Rethinking Spatial Dimensions of Vision Transformers

School of Industrial and Management Engineering, Korea University

Jae Hoon Kim

KOREA UNIVERSITY

DMQA

Contents

Research Purpose

❖ Rethinking Spatial Dimensions of Vision Transformers (arXiv, 2021)

- Naver AI Lab에서 연구하였으며 2021년 07월 11일 기준으로 15회 인용됨

Rethinking Spatial Dimensions of Vision Transformers

Byeongho Heo Sangdoo Yun Dongyoon Han Sanghyuk Chun Junsuk Choe Seong Joon Oh

NAVER AI Lab

Abstract

Vision Transformer (ViT) extends the application range of transformers from language processing to computer vision tasks as being an alternative architecture against the existing convolutional neural networks (CNN). Since the transformer-based architecture has been innovative for computer vision modeling, the design convention towards an effective architecture has been less studied yet. From the successful design principles of CNN, we investigate the role of the spatial dimension conversion and its effectiveness on the transformer-based architecture. We particularly attend the dimension reduction principle of CNNs; as the depth increases, a conventional CNN increases channel dimension and decreases spatial dimensions. We empirically show that such a spatial dimension reduction is beneficial to a transformer architecture as well, and propose a novel Pooling-based Vision Transformer (PiT) upon the original ViT model. We show that PiT achieves the improved model capability and generalization performance against ViT. Throughout the extensive experiments, we further show PiT outperforms the baseline on several tasks such as image classification, object detection and robustness evaluation. Source codes and ImageNet models are available at https://github.com/naver-ai/pit.

convolution operation, has emerged in computer vision.

ViT is quite different from convolutional neural networks (CNN). Input images are divided into 16×16 patches and fed to the transformer network; except for the first embedding layer, there is no convolution operation in ViT, and the position interactions occur only through the self-attention layers. While CNNs have restricted spatial interactions, ViT allows all the positions in an image to interact through transformer layers. Although ViT is an innovative architecture and has proven its powerful image recognition ability, it follows the transformer architecture in NLP [35] without any changes. Some essential design principles of CNNs, which have proved to be effective in the computer vision domain over the past decade, are not sufficiently reflected. We thus revisit the design principles of CNN architectures and investigate their efficacy when applied to ViT architectures.

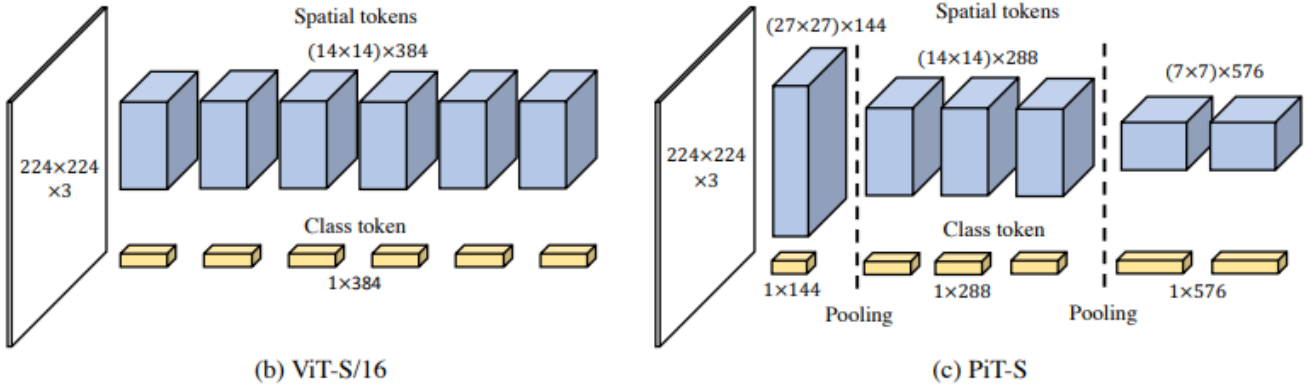
CNNs start with a feature of large spatial sizes and a small channel size and gradually increase the channel size while decreasing the spatial size. This dimension conversion is indispensable due to the layer called spatial pooling. Modern CNN architectures, including AlexNet [21], ResNet [13], and EfficientNet [33], follow this design principle. The pooling layer is deeply related to the receptive field size of each layer. Some studies [6, 27, 5] show that the pooling layer contributes to the expressiveness and gen-

DMQA

Research Purpose

❖ Rethinking Spatial Dimensions of Vision Transformers (arXiv, 2021)

- Pooling layer를 추가함으로써 ViT의 예측 및 일반화 성능이 향상됨을 **실험으로 증명**한 논문
- 현재 이미지 분석에 좋은 성능을 보이는 CNN 기반의 모델은 대부분 pooling layer를 포함하고 있으며 **pooling layer는 CNN 필터의 인식 범위(receptive field)를 조절하는 주요 기능**에 해당함
- **ViT는 모델 중간에 이미지 크기를 다루는 과정이 없으므로** 기존 이미지 분석에서 효과적이었던 기법인 pooling을 도입하면 성능이 향상될 것으로 가정함
- 이에 Vision Transformer(ViT)에 pooling 기법을 적용한 Pooling-base Vision Transformer(PiT)를 제안함

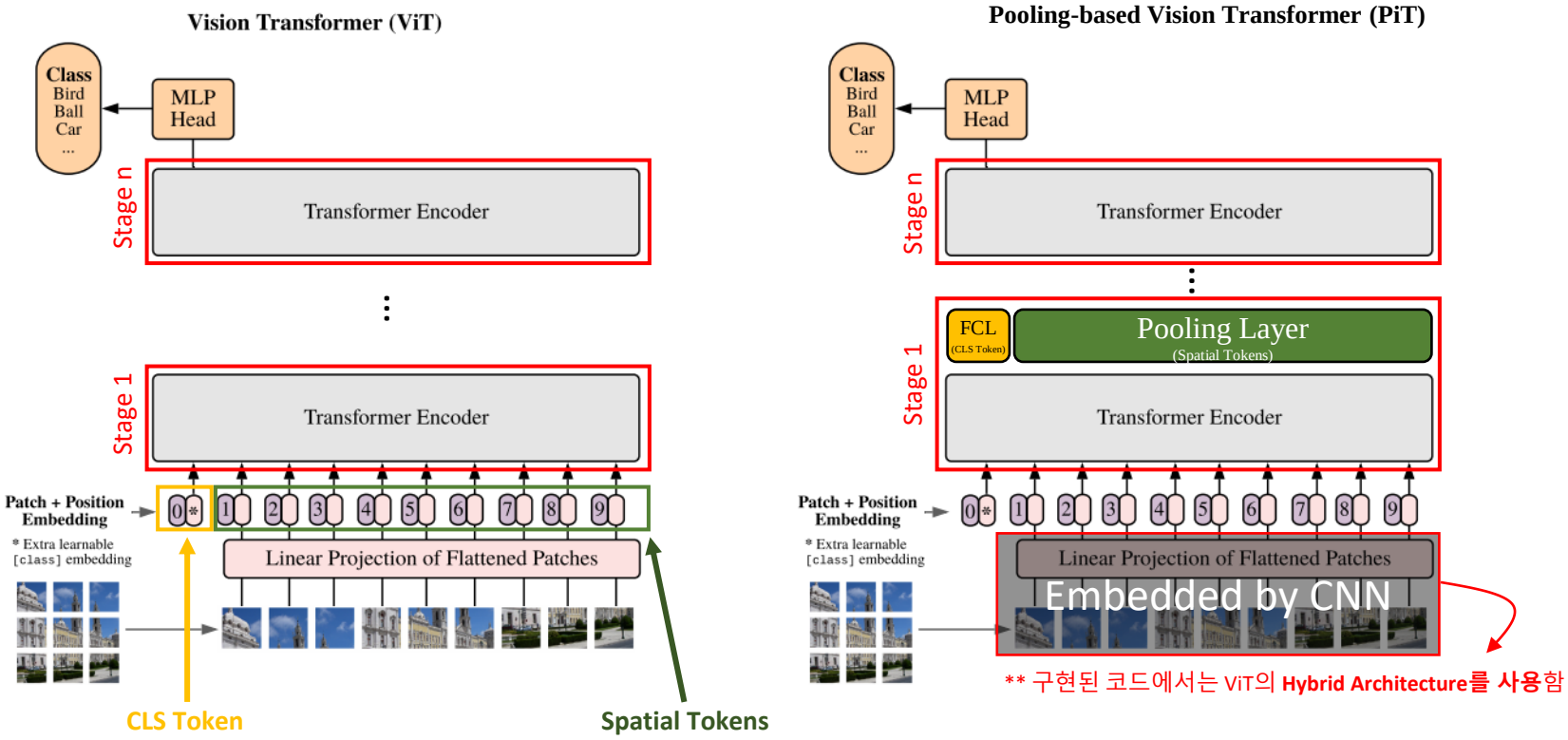


DMQA

Pooling-based Vision Transformer (PiT)

❖ Diagram of PiT (vs. ViT)

- PiT는 ViT의 아키텍처에서 pooling layer를 추가한 모델
- CLS token은 fully-connected layer (FCL)에서 처리하며 spatial token은 pooling layer에서 처리

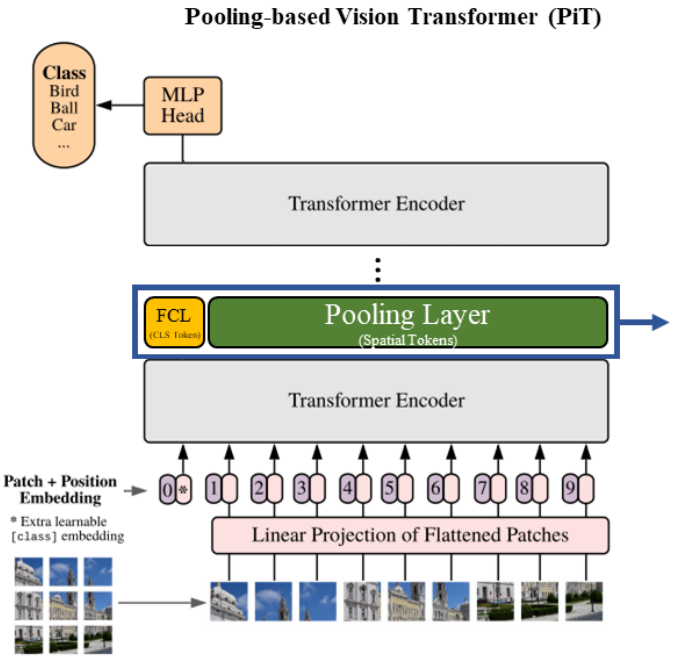


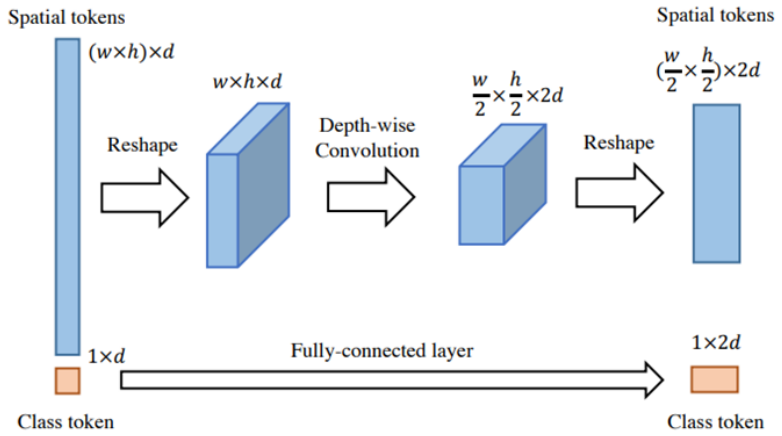
DMQA

Pooling-based Vision Transformer (PiT)

❖ Diagram of PiT (Pooling Layer)

- Transformer로부터 계산된 Spatial tokens와 CLS token을 입력 값으로 받음
- Spatial tokens는 pooling 연산을 수행하기 위하여 2D-matrix 형태에서 3D-tensor 형태로 변환
- Convolution 연산 시 stride 값을 2로 설정하여 pooling 연산을 대체함
- CLS token은 FC layer를 통해서 pooling 연산을 마친 spatial tokens와 같은 채널 사이즈로 조정함





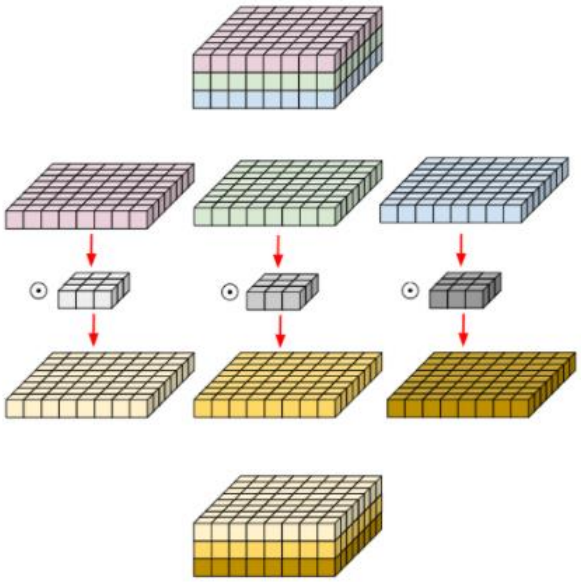
** **Pooling 연산 시** 일반적으로 쓰이는 average, max pooling이 아닌 **stride를 2 이상으로 설정한 depth-wise convolution**을 사용함

DMQA

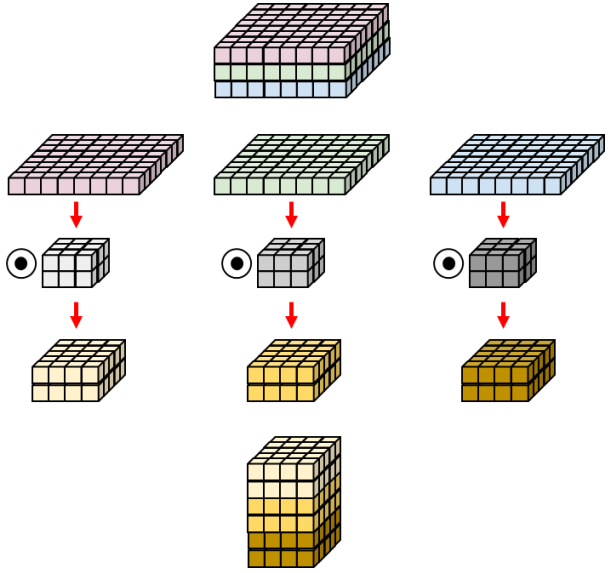
Pooling-based Vision Transformer (PiT)

❖ Diagram of PiT (Depth-wise Convolution)

- 일반적인 convolution 필터는 입력 이미지의 모든 채널을 반영하여 연산을 함
- Depth-wise convolution은 각 단일 채널에 대해서만 수행되는 필터를 사용함
  - ✓ 채널 방향의 convolution은 진행하지 않으며, 공간 방향의 convolution만을 진행
  - ✓ PiT에서는 필터의 하이퍼 파라미터를 조정하여 채널 수가 증가하고 이미지 크기가 줄어들도록 설정



Depth-wise Convolution
(Ordinary)



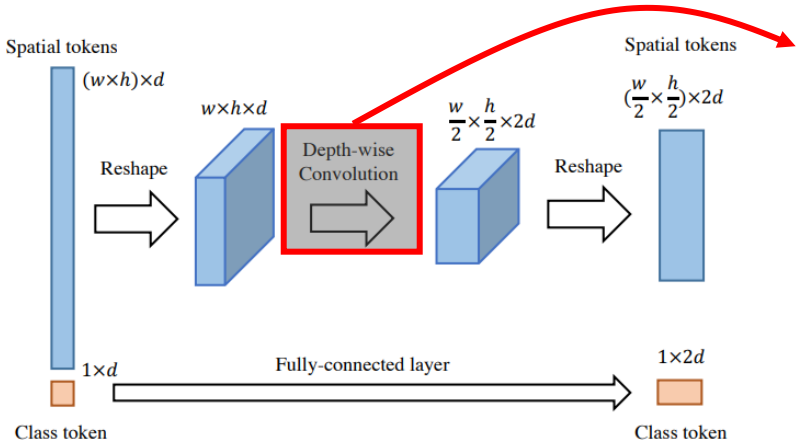
Depth-wise Convolution
(PiT)

출처: https://eehoeskrap.tistory.com/431

Pooling-based Vision Transformer (PiT)

❖ Diagram of PiT (Depth-wise Convolution)

- Pooling은 채널 간 연산을 하지 않기 때문에 depth-wise convolution 연산을 진행한 것으로 **추정**




```
class conv_head_pooling(nn.Module):
    def __init__(self, in_feature, out_feature, stride,
                 padding_mode='zeros'):
        super(conv_head_pooling, self).__init__()

        self.conv = nn.Conv2d(in_feature, out_feature, kernel_size=stride + 1,
                              padding=stride // 2, stride=stride,
                              padding_mode=padding_mode, groups=in_feature)
        self.fc = nn.Linear(in_feature, out_feature)
```

Spatial Tokens 연산

CLS Tokens 연산

groups의 인자가 **입력 채널 수와 동일할 경우 depth-wise 연산** 수행

출처: https://github.com/naver-ai/pit/blob/master/pit.py (line 54)

```
if stage < len(heads) - 1:
    self.pools.append(
        conv_head_pooling(base_dims[stage] * heads[stage],
                          base_dims[stage + 1] * heads[stage + 1],
                          stride=2
        )
```

Stride 크기는 2로 고정하여 사용

출처: https://github.com/naver-ai/pit/blob/master/pit.py (line 130)

DMQA

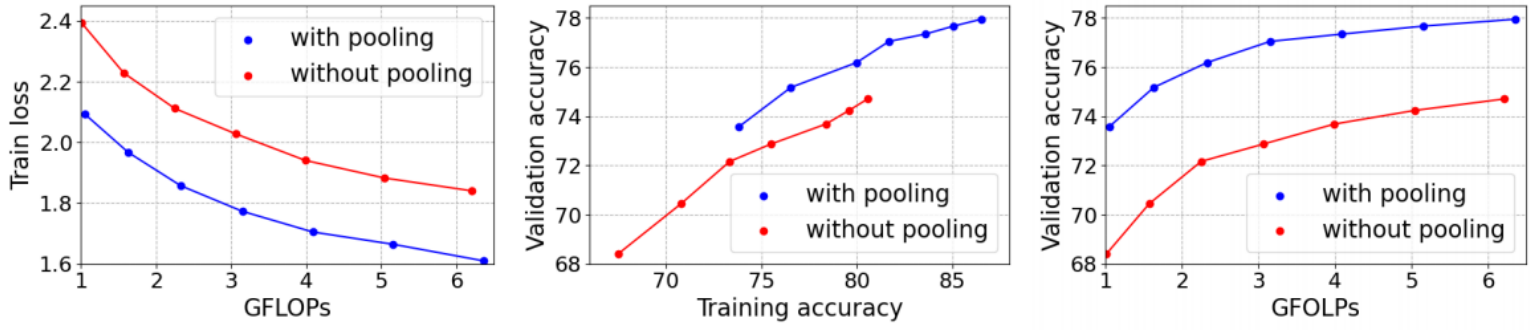
Experiments

❖ ResNet-50

- 우선 CNN 기반 모델에서 pooling layer 포함 여부에 따른 비교 실험을 세 가지 관점에서 진행함
  - ✓ Model capability : 같은 연산량에서 얼마나 더 효과적으로 학습하는가?
  - ✓ Generalization performance : 훈련데이터셋 성능 대비 검증데이터셋에서의 성능은?
  - ✓ Model performance : 같은 연산량에서 얼마나 더 좋은 성능을 보이는가?

- 모든 실험에서 pooling layer를 포함한 경우가 더 좋은 성능을 보임
- 따라서 pooling layer가 모델의 예측 및 일반화 성능의 향상에 효과적임

** 실험에는 ImageNet dataset이 쓰임



(a) Model capability (b) Generalization performance (c) Model performance

** GFLOPs (GPU FLoating point OPerations): GPU의 부동소숫점 연산량을 의미함

DMQA

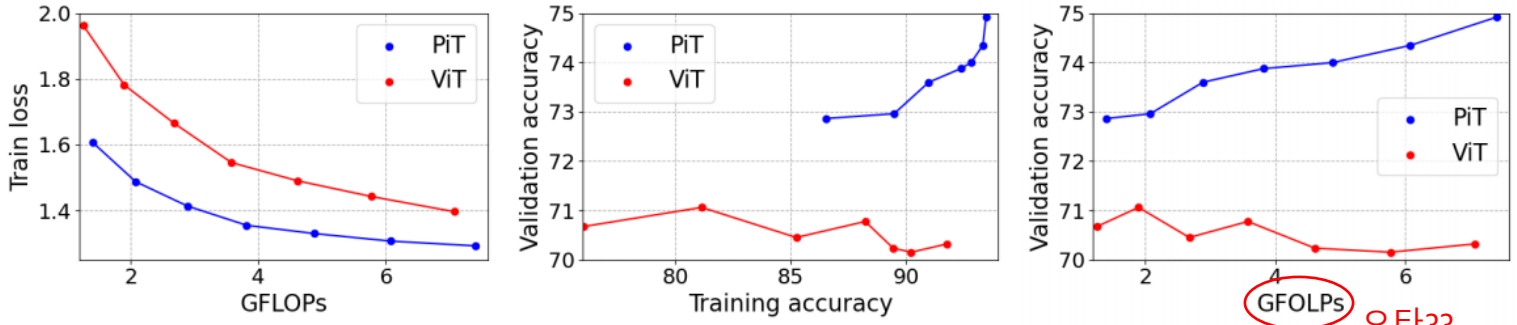
Experiments

❖ Pooling-based Vision Transformer (PiT)

- 마찬가지로 PiT에서 pooling layer 포함 여부에 따른 비교 실험을 세 가지 관점에서 진행함
  - ✓ Model capability : 같은 연산량에서 얼마나 더 효과적으로 학습하는가?
  - ✓ Generalization performance : 훈련데이터셋 성능 대비 검증데이터셋에서의 성능은?
  - ✓ Model performance : 같은 연산량에서 얼마나 더 좋은 성능을 보이는가?

- 모든 실험에서 pooling layer를 포함한 경우가 더 좋은 성능을 보임
- 따라서 pooling layer가 모델의 예측 및 일반화 성능의 향상에 효과적임

** 실험에는 ImageNet dataset이 쓰임



(a) Model capability (b) Generalization performance (c) Model performance

** GFLOPs (GPU FLoating point OPerations): GPU의 부동소숫점 연산량을 의미함

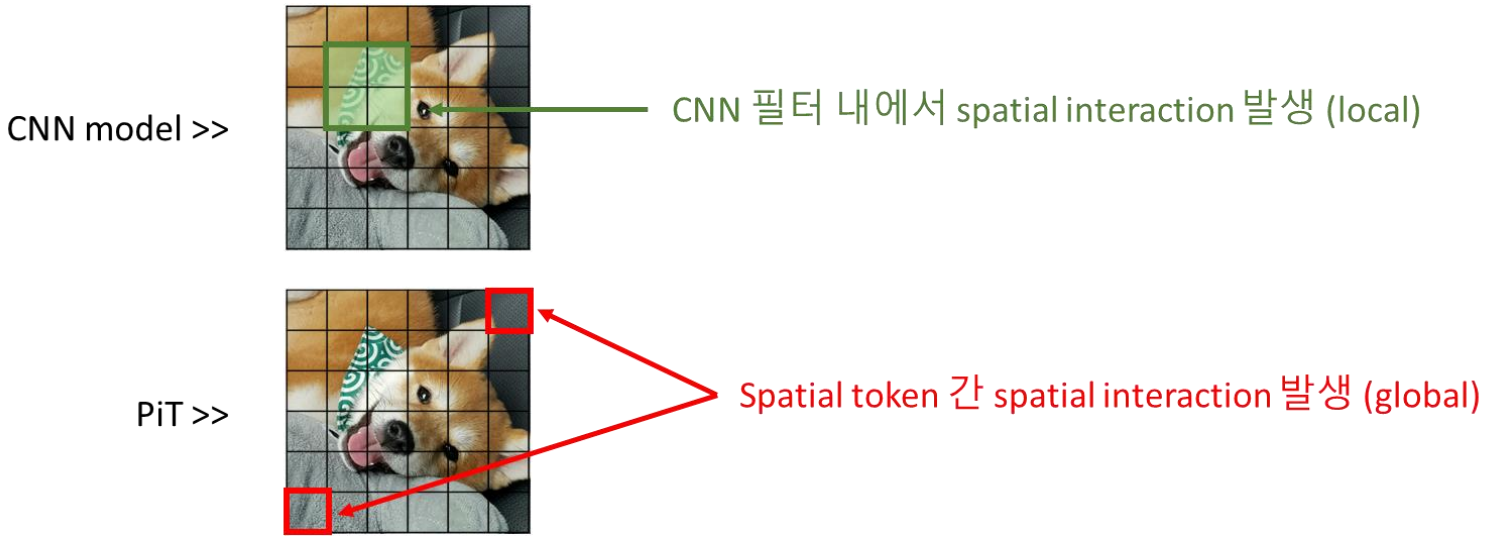
** PiT에서 pooling layer를 뺀 경우는 ViT가 됨

DMQA

Experiments

❖ Spatial interaction (definition)

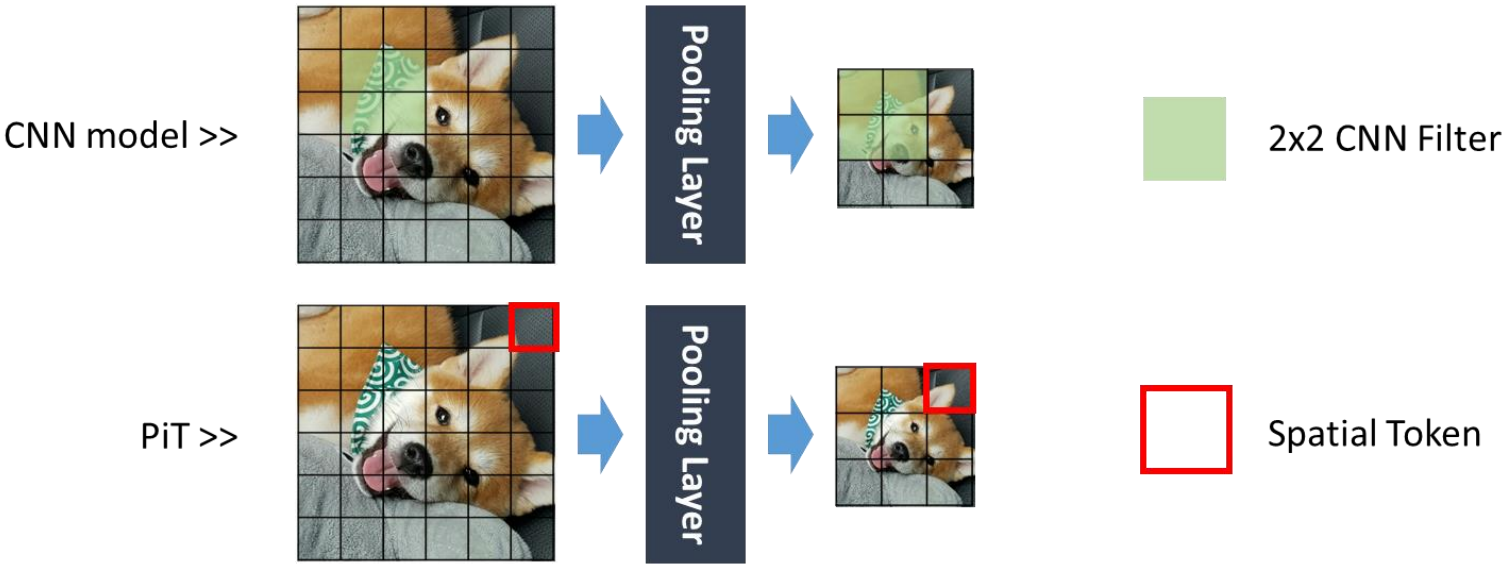
- Token 혹은 receptive field 크기의 단위에서 발생하는 입력 이미지와의 연산
  - ✓ CNN의 경우 3x3 convolution filter는 3x3 spatial interaction을 의미함.
    즉, 근처의 일부 이미지에 대하여 발생함 (local)
  - ✓ ViT (PiT)의 경우 self-attention layer에서 token간의 spatial distance와 관계 없이
    전체 이미지 패치 중 하나에 대하여 발생함 (global)



Experiments

❖ Spatial interaction (definition)

- CNN 기반 모델에서 pooling layer를 통과하면 receptive field의 크기를 늘리는 효과가 있음
- PiT의 경우 pooling layer를 거치면서 입력 이미지의 크기가 줄어들고 spatial token의 크기는 고정되어 있기 때문에 token의 개수는 줄어들면서 한 token이 나타내는 이미지의 범위가 늘어나는 효과가 있음



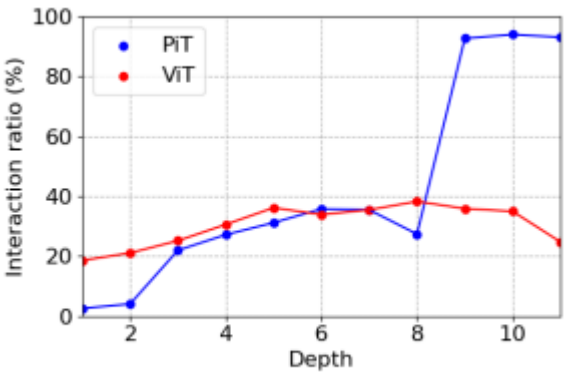
이미지 출처: https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=djnice2503&logNo=221579326685

Experiments

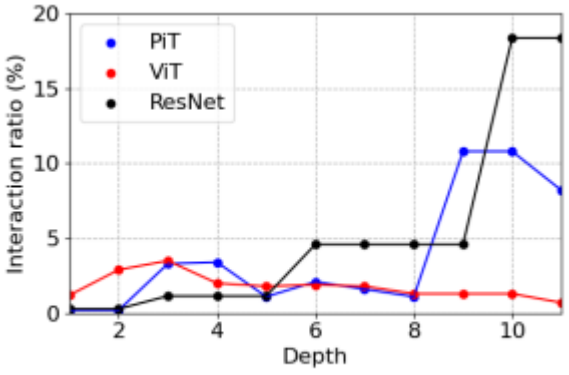
❖ Spatial interaction (measure)

- ViT와 PiT에서 spatial interaction이 발생하는 영역은 attention matrix에 softmax를 취하여 특정 threshold를 넘기는 값의 비율로 계산
  - ✓ Pooling layer가 없는 ViT는 해당 비율이 일정
  - ✓ Pooling layer가 있는 PiT는 해당 비율이 상승

- PiT에서 spatial interaction 비율이 상승하는 패턴은 CNN 기반의 모델의 패턴과 유사함
  - ✓ Transformer 기반의 모델에서도 pooling layer가 의도한 대로 잘 작동함을 확인



(a) Spatial interaction (over 1%)



(b) Spatial interaction (over 10%)

DMQA

Experiments

❖ Architectures

- 실험에 사용할 ViT와 PiT 모델 크기 별 특징 비교
- 동일 크기라도 PiT가 GPU 연산량을 보다 가볍게 가져가도록 설계된 것을 알 수 있음

Network	Spatial size	# of blocks	# of heads	Channel size	FLOPs
ViT-Ti [34]	14 x 14	12	3	192	1.3B
PiT-Ti	27 x 27	2	2	64	0.7B
	14 x 14	6	4	128	
	7 x 7	4	8	256	
PiT-XS	27 x 27	2	2	96	1.4B
	14 x 14	6	4	192	
	7 x 7	4	8	384	
ViT-S [34]	14 x 14	12	6	384	4.6B
PiT-S	27 x 27	2	3	144	2.9B
	14 x 14	6	6	288	
	7 x 7	4	12	576	
ViT-B [9]	14 x 14	12	12	768	17.6B
PiT-B	31 x 31	3	4	256	12.5B
	16 x 16	6	8	512	
	8 x 8	4	16	1024	

Experiments

❖ Classification

- 동일 크기라도 PiT가 GPU 연산량을 보다 가볍게 가져가도록 설계된 것을 알 수 있음
- 성능 지표는 accuracy

Architecture	FLOPs	# of params	Throughput (imgs/sec)	Vanilla	+CutMix [41]	+DeiT [34]	+Distill [34]
ViT-Ti [34]	1.3 B	5.7 M	2564	68.7%	68.5%	72.2%	74.5%
PiT-Ti	0.7 B	4.9 M	3030	71.3%	72.6%	73.0%	74.6%
PiT-XS	1.4 B	10.6 M	2128	72.4%	76.8%	78.1%	79.1%
ViT-S [34]	4.6 B	22.1 M	980	68.7%	76.5%	79.8%	81.2%
PiT-S	2.9 B	23.5 M	1266	73.3%	79.0%	80.9%	81.9%
ViT-B [9]	17.6 B	86.6 M	303	69.3%	75.3%	81.8%	83.4%
PiT-B	12.5 B	73.8 M	348	76.1%	79.9%	82.0%	84.0%

Experiments

❖ Object Detection

- DETR의 backbone으로서 ResNet50, ViT, PiT를 사용하여 학습한 뒤 성능을 비교
- 성능 지표는 Average Precision (AP)

Backbone	Avg. Precision at IOU			Params.	Throughput
	AP	AP_{50}	AP_{75}		(imgs/sec)
ResNet50	36.7	56.8	38.1	41.3 M	22.8
ViT-S	32.1	52.4	32.3	39.3 M	26.1
PiT-S	34.4	54.7	35.3	40.6 M	26.3

Table 4. **COCO detection performance based on DETR [4]**. We evaluate the performance of PiT as a pretrained backbone for object detection.

Experiments

❖ Robustness

- 분류하려는 객체의 배경이 바뀜에 따라 얼마나 강건한 성능을 보이는지 실험
- 성능 지표는 accuracy
- ViT 모델은 spatial dimension이 일정하기 때문에 다음과 같은 취약점이 있다고 가정
  - ✓ 배경이 바뀌면 동일한 객체라도 분류하는 성능이 떨어질 것
  - ✓ 객체의 전반적인 특징보다 다른 객체와 구분되는 지엽적인 특징에 의존할 것

	Standard	Occ	IN-A [15]	BGC [40]	FGSM [10]
PiT-S	80.8	74.6	21.7	21.0	29.5
ViT-S [34]	79.8	73.0	19.1	17.6	27.2
ResNet50 [13]	76.0	52.2	0.0	22.3	7.1
ResNet50† [38]	79.0	67.1	5.4	32.7	24.7

Table 5. **ImageNet robustness benchmarks.** We compare three comparable architectures, PiT-B, ViT-S, and ResNet50 on various ImageNet robustness benchmarks, including center occlusion (Occ), ImageNet-A (IN-A), background challenge (BGC), and fast sign gradient method (FGSM) attack. We evaluate two ResNet50 models from the official PyTorch repository, and the well-optimized implementation [38], denoted as †.

Conclusion

- CNN 모델 구조의 특징(pooling layer)을 Transformer 모델 구조에 적용한 성공적인 사례
- Pooling layer를 추가함으로써 ViT 대비 연산량은 줄이면서 일반화 성능을 높임
- Transformer 기반의 모델에서도 spatial interaction을 고려해야함을 증명함

Reference

1. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., & Oh, S. J. (2021). Rethinking spatial dimensions of vision transformers. *arXiv preprint arXiv:2103.16302*.


Thank You