sánh với kết quả của mô hình *Pytorch-tutorial (ResNet152+LSTM)* được thực hiện trong bài báo khoa học “ *Dataset for the First Evaluation on Vietnamese Image Captioning*”.

Kết quả đánh giá với mô hình *Pytorch-tutorial (ResNet152+LSTM)*:

Dataset	Tokenizer	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr-D
English-sportball	nltk	0.761	0.562	0.405	0.289	0.560	0.668
GT-sportball	PyVI	0.596	0.455	0.341	0.254	0.522	0.578
UIT-ViIC	PyVI	0.710	0.575	0.476	0.394	0.626	1.005

Kết quả đánh giá với mô hình *EfficientNetV2 + Transformer*:

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	CIDEr
<i>UIT - ViIC</i>	0.6965	0.5647	0.4671	0.395	0.5821	1.0668
<i>UIT-ViIC + Flickr_sportballs</i>	0.6957	0.5562	0.4576	0.3863	0.5802	1.0703

Thông qua kết quả từ 2 bảng trên, chúng em rút ra được một số nhận xét:

- Xét về mức độ hiệu quả của mô hình trên cùng bộ dữ liệu *UIT-ViIC*, chúng em nhận thấy so với mô hình *Pytorch-tutorial* thì mô hình *EfficientNetV2 + Transformer* có một số vượt trội về độ đo **BLEU-4** (*Pytorch-tutorial*: 0.394, *EfficientNetV2 +*

Transformer: 0.395) và **CIDEr** (*Pytorch-tutorial*: 1.005, *EfficientNetV2 + Transformer*: 1.0668). Song vẫn còn hạn chế về các độ đo khác như **BLEU-1**, **BLEU-2**, **BLEU-3** và **ROUGE**.

- Xét về mức độ hiệu quả của mô hình *EfficientNetV2 + Transformer* trên bộ dữ liệu *UIT-ViC + Flickr_sportballs*, chúng em nhận thấy so với bộ dữ liệu chuẩn *UIT-ViC* được cung cấp thì bộ dữ liệu mà nhóm đã xây dựng thêm mang lại những kết quả đánh giá không cải thiện nhiều so với kết quả mô hình hoạt động trên bộ dữ liệu chuẩn *UIT-ViC*. Điều này có thể được giải thích bởi bộ dữ liệu mà nhóm xây dựng thêm có số lượng khá ít và chất lượng chưa tốt do việc tạo câu mô tả cho bức ảnh đều được tiến hành thủ công bởi những thành viên trong nhóm.
- Tuy nhiên, nhìn chung về độ đánh giá và một số hình ảnh thử nghiệm thì mô hình *EfficientNetV2 + Transformer* vẫn mang lại hiệu quả khá tốt trong việc *Image Captioning*. Mô hình này có nhiều tiềm năng để nghiên cứu thêm và cải tiến về mức độ hiệu quả trong tương lai.

7. Hướng phát triển và cải tiến

7.1. Về data

- Thu thập hình ảnh nhiều hơn, đa dạng hơn về các hoạt động thể thao với bóng.
- Sử dụng công nghệ trong việc hỗ trợ tạo câu caption mang tính khoa học với quy luật chặt chẽ thay vì tạo caption thủ công tốn nhiều thời gian và công sức.
- Tìm hiểu và nghiên cứu thêm về các quy tắc tạo câu mô tả bằng tiếng Việt để nâng cao chất lượng bộ dữ liệu, phù hợp hơn với ngôn ngữ thực tế.

7.2. Về model

- Tiến hành cài đặt và thử nghiệm các mô hình mạng học sâu khác như ResNet, Xception, Inception ... cho phần encoder trích xuất đặc trưng ảnh.
- Việc sử dụng pretrained model cũng ảnh hưởng đến kết quả trích xuất đặc trưng ảnh, nên cân nhắc sử dụng pretrained model.

Tài liệu tham khảo

- [1] Lam, Q.H., Le, Q.D., Nguyen, V.K., Nguyen, N.L.T. (2020). *UIT-ViIC: A Dataset for the First Evaluation on Vietnamese Image Captioning*. In: Nguyen, N.T., Hoang, B.H., Huynh, C.P., Hwang, D., Trawiński, B., Vossen, G. (eds) Computational Collective Intelligence. ICCCI 2020. Lecture Notes in Computer Science (), vol 12496. Springer, Cham. [Paper]
- [2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin. (2017). *Attention Is All You Need*. In: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.
- [3] Mingxing Tan, Quoc V. Le. *EfficientNetV2: Smaller Models and Faster Training*. Proceedings of the 38th International Conference on Machine Learning, PMLR 139:10096-10106, 2021.
- [4] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.
- [5] Lin, C.Y., 2004. *Rouge: A package for automatic evaluation of summaries*. In Text summarization branches out (pp. 74-81).
- [6] Ramakrishna Vedantam, C. Lawrence Zitnick, Devi Parikh. *CIDEr: Consensus-based Image Description Evaluation*. CoRR abs/1411.5726, 2014.
- [7] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T., 2015. *Long-term recurrent convolutional networks for visual recognition and description*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).
- [8] He, K., Zhang, X., Ren, S. and Sun, J., 2016. *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [9] Hossain, M.D., Sohel, F., Shiratuddin, M.F. and Laga, H., 2019. *A comprehensive survey of deep learning for image captioning*. ACM Computing Surveys (CSUR), 51(6), p.118.
- [10] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. *Imagenet large scale visual recognition challenge*. International journal of computer vision, 115(3), pp.211-252.