Vision AI

2021 arXiv Trends

2021-12

hutom

no.	Paper Title	Correspondence	h-index
1	Benchmarking Detection Transfer Learning with Vision Transformers	Ross Girshick	74
2	Object-Aware Cropping for Self-Supervised Learning	Dilip Krishnan	30
3	MOBILEVIT: Light-weight, general-purpose, and mobile-friendly vision transformer	Mohammad Rastegari	23
4	Florence: A New Foundation Model for Computer Vision	Lu Yuan	36

https://arxiv.org/pdf/2111.11429.pdf

Benchmarking Detection Transfer Learning with Vision Transformers

Yanghao Li Saining Xie Xinlei Chen Piotr Dollár Kaiming He Ross Girshick

Facebook AI Research (FAIR)

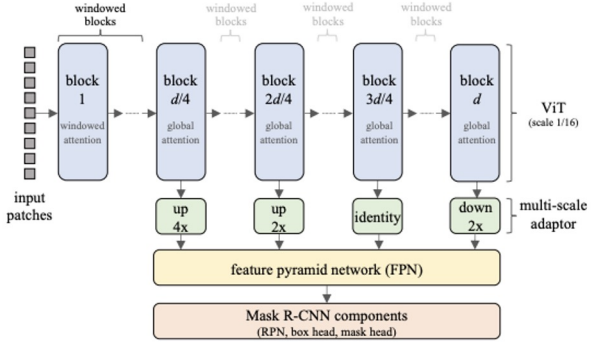


Figure 1. **ViT-based Mask R-CNN.** In §2 we describe how a standard ViT model can be used effectively as the backbone in Mask R-CNN. To save time and memory, we modify the ViT to use non-overlapping windowed attention in all but four of its Transformer blocks, spaced at an interval of $d/4$, where d is the total number of blocks (blue) [26]. To adapt the single-scale ViT to the multi-scale FPN (yellow), we make use of upsampling and downsampling modules (green) [11]. The rest of the system (light red) uses upgraded, but standard, Mask R-CNN components.

<Abstract>

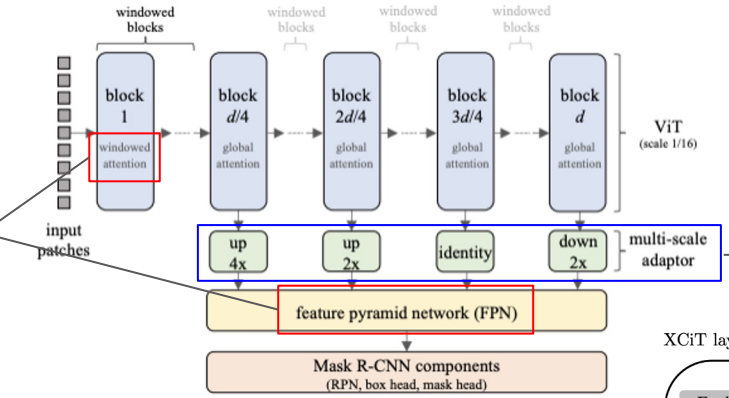
- 아키텍처 비호환성, 느린 훈련, 높은 메모리 소비, unknown training formulae등으로 인해 standard ViT model을 통한 transfer learning을 벤치마킹하지 못했다면
- 본 논문에서는 이러한 과제를 극복하는 훈련 기술을 제시하여 표준 ViT 모델을 Mask R-CNN의 백본으로 사용할 수 있도록 하겠다
- **we compare five ViT initializations, including recent state-of-the-art self-supervised learning methods, supervised initialization, and a strong random initialization baseline**
- Our results show that recent masking-based unsupervised learning methods may, for the first time, provide convincing transfer learning improvements on COCO, increasing APbox up to 4% (absolute) over supervised and prior self-supervised pre-training methods.
- 주제는 오히려 vit로 object detection을 하는 것 자체가 좋더라보다도 mask prediction 기반 방법(BEiT, MAE)들이 object detection에 transfer 됐을 때 supervised pretraining이나 moco 같은 contrastive pretraining에 비해서 성능이 좋더라

https://arxiv.org/pdf/2111.11429v1.pdf

ViT Backbone

1. FPN Compatibility

block1: stride-two 2×2 transposed convolution, and finally another stride-two 2×2 transposed convolution
block d/4: single stride-two 2 × 2 transposed convolution (without normalization and non linearity)
각 모듈은 ViT의 embedding/channel dimension를 보존



window attention을 사용하고 fpn 붙임

result

FPN	AP^{box} ViT-B	ViT-L
yes	**50.1**	**53.3**
no	48.4	52.0

Table 2. **Single-scale *vs*. multi-scale (FPN) ablation.** FPN yields consistent improvements. Our default setting is marked in gray.

ViT는 일반적인 CNN에서 생성된 다중 축척 특징 맵과 달리 단일 축척(예: 1/16)
⇒ 해결: single-scale XCiT backbone의 기술을 사용하여 4개의 resolution-mo
여 중간 ViT feature map을 upsampling 또는 downsampling (d는 총 블록 수)

패치 크기가 16이라고 가정하면 이 모듈은 FPN에 input으로 4, 8, 16, 32 픽셀의 입력 이미지를 사용ㅎ

Swin 및 MViT와 같은 최근 연구가 핵심 ViT architecture (in pretraining)를 수정
해결

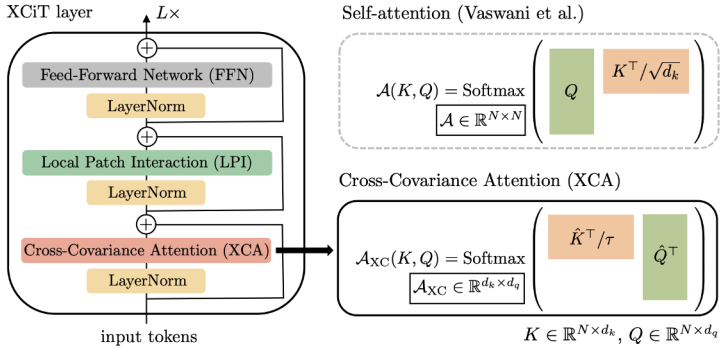


Figure 1: Our XCiT layer consists of three main blocks, each preceded by LayerNorm and followed by a residual connection: (i) the core cross-covariance attention (XCA) operation, (ii) the local patch interaction (LPI) module, and (iii) a feed-forward network (FFN). By transposing the query-key interaction, the computational complexity of XCA is linear in the number of data elements N, rather than quadratic as in conventional self-attention.

ViT Backbone

2. Reducing Memory and Time Complexity

ViT를 Mask R-CNN의 백본으로 사용하면 메모리 및 런타임 문제가 발생

- self-attention operation in ViT에는 image tiled(or "patchified")이 겹치지 않는 h x $O(h^2w^2)$개 만큼의 공간과 시간이 소요

To reduce space and time complexity we use restricted (or "windowed") self-attention

- **replacing global computation with local computation**
- h × w 패치화된 이미지를 r $O(r^2hw)$h non-overlapping windows으로 분할하고 이 $O(r^4)$ indow 내에서 $h/r \times w/r$ 로 self-attention을 계산
- windowed self-attention는 time complexity 가짐(per-window complexity: , window수:
  - r = the global self-attention size used in pre-training
- windowed self-attention 단점: 백본이 윈도우 간에 정보를 통합하지 않는다는 것 ⇒ Therefore we adopt the hybrid approach from that includes four global self-attention blocks placed evenly at each d/4th block (앞 슬라이드 그림의 파란 블록)

self-attention	act checkpt	AP^{box}	train mem	train time	test time
(1) windowed	no	50.7	16GB	**0.67s**	**0.34s**
(2) windowed, 4 global	no	**53.3**	27GB	0.93s	0.40s
(3) global	yes	53.1	**14GB**	2.26s	0.65s
(4) global	no	-	OOM	-	-

Table 3. **Memory and time reduction strategies.** We compare methods for reducing memory and time when using ViT-L in Mask R-CNN. The strategies include: (1) replace *all* global self-attention with 14 × 14 non-overlapping windowed self-attention, (2) a hybrid that uses both windowed and global self-attention, or (3) all global attention with activation checkpointing. Without any of these strategies (row 4) an out-of-memory (OOM) error prevents training. We report AP^{box}, peak GPU training memory, average per-iteration training time, and average per-image inference time using NVIDIA V100-32GB GPUs. The per-GPU batch size is 1. Our defaults (row 2) achieves a good balance between memory, time, and AP^{box} metrics. In fact, our hybrid approach achieves comparable AP^{box} to full global attention, while being much faster.

https://arxiv.org/pdf/2111.11429v1.pdf

Upgraded Modules: MAE에서의 원본 Mask R-CNN과 비교하여, 몇 개의 모듈을 modernize

(1) following the convolutions in FPN with batch normalization (BN)
(2) using two convolutional layers in the region proposal network (RPN) instead of one
(3) using four convolutional layers with BN followed by one linear layer for the region-of-interest (RoI) classification and box regression head
instead of a two-layer MLP without normalization, (4) and following the convolutions in the standard mask head with BN.

Training Formula

기존 Mask R-CNN에 비해 업그레이드된 교육 공식을 채택
목표: hyperparameters 수를 낮게 유지하여 추가 데이터 확대 및 정규화 기술을 채택하지 않는
그러나 drop path regularization가 ViT 백본에 매우 효과적이라는 것을 발견하여 포함

In summary, we train all models with the same simple formula: LSJ (1024×1024 resolution, scale range $[0.1, 2.0]$), AdamW [30] ($\beta_1, \beta_2 = 0.9, 0.999$) with half-period cosine learning rate decay, linear warmup [15] for 0.25 epochs, and drop path regularization.

https://arxiv.org/pdf/2111.11429v1.pdf

3. Initialization Methods

We compare five initialization methods, which we briefly summarize below.

Random: All network weights are randomly initialized and no pre-training is used. The ViT backbone initialization follows the code of [1] and the Mask R-CNN initialization uses the defaults in Detectron2 [40].

Supervised: The ViT backbone is pre-trained for supervised classification using ImageNet-1k images *and* labels. We use the DeiT released weights [36] for ViT-B and the ViT-L weights from [16], which uses an even stronger training formula than DeiT to avoid overfitting (moreover, the DeiT release does not include ViT-L). ViT-B and ViT-L were pre-trained for 300 and 200 epochs, respectively.

MoCo v3: We use the unsupervised ImageNet-1k pre-trained ViT-B and ViT-L weights from the authors of [7] (ViT-B is public; ViT-L was provided via private communication). These models were pre-trained for 300 epochs.

BEiT: Since ImageNet-1k pre-trained weights are not available, we use the official BEiT code release [1] to train ViT-B and ViT-L ourselves for 800 epochs (the default training length used in [1]) on unsupervised ImageNet-1k.

MAE: We use the ViT-B and ViT-L weights pre-trained on unsupervised ImageNet-1k from the authors of [16]. These models were pre-trained for 1600 epochs using *normalized* pixels as the target.

initialization	pre-training data	AP^{box} ViT-B	AP^{box} ViT-L	AP^{mask} ViT-B	AP^{mask} ViT-L
supervised	IN1k w/ labels	47.9	49.3	42.9	43.9
random	*none*	48.9	50.7	43.6	44.9
MoCo v3	IN1k	47.9	49.3	42.7	44.0
BEiT	IN1k+DALL·E	49.8	**53.3**	44.4	47.1
MAE	IN1k	**50.3**	**53.3**	**44.9**	**47.2**

Table 1. **COCO object detection and instance segmentation** using our ViT-based Mask R-CNN baseline. Results are reported on COCO 2017 `val` using the best schedule length (see Figure 2). Random initialization does not use any pre-training data, supervised initialization uses IN1k *with* labels, and all other initializations use IN1k *without* labels. Additionally, BEiT uses a dVAE trained on the proprietary DALL·E dataset of ~250M images [32].

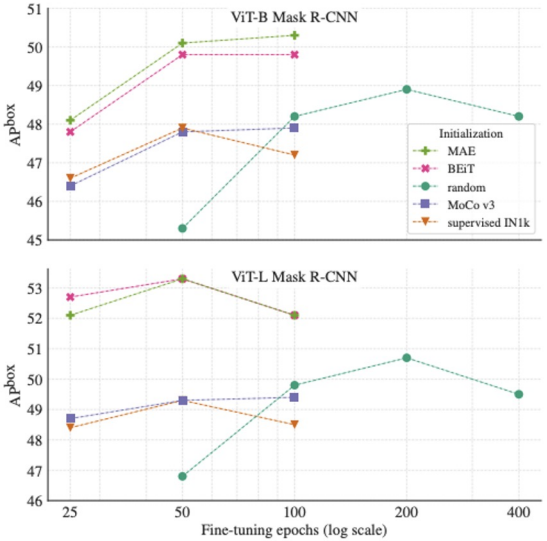


Figure 2. **Impact of fine-tuning epochs.** Convergence plots for fine-tuning from 25 and 400 epochs on COCO. All pre-trained initializations converge much faster (~4×) compared to random initialization, though they achieve varied peak AP^{box}. The performance gap between the masking-based methods (MAE and BEiT) and all others is visually evident. When increasing model scale from ViT-B (top) to ViT-L (bottom), this gap also increases, suggesting that these methods may have superior scaling properties.

MOBILEVIT: LIGHT-WEIGHT, GENERAL-PURPOSE, AND MOBILE-FRIENDLY VISION TRANSFORMER

Sachin Mehta
Apple

Mohammad Rastegari
Apple

https://arxiv.org/pdf/2110.02178v1.pdf

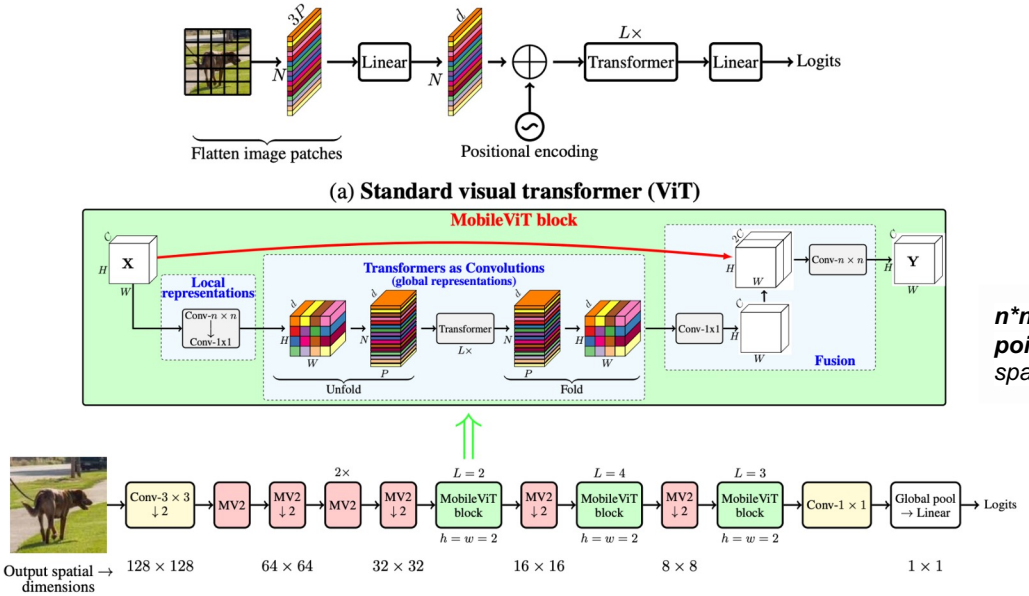
APPLE / technical report

- **MobileViT**
  - **A light-weight ViT model**

is it possible to combine the strengths of CNNs and ViTs to build a light-weight and low latency network for mobile vision tasks?

- Mobile vision task
  - CNNs (de-facto)
    - Spatial inductive biases allow them to learn representations with fewer parameters across different vision tasks.
    - However, these networks are spatially **local**.
  - ViTs
    - To learn **global** representations.
    - Unlike CNNs, ViTs are heavy-weight.
- Results
  - MobileViT significantly outperforms CNN- and ViT-based networks. (across different tasks and datasets.)
    - ImageNet-1k dataset
    - MS-COCO object detection task

N E



(b) **MobileViT**. Here, Conv-$n \times n$ in the MobileViT block represents a standard $n \times n$ convolution and MV2 refers to MobileNetv2 block. Blocks that perform down-sampling are marked with $\downarrow 2$.

Figure 1: **Visual transformers vs. MobileViT**

MobileViT Architecture (MobileViT block)
encodes both local and global information in a tensor effectively.

1) **Local Representations** *(CNN style)*
   a) n*n conv
   b) pointwise conv (1*1)

n*n convolutional layer *: encodes local spatial information*
point-wise convolutional layer*: projects the tensor to a high-dimensional space (d-dimensional) by learning linear combinations of the input channels.*

1) **Global Representations** *(ViT style)*
   a) unfold - fold
   b) 1*1 conv
   c) n*n conv

CORE IDEA
- transformer 를 이용하여 global representations 학습 → convolutions 과 같음.
- Learn global representations with transformers as convolutions.

Depth-wise conv & Point-wise conv
https://sotudy.tistory.com/10

3. Depth-wise Convolution

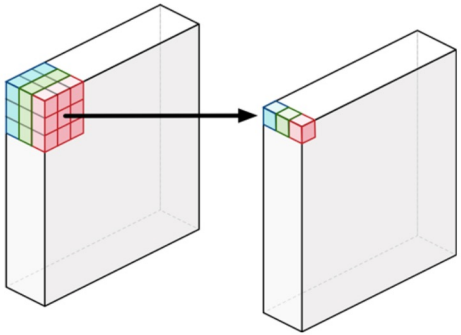


그림 5 Depth-wise Convolution [6]

채널마다 따로 필터를 학습하는 것입니다.
일반적인 convolution filter는 입력의 모든 채널의 영향을 받게 되므로 완벽히 특정 채널만의 Spatial feature를 추출하는 것이 불가능합니다.
Depth-wise convolution은 각 단일 채널에 대해서만 수행되는 필터들을 사용합니다. 그렇기에 필터 수는 입력 채널의 수와 동일합니다.
결과적으로 입력 채널 수만큼 그룹을 나눈 Grouped Convolution과 같습니다.

4. Point-wise Convolution (1 x 1 convolution)

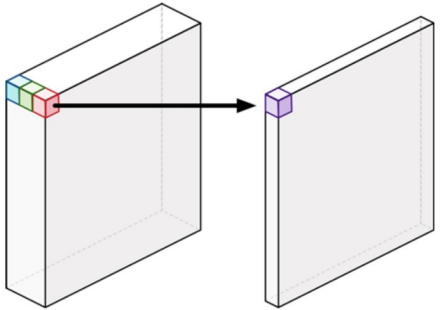
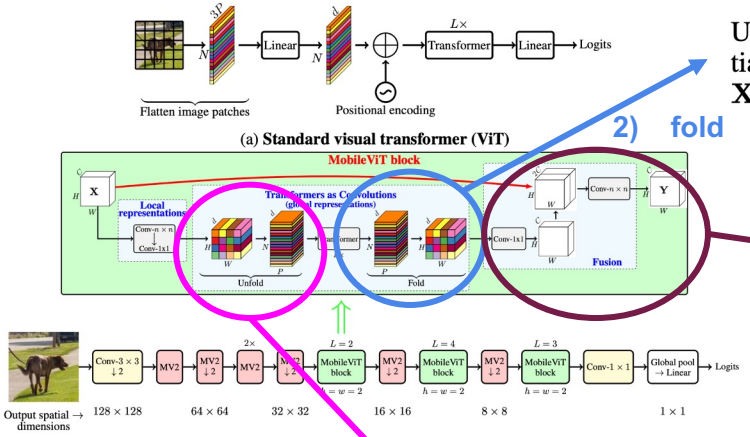


그림6 Point-wise Convolution [6]

Point-wise convolution은 커널 크기(filter크기)가 1x1로 고정된 convolution layer를 말합니다.
input에 대한 Spatial Feature는 추출하지 않은 상태로, 각 채널에 대한 연산만 수행합니다. 따라서 output의 크기는 변하지 않고, channel의 수는 자유롭게 조절할 수 있습니다.
하나의 필터는 각 입력 채널별로 하나의 가중치만을 가집니다. 이 가중치는 해당 채널의 모든 영역에 동일하게 적용됩니다. 즉, 입력 채널들에 대한 Linear Combination과 같습니다.

보통 dimensional reduction을 위해 많이 쓰입니다. 이것은 channel의 수를 줄이는 것을 의미하는데 연산량을 많이 줄여줄 수 있어 중요한 역할을 하게 됩니다.



(b) **MobileViT**. Here, Conv-$n \times n$ in the MobileViT block represents a standard $n \times n$ convolution and MV2 refers to MobileNetv2 block. Blocks that perform down-sampling are marked with $\downarrow 2$.

Figure 1: **Visual transformers vs. MobileViT**

2) fold

Unlike ViTs that lose the spatial order of pixels, MobileViT neither loses the patch order nor the spatial order of pixels within each patch (Figure 1b). Therefore, we can fold $\mathbf{X}_G \in \mathbb{R}^{P \times N \times d}$ to obtain $\mathbf{X}_F \in \mathbb{R}^{H \times \tilde{W} \times d}$. $\mathbf{X}_F$ is then projected to low C-dimensional space using a point-wise convolution and combined with $\mathbf{X}$ via concatenation operation. Another $n \times n$ convolutional layer is then used to fuse local and global features in the concatenated tensor. Note that because $\mathbf{X}_U(p)$ encodes local information from $n \times n$ region using convolutions and $\mathbf{X}_G(p)$ encodes global information across P patches for the p-th location, each pixel in $\mathbf{X}_G$ can encode information from all pixels in $\mathbf{X}$, as shown in Figure 4. Thus, the overall effective receptive field of MobileViT is $H \times W$.

3) 1*1 conv
4) n*n conv

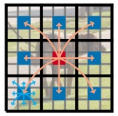


Figure 4: **Every pixel sees every other pixel in the MobileViT bloc** pixel attends to **blue** pixels (pixels at the corresponding location in other Because **blue** pixels have already encoded information about the neight tions, this allows the **red** pixel to encode information from all pixels in **black** and **gray** grids represents a patch and a pixel, respectively.

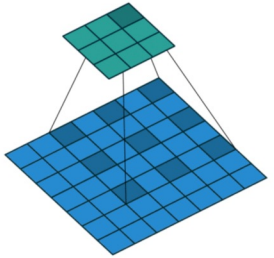
1) unfold

To enable MobileViT to learn global representations with spatial inductive bias, we unfold $\mathbf{X}_L$ into N non-overlapping flattened patches $\mathbf{X}_U \in \mathbb{R}^{P \times N \times d}$. Here, $P = wh$, $N = \frac{HW}{P}$ is the number of patches, and $h \leq n$ and $w \leq n$ are height and width of a patch respectively. For each $p \in \{1, \cdots, P\}$, inter-patch relationships are encoded by applying transformers to obtain $\mathbf{X}_G \in \mathbb{R}^{P \times N \times d}$ as:

$$\mathbf{X}_G(p) = \text{Transformer}(\mathbf{X}_U(p)), 1 \leq p \leq P \quad (1)$$

Dilated Convolutions (확장된 Convolution)

(a.k.a. atrous convolutions)



2D convolution using a 3 kernel with a dilation rate of 2 and no padding

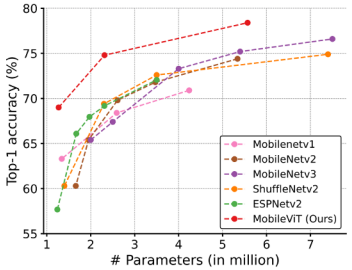
Dilated Convolution은 Convolutional layer에 또 다른 파라미터인 dilation rate를 도입했습니다. dilation rate은 커널 사이의 간격을 정의합니다. dilation rate가 2인 3x3 커널은 9개의 파라미터를 사용하면서 5x5 커널과 동일한 시야(view)를 가집니다.

5x5 커널을 사용하고 두번째 열과 행을 모두 삭제하면 (3x3 커널을 사용한 경우 대비)동일한 계산 비용으로 더 넓은 시야를 제공합니다.

Dilated convolution은 특히 real-time segmentation 분야에서 주로 사용됩니다. 넓은 시야가 필요하고 여러 convolution이나 큰 커널을 사용할 여유가 없는 경우 사용합니다

Experiments

- CNNs & ViTs (on ImageNet-1k)
- MobileViT significantly outperforms CNN- and ViT-based networks across different tasks and datasets.
  On the ImageNet-1k dataset, MobileViT achieves top-1 accuracy of 78.4% with about 6 million parameters, which is 3.2% and 6.2 more accurate than MobileNetv3 (CNN-based) and DeIT (ViT-based)



(a) Comparison with light-weight CNNs

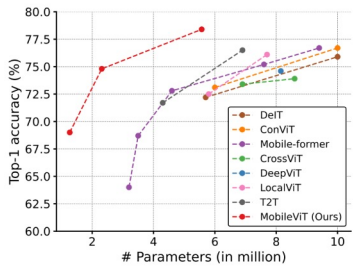
Model	# Params. ⇓	Top-1 ⇑
MobileNetv1	2.6 M	68.4
MobileNetv2	2.6 M	69.8
MobileNetv3	2.5 M	67.4
ShuffleNetv2	2.3 M	69.4
ESPNetv2	2.3 M	69.2
MobileViT-XS (Ours)	2.3 M	**74.8**

(b) Comparison with light-weight CNNs (similar parameters)

Model	# Params. ⇓	Top-1 ⇑
DenseNet-169	14 M	76.2
EfficientNet-B0	5.3 M	76.3
ResNet-101	44.5 M	77.4
ResNet-101-SE	49.3 M	77.6
MobileViT-S (Ours)	5.6 M	**78.4**

(c) Comparison with heavy-weight CNNs

Figure 6: **MobileViT vs. CNNs** on ImageNet-1k validation set. All models use basic augmentation.



(a)

Row #	Model	Augmentation	# Params. ⇓	Top-1 ⇑
R1	DeIT	Basic	5.7 M	68.7
R2	T2T	Advanced	4.3 M	71.7
R3	DeIT	Advanced	5.7 M	72.2
R4	PiT	Basic	10.6 M	72.4
R5	Mobile-former	Advanced	4.6 M	72.8
R6	PiT	Advanced	4.9 M	73.0
R7	CrossViT	Advanced	6.9 M	73.4
R8	MobileViT-XS (Ours)	Basic	2.3 M	**74.8**
R9	CeiT	Advanced	6.4 M	76.4
R10	DeIT	Advanced	10 M	75.9
R11	T2T	Advanced	6.9 M	76.5
R12	ViL	Advanced	6.7 M	76.7
R13	LocalVit	Advanced	7.7 M	76.1
R14	Mobile-former	Advanced	9.4 M	76.7
R15	PVT	Advanced	13.2 M	75.1
R16	ConViT	Advanced	10 M	76.7
R17	PiT	Advanced	10.6 M	78.1
R18	BoTNet	Advanced	20.8 M	78.3
R19	MobileViT-S (Ours)	Basic	5.6 M	**78.4**

(b)

Figure 7: **MobileViT vs. ViTs** on ImageNet-1k validation set. Here, **basic** means ResNet-style augmentation while **advanced** means a combination of augmentation methods with basic (e.g., MixUp (Zhang et al., 2018), RandAugmentation (Cubuk et al., 2019), and CutMix (Zhong et al., 2020)).

Experiments

MobileViT as a general-purpose backbone

- **Mobile Object Detection (Table 1)**
  - SSDLite with MobileViT outperforms SSDLite with other light-weight CNN models (MobileNetv1, 2, 3, MNASNet, and MixNet)
- **Mobile Semantic Segmentation (Table 2)**
  - DeepLabv3 with MobileViT is smaller and better. The performance of DeepLabv3 is improved by 1.4%, and its size is reduced by 1.6x when MobileViT is used as a backbone instead of MobileNetv2.
- **Performance on mobile devices**

Feature backbone	# Params. ⇓	mAP ⇑
MobileNetv3	4.9 M	22.0
MobileNetv2	4.3 M	22.1
MobileNetv1	5.1 M	22.2
MixNet	4.5 M	22.3
MNASNet	4.9 M	23.0
MobileViT-XS (Ours)	**2.7 M**	24.8
MobileViT-S (Ours)	5.7 M	**27.7**

(a) Comparison w/ light-weight CNNs

Feature backbone	# Params. ⇓	mAP ⇑
VGG	35.6 M	25.1
ResNet50	22.9 M	25.2
MobileViT-S (Ours)	**5.7 M**	**27.7**

(b) Comparison w/ heavy-weight CNNs

Table 1: **Detection w/ SSDLite.**

Feature backbone	# Params. ⇓	mIOU ⇑
MobileNetv1	11.2 M	75.3
MobileNetv2	4.5 M	75.7
MobileViT-XXS (Ours)	1.9 M	73.6
MobileViT-XS (Ours)	2.9 M	**77.1**
ResNet-101	58.2 M	**80.5**
MobileViT-S (Ours)	6.4 M	79.1

Table 2: **Segmentation w/ DeepLabv3.**

(a) Classification @ 256 × 256 (b) Detection @ 320 × 320 (c) Segmentation @ 512 × 512

Figure 8: **Inference time of MobileViT models on different tasks.** Here, dots in green color region represents that these models runs in real-time (inference time < 33 ms).

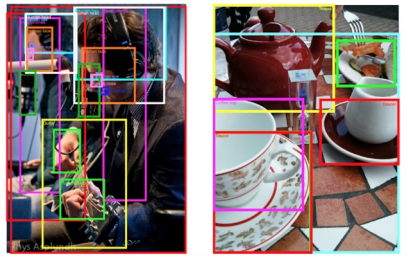
Model	# Params. ⇓	Time ⇓	Top-1 ⇑
MobileNetv2†	3.5 M	**0.92 ms**	73.3
DeIT	5.7 M	10.99 ms	72.2
PiT	4.9 M	10.56 ms	73.0
MobileViT (Ours)	**2.3 M**	7.28 ms	**74.8**

Table 3: **ViTs are slower than CNNs.**
†Results with multi-scale sampler (§B).

OBJECT-AWARE CROPPING FOR SELF-SUPERVISED LEARNING

https://arxiv.org/pdf/2112.00319v1.pdf

Shlok Mishra[1], **Anshul Shah**[2], **Ankan Bansal**[1], **Abhyuday Jagannatha**[3]
Abhishek Sharma, **David Jacobs**[1] , **Dilip Krishnan**[4]
[1]University of Maryland, College Park,
[2]Johns Hopkins University, [3]University of Massachusetts Amherst [4]Google Research
{shlokm,dwj}@cs.umd.edu, dilipkay@google.com



Annotated images from the Open Images dataset. Left: Mark Paul Gosselaar plays the guitar by Rhys A. Right: Civilization by Paul Downey. Both images used under CC BY 2.0 license.

- In self-supervised learning, cropping data augmentation 을 통해 성능향상
  - SSL loss에서 사용할 Positive views image에 사용하기 위해 sub-regions 방식으로 random cropping
- 주어진 image에 대해 randomly cropped and resized regions 하는 것은 object 의 다양성을 부여하여 좀더 representation을 잘 capture하기 위함.
  - 이러한 가정은, ImageNet과 같은 (중앙에 큰 객체가 있는 데이터 셋) 에서 만족 // corp하여도 object가 존재할 가능성이 있는 데이터셋
- However, OpenImages or COCO와 같이 (real world 를 더 잘 나타내는 데이터 셋) 에서는 일반적으로 여러개의 작은 개체 존재.
- In this work, we show that self-supervised learning based on the usual random cropping performs poorly on such datasets.
- We propose “object-aware cropping”
- 해당 Approach를 이용하여 OpenImages에서 MoCo-v2 (random cropping 대비) 8.8% mAP 성능향상 달성
- COCO나 PASCAL-VOC object detection and segmentation task에서도 SOTA SSL approaches.
- “object-aware cropping” = Efficient, Simple and General 한 Approach.

Object-Aware Cropping for Self-Supervised Learning

Random Cropping

"object-aware cropping"

(Efficient, Simple and General 한 Approach.)

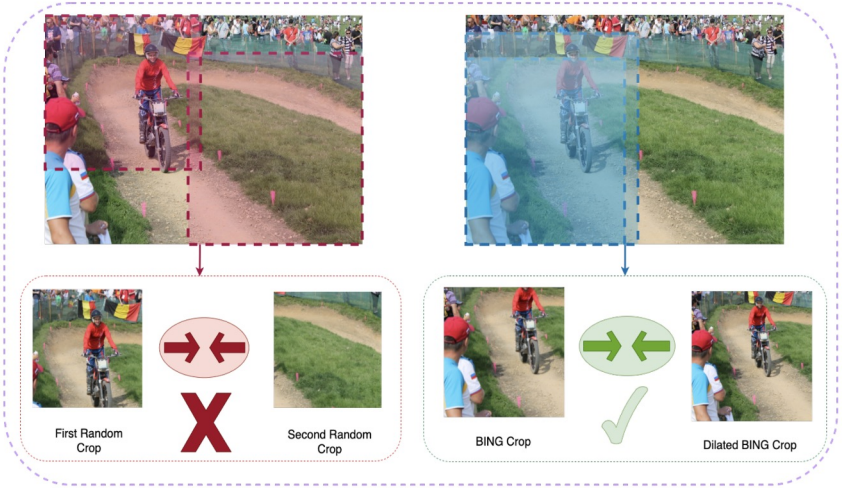


Figure 1: Illustration of object aware cropping. Top-Left: We show the original image with random crops overlaid. Bottom (red panel): Overlap between random crops tend to miss the object of interest. Top-Right: We show crops generated from the BING (Cheng et al., 2014) algorithm and also the dilated BING crop. Bottom-Right (green panel): Instead, we use BING-based object-aware crops. This incorporates both object and scene information into the MoCo-v2 (or other SSL frameworks).

https://mmcheng.net › Papers › ObjectnessBING PDF

Binarized normed gradients for objectness estimation at 300fps

MM Cheng 저술 · 1223회 인용 — On the challenging PASCAL VOC2007 dataset, using 1000 proposals per **image** and intersection- over-union threshold of 0.5, our proposal method achieves a 95.6% ...

페이지 26개

In this paper, we propose a surprisingly simple and powerful feature which we call "BING", to help search for objects using objectness scores. Our work

(see Section 5.4). Its poor generalization ability has restricted its usage, so *RPN is usually only used in object detection*. In comparison, BING is based on low-level cues concerning enclosing boundaries and thus can produce category independent object proposals, which has demonstrated applications in multi-label image classification [23], semantic segmentation [25], video classification [24], co-salient object detection [29], deep multi-instance learning [26], and video summarisation [27]. However, several researchers [34–37] have noted that BING's proposal localization is weak.

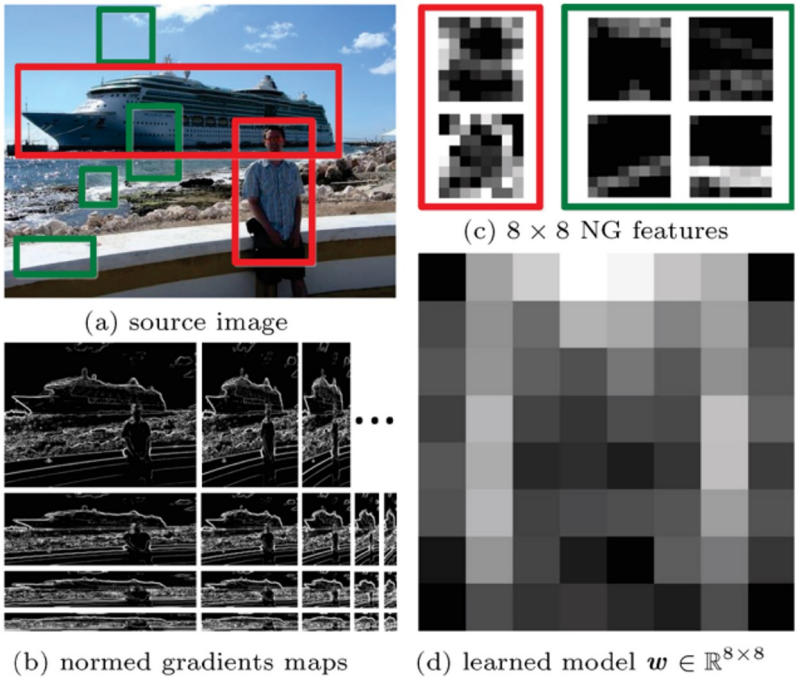


Fig. 1 Although object (red) and non-object (green) windows vary greatly in image space (a), at proper scales and aspect ratios which correspond to a small fixed size (b), their corresponding normed gradients (NG features) (c), share strong correlation. We learn a single 64D linear model (d) for selecting object proposals based on their NG features.

https://mmcheng.net › Papers › ObjectnessBING PDF

Binarized normed gradients for objectness estimation at 300fps

MM Cheng 저술 · 1223회 인용 — On the challenging PASCAL VOC2007 dataset, using 1000 proposals per **image** and intersection- over-union threshold of 0.5, our proposal method achieves a 95.6% ...

페이지 26개

In this paper, we propose a surprisingly simple and powerful feature which we call "BING", to help search for objects using objectness scores. Our work

(see Section 5.4). Its poor generalization ability has restricted its usage, so *RPN is usually only used in object detection*. In comparison, BING is based on low-level cues concerning enclosing boundaries and thus can produce category independent object proposals, which has demonstrated applications in multi-label image classification [23], semantic segmentation [25], video classification [24], co-salient object detection [29], deep multi-instance learning [26], and video summarisation [27]. However, several researchers [34–37] have noted that BING's proposal localization is weak.

Object-Aware Cropping for Self-Supervised Learning

Random Cropping

"object-aware cropping"

(Efficient, Simple and General 한 Approach.)

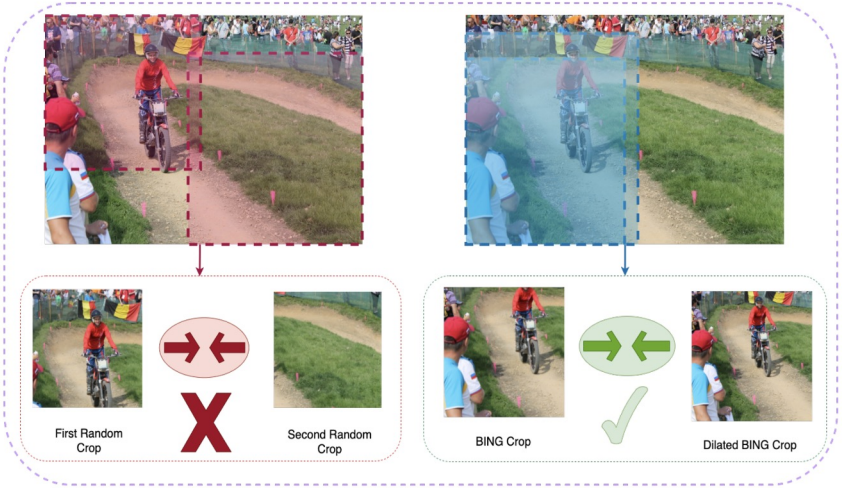


Figure 1: Illustration of object aware cropping. Top-Left: We show the original image with random crops overlaid. Bottom (red panel): Overlap between random crops tend to miss the object of interest. Top-Right: We show crops generated from the BING (Cheng et al., 2014) algorithm and also the dilated BING crop. Bottom-Right (green panel): Instead, we use BING-based object-aware crops. This incorporates both object and scene information into the MoCo-v2 (or other SSL frameworks).

- Random Crops: tends to miss the object of interest.
- dilated BING crop: object of interest 유지

- 빨간색 패널(random coprs)과 같은 현상이 Self-supervised contrastive learning에 크게 영향을 미칠 것.
- 해당 부분 object-aware cropping으로 바꾸어 실험시 MoCo-V2 pipline 에서 16.5 % mAP 차이확인, BYOL이나 CMC같은 경우도 비슷한 현상.

⇒ 그래서 Core Problem은 Random scene crops

- do not contain enough information about objects.
- cusing degarded representations quiality

Methods

1. BING algorlithm outputs multiple object proposals, one of which we pick at random as the first view for SSL loss
2. Second view, object-aware cropping
   a. scene-level random crop [obj - scene]
   b. obj + dilate [obj - obj + dilate]
   c. obj + shift [obj - obj + shift]

⇒ Baseline (scene-level random crops to both views) [scene - scene]

Object-Aware Cropping for Self-Supervised Learning

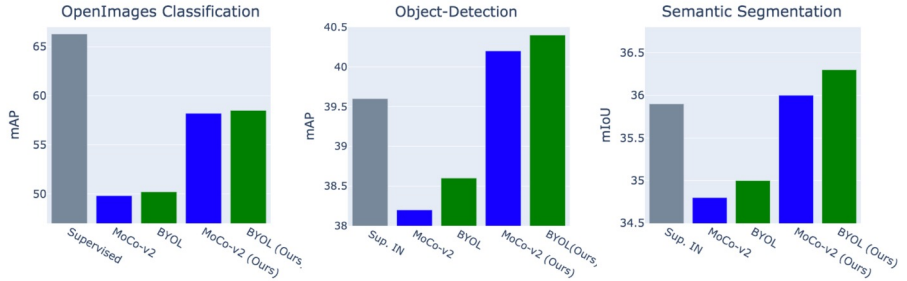
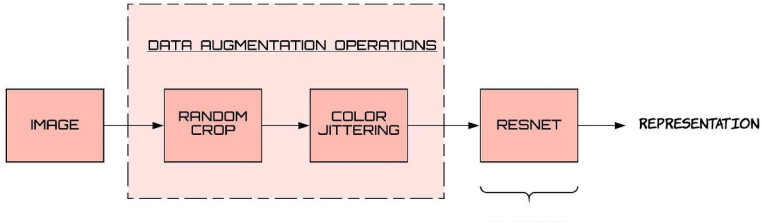
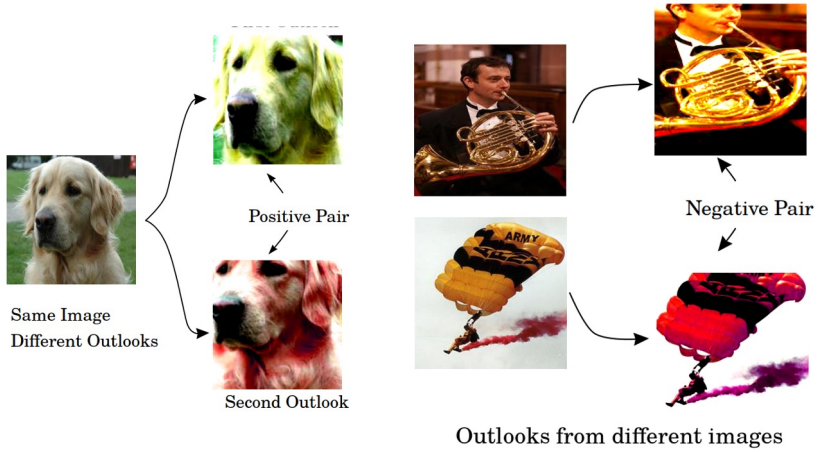


Figure 2: Our object-aware cropping approach can be easily plugged into self-supervised learning pipelines and achieves excellent results for classification on OpenImages (left), COCO object detection (middle) and COCO semantic segmentation (right). Using object-aware cropping instead of scene-level cropping provides a consistent boost on BYOL (Richemond et al., 2020) and MoCo-v2 (Chen et al., 2020b), two of the top SSL methods. In the case of COCO object detection and semantic segmentation, this boost allows us to beat pre-training on supervised ImageNet (denoted "Sup. IN"). In the case of classification on OpenImages, we reduce the gap to supervised training by nearly 50%.

Model	OpenImages (mAP)	ImageNet (Top-1 %)
Supervised Performance	66.3	76.2
CMC (Tian et al., 2019)	48.7 (-17.6)	60.0 (-16.2)
BYOL (Grill et al., 2020)	50.2 (-16.1)	70.7 (-5.5)
SwAV (Caron et al., 2020)	51.3 (-15.0)	72.7 (-3.5)
MoCo-v2 (Scene-Scene crop)	49.8 (-16.5)	67.5 (-8.7)
MoCo-v2 (Object-Object+Dilate crop) (Ours)	58.6 (-7.7)	68.0 (-8.2)

Table 1: Classification results on OpenImages and Imagenet. For each SSL method, we show in parentheses the gap to fully supervised training (same number of epochs). The last row shows that our proposed approach using *obj-obj+dilate* cropping reduces the gap on OpenImages by nearly half compared to the baselines, improving over the *scene-scene* cropping based SSL methods by between 6.8 to 9.4 mAP points. We also observe improvements on ImageNet as well.





Moco-V2, default Positive Sampling [scene - scene]

Model	Crops	Obj-Obj+Dilate	Obj-Scene	Scene-Scene	mAP
Supervised	-	-	-	-	66.3
MoCo-v2	Ground Truth boxes	-	✓	-	58.9
MoCo-v2	Ground Truth boxes	✓	-	-	60.2
MoCo-v2	-	-	-	✓	49.8
BYOL	-	-	-	✓	50.2
MoCo-v2	Unsupervised proposal boxes	-	✓	-	58.0
MoCo-v2	EdgeBoxes crops	-	✓	-	57.1
MoCo-v2	BING crops	-	✓	-	58.1
BYOL	BING crops	-	✓	-	58.5
MoCo-v2	BING crops	✓	-	-	58.6
BYOL	BING crops	✓	-	-	59.1

Table 3: Crop approaches on OpenImages: using BING crops to generate one view, and a dilated crop or a scene crop for the other positive, we are able to reduce the difference between SSL and Supervised Learning by close to 50% (compare the last two rows to the first row). Using ground-truth boxes to generate crops from OpenImages improves the pre-training performance marginally compared to BING crops. Obj-Obj+Dilate (last two rows) have the best performance, although Obj-Scene also does well compared to Scene-Scene.

Description	AP	AP_{50}	AP_{75}	AP_s	AP_l	AP_m
Supervised (Random Initialization)	32.8	50.9	35.3	29.9	47.9	32.0
Supervised (ImageNet Pre-trained)	39.7	59.5	43.3	35.9	56.6	38.6
MoCo-v2 (Chen et al., 2020b)	38.2	58.9	41.6	34.8	55.3	37.8
BYOL (Hénaff et al., 2021)	38.8	58.5	42.2	35.0	55.9	38.1
Dense-CL (Wang et al., 2021)	39.6	59.3	43.3	35.7	56.5	38.4
CAST (Selvaraju et al., 2020) (180K steps)	39.4	60.0	42.8	35.8	57.1	37.6
Self-EMD (Liu et al., 2021a) (Uses BYOL)	39.8	60.0	43.4	-	-	-
MoCo-V2 (Obj-Scene) (Ours)	39.4	59.8	42.9	35.8	57.8	38.7
MoCo-v2 (Obj-Obj+Dilate) (Ours)	**39.7**	**60.1**	**43.4**	**36.0**	**57.3**	**38.8**
MoCo-v2 (Obj-Obj+Dilate) (180k steps)	**40.2**	**60.6**	**43.6**	**36.3**	**57.4**	**39.0**
BYOL (Obj-Obj+Dilate) (Ours)	**40.1**	**60.8**	**43.6**	**36.4**	**58.4**	**39.5**
Dense-CL (Obj-Obj+Dilate) (Ours)	**40.4**	**60.4**	**44.0**	**36.6**	**57.9**	**39.5**

Table 4: Object detection (first 3 columns) and Semantic Segmentation (last 3 columns) results on COCO dataset. All SSL models have been pre-trained on COCO dataset and then finetuned on COCO. Note that for the same number of finetuning iterations (180K), we outperform CAST (Selvaraju et al., 2020) which also relies on localized crops. All other methods are run for 90K, finetuning iterations. For any SSL method, we compare (BYOL, Moco-v2, Dense-CL) adding Obj-Obj+Dilate crop improves performance.

Description	AP	AP_{50}	AP_{75}	AP_s	AP_l	AP_m
COCO: MoCo-v2 (Scene-Scene crop)	38.2	58.9	41.6	34.8	55.3	37.8
COCO: MoCo-v2 (Obj-Obj+Dilate crop)	**40.2**	**60.6**	**43.6**	**36.3**	**57.4**	**39.0**
VOC: MoCo-v2 (Scene-Scene crop)	56.1	81.3	61.3	-	-	-
VOC: MoCo-v2 (Obj-Obj+Dilate crop)	**57.6**	**82.5**	**63.8**	-	-	-

Table 5: Object detection (first 3 columns) and semantic segmentation (last 3 columns) results on COCO (first 2 rows) and VOC (last 2 rows). All SSL models have been pre-trained on complete OpenImages dataset(1.9 million images) for 75 epochs and then finetuned on COCO and VOC dataset.

Florence: A New Foundation Model for Computer Vision

https://arxiv.org/pdf/2111.11432v1.pdf

Lu Yuan [1] Dongdong Chen [*1] Yi-Ling Chen [*1] Noel Codella [*1] Xiyang Dai [*1] Jianfeng Gao [*2] Houdong Hu [*1]
Xuedong Huang [*1] Boxin Li [*1] Chunyuan Li [*2] Ce Liu [*1] Mengchen Liu [*1] Zicheng Liu [*1] Yumao Lu [*1]
Yu Shi [*1] Lijuan Wang [*1] Jianfeng Wang [*1] Bin Xiao [*1] Zhen Xiao [*1] Jianwei Yang [*2] Michael Zeng [*1]
Luowei Zhou [*1] Pengchuan Zhang [*2]

- Vision foundation models 인 CLIP을 대체할 만 할(더 좋은) Florence 발표
- We introduce a new computer vision foundation model, 'Florence'

video retrieval and action recognition. Moreover, *Florence* demonstrates outstanding performance in many types of transfer learning: fully sampled fine-tuning, linear probing, few-shot transfer and zero-shot transfer for novel images and objects. All of these properties are critical for our vision foundation model to serve general purpose vision tasks. *Florence* achieves new state-of-the-art results in majority of 44 representative benchmarks, *e.g.* ImageNet-1K zero-shot classification with top-1 accuracy of 83.74 and the top-5 accuracy of 97.18, 62.4 mAP on COCO fine tuning, 80.36 on VQA, and 87.8 on Kinetics-600.

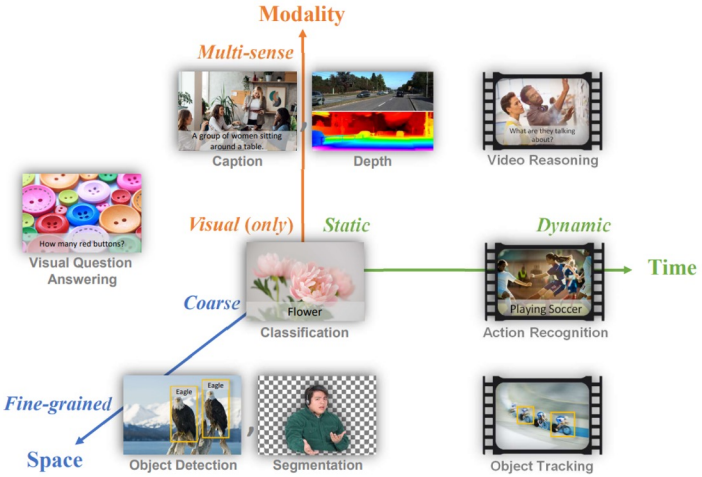
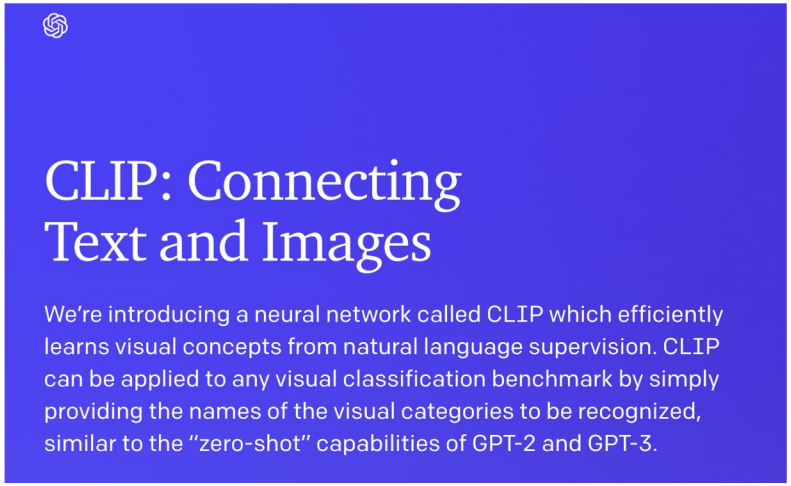
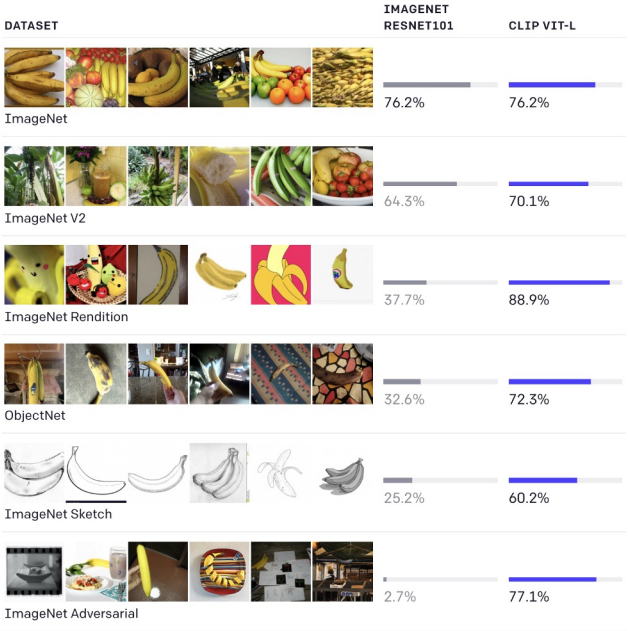


Figure 1. Common computer vision tasks are mapped to a *Space-Time-Modality* space. A computer vision foundation model should serve as general purpose vision system for all of these tasks.

- CLIP

https://openai.com/blog/clip/

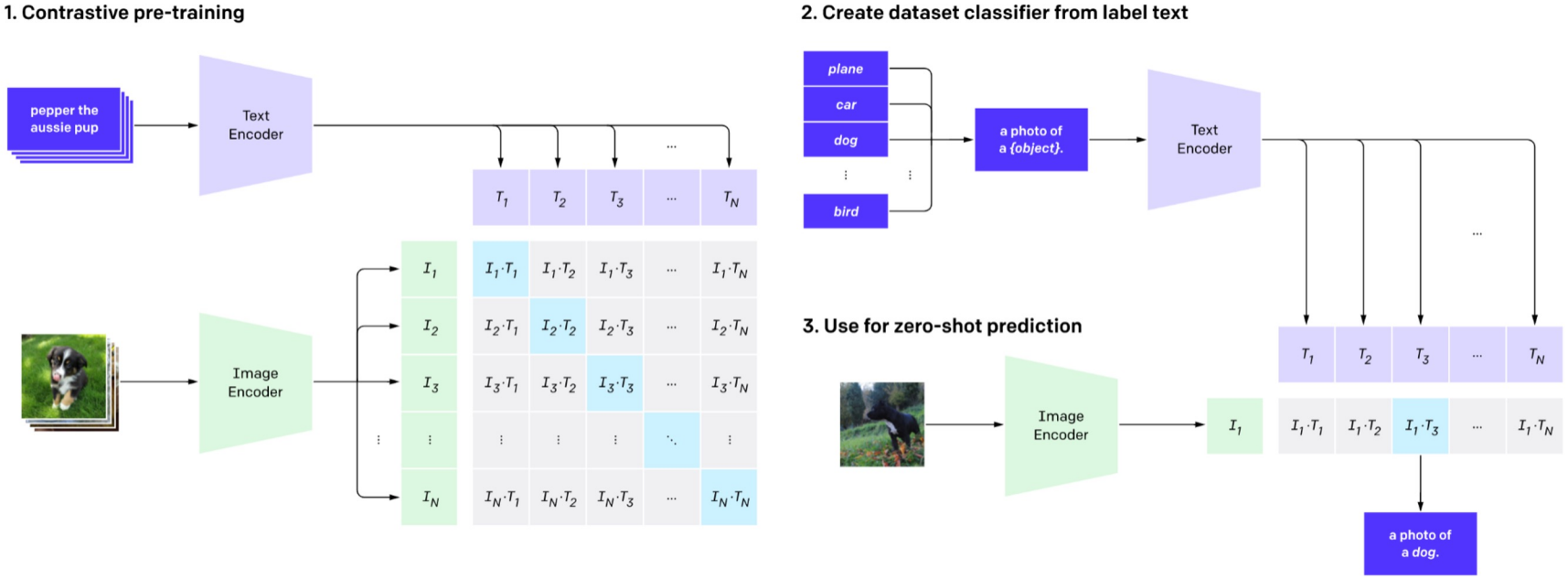




- 4000억개 training Dataset image
- 256개의 V100 12일간 학습
- (이미지, 물체분류) 데이터 대신 (이미지, 텍스트) 데이터 사용
- 수작업 labeling 없이 웹 크롤링을 통해 자동으로 이미지와 그와 연관된 자연어 텍스트 추출.
- (이미지, 텍스트)로 구성된 데이터셋은 정해진 Class label이 없기 때문에 classification 문제로 학습할 수는 없음.

⇒ N개의 이미지들과 N개의 텍스트들 사이의 올바른 연결관계를 찾는 문제로 네트워크 학습

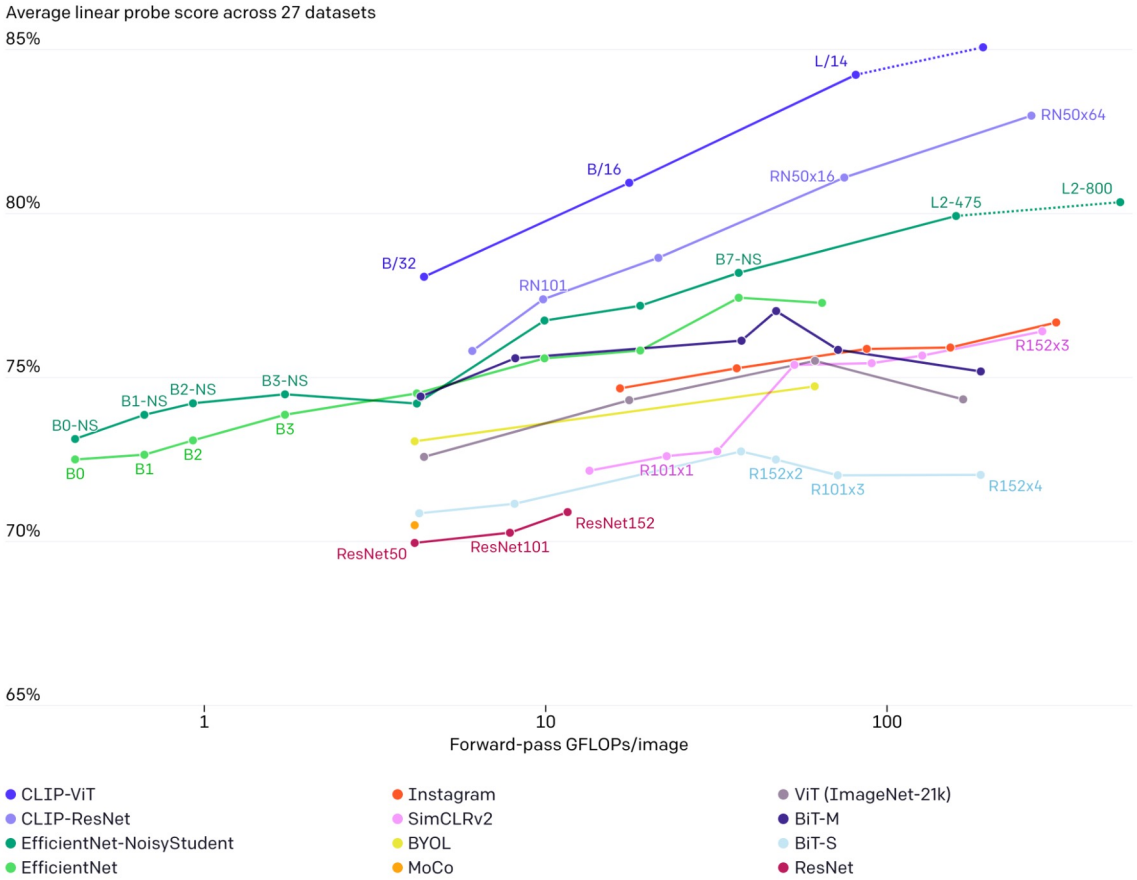
Florence: A New Foundation Model for Computer Vision



1. 이미지 인코더, 텍스트 인코더가 존재, 각 인코더를 통과해서 나온 N개의 이미지, 텍스트 특징 벡터들 사이의 올바른 연결관계를 학습.
2. zero-shot learning 가능성 테스트 (학습과정에서 한번도 보지 못한 문제 및 데이터셋에 대해 성능평가)
    a. 학습된 CLIP 모델을 이용해 ImageNet 데이터의 분류과정 수행
    b. 평가 대상이 되는 데이터셋이 다름, 이미지-텍스트 연결이 아닌 이미지 분류 문제에 적용하는 것이므로 문제의 타입또한 다른상황
    c. Classification 문제를 위해 그림과 위 그림과 같은 방법으로 CLIP 모델 적용

⇒ ImageNet에 대한 zero-shot learning 결과 76.2%로 굉장히 높은 성능.

Florence: A New Foundation Model for Computer Vision



Fully Supervised learning (RESNET) 보다 성능이 좋음

ImageNet에서 가장 좋은 결과를 보여준 EfficientNet보다 높은 성능

- pretrained 된 이미지 인코더 모델의 마지막 단계에 linear classifier를 추가한 모델로도 평가 수행

- CLIP과 비슷하게 (이미지, text)를 사용하여 서로의 관계를 학습 (Contrastive Learning Pretrained)
- Vision task에 맞게 Adaptive Models 적용

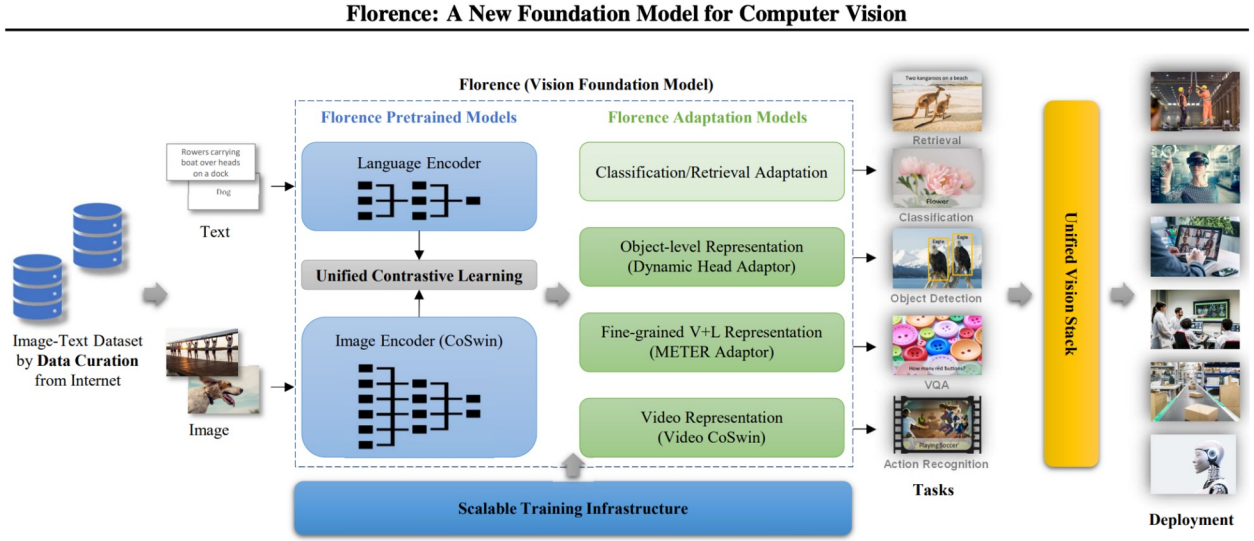


Figure 2. Overview of building Florence. Our workflow consists of data curation, unified learning, Transformer architectures and adaption. It shows the foundation model can be adapted to various downstream tasks and finally integrated into modern computer vision system to power real-world vision and multimedia applications. Compared with existing image-text pretraining models (Radford et al., 2021; Jia et al., 2021; Wud), mainly limited on cross-modal shared representation for classification and retrieval (illustrated by *light-green* adaptation module), Florence expands the representation to support object level, multiple modality, and videos respectively.

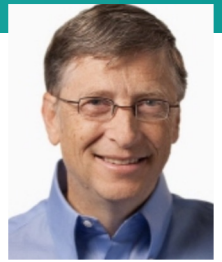
We raise the question: "*What is the foundation model for computer vision*?". But first, in order to better define what

we redefine ***foundation models for computer vision*** to be *a pre-trained model and its adapters* for solving all vision tasks in this Space-Time-Modality space, with transferability such as zero-/few-shot learning and fully fine tuning, etc.

- vision 계의 sensation? 논문에 저런 문구를 사용한다는건…. (оДO;)
- I Am "Microsoft" !!!

*Florence Team member in alphabetic order [1]Microsoft Cloud and AI [2]Microsoft Research Redmond. Correspondence to: Lu Yuan <luyuan@microsoft.com>.

Florence: A New Foundation Model for Computer Vision



- Data Curation - 9억개 사용 (FLD-900M) [해당 dataset을 만들기 위해 Internet Image 약 30억개를 처리]
- The model takes 10 days to train on 512 NVIDIA-A100 GPUs with 40GB memory per GPU. ⇒ 빌게이츠 파워

Action Classification

Edit

128 papers with code • 14 benchmarks • 19 datasets

This task has no description! Would you like to contribute one?

Benchmarks

Add a Result

Trend	Dataset	Best Model	Paper Title	Paper	Code	Compare
	Kinetics-400	Florence (curated FLD-900M pretrain)	Florence: A New Foundation Model for Computer Vision			See all
	Kinetics-600	Florence (curated FLD-900M pretrain)	Florence: A New Foundation Model for Computer Vision			See all
	Charades	TokenLearner	TokenLearner: What Can 8 Learned Tokens Do for Images and Videos?			See all
	Moments in Time	VATT-Large	VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text			See all
	Kinetics-700	MViT-L (ImageNet-21k pretrain)	Improved Multiscale Vision Transformers for Classification and Detection			See all

Classification (Zero-shot transfer of image classification)

Florence: A New Foundation Model for Computer Vision

	Food101	CIFAR10	CIFAR100	SUN397	Stanford Cars	FGVC Aircraft	VOC2007	DTD	Oxford Pets	Caltech101	Flowers102	ImageNet
CLIP-ResNet-50x64	91.8	86.8	61.3	48.9	76.0	35.6	83.8	53.4	93.4	90.6	77.3	73.6
CLIP-ViT-L/14 (@336pix)	93.8	**95.7**	77.5	68.4	78.8	37.2	84.3	55.7	93.5	92.8	78.3	76.2
FLIP-ViT-L/14	92.2	**95.7**	75.3	73.1	70.8	**60.2**	-	60.7	92.0	93.0	**90.1**	78.3
Florence-CoSwin-H (@384pix)	**95.1**	94.6	**77.6**	**77.0**	**93.2**	55.5	**85.5**	**66.4**	**95.9**	**94.7**	86.2	**83.7**

Table 1. Zero-shot transfer of image classification comparisons on 12 datasets: CLIP-ResNet-50x64 (Radford et al., 2021), FLIP-ViT-L/14 (Yao et al., 2021).

Action Recognition

Method	Pretraining Data	Kinetics-400		Kinetics-600		Views	Params
		Top-1	Top-5	Top-1	Top-5		
ViViT-H/16x2	JFT-300M	84.8	95.8	85.8	96.5	4 × 3	648M
VideoSwin-L	ImageNet-22K	84.6	96.5	85.9	97.1	4 × 3	200M
VideoSwin-L	ImageNet-22K	84.9	96.7	86.1	97.3	10 × 5	200M
TokenLearner 16at18+L/10	JFT-300M	85.4	96.3	86.3	97.0	4 × 3	460M
Florence	FLD-900M	**86.5**	**97.3**	**87.8**	**97.8**	4 × 3	647M

Table 10. Comparison to state-of-the-art methods, including ViViT (Arnab et al., 2021), VideoSwin (Liu et al., 2021b), TokenLearner (Ryoo et al., 2021), on Kinetics-400 and Kinetics-600. Views indicate *#temporal clip* × *#spatial crop*.

Benchmark	Model	AP
COCO miniVal	DyHead	60.3
	Soft Teacher	60.7
	Florence	**62.0**
COCO test-Dev	DyHead	60.6
	Soft Teacher	61.3
	Florence	**62.4**
Object365	Multi-dataset Detection	33.7
	Florence	**39.3**
Visual Genome	VinVL	13.8
	Florence	**16.2**

Table 6. Object detection fine tuning comparisons with state-of-the-art methods, including DyHead (Dai et al., 2021a), Soft Teacher (Xu et al., 2021b), Multi-dataset Detection (Zhou et al., 2021), VinVL (Zhang et al., 2021b).

Object Detection

Florence: A New Foundation Model for Computer Vision

4. Conclusion and Future Work

In this paper we investigated a new paradigm of building a computer vision foundation model, *Florence*, as a general-

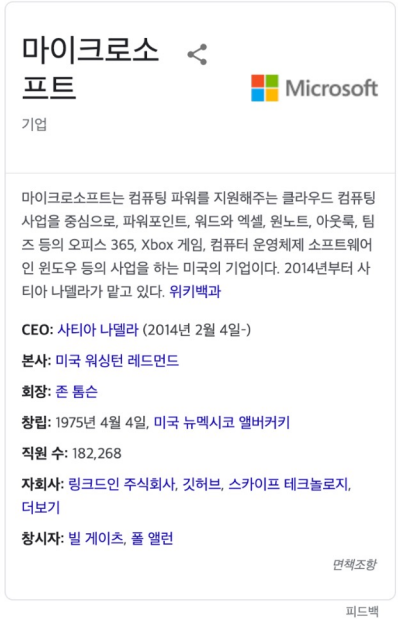
For the future work, we plan to include more vision tasks and applications, such as depth/flow estimation, tracking, and additional vision+language tasks. *Florence* is designed

- MS 주식을 사야하나..

[v1] Mon, 22 Nov 2021 18:59:55 UTC (6,636 KB)

public 1/8

- 그래서 model public?
- 그런이야기는 없는것같아요..(>﹏<)



End of the Document