Vision AI

2021 arXiv Trends

2021-09

hutom

no.	Paper Title	Correspondence	h-index
1	Vision Transformer with Progressive Sampling	Dahua Lin	51
2	CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows	Baining Guo	72
3	Learning with Noisy Labels for Robust Point Cloud Segmentation	Jing Liao	17
4	EvilModel: Hiding Malware Inside of Neural Network Models	Xiang Cui	

no.	Paper Title	Correspondence	h-index
5	Mobile-Former: Bridging MobileNet and Transformer	Zicheng Liu	65
6	Do Vision Transformers See Like Convolutional Neural Networks?	Alexey Dosovitskiy	43
7	Towards Robust Classification Model by Counterfactual and Invariant Data Generation		
8	Eyes Tell All: Irregular Pupil Shapes Reveal GAN-generated Faces	Siwei Lyu	46

Vision Transformer with Progressive Sampling

Xiaoyu Yue*[1] Shuyang Sun*[2] Zhanghui Kuang[3] Meng Wei[4]
Philip Torr[2] Wayne Zhang[3,6] Dahua Lin[1,5]
[1]Centre for Perceptual and Interactive Intelligence [2]University of Oxford
[3]SenseTime Research [4]Tsinghua University [5]The Chinese University of Hong Kong
[6]Qing Yuan Research Institute, Shanghai Jiao Tong University

https://arxiv.org/pdf/2108.01684.pdf

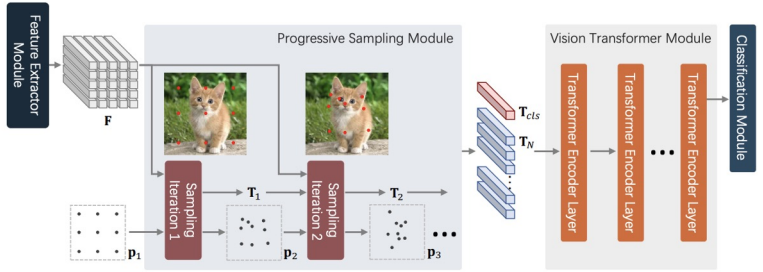


Figure 4. Overall architecture of the proposed Progressive Sampling Vision Transformer (PS-ViT). Given an input image, its feature map $\mathbf{F}$ is first extracted by the feature extractor module. Tokens $\mathbf{T}_i$ are then sampled progressively and iteratively at adaptive locations $\mathbf{p}_i$ over $\mathbf{F}$ in the progressive sampling module. The final output tokens $\mathbf{T}_N$ of the progressive sampling module are padded with the classification token $\mathbf{T}_{cls}$ and further fed into the vision transformer module to refine $\mathbf{T}_{cls}$, which is finally classified in the classification module.

- **Accepted to ICCV 2021**

- 기존의 vision transformer 는 고정된 length 로 token 을 split, token 사이의 relationship 을 이용해 학습
  - 이러한 naive 한 tokenization은 객체구조를 파괴하고, 관심없는 영역 (배경 등) 에 grid 를 할당하여, inference signal (간섭 신호) 를 야기할 수 있음.

- **본 논문은 discriminative regions 을 찾기 위해, iterative and progressive sampling 전략 제안**
- Overall architecture
  1. feature extraction module
  2. **progressive sampling module**
  3. vision transformer module (ViT, DeiT 와 같이 cls Token과 embedding Token 이 input 인 구조)
  4. classification module

- PS-ViT
  - iterative and progressive sampling 전략을 vision transformer 에 결합
  - adaptively 하게 이미지의 어디를 찾아야 하는지 학습 가능

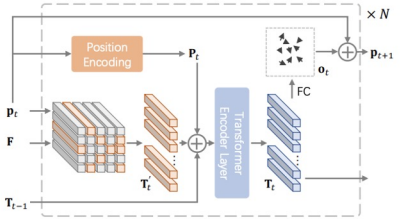


Figure 3. The architecture of the progressive sampling module. At each iteration, given the sampling location $\mathbf{p}_t$ and the feature map $\mathbf{F}$, we sample the initial tokens $\mathbf{T}'_t$ at $\mathbf{p}_t$ over $\mathbf{F}$, which are element-wisely added with the positional encodings $\mathbf{P}_t$ generated based on $\mathbf{p}_t$, and the output tokens $\mathbf{T}_{t-1}$ of the last iteration, and then fed into one transformation encoder layer to predict the tokens $\mathbf{T}_t$ of the current iteration. The offset matrix $\mathbf{o}_t$ is predicted via one fully-connected layer based on $\mathbf{T}_t$, which is added with $\mathbf{p}_t$ to obtain the sampling positions $\mathbf{p}_{t+1}$ for the next iteration. The above procedure is iterated N times.

Figure 5. Visualization of sampled locations in the proposed progressive sampling module. The start points of arrows are initial sampled locations ($\mathbf{p}_1$) while the end points of arrows are the final sampled locations ($\mathbf{p}_4$).

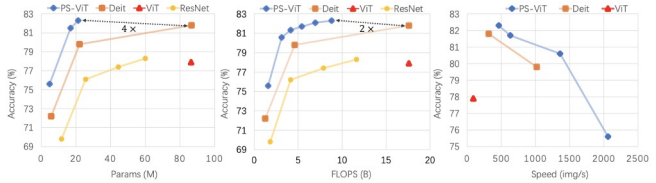


Figure 2. Comparisons between PS-ViT with state-of-the-art networks in terms of top-1 accuracy on ImageNet, parameter number, FLOPs, and speed. The chart on the left, middle and right show top-1 accuracy *vs.* parameter numbers, FLOPs and speed respectively. The speed is tested on the same V100 with a batch size of 128 for fair comparison.

Model	FLOPs (B)	Speed (img/s)	Top-1
RegNetY-4.0GF	4.0	1097.6	79.4
RegNetY-6.4GF	6.4	487.0	79.9
RegNetY-16GF	15.9	351.0	80.4
ViT-B/16	55.5	92.4	77.9
DeiT-S	4.6	1018.2	79.8
DeiT-B	17.6	316.1	81.8
PS-ViT-Ti/14	1.6	1955.3	75.6
PS-ViT-B/10	3.1	1348.0	80.6
PS-ViT-B/14	5.4	765.6	81.7
PS-ViT-B/18	8.8	463.8	82.3

Table 9. Comparison the efficiency of PS-ViT, and that of state-of-the-art networks in terms of FLOPs and speed.

- **각 iteration 마다 현재 sampling 단계의 embedding 은 transformer encoder layer 로 공급**
- **다음 단계의 sampling 위치를 업데이트하기 위해 sampling off-sets group 이 예측됨.**
- **각각의 token에 off-set 모듈을 이용해서 뽑아낸 정보를 position embedding 에 적용을 함**

- Experiments
  - ImageNet 으로 scratch training 시, Vanilla ViT 보다 Top-1 acc 3.8% 증가
  - 4배 적은 파라미터와 10배 적은 FLOPs 달성

- ablation study 포함
- git code 공유
  - https://github.com/yuexy/PS-ViT

CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows

Xiaoyi Dong[1]*, **Jianmin Bao**[2], **Dongdong Chen**[3], **Weiming Zhang**[1], **Nenghai Yu**[1], **Lu Yuan**[3], **Dong Chen**[2], **Baining Guo**[2]
[1]University of Science and Technology of China
[2]Microsoft Research Asia [3]Microsoft Cloud + AI
{dlight@mail., zhangwm@, ynh@}.ustc.edu.cn cddlyf@gmail.com
{jianbao, luyuan, doch, bainguo }@microsoft.com

https://arxiv.org/pdf/2107.00652.pdf

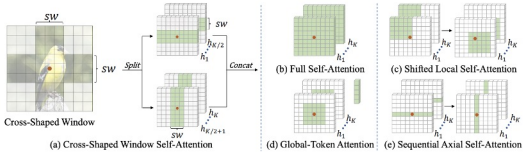


Figure 1: (a) Left: the illustration of the Cross-Shaped Window (CSWin) with stripe width sw for the query point(red dot). Right: the computing of CSWin self-attention, where multi-heads $([h_1, \ldots, h_K])$ is first split into two groups, then two groups of heads perform self-attention in horizontal and vertical stripes respectively, and finally are concatenated together. (b), (c), (d), and (e) are existing self-attention mechanisms.

Transformer architecture but adopt different self-attention mechanisms. The main benefit of the hierarchical design is to utilize the multi-scale features and reduce the computation complexity by progressively decreasing the number of tokens. In this paper,we propose a new hierarchical vision Transformer backbone by introducing cross-shaped window self-attention and locally-enhanced positional encoding.

Figure 3: Comparison among different positional encoding mechanisms: APE and CPE introduce the positional information before feeding into the Transformer blocks, while RPE and our LePE operate in each Transformer block. Different from RPE that adds the positional information into the attention calculation, our LePE operates directly upon V and acts as a parallel module. * Here we only draw the self-attention part to represent the Transformer block for simplicity.

- **Vision Transformer**
  - **self-attention operation 은 permutation-invariant 하기 때문에, 2D 이미지에서 중요한 positional information 을 무시하는 경향이 있음**
    - permutation-invariant: 입력 벡터 요소의 순서와 상관없이 같은 출력을 생성하는 모델, 즉 픽셀의 순서를 고려하지 않음.
  - **2가지 핵심 아이디어 제안**

- **Cross-Shaped Window self-attention mechanism**
  - cross-shaped window 형태: parallel 하게 horizontal and vertical strips 의 self-attention 수행

- **Locally-enhanced Positional Encoding (LePE)**
  - 현재 사용하는 local positional information 을 더 잘 다루는 encoding 기법
  - arbitrary input resolution support
  - especially effective and friendly for downstream tasks

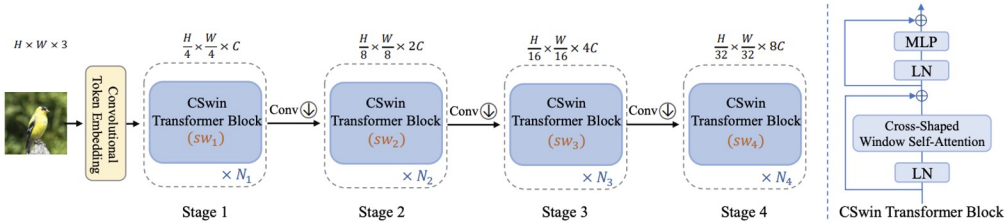


Figure 2: Left: the overall hierarchical architecture of our proposed CSWin Transformer, Right: the illustration of our proposed CSWin Transformer block.

Self-Attention; 2) In order to introduce the local inductive bias, LePE is added as a parallel module to the self-attention branch.

Method	#Param.	Input Size	FLOPs	Top-1	Method	#Param.	Input Size	FLOPs	Top-1
R-101x3 [35]	388M	384^2	204.6G	84.4	R-152x4 [35]	937M	480^2	840.5G	85.4
ViT-B/16 [18]	86M	384^2	55.4G	84.0	ViT-L/16 [35]	307M	384^2	190.7G	85.2
ViL-B [70]	56M	384^2	43.7G	86.2	—	—	—	—	—
Swin-B [39]	88M	224^2 384^2	15.4G 47.1G	85.2 86.4	Swin-L [39]	197M	224^2 384^2	34.5G 103.9G	86.3 87.3
CSWin-B(ours)	78M	224^2 384^2	15.0G 47.0G	85.9 **87.0**	CSWin-L(ours)	173M	224^2 384^2	31.5G 96.8G	86.5 **87.5**

Table 3: ImageNet-1K fine-tuning results by pre-training on ImageNet-21K datasets.

Backbone	#Params (M)	FLOPs (G)	Cascade Mask R-CNN 3x +MS AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Res50 [23]	82	739	46.3	64.3	50.5	40.1	61.7	43.4
Swin-T [39]	86	745	50.5	69.3	54.9	43.7	66.6	47.1
CSWin-T	80	757	**52.5**	**71.5**	**57.1**	**45.3**	**68.8**	**48.9**
X101-32 [64]	101	819	48.1	66.5	52.4	41.6	63.9	45.2
Swin-S [39]	107	838	51.8	70.4	56.3	44.7	67.9	48.5
CSWin-S	92	820	**53.7**	**72.2**	**58.4**	**46.4**	**69.6**	**50.6**
X101-64 [64]	140	972	48.3	66.4	52.3	41.7	64.0	45.1
Swin-B [39]	145	982	51.9	70.9	56.5	45.0	68.4	48.7
CSWin-B	135	1004	**53.9**	**72.6**	**58.5**	**46.4**	**70.0**	**50.4**

Table 5: Object detection and instance segmentation performance on the COCO val2017 with Cascade Mask R-CNN.

Backbone	Semantic FPN 80k #Param.(M)	FLOPs(G)	mIoU(%)	Upernet 160k #Param.(M)	FLOPs(G)	mIoU/MS mIoU(%)
Res50 [23]	28.5	183	36.7	—	—	—/—
PVT-S [59]	28.2	161	39.8	—	—	—/—
TwinsP-S [12]	28.4	162	44.3	54.6	919	46.2/47.5
Twins-S [12]	28.3	144	43.2	54.4	901	46.2/47.1
Swin-T [39]	31.9	182	41.5	59.9	945	44.5/45.8
CSWin-T (ours)	26.1	202	**48.2**	59.9	959	**49.3/50.4**
Res101 [23]	47.5	260	38.8	86.0	1029	—/44.9
PVT-M [59]	48.0	219	41.6	—	—	—/—
TwinsP-B [12]	48.1	220	44.9	74.3	977	47.1/48.4
Twins-B [12]	60.4	261	45.3	88.5	1020	47.7/48.9
Swin-S [39]	53.2	274	45.2	81.3	1038	47.6/49.5
CSWin-S (ours)	38.5	271	**49.2**	64.6	1027	**50.0/50.8**
X101-64 [64]	86.4	—	40.2	—	—	—/—
PVT-L [59]	65.1	283	42.1	—	—	—/—
TwinsP-L [12]	65.3	283	46.4	91.5	1041	48.6/49.8
Twins-L [12]	103.7	404	46.7	133.0	1164	48.8/50.2
Swin-B [39]	91.2	422	46.0	121.0	1188	48.1/49.7
CSWin-B (ours)	81.2	464	**49.9**	109.2	1222	**50.8/51.7**
Swin-B† [39]	—	—	—	121.0	1841	50.0/51.7
Swin-L† [39]	—	—	—	234.0	3230	52.1/53.5
CSWin-B† (ours)	—	—	—	109.2	1941	51.8/52.6
CSWin-L† (ours)	—	—	—	207.7	2745	**54.0/55.7**

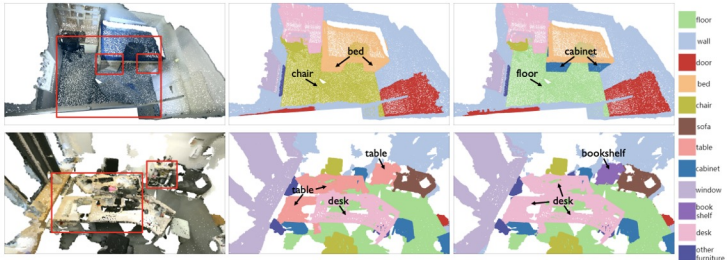
Table 6: Performance comparison of different backbones on the ADE20K segmentation task. Two different frameworks semantic FPN and Upernet are used. FLOPs are calculated with resolution 512×2048. ResNet/ResNeXt results and Swin FPN results are copied from [59] and [12] respectively. † means the model is pretrained on ImageNet-21K and finetuned with 640×640 resolution.

- **Experiments (common vision tasks)**
  - ImageNet-1k (85.4% top-1 accuracy without any extra training data or label)
  - COCO detection task (53.9 box AP and 46.6 mask AP)
  - ADE20k semantic segmentation task (51.7 mIOU)

Learning with Noisy Labels for Robust Point Cloud Segmentation

Shuquan Ye[1] Dongdong Chen[2] Songfang Han[3] Jing Liao[1]*
[1] City University of Hong Kong [2] Microsoft Cloud AI [3] University of California San Diego
shuquanye2-c@my.cityu.edu.hk, cddlyf@gmail.com, s5han@eng.ucsd.edu, jingliao@cityu.edu.hk

https://arxiv.org/pdf/2107.14230.pdf



Input Scenes Real-World Noisy GT Labels Predictions of PNAL

Figure 1. Illustration of the instance-level label noise concept in point cloud segmentation. From left to right are the input (noisy instances highlighted red boxes), the manual annotation given by the real-world dataset ScanNetV2, and the prediction of PNAL (more in line with the real category). It is noticeable that this popular dataset suffers from label noise, such as mislabeling the floor as a chair, even that it is already a re-labeled version of ScanNet. Our PNAL framework is trained on this noisy dataset but still achieves correct predictions.

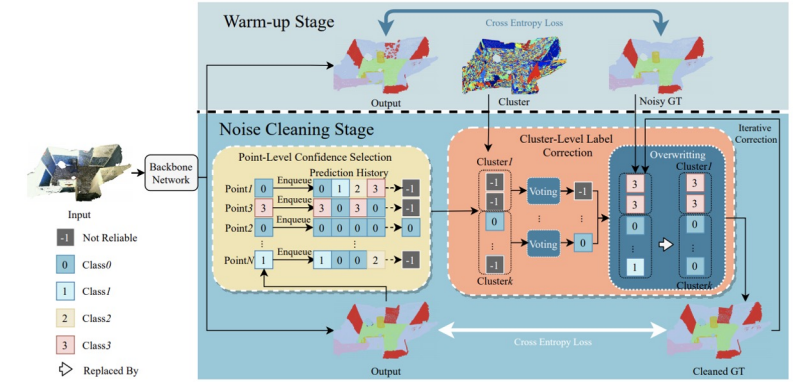


Figure 2. System pipeline. In the warm-up stage, the network is updated with CE as usual. In the noise cleaning stage, we enqueue the output to the prediction history and point wisely perform confidence selection to get reliable labels. With these results, we do voting at cluster level, then correct the original noisy GT or the previously cleaned GT. Finally the obtained cleaned GT guides the network update.

- **Point Cloud Segmentation**
  - 깨끗한 label 가정에 기반한 현재의 deep learning 학습 방법은 noise 가 많은 label 에서 성능이 좋지 못함
  - noise 가 많은 label 에서도 성능이 잘 나올 수 있는 새로운 PNAL (Point Noise-Adaptive Learning) 프레임워크 제안

- **PNAL (Point Noise-Adaptive Learning)**
  - noise-rate blind methods 기반 (기존 방법은 noise-adaptive learning) 을 사용하여 특유의 공간적 변형 noise 속도 문제에 대처
  - 각 point 의 이전 예측을 바탕으로 신뢰할 수 있는 label 을 다시 얻기 위해, point 별 신뢰도 선택을 제안

Methods	0%	Symmetric Noise (τ)				Asymmetric Noise
		20%	40%	60%	80%	$\tau_{pair} = 40\%, \tau = 60\%$
DGCNN[31]+CE	**0.8692**	0.7506	0.6732	0.6390	0.5060	0.5634
DGCNN[31]+SCE[30]	0.7768	0.7524	0.7230	0.6509	0.5705	0.7084
DGCNN[31]+GCE[35]	0.7067	0.7003	0.6997	0.6967	0.6880	0.6614
DGCNN[31]+SELFIE[26]*	0.8673	0.8158	0.7914	0.7725	0.7163	0.7500
DGCNN[31]+PNAL	0.8686	**0.8569**	**0.8378**	**0.8236**	**0.7651**	**0.7968**
PointNet2[22]+CE	**0.8898**	0.7008	0.6796	0.5850	0.5204	0.5648
PointNet2[22]+PNAL	0.8852	**0.8385**	**0.8271**	**0.8067**	**0.7708**	**0.8202**

Table 1. OA Comparison of different methods on artificially created noisy S3DIS. The tops with different backbones are shown in bold.

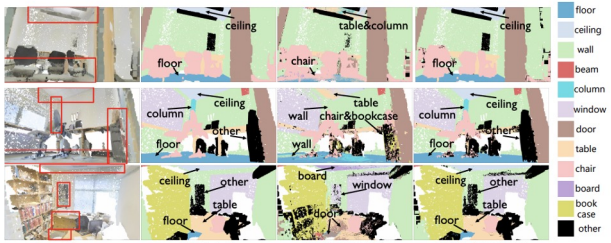


Figure 3. From left to right: Scenes in S3DIS testset, clean GTs, predictions of DGCNN+CE, and DGCNN+PNAL.

- **Experiments**
  - DGCNN이 backbone 인 경우, SELFIE와 PNAL 의 성능 비교
    - PNAL 이 4% 이상의 더 좋은 성능을 보임
  - PointNet2 가 backbone 경우에도 동일한 결과

Methods	real-world noisy ScanNetV2		our re-labeled ScanNetV2	
	mIoU	OA	mIoU	OA
SparseConvNet[11]	0.7250	0.8928	0.7103	0.8807
SparseConvNet[11]+PNAL	0.7298	0.8979	0.7416	0.9211

Table 2. The mIoU and OA comparison on real-world noisy ScanNetV2 validation set and our re-labeled ScanNetV2 validation set.

- ScanNet V2의 Real-world noisy dataset 으로 test 해 보았을 때, dramatic 한 advantage 을 얻지 못함
- dataset 을 re-labeled 한 후, test 해 보니, PNAL 알고리즘의 성능이 더 좋음.

EvilModel: Hiding Malware Inside of Neural Network Models

Zhi Wang, Chaoge Liu, Xiang Cui

https://arxiv.org/pdf/2107.08590.pdf

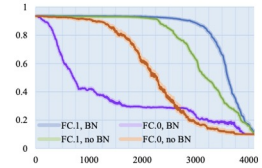


Fig. 5. Accuracy with different neurons replaced on different layers

TABLE III
ACCURACY WITH DIFFERENT NUMBER OF NEURONS REPLACED

Struc.	Initial Acc.	Layer	No. of replaced neurons with Acc.			
			93%	(-1%)	90%	80%
BN	93.75%	FC.1	2105	2285	2900	3290
		FC.0	40	55	160	340
no BN	93.44%	FC.1	1785	2020	2305	2615
		FC.0	220	600	1060	1550

TABLE IV
TESTING ACCURACY ON DIFFERENT MALWARE

	Method	Model	Base	EquationDrug 372KB	ZeusVM 405KB	NSIS 1.7MB	Mamba 2.3MB	WannaCry 3.4MB	VikingHorde 7.1MB	Artemis 12.8MB
EvilModel	Fast Substitution	Vgg19	74.2%	74.2%	74.2%	74.3%	74.2%	74.2%	74.2%	74.2%
		Vgg16	73.4%	73.4%	73.4%	73.4%	73.4%	73.4%	73.4%	73.4%
		Alexnet	56.5%	56.5%	56.5%	56.5%	56.5%	56.4%	56.4%	56.4%
		Resnet101	77.4%	77.3%	77.3%	77.2%	77.1%	77.0%	76.7%	74.5%
		Inception	69.9%	69.9%	69.9%	69.5%	69.6%	69.1%	64.7%	61.3%
		Resnet18	69.8%	69.7%	69.6%	69.2%	69.1%	68.9%	67.4%	60.3%
		Mobilenet	71.9%	71.0%	71.1%	68.5%	**60.6%**	**39.7%**	**0.1%**	-
StegoNet	LSB Substitution	Inception	78.0%	78.2%	77.9%	78.0%	78.3%	78.2%	78.1%	77.3%
		Resnet18	70.7%	69.3%	71.2%	70.5%	72.1%	71.3%	69.3%	61.3%
		Mobilenet	70.9%	**0.2%**	**0.2%**	**0.2%**	**0.2%**	**0.1%**	-	-
	Resilience Training	Inception	78.0%	78.3%	78.4%	78.4%	77.6%	78.4%	77.8%	78.1%
		Resnet18	70.7%	71.1%	71.2%	70.4%	70.9%	71.3%	68.2%	69.7%
		Mobilenet	70.9%	71.2%	68.5%	**32.5%**	**6.1%**	**0.7%**	-	-
	Value-Mapping	Inception	78.0%	78.3%	78.4%	77.2%	78.4%	78.1%	77.6%	77.3%
		Resnet18	70.7%	71.1%	70.2%	72.1%	71.0%	70.4%	70.3%	70.9%
		Mobilenet	70.9%	69.2%	71.0%	**54.7%**	**49.3%**	-	-	-
	Sign-Mapping	Inception	78.0%	77.4%	78.2%	78.0%	-	-	-	-
		Resnet18	70.7%	71.1%	70.8%	69.5%	-	-	-	-
		Mobilenet	70.9%	68.3%	-	-	-	-	-	-

- **Malware 를 Neural Network model 을 통해 전파하는 새로운 방법 제안**
  - malware 를 neurons 에 embedding 함으로써, neural network performance 에는 거의 영향을 미치지 않으면서도
  - malware 가 전파될 수 있음
  - 추가적으로, neural network model 구조는 변하지 않기 때문에, antivirus engines security scan을 통과할 수 있음

- Experiments
  - 36.9MB의 malware가 178MB-Alexnet model에 embedded 되었을 때, 1% accuracy loss + VirusTotal 의 anti-virus 에 no-suspicious

- Conclusion
  - AI application이 퍼짐에 따라, NN을 사용한 공격을 앞으로 trend 가 될 것임
  - NN으로 발생하는 공격에 대응하기 위한 reference scenario가 되길 바람

End of the Document

Towards Robust Classification Model by Counterfactual and Invariant Data Generation

https://arxiv.org/pdf/2106.01127v2.pdf

Towards Robust Classification Model by Counterfactual and Invariant Data Generation

Chun-Hao Chang, George Alexandru Adam, Anna Goldenberg
University of Toronto, Vector Institute, The Hospital for Sick Children
{kingsley.chang,alex.adam}@mail.utoronto.ca,anna.goldenberg@utoronto.ca

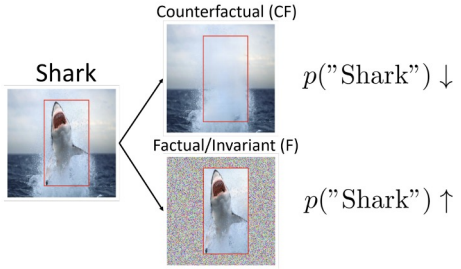


Figure 1: We propose 2 data augmentations: Counterfactual (CF) and Factual (F). CF image keeps backgrounds and reduces the probability of target class, while F image keeps foreground and increases the probability.

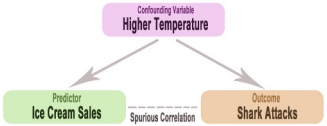
"만약 Shark가 없었더라면 Shark라고 Label을 주지 않았을 텐데"

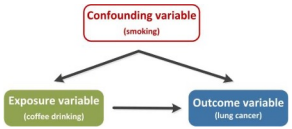
"Shark가 있으므로 Shark라고 Label을 주어야지"

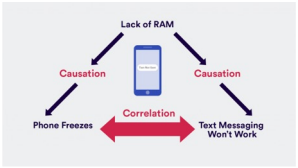
"Correlation Doesn't Imply Causation" (상관관계는 인과관계(원인)를 의미하지 않는

- **Superious Relationship**

"우연의 일치 또는 보이지 않는 특정 세 번째 요인으로 인해 둘 이상의 이벤트 또는 변수가 연관되어 있지만 인과관계가 없는 수학적 관계"







Superious Relationship Issue of Machine Learning Field

"일부 Feature는 Label과 상관관계는 있지만 인과관계는 없다."
-> 이에 Bounding Box 를 이용해 2가지 Data Augmentation

(이 외에도 다음과 같은 Issue 존재 : artifacts, lack of robustness, discrimination)

- **Counterfactual Explanations (반사실적 설명)**
  "x가 벌어지지 않았다면 Y는 발생하지 않았을 것이다."

  e.g, 만약 내가 뜨거운 커피를 한 모금도 마시지 않았다면, 나는 내 혀에 화상을 입지 않았을 것이다.
  이 때, 실제(= 커피를 마신상황) 상황과 다른 관찰된 사실 (= 반사실적 상황)과 모순된 가상의 현실을 상상해야 하므로 반사실적 "Conuterfactual"이라는 단어를 사용.

Towards Robust Classification Model by Counterfactual and Invariant Data Generation

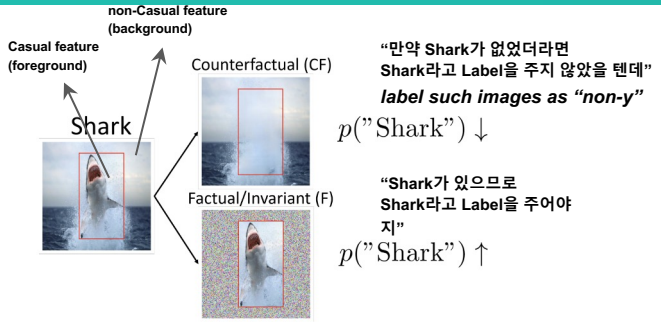


Figure 1: We propose 2 data augmentations: Counterfactual (CF) and Factual (F). CF image keeps backgrounds and reduces the probability of target class, while F image keeps foreground and increases the probability.

a causal region $\boldsymbol{r} \subseteq \{0,1\}^U$ (1 means causal).
image $\boldsymbol{x}$ with U pixels
a label y

To break the correlation between non-casual features (background) and labels
= keep background + remove foregrounds

$$\phi_{cf}(\boldsymbol{x}, \boldsymbol{r}) = (1-\boldsymbol{r}) \odot \boldsymbol{x} + \boldsymbol{r} \odot \hat{\boldsymbol{x}} \quad , \text{where } \hat{\boldsymbol{x}} \sim p_{\text{infill}}(\hat{\boldsymbol{x}} | \boldsymbol{x}_{\boldsymbol{r}=\boldsymbol{0}}).$$

$$L_{cf} = (-log(1 - P_f(\hat{y} = c | \phi_{cf}(x))))$$

To make a classifier immune to background shift

$$\Phi_f(\boldsymbol{x}, \boldsymbol{r}) = \boldsymbol{r} \odot \boldsymbol{x} + (1-\boldsymbol{r}) \odot \hat{\boldsymbol{x}} \quad \text{where } \hat{\boldsymbol{x}} \sim p_{\text{infill}}(\hat{\boldsymbol{x}} | \boldsymbol{x}_{\boldsymbol{r}=\boldsymbol{1}}).$$

- **3 different option of counterfactual loss function**
  1. **Negative log likelihood (better than 3)**
     $P(\hat{y} \neq y)$ i.e. $-log(1 - P(\hat{y} = y|x))$;
  1. **KL divergence between the uniform distribution and the predicted probability (worst)**
     $\text{KL}(\text{Uniform}(y) || P(\hat{y}|x))$.
  1. **KL divergence between the uniform distribution except original class y and the predicted prob.**

final objective (with cross entropy loss)

$$L = L_{CE}(y, \hat{y}(x)) + L_{cf} + \overbrace{L_{CE}(y, \hat{y}(\Phi_f(\boldsymbol{x}, \boldsymbol{r}))}^{\text{Factual Loss}}$$

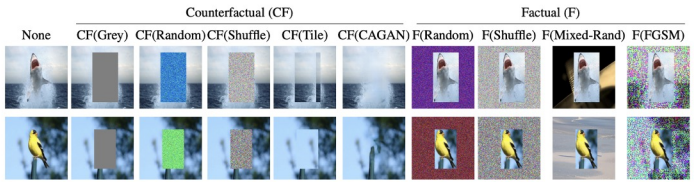


Figure 2: Examples of Counterfactual (CF) or Factual (F) data generation.

Choice of infilling value (CF)

- **Grey : 0.5, which is 0 after being normalized [-1,1]**
- **Random : uniform distribution that resembles low-frequency, and adds Gasussians of gamma=0.2 per-channel per-pixel as high-frequency noise.**
- **Shuffle: randomly shuffle all pixel values in the specified region;**
- **Tile : first extracts the largest rectangle from the background that does not intersect with the foreground**
- **CAGAN(Authors' pretrained Contextual Attention GAN)**

Choice of infilling value (F)

- **Mixed-Rand : randomly-chosen tiled background from images of other classes within the same training batch.**
- **FGSM : adversarial attach only on the background region.**

- **Contribution**

1. We use various **counterfactual and invariant data generations to augment training datasets** which makes models more robust to spurious correlations
2. We show that **our augmentations lead to similar or better accuracy than state-of-art saliency regularization and other robustness baseline on challenging dataset in the presence of background shifts.** We also find combining our augmentations with saliency regularization can further improve performance.
3. Our methods have **stronger salience focus on casual features that provide better explanations**, although we find strong salience on casual features only correlates weakly with good generalization.

Towards Robust Classification Model by Counterfactual and Invariant Data Generation

- **IN-9 Dataset : a synthetic perturbed background challenge dataset (link)**
- **(a) (b) same class bg | [c] [d] diff class bg**

Figure 3: Examples of IN9 test sets. Original and Mixed-Same have backgrounds of the same class; Mixed-Rand and Mixed-Next have backgrounds of different classes.

- **Experimental Results**

1. **Do they improve the accuracy under shifted distribution?**
2. **Do they make model focus more on foregrounds instead of backgrounds measured by saliency map?**
3. **Does focusing on foregrounds indicate better accuracy?**
4. **Do our augmentations make model's predictions less affected by changed background?**

Mobile-Former: Bridging MobileNet and Transformer

https://arxiv.org/pdf/2108.05895.pdf

Mobile-Former: Bridging MobileNet and Transformer

Yinpeng Chen[1] Xiyang Dai[1] Dongdong Chen[1] Mengchen Liu[1] Xiaoyi Dong[2]
Lu Yuan[1] Zicheng Liu[1]

[1] Microsoft [2] University of Science and Technology of China

{yiche,xidai,dochen,mengcliu,luyuan,zliu}@microsoft.com, dlight@mail.ustc.edu.cn

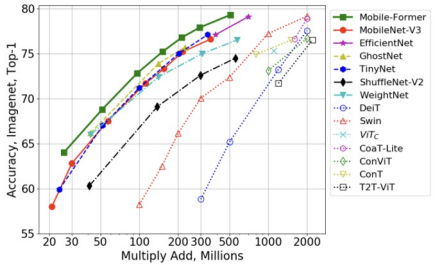


Figure 2. **Comparison among Mobile-Former, efficient CNNs and Vision Transformers**, in terms of accuracy-flop tradeoff. The comparison is performed on ImageNet classification. Mobile-Former consistently outperforms both efficient CNNs and vision transformers in low FLOP regime (from 25M to 500M MAdds). Note that we implement Swin [20] and DeiT [29] at low computational budget from 100M to 2G FLOPs. Best viewed in color.

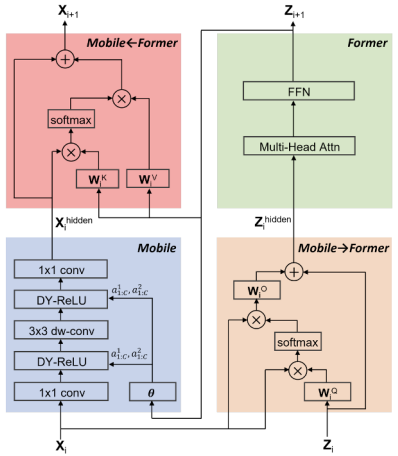
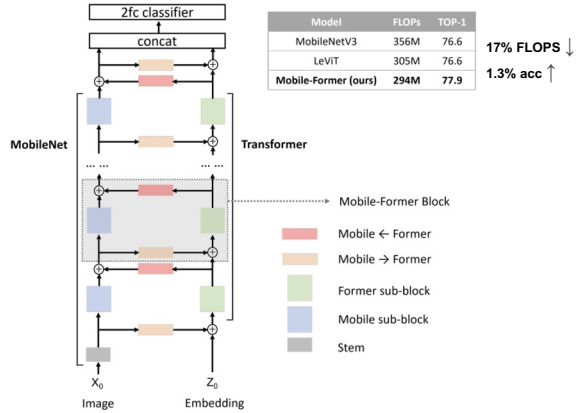


Figure 3. **Mobile-Former Block** that includes four modules: ***Mobile*** sub-block modifies inverted bottleneck block in [24] by replacing ReLU with dynamic ReLU [3]. ***Mobile→Former*** uses light-weight cross attention to fuse local features into global features. ***Former*** sub-block is a standard transformer block including multi-head attention and FFN. Note that the output of *Former* is used to generate parameters for dynamic ReLU in *Mobile* sub-block. ***Mobile←Former*** bridges from global to local features.



Model	FLOPs	TOP-1
MobileNetV3	356M	76.6
LeViT	305M	76.6
Mobile-Former (ours)	294M	77.9

Figure 1. **Overview of Mobile-Former**, which parallelizes MobileNet [24] (on the left side) and Transformer [33] (on the right side). Different with vision transformer [8] that uses image patches to form tokens, the transformer in Mobile-Former takes *very few learnable tokens* as input that are randomly initialized. *Mobile* (refers to MobileNet) and *Former* (refers to transformer) are communicated by a bidirectional bridge, which is modeled by the proposed light-weight across attention. Best viewed in color.

"How to design efficient networks to **efficient** networks to **effectively** encode both local processing and global interaction?"

1. **a parallel design of MobileNet and Transformer (two-way bidirectional bridge)**
2. **MobileNet (local feature) + Transformer (global feature) => more representation power**
3. **Mobile-Former contains very few tokens (e.g, less than 6 tokens) => low computational cost**
4. **proposed light-weight cross attention to model the bridge**
5. **Mobile-Former는 25M to 500M FLOPs on ImageNet Classification (lower than MobileNet v3)**
6. **기존 MobileNet v3보다 8.6AP ↑ on Object detection**

Mobile-Former: Bridging MobileNet and Transformer

https://arxiv.org/pdf/2108.05895.pdf

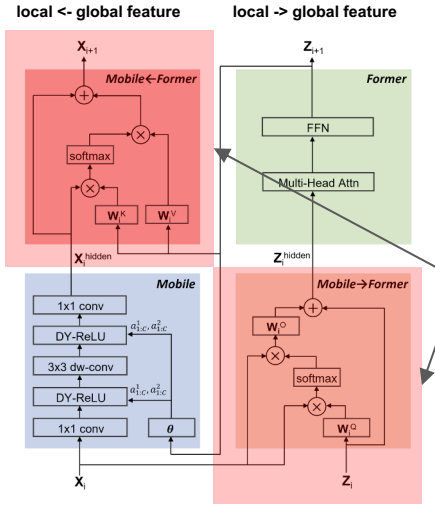


Figure 3. **Mobile-Former Block** that includes four modules: ***Mobile*** sub-block modifies inverted bottleneck block in [24] by replacing ReLU with dynamic ReLU [3]. ***Mobile→Former*** uses light-weight cross attention to fuse local features into global features. ***Former*** sub-block is a standard transformer block including multi-head attention and FFN. Note that the output of *Former* is used to generate parameters for dynamic ReLU in *Mobile* sub-block. ***Mobile←Former*** bridges from global to local features.

1. Low cost Two-way Bridge (Cross attention)

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V.$$

local features

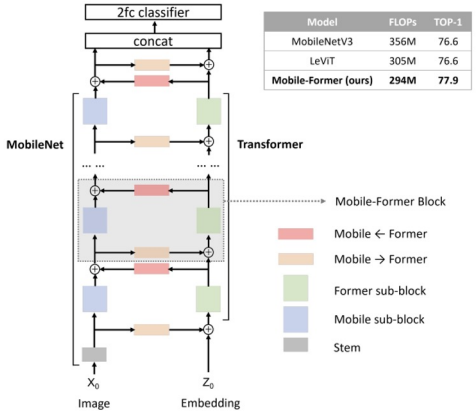
$$\boldsymbol{z}^{out} = \boldsymbol{z} + \Big[Attention(\boldsymbol{z}_h\boldsymbol{W}_h^Q, \boldsymbol{x}_h, \boldsymbol{x}_h)\Big]_{h=1:H}\boldsymbol{W}^O,$$

$$\boldsymbol{x}^{out} = \boldsymbol{x} + \Big[Attention(\boldsymbol{x}_h, \boldsymbol{z}_h\boldsymbol{W}_h^K, \boldsymbol{z}_h\boldsymbol{W}_h^V)\Big]_{h=1:H}, \quad (3)$$

from global tokens

Table 1. **Specification for Mobile-Former-294M**. "bneck-lite" denotes light bottleneck block. "Mobile-Former$^{\downarrow}$" denotes the Mobile-Former block for downsampling.

Stage	Input	Operator	exp size	#out	Stride
tokens	6×192	–	–	–	–
stem	$224^2\times3$	conv2d, 3×3	–	16	2
1	$112^2\times16$	bneck-lite	32	16	1
2	$112^2\times16$	Mobile-Former$^{\downarrow}$	96	24	2
	$56^2\times24$	Mobile-Former	96	24	1
3	$56^2\times24$	Mobile-Former$^{\downarrow}$	144	48	2
	$28^2\times48$	Mobile-Former	192	48	1
4	$28^2\times48$	Mobile-Former$^{\downarrow}$	288	96	2
	$14^2\times96$	Mobile-Former	384	96	1
	$14^2\times96$	Mobile-Former	576	128	1
	$14^2\times128$	Mobile-Former	768	128	1
5	$14^2\times128$	Mobile-Former$^{\downarrow}$	768	192	2
	$7^2\times192$	Mobile-Former	1152	192	1
	$7^2\times192$	Mobile-Former	1152	192	1
	$7^2\times192$	conv2d, 1×1	–	1152	1
head	$7^2\times1152$	pool, 7×7	–	–	1
	$1^2\times1152$	concat w/ cls token	–	1344	1
	$1^2\times1344$	FC	–	1920	1
	$1^2\times1920$	FC	–	1000	1



Model	FLOPs	TOP-1
MobileNetV3	356M	76.6
LeViT	305M	76.6
Mobile-Former (ours)	**294M**	**77.9**

Figure 1. **Overview of Mobile-Former**, which parallelizes MobileNet [24] (on the left side) and Transformer [33] (on the right side). Different with vision transformer [8] that uses image patches to form tokens, the transformer in Mobile-Former takes *very few learnable tokens* as input that are randomly initialized. *Mobile* (refers to MobileNet) and *Former* (refers to transformer) are communicated by a bidirectional bridge, which is modeled by the proposed light-weight across attention. Best viewed in color.

Why? Low Computational Cost?

- **Expansion ratio 2 instead of 4 in FFN**
- **The Number of VIT tokens is only 6**
- **Cross Attention**

- **11 Mobile-Former blocks at different input resolutions.**
- **Mobile-Former blocks have 6 global tokens with dimension 192.**
- **2-5 stage has a downsample Mobile-Former block**

Mobile-Former: Bridging MobileNet and Transformer

https://arxiv.org/pdf/2108.05895.pdf

ImageNet Classification

Model	Input	#Param	MAdds	Top-1
MobileNetV3 Small 1.0× [14]	160^2	2.5M	30M	62.8
Mobile-Former-26M	224^2	3.2M	**26M**	**64.0**
MobileNetV3 Small 1.0× [14]	224^2	2.5M	57M	67.5
Mobile-Former-52M	224^2	3.5M	**52M**	**68.7**
MobileNetV3 1.0× [14]	160^2	5.4M	112M	71.7
Mobile-Former-96M	224^2	4.6M	**96M**	**72.8**
ShuffleNetV2 1.0× [23]	224^2	2.2M	**138M**	69.1
ShuffleNetV2 1.0×+WeightNet 4× [22]	224^2	5.1M	141M	72.4
MobileNetV3 0.75× [14]	224^2	4.0M	155M	73.3
Mobile-Former-151M	224^2	7.6M	151M	**75.2**
MobileNetV3 1.0× [14]	224^2	5.4M	217M	75.2
Mobile-Former-214M	224^2	9.4M	**214M**	**76.7**
ShuffleNetV2 1.5× [23]	224^2	3.5M	299M	72.6
ShuffleNetV2 1.5×+WeightNet 4× [22]	224^2	9.6M	307M	75.0
MobileNetV3 1.25× [14]	224^2	7.5M	356M	76.6
EfficientNet-B0 [26]	224^2	5.3M	390M	77.1
Mobile-Former-294M	224^2	11.4M	**294M**	**77.9**
ShuffleNetV2 2× [23]	224^2	5.5M	557M	74.5
ShuffleNetV2 2×+WeightNet 4× [22]	224^2	18.1M	573M	76.5
Mobile-Former-508M	224^2	14.0M	**508M**	**79.3**

Table 2. **Comparing Mobile-Former with efficient CNNs** evaluated on ImageNet [6] classification.

Model	Input	#Param	MAdds	Top-1
T2T-ViT-7 [40]	224^2	4.3M	1.2G	71.7
DeiT-Tiny [29]	224^2	5.7M	1.2G	72.2
ConViT-Tiny [5]	224^2	6.0M	1.0G	73.1
ConT-Ti [38]	224^2	5.8M	0.8G	74.9
ViT$_C$ [36]	224^2	4.6M	1.1G	75.3
ConT-S [38]	224^2	10.1M	1.5G	76.5
Swin-1G [20] ‡	224^2	7.3M	1.0G	77.3
Mobile-Former-294M	224^2	11.4M	**294M**	**77.9**
PVT-Tiny [34]	224^2	13.2M	1.9G	75.1
T2T-ViT-12 [40]	224^2	6.9M	2.2G	76.5
CoaT-Lite Tiny [37]	224^2	5.7M	1.6G	76.6
ConViT-Tiny+ [5]	224^2	10.0M	2G	76.7
DeiT-2G [29] ‡	224^2	9.5M	2.0G	77.6
CoaT-Lite Mini [37]	224^2	11.0M	2.0G	78.9
BoT-S1-50 [25]	224^2	20.8M	4.3G	79.1
Swin-2G [20] ‡	224^2	12.8M	2.0G	79.2
Mobile-Former-508M	224^2	14.0M	**508M**	**79.3**

Table 3. **Comparing Mobile-Former with Vision Transformer variants** evaluated on ImageNet [6] classification. Here, we choose ViT variants that use image resolution 224×224 and are trained *without* distillation from a teacher network. We split ViT models based on FLOPs (using 1.5G as threshold) and rank them based on top-1 accuracy. ‡ indicates our implementation.

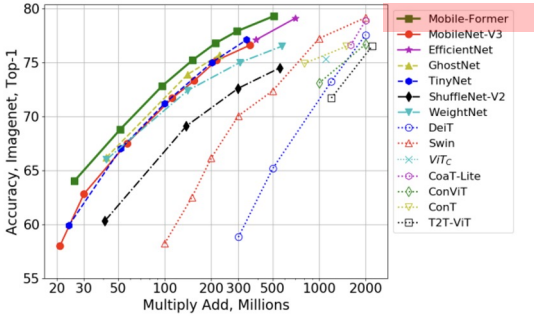


Figure 2. **Comparison among Mobile-Former, efficient CNNs and Vision Transformers**, in terms of accuracy-flop tradeoff. The comparison is performed on ImageNet classification. Mobile-Former consistently outperforms both efficient CNNs and vision transformers in low FLOP regime (from 25M to 500M MAdds). Note that we implement Swin [20] and DeiT [29] at low computational budget from 100M to 2G FLOPs. Best viewed in color.

Mobile-Former: Bridging MobileNet and Transformer

https://arxiv.org/pdf/2108.05895.pdf

Object Detection

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	MAdds (G) backbone	MAdds (G) all	#Params (M) backbone	#Params (M) all
ShuffleNetV2 [23]	25.9	41.9	26.9	12.4	28.0	36.4	2.6	161	0.8	10.4
Mobile-Former-151M	**34.2**	53.4	36.0	19.9	36.8	45.3	**2.4**	161	4.9	14.4
MobileNetV3 [14]	27.2	43.9	28.3	13.5	30.2	37.2	4.7	162	2.8	12.3
Mobile-Former-214M	**35.8**	55.4	38.0	21.8	38.5	46.8	**3.6**	162	5.7	15.2
ResNet18 [12]	31.8	49.6	33.6	16.3	34.3	43.2	29	181	11.2	21.3
Mobile-Former-294M	**36.6**	56.6	38.6	21.9	39.5	47.9	**5.2**	164	6.5	16.1
ResNet50 [12]	36.5	55.4	39.1	20.4	40.3	48.1	84	239	23.3	37.7
PVT-Tiny [34]	36.7	56.9	38.9	22.6	38.8	50.0	70	221	12.3	23.0
ConT-M [38]	37.9	58.1	40.2	23.0	40.6	50.4	65	217	16.8	27.0
Mobile-Former-508M	**38.0**	58.3	40.3	22.9	41.2	49.7	**9.4**	168	8.4	17.9

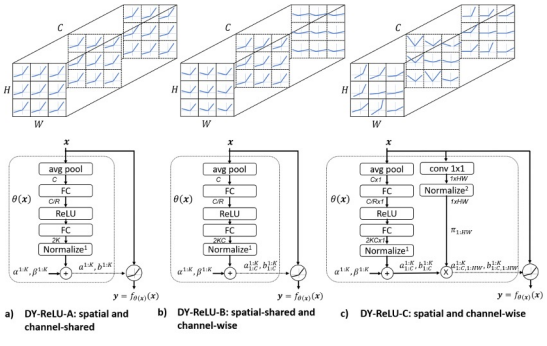
Table 4. **COCO object detection results**. All models are trained on `train2017` and tested on `val2017`. All models are trained for 12 epochs (1×) from ImageNet pretrained weights.

I think using "DY-ReLU" is better in MobileNet's inverted bottleneck

Model	#Param	MAdds	Top-1	Top-5
Mobile (using ReLU)	6.1M	259M	74.2	91.8
+ *Former* and Bridge	10.1M	290M	$76.8_{(+2.6)}$	$93.2_{(+1.4)}$
+ DY-ReLU in *Mobile*	11.4M	294M	$77.8_{(+1.0)}$	$93.7_{(+0.5)}$

Table 5. **Ablation of Former, Bridge and DY-ReLU** evaluated on ImageNet classification. Mobile-Former-294M is used.

"DY-ReLU"



Mobile-Former: Bridging MobileNet and Transformer

Mobile-Former is Explainable

(Mobile->Former) is normalized pixels, showing focused region per token.
(Mobile<-Former) is normalized over tokens showing the contribution per token at each pixel.

https://arxiv.org/pdf/2108.05895.pdf

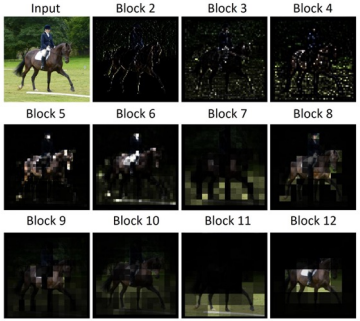


Figure 5. **Cross attention over featuremap for the first token in *Mobile→Former*** across all Mobile-Former blocks. Attention is normalized over pixels, showing focused regions. The focused regions change from low to high levels. The token starts paying more attention to edges/corners at block 2–4. Then it focused more on bigger region rather than scattered small pieces at block 5–12. The focused region shifts between the foreground (person and horse) and background (grass). Finally, it locks the most discriminative part (horse body and head) for classification.

Low level -> High level

- 2-4 blocks = edges/corners
- 5-12 = more bigger region
- finally = criminative part (horse)

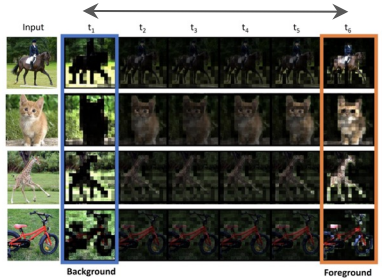
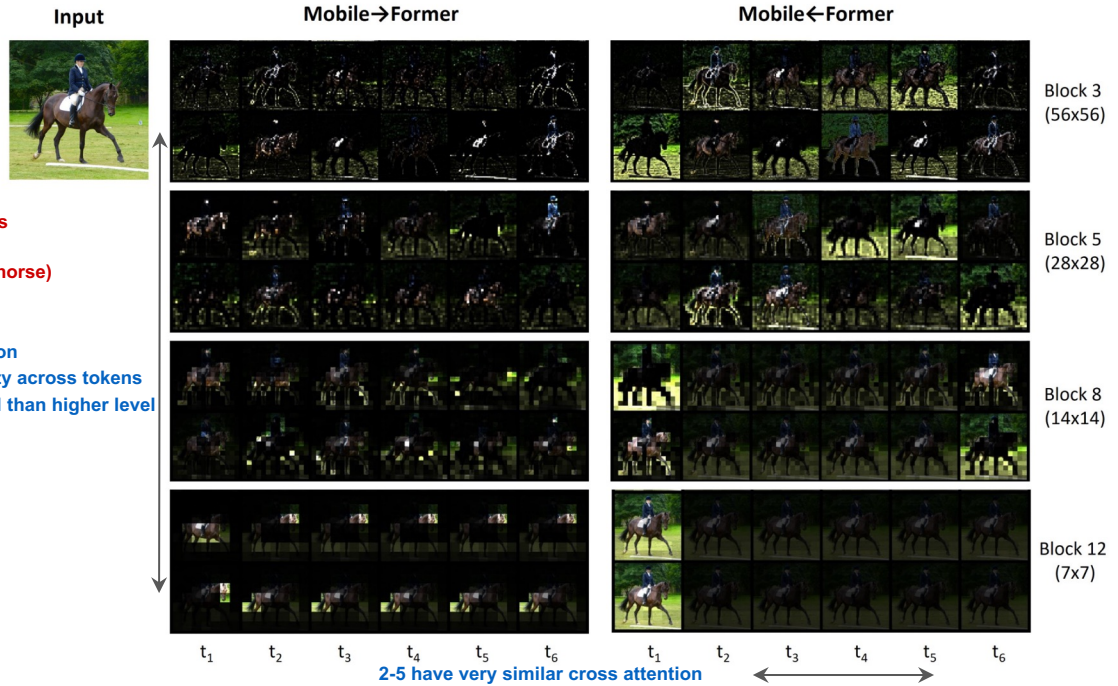


Figure 6. **Cross attention in *Mobile←Former* separates foreground and background at middle layers**. Attention is normalized over tokens showing the contribution of different tokens at each pixel. Block 8 is chosen where the background pixels pay more attention to the first token and the foreground pixels pay more attention to the last token.

In Block 8,

- first token : attention bg
- last token : attention fg

cross attention
more diversity across tokens
at lower level than higher level



2-5 have very similar cross attention

Figure 4. **Visualization of cross attention** on the two-way bridge: *Mobile→Former* and *Mobile←Former*. Mobile-Former-294M is used, which includes 6 tokens (each corresponds to a column). Four blocks with different input resolutions are selected and each has two attention heads that are visualized in two rows. Attention in *Mobile→Former* (left half) is normalized over pixels, showing the focused region per token. Attention in *Mobile←Former* (right half) is normalized over tokens showing the contribution per token at each pixel. The cross attention has more diversity across tokens at lower levels than higher levels. At the last block, token 2–5 have very similar cross attention.

Eyes Tell All: Irregular Pupil Shapes Reveal GAN-generated Faces

https://arxiv.org/pdf/2109.00162v1.pdf

EYES TELL ALL: IRREGULAR PUPIL SHAPES REVEAL GAN-GENERATED FACES

Hui Guo[1], Shu Hu[2], Xin Wang[3], Ming-Ching Chang[1], Siwei Lyu[2]

[1]University at Albany, SUNY, USA. {hguo,mchang2}@albany.edu
[2]University at Buffalo, SUNY, USA. {shuhu,siweilyu}@buffalo.edu
[3]Keya Medical, Settle, USA. xinw@keyamedna.com

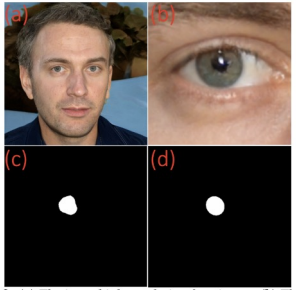
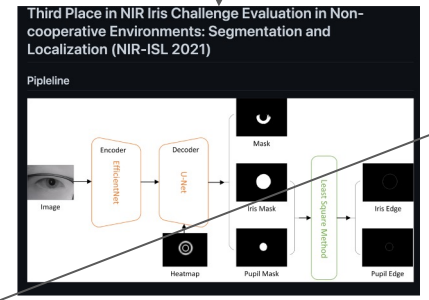


Fig. 2: ***(a)*** *The input high-resolution face image,* ***(b)*** *The the cropped eye image using landmarks,* ***(c)*** *Predicted pupil mask of image (b),* ***(d)*** *Ellipse fitted pupil mask of (c). Note that this example is a GAN-generated face.*

- **Pupil Segmentation and Boundary Detection**

1. **face detector [a]**
2. **two eyes are properly cropped [b]**
   (using DLlib)
3. **the inner and outer boundary mask of both pupil and iris [c]**
   (using EyeCool segments)
4. **Ellipse Fitted Pupil Mask [d]**
   (Square-based ellipse fitting method)



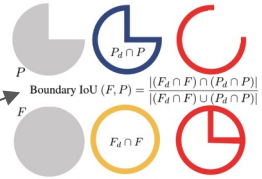


Fig. 3: *A toy example to explain the Boundary IoU.* ***Left:*** *The predicted pupil mask P and the ellipse fitted pupil mask F.* ***Middle:*** *P_d and F_d are the mask pixels within distance d from the boundaries (blue and yellow).* ***Right:*** *Boundary IoU calculation between predicted pupil mask and the ellipse fitted pupil mask with distance parameter d.*

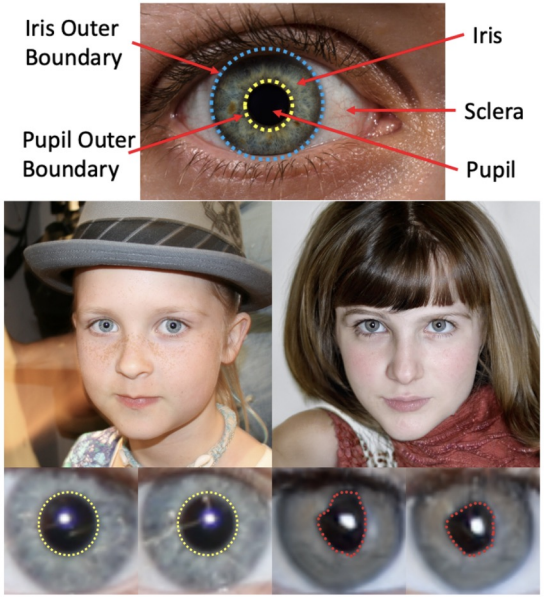


Fig. 1: ***Top:*** *Anatomy structures of a human eye.* ***Bottom:*** *Examples of pupils of real human (left) and GAN-generated (right). Note that the pupils for the real eyes have a strong circular or elliptical shapes (**yellow**) while those for the GAN-generated pupils are with irregular shapes (**red**). And also the shapes of both pupils are very different from each other in the GAN-generated face image.*

- GAN 생성 얼굴의 불규칙한 눈동자 모양을 통해 실제 얼굴과 GAN으로 생성된 얼굴을 구별하는 지표가 될 ㅅ
- BIuU Score로 GAN으로 생성된 사람 얼굴과 실제 사람얼굴을 동공으로 구별하겠다고 주장
- 한계: 실제 사람의 동공도 모종의 병 때문에 동공이 원의 형태가 아닌 경우도 있음.

Eyes Tell All: Irregular Pupil Shapes Reveal GAN-generated Faces

실제 사진
(B IoU high)

StyleGAN2
(B IoU low)

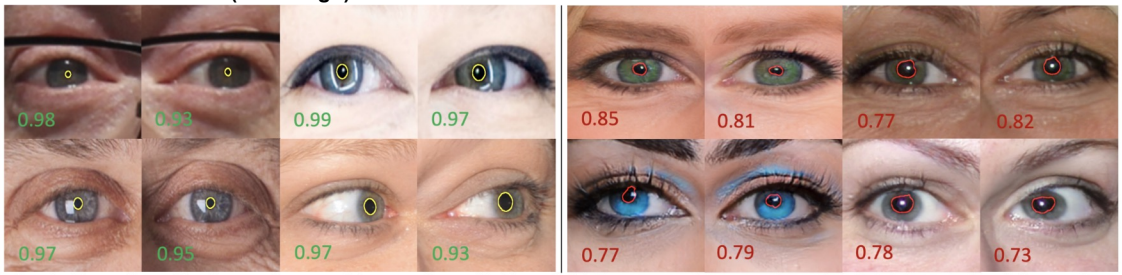


Fig. 4: *Examples of both eyes from real human faces (**left**) and GAN generated human faces (**right**). The pixels of the predicted pupil mask within a distance $d = 4$ from the prediction boundary contours are highlighted. The Boundary IoU score ($d = 4$) between the predicted pupil mask and the ellipse-fitted pupil mask for each pupil is shown on the images.*

https://arxiv.org/pdf/2109.00162v1.pdf

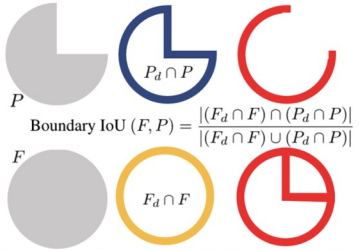


Fig. 3: *A toy example to explain the Boundary IoU. **Left:** The predicted pupil mask P and the ellipse fitted pupil mask F. **Middle:** P_d and F_d are the mask pixels within distance d from the boundaries (blue and yellow). **Right:** Boundary IoU calculation between predicted pupil mask and the ellipse fitted pupil mask with distance parameter d.*

- **Dataset**
  - **FilckrFaces-HQ (FFHQ) dataset**
  - **GAN-generated human faces are created by StyleGAN2 (1000 images 1024*1024)**

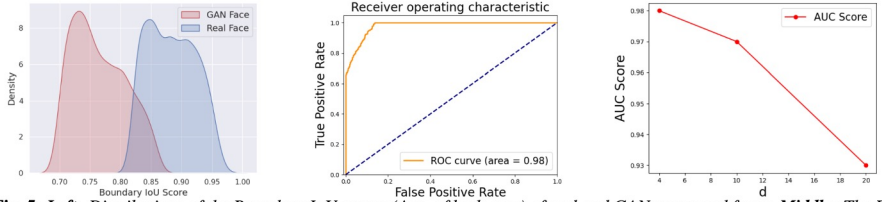


Fig. 5: ***Left:** Distributions of the Boundary IoU scores (Ave. of both eyes) of real and GAN-generated faces. **Middle:** The ROC curve is based on the Boundary IoU scores. The $d = 4$ for both figures. **Right:** BIoU hyper-parameter analysis, where x axis indicates the variation of hyper-parameter d and y axis is the AUC score.*

- **Boundary IoU**
  - **얼마나 두 객체간 boundary 가 겹치는가 ?**
  - **detected pipil mask v.s. ellipse fitted pupil mask**
  - **[0,1] : 1에 가까울수록 겹치는 영역이 많음 (= pupil이 타원에 가까움)**
  - **distance parameter d.(distance to boundary that contorls the measure's sentitivity to the boundary)**

M- Mask IoU 사용시 pixel equally, Thus, it is less sensitive to boundary quality.

1. **MIoU calculation을 이용해 확인한 결과, GAN vs Real 둘사이의 B IoU 분포차이가 존재 (Avg, of both eyes)**

- **artifacts of irregular pupil shapes lead to siginificantly lower BIoU scores.**

1. **In ROC Curve based on the Boundary IoU scores, AUC 0,96로 irregular pupil shape가 GAN에서 생성된 얼굴을 식별하는데 효과적이라는 것**
2. **d 에 따른 AUC Score (B IoU를 계산하기 위한 hyper parameter d=4 가 가장 optimal)**

- **d가 충분히 커지면 BIoU 는Mask IoU와 동일한 효과, Mask IoU를 계산하기 위한 Pixel이 충분히 많아져 B IoU는 Mask IoU와 같이 boundary sensitivity 에 덜 민감 (d가 증가할 수록 AUC Score가**

- **해당 지표가 GAN과 Real이미지를 구분하기 힘든 limitation sample**

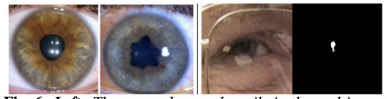


Fig. 6: ***Left:** There are abnormal pupils in the real images with non-elliptical shapes due to diseases and infection at pupil and iris regions, these real images examples are from [31]. **Right:** Occlusions, noises around the pupil and fail pupil segmentation.*

Do Vision Transformers See Like Convolutional Neural Networks?

https://arxiv.org/pdf/2108.08810.pdf

Do Vision Transformers See Like Convolutional Neural Networks?

Maithra Raghu[1] Thomas Unterthiner[1] Simon Kornblith[1] Chiyuan Zhang[1] Alexey Dosovitskiy[1]

[1]*Google Research, Brain Team*

- We investigate the internal representation structure of ViTs and CNNs, finding striking differences between the two models, such as ViT having more uniform representations, with greater similarity between lower and higher layers.
- Analyzing how local/global spatial information is utilised, we find ViT incorporates more global information than ResNet at lower layers, leading to quantitatively different features.

Visual data (image classification tasks)

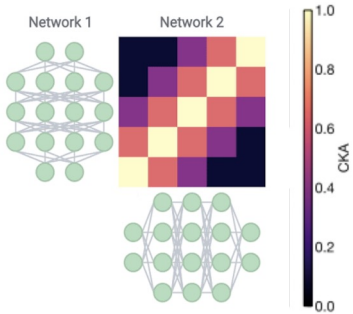
- CNNs → de-facto model
- ViT → achieve comparable or even superior performance

how are Vision Transformers solving these tasks?
uncovering insights about key differences between ViTs and CNNs

- ***CNN 과 비슷하게 작동하는가? 완전히 다른 visual representations을 학습하는가?***

두 아키텍처 간의 차이 연구

- Nevertheless, we find that incorporating local information at lower layers remains vital, with large-scale pre-training data helping early attention layers learn to do this
- We study the uniform internal structure of ViT, finding that skip connections in ViT are even more influential than in ResNets, having strong effects on performance and representation similarity.
- Motivated by potential future uses in object detection, we examine how well input spatial information is preserved, finding connections between spatial localization and methods of classification.
- We study the effects of dataset scale on transfer learning, with a linear probes study revealing its importance for high quality intermediate representations.



CKA (Centered Kernel Alignment)
layer 간 representation similarity 비교

- single model
- across model (네트워크 1과 네트워크 2가 random initializations 으로 학습 되었다거나, 다른 아키텍쳐를 가졌을 경우)

input

- representations from two layers

output

- a similarity score between 0 (not at all similar) and 1 (identical representations)

Do Vision Transformers See Like Convolutional Neural Networks?

두 아키텍처 간의 차이 연구

1. **Single model 비교**
   - ViT 가 전반적인 layer 에 걸쳐 uniform representations 을 가짐. (lower layer와 higher layer 간 유사도가 높음)
   - ResNet 모델의 경우, lower layer 와 higher layer 의 similarity 가 낮음.

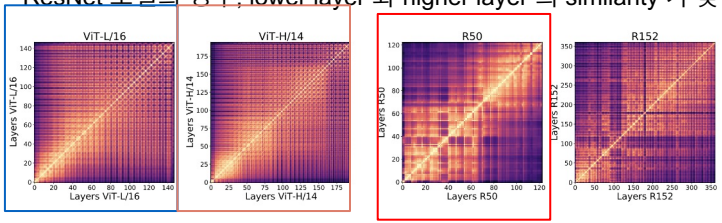


Figure 1: Representation structure of ViTs and convolutional networks show significant differences, with ViTs having highly similar representations throughout the model, while the ResNet models show much lower similarity between lower and higher layers. We plot CKA similarities between all pairs of layers across different model architectures. The results are shown as a heatmap, with the x and y axes indexing the layers from input to output. We observe that ViTs have relatively uniform layer similarity structure, with a clear grid-like pattern and large similarity between lower and higher layers. By contrast, the ResNet models show clear stages in similarity structure, with smaller similarity scores between lower and higher layers.

2. **cross-model 비교**
   - all layers X from ViT and compare to all layers Y from ResNet
     - lower half of ResNet layer 와 lowest quarter of ViT layers 비슷
       - 동일한 representations 계산을 위해 ResNet 은 ViT lower layers 에 비해, 더 많은 lower layers를 필요로 함
     - remaining half of the ResNet 과 the next third of ViT layer 비슷
     - highest ViT layer 는 lower and higher ResNet layer 와 비슷하지 않음

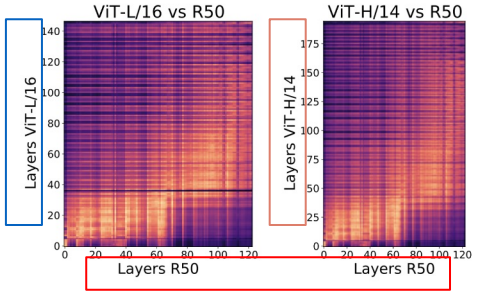
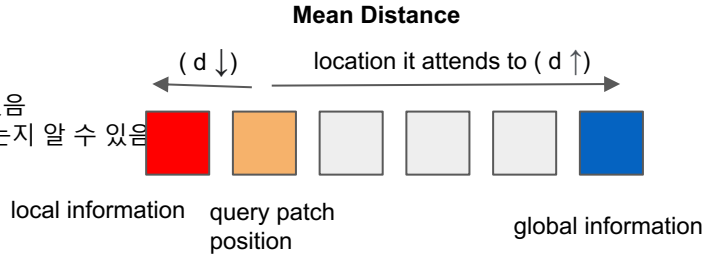


Figure 2: Cross model CKA heatmap between ViT and ResNet illustrate that a larger number of lower layers in the ResNet are similar to a smaller set of the lowest ViT layers. We compute a CKA heatmap comparing all layers of ViT to all layers of ResNet, for two different ViT models. We observe that the lower half of ResNet layers are similar to around the lowest quarter of ViT layers. The remaining half of the ResNet is similar to approximately the next third of ViT layers, with the highest ViT layers dissimilar to lower and higher ResNet layers.

Do Vision Transformers See Like Convolutional Neural Networks?

3. Local and Global information in layer representations

- 얼마나 많은 global information 이 ViT early self-attention layer 에 포함되어 있을까?
- 각 self-attention head 에서, query patch position 과 locations it attends to 간의 거리를 구할 수 있음
  → 이는 얼마나 많은 local vs global information 이 각 self-attention layer 에 aggregating 되어 있는지 알 수 있음



JFT - ImageNet (bigger dataset)

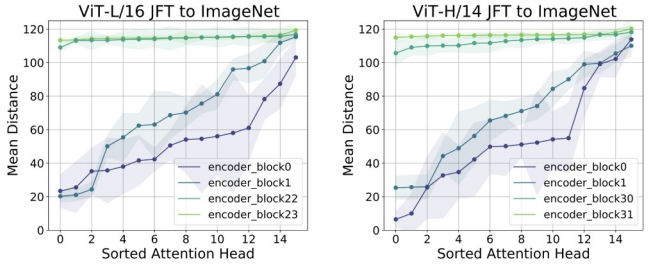


Figure 3: Plotting attention head mean distances shows lower ViT layers attend both locally and globally, while higher layers primarily incorporate global information. For each attention head, we compute the pixel distance it attends to, weighted by the attention weights, and then average over 5000 datapoints to get an average attention head distance. We plot the heads sorted by their average attention distance for the two lowest and two highest layers in the ViT, observing that the lower layers attend both locally and globally, while the higher layers attend entirely globally.

lower ViT layers 는 locally, globally 를 나타내는 반면,
higher layers 는 대부분 global information 만을 포함함.

ImageNet (smaller dataset)

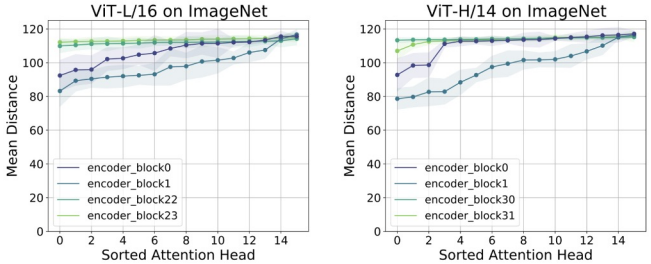


Figure 4: With less training data, lower attention layers do not learn to attend locally. Comparing the results to Figure 3, we see that training only on ImageNet leads to the lower layers not learning to attend more locally. These models also perform much worse when only trained on ImageNet, suggesting that incorporating local features (which is hardcoded into CNNs) may be important for strong performance. (See also Figure C.5.)

적은 양의 dataset (ImageNet) 에서는
lower attention layer 가 locally 정보를 거의 학습하지 않음

Do Vision Transformers See Like Convolutional Neural Networks?

4. Does access to global information result in different features?

- ResNet 의 lower layer representation 의 경우, (block1unit1 - block2unit2)
  ViT local attention head (smallest mean distance) 와 유사함.

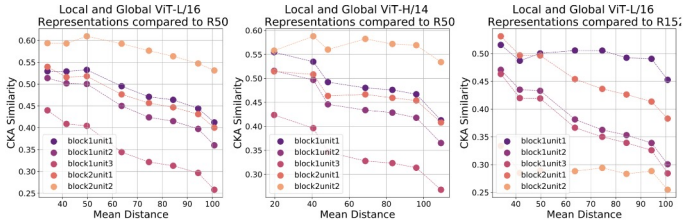


Figure 5: Lower layer representations of ResNet are most similar to representations corresponding to local attention heads of ViT. We take subsets of ViT attention heads in the first encoder block, ranging from the most locally attending heads (smallest mean distance) to the most global heads (largest mean distance). We then compute CKA similarity between these subsets and lower layer representations in the ResNet. We observe that lower ResNet layers are most similar to the features learned by local attention heads of ViT, and decrease monotonically in similarity as more global information is incorporated, demonstrating that the global heads learn quantitatively different features.

⇒ global information 의 경우,
ResNet 의 lower layer 에서 계산되는 것과는 quantitatively 한 다른 feature 를 가짐.

5. Effective Receptive Fields

- ViT가 lower layer 에서 ResNet 보다 더 큰 effective receptive fields
- ResNet는 effective receptive fields grow gradually, ViT receptive fields become much more global midway through the network
- ViT 는 strong dependence on their center patch

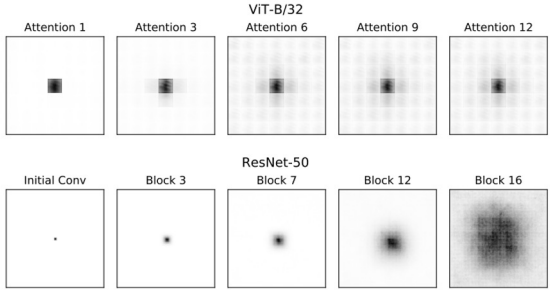


Figure 6: ResNet effective receptive fields are highly local and grow gradually; ViT effective receptive fields shift from local to global. We measure the effective receptive field of different layers as the absolute value of the gradient of the center location of the feature map (taken after residual connections) with respect to the input. Results are averaged across all channels in each map for 32 randomly-selected images.

ViT

Resnet

Dipper layer →

“Receptive field는 각 단계의 입력 이미지에 대해 하나의 필터가 커버할 수 있는 이미지 영역의 일부를 **뜻**한다.”

End of the Document