ĐÁNH GIÁ CÁC MÔ HÌNH PHÁT HIỆN ĐỐI TƯỢNG DỰA VÀO TRANSFORMER

Lê Nguyễn Thủy Nhi[1], Nguyễn Dũng[1*], Võ Văn Thành[1], Hoàng Văn Dũng[2], Lê Văn Tường Lân[3]

[1]Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế

[2]Trường Đại học Sư phạm Kỹ thuật TP. HCM

[3]Đại học Huế

*Email: nguyendung@hueuni.edu.vn

Ngày nhận bài: 8/12/2024; ngày hoàn thành phản biện: 20/12/2024; ngày duyệt đăng: 20/12/2024

TÓM TẮT

Thành công của Transformer trong xử lý ngôn ngữ tự nhiên đã thúc đẩy việc ứng dụng chúng vào thị giác máy tính, đặc biệt là trong bài toán phát hiện đối tượng. Khác với các mô hình truyền thống dựa trên CNN vốn cần bước xử lý hậu kỳ phức tạp, DETR (DEtection TRansformer) xem phát hiện đối tượng là một bài toán dự đoán tập hợp, cho phép huấn luyện đầu-cuối. Mặc dù đạt được AP0.5 từ 55.7–64.7%, DETR gặp khó khăn về tốc độ hội tụ chậm và hiệu suất kém đối với các đối tượng nhỏ (APS 15.2–23.7%). Các biến thể gần đây như Deformable DETR và RT-DETR (với AP0.5 lên đến 72.7%) đã khắc phục những hạn chế này thông qua tối ưu hóa backbone, thiết kế lại truy vấn và cải tiến cơ chế attention. Nghiên cứu này đánh giá các thành phần của DETR, so sánh các mô hình dựa trên Transformer, đồng thời đề xuất các chiến lược nhằm nâng cao hiệu quả và độ chính xác trong việc giải quyết các thách thức như phát hiện đối tượng nhỏ.

Từ khóa: Transformer, Thị giác máy tính, Phát hiện đối tượng

1. GIỚI THIỆU

Phát hiện đối tượng là một nhiệm vụ quan trọng trong thị giác máy tính, liên quan đến việc nhận diện và phân loại các đối tượng trong hình ảnh hoặc video. Nhiệm vụ này có ý nghĩa lớn vì nó là nền tảng cho nhiều ứng dụng quan trọng, chẳng hạn như xe tự hành [1-3], giám sát an ninh [4, 5], nhận diện khuôn mặt [6-9], và phân tích hình ảnh y tế [10-13]. Các hệ thống phát hiện đối tượng không chỉ xác định vị trí các đối tượng mà còn nhận diện chính xác chúng, từ đó giúp các ứng dụng hoạt động hiệu quả và chính xác hơn.

Mô hình **Transformer** [14], ban đầu được thiết kế cho xử lý ngôn ngữ tự nhiên (NLP), đã chứng minh hiệu quả vượt trội trong nhiều lĩnh vực, bao gồm cả phát hiện đối tượng. Việc áp dụng Transformer trong phát hiện đối tượng được thúc đẩy bởi cơ chế attention mạnh mẽ, cho phép mô hình tập trung vào các vùng quan trọng của hình ảnh mà không cần đến quy trình xử lý phức tạp như các mô hình truyền thống. Điều này giúp giảm sự phụ thuộc vào các bước tiền xử lý và hậu xử lý, đồng thời nâng cao hiệu suất và độ chính xác trong việc nhận diện và định vị đối tượng.

Mục tiêu của bài báo này là cung cấp một cái nhìn tổng quan toàn diện về những tiến bộ gần đây của mô hình DETR gốc [15]. Cụ thể, bài báo tập trung vào ba khía cạnh chính: (1) Khám phá và phân tích các mô-đun cơ bản của DETR cũng như các cải tiến gần đây, chẳng hạn như thay đổi kiến trúc backbone, chiến lược thiết kế query, và cải thiện cơ chế attention; (2) So sánh và đánh giá hiệu suất của các mô hình phát hiện đối tượng dựa trên Transformer, đồng thời phân tích kiến trúc mạng của chúng; (3) Đề xuất các hướng nghiên cứu trong tương lai nhằm khuyến khích các nhà nghiên cứu giải quyết những thách thức hiện tại và tiếp tục khám phá tiềm năng của Transformer trong lĩnh vực phát hiện đối tượng.

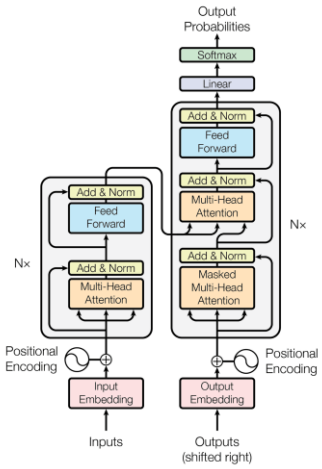
2. CƠ SỞ LÝ THUYẾT

2.1. Phát hiện đối tượng truyền thống dựa vào mạng CNN

Các phương pháp phát hiện đối tượng truyền thống dựa trên mạng nơ-ron tích chập (CNN) đã đạt nhiều thành tựu và đóng vai trò nền tảng cho lĩnh vực thị giác máy tính. Nhóm mô hình **R-CNN** [16-19] gồm R-CNN, Fast R-CNN và Faster R-CNN sử dụng phương pháp đề xuất vùng để phát hiện đối tượng. R-CNN xử lý từng vùng đề xuất một cách riêng lẻ, gây tốn thời gian. Fast R-CNN cải tiến bằng cách trích xuất đặc trưng toàn ảnh, còn Faster R-CNN sử dụng Region Proposal Network (RPN) [18] để tăng tốc và cải thiện hiệu quả, tuy nhiên vẫn còn phức tạp. Trong khi đó, **YOLO (You Only Look Once) [20]** đưa ra cách tiếp cận đơn bước: chia ảnh thành lưới và dự đoán các hộp giới hạn cùng xác suất lớp trực tiếp từ toàn ảnh. Nhờ đó, YOLO rất nhanh và phù hợp với các ứng dụng thời gian thực như giám sát và xe tự hành. Tuy nhiên, mô hình gặp khó khăn khi phát hiện các đối tượng nhỏ hoặc chồng chéo. Qua nhiều phiên bản, YOLO đã liên tục cải thiện về độ chính xác và hiệu quả. Trong khi đó, **SSD (Single Shot MultiBox Detector)** [21] cũng là một mô hình phát hiện một bước, tận dụng đặc trưng từ nhiều lớp trong mạng CNN để phát hiện ở nhiều tỉ lệ. SSD nhanh và phù hợp với các ứng dụng thời gian thực, nhưng kém hiệu quả với đối tượng nhỏ và khó điều chỉnh các tham số. Ngoài ra, mô hình **RetinaNet** [22] sử dụng kiến trúc backbone kết hợp với Feature Pyramid Network và hàm lỗi Focal Loss để cải thiện việc phát hiện các lớp mất cân bằng và đối tượng nhỏ. Mô hình này cân bằng tốt giữa tốc độ và độ chính xác, dù việc huấn luyện khá tốn tài nguyên.

2.2. Transformer và ứng dụng trong thị giác máy tính

Transformer [14] là một mô hình học sâu được giới thiệu lần đầu trong bài báo "Attention is All You Need" của Vaswani và cộng sự vào năm 2017. Transformer nhanh chóng trở thành một công cụ mạnh mẽ trong xử lý ngôn ngữ tự nhiên và đã được áp dụng thành công trong nhiều lĩnh vực khác, bao gồm cả thị giác máy tính. Hình 1 dưới đây mô tả kiến trúc của mô hình Tranformer.



Hình 1. Kiến trúc mô hình Transformer [14]

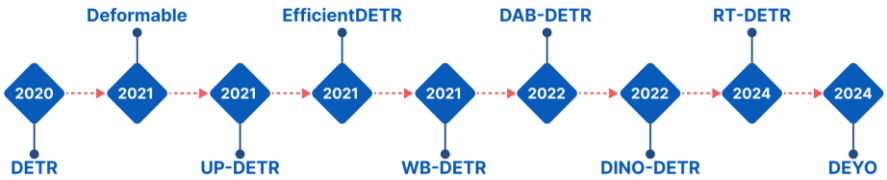
Những đặc điểm chính của Transformer: **(1) Cơ chế chú ý:** Transformer sử dụng cơ chế chú ý, đặc biệt là cơ chế chú ý nhiều đầu, để tính toán tầm quan trọng của mỗi phần tử trong dữ liệu đầu vào. Cơ chế này cho phép mô hình tự động tập trung vào các phần quan trọng của dữ liệu, giúp cải thiện hiệu suất trong việc nắm bắt mối quan hệ dài hạn và ngữ cảnh; **(2) Không cần xử lý tuần tự:** Không giống như các mô hình RNN [23-25] (Recurrent Neural Network) truyền thống, Transformer không yêu cầu xử lý tuần tự dữ liệu đầu vào. Nhờ đó Transformer có thể tận dụng khả năng xử lý song song của phần cứng hiện đại, đặc biệt là GPU và TPU, để tăng tốc độ huấn luyện và dự đoán; **(3) Kiến trúc Encoder-Decoder:** Transformer bao gồm hai phần chính: Encoder và Decoder. Encoder mã hóa thông tin từ chuỗi đầu vào thành một tập hợp các biểu diễn ngữ cảnh, trong khi Decoder sử dụng các biểu diễn này để tạo ra chuỗi đầu ra; **(4) Tính linh hoạt:** Mô hình Transformer rất linh hoạt và có thể được tùy chỉnh cho nhiều nhiệm vụ khác nhau. **(5) Khả năng mở rộng:** Transformer dễ dàng mở rộng với số lượng lớn các lớp và tham số, giúp nó xử lý được các tập dữ liệu lớn và phức tạp

Trong thị giác máy tính, Transformer đã được sử dụng để thay thế hoặc bổ sung cho các mạng CNN truyền thống, đặc biệt trong các nhiệm vụ như phân loại hình ảnh và phát hiện đối tượng. Các mô hình như Vision Transformer (ViT) [26] và Swin Transformer [27, 28] đã đạt được hiệu suất vượt trội trong phân loại hình ảnh trên các bộ dữ liệu lớn. Đặc biệt, mô hình DETR đã mang lại một cách tiếp cận mới cho nhiệm

vụ phát hiện đối tượng, tận dụng sức mạnh của Transformer để loại bỏ sự phụ thuộc vào các cấu trúc CNN truyền thống. Những thành công này đánh dấu một bước tiến lớn trong việc ứng dụng Transformer vào các bài toán thị giác máy tính, mở ra hướng đi mới cho các nghiên cứu và ứng dụng thực tiễn.

3. CÁC MÔ HÌNH PHÁT HIỆN ĐỐI TƯỢNG DỰA VÀO TRANSFORMER

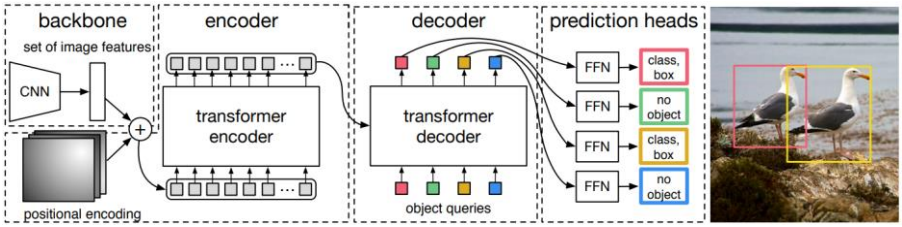
Bài báo này tập trung khảo sát và đánh giá các mô hình phát hiện đối tượng dựa vào Transformer, một lĩnh vực đang phát triển mạnh mẽ trong thị giác máy tính. Trong bối cảnh nhiều mô hình khác nhau đã được công bố, nghiên cứu này đặc biệt chú trọng đến các kiến trúc sử dụng Transformer như mô tả ở hình 2. Bằng cách đánh giá những cải tiến và kết quả đạt được của các mô hình này, bài báo nhằm cung cấp cái nhìn tổng quan và sâu sắc về hiệu quả và tiềm năng của Transformer trong nhiệm vụ phát hiện đối tượng.



Hình 2. Mốc thời gian ra đời của các mô hình dựa trên Transformer

3.1. DETR

A. Chi tiết mô hình: DETR là một mô hình phát hiện đối tượng dựa trên kiến trúc Transformer, bao gồm ba thành phần chính: Encoder, Decoder và Positional Encoding. Quá trình bắt đầu với Encoder bằng cách chia hình ảnh đầu vào thành các patch nhỏ (thường là 16 × 16 pixel) và chuyển đổi chúng thành các vector đặc trưng thông qua một mạng CNN. Để duy trì thứ tự không gian, thông tin vị trí được thêm vào các vector đặc trưng dưới dạng Positional Encoding, cho phép mô hình nhận diện được mối quan hệ không gian giữa các vùng khác nhau trong hình ảnh. Các vector đặc trưng này sau đó đi qua nhiều lớp tự chú ý (Self-attention), giúp mô hình học các phụ thuộc không gian dài hạn. Kết quả cuối cùng của Encoder được chuyển qua các lớp mạng nơ-ron truyền thẳng để tăng cường khả năng biểu diễn đặc trưng. Hình 3 mô tả kiến trúc tổng quát mô hình DETR.



Hình 3. Kiến trúc mô hình DETR

Decoder của DETR bắt đầu với một tập hợp các vector truy vấn đối tượng (object queries), đại diện cho các đối tượng tiềm năng cần được phát hiện. Các object queries này được bổ sung Positional Encoding, sau đó trải qua cơ chế cross-attention để kết nối với các vector đặc trưng từ Encoder nhằm xác định đối tượng. Các object queries này tiếp tục qua nhiều lớp tự chú ý để học các mối quan hệ giữa các đối tượng và cuối cùng được đưa qua các lớp mạng nơ-ron truyền thẳng để dự đoán các hộp giới hạn và nhãn đối tượng. Positional Encoding được dùng trong cả Encoder và Decoder, cung cấp thông tin vị trí về các patch và object queries, giúp mô hình nắm bắt thông tin không gian quan trọng.

Self-attention trong Encoder cho phép mỗi vector đặc trưng tương tác với tất cả các vector khác để tính toán trọng số chú ý, tạo điều kiện cho mô hình học được các mối quan hệ dài hạn và phụ thuộc không gian. Trong Decoder, cơ chế cross-attention kết nối các object queries với các vector đặc trưng từ Encoder để xác định chính xác các đối tượng, cho phép mô hình tập trung vào các vùng có khả năng chứa đối tượng.

Về hiệu quả, DETR có khả năng đạt độ chính xác cao ở các mức IoU (Intersection over Union) khác nhau. Với IoU trung bình, mô hình đạt *AP* từ 35.3% đến 44.9%, và ở mức IoU 0.5, độ chính xác tăng lên từ 55.7% đến 64.7%. Tuy nhiên, khi gặp các đối tượng nhỏ, hiệu quả của DETR giảm đáng kể, chỉ đạt AP^S từ 15.2% đến 23.7%, phản ánh một hạn chế khi xử lý chi tiết nhỏ trong ảnh. Về mặt tài nguyên tính toán, DETR yêu cầu *GFlop* dao động từ 187 đến 253 và số lượng tham số từ 41 triệu đến 60 triệu, chủ yếu do cơ chế tự chú ý và cross-attention phức tạp cùng với yêu cầu xử lý các lớp Positional Encoding và các lớp nơ-ron truyền thẳng ở cả Encoder và Decoder.

B. Đánh giá ưu điểm và hạn chế: DETR loại bỏ nhu cầu sử dụng các thành phần truyền thống như hộp neo và Non-Maximum Suppression (NMS), đơn giản hóa quá trình huấn luyện và dự đoán. Cơ chế tự chú ý giúp mô hình học các mối quan hệ dài hạn và phụ thuộc không gian, cải thiện độ chính xác phát hiện đối tượng. Đồng thời, DETR có tính thống nhất và linh hoạt, phù hợp cho nhiều tác vụ mà không cần thay đổi cấu trúc cơ bản. Tuy nhiên, một hạn chế lớn của DETR là yêu cầu tài nguyên tính toán cao do cơ chế tự chú ý phức tạp, dẫn đến thời gian huấn luyện lâu hơn so với các mô hình truyền thống. Hơn nữa, thiếu các thành phần đặc trưng như Feature Pyramid Network (FPN) làm giảm khả năng phát hiện đối tượng nhỏ, gây bất lợi cho các ứng dụng yêu cầu độ chính xác cao trên các chi tiết nhỏ. Tuy nhiên nhờ ưu điểm về độ chính xác cao ở các mức IoU lớn và khả năng học mối quan hệ dài hạn, DETR là lựa chọn tối ưu cho các ứng dụng đòi hỏi độ chính xác cao trên đối tượng kích thước trung bình và lớn. Tuy nhiên, hạn chế về tài nguyên và độ chính xác đối với các đối tượng nhỏ cần được cân nhắc khi áp dụng vào các hệ thống yêu cầu thời gian thực hoặc tiết kiệm tài nguyên.

3.2. Một số cải tiến tiêu biểu gần đây của DETR

Deformable DETR [29] là một cải tiến quan trọng nhằm khắc phục nhược điểm về tốc độ huấn luyện chậm và khả năng nhận diện kém đối với các đối tượng nhỏ trong DETR gốc. Thay vì sử dụng cơ chế tự chú ý toàn cục, Deformable DETR áp dụng cơ chế tự chú ý biến dạng, chỉ tập trung vào một số vị trí không gian được chọn lựa linh hoạt. Điều này không chỉ giúp giảm đáng kể chi phí tính toán mà còn tăng tốc độ hội tụ trong quá trình huấn luyện. Bên cạnh đó, khả năng thích ứng theo hình dạng và vị trí của các đối tượng giúp mô hình cải thiện hiệu suất phát hiện các đối tượng nhỏ và phức tạp. Việc sử dụng tài nguyên bộ nhớ hiệu quả hơn cũng cho phép mô hình xử lý hình ảnh có độ phân giải cao mà không gặp giới hạn bộ nhớ như trước.

DAB-DETR [30] là một biến thể của DETR, tập trung vào việc cải thiện chất lượng dự đoán thông qua việc sử dụng hộp neo động thay vì các object queries cố định. Các hộp neo trong DAB-DETR được điều chỉnh linh hoạt trong quá trình huấn luyện để phù hợp với đặc điểm của đối tượng, từ đó nâng cao độ chính xác của hộp giới hạn. Ngoài ra, mô hình còn tích hợp phương pháp mã hóa vị trí nâng cao, giúp hiểu rõ hơn về mối quan hệ không gian giữa các đối tượng. Một điểm nổi bật khác là việc sử dụng các queries động thay vì cố định, cho phép mô hình tập trung vào các vùng quan trọng hơn trong ảnh. Cuối cùng, mô-đun tinh chỉnh hộp giới hạn giúp cải thiện thêm độ chính xác trong việc xác định vị trí và kích thước đối tượng.

RT-DETR [31] được thiết kế để phục vụ các ứng dụng yêu cầu xử lý thời gian thực, với mục tiêu giảm độ trễ và tăng tốc độ dự đoán mà vẫn duy trì hiệu suất cao. Mô hình sử dụng kiến trúc Transformer nhẹ hơn, giảm số lớp và tham số để rút ngắn thời gian xử lý. Đồng thời, các kỹ thuật mã hóa vị trí patch hiệu quả được áp dụng nhằm giảm chi phí tính toán nhưng vẫn đảm bảo mô hình học được các mối quan hệ không gian. Các object queries cũng được tối ưu để rút gọn quá trình dự đoán mà không làm giảm đáng kể độ chính xác. Nhờ việc tích hợp xử lý song song và sử dụng tài nguyên phần cứng hợp lý, RT-DETR có thể vận hành tốt trên cả các thiết bị hạn chế và đáp ứng tốt yêu cầu của các ứng dụng thời gian thực.

4. SO SÁNH VÀ PHÂN TÍCH

Để có cái nhìn tổng quát về các kết quả đạt được của các mô hình, bài báo tiến hành thống kê một vài chỉ số của các mô hình như mô tả ở Bảng 1. Các mô hình này được đánh giá trên bộ dữ liệu COCO (Common Objects in Context) [32] – một tập dữ liệu tiêu chuẩn trong lĩnh vực thị giác máy tính, bao gồm hơn 200.000 hình ảnh với 80 lớp đối tượng khác nhau, hỗ trợ đánh giá hiệu quả phát hiện ở các mức độ khác nhau như đối tượng nhỏ, trung bình và lớn. Các chỉ số được sử dụng bao gồm:

(1) **GFlop** (Giga Floating Point Operations Per Second): Số lượng phép tính trong một giây.

(2) **Tham số**: Số lượng tham số của mô hình.

(3) **AP (Average precision):** Độ chính xác trung bình

(4) **$AP^{0.5}$ (Average precision at IoU threshold 0.5):** Độ chính xác trung bình với ngưỡng IoU là 0.5

(5) **AP^{S} (Average Precision with small objects):** Độ chính xác trung bình trên các đối tượng có diện tích nhỏ.

Bảng 1. So sánh hiệu năng và độ chính xác của các mô hình phát hiện đối tượng

Mô hình	Backbone	GFlop	Tham số	AP (%)	$AP^{0.5}$ (%)	AP^{S} (%)
Faster R-CNN	Resnet	180-320	42-166M	41.1-44.0	61.4-63.9	22.9-27.2
YOLOv7	CSPDarknet	104-189	36-71	51.2-52.9	69.7-71.1	35.2-36.9
YOLOv8	EfficientNet	165-257	43-68M	52.9-53.9	69.8-71.0	35.3-35.7
YOLOv10	MixNet	6.7-160.4	2.3-29.5M	38.5-54.4	53.8-71.3	18.9-37.0
DETR	Resnet	187-253	41-60M	35.3-44.9	55.7-64.7	15.2-23.7
Deformable-DETR	Resnet	173	40M	46.2	65.2	26.4-28.8
UP-DETR	Resnet	86	41M	42.8	63.0	19.0-20.8
Efficient-DETR	Resnet	159-289	32-54M	44.2-45.7	62.2-64.1	28.2-28.8
WB-DETR	None	98	24M	41.8	63.2	19.4
DAB-DETR	Resnet	216-296	44-63M	45.7-46.6	66.2-67.0	26.1-28.1
DINO	Resnet	279-860	47M	49.0-51.3	66.6-66.9	32.0-35.0
RT-DETR	Resnet	136-259	42-76M	53.1-54.3	71.3-72.7	57.7-58.6
DEYO	Resnet	8-242	4-78M	37.6-53.7	52.8-71.3	17.9-36.0

Dựa trên kết quả trong Bảng 1, có thể thấy các mô hình phát hiện đối tượng có sự khác biệt rõ rệt về hiệu suất và mức độ phức tạp tính toán. Trong đó, **RT-DETR** và **YOLOv8** nổi bật với khả năng xử lý nhanh và độ chính xác cao. RT-DETR đạt **AP0.5 72.7%** và **APS 58.6%**, cho thấy hiệu quả vượt trội trong phát hiện đối tượng nhỏ. YOLOv8 cũng cung cấp tốc độ suy luận ấn tượng, phù hợp với các ứng dụng thời gian thực yêu cầu độ chính xác cao. Ở nhóm mô hình chính xác cao nhưng tiêu tốn tài nguyên, **DINO** và **YOLOv7** là đại diện tiêu biểu với **AP** lần lượt là **51.3%** và **52.9%**. Tuy nhiên, DINO có mức **GFlops** cao nhất (279–860), nên thích hợp hơn cho các hệ thống mạnh mẽ

không yêu cầu thời gian thực. Nếu xét đến hiệu suất tính toán, **WB-DETR** và **DEYO** nổi bật nhờ số lượng tham số và GFlops thấp, phù hợp với các thiết bị nhúng hoặc hệ thống di động. Tuy nhiên, độ chính xác của chúng thấp hơn do cấu trúc đơn giản. **Efficient-DETR** và **DAB-DETR** cung cấp sự cân bằng tốt giữa độ chính xác và tốc độ xử lý, với cấu hình nhẹ hơn nhưng vẫn đảm bảo hiệu suất đáng kể, thích hợp cho các ứng dụng linh hoạt không quá khắt khe về thời gian thực.

Tổng kết, RT-DETR là lựa chọn tối ưu cho các bài toán yêu cầu cao về hiệu suất và phát hiện vật thể nhỏ. YOLOv8, Efficient-DETR và DAB-DETR mang lại sự cân bằng giữa độ chính xác và tốc độ, trong khi WB-DETR và DEYO phù hợp với môi trường hạn chế tài nguyên, còn DINO và YOLOv7 phù hợp cho các hệ thống ưu tiên độ chính xác.

5. KẾT LUẬN

Các mô hình phát hiện đối tượng dựa trên Transformer, đặc biệt là DETR và các biến thể, đã mở ra hướng tiếp cận mới trong thị giác máy tính với khả năng học End-to-End và hiệu quả cao nhờ cơ chế tự chú ý mạnh mẽ. Những cải tiến gần đây giúp khắc phục phần nào các vấn đề về tốc độ huấn luyện và khả năng phát hiện đối tượng nhỏ. Tuy vậy, các mô hình này vẫn đối mặt với thách thức về tiêu tốn tài nguyên, tốc độ suy luận và khả năng tổng quát hóa trong môi trường thực tế. Hướng phát triển tương lai sẽ tập trung vào tối ưu hóa kiến trúc để giảm chi phí tính toán, cải thiện khả năng phát hiện đối tượng nhỏ và tăng tính minh bạch, từ đó thúc đẩy Transformer trở thành giải pháp hiệu quả và linh hoạt hơn cho các bài toán phát hiện đối tượng trong thực tế.

LỜI CẢM ƠN

Cảm ơn trường Đại học Khoa học, Đại học Huế đã tài trợ cho nghiên cứu này theo mã số đề tài ĐHKH2025A-02.

TÀI LIỆU THAM KHẢO

[1] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017: IEEE, pp. 1025-1032.

[2] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, "Self-driving cars," *Computer*, vol. 50, no. 12, pp. 18-23, 2017.

[3] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars: Chances and challenges," in *Proceedings of the 1st international workshop on software engineering for AI in autonomous systems*, 2018, pp. 35-38.

[4] S. Gong, C. C. Loy, and T. Xiang, "Security and Surveillance," 2011, pp. 455-472.

[5] V.-D. Hoang and K.-H. Jo, "Accelerative Object Classification Using Cascade Structure for Vision Based Security Monitoring Systems," in *Intelligent Information and Database Systems: 8th Asian Conference, ACIIDS 2016, Da Nang, Vietnam, March 14–16, 2016, Proceedings, Part I 8*, 2016: Springer, pp. 790-800.

[6] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730-3738.

[7] W. Liu, I. Hasan, and S. Liao, "Center and scale prediction: Anchor-free approach for pedestrian and face detection," *arXiv preprint arXiv:1904.02948,* 2019.

[8] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, "Edgeface: Efficient face recognition model for edge devices," *IEEE Transactions on Biometrics, Behavior, and Identity Science,* 2024.

[9] H. O. Ikromovich and B. B. Mamatkulovich, "Facial recognition using transfer learning in the deep cnn," *Open Access Repository,* vol. 4, no. 3, pp. 502-507, 2023.

[10] T.-C. Pham, A. Doucet, C.-M. Luong, C.-T. Tran, and V.-D. Hoang, "Improving skin-disease classification based on customized loss function combined with balanced mini-batch logic and real-time image augmentation," *IEEE Access,* vol. 8, pp. 150725-150737, 2020.

[11] V.-D. Hoang, X.-T. Vo, K.-A. Phu, and K.-H. Jo, "Fusion of Segmentation and Classification for Improving Skin Disease Diagnosis," in *International Conference on Green Technology and Sustainable Development*, 2022: Springer, pp. 144-154.

[12] A. T. Huynh, V.-D. Hoang, S. Vu, T. T. Le, and H. D. Nguyen, "Skin Cancer Classification Using Different Backbones of Convolutional Neural Networks," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2022: Springer, pp. 160-172.

[13] V.-D. Hoang, X.-T. Vo, and K.-H. Jo, "Categorical weighting domination for imbalanced classification with skin cancer in intelligent healthcare systems," *IEEE Access,* 2023.

[14] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, 2020: Springer, pp. 213-229.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[17] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* vol. 28, 2015.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016: Springer, pp. 21-37.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.

[23] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[24] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[25] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU," *arXiv preprint arXiv:2305.17473*, 2023.

[26] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[27] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10012-10022.

[28] Z. Liu *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12009-12019.

[29] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[30] S. Liu *et al.*, "Dab-detr: Dynamic anchor boxes are better queries for detr," *arXiv preprint arXiv:2201.12329*, 2022.

[31] Y. Zhao *et al.*, "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16965-16974.

[32] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014: Springer, pp. 740-755.

EVALUATION OF TRANSFORMER-BASED OBJECT DETECTION MODELS

Le Nguyen Thuy Nhi[1], Nguyen Dung[1*], Vo Van Thanh[1], Hoang Van Dung[2], Le Van Tuong Lan[3]

[1]University of Sciences, Hue University

[2]HCMC University of Technology and Education, Ho Chi Minh city

[3]Hue University

*Email: nguyendung@hueuni.edu.vn

ABSTRACT

The success of Transformers in natural language processing has spurred their application in computer vision, particularly object detection. Unlike traditional CNN-based models requiring complex post-processing, DETR (Detection Transformer) treats object detection as a set prediction task, enabling end-to-end training. Despite achieving AP0.5 of 55.7–64.7%, DETR struggles with slow convergence and small object detection (APS 15.2–23.7%). Recent variants, such as Deformable DETR and RT-DETR (AP0.5 up to 72.7%), address these issues through optimized backbones, redesigned queries, and refined attention mechanisms. This study evaluates DETR's components, compares Transformer-based models, and proposes strategies to enhance efficiency and accuracy in addressing challenges like small object detection.

Keywords: Transformer, Computer Vision, Object detection.

Lê Nguyễn Thủy Nhi sinh năm 1998 tại Huế. Bà tốt nghiệp Cử nhân ngành Công nghệ thông tin năm 2020 và Thạc sĩ ngành Khoa học máy tính năm 2023 tại Trường Đại học Khoa học, Đại học Huế. Bà hiện đang là giảng viên Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Công nghệ phần mềm, học máy, học sâu, trí tuệ nhân tạo.

Nguyễn Dũng sinh ngày 13/06/1988 tại Thừa Thiên Huế. Ông tốt nghiệp cử nhân Tin học tại trường Đại học Khoa học, Đại học Huế năm 2010. Năm 2013, ông tốt nghiệp thạc sĩ chuyên ngành Khoa học máy tính tại trường Đại học Khoa học, Đại học Huế. Hiện nay, ông là giảng viên Khoa Công nghệ thông tin, trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Công nghệ phần mềm, trí tuệ nhân tạo, học máy, học sâu, cơ sở dữ liệu.

Võ Văn Thành sinh năm 2000 tại thành phố Huế. Ông tốt nghiệp Cử nhân ngành Công nghệ thông tin năm 2022 tại trường Đại học Khoa học, Đại học Huế. Ông hiện đang là trợ giảng Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế.

Lĩnh vực nghiên cứu: Học máy, học sâu, thị giác máy tính.

Hoàng Văn Dũng nhận bằng tiến sĩ ngành Kỹ thuật điện tử và Hệ thống thông tin tại Trường Đại học Ulsan, Hàn Quốc, năm 2015. Hiện nay, ông công tác tại Khoa Công nghệ thông tin, Trường ĐH Sư phạm Kỹ thuật TP. HCM.

Lĩnh vực nghiên cứu: Trí tuệ nhân tạo, Thị giác máy tính

Lê Văn Tường Lân sinh ngày 10/11/1974 tại Thừa Thiên Huế. Năm 1996, ông tốt nghiệp Đại học ngành Toán - Tin tại Trường Đại học Khoa học, Đại học Huế. Ông nhận bằng thạc sỹ Công nghệ thông tin tại Trường Đại học Bách Khoa Hà Nội năm 2002 và nhận học vị Tiến sĩ ngành Khoa học máy tính tại Trường Đại học Khoa học, Đại học Huế năm 2018. Hiện nay, ông công tác tại Ban Đào tạo và Công tác sinh viên, Đại học Huế.

Lĩnh vực nghiên cứu: Cơ sở dữ liệu, Công nghệ phần mềm, Khai phá dữ liệu.