Locally enhanced Convolutional Transformer with appropriate Inductive Bias(LeCT)

https://github.com/ysj9909/Vision_Backbone_projects

Abstract

Vision Transformer(ViT)가 소개된 후부터 CNN, MLP 모델들 중에서 어떤 모델이 더 좋은가에 대한 논의는 계속되어왔다. 하지만 논의가 정리되지 않고 계속 연구들이 이어지고 있는 상황이다. 이러한 상황에서 여러 pure model들에 대한 연구들을 보며 느낀 점은 각 모델들이 가지고 있는 서로 다른 강점들이 있다는 것이고 우리가 제안하는 모델은 각 모델의 강점을 적절히 이용한다. 특히 우리 모델은 CNN과 ViT의 강점들이 적절히 혼합되어 있다. 해당 report에서 우리는 Locally enhanced Convolutional Transformer(LeCT)라는 모델을 제안하려 한다. 이 모델에서 사용하는 Locally enhanced Attention 모듈이 기존의 Attention 모듈과 구별되는 강점을 크게 세 가지로 볼 수 있다. 1) query, key, value 로의 projection을 convolution을 이용해서 "***Sequentially***" 계산한다. 이를 통해 지역적으로 더 강화된 representation을 통해서 attention을 수행할 수 있다. 2) projection을 위해 계산된 features가 skip connection을 통해서 attention feature와 연결됨으로써 long-range feature와 local feature를 동시에 활용할 수 있다. 3) 2D 이미지상에서 인접한 픽셀 간의 상관관계가 높다는 inductive bias를 활용함으로써 sequence 길이에 선형적인 computational costs를 갖는 Attention을 수행한다. 우리의 실험환경에서 여러 최근 모델들보다 좋은 성능을 보이는 것을 확인할 수 있다.

1.Introduction

최근 10년동안 CNNs[2, 13]는 Vision Bcakbone으로서 굉장히 좋은 성능을 보이고 있다. 하지만 최근 ViT[1]가 CNN의 성능을 뛰어넘으며 발표되었다. ViT[1]는 weak inductive bias를 가지고 있으며 Multi-head Self Attention(MSA)를 이용해서 long-range feature를 잘 계산할 수 있다. 하지만 ViT의 weak inductive bias때문에 inductive bias를 implicitly 학습하기 위해서 많은 양의 학습 데이터와 더 긴 학습 스케줄링 그리고 여러 data augmentation이 필요했다. 즉, 세심하게 구성된 학습 환경에서는 ViT가 CNNs 보다 더 좋은 성능을 보인다. 이런 면에서 성능에 있어서는 ViTs가 CNNs 보다 upper bound가 더 높다고 할 수 있다. 반면에 CNNs의 경우 작은 크기의 데이터에 대해서도 빠르게 학습해 좋은 성능을 보인다. 따라서 성능의 lower bound가 높다고 할 수 있다. 따라서 우리는 각 모델의 장점을 적절히 이용한 새로운 모델을 제안해서 작은 크기의 데이터에 대해서도 적절한 수준의 inductive bias를 가지고 안정적으로 학습하고 성능의 upper bound 또한 높였다.

우리는 ***ViT의 capacity를 제안하지 않는 선에서 2D 이미지를 다루기에 적절한 inductive bias를 활용한 Attention을 고안할 수 있을까*** 라는 문제의식에서 연구를 시작했다. 우리 모델은 다음의 디자인 규칙들을 따른다. 첫 번째, ViT[1]에서는 single scale features를 계산하는 반면 우리 모델은 multi-scale features를 각 stage에서 계산한다. (이미 선행 연구에서 multi-scale feature information이 single scale에 비해 좋은 성능을 보인다는 것이 밝혀졌다.[12]) 다음 stage로 넘어가기 전에 feature map의 해상도를 줄이기 위해 단순히 non-overlap하게 패치들을 병합하는 것이 아닌 Conv layer를 이용해서 local feature를 강화한다. 두 번째, Locally enhanced Mlp[8, 9]를 Attention 모듈 뒤에 배치함으로써 local feature를 계산한다. 세 번째, Attention 모듈에서 계산되고 있는 토큰 입장에서 중요한 토큰을 attend할 때 중요한 토큰 주변의 토큰들도 비슷한 attention weight를 가지고

feature들이 가중합될 수 있도록 한다.(2D 이미지상에서 인접한 픽셀간의 상관관계가 높다는 정보를 통해 자연스럽게 생각해볼 수 있다.) 그리고 Attention을 위한 Projection(using convolution)[9]에 쓰인 features를 skip connection으로 Attention features와 연결함으로써 high frequency feature[10]를 보완적으로 사용할 수 있다. (해당 모듈을 Locally enhanced Attention, e.g. LeAtt라 칭한다.) LeAtt 안에서 high-frequency feature(local), low-frequency feature(global)를 동시에 다룬다. (discriminative power를 위해서 두 종류의 feature는 모두 필요하다.[18]) 한 레이어에서 두 가지 feature를 모두 다룸으로써 서로 다른 레이어에서 순차적으로 두 feature를 다루는 AlterNet[10]보다 더 효율적이고 성능 또한 더 좋다.(뒤에서 결과를 통해 확인할 수 있다.)하나의 모듈에서 앞의 두 개의 feature 중 한가지만 이용할 경우[6, 16] sub-optimal할 수 밖에 없다. 왜냐하면 local modeling을 하는 레이어에서는 중요한 global feature를 놓칠 수 있고 그 반대의 상황도 일어날 수 있기 때문이다. 또한 LeAtt는 input sequence 길이의 linear computational costs를 갖기 때문에 고해상도 이미지를 다루기 적절하다.

우리 모델의 성능을 비교하기 위해서 제한된 training 환경을 설정했다. 자세한 configuration은 실험 부분에서 확인할 수 있다. 그리고 비교 대상은 ResNet[2], ConvNeXt[13], ViT[1], CeiT[8], AlterNet[10]이다. 똑같은 환경에서 성능을 비교해본 결과 제일 좋은 성능을 보이는 것을 확인할 수 있다.

2.Related Work

Vision Transformer(ViT)[1]는 image classification task에서 pure transformer를 사용해서 SOTA CNN 모델들의 성능보다 더 좋은 성능을 낼 수 있음을 보였다. 그 이후 DeiT[14]에서는 distillation 기법을 도입해 ViT의 성능을 더 높였다. DeiT[14] 이외에도 ViT의 문제점을 보완하기 위한 연구들이 활발하게 이뤄졌다. MSAs는 특히 고해상도 이미지를 다룰 때 sequence 길이의 제곱에 비례하는 Computational costs를 갖기 때문에 적절하지 못하다. 따라서 이런 문제점을 해결하기 위해서 local window내에서 self-attention을 진행하는 방법을 사용하는 모델[5, 11]들이 제안되었다. 이러한 기법들은 기존의 MSAs에서의 Computational Costs문제를 해결하고 인접한 레이어(혹은 서로 다른 head)에서의 window를 이동시켜 receptive size를 확장시킨다. 하지만 local Attention은 long-range interactions을 제한시킨다. 따라서 global MSAs에 비하면 sub-optimal이라고 할 수 있다. 반면에 우리의 모델은 적절한 inductive bias를 갖는 global attention을 통해서 효율적이고(has linear computational costs w.r.t sequence length)효과적이다(global feature를 계산함으로써 성능의 upper bound 또한 높다.).

최근에 Pyramid Vision Transformer(PVT)[4]에서 CNNs에서와 같이 pyramid feature를 계산한다. 그리고 spatial reduction module[4]을 통해서 global attention을 수행한다. 또한 최근 연구들에서 ViT에 convolution의 속성을 첨가[3-12,16]함으로써 학습의 안정성[12]과 성능 향상을 꾀했다. DeiT[14]에서도 CNN teacher model을 사용하는 것을 생각해보면 local feature를 적절히 이용해주는 것은 성능 향상을 위해 필수적이라는 것을 알 수 있다[12]. 따라서 CNN과 ViT가 서로 상호보완적[10]인 특징을 갖고 있다는 것을 이용해서 우리는 ViT[1]에 Convolutional 요소를 적극적으로 삽입한 Attention 기법을 제안한다. 또한 Positional encoding이 제거되어도 성능에 영향이 거의 없다.[15]

MSAs가 여러 Vision tasks에 대해서 좋은 성능을 보이고 있다. 하지만 정작 MSAs가 근본적으로 왜 좋은 성능을 보이는가에 대한 연구는 제대로 이루어지지 않았다. 그러던 와중에 [10]에서는 Convs와 MSAs는 서로 상호보완적인 기능을 한다는 밝혔다. 또한 ViTAE[3] 에서는 ViT[1]의 부족한 intrinsic inductive bias를 보완하기 위해서 다양한 dilation ratio를 가지는 Conv features를 Concat하여 MSAs를 적용하기 전에 local features를 강화한다. 또한 CNN feature를 skip connection으로 연결함으로써 local & multi-scale context 정보를 이용할 수 있다. 우리는 ViTAE의 skip connection에서 더 나아가 convolutional feature들을 attention map을 생성하기 위한 feature를 projection할 때에도 이용해줌으로써 Locally enhanced Attention을 수행할 수 있다. Global feature와 local feature를 적절히 사용해줌으로써 모델이 적절한 Inductive Bias를 갖게 되고 적은 parameter수로도 small

dataset에 대해서 최근 모델들 보다 성능이 좋은 것을 확인할 수 있다.

3.Method

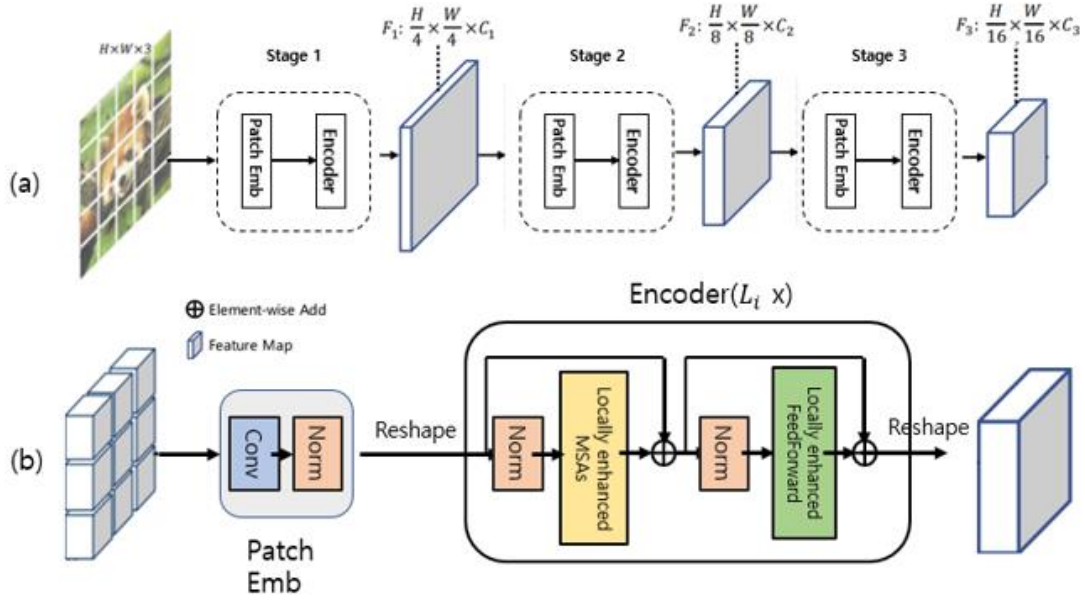

Figure 1: Overall architecture of Locally enhanced Convolutional Transformer(LeCT). (a)는 전체 모델의 개괄적인 구조이다. 이렇게 구한 features를 global average pooling – linear layer를 거쳐 최종 예측을 진행한다.(b)는 각 스테이지의 세부 사항을 보여준다.

3.1 Overall Architecture

우리의 LeCT는 ViT[1]에서 여러 모듈들을 수정해가면서 디자인되었다. 또한 여러 ViTs[1, 6, 8]에서 class token을 이용해서 마지막 classification task를 진행한다. 하지만 우리의 모델은 마지막 features를 spatial 축으로 global average pooling을 한 후 classification을 진행함으로써 성능을 향상시켰다.[10] 먼저 Fig1(a) 는 전체 모델의 개괄적인 pipeline이다. 또한 Fig1(b) 는 각 스테이지가 어떻게 구성되어 있는지 설명한다. 3.2에서 Patch Embedding에 대해서 설명하고, 3.3에서 Locally enhanced MSAs를 설명하고, 마지막으로 3.4에서 Locally enhanced Feed Forward layer[8]를 설명하려 한다.

3.2 Convolutional Patch Embedding

ViT[1]에서는 visual tokens을 만들기 위해서 중복되지 않도록 입력 이미지의 인접한 픽셀들끼리 채널 축으로 Concat operation을 진행해준 뒤 Linear layer를 거친다. 다음과 같이 단순한 토큰화 기법은 인접한 픽셀간의 중요한 정보(경계, 선)를 놓칠 수 있는 단점을 가지고 있다. 따라서 단순한 토큰화 기법 대신에 각 스테이지의 처음에 overlapping convolution operation을 수행하는 Convolutional Patch Embedding layer를 사용한다. 해당 레이어를 통해서 local information을 계산할 수 있고 또한 CNN과 같이 점진적으로 sequence 길이를 줄이고 레이어의 width를 늘릴 수 있다. 첫 번째 스테이지에서만 kernel size를 7로하고 이후에는 kernel size를 3으로 한다.

3.3 Locally enhanced MSAs(LeMSA)

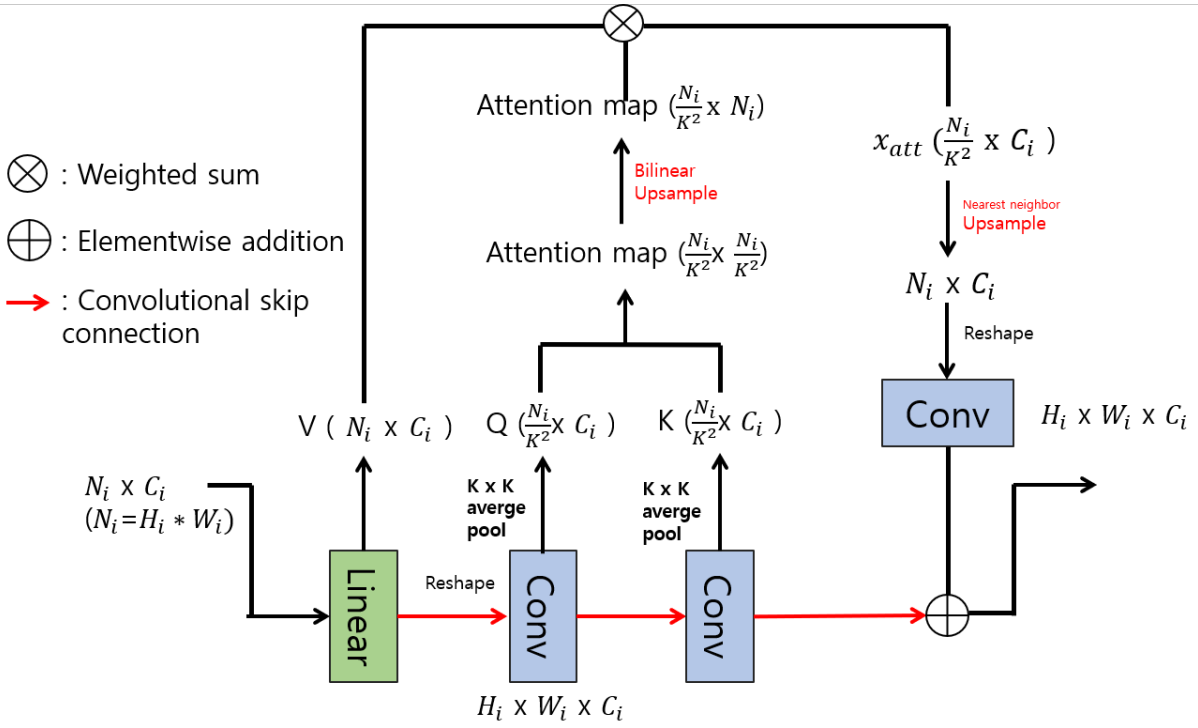

Figure 2: Details of Locally enhanced Multi-head Self Attention. 본문에서의 spatial reduction ratio = K이고 Conv는 Depthwise Convolution이다.Fig의 가독성을 높이기 위해서 Normalization, activation function은 생략했다.(자세한 내용은 아래의 수식을 참고.) Skip connection을 통해서 CNN, MSA features를 동시에 다룬다. 상호보완적인 features를 다루기 때문에 성능을 향상시킬 수 있다.[18]

Global Attention을 수행한다고 했을 때 sequence length의 제곱에 비례하는 Computational costs가 발생한다. 특정 토큰이 모든 토큰에 대한 attention score를 구한다고 했을 때, attention map을 확인해보면 attention weight가 높은 토큰들이 모여 있는 것을 자주 관찰할 수 있다. 이러한 사실을 이용해서 attention score를 더 큰 영역에 대해서 구하고 bilinear interpolation을 통해서 sub-region의 attention score를 근사할 수 있다. 2D 이미지에서 인접한 픽셀 간의 상관관계가 높기 때문에 해당 Attention map을 upsample하더라도 information loss가 적고 Computational cost 문제도 해결할 수 있다.

Weak inductive bias를 갖는 vanilla Attention 모듈에 적절한 inductive bias를 주는 것이 목표이므로 attention map을 upsample하는 것 외에도 추가적으로 최대한 convolution operation을 활용하여 MSAs를 변형시켰다. 먼저 선행 연구[6]에서는 convolutional Projection을 통해서 query(q), key(k), value(v) feature를 생성한다. 이 때, 각 features는 병렬적으로 서로 다른 weight로 계산되어 feature가 구해진다. 하지만 LeMSA는 q, k, v feature들을 *Sequentially* 계산한다.(v, q, k 순서로) 여러 번 계산된 feature를 통해서 q, k를 위한 features를 구하므로 더 강력한 representation을 이용해서 MSAs를 수행할 수 있다.

$$Attention(q,k,v) = softmax\left(\frac{M_{up}}{\sqrt{C}}\right)v \qquad (1)$$

v, q, k를 구하기 위한 feature를 계산하고 skip connection을 위한 Convolutional features를 계산하는 과정은 다음과 같다.(x는 LeMSAs의 input이다.)

$$x_1 = linear(x) \qquad (2)$$

$$x_1 = Reshape(x_1) \qquad (3)$$

$$x_2 = BatchNorm\big(DWConv1(x_1)\big) \qquad (4)$$

$$x_3 = BatchNorm\Big(DWConv2\big(GELU(x_2)\big)\Big) \qquad (5)$$

$$skip = Reshape\big(GELU(x_3)\big) \qquad (6)$$

또한 [4, 6]에서는 key, value를 위한 projection으로 stride 2의 convolution layer를 이용해서 4배 빠른 MSAs를 수행한다. 우리 모델은 value의 spatial 차원을 유지하고 앞선 기법과 다르게 key, query의 spatial 차원을 average pooling layer(stride = K)에서 spatial reduction ratio(= K)만큼 축소시킨다.

$$query = K \times K\ Average\ Pooling(x_2) \qquad (7)$$

$$key = K \times K\ Average\ Pooling(x_3) \qquad (8)$$

$$value = x_1 \qquad (9)$$

Fig2와 (1) 수식을 통해 알 수 있듯이 spatially 축소된 attention weight를 계산함으로써 sequence length의 제곱에 비례하는 computational costs가 발생한다는 문제도 해결할 수 있고 dense prediction도 다룰 수 있게 된다.

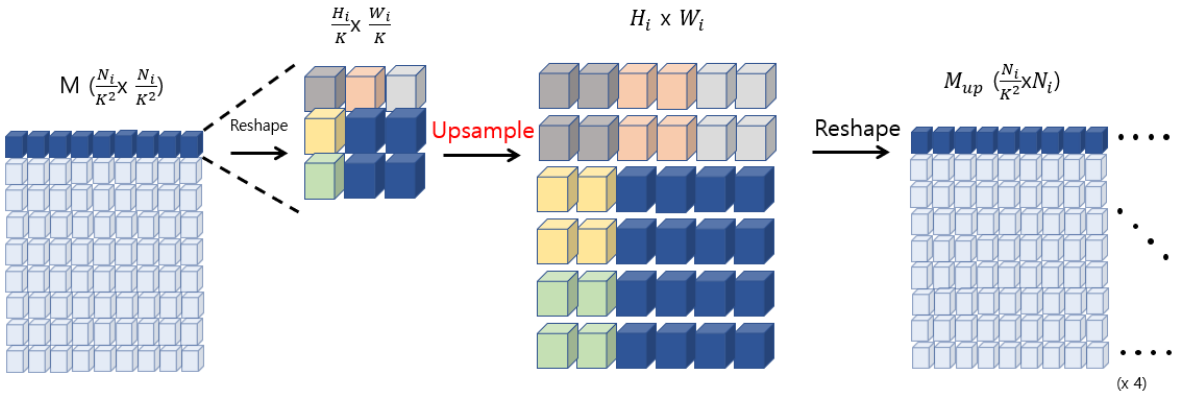

Figure 3: Details of Upsample layer in LeMSAs(Fig2). 진하게 표현된 부분이 특정 토큰이 모든 토큰에 대해 attend하는 정보라 할 때 다른 모든 토큰에 대한 attention energy들의 spatial 정보를 복원하고 bilinear upsample을 적용한다. 마지막으로 다시 flatten layer를 거치고 softmax를 거쳐 attention weights를 구하게 된다.

Bilinear Upsample layer를 통해서 Attention map의 key 축을 input spatial dimension으로 확대시킨다. 그리고 V(value features)와 weighted sum operation을 통해서 최종적으로 self-attention이 적용된features x_{att}를 구한다.

$$M = qk^T \qquad (10)$$

$$M_{up} = Reshape\left(Upsample\big(Reshape(M)\big)\right) \qquad (11)$$

(11)에서 구한 attention map과 v features를 이용해서 attention feature(1)를 구해준다. 그리고 축소되었던 q 축을 원래 sequence length로 upsample하기 위해서 Fig3 에서의 연산을 query 축으로 다시 진행해준다. 이때 attention map을 upsampling하는 과정과 달리 해당 스텝에서는 Depthwise convolution – BatchNorm을 추가해서 local representations을 강화한다. (1)을 통해 구한 global attention feature와 (6) feature(convolutional skip connection)를 원소별로 더해준다.

3.4 Locally enhanced FFNs(LeFFN)

해당 모듈은 CeiT[8]의 Locally enhanced Feed-Forward Network와 동일한 연산을 진행한다.[9](단 유일한 차이점은 input token 중 class token을 우리 모델에서는 사용하지 않기 때문에 따로 split 연산이 필요하지 않다.) 기존의 ViT[1]에서는 단순히 pointwise 연산만 진행했다면 LeFFN의 경우 특정 비율로 채널이 확장된 feature에서 Depthwise convolution을 통해서 local 정보를 더 사용할 수 있도록 한다.(Appendix에 CeiT 논문의 Figure를 통해서 자세한 구조를 확인할 수 있다.) CVPT[7] 에서는 3x3 conv를 conditional PE를 각 해상도에 맞춰서 구하고 이를 feature map에 더하는 방식으로 patch flattening 에서 생기는 위치 정보를 보충한다. 하지만 LeFFN의 3x3 depthwise convolution을 통해서 위치 정보를 처리하기 때문에 기존의 Positional Encoding(PE)

을 제거해도 성능 하락이 없음을 알 수 있다.[15]

4.Experiments

우리의 모델을 ResNet[2]을 포함해서 최근에 발표된 ConvNeXt[13], ViT[1], CeiT[8]와 비교하였다. 특히 CeiT[8]의 경우 Attention module, single-scale feature를 다룬다는 점 제외하고는 모델 구성이 매우 비슷하다. 따라서 우리가 제안한 LeMSAs의 성능을 확인하기 위해서 최대한 모델 구조가 비슷할 수 있도록 CeiT에서 사용하던 class token을 없애고 마지막 feature map에 global average pooling(GAP)-linear layer를 통해서 예측할 수 있도록 모델 구조를 수정해서 비교했다.(GAP 모듈을 통해서 성능이 더 향상된 것을 확인할 수 있었다.[10]) 그리고 각 baseline 모델들은 해당 논문 저자들의 코드를 바탕으로 작성하여 from scratch로 학습했다. 그리고 사용한 데이터셋은 CIFAR10, CIFAR100, STL10을 사용해서 비교했다. 또한 학습을 위한 여러 환경 configuration은 Appendix에서 확인할 수 있다.

비교에 사용되는 데이터셋에 대한 설명은 Table1에서 확인할 수 있다.

Dataset	Classes	train data	val data	img size
CIFAR10	10	50000	10000	32
CIFAR100	100	50000	10000	32
STL10	10	5000	8000	96

Table1. Details of used visual datasets.

CIFAR100

Model	#param	Accuracy
ResNet	1.2M	69.68
ConvNeXt	1.2M	71.84
ViT	1.7M	59.46
CeiT	1.2M	74.98
AlterNet	9.5M	73.16
LeCT	1.2M	**76.53**
LeCTv2	1.1M	74.46
LeCTv3	1.3M	76.08

CIFAR10

Model	#param	Accuracy
ResNet	-	92.83
ConvNeXt	-	93.52
ViT	-	79.53
CeiT	-	93.82
AlterNet	-	-
LeCT	-	**94.41**
LeCTv2	-	-
LeCTv3	-	

STL10

Model	#param	Accuracy
ResNet	-	79.71
ConvNeXt	-	73.95
ViT	-	61.78
CeiT	-	80.53
AlterNet	-	-
LeCT	-	**82.94**
LeCTv2	-	-
LeCTv3	-	81.85

Table 2. Image classification performance on validation set. LeCTv2, v3은 Appendix에서 확인할 수 있다. 모든 데이터셋에 대해서 각 모델의 parameter 개수를 갖게 설정했다. 위의 table의 CeiT[8]은 해당 논문의 코드에서 class token을 사용하는 부분을 Global Average Pooling(GAP)로 대체해서 학습한 결과이다.(더 공정한 비교를 위해서)

4.1 Results on visual datasets(CIFAR, STL10)

Table2 을 보면 일단 우리의 LeCT 모델이 최신 모델들보다 제한된 학습 환경에서 더 좋은 성능을 보이는 것을 확인할 수 있다. 특히 여태까지 Computer vision에서 표준으로 쓰이던 CNNs model(ResNet[2], ConvNeXt[13])보다 모든 작은 크기의 데이터셋(CIFAR10, 100, STL10)에 대해서 좋은 성능을 보이는 것을 확인할 수 있다. 그리고 ViT[1] 는 weak inductive bias로 인해 작은 크기의 데이터셋에 대해서 대체로 안 좋은 성능을 보이는 것을 확인할 수 있다. 또한 CNN과 attention의 hybrid model(CeiT[8], AlterNet[10])들 보다도 좋은 성능을 보이는 것을 확인할 수 있는데 AlterNet[10]이 parameter 개수가 8배 정도 많음에도 우리의

모델이 더 좋은 성능을 보이는 것을 확인할 수 있다. AlterNet의 경우 상호보완적인 필터(CNN, MSA)를 layerwise하게 사용한다. 하지만 하나의 레이어에서 두 필터를 같이 사용해준다면 다양한 상황에서 더 최적일 것이다. 이러한 면에서는 우리의 모델이 AlterNet[10]보다 더 우수하다고 할 수 있다. 또한 CeiT와 우리의 모델의 차이는 attention module, feature-scale이다. 위의 결과를 통해서 LeAtt를 통해 Computational costs 문제를 해결하고 multi-scale feature를 이용해서 우리가 제안한 방법론이 우수함을 확인할 수 있다.

4.2 Ablation Study

Model	CIFAR10	CIFAR100	STL10
LeCT	**94.41**	**76.53**	**82.94**
LeCT (w/o Convolutional skip connection)	92.31	71.68	79.88

Table3. Effect of Convolutional skip connection with sequentially projecting q, k, v.

Convolutional skip connection(+ sequentially projecting q, k, v)의 영향력을 확인하기 위해서 CNN feature를 사용하지 않고 q, k, v를 위한 projection 역시 독립적으로 이루어지도록 모델을 구성해서 학습시킨 결과를 LeCT와 비교한 것을 Table3 에서 확인할 수 있다. Table3를 통해 알 수 있듯이 CNN features와 MSA features를 원소별로 더해줌으로써 두 가지 서로 다른 종류의 정보를 모델이 잘 이용해서 더 좋은 성능을 보이는 것을 알 수 있다. 또한 CNN features를 계산하는 과정에서의 feature를 MSA를 위한 q, k, v projection에 사용함으로써 추가적인 overhead도 무시할 만한 수치이면서 큰 성능 향상을 이룬 것을 확인할 수 있다.

4.3 Fast Convergence

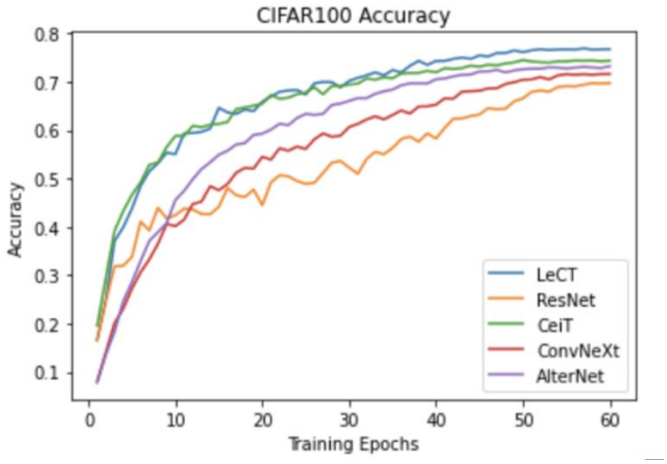

Figure4. Comparisons of the ability of convergence with beselines.

Fig4 를 통해서 우리의 모델 LeCT가 CeiT[8]를 제외한 모든 모델들 보다 acc 60% 를 10 ~ 20 epochs 먼저 달성하는 것을 확인할 수 있다. 우리의 모델은 CeiT[8]와 더불어 MSAs 에 convolution을 적절히 활용한 모델로 다른 비교 모델들 보다 훨씬 빠른 학습 수렴 속도를 보인다.

5.Conclusions

이 리포트를 통해서 우리는 low-level feature를 추출하는 CNNs 과 global feature를 추출하는 MSAs를 적절히 활용하고 적절한 inductive bias를 가지고 기존의 MSAs 의 Computational complexity 문제를 해결한 Locally enhanced MSAs를 통해서 여러 small dataset에 대해서 우수한 성능을 보였다.

CNNs과 MSAs 를 복합적으로 적절하게 활용하는 모델에 대한 연구가 부족한 상황에서 우리는 다양한 해답을 제시한다. 특히 MSAs 에서 관찰되는 계산 비용 문제를 해결하고 CNN features를 적절하게 활용해줌으로써 학습 속도 또한 높였다. 우리의 모듈이 visual transformer 연구에 새로운 방향성을 제시할 수 있을 것이다.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[3] Yufei Xu, Qiming Zhang, Jing Zhang, Dacheng Tao. ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias. In NeurIPS, 2021.

[4] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122, 2021.

[5] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030, 2021.

[6] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. arXiv preprint arXiv:2103.15808, 2021.

[7] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. arXiv preprint arXiv:2102.10882, 2021.

[8] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu and Wei Wu. Incorporating Convolution Designs into Visual Transformers. In ICCV, 2021.

[9] Jianyuan Guo, Kai Han , Han Wu , Chang Xu , Yehui Tang , Chunjing Xu , Yunhe Wang. CMT: Convolutional Neural Networks Meet Vision Transformers. arXiv preprint arXiv:2107.06263, 2021.

[10] Namuk Park, Songkuk Kim. HOW DO VISION TRANSFORMERS WORK?., In ICLR, 2022.

[11] Xiaoyi Dong, Jianmin Bao , Dongdong Chen , Weiming Zhang , Nenghai Yu , Lu Yuan , Dong

Chen , Baining Guo. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. In CVPR, 2021.

[12] Yucheng Zhao, Guangting Wang, Chuanxin Tang, Chong Luo, Wenjun Zeng, Zheng-Jun Zha. A Battle of Network Structures: An Empirical Study of CNN, Transformer, and MLP. arXiv preprint arXiv:2108.13002, 2021.

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie. A ConvNet for the 2020s. In CVPR, 2022.

[14] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. arXiv preprint arXiv:2012.12877, 2020.

[15] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In NeurIPS, 2021.

[16] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, Ross Girshick. Early Convolutions Help Transformers See Better. In NeurIPS, 2021.

[17] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, Alexey Dosovitskiy. Do Vision Transformers See Like Convolutional Neural Networks?. In NeurIPS, 2021.

[18] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, Shuicheng Yan. Inception Transformer. arXiv preprint arXiv:2205.12956, 2022.

Appendix

Updating...