Incorporating Convolution Designs into Visual Transformers

Proceeding in ICCV 2021
SenseTime Research & NTU

Abstract

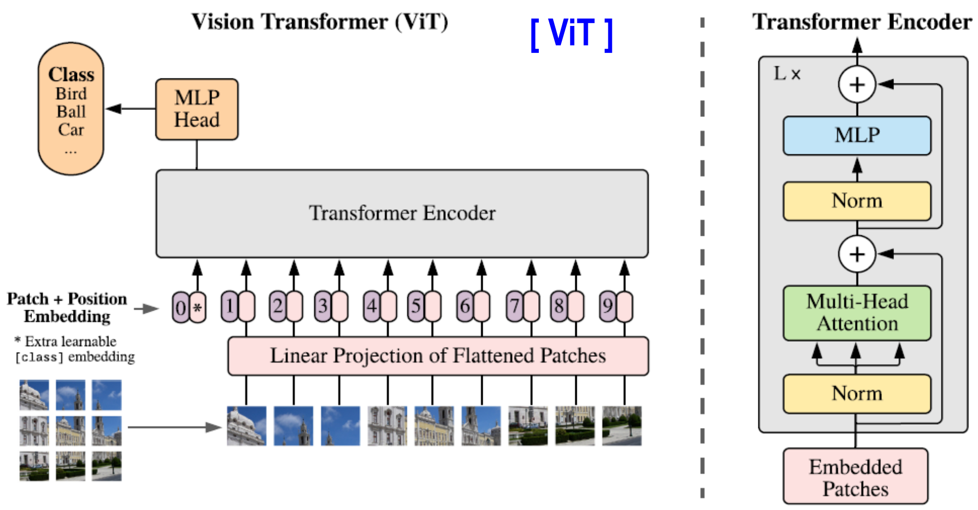
- Transformer in vision task
  - CNNs : extracting low-level features, strengthening locality
  - Transformer : long-range dependency
  - disadvantage : using large training data or extra supervision

- Convolution-enhanced image Transformer (CeiT)
  1) Image-to-Tokens (I2T) module
     - original transformers : tokenization from raw input image
     - CeiT : extract patch from generated low-level features
  2) Locally–enhanced Feed-Forward (LeFF) layer
     - original transformers: encoder blocks
     - CeiT : correlation among neighboring tokens in the spatial dimension
  3) Layer-wise Class token Attention (LCA)
     : multi-level representation

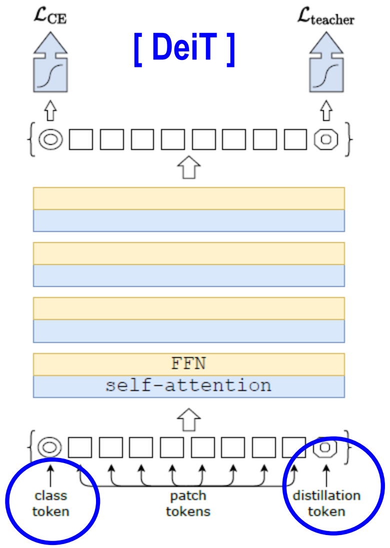
ImageNet & 7-downstream task s show effectiveness and generalization ability

Introduction

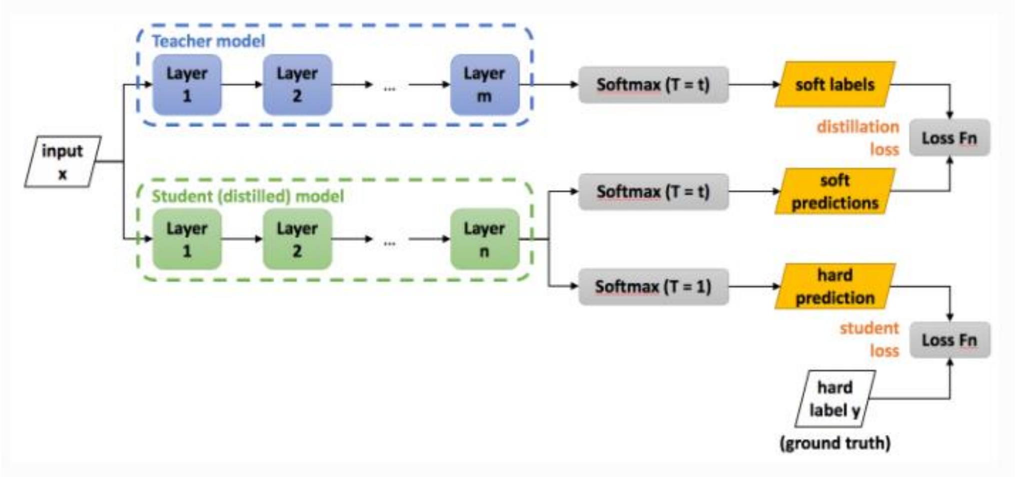
[ViT]



[DeiT]



[Knowledge Distillation]



Model	Architecture	Training Data	Weak point
ViT (Vision Transformer)	first pure transformer architecture - (image patch + position embedding) + class token - No convolution - pre-training + fine-tuning	JFT-300M : large amount of dataset	Limited computing resource or labeled training data ~ 10M training samples: underperforms CNNs ⇒ Transformers lack of the inductive biases inherent to CNNs
DeiT (Data-efficient image Transformers)	teacher : CNN student : ViT - knowledge distillation - class token + distillation token	ImageNet-1k	computation burden engineering issue : teacher model, distillation type CNN teacher better performance than transformer ⇒ the inductive bias inherited by the transformer through distillation

Introduction

Convolution

1. Translation invariance
- weight sharing mechanism과 관련
- geometry and topology information capture

2. Locality
- neighboring pixel tend to be correlated

Transformer

1. Tokenization of patches (@ViT)
- difficult to extract low-level feature

2. Self-attention Module
- building long-range dependence
- ignoring the locality in the spatial dimension

CeiT

1. I2T (image to tokens)
- extract patch from generated low-level features

2. LeFF (Locally-enhanced Feed-Forward)
- promote the correlation among neighboring tokens in the spatial dimension

3. LCA (Layer-wise Class token Attention)
- exploit the ability of self-attention to improve the final representation

Introduction

- **Contribution**

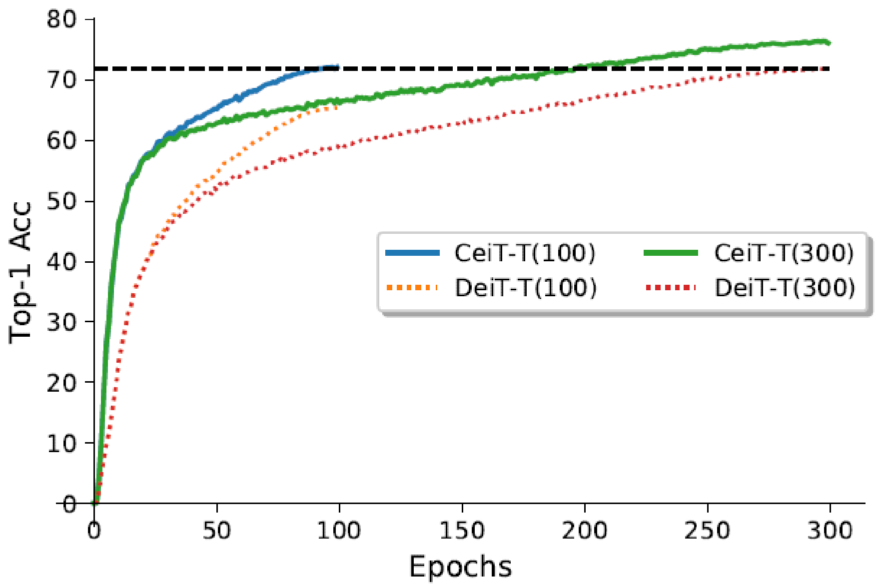
  1) Advantage of CNNs + Transformers

  2) Experimental result on ImageNet + 7 downstream task

  : show effectiveness & generalization ability

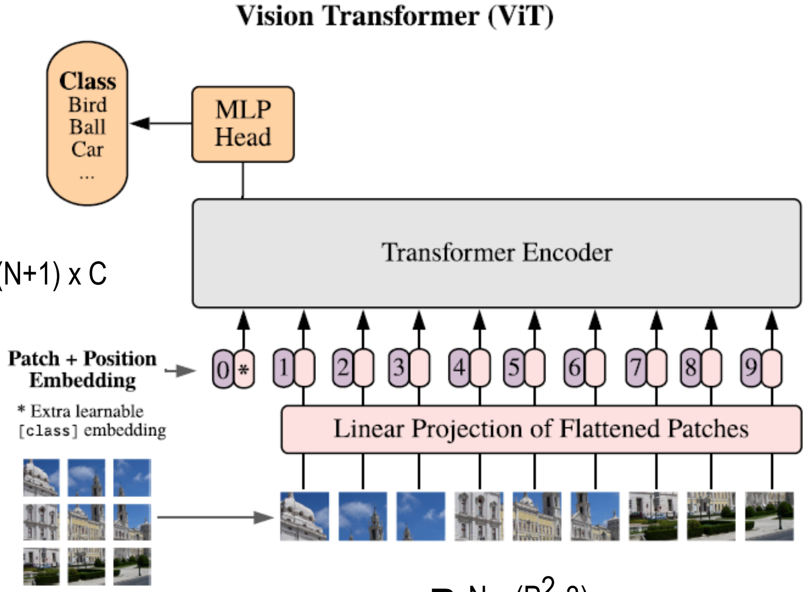
Group	Model	FLOPs (G)	Params (M)	input size	ImageNet Top-1	ImageNet Top-5	Real Top-1
CNNs	ResNet-18 [12]	1.8	11.7	224	70.3	86.7	77.3
	ResNet-50 [12]	4.1	25.6	224	76.7	93.3	82.5
	ResNet-101 [12]	7.8	44.5	224	78.3	94.1	83.7
	ResNet-152 [12]	11.5	60.2	224	78.9	94.4	84.1
Transformers	CeiT-T	1.2	6.4	224	**76.4**	93.4	83.6
	CeiT-S	4.5	24.2	224	**82.0**	95.9	87.3

3) fast convergence than pure Transformer models



Method / Vision Transformer (ViT)

[Tokenization]



$x_t \in R^{(N+1) \times C}$

$x \in R^{H \times W \times 3}$

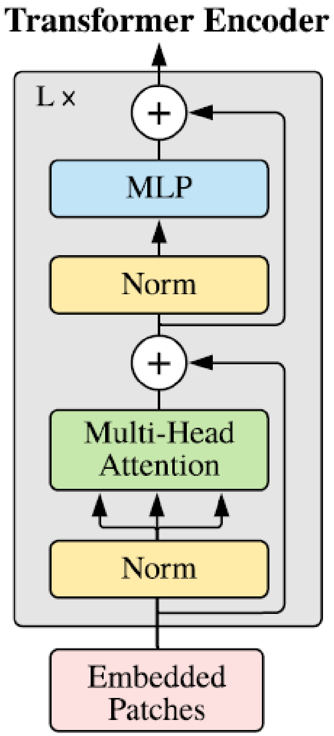
$x_p \in R^{N \times (P^2 \cdot 3)}$

1) Image를 patch 단위로 잘라서 flatten 후 linear projection 하여 encoder 에 feeding
   $N = HW / P^2$ (N : # of patch)

2) Flatted and mapped to latent embedding + position embedding

3) Add class token

[Transformer Encorder]



$$\mathbf{y} = \mathrm{LN}(\mathbf{x}' + \mathrm{FFN}(\mathbf{x}')), \text{ and } \mathbf{x}' = \mathrm{LN}(\mathbf{x} + \mathrm{MSA}(\mathbf{x}))$$

FFN $\mathrm{FFN}(\mathbf{x}) = \sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$

→ 2번의 선형변환 with GELU

MSA $\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{QK}^T}{\sqrt{C}})\mathbf{V}$

→ token 사이의 유사도 계산을 통해 long-range & global attention 을 얻음

LN Layer Normalization is applied before every layer

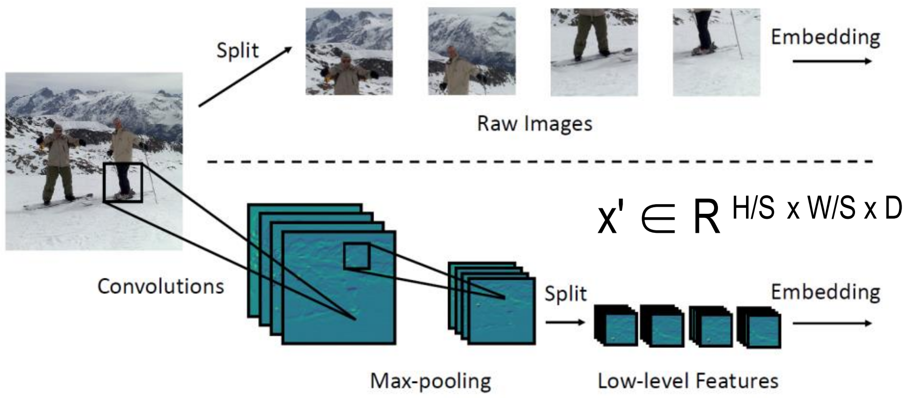
1) 각 encoder는 MSA와 FFN layer 구성되어 있으며, 각 layer 전에 LN 수행

2) MSA : token 사이의 유사도를 계산하여 long-range & global attention 을 얻음

3) FFN : 차원 변환 및 비선형 변환을 통해 token의 representation ability 향상됨

Method / I2T module on CeiT

[I2T w/ Low-level Feature]



$x' \in R^{H/S \times W/S \times D}$

$$\mathbf{x}' = \mathrm{I2T}(\mathbf{x}) = \mathrm{MaxPool}(\mathrm{BN}(\mathrm{Conv}(\mathbf{x})))$$

- Patch의 resolution을 줄여 ViT의 token 수를 맞춤으로써, training difficulty를 줄임

- low-level feature를 이용하으로써 ViT 의 hybrid type의 모델에 비해 light (cf. ResNet last 2-stage 이용함)

▪ Ablation study

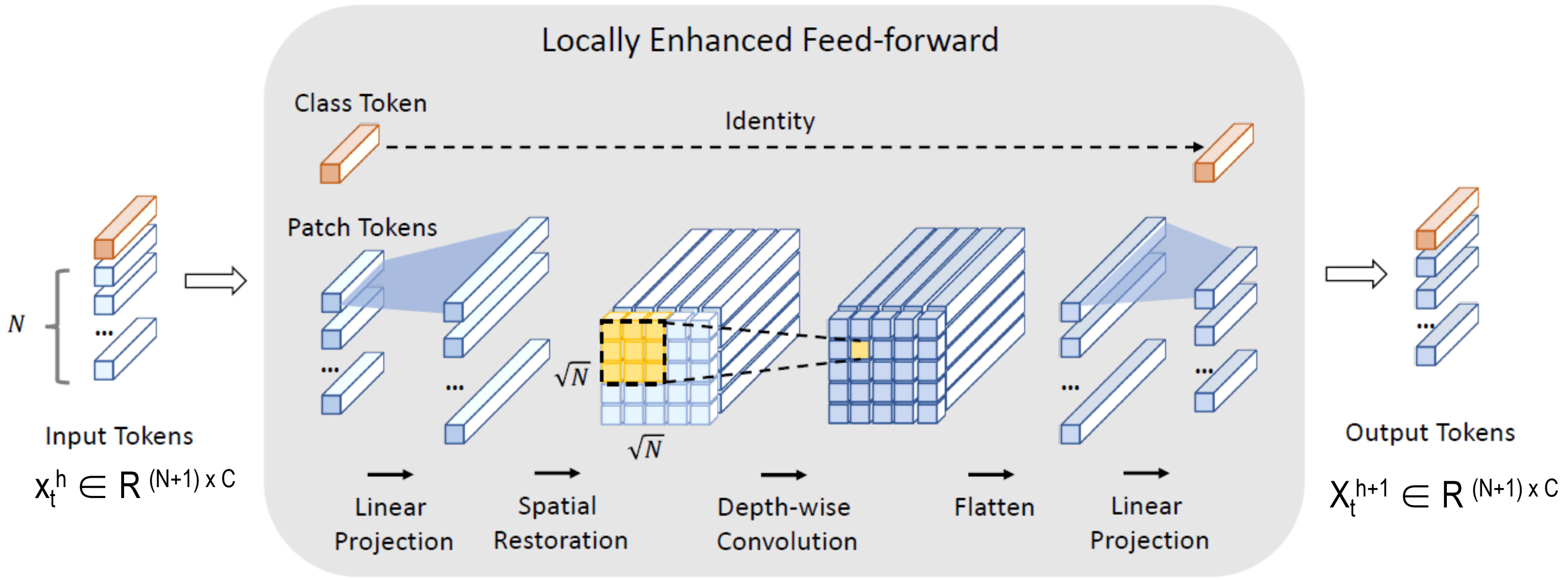
I2T Type				Top-1
conv	maxpool	BN	channels	
✗	✗	✗	3	72.2
$k7s4$	✗	✗	64	71.4 (-0.8)
$k5s4$	✗	✗	64	71.1 (-1.1)
$k3s2 + k3s2$	✗	✗	64	70.4 (-1.8)
$k7s2$	$k3s2$	✗	32	72.9 (+0.7)
$k7s2$	$k3s2$	✓	32	73.4 (+1.2)

- Factor in I2T module

: kernel size & max-pooling & Batch-Norm

Method / LeFF module on CeiT

[LeFF]



$$
\begin{aligned}
\mathbf{x}_c^h, \mathbf{x}_p^h &= \mathrm{Split}(\mathbf{x}_t^h) \\
\mathbf{x}_p^{l_1} &= \mathrm{GELU}(\mathrm{BN}(\mathrm{Linear1}(\mathbf{x}_p^h))) \\
\mathbf{x}_p^s &= \mathrm{SpatialRestore}(\mathbf{x}_p^{l_1}) \\
\mathbf{x}_p^d &= \mathrm{GELU}(\mathrm{BN}(\mathrm{DWConv}(\mathbf{x}_p^s))) \\
\mathbf{x}_p^f &= \mathrm{Flatten}(\mathbf{x}_p^d) \\
\mathbf{x}_p^{l_2} &= \mathrm{GELU}(\mathrm{BN}(\mathrm{Linear2}(\mathbf{x}_p^f))) \\
\mathbf{x}_t^{h+1} &= \mathrm{Concat}(\mathbf{x}_c^h, \mathbf{x}_p^{l_2})
\end{aligned}
$$

3) Spatial Restoration
: original image 위치를 기반으로 image 로 복원

4) Depth-wise Conv.
: patch token 으로 복원

1) split
: class와 patch token 분리함

5) Flatten

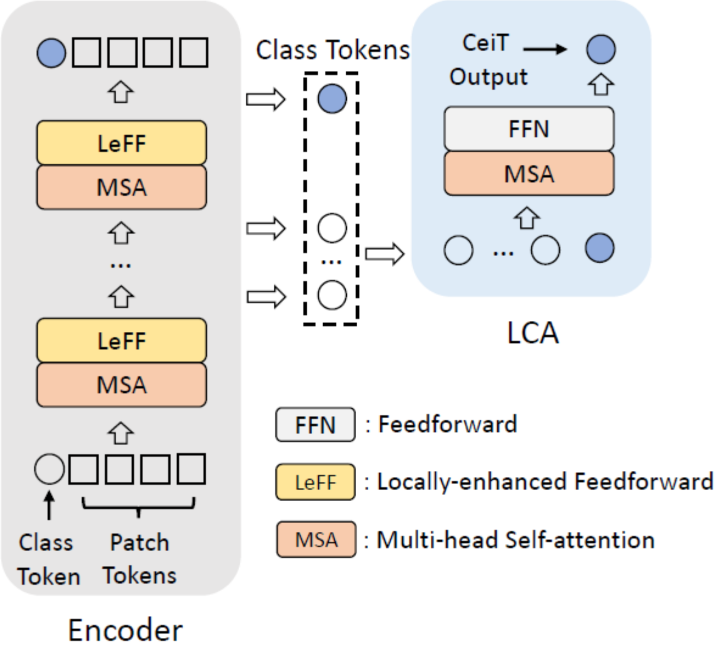
7) Concatenate
: class token과 concatenate 하여 X_t^{h+1}

2) Linear Projection
: Expanding embedding

6) Linear Projection
: initial dimension으로 축소

Method/ Ablation Study

[Layer-wise Class-token Attention]



- 목적 : 다른 layer 의 정보를 합치기 위해서
  - -. CNN : deep 할수록 receptive field 증가
  - -. ViT : deep 할수록 attention distance 증가

- 단방향의 유사성만 계산함

[Ablation Study]

- **LeFF Module**
  - depth-wise convolution kernel size
    : patch token의 correlation 설정하는 영역 크기를 결정
    : kernel 크기가 클수록 accuracy ↑
  - Batch Norm
    : w/BN accuracy 개선함

LeFF Type kernel size	BN	Top-1
✗	✗	72.2
1 × 1	✗	70.3 (-1.9)
3 × 3	✗	72.7 (+0.5)
5 × 5	✗	73.1 (+0.9)
3 × 3	✓	74.3 (+2.1)
5 × 5	✓	74.4 (+2.2)

- **LCA**
  - Adopting LCA, 72.2% → 72.8%

Experimental Setting

[Network architecture]

Model	I2T conv	I2T maxpool	channels	encoder blocks	embedding dimension	heads	LeFF e	LeFF k	Params (M)	FLOPs (G)
CeiT-T	$k7s2$	$k3s2$	32	12	192	3	4	3	6.4	1.2
CeiT-S	$k7s2$	$k3s2$	32	12	384	6	4	3	24.2	4.5
CeiT-B	$k7s2$	$k3s2$	32	12	768	12	4	3	86.6	17.4

1) I2T module : conv layer + BatchNorm + Maxpooling
   → input imgae 대비 4배 작은 feature map

2) Encoder block : 12개

3) LeFF
   - expand ratio e=4
   - depth-wise conv. : 3x3

[Dataset]

Task	dataset	input size	Epochs	batch size	learning rate	LR scheduler	warmup epoch	weight decay	repeated aug [15]
training	ImageNet	224	300	1024	1e-3	cosine	5	0.05	✓
fine-tuning	ImageNet	384	30	1024	5e-6	constant	0	1e-8	✓
transferring	downstream	224&384	100	512	5e-4	cosine	2	1e-8	✗

dataset	classes	train data	val data
ImageNet	1000	1,281,167	50000
iNaturalist2018	8142	437513	24426
iNaturelist2019	1010	265240	3003
Standford Cars	196	8133	8041
Oxford-102 Followers	102	2040	6149
Oxford-IIIT-Pets	37	3680	3669
CIFAR100	100	50000	10000
CIFAR10	10	50000	10000

Result

Group	Model	FLOPs (G)	Params (M)	input size	ImageNet Top-1	ImageNet Top-5	Real Top-1
CNNs	ResNet-18 [12]	1.8	11.7	224	70.3	86.7	77.3
	ResNet-50 [12]	4.1	25.6	224	76.7	93.3	82.5
	ResNet-101 [12]	7.8	44.5	224	78.3	94.1	83.7
	ResNet-152 [12]	11.5	60.2	224	78.9	94.4	84.1
	EfficientNet-B0 [35]	0.4	5.3	224	77.1	93.3	83.5
	EfficientNet-B1 [35]	0.7	7.8	240	79.1	94.4	84.9
	EfficientNet-B2 [35]	1.0	9.1	260	80.1	94.9	85.9
	EfficientNet-B3 [35]	1.8	12.2	300	81.6	95.7	86.8
	EfficientNet-B4 [35]	4.4	19.3	380	82.9	96.4	88.0
	RegNetY-4GF [30]	4.0	20.6	224	80.0	-	86.4
	RegNetY-8GF [30]	8.0	39.2	224	81.7	-	87.4
Transformers	ViT-B/16 [10]	18.7	86.5	384	77.9	-	-
	ViT-L/16 [10]	65.8	304.33	384	76.5	-	-
	DeiT-T [36]	1.2	5.7	224	72.2	91.1	80.6
	DeiT-S [36]	4.5	22.1	224	79.9	95.0	85.7
	DeiT-B [36]	17.3	86.6	224	81.8	95.6	86.7
	DeiT-T + Teacher [36]	1.2	5.7	224	74.5	91.9	82.1
	DeiT-S + Teacher [36]	4.5	22.1	224	81.2	95.4	86.8
	DeiT-B↑384 [36]	52.8	86.6	384	83.1	96.2	87.7
	T2T-ViT-14 [47]	5.2	21.5	224	81.5	-	-
	T2T-ViT-19 [47]	8.9	39.2	224	81.9	-	-
	T2T-ViT-24 [47]	14.1	64.1	224	82.3	-	-
	PVT-T [40]	1.9	13.2	224	75.1	-	-
	PVT-S [40]	3.8	24.5	224	79.8	-	-
	PVT-M [40]	6.7	44.2	224	81.2	-	-
	PVT-L [40]	9.8	61.4	224	81.7	-	-
	CeiT-T	1.2	6.4	224	**76.4**	93.4	83.6
	CeiT-S	4.5	24.2	224	**82.0**	95.9	87.3
	CeiT-T↑384	3.6	6.4	384	**78.8**	94.7	85.6
	CeiT-S↑384	12.9	24.2	384	**83.3**	96.5	88.1

1) CeiT vs CNN

- ResNet-50 vs CeiT-S

: similar size & 5.3% higher performance

- EfficientNet-B4 vs CeiT-S 384

: similar performance

2) CeiT vs ViT

- ViT-L/16 vs CeiT-T

: 1/5 size & 성능이 유사함

3) CeiT vs DeiT

- CeiT-S vs DeiT-T&-B

- CeiT-S vs DeiT-Teacher

: computation cost efficient

: w/o additional CNN model

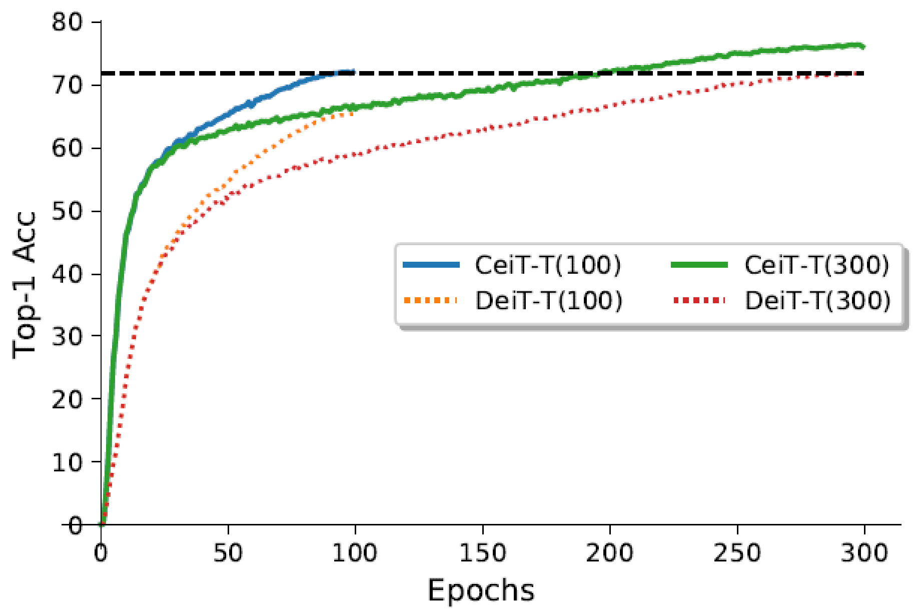
Result / Transfer learning

Model	FLOPs	ImageNet	iNat18	iNat19	Cars	Followers	Pets	CIFAR10	CIFAR100
Grafit ResNet-50 [37]	4.1G	79.6	69.8	75.9	92.5	98.2	-	-	-
Grafit RegNetY-8GF [37]	8.0G	-	76.8	80.0	94.0	**99.0**	-	-	-
EfficientNet-B5 [35]	10.3G	83.6	-	-	-	98.5	-	98.1	**91.1**
EfficientNet-B7 [35]	37.3G	**84.3**	-	-	**94.7**	**98.8**	-	98.9	**91.7**
ViT-B/16 [10]	18.7G	77.9	-	-	-	89.5	93.8	98.1	87.1
ViT-L/16 [10]	65.8G	76.5	-	-	-	89.7	93.6	97.9	86.4
Deit-B [36]	17.3G	81.8	73.2	77.7	92.1	98.4	-	**99.1**	90.8
Deit-B↑384 [36]	52.8G	83.1	**79.5**	**81.4**	93.3	98.5	-	**99.1**	90.8
CeiT-T	1.2G	76.4	64.3	72.8	90.5	96.9	93.8	98.5	88.4
CeiT-T↑384	3.6G	78.8	72.2	77.9	93.0	97.8	94.5	98.5	88.0
CeiT-S	4.5G	82.0	73.3	78.9	93.2	98.2	**94.6**	99.0	90.8
CeiT-S↑384	12.9G	**83.3**	**79.4**	**82.7**	**94.1**	98.6	**94.9**	**99.1**	90.8

⇒ pre-trained CeiT model can be showing strong potential of visual Transformer

Result / Fast Convergence

3×	Top-1	1×	Top-1	1×	Top-1
DeiT-T	72.2	DeiT-T	65.3	CeiT-T	72.2 (+6.9)
DeiT-S	79.9	DeiT-S	74.5	CeiT-S	78.9 (+4.4)
DeiT-B	81.8	DeiT-B	76.8	CeiT-B	81.8 (+5.0)



⇒ DeiT model 대비, 3배 적은 epoch 에서 동등한 성능 구현