Hello!

We are Spencer King & Sixiang Zhang

Transforming Computer Vision

Demo

DeepAI - Text to Image

Supporting Papers

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Alexey Dosovitskiy[*,†], Lucas Beyer[*], Alexander Kolesnikov[*], Dirk Weissenborn[*], Xiaohua Zhai[*], Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby[*,†]

[*]equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Ze Liu[†*] Yutong Lin[†*] Yue Cao[*] Han Hu[*‡] Yixuan Wei[†]

Zheng Zhang Stephen Lin Baining Guo

Microsoft Research Asia

{v-zeliu1,v-yutlin,yuecao,hanhu,v-yixwe,zhez,stevelin,bainguo}@microsoft.com

Agenda

1. Background
2. Related Work
3. Motivation
4. Methods
5. Evaluation
6. Conclusion
7. Swin Transformer

1.

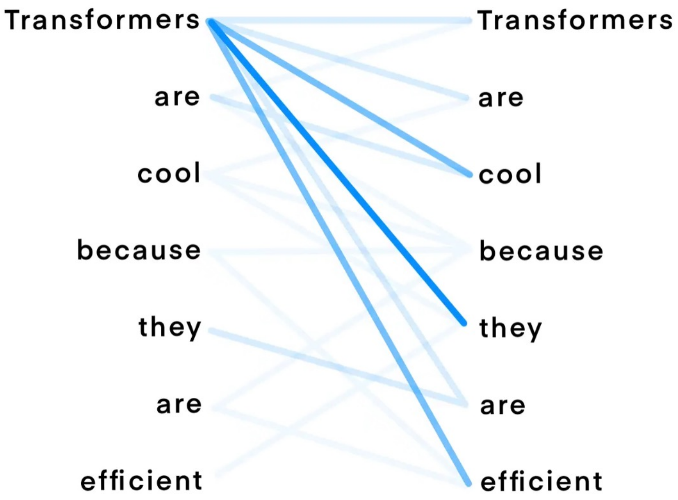
Background

The problem & why it is important

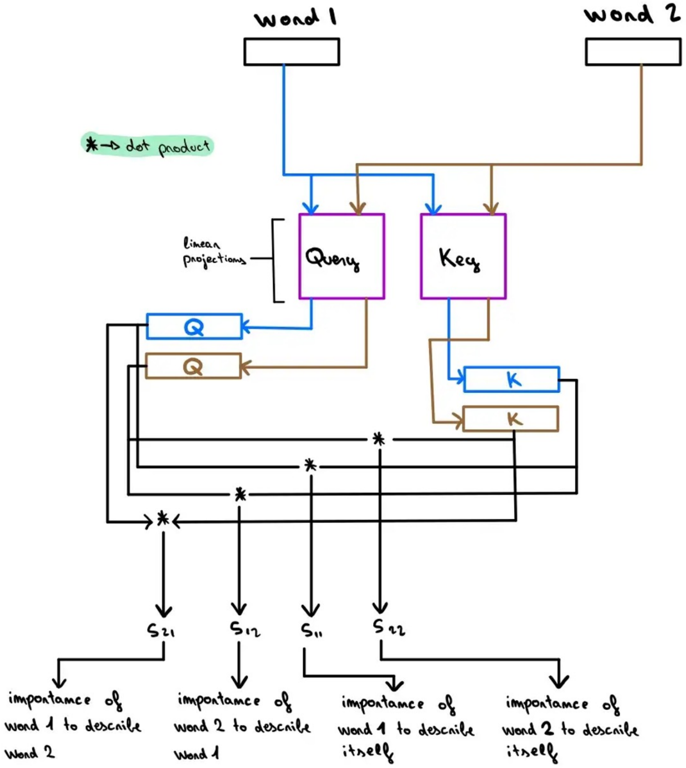
Transformers in NLP

- Transformers were mostly used for NLP problems (model of choice)
- Very computationally efficient and scalable
- Ability to handle long-term dependencies (better than RNNs)
- Allowed for the training of model of unprecedented size with over 100B parameters

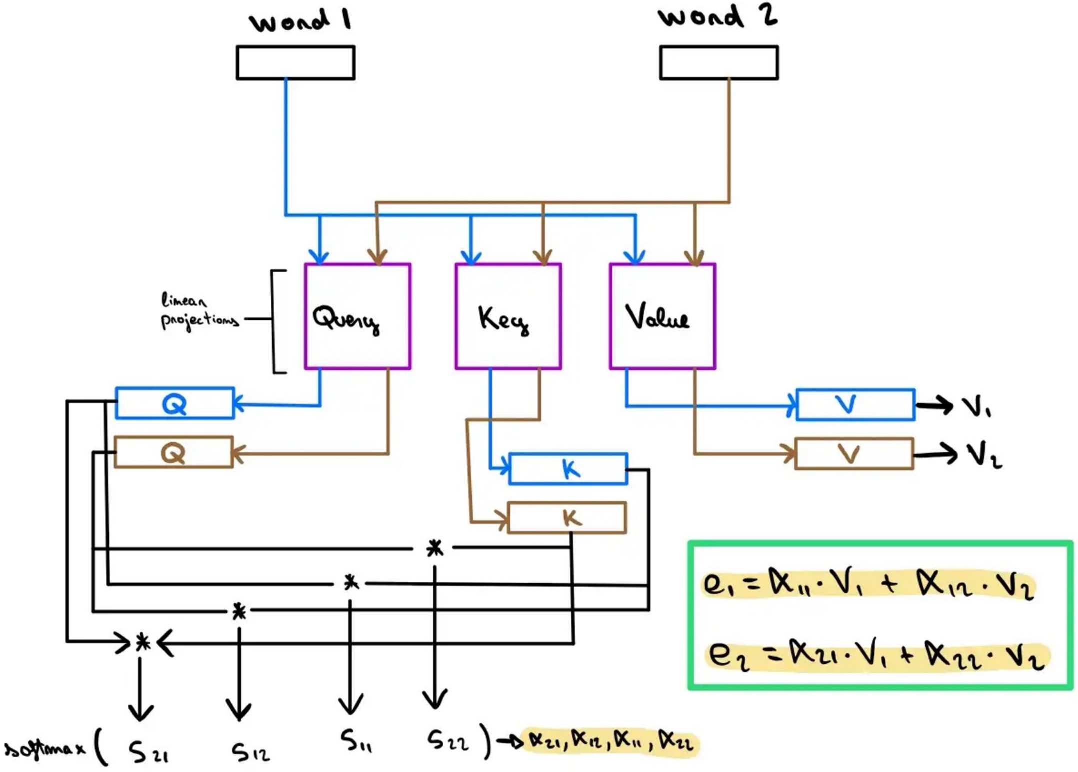
Self-Attention



Attention Scores Between 2 Words



Final Word Embeddings



Toy Example

https://drive.google.com/file/d/1gL0JoHm3KdN8yYMKhszrtNuAqlMjKgJ1/view?usp=share_link

Computer Vision Before Application of Transformers

- Computer vision tasks were dominated by various CNN architectures (AlexNet, VGG-16, ResNet, etc)
- Some newer works tried combining CNN with self-attention but could not scale effectively
- Issue is CNN architecture do not scale effectively on modern hardware accelerators

Application of Transformers to Computer Vision Tasks

- Trained on mid-sized data sets (ImageNet ~ 14M)
  - Poor results
- Trained on larger datasets (14M - 300M)
  - Excellents results
- Transformers lack inductive biases present in CNNs
- “ **Training trumps inductive bias!**”

Overarching Problem

Is there a more scalable solution to compete with state-of-the-art CNNs on computer vision tasks?

Main Idea

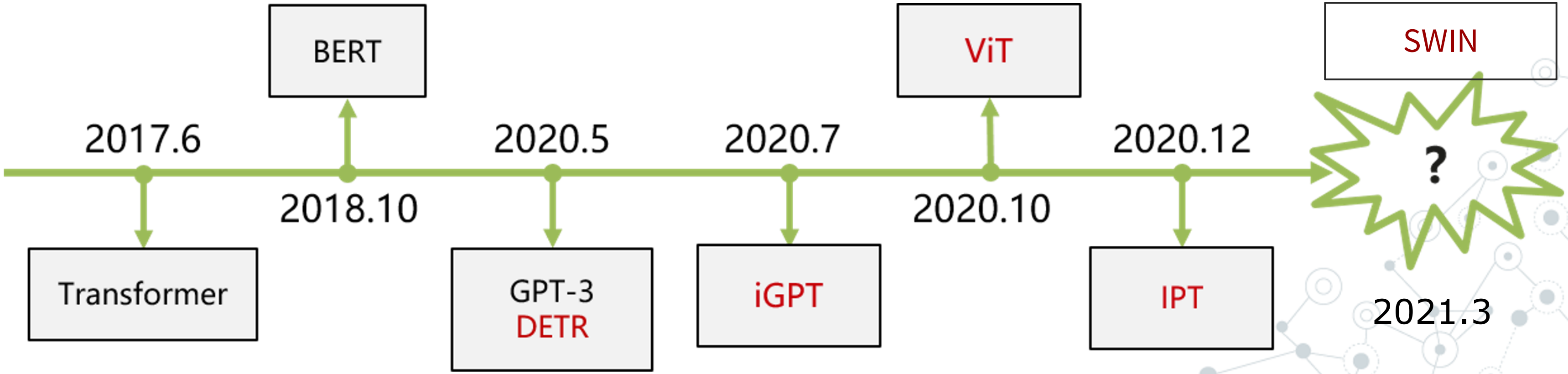
Use the scalability of transformers to more efficiently solve computer vision problems

2.

Related Work

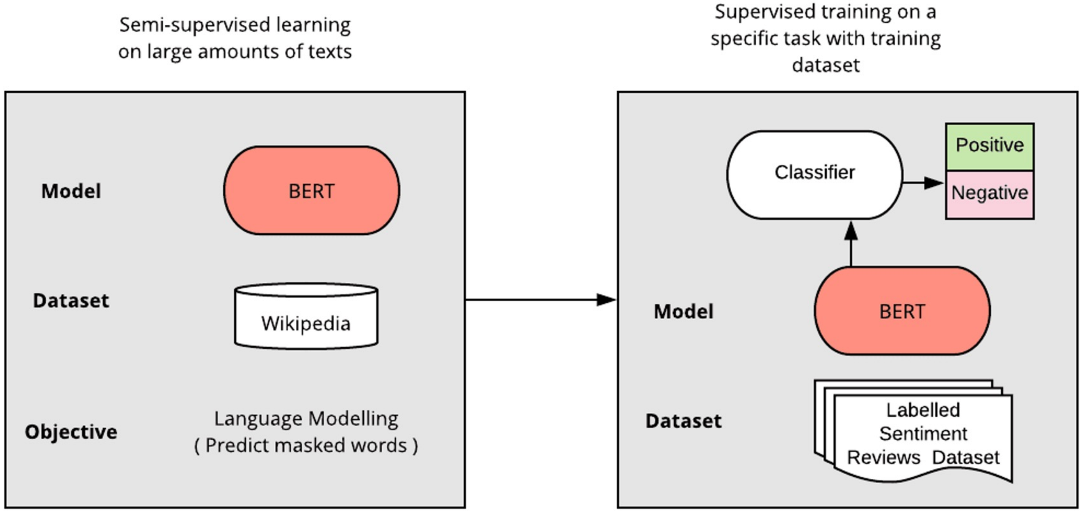
A review of work related to our problem

Timeline



Source: https://www.arxiv-vanity.com/papers/2012.12556/

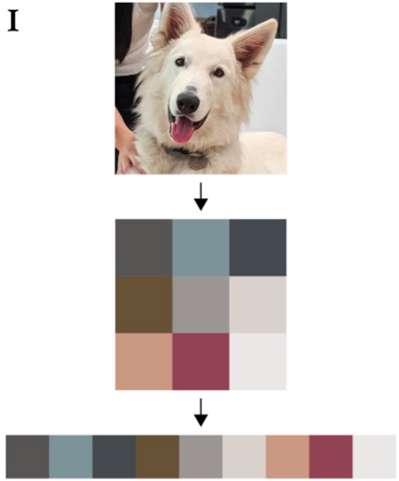
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding



"AS close as possible to the Bert" — By ViT

Source:https://nish-19.github.io/posts/2021/02/blog-post-4/

iGPT (Generative Pretraining from Pixels)



1)Reducing image resolution and color space
2)a generative model based on Transformers

Source: https://proceedings.mlr.press/v119/chen20s.html

3.

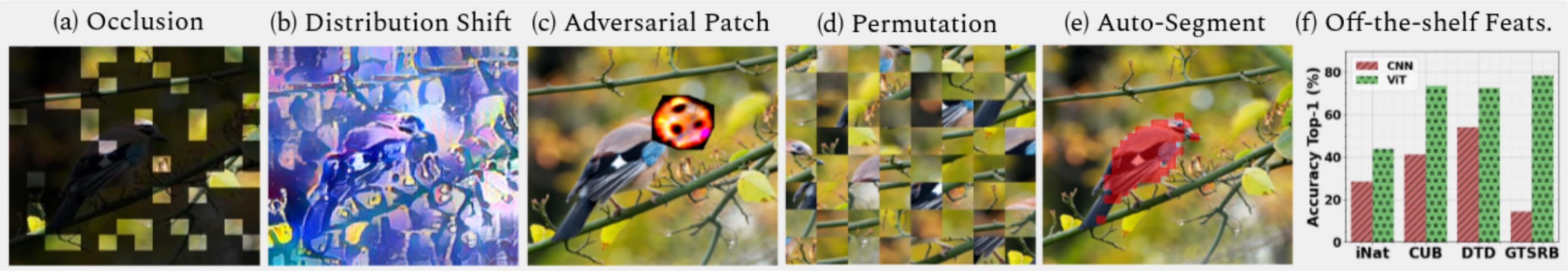
Motivation

Why was this work proposed?

We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks…

…Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.

CNN and ViT



https://arxiv.org/abs/2105.10497

4.

Methods

Outlining the work's procedures

Question?

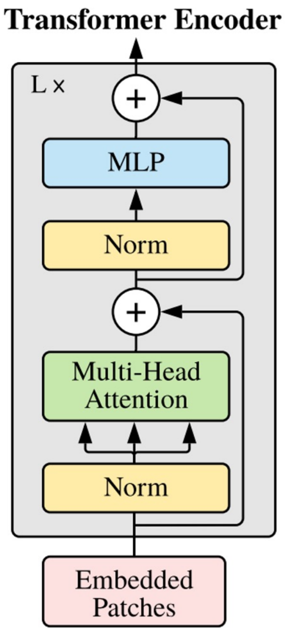
WHY DON'T WE USE A FULL IMAGE FOR TRANSFORMER?

Recall: Self-Attention

Complexity!

$O(n^2)$

Method



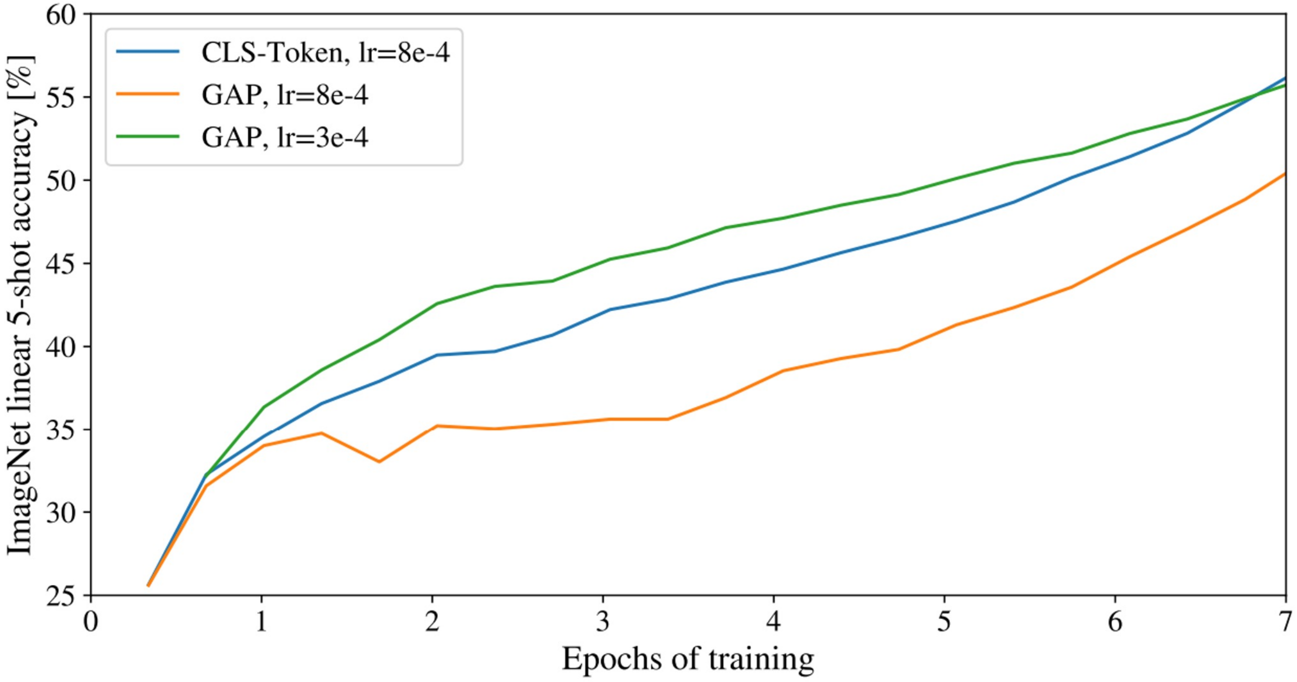
Source: lucidrains/vit-pytorch

Details of ViT variants

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

ViT-L/16 means the “Large” variant with 16×16 input patch size.

[Class] Token VS GAP(globally average-pooling)



Why needs Position embeddings?

Positions embedding(cont.)

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

5.

Evaluation

What are the results?

Dataset

	# of Images	# of Classes
ImageNet (Small)	1.3 Million	1 Thousand
ImageNet-21K (Medium)	14 Million	21 Thousand
JFT (Big)	300 Million	18 Thousand

Source:https://github.com/wangshusen/DeepLearning/blob/master/Slides/10_ViT.pdf

6.

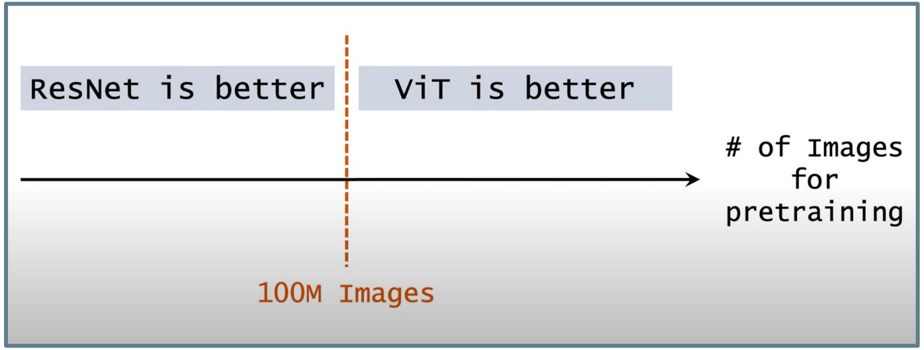
Conclusion

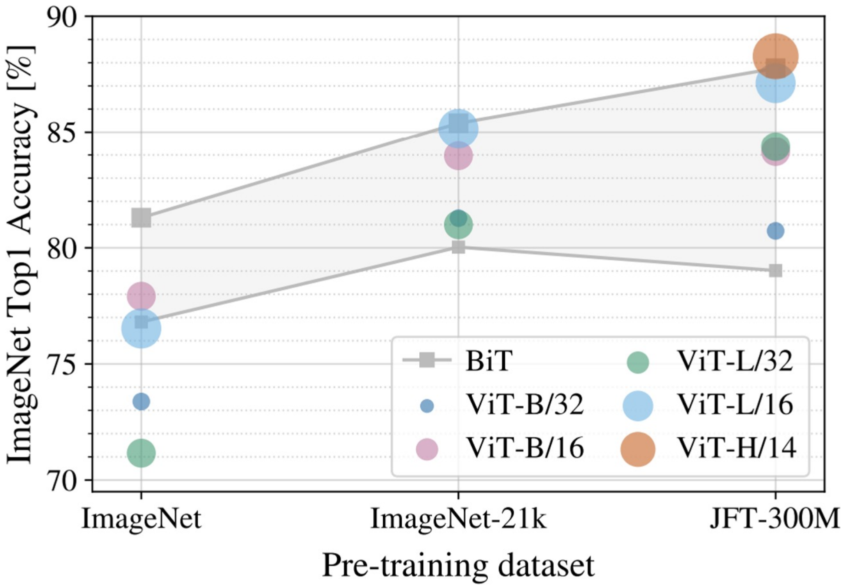
The key takeaways

Summary

- Simple
- Scalable
- Accuracy comparable to SOTA CNN models while computational less expensive to train
- Requires large amount of data for SOTA performance
- Unlike prior works, no image-specific inductive bias

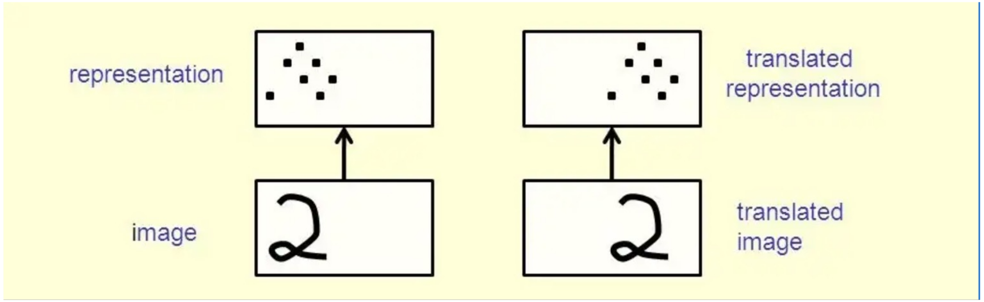
ResNet vs Transformer



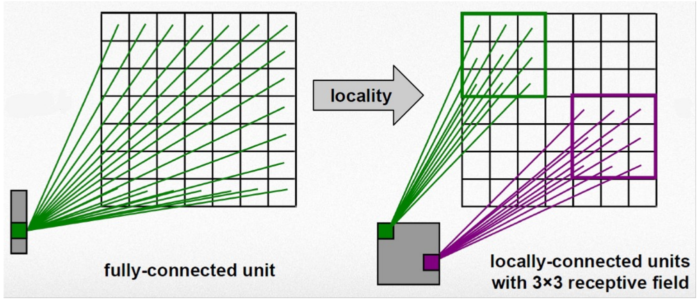


Why is ViT worse than ResNets at a small dataset?

CNN's inductive biases



Translation equivariance



locality

https://oi.readthedocs.io/en/latest/computer_vision/cnn/intro.html
https://towardsdatascience.com/translational-invariance-vs-translational-equivariance-f9fbc8fca63a

Best Model Performance Comparison

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Future Work

- Great results but challenges remain
- Apply ViT to other computer vision tasks not just image classification
  - Object detection
  - Image segmentation
- Improve pre-training to accommodate larger scale
- Further scaling of ViT itself
  - Structurally?

7.

Swin Transformer

Interesting follow up work extending ViT

…Swin Transformer, that capably serves as a general-purpose backbone for computer vision.

Motivation

- Address shortcomings of ViT
  - Can preform dense prediction tasks - object detection & image segmentation
  - Increases scalability - complexity scales linearly rather than quadratically with image size
- Create general purpose computer vision backbone

There exist many vision tasks such as semantic segmentation that require dense prediction at the pixel level, and this would be intractable for [the] Transformer on high-resolution images, as the computational complexity of its self-attention is quadratic to image size.

Shifted Window

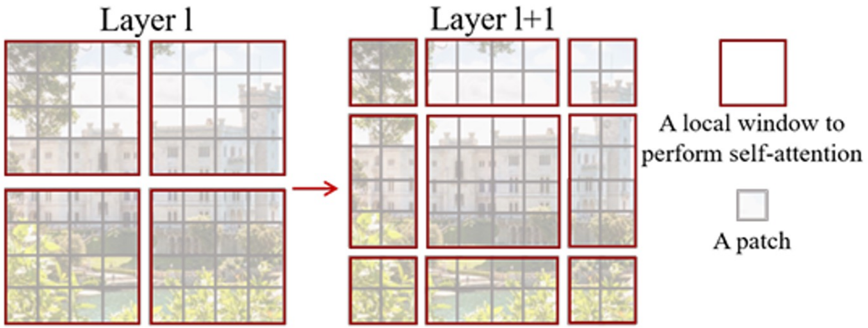


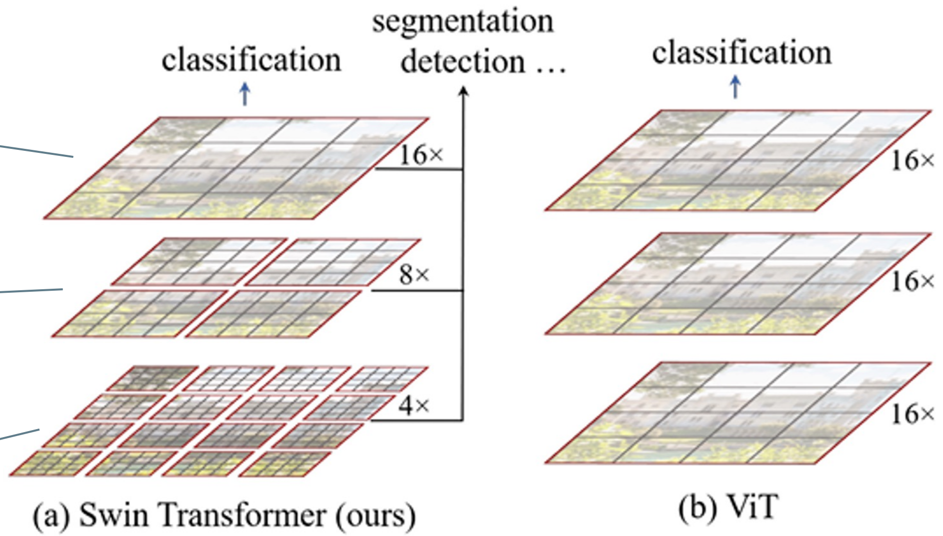
Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

Creating Patches

Dimensionality
16 x 16 x 3 = 768

Dimensionality
8 x 8 x 3 = 192

Dimensionality
4 x 4 x 3 = 48



Architecture

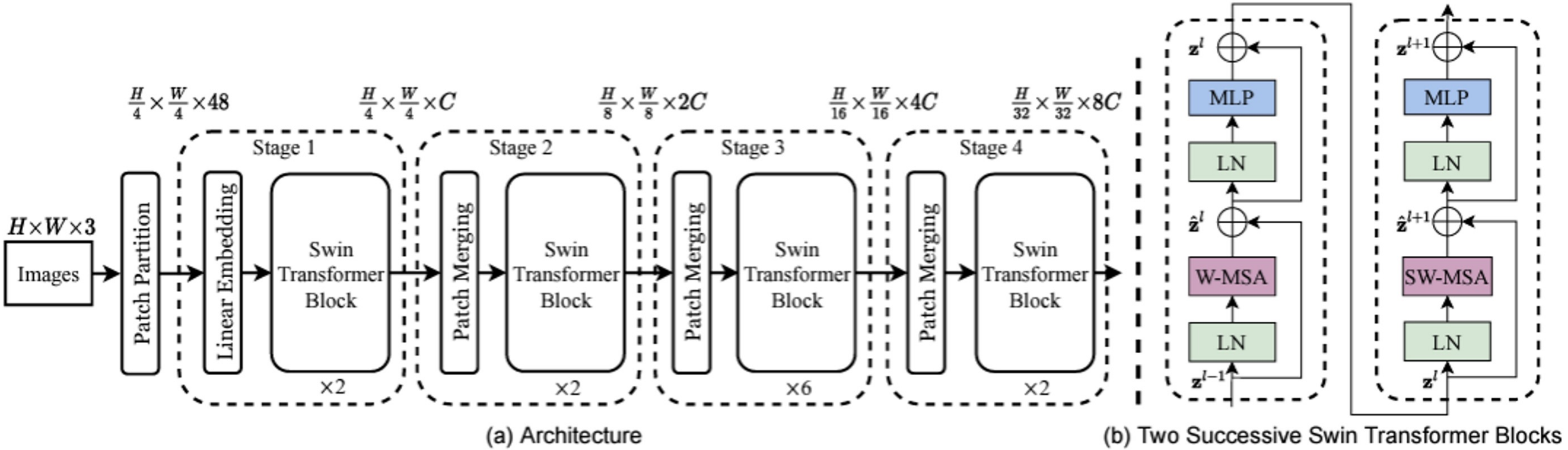


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Performance - Image Classification

(a) Regular ImageNet-1K trained models					
method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8
DeiT-B [63]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5

(b) ImageNet-22K pre-trained models					
method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

Performance - Object Detection

(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS
Cascade Mask R-CNN	R-50	46.3	64.3	50.5	82M	739G	18.0
	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0
Sparse R-CNN	R-50	44.5	63.4	48.2	106M	166G	21.0
	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6

Performance - Image Segmentation

ADE20K Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DANet [23]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [11]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [24]	ResNet-101	45.9	38.5	-		
DNL [71]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [73]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [69]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [73]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [11]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [11]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [81]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

Thanks!

Any questions?

The related papers can be found at the links below:

1. https://arxiv.org/abs/2010.11929
2. https://arxiv.org/abs/2103.14030

Resources

- General Overview of Transformers in Various Applications
  https://towardsdatascience.com/transformers-in-computer-vision-farewell-convolutions-f083da6ef8ab
- Short Overview of ViT Paper
  https://www.youtube.com/watch?v=HZ4j_U3FC94
- Complete Coverage of ViT Paper
  https://www.youtube.com/watch?v=TrdevFK_am4
- Explanation of the Swin Transformer Paper
  https://www.youtube.com/watch?v=SndHALawoag
- Second explanation of the Swin Transformer Paper
  https://www.youtube.com/watch?v=tFYxJZBAbE8

Resources Cont.

- About Metrics of AP and mAP for Object Detection / Instance Segmentation https://yanfengliux.medium.com/the-confusing-metrics-of-ap-and-map-for-object-detection-3113ba0386ef