논문 2022-17-38

객체 탐지 과업에서의 트랜스포머 기반 모델의 특장점 분석 연구

(A Survey on Vision Transformers for Object Detection Task)

하 정 민, 이 현 종, 엄 정 민, 이 재 구*
(Jungmin Ha, Hyunjong Lee, Jungmin Eom, Jaekoo Lee)

Abstract : Transformers are the most famous deep learning models that has achieved great success in natural language processing and also showed good performance on computer vision. In this survey, we categorized transformer-based models for computer vision, particularly object detection tasks and perform comprehensive comparative experiments to understand the characteristics of each model. Next, we evaluated the models subdivided into standard transformer, with key point attention, and adding attention with coordinates by performance comparison in terms of object detection accuracy and real-time performance. For performance comparison, we used two metrics: frame per second (FPS) and mean average precision (mAP). Finally, we confirmed the trends and relationships related to the detection and real-time performance of objects in several transformer models using various experiments.

Keywords : Object Detection, Transformer, Inductive Bias, Computer Vision, Deep Learning

Ⅰ. 서 론

객체 탐지 (Object Detection)혹은 검출은 그림 1과 같이 복수 객체의 위치를 찾고 이를 분류 (Classification)하는 과업이다. 과거의 객체 탐지 기법은 대부분 합성곱 신경망을 사용하여 격자형 데이터인 이미지의 특성을 반영하였다. 따라서 귀납적 편향 (Inductive Bias)이 강하다.

대표적인 합성곱 신경망 기반 객체 탐지 방식은 2-Stage와 1-Stage 방식이 있다. Faster R-CNN [1]과 같은 2-Stage 방식은 영역 제안 (Region Proposal)과 객체 분류를 따로 진행한다. 특히, 객체 탐지 과업은 실시간성이 필요한 자율주행, 지능형 공장 등에 사용되기 때문에 실시간성 또한 중요한 요소이다. 따라서 영역 제안과 객체 분류를 결합하여 한 번에 진행하는 YOLO [2]와 같은 1-Stage 방식이 제안되었다. 1-Stage 방식은 속도가 빠르지만 2-Stage 방식보다 상대적으로 정확성이 낮은 단점이 있다.

하지만, 합성곱 신경망을 사용한 방식들은 구조적으로 지역성 (Locality)에 대한 귀납적 편향이 높은 문제가 존재하였다. 이미지의 지역성처럼 귀납적 편향이란 모델을 학습시키기 위해 가정하는 데이터 특성을 의미한다. 귀납적 편향을 활용하여 이미지 분야에서 좋은 성능을 보여준 합성곱

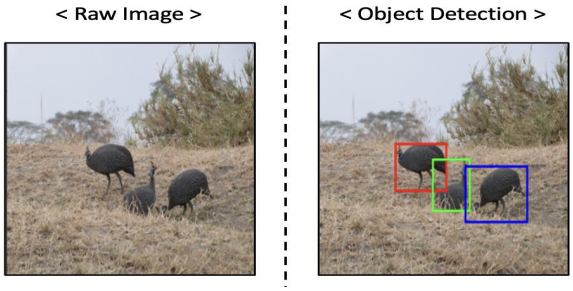

그림 1. 원본 이미지와 객체 탐지 결과
Fig. 1. Original Image and the Result of Object Detection

신경망은 구조적으로 지역성에 대한 유도 편향 (Deductive Bias)이 존재하기 때문에 강한 폐색 (Occlusion)이나 극심한 조명 조건에서 성능이 감소하는 문제가 존재한다. 이를 극복하기 위해 최근에는 귀납적 편향을 최소화하여 성능의 한계를 높인 트랜스포머 기반 모델이 대두되고 있다 [3-5].

본 논문은 객체 탐지 분야에서 트랜스포머 기반 모델의 구조와 특장점을 파악하고, 인식과 실시간 성능을 포괄적으로 비교 분석하였다.

기존 트랜스포머 모델 분류 [3]는 과업에 따른 분류 방법과 다르게 본 논문에서는 그림 2와 같이 객체 탐지 특화된 트랜스포머 모델의 구조적 특징 분류를 제시하였으며, 기본 트랜스포머 (Standard Transformer), with Key Points Attention, Adding Attention with Coordinates와 같이 구조적인 특징으로 분류하고, 각 분류별 특장점들을 분석하였다.

정량적 지표로는 객체 탐지 분야에서 객체 인식 정확도를 측정하기 위해 널리 사용되는 mAP (Mean Average Precision)를 사용하였다. 또한 각 모델의 FPS (Frame Per Sec

*Corresponding Author (jaekoo@kookmin.co.kr)
Received: Oct. 12, 2022, Revised: Nov. 16, 2022, Accepted: Nov. 21, 2022.
J. M. Ha: Kookmin University (M.S. Student)
H. J. Lee: Kookmin University (M.S. Student)
J. M. Eom: Kookmin University (M.S. Student)
J. K. Lee: Kookmin University (Assoc. Prof.)
※ 이 논문은 2022년도 정부 (과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00994, 지속가능하고 견고한 자율주행 인공지능 교육/개발 통합 플랫폼과 No.RS-2022-00167194,미션 크리티컬 시스템을 위한 신뢰 가능한 인공지능).

ⓒ IEMEK J. Embed. Sys. Appl. 2022 Dec. 17(6) 319-327
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2022.17.6.319

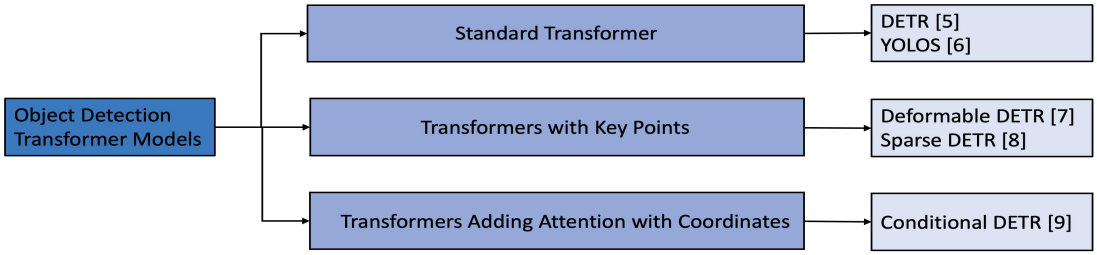

그림 2. 트랜스포머 기반 객체 탐지 모델 분류
Fig. 2. Taxonomy of Transformer based Object detection models

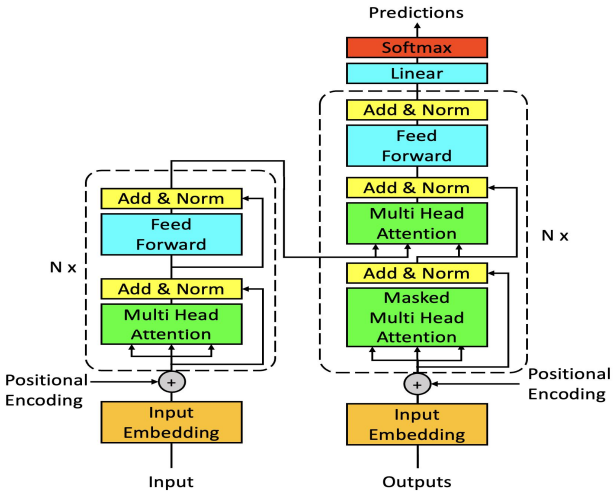

그림 3. 기본 트랜스포머의 구조
Fig. 3. Model Architecture of Basic Transformer

ond)를 통해 실시간 객체 탐지 사용 가능성을 확인했다.

본 논문은 인식 성능을 비교하기 위해 대표적인 객체 인식 공개 데이터인 MS-COCO (MicroSoft-Common Object in COntext) [10]와 Pascal VOC [11]를 사용하였다. 또한 같은 실험 환경 조건을 활용하여 정량적 성능을 측정하였다.

그 결과 대부분 모델이 탐지 정확도 측면에서 높은 성능을 보였지만, 실시간성을 보장하기 어려운 경향을 확인할 수 있었다.

Ⅱ. 모델 설명

본 논문에 사용된 모델들은 트랜스포머 기반 객체 탐지 모델이다. 일반적인 트랜스포머는 어텐션 (Attention) [12] 기반으로 모델이 확장되는 구조이다. 어텐션은 문장 전체에서 핵심 부분을 추상화하며 특징을 학습하기 위해 개발되었다.

트랜스포머는 어텐션을 사용해 전체 문장의 문맥을 잘 파악하여 자연어 처리 분야에서 좋은 성능을 내었다. 그래서 NLP (Natural Language Processing) 분야에서 최근 개발된 모델들은 대부분 트랜스포머 구조로 되어 있다 [13, 14].

NLP 분야 이외에 최근 트랜스포머 구조가 이미지 분야에서 처음으로 사용된 모델이 ViT (Vision Transformer) [4]이다. ViT는 그림 3의 트랜스포머 구조를 이미지 분류 과업에 적용해 좋은 성능을 보여준 모델이다. ViT에서는 이미지를 작은 패치들로 나누어 서열 형태로 처리하는 방법을 제시하였다. ViT는 컴퓨터 비전 분야에서 트랜스포머 구조가 사용될 수 있다는 가능성을 제시했으며 이후 개발된 비전 트랜스포머 모델들에게 큰 영향을 주었다 [3, 5, 6].

본 논문에서 사용한 모델들은 기본 트랜스포머를 사용하는 Standard Transformers, 여러 해상도의 특징맵 (Feature Map)에서 특징 핵심점들 (Key Points)만 어텐션을 적용한 Transformers with Key Points Attention, 객체의 위치를 임베딩하여 학습에 사용하는 Transformer Adding Attention with Coordinates로 분류하였다.

1. 기본 트랜스포머 (Standard Transformers)

기본 트랜스포머는 인코더만 사용하는 모델과 인코더와 디코더 모두를 사용하는 모델이 있다. 두 방법 모두 ViT의 기본 인코더에 기반하고 있어 ViT처럼 이미지 전체에 대한 문맥을 잘 파악할 수 있다는 장점이 있다.

1.1. DETR (DEtection TRansformer)[1] [5]

DETR은 자연어 처리 분야의 트랜스포머 구조를 객체 탐지 과업에 특화하여 기존의 합성곱 모델들보다 좋은 성능을 보여준 대표적인 모델이다. 또한, 객체 탐지 과업에서 합성곱 신경망이 주를 이룬 패러다임 (Paradigm)을 바꾼 초기 모델로 이후 DETR로부터 파생된 Deformable DETR [7], Sparse DETR [8], Conditional DETR [9]과 같은 객체 탐지 모델들이 연구 개발되었다.

DETR의 큰 특징은 그림 4에서 보이듯 객체 쿼리들 (Object Queries)로 객체를 탐지하는 것이다. 하나의 객체 쿼리에 하나의 객체가 이분 매칭 (Bipartite Matching)되어 학습하게 된다. 이미지 추론 시에는 하나의 객체 쿼리가 하나의 객체를 탐지하게 된다. 이분 매칭 기법을 통해 기존에 합성곱 신경망 모델에서 사용되었던 Anchor Box와 NMS (Non Maximum Suppression)를 대체하여 연산량을 줄였다.

그리고 별도의 '객체 없음' 클래스를 생성하여 객체 쿼리의 개수를 충분히 크게 설정한다. 하지만 이런 방식은 객체 쿼리가 '객체 없음' 클래스에 연결되는 비율을 증가시켜 클래스 불균형을 발생시킨다. 따라서 객체 쿼리가 '객체 없음'에 연결되면 역전파되는 값에 0.1을 곱하는 방법으로 클래스 불균형을 해소한다.

1) https://github.com/facebookresearch/detr

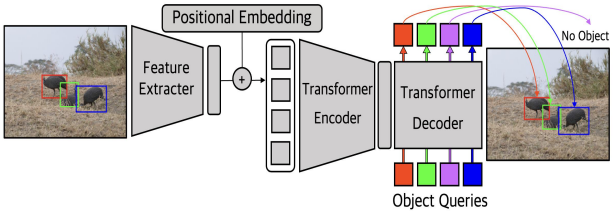

그림 4. DETR의 구조
Fig. 4. Overview of DETR

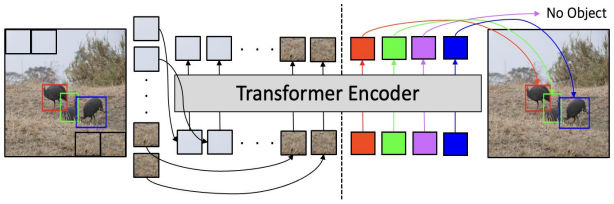

그림 5. YOLOS의 구조
Fig. 5. Overview of YOLOS

1.2. YOLOS (You Only Look One Sequence)[2] [6]

트랜스포머는 자연어 처리 분야에서 지식 전달을 목적으로 개발되었다. 자연어 처리 분야에서는 방대한 말뭉치에서 사전학습 (Pretraining)한 트랜스포머 모델을 사용하고자 하는 과업에 맞춰 미세조정 (fine-tuning)을 수행한다. 그리고 이미지 분류 분야에서도 방대한 이미지 데이터를 통해 학습한 ViT를 과업 전이하여 사용한다.

그러나 객체 탐지 과업은 이미지 분류보다 어려운 과업이며, 트랜스포머 기반의 객체 탐지 모델인 DETR은 ViT와 다른 구조를 갖는다. 따라서 ViT의 구조를 그대로 가져오며 ViT가 학습한 지식을 이용해서 객체 탐지를 할 수 있는지 가능성을 확인하기 위해 YOLOS가 제안되었다.

YOLOS는 ViT 구조를 객체 탐지 과업으로 확장한 모델이다. 따라서 그림 5에서 보이는 것과 같이 ViT에서 사용한 트랜스포머 인코더를 사용했다. ViT에서 분류 예측을 위해 사용한 [CLS] 토큰을 없애고 DETR의 객체 쿼리의 역할을 하는 [DET] 토큰을 생성하여 트랜스포머 인코더의 입력으로 넣어준다. 트랜스포머 인코더를 통과한 [DET] 토큰들을 각자 하나의 객체에 연결해주어 학습을 진행한다. 그 결과 ViT의 지식을 활용하여 성공적으로 객체 탐지를 할 수 있다는 것을 확인하였다.

2. Transformers with Key Points Attention

Key Points Attention은 어텐션을 계산할 때 여러 해상도의 특징맵에서 몇 개의 핵심 특징점들로 어텐션 연산을 하는 것이다. 이런 방법은 초기 어텐션 가중치가 고르게 분포하는 것을 방지하여 학습 수렴 속도가 향상하는 장점이 있다.

2.1. Deformable DETR[3]

DETR의 단점은 처음에 어텐션 가중치들 (Attention Weights)이 모든 픽셀의 평균값을 갖고 진행되기 때문에 학습의 수렴 속도가 느리다는 것이다. 또한 단일 해상도 특징맵을 사용하기 때문에 작은 물체에 대한 탐지 성능이 좋지 않다. Deformable DETR은 이런 문제를 해결하기 위해 Multi Scale Deformable Attention을 제안했다.

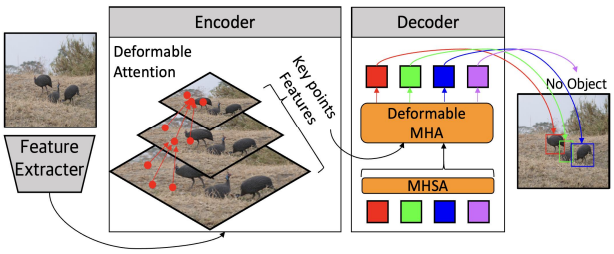

그림 6. Deformable DETR의 구조
Fig. 6. Overview of Deformable DETR

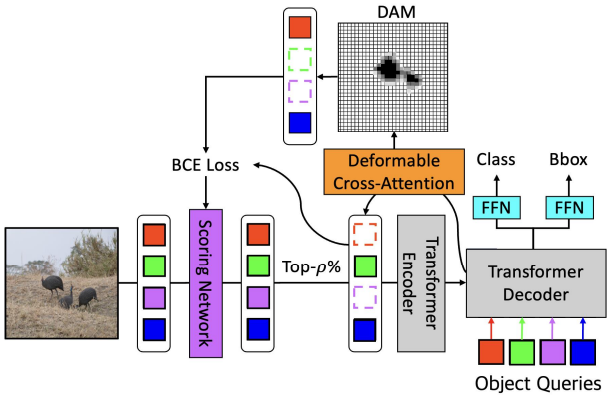

그림 7. Sparse DETR의 구조
Fig. 7. Overview of Sparse DETR

Multi Scale Deformable Attention은 백본망을 통해 여러 해상도의 특징맵을 추출한다. 추출된 특징맵들에서 몇 개의 핵심점들을 선택한다. 그 후 그림 6과 같이 선택된 핵심점들을 사용해서 어텐션 연산을 수행한다.

여러 해상도에서 어텐션을 연산하기 때문에 연산량이 증가하는 단점이 있다. 그러나 학습 초기에 어텐션 가중치 값이 고르게 분포하는 것을 방지해 학습 수렴 속도를 높였다. 또한 여러 해상도의 특징맵에서 객체에 대한 정보를 활용해 작은 물체에 대한 탐지 능력이 향상되는 경향을 보였다.

2.2. Sparse DETR[4]

기존의 DETR 계열의 객체 탐지 모델들은 객체 쿼리를 사용하여 객체를 탐지했다. 그러나 이 과정에서 각 객체 쿼리들이 객체 탐지를 위해 더 비중 있게 참조하는 패치 토큰이 존재함을 발견했다. 또한 객체 쿼리가 참조하지 않는 패치 토큰을 제외하고 학습시켰을 때 전체 패치 토큰을 사용하여 학습했을 때와 성능 차이가 나지 않음을 확인하였다.

Sparse DETR에서는 인코더의 패치 토큰 중 어느 토큰

2) https://github.com/hustvl/YOLOS
3) https://github.com/fundamentalvision/Deformable-DETR
4) https://github.com/kakaobrain/sparse-detr

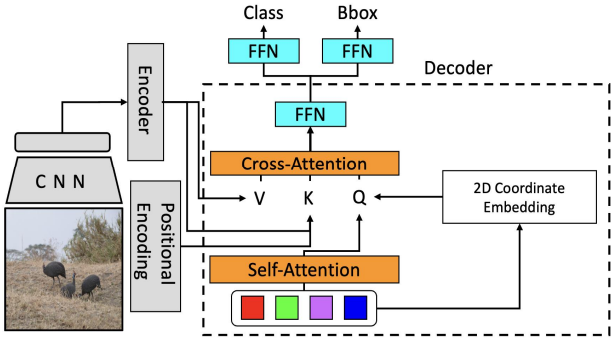

그림 8. Conditional DETR의 구조
Fig. 8. Overview of Conditional DETR

을 디코더에서 참조하여 학습에 사용할지 판단하는 Scoring Network를 사용하였다. 패치 토큰에 대한 참조 점수를 구하고 전체 패치 중 비율을 통해 참조 점수가 높은 상위 토큰들을 결정한 후 선택된 토큰만 사용하여 탐지하는 방법을 제안했다. 또한 학습 시 그림 7과 같이 만들어지는 어텐션 맵들을 더해서 학습에 영향을 주는 패치들을 나타내는 DAM (Decoder cross-Attention Map)을 생성한다. 이후 Scoring Network에서 생성한 참조 점수들과 BCE (Binary Cross Entropy) 손실함수를 사용해 Scoring Network를 학습한다.

그 결과 일부 패치만을 사용하기 때문에 기본 DETR보다 10배 빠른 학습 수렴 속도와 성능 향상을 보여주었다.

3. Transformers Adding Attention with Coordinates

Add Coordinates는 객체의 좌표를 임베딩하여 어텐션의 입력에 더해주는 것이다. 좌표 정보를 더해주면 정확한 바운딩 박스 영역을 예측하여 객체를 탐지하며 학습 수렴 속도가 향상하는 장점이 있다.

3.1. Conditional DETR[5)]

DETR은 학습이 느리게 수렴하는 문제점이 있다. 이는 처음 어텐션 맵의 값이 모두 비슷하게 계산되어 특정 부분에 집중하도록 학습되기까지 오래 걸리기 때문이다. 따라서 Deformable DETR에서는 어텐션 맵을 계산할 때 특정 핵심 점들을 이용했고 Sparse DETR은 참조하는 패치를 선택적으로 결정했다. 그러나 Conditional DETR은 객체가 존재할 수 있는 Anchor Points를 이용하는 방법을 사용했다.

객체 좌표를 전달하기 위해 그림 8과 같은 구조를 사용했다. 먼저, 객체 쿼리를 통해 학습 가능한 Anchor Points를 생성한다. 또한 이전 디코더를 통해 생성된 출력값을 완전연결층을 이용하여 임베딩한 후 Anchor Points와 곱해준다. 이렇게 생성된 좌표 임베딩 값은 트랜스포머에서 Cross Attention을 계산할 때 입력에 더해준다.객체의 좌표값을 전달하는 방법을 통해 기존의 DETR보다 학습 수렴 속도를 약 6.7배 빠르게 향상했다. 또한 객체의 위치에 대한 정보를 제공해 탐지 성능이 향상했다.

Ⅲ. 데이터 집합

1. MS-COCO

MS-COCO는 객체 탐지 과업에서 널리 사용되는 벤치마크 (Benchmark) 데이터집합이다. 총 80개의 분류가 있으며, 사람, 버스, 자전거와 같은 일상에서 흔히 볼 수 있는 객체들을 포함하고 있다.

표 1과 같이 MS-COCO의 학습 이미지는 118,287개이며 검증 이미지와 테스트 이미지는 각각 5,000개, 40,670개이다.

2. Pascal VOC

Pascal VOC는 객체 탐지 과업의 벤치마크 데이터 집합이며 20개의 분류가 있다. 데이터는 2007년 제공된 2501개의 학습 이미지, 2510개의 검증 이미지, 4952개의 테스트 이미지와 2012년에 제공된 5717개의 학습 이미지, 5823개의 검증 이미지, 10,991개의 테스트 이미지가 있다.

본 논문에서는 2007년과 2012년의 학습 이미지를 합쳐 학습을 진행했다. 또한 테스트 이미지에 대한 주석이 없기 때문에 2007년과 2012년 검증 이미지를 통해 검증을 진행했다.

Ⅳ. 평가 지표

1. mAP (mean Average Precision)

본 논문에서는 모델의 평가 지표로 객체 탐지 과업에서 널리 사용되는 mAP를 사용하였다. mAP는 각 클래스의 AP 값들의 평균을 낸 값을 의미한다. 모델의 성능을 평가할 때 정밀도 (Precision)와 재현율 (Recall)을 이용한다. 한 클래스의 정밀도는 $\frac{TP}{TP+FP}$, 재현율은 $\frac{TP}{TP+FN}$를 통해 구한다.

객체를 탐지할 때 검출이 되었으면 Positive, 검출되지 않았으면 Negative로 구분한다. 또한 정답값과 검출 결과와

표 1. 실험에 사용된 데이터집합의 상세 정보
Table 1. Details of used Datasets in Experiments

Dataset	Classes	Train Data		Validation Data		Test Data
		Images	Instances	Images	Instances	Images
MS-COCO	80	118,287	860,001	5,000	36,781	40,670
Pascal VOC	20	8,218	23,618	8,333	23,605	15,943

5) https://github.com/Atten4Vis/ConditionalDETR

표 2. COCO 검증 데이터집합에서 실험 결과
Table 2. Experimental Results on COCO Dataset

Model \ Metric	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l	FPS
DETR	0.421	0.623	0.442	0.214	0.460	0.611	29.1
YOLOS-Tiny	0.289	0.474	0.293	0.100	0.296	0.460	124.8
YOLOS-Small	0.361	0.565	0.371	0.154	0.386	0.562	11.9
YOLOS-Small (dwr)	0.375	0.575	0.391	0.158	0.401	0.573	13.5
YOLOS-Base	0.420	0.622	0.445	0.195	0.453	0.622	4.96
Deformable DETR (Single Scale)	0.394	0.597	0.422	0.207	0.429	0.558	27.0
Deformable DETR (Single Scale dc5)	0.414	0.618	0.449	0.237	0.453	0.560	23.1
Deformable DETR (base)	0.445	0.635	0.487	0.268	0.477	0.595	18.5
Deformable DETR (Refine)	0.463	0.650	0.501	0.285	0.492	0.615	18.0
Deformable DETR (Refine Two-Stage)	0.469	0.657	0.511	0.296	0.503	0.616	17.2
Sparse DETR (Swin Transformer ρ=10%)	0.483	0.693	0.525	0.300	0.514	0.642	18.0
Sparse DETR (Swin Transformer ρ=20%)	0.488	0.693	0.530	0.308	0.519	0.646	17.6
Sparse DETR (Swin Transformer ρ=30%)	**0.492**	**0.695**	**0.535**	**0.313**	**0.526**	**0.651**	17.0
Sparse DETR (RetinaNet ρ=10%)	0.455	0.659	0.464	0.286	0.482	0.601	18.2
Sparse DETR (RetinaNet ρ=20%)	0.457	0.658	0.496	0.287	0.486	0.604	17.7
Sparse DETR (RetinaNet ρ=30%)	0.461	0.661	0.498	0.291	0.492	0.608	17.3
Conditional DETR	0.409	0.619	0.434	0.207	0.442	0.595	26.0
Conditional DETR (dc5)	0.437	0.644	0.467	0.238	0.476	0.602	12.9
Average	0.430	0.630	0.460	0.241	0.460	0.600	24.0

같으면 True, 다르다면 False로 구분한다. 모델이 객체를 제대로 검출하면 TP (True Positive), 모델이 객체를 오검출했을 때 FP (False Positive), 모델이 검출했어야 하는 객체를 검출하지 못했을 때 FN (False Negative)이라고 한다.

재현율과 정밀도는 일반적으로 반비례 관계를 갖는다. 그래서 모델의 전체적인 성능 분석을 위해 x축을 재현율, y축을 정밀도로 설정한 PR (Precision-Recall) 곡선을 사용한다.

2. FPS (Frame Per Second)

FPS는 초당 프레임 (Frame)을 얼마나 처리할 수 있는지 나타내는 지표로 모델의 추론 시간을 정량적으로 나타낼 때 사용하는 평가 지표이다. 객체 탐지는 실시간에서 사용되는 경우가 많으므로 모델의 정확성과 더불어 중요한 지표이다.

V. 실험 결과

실험에서 사용한 데이터 집합은 COCO 데이터 집합과 Pascal VOC 데이터 집합이며 GPU는 NVIDIA RTX A6000 1개를 사용하여 성능을 측정하였다. 표 2의 결과를 보면 모델의 용량이 커짐에 따라 객체의 특징을 더 잘 학습하기 때문에 mAP가 증가했다. 그러나 모델의 용량이 증가하면 연산량 또한 증가하기 때문에 FPS는 낮아짐을 확인하였다.

각 모델은 성능이 수렴할 때까지 학습을 진행하였기 때문에 모델마다 학습한 에폭 (epoch)이 다르다. 또한 백본망을 사용하는 모델들은 공통적으로 ResNet50 [15]을 사용했다.

이를 통해 mAP와 FPS가 상충관계에 있다는 것을 알 수 있다. 모델의 용량이 커지면 모델의 일반화 성능이 좋아지지만 연산량이 증가하여 FPS는 감소하기 때문이다. 이를 통해 트랜스포머 구조를 사용한 객체 탐지 모델들의 mAP와 FPS 성능 사이에 상충 관계가 있음을 알 수 있다. 따라서 효율적인 성능을 찾기 위해 최적의 mAP와 FPS를 갖는 모델의 용량을 찾는 것이 중요하다.

1. COCO 데이터 집합

DETR은 DETR 계열 모델의 기본이 되는 모델이기 때문에 성능은 표 2의 평균에 비해 높지 않은 것을 볼 수 있다. 그러나 성능을 높이기 위해 추가적인 모듈 (Module)을 더한 다른 모델들과 비교했을 때 FPS가 높게 측정된 것을 확인할 수 있었다.

표 2의 YOLOS 모델들 중 Tiny 모델의 FPS가 124.8로 가장 높게 측정되었다. 이는 YOLOS-Tiny가 ViT Tiny모델에 [DET]토큰만 추가한 모델이기 때문에 연산량이 크지 않아 나타난 결과이다. 그러나 Small, Smal l(dwr), Base 모델은 연산량이 많이 늘어나기 때문에 FPS가 급격히 감소하는 것을 확인할 수 있다.

또한 Small 모델에는 Depth, Width, Resolution을 조절하는 최적화 기법을 적용했다 [16]. 그 결과 Small (dwr) 모델의 성능이 Small 모델보다 mAP가 0.014 향상되었다.

Deformable DETR에서는 Deformable Attention을 사용해 기본 DETR보다 성능을 향상했다. 이를 통해 다양한 해상도를 사용하는 것이 성능 향상에 중요한 요소라는 것을 확인할

표 3. Pascal VOC 검증 데이터집합에서 실험 결과
Table 3. Experimental Results on Pascal VOC Dataset

Model \ Metric	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l	FPS
DETR	0.470	0.725	0.496	0.058	0.315	0.621	30.3
YOLOS-Tiny	0.260	0.460	0.259	0.007	0.075	0.387	**141.2**
YOLOS-Small	0.383	0.608	0.396	0.027	0.165	0.531	12.0
YOLOS-Small (dwr)	0.385	0.608	0.398	0.033	0.179	0.532	18.9
YOLOS-Base	0.424	0.662	0.445	0.049	0.220	0.585	5.7
Deformable DETR (Single Scale)	0.558	0.802	0.614	0.192	0.438	0.668	29.0
Deformable DETR (Single Scale dc5)	0.569	0.810	0.629	0.225	0.452	0.671	14.8
Deformable DETR (base)	0.586	0.814	0.653	0.228	0.469	0.693	18.7
Deformable DETR (Refine)	0.603	0.824	0.665	0.240	0.474	0.707	24.4
Deformable DETR (Refine Two-Stage)	**0.605**	0.830	**0.669**	0.252	**0.482**	**0.713**	22.8
Sparse DETR (Swin Transformer ρ=10%)	0.520	0.788	0.562	0.154	0.377	0.629	20.5
Sparse DETR (Swin Transformer ρ=20%)	0.529	0.794	0.570	0.159	0.377	0.637	20.5
Sparse DETR (Swin Transformer ρ=30%)	0.533	0.796	0.578	0.173	0.384	0.638	19.7
Sparse DETR (RetinaNet ρ=10%)	0.587	0.825	0.647	0.256	0.476	0.691	23.4
Sparse DETR (RetinaNet ρ=20%)	0.589	0.828	0.655	0.250	0.474	0.693	23.1
Sparse DETR (RetinaNet ρ=30%)	0.590	**0.831**	0.648	**0.257**	0.481	0.697	22.8
Conditional DETR	0.562	0.795	0.617	0.154	0.409	0.682	28.0
Conditional DETR (dc5)	0.579	0.802	0.634	0.186	0.426	0.692	14.9
Average	0.540	0.756	0.563	0.161	0.370	0.637	27.3

수 있다. 그러나 여러 해상도에서 어텐션을 진행하기 때문에 기본 어텐션보다 연산량이 증가하여 FPS가 낮게 측정된다.

Deformable DETR은 이전 디코더 층에서 예측한 바운딩 박스를 앵커박스 (Anchor Box)로 사용하는 Iterative Refinement 방법을 사용하여 성능을 향상했다. 또한 인코더에서 영역 제안을 사용하는 Two-Stage Deformable DETR을 통해 성능을 향상했다.

Sparse DETR은 임베딩된 패치 중 객체 쿼리에서 주로 참조하는 이미지 패치만을 사용해 Deformable DETR에서 성능을 향상했다. 그 중 Swin Transformer [17]를 백본망으로 사용하고 상위 30%의 패치만 참조하는 모델이 mAP 0.492로 전체 모델 중 가장 좋은 성능을 보였다. 또한 백본망으로 Swin Transformer를 사용하는 모델들이 백본망으로 RetinalNet을 사용하는 모델들보다 좋은 성능을 보였다. 이는 Swin Transformer가 RetinaNet [18]보다 이미지의 특징을 추출하는 백본망의 역할을 잘 수행하는 것을 의미한다. 또한 이미지의 패치 중 일부 패치만을 참조하여 Deformable DETR에 비해 연산량을 감소시켰다.

표 2에서 전체적으로 참조하는 패치의 개수가 많아질수록 FPS가 감소하는 경향을 보였다. 이는 사용하는 패치의 개수가 많아지는 만큼 연산량이 증가하기 때문에 나타나는 결과이다.

Conditional DETR은 객체의 좌표에 대한 정보를 제공하여 기본 DETR보다 성능이 향상되었다. 그러나 좌표에 대한 정보만 추가로 제공해주기 때문에 FPS는 기본 DETR과 비슷하게 측정되었다. 또한, IoU (Intersection over Union)가 0.5 이상, 0.75 이상인 객체에 대한 성능을 측정하는 mAP_50과 mAP_75의 성능도 차이가 없었다.

2. Pascal VOC 데이터 집합

표 3의 결과를 보면 대체로 성능이 높게 나온 것을 볼 수 있으며 FPS또한 높게 측정된 것을 확인할 수 있다.

Pascal VOC는 한 이미지 안에 많은 객체가 존재하지 않기 때문에 COCO 데이터 집합과 비교했을 때 평균적으로 더 높은 성능을 보여준다. 또한 이미지의 크기가 비교적 작기 때문에 이미지 크기에 영향을 받는 연산량이 감소하여 FPS가 높게 측정되었다.

또한 표 3의 작은 객체에 대한 성능을 측정하는 mAP_s를 보면 단일 해상도의 특징 맵만을 사용하는 모델이 대체로 성능이 낮은 것을 볼 수 있다.

그리고 특징으로 볼 수 있는 것은 트랜스포머의 인코더만 사용하는 YOLOS는 COCO 데이터 집합에서의 성능과 비슷하거나 더 낮은 성능을 보였다. 이는 Pascal VOC 이미지의 크기가 작아서 패치를 나눌 때 동일한 개수로 나누면 하나의 패치가 이미지에서 차지하는 영역이 넓어지게 된다. 따라서 이미지에 대한 세밀한 예측 (Dense Prediction)이 어려워진다.

또한 YOLOS는 [DET]토큰과 함께 Self Attention연산이 이뤄지기 때문에 이미지 내부에서 객체를 분리해주지 못하게 된다. 그러나 인코더-디코더의 형태로 사용되는 모델은 인코더에서 추출한 이미지 전체 정보와 객체 쿼리를 통해 각 객

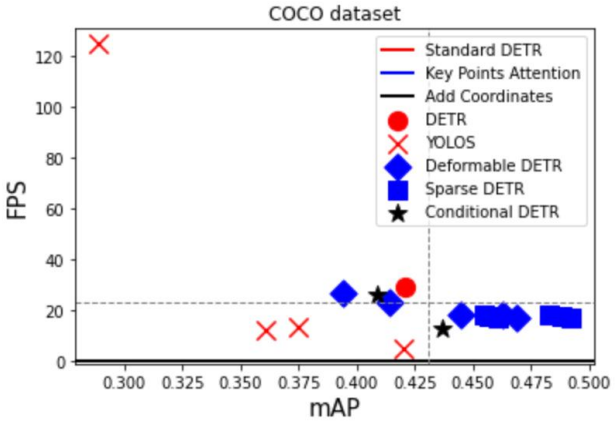

그림 9. 모델별 mAP, FPS의 관계 산점도
Fig. 9. mAP and FPS of various Vision Transformer Models for Object Detection

체 사이를 분리시켜주는 역할을 한다. 따라서 인코더만 사용하는 모델보다 세밀한 예측에서 좋은 성능을 보여준다.

표 3의 결과에서 Deformable DETR이 대체로 가장 높은 성능을 보여주었다. 이는 이미지의 크기가 작기 때문에 여러 해상도의 특징 맵을 활용하며 이미지 패치들을 모두 사용하기 때문이다. 반면 SparseDETR은 이미지의 패치 중 중요한 Top-ρ%를 사용한다. 그러나 이미지 크기가 작은 Pascal VOC 데이터 집합에서는 역설적으로 정보의 손실이 발생한다. 따라서 Deformable DETR이 가장 높은 성능을 보여준다.

Ⅵ. 논의 및 결론

본 논문에서는 트랜스포머 기반의 객체 탐지 모델들을 특징에 따라 분류하고 같은 데이터를 사용하여 mAP와 FPS를 측정하였다. 이를 통해 모델의 구조와 실시간 탐지 가능성의 관계에 대해 분석했다.

그림 9의 분포는 mAP와 FPS를 축으로 하는 모델들의 산점도이다. 그림 9의 점선은 각 축에서 실험에 사용된 전체 모델의 평균을 의미한다. 그 결과 mAP와 FPS가 모두 평균 이상인 모델은 존재하지 않았다. 또한 DETR은 mAP, FPS 모두 평균에 가깝게 측정되었다.

기본 DETR에서 큰 변형을 주지 않고 객체의 좌표에 대한 정보만을 더해준 Conditional DETR도 평균에 가깝게 측정되었다. 또한 대부분 모델이 그래프의 오른쪽 아래에 있는 것을 볼 수 있다. 이는 기존의 DETR의 성능 향상을 위해 연산량을 고려하지 않았기 때문에 발생한 결과이다.

최근에는 모델의 성능 향상을 위해 여러 해상도를 사용하거나 핵심점들로 어텐션을 계산하는 방법을 사용한다. 또한 YOLOS의 dwr 최적화 기법, Deformable DETR의 Iterative Refinement와 Two-Stage 기법을 통해 성능 향상을 이루었다. 그 중 핵심점들로 어텐션을 사용한 모델들이 동일한 FPS에서 높은 mAP를 달성하는 경향을 확인했다.

그림 10은 GPU의 성능을 측정하기 위해 사용되는 GFLOPS (Giga FLoating point Operations Per Second)를 통해 각 모델들의 연산량을 측정한 결과이다. 표 2와 표 3에서 가장 낮은 FPS를 갖는 YOLOS-base가 가장 높은 연산량을 갖고 FPS가 가장 낮은 YOLOS-tiny가 가장 낮은 연산량을 갖는 것을 볼 수 있다. 그러나 반대로 DETR과 Deformable DETR Single Scale을 비교했을 때는 연산량이 높은 모델이 FPS가 더 높게 나온 것을 확인할 수 있다. Deformable DETRSingle Scale은 DETR에 Deformable Attention Module만 더해준 것이다. 따라서 연산량 자체는 낮게 나오지만 FPS가 느린 원인이 Deformable Attention Module의 최적화가 잘 이루어져있지 않아 발생하는 문제로 볼 수 있다. 위와 같은 결과를 통해 GFLOPS와 FPS 사이에 대략적인 상충관계를 갖지만 부가적인 모듈로 인한 FPS저하가 있을 수 있다는 것을 알 수 있다.

그림 9를 통해 대체로 트랜스포머 구조의 모델들이 mAP와 FPS사이에 상충관계가 있는 것을 확인했다. 그러나 정량적인 지표를 통해 어떤 상관관계가 있는지 확인하기 위해 FPS와 mAP사이의 피어슨 (Pearson) 상관계수를 계산하였다. 또한 연산량과 모델의 FPS와의 연관성을 확인하기 위해 m

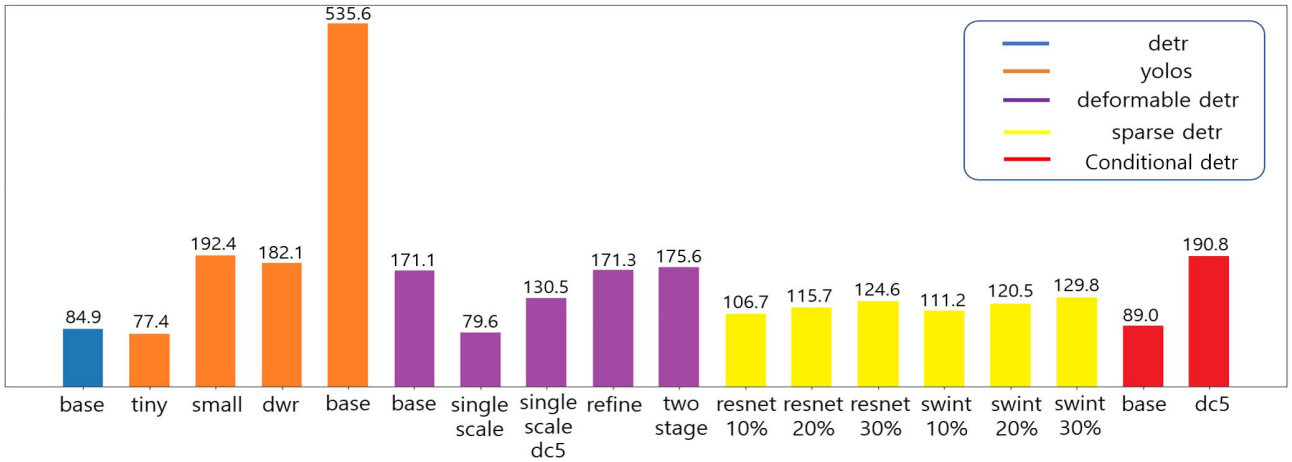

그림 10. 모델 별 GFLOPS
Fig. 10. GFLOPS of various Vision Transformer Models

표 4. FPS, mAP, GFLOPS 사이의 피어슨 상관계수
Table 4. Pearson Correlation Coefficient among FPS, mAP and GFLOPS

	FPS	mAP	GFLOPS
FPS	1.000	-0.667	-0.340
mAP	-0.667	1.000	-0.029
GFLOPS	-0.340	-0.029	1.000

AP, FPS, GFLOPS를 변수로 설정하여 서로와의 상관관계를 계산하였다. 그 결과 표 4와 같이 mAP와 FPS사이에 상관계수가 -0.667로 강한 상충관계에 있음을 확인할 수 있었으며, GFLOPS와 다른 두 변수인 FPS의 상관계수는 각각 -0.340으로 약한 상충관계에 있음을 알 수 있었다. 따라서 높은 성능을 달성한 모델이 효율적으로 성능을 유지하며 연산량을 감소시키는 방법에 관한 연구가 필요하다.

현재 개발되는 인공지능에 사용되는 임베디드 보드는 TFLOPS (Tera FLoating point Operations Per Second)단위의 연산이 가능하다. 따라서 트랜스포머 기반의 모델들 또한 임베디드 보드에 적용하여 실생활에 사용 가능하다. 그러나 합성곱 신경망 기반의 모델들에 비해 아직까지 효율적이지 못하다.

본 논문에서는 두 개의 다른 데이터 집합에 다양한 트랜스포머 기반의 모델의 성능을 분석하였다. 이 결과를 기반으로 추후 실시간성을 보장하는 경량화된 트랜스포머 기반의 객체 탐지 모델 연구를 기대하는 바이다.

References

[1] S. Ren, K. He, R. Girshick, J. Sun, “Faster r-cnn: Towards Real-time Object Detection with Region Proposal Networks,” Proceedings of Advances in Neural Information Processiong Systems, 2015.

[2] J, Redmon, S. Divvala, R. Girshick, A. Farhadi, “You Only Look Once: Unified, Real-time Object Detection,” Proceedings of Computer Vision and Pattern Recognition. pp. 779-788, 2016.

[3] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, D. Tao, “A survey on Vision Transformer,” Journals of IEEE Transections on Pattern Analysis and Machine Intelligence, Vol.45, No. 1, pp. 73-86, 2023.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Proceedings of International Conference on Learning Representations, 2021.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, “End-to-end Object Detection with Transformers,” Proceedings of European Conference on Computer Vision, pp. 213-229, 2020.

[6] Y. Fang, B. Liao, X. Wang, J. Fang, .J Qi, “You Only Look at one Sequence: Rethinking Transformer in Vision Through Object Detection,” Proceedings of Advances in Neural Information Processiong Systems, Vol. 34, pp. 26183-26197, 2021.

[7] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai “Deformable Detr: Deformable Transformers for End-to-end Object Detection,” Proceedings of International Conference on Learning Representations, 2021.

[8] B. Roh, J. W. Shin, W. Shin, S. Kim, “Sparse DETR: Efficient End-to-End Object Detection with Learnable Sparsity,” Proceedings of International Conference on Learning Representations, 2022.

[9] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, J. Wang “Conditional DETR for Fast Training Convergence,” Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3651-3660 2021.

[10] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” Proceedings of European Conference on Computer Vision, pp. 740-750, 2014.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” International Journal of Computer Vision, Vol. 88, No. 2, pp. 303-338, 2010.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, “Attention is all you Need,” Advances in Neural Information Processiong Systems, pp. 6000-6010, 2017.

[13] J. Devlin, M.. W. Chang, K. Lee, K. Toutanova, “Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics, Vol. 1, pp. 4171-4186, 2019.

[14] D. W. Otter, J. R. Medina, J. K. Kalita, “A Survey of the Usages of Deep Learning for Natural Language Processing,” Journal of IEEE transactions on neural networks and learning systems, Vol. 32, No. 2, pp. 604-624, 2020.

[15] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition,” Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[16] P. Dollár, M. Singh, R. Girshick “Fast and Accurate Model Scaling,” Proceedings of Computer Vision and Pattern Recognition, pp. 924-932, 2021.

[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows,” Proceedings of International Conference on Computer Vision, pp. 10012-10022, 2021.

[18] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, “Focal Loss for Dense Object Detection,” Proceedings of IEEE International Conference on Computer Vision, pp. 2980-2988, 2017.

Jungmin Ha (하 정 민)

2022 Automobile and IT Convergence from Kookmin University (B.S.)
2022~Computer Science from Kookmin University (M.S.)

Field of Interests: Object Detection, Computer Vision, Transformer
Email: 20173430@kookmin.ac.kr

Jungmin Eom (엄 정 민)

2021 Automotive Engineering from Kookmin University (B.S.)
2021~Computer Science from Kookmin University (M.S.)

Field of Interests: Computer Vision, Continual Learning
Email: sozzso@kookmin.ac.kr

Hyunjong Lee (이 현 종)

2022 Automobile and IT Convergence from Kookmin University (B.S.)
2022~Computer Science from Kookmin University (M.S.)

Field of Interests: Autonomous Driving, Vision Segmentation
Email: dlrhkswhd10@kookmin.ac.kr

Jaekoo Lee (이 재 구)

2018 Electrical and Computer Engineering from Seoul National University (Ph.D)
2018~College of Computer Science at Kookmin Univsersity (Assistant Professor)

Career:
2011~2013 Assistance Research Engineer, LG Electronics R&D Campus
2018 Data Scientist, SK Telecom Research Center

Field of Interests: Artificial Intelligence(AI), Machine Learning, Deep Learning and Data Science, Autonomous Driving, Object Detection
Email: jaekoo@kookmin.ac.kr