XCiT: Cross-Covariance Image Transformers

School of Industrial and Management Engineering, Korea University

Jong Kook, Heo

KOREA UNIVERSITY

DMQA

Contents

Research Purpose

❖ XCiT : Cross Covariance Image Transformers

- Facebook AI Resarch 에서 연구, 2021년 08월 11일 기준 약 2회 인용
- 기존 Self-Attention 은 모든 토큰들 사이의 상호 작용을 계산하므로 quadratic complexity 를 가지고 있다는 것을 지적
- "Transposed Version of Self-Attention : token 이 아니라 feature channel 간의 Self-attention!!

XCiT: Cross-Covariance Image Transformers

Alaaeldin El-Nouby[1,2] Hugo Touvron[1,3] Mathilde Caron[1,2] Piotr Bojanowski[1]
Matthijs Douze[1] Armand Joulin[1] Ivan Laptev[2] Natalia Neverova[1]
Gabriel Synnaeve[1] Jakob Verbeek[1] Hervé Jégou[1]

[1]Facebook AI [2]Inria [3]Sorbonne University

Abstract

Following tremendous success in natural language processing, transformers have recently shown much promise for computer vision. The self-attention operation underlying transformers yields global interactions between all tokens, *i.e.* words or image patches, and enables flexible modelling of image data beyond the local interactions of convolutions. This flexibility, however, comes with a quadratic complexity in time and memory, hindering application to long sequences and high-resolution images. We propose a "transposed" version of self-attention that operates across feature channels rather than tokens, where the interactions are based on the cross-covariance matrix between keys and queries. The resulting cross-covariance attention (XCA) has linear complexity in the number of tokens, and allows efficient processing of high-resolution images. Our cross-covariance image transformer (XCiT) – built upon XCA – combines the accuracy of conventional transformers with the scalability of convolutional architectures. We validate the effectiveness and generality of XCiT by reporting excellent results on multiple vision benchmarks, including (self-supervised) image classification on ImageNet-1k, object detection and instance segmentation on COCO, and semantic segmentation on ADE20k.

DMQA

Overview

XCiT

❖ Main Contribution

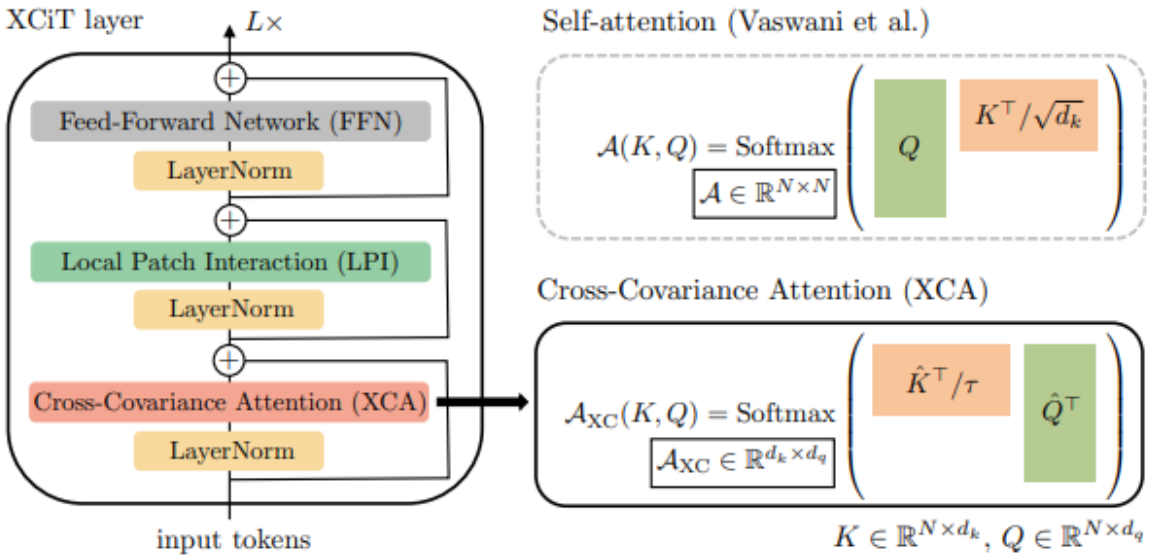
- Self-Attention 을 Token-wise 가 아닌 channel-wise 로 진행하여 계산 복잡도가 patch 개수에 대해 quadratic 하지 않고 linear 함
- Token 개수에 상관없이 고정된 channel 개수에 대해 계산하므로 해상도(resolution) 변화에 robust
- 고해상도 이미지를 이용한 dense prediction task 에서 ResNet 이나 다른 transformer backbone 의 성능을 뛰어넘었다고 함
  - ✓ ADE20k(Semantic Segmentation 벤치마크)에서 SOTA 였던 Swin Transformer 기반 모델 뛰어넘음
- DINO(Self-Distillation with no-labels) 를 통한 자가지도학습으로 학습할 시 ImageNet Top1 Acc 80.9 보여줬다고 함(linear evaluation)

Overview

XCiT

❖ XCiT layer

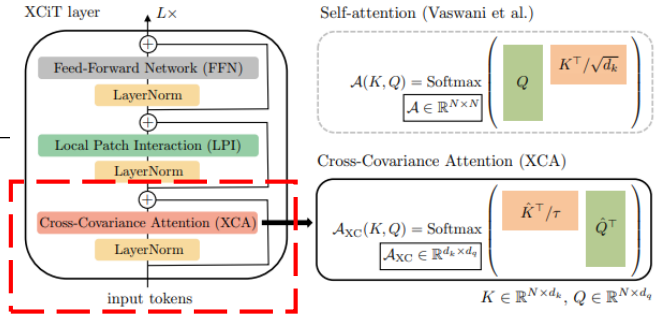
- Can be regarded as "dynamic" 1 by 1 convolution
  - ✓ (B, N, C) 사이즈의 input patch sequence 를 (B, N, C) 의 output patch sequence 로 뱉어냄으로써 채널 개수가 같은 1 x 1 Conv 와 같다고 볼 수 있음
  - ✓ XCiT layer 는 Convolution filter 처럼 static 하지않고, query 와 key 로부터 구한 data-dependent filter 라는 것이 차이점!



논문 표기 및 그림 비례가 이상하게 된 듯...

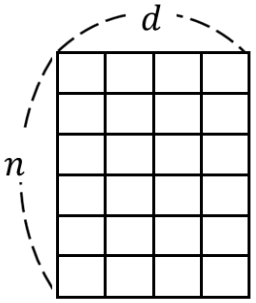
Overview

XCiT

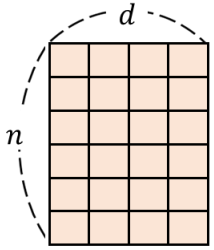


❖ XCiT layer : Cross-Covariance Attention (XCA)

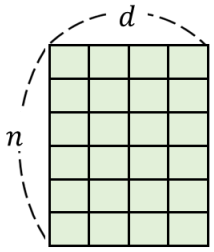
- Preliminaries(Self-Attention)
  - ✓ **Matrix X : n * d matrix (patch 개수, embedding dim)**



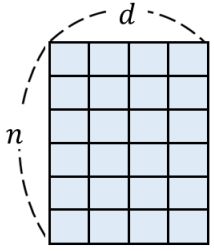
X



$Q = XW_q$



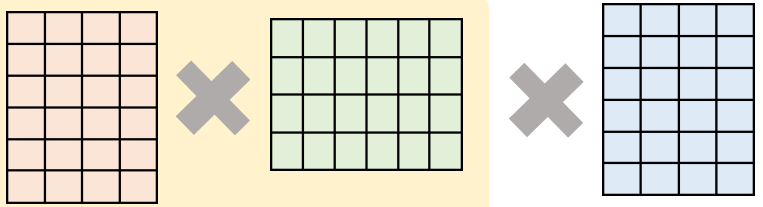
$K = XW_k$



$V = XW_v$

$W_q, W_k, W_v : R^{d*d}$

Softmax & scaling



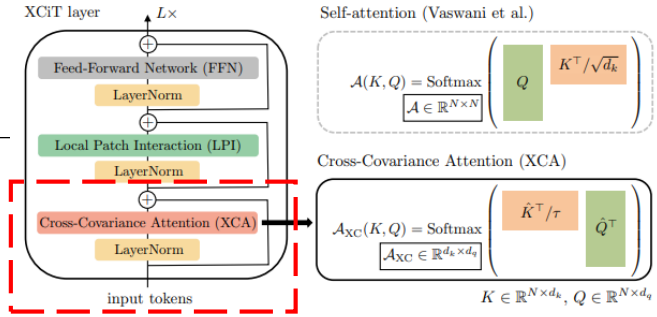
n by n self attention score map(patches)

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$

DMQA

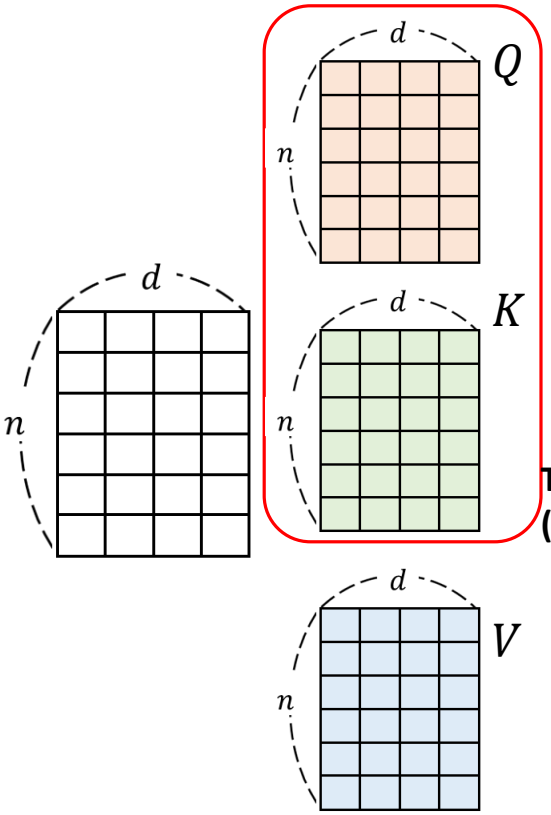
Overview

XCiT

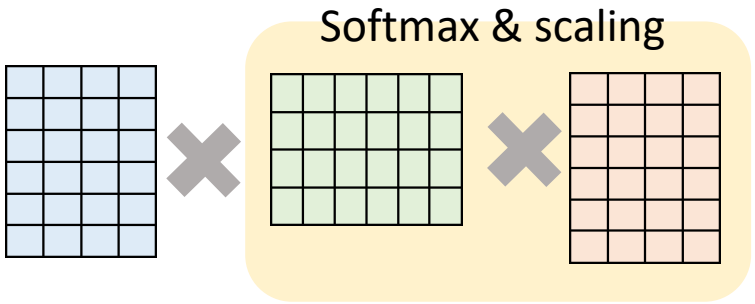


❖ XCiT layer : Cross-Covariance Attention (XCA)

- Cross-Covariance Attention(XCA)



Transpose & L2 normalize
(across patch length dimension)



d by d self attention score map(channel)

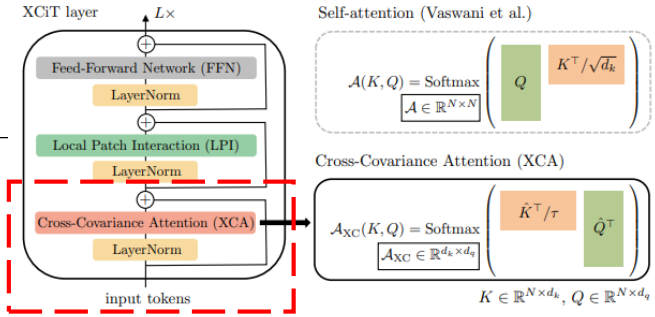
$$XCA(Q,K,V) = V * Softmax\left(\frac{\hat{K}^T\hat{Q}}{t}\right)$$

1. $\hat{Q}$ *is L2 normalized version of* Q
2. t *is learnable parameter*

DMQA

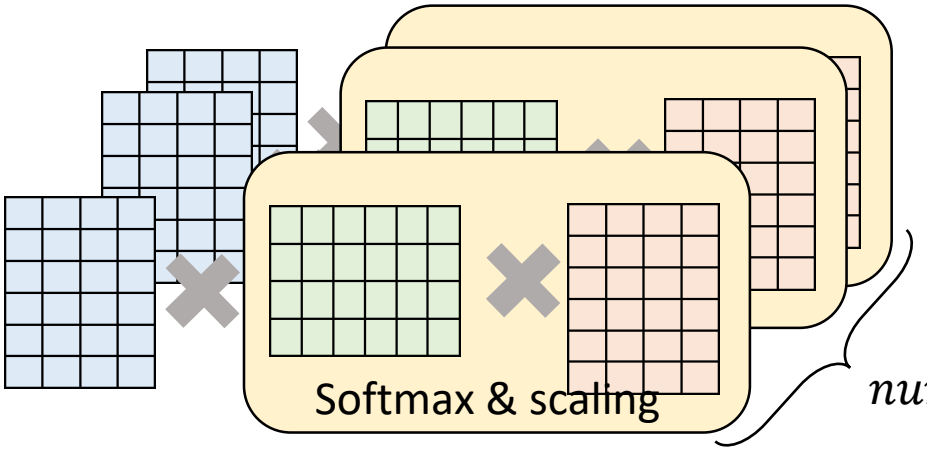
Overview

XCiT



❖ XCiT layer : Cross-Covariance Attention (XCA)

- Cross-Covariance Attention(XCA)
  - ✓ Remind 1 : XCiT Layer 는 channel-wise 1 by 1 convolution 의 역할
  - ✓ Remind 2 : 이 때, 1 by 1 convolution 으로 작용하는 Attention 은 data-dependent 한 것이 차이점!
  - ✓ Block-diagonal Cross-Covariance attention(additional)
    - channel dimension(d) 를 단일 head 가 아닌 multi-head 로 각각 처리
    - 계산 복잡도가 head 개수(h)의 비율로 감소
    - 단일 head 버전보다 최적화하기 더 쉬움을 발견



$h * d' = d$

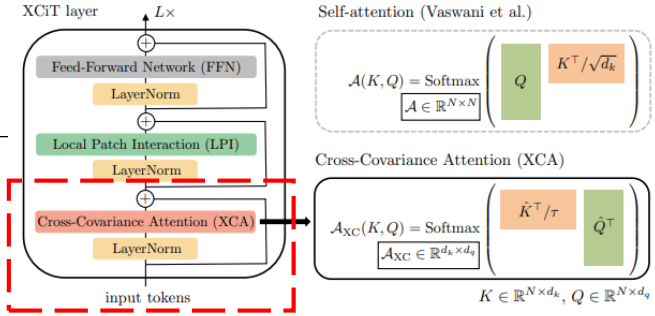
$num\ heads : h$

$V \in R^{n*d'}$ $\quad Attention \in R^{d'*d'}$

DMQA

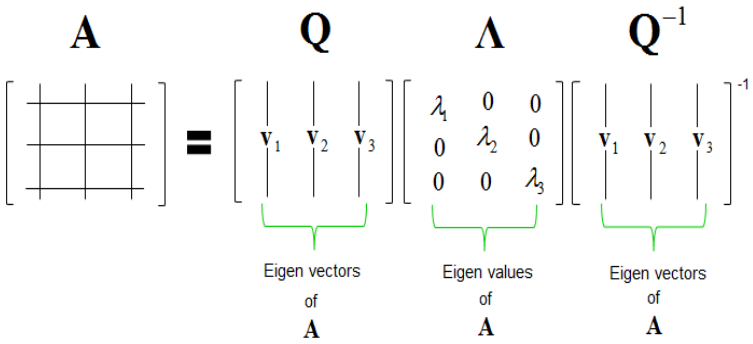
Overview

XCiT



❖ XCiT layer : Cross-Covariance Attention (XCA)

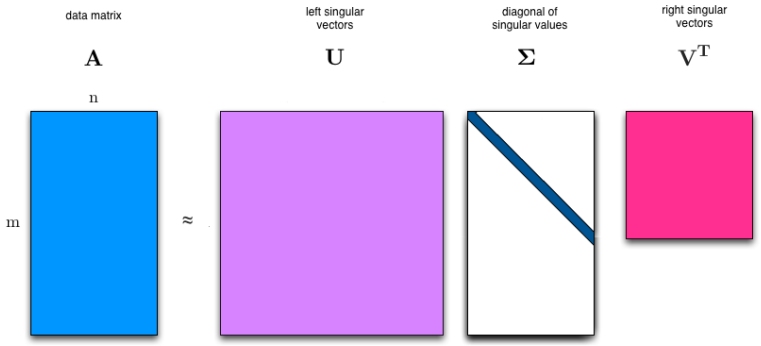
- Motivation : Preliminaries (Eigen-value Decomposition & SVD)



각 eigenvector 는 방향만 있고 크기는 1 이며 항상 직교!

$$if\ A\ is\ symmetric,$$
$$then\ Q^T = Q^{-1} \Rightarrow Q^TQ = QQ^T = I$$

Eigen-value Decomposition



U 와 V 는 각각 서로 직교하는 단위 벡터의 집합

$$\rightarrow\ U^T = U^{-1} \Rightarrow U^TU = UU^T = I$$

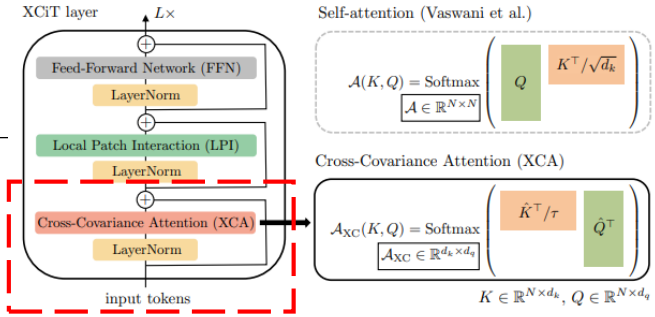
★ $U: Eigenvectors\ of\ AA^T$
$V: Eigenvectors\ of\ A^TA$

Singular Value Decomposition

DMQA

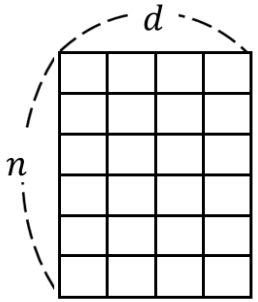
Overview

XCiT

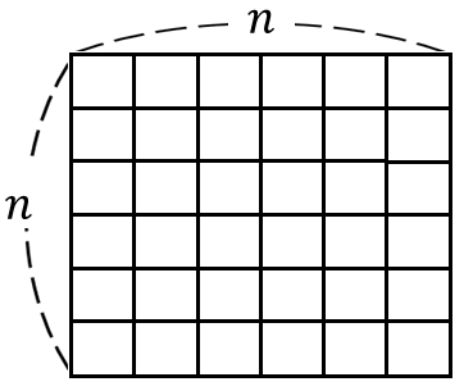


❖ XCiT layer : Cross-Covariance Attention (XCA)

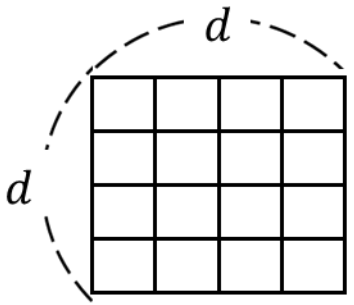
- Motivation : Preliminaries(Gram Matrix & Covariance Matrix)
  - ✓ 저자들은 Gram Matrix 와 Covariance Matrix 의 관계를 조명
  - ✓ **앞서 살펴 본 고윳값 분해와 특잇값 분해의 성질을 이용해 둘 중 하나를 알면, "적은 계산량으로" 나머지 하나를 유추할 수 있다!!**



X



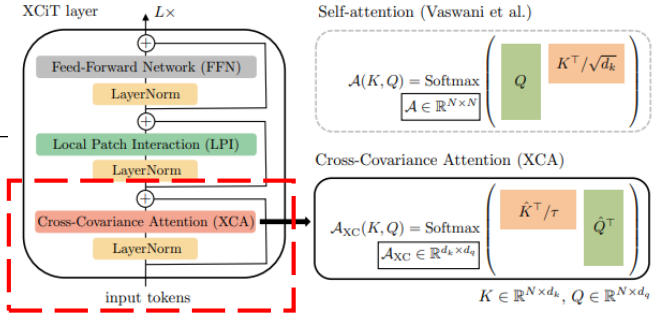
$Gram\ Matrix = XX^T$



$Covariance\ \ Matrix = X^TX$

Overview

XCiT



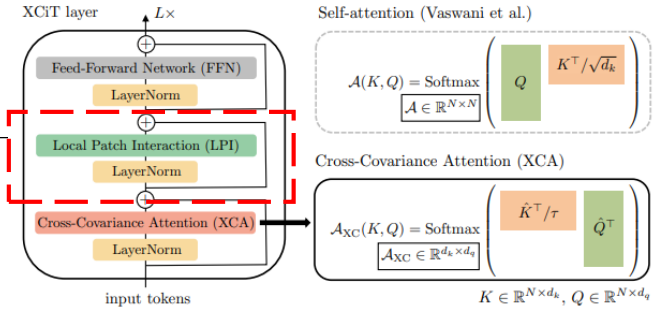
❖ XCiT layer : Cross-Covariance Attention (XCA)

- Motivation
  - ✓ **Self-Attention Matrix : $QK^T = XW_qW_kX^T \in R^{n*n}$**
  - ✓ **(XCA)Cross-Covariance Attention Matrix : $K^TQ = XW_qW_kX^T \in R^{n*n}$**
  - ✓ **Self-Attention 을 구하는데는 patch 개수에 대해 Quadratic Complexity 가 소모되지만, XCA 는 Linear Complexity 를 가진다. 따라서 더 적은 계산량으로 Query, Key Matrix 를 잘 학습시킬 수 있다!**

DMQA

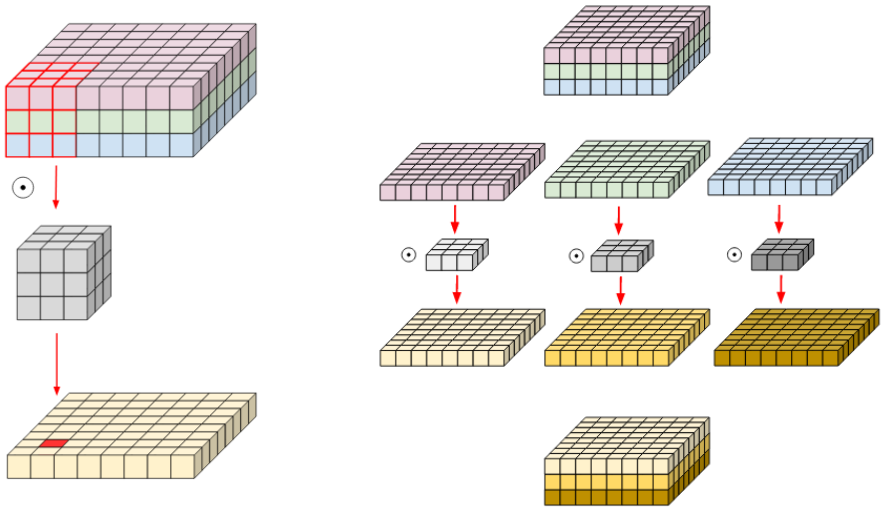
Overview

XCiT



❖ XCiT layer : Local Patch Interaction(LPI)

- XCA 모듈은 channel-wise Attention 이기 때문에 patch-wise communication 을 포착하기는 어려움
  - ✓ Q, K, V 가 data-dependent 한 matrix 라서 어느 정도 포착할 수 있긴함(implicit shared statistic)
  - ✓ 좀 더 explicit 한 patch-wise communication 을 잡아내고자 LPI 라는 모듈 도입
- LPI Explanation
  - ✓ 사실 그냥 depth-wise 3 by 3 Convolution 2개와, BN, GELU 를 섞은 Layer
  - ✓ Convolution Layer 를 통해 지역적인 특성 포착
  - ✓ Depthwise Convolution 은 Convolution Filter 가 각각 group 내에 있는 channel 만 계산하는 것



일반 Conv filter 는 모든 입력 채널에 대하여 계산하지만, depth-wise conv filter 는 filter 별로 계산하는 채널이 다름

DMQA

Details

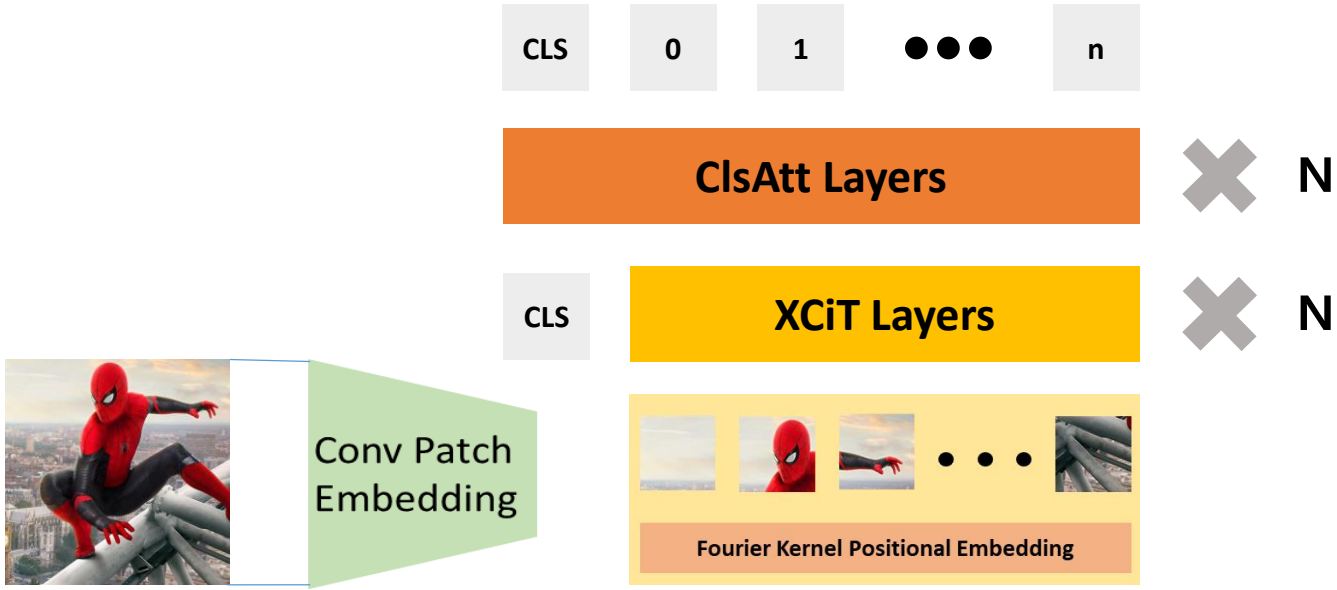
Architecture Details

❖ Class Attention Module

- CaiT(Tourvron et al, 2021) 에서 제안된 모듈, [CLS] 의 Query 에 대해서만 Attention 을 계산
- CLS token 은 XCiT Layer 를 다 통과한 후, Class Attention Module 을 통과하기 전에 생성

❖ Patch Embedding

- Conv 3 by 3 과 GELU 로 이루어진 embedding layers
- Positional embedding 은 DETR 의 Fourier Kernel Encoding 을 사용



Experiments

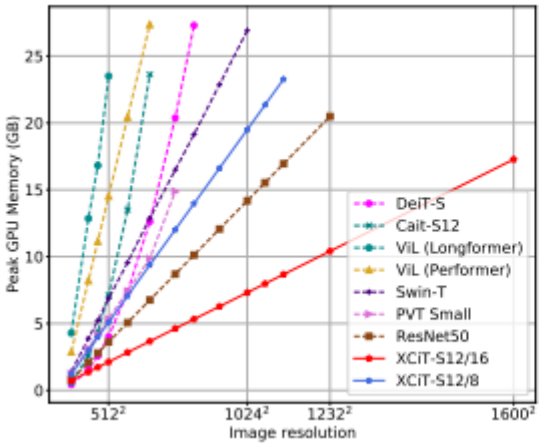
Handling Variable Resolutions

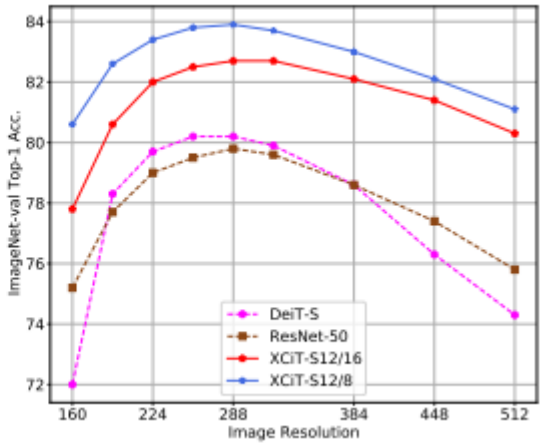
❖ XCiT scale linearly in the number of tokens(Fig. 2)

- 더 큰 이미지도 효율적으로 처리 할 수 있음

❖ XCiT shows more robust performance when train-test resolution discrepancy exists(Fig. 3)

- 모든 네트워크를 224 by 224 에 train 한 후, test 단계에서 해상도를 변화시키며 성능을 관찰
- Train 224/ test 288 일 때 성능이 제일 좋은데, 이러한 효과를 FixRes effect 라고 칭함*





*H Touvron, A Vedaldi, M Douze, and H Jégou. Fixing the train-test resolution discrepancy. Advances in Neural Information Processing Systems, 2019.

DMQA

Experiments

Visualization of the attention map

❖ Attention Score map with CLS token at Class Attention Stage

- 각각의 헤드가 이미지 내에서 의미적으로 비슷한 feature 를 잘 잡음(Each head seems salient to semantically coherent regions)
  - ✓ attention head 가 동일 이미지에 존재하는 우산에 대해서 잘 탐지 (초록색)
- 모든 헤드는 동일 이미지 뿐만 아니라 다른 이미지 내에서도 비슷한 feature 를 찾으려고 노력(Heads are sensitive to similar features within the same or across images)
  - ✓ 사람 머리를 탐지한 head 가 새의 머리를 탐지(노란색)

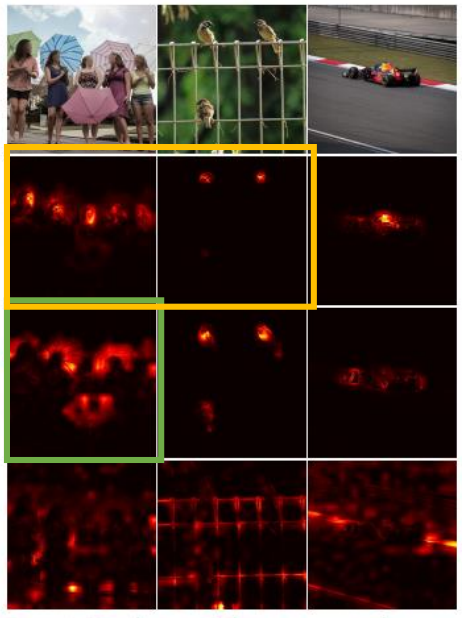


Figure 4: Visualization of the attention map between the CLS token and individual patches in the class-attention stage. For each column, each row represents the attention map w.r.t. one head, corresponding to the image in the first raw. Each head seems salient to semantically coherent regions. Heads are sensitive to similar features within the same or across images (*e.g.* people or bird faces). They trigger on different concepts when such features are missing (*e.g.*, cockpit for race cars).

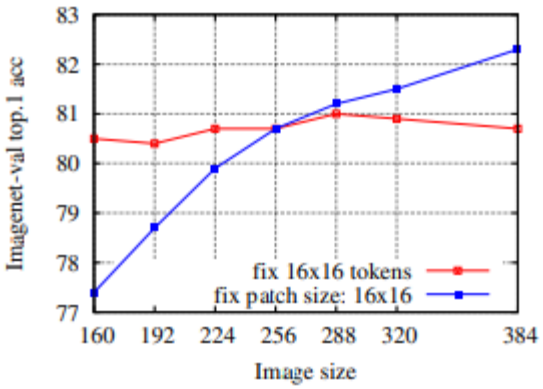
Experiments

Additionals

❖ Impact of resolution versus patch-size(Appendix A.1)

- ViT 아키텍처에 관한 실험(XCiT 에서도 비슷한 트렌드를 발견 했다고함)
- When Resolution is increased..
  - Patch 크기 ↑ token 개수 fix vs patch 크기 fix token 개수 ↑
  - 성능 향상의 주 요인(main driver) 는 patch 개수!!

A.1 Impact of resolution versus patch size



Variable patch size

Image Size	80	112	160	256	320	384
Patch Size	5	7	10	16	20	24
Top-1	78.2	79.7	80.5	80.7	80.9	80.7

Variable number of tokens size

Image Size	160	224	256	288	320	384
# of tokens	100	196	256	324	400	576
Top-1	77.4	79.9	80.7	81.2	81.5	82.3

Figure A.1: **Impact of input resolution on accuracy for DeiT-S**. We consider different image resolutions, and either (1) increase the patch size while keeping the number of tokens fixed; or (2) keep the patch size fixed and use more tokens. Larger input images are beneficial if the number of tokens increases. The impact of a change of a resolution for a constant number of patches (of varying size) is almost neutral. As one can observe, the main driver of performance is the number of patches. The patch size has a limited impact on the accuracy, except when considering very small ones. We have observed and confirmed similar trends with XCiT models.

Conclusion

- ViT 연산 중 대부분을 차지하는 Attention 의 계산복잡도를 줄이고, 해상도 변화에 Robust 하다는 것이 Main Contribution

- Swin Transformer 와 마찬가지로 Object Detection, Instance/Semantic Segmentation 의 backbone 으로 활용, 일부 벤치마크에서 SOTA 였던 Swin Transformer backbone 을 뛰어넘었다고 주장
  - 근데 paperswithcode 보니 해당 벤치마크에서도 항상 상회하는 것은 아니었음
  - (참고) 인용수 Swin Transformer 139회 > XCiT 2회

- Yannic Kilcher says, "XCiT 가 Transformer 로 볼 수 있는 지 애매하다. 이건 patch 간의 관계를 보는 Self-Attention 이 아니기 때문에 ConvNet 에 더 가까운 것 같다"(필자도 일부 동의)

- 논문에서 Cross-Covariance 에 대한 명확한 정의를 해주지 않는게 좀 아쉬움
  - Cross-Covariance 는 주로 시계열 데이터에 "서로 다른 두 시그널의 다른 시점에 대한 공분산" 에서 주로 쓰이는데, 해당 논문에서는 그런 의미와는 살짝 다른 듯