RT-DETR

(DETRs Beat YOLOs on Real-time Object Detection)

김동원

목차

RT-DETR

RT-DETR개요

DETR을 이용하여 YOLOs에 비견될 만한 실시간 object detection모델로 발전시킨 방법에 대한 논문

실시간 객체 탐지를 위한 추론속도와, 학습속도 개선에 대한 내용이다.

DETR의 문제점

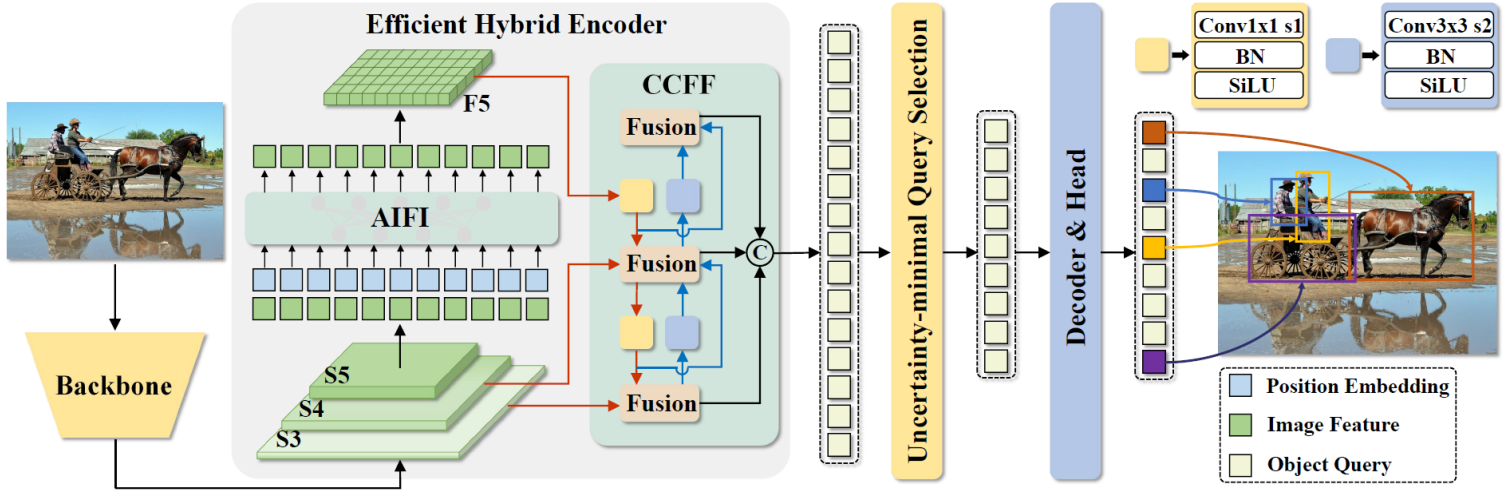
DETR은 느린 학습속도와 transformer block의 연산량에 의한 느린 추론이 문제이다

제안한 해결법

RT DETR은 이러한 문제를 해결하기 위해 Efficient Hybrid Encoder와 uncertainty minimal selection을 사용하였다.

RT-DETR – 구조, efficient hybrid-encoder

RT-DETR은 backbone, efficient hybrid-encoder, decoder & prediction heads로 이루어져 있다.



RT-DETR은 Deformable DETR에서 Backbone 피처 맵에서 특정 위치를 샘플링하여 Attention 메커니즘을 적용하는 방식이 Attention 효율성을 증가시키는 것은 인정하였다. 하지만, backbone 네트워크의 여러 피쳐를 사용함으로 인해 encoder의 입력 시퀀스 길이가 증가하고, 연산량이 증가하여 모델 속도에 대해 병목이라 생각하여 이를 개선하기 위해 Efficient Hybrid-Encoder를 제안하였다.

RT-DETR – efficient hybrid-encoder 도입을 위한 실험

Multi-scale feature(여러 피쳐맵) 사용하는 것이 속도와 성능에 영향을 미치는지에 대해 실험을 진행

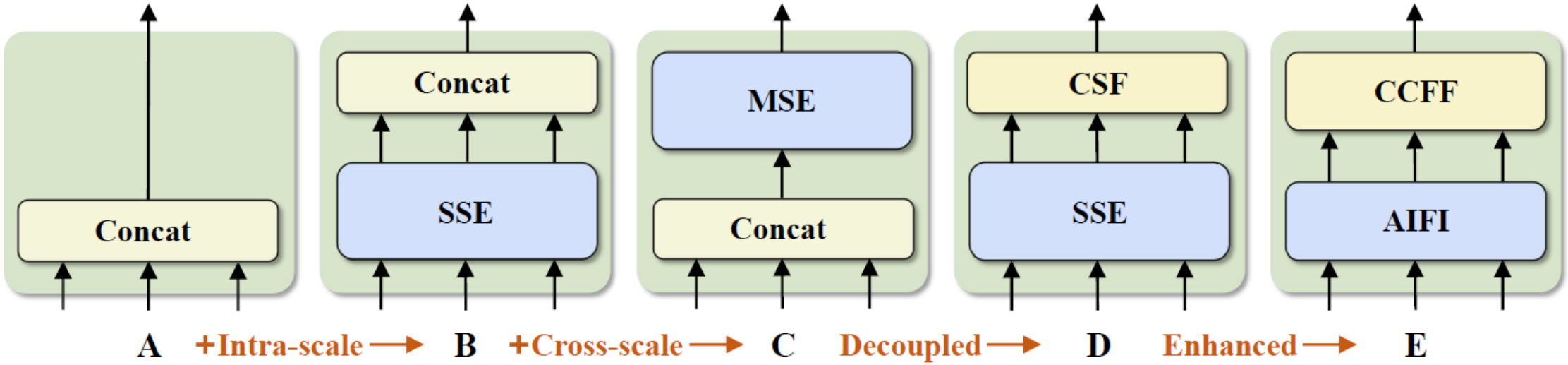


Figure 3. The encoder structure for each variant. **SSE** represents the single-scale Transformer encoder, **MSE** represents the multi-scale Transformer encoder, and **CSF** represents cross-scale fusion. **AIFI** and **CCFF** are the two modules designed into our hybrid encoder.

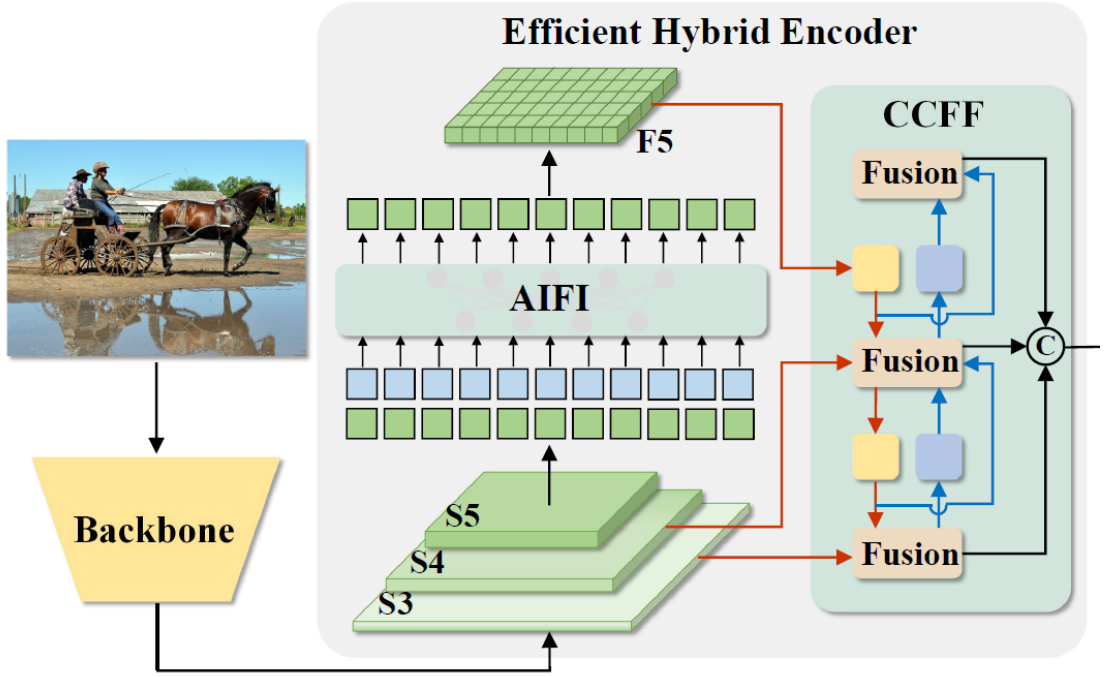
A -〉 B : A에 단일 피쳐맵(intra-scale)에 대한 transformer block을 추가

B -〉 C : 여러 피쳐맵을 합치고 transformer block에 적용시키므로 cross-scale feature fusion을 도입

C -〉 D : intra-scale, cross-scale feature interaction에 대해 독립적인 구조로 구성한다

D -〉 E : D를 발전시켜 efficient hybrid-encoder가 사용하는 방법이다.

RT-DETR – efficient hybrid-encoder



Intra-scale feature를 위한 AIFI와 cross-scale feature를 CCFF로 구성된다. D구조에서 발전시킨 부분은 AIFI에서 backbone network의 최상단의 피쳐맵 만을 transformer encoder에 적용시킨다는 점과, CCFF에서 fusion block을 사용하였다는 것이다.

AIFI는 transformer encoder 구조이고, 이에 최상단 피쳐맵만을 적용시키는 것은 연산량을 줄이고, 정보가 가장 풍부한 가장 최상단의 피쳐맵을 사용하는 것이 중복적인 내용의 학습을 줄일 것이라 예상된다.

RT-DETR – efficient hybrid-encoder

Variant	AP (%)	#Params (M)	Latency (ms)
A	43.0	31	7.2
B	44.9	32	11.1
C	45.6	32	13.3
D	46.4	35	12.2
$D_{\boldsymbol{s}_5}$	46.8	35	7.9
E	47.9	42	9.3

Table 3. The indicators of the set of variants illustrated in Figure 3.

Intra scale feature를 하나의 피쳐맵에 적용시키는 것이 latency가 줄어들고, efficient hybrid encoder는 학습가능한 파라미터가 개수가 증가하지만 latency는 단축되었고, 성능과 latency의 trade-off를 보여주었다.

RT-DETR – efficient hybrid-encoder

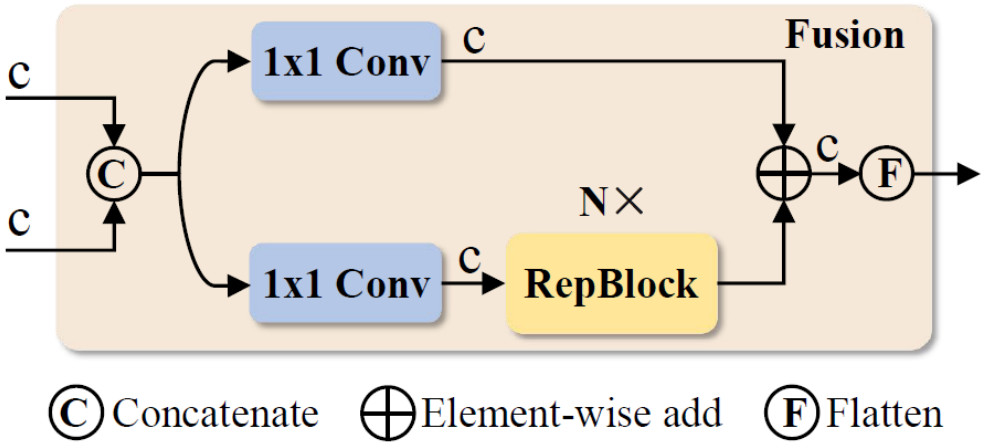
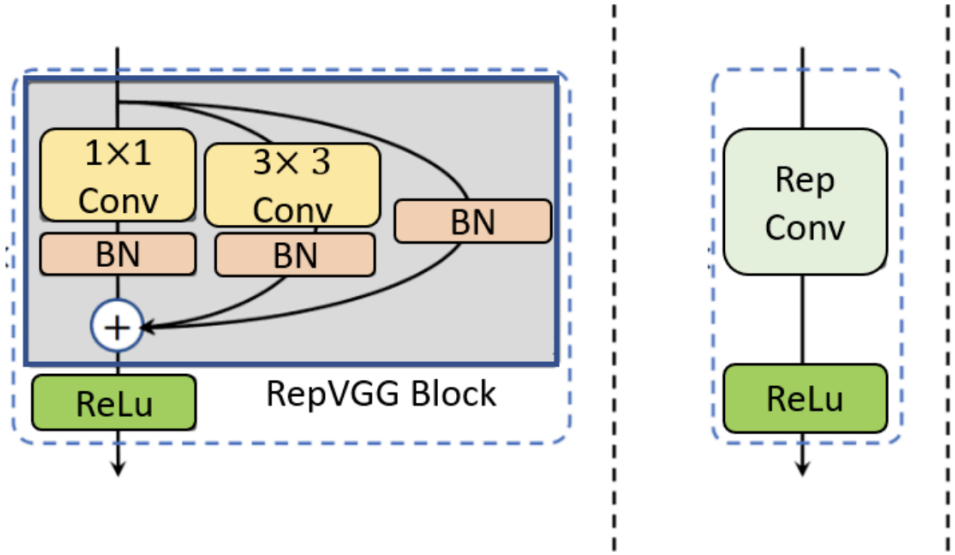
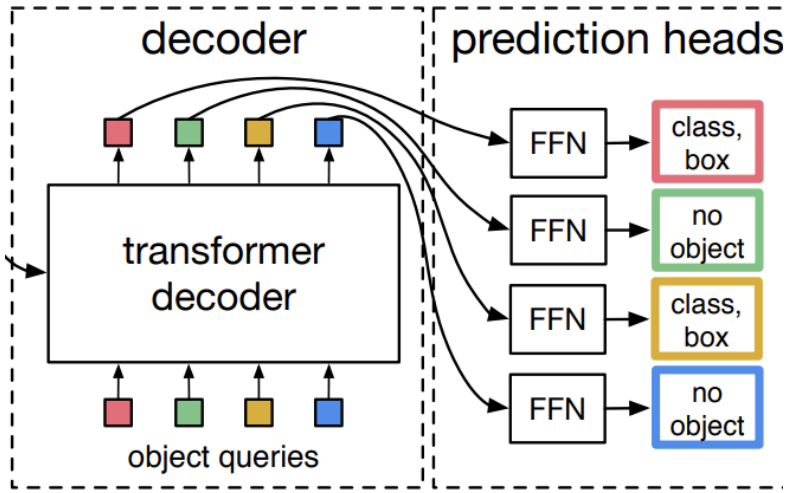


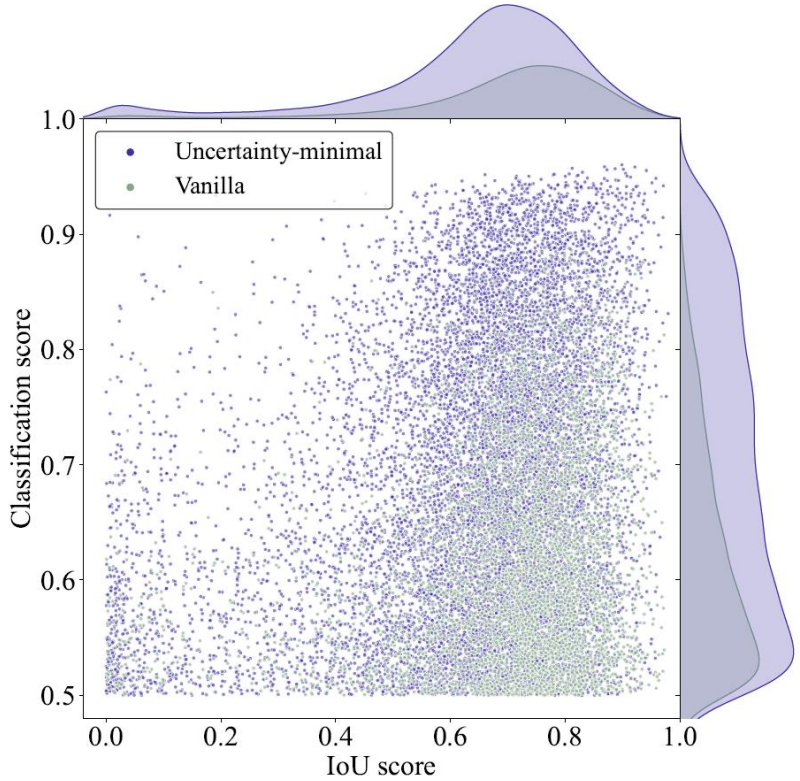
Figure 5. The fusion block in CCFF.



여러 개의 피쳐맵을 합칠때 1x1 conv를 통해 채널수를 조정하고, repconv를 통해 원하는 방향에 따라 학습하고, 추론시에는 빠르게 작동하게 할 수 있다.

RT-DETR – uncertainty minimal query selection





Object query들을 최적화 시키는 것이 어렵기 때문에 uncertainty를 포함한 loss 함수를 정의하고 학습하였을시 top K개의 featur를 사용하여 object query를 초기화 시켰다.

$$\mathcal{U}(\hat{\boldsymbol{x}}) = \|\mathcal{P}(\hat{\boldsymbol{x}}) - \mathcal{C}(\hat{\boldsymbol{x}})\|, \hat{\boldsymbol{x}} \in \mathbb{R}^D$$

$$\mathcal{L}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \boldsymbol{y}) = \mathcal{L}_{box}(\hat{\mathbf{b}}, \mathbf{b}) + \mathcal{L}_{cls}(\mathcal{U}(\hat{\boldsymbol{x}}), \hat{\mathbf{c}}, \mathbf{c})$$

RT-DETR – ablation study

Decoder 부분

ID	AP(%)				Latency (ms)
	Det^4	Det^5	Det^6	Det^7	
7	-	-	-	52.6	9.6
6	-	-	**53.1**	52.6	9.3
5	-	52.9	53.0	52.5	8.8
4	52.7	52.7	52.7	52.1	8.3
3	52.4	52.3	52.4	51.5	7.9
2	51.6	51.3	51.3	50.6	7.5
1	49.6	48.8	49.1	48.3	7.0

Table 5. Results of the ablation study on decoder. **ID** indicates decoder layer index. $\textbf{Det}^k$ represents detector with k decoder layers. All results are reported on RT-DETR-R50 with $6\times$ configuration.

낮은 레이어의 decode일 수록 성능을 떨어지나 latency를 줄어듦. 추가적인 학습을 더 수행하지 않고, 낮은 레이어의 decoder를 사용하여 추론속도 개선이 가능하다.

RT-DETR – 한계, 요약

한계

DETR과 마찬가지로 작은 크기의 이미지에서 성능이 떨어진다

요약

Multi-scale feature를 처리하기 위한 efficient hybrid encode를 사용하여 연산량을 줄이고 빠른 추론속도를 가능하게 하였고, uncertainty-minimal query selection를 통해, 빠른 추론속도와 성능 향상을 보였다. 또한 decoder의 추론하는 layer를 조정 하므로서 latency를 조정 가능하다.

참고자료link

Repblock image