UAB UOC UPC upf.

Master in Computer Vision Barcelona

[http://pagines.uab.cat/mcv/]

Module 6 - Day 7

The Transformer in Vision

28th March 2023

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Acknowledgments

Xavier Giro-i-Nieto

@DocXavi
xavier.giro@upc.edu

Applied Scientist
Amazon Science Barcelona.

Associate Professor
Universitat Politècnica de Catalunya

Outline

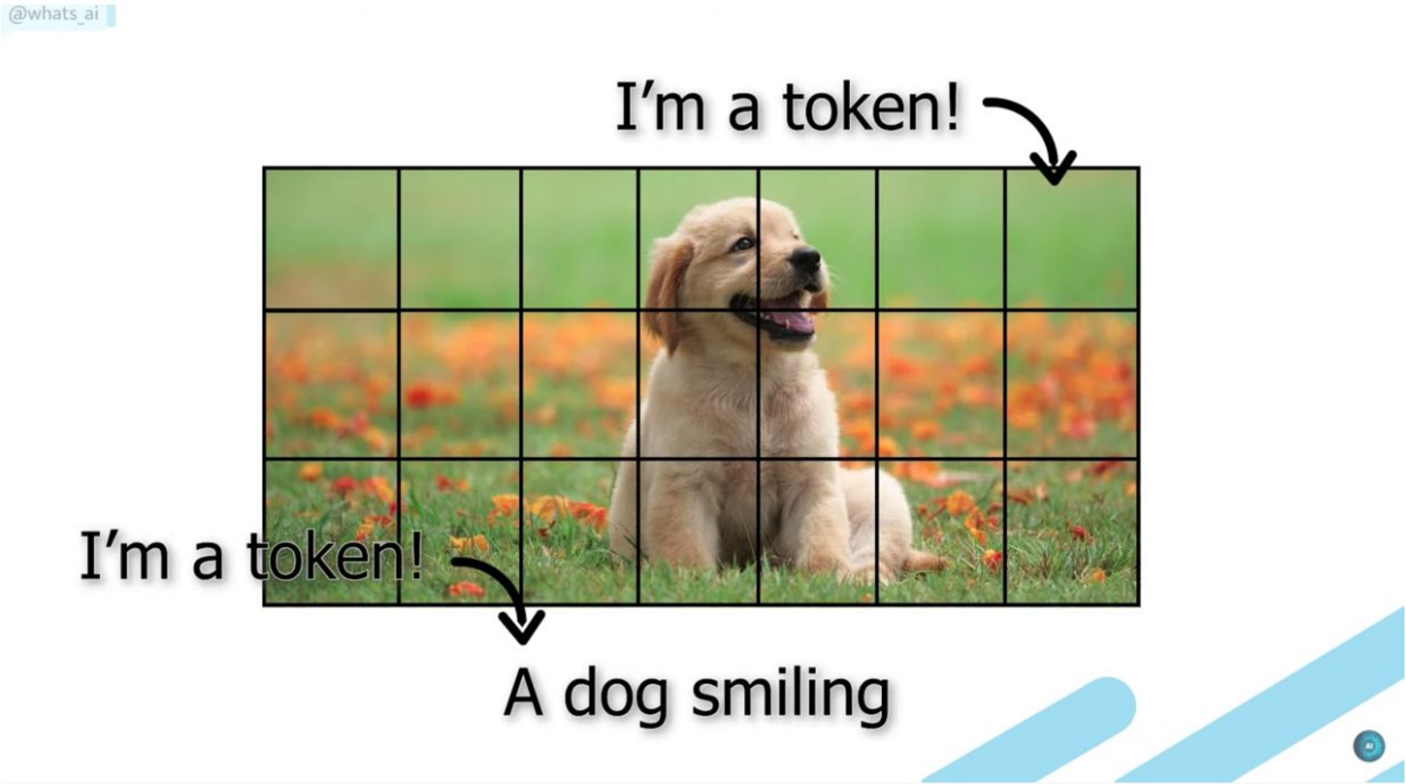
1. Vision Transformer (ViT)
2. Beyond ViT

The Transformer for Vision: ViT

#ViT Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." ICLR 2021. [blog] [code] [video by Yannic Kilcher]

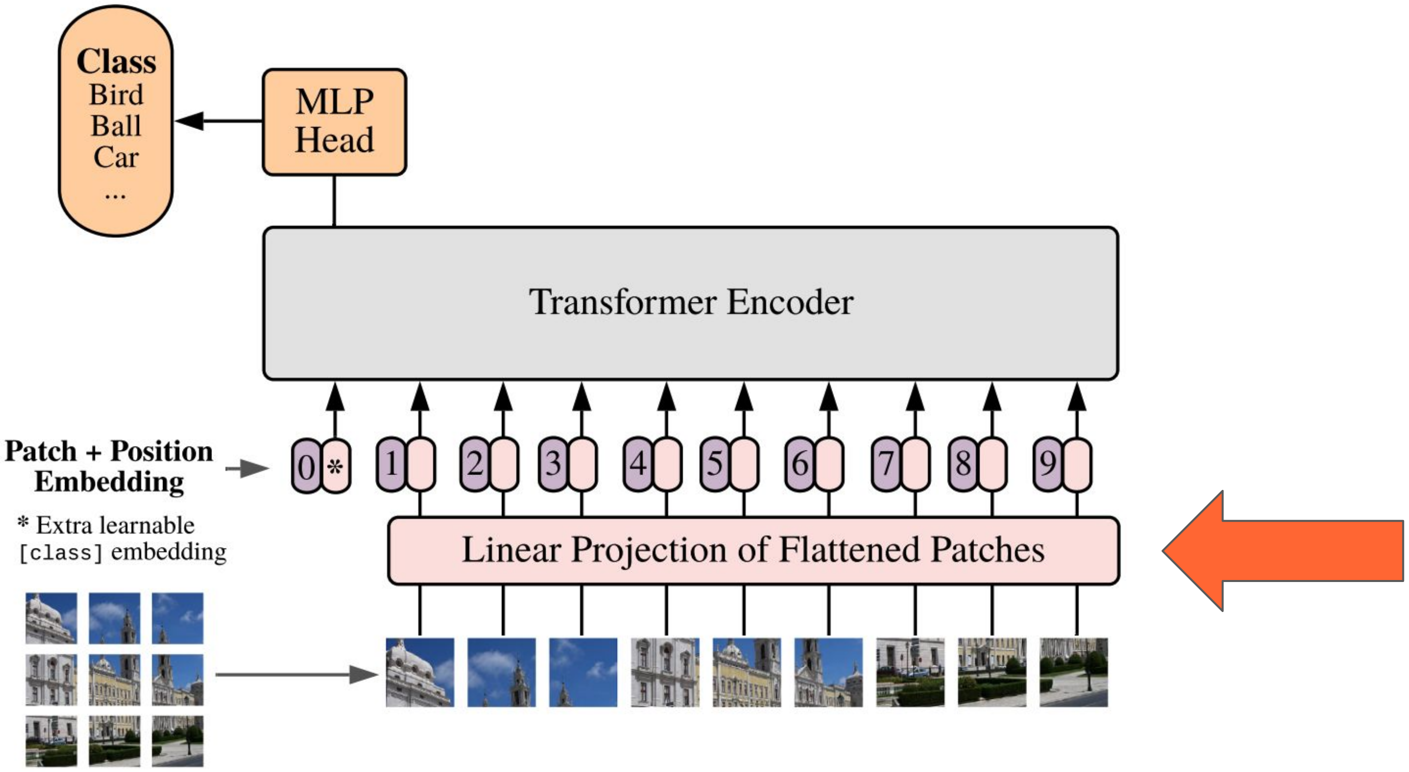
Outline

The Transformer for Vision



Source: What's AI, Will Transformers Replace CNNs in Computer Vision? + NVIDIA GTC Giveaway (2021)

The Transformer for Vision: ViT



#ViT Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." ICLR 2021. [blog] [code] [video by Yannic Kilcher]

Linear projection of Flattened Patches

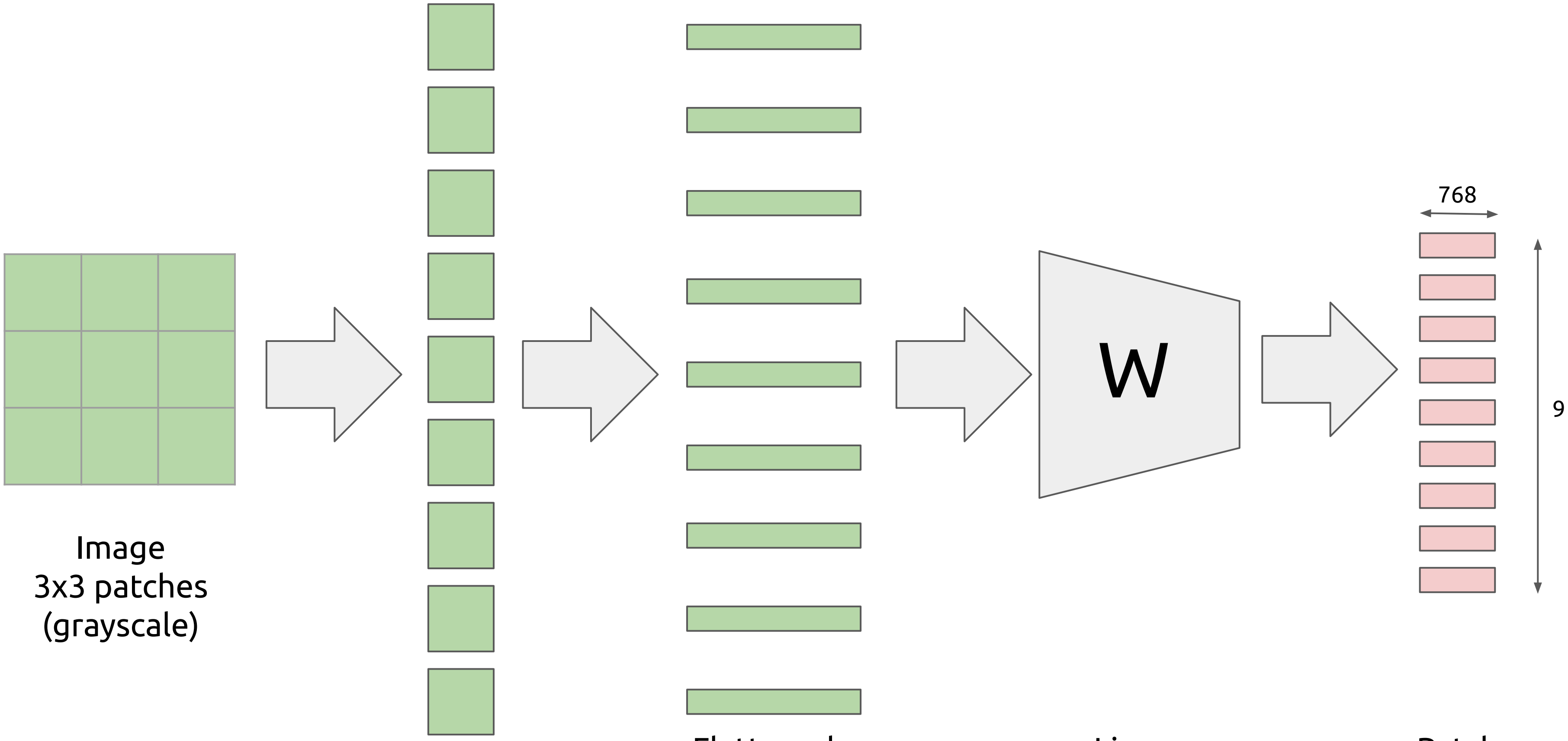


Image
3x3 patches
(grayscale)

Patches

Flattened
patches

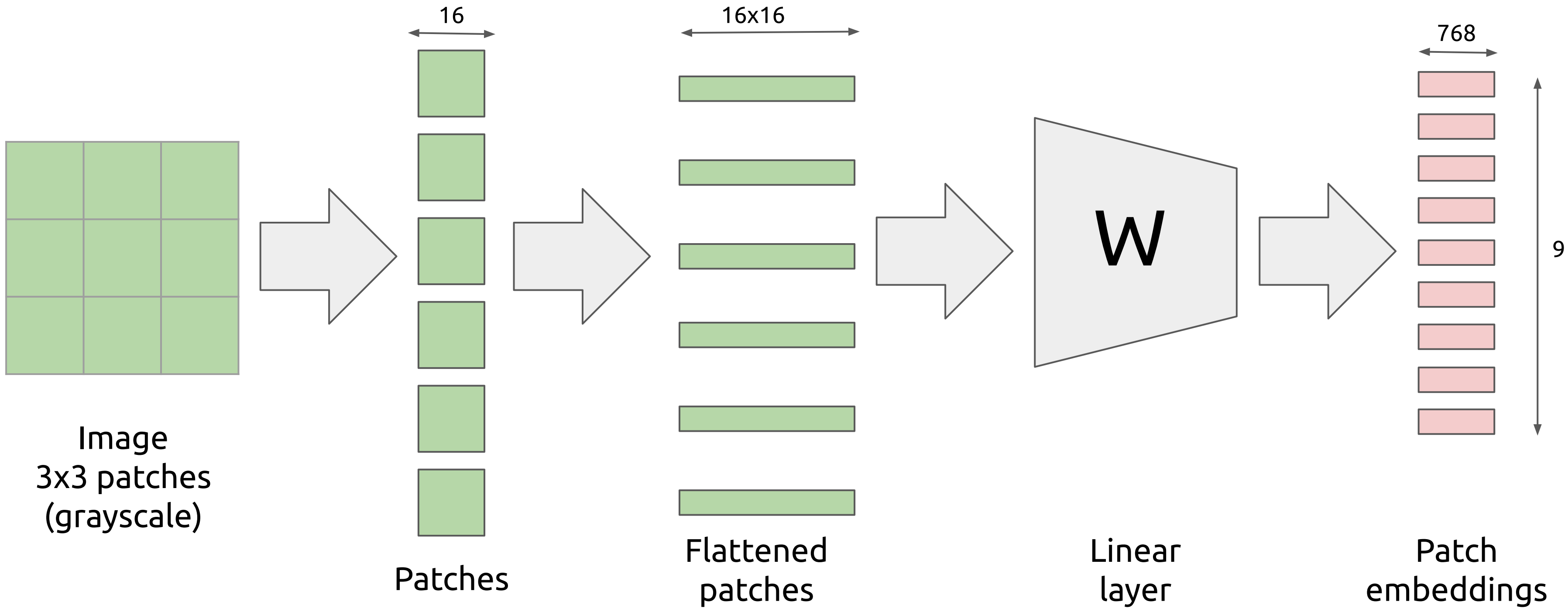
Linear
layer

Patch
embeddings

The Transformer for Vision

...

Consider the case of patches of 16x16 pixels and their embedding size of D=768, as in ViT-Base. How could the linear layer be implented with a convolutional layer ?



The Transformer for Vision

Consider the case of patches of 16x16 pixels and their embedding size of D=768, as in ViT-Base. How could the linear layer be implemented with a convolutional layer ?

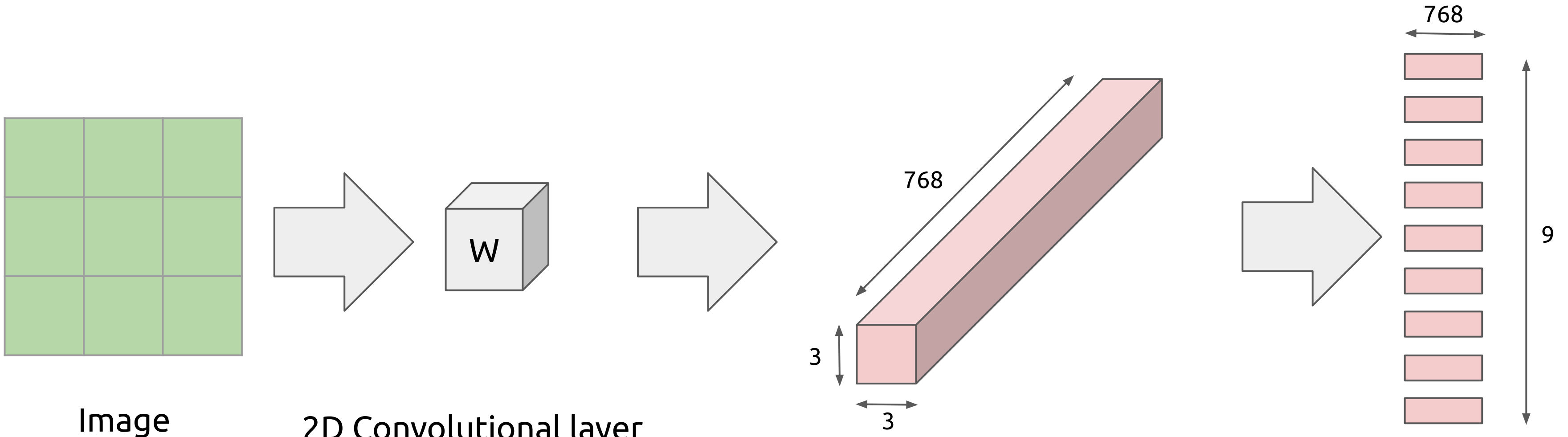


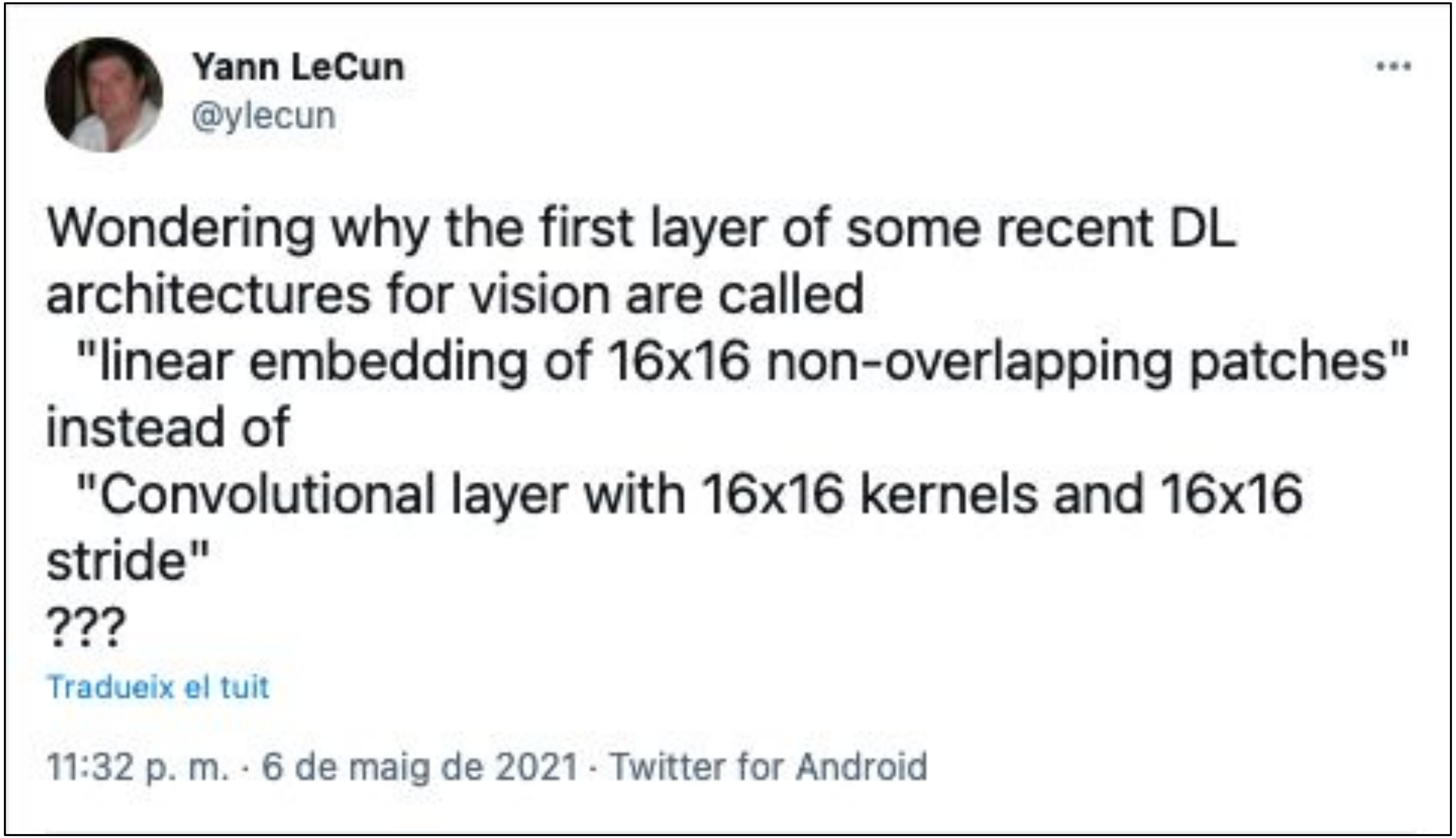
Image
3x3 patches
(grayscale)

<u>2D Convolutional layer</u>

768 filters
Kernel size = 16x16
Stride = 16x16

Patch
embeddings

The Transformer for Vision



Yann LeCun
@ylecun

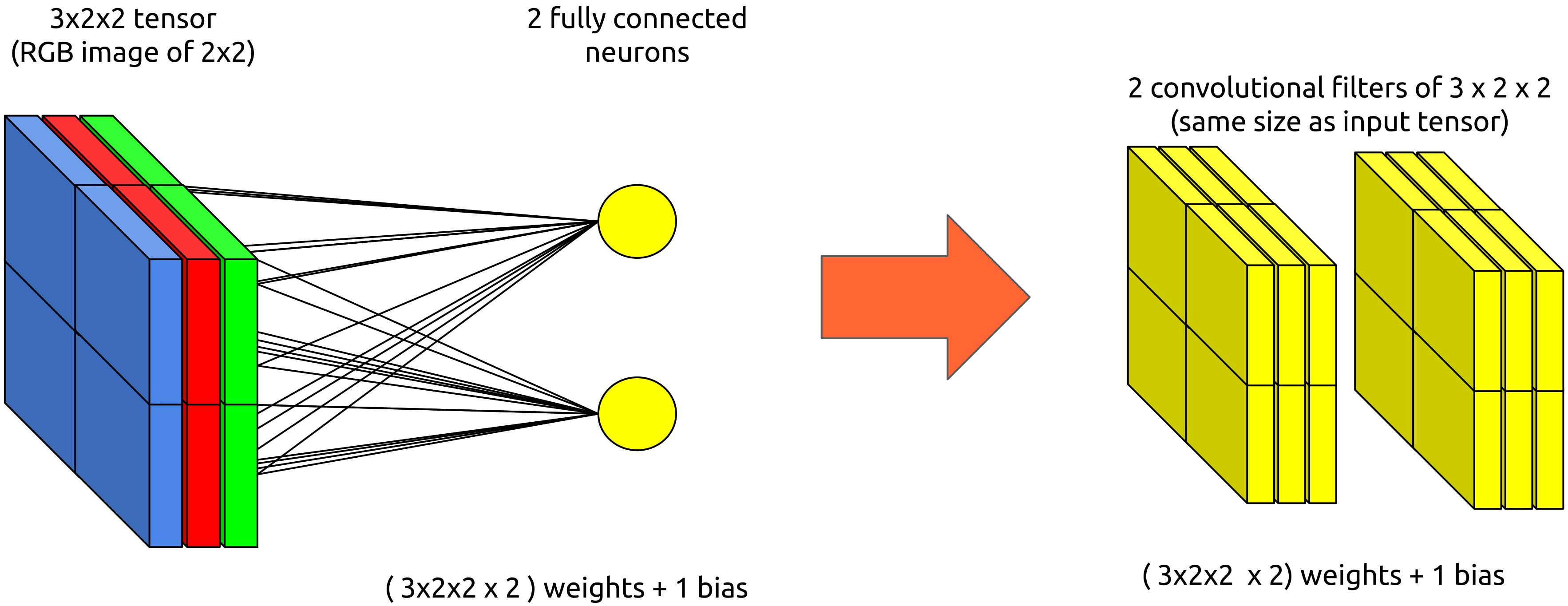
Wondering why the first layer of some recent DL architectures for vision are called
"linear embedding of 16x16 non-overlapping patches"
instead of
"Convolutional layer with 16x16 kernels and 16x16 stride"
???

Tradueix el tuit

11:32 p. m. · 6 de maig de 2021 · Twitter for Android

The Transformer for Vision

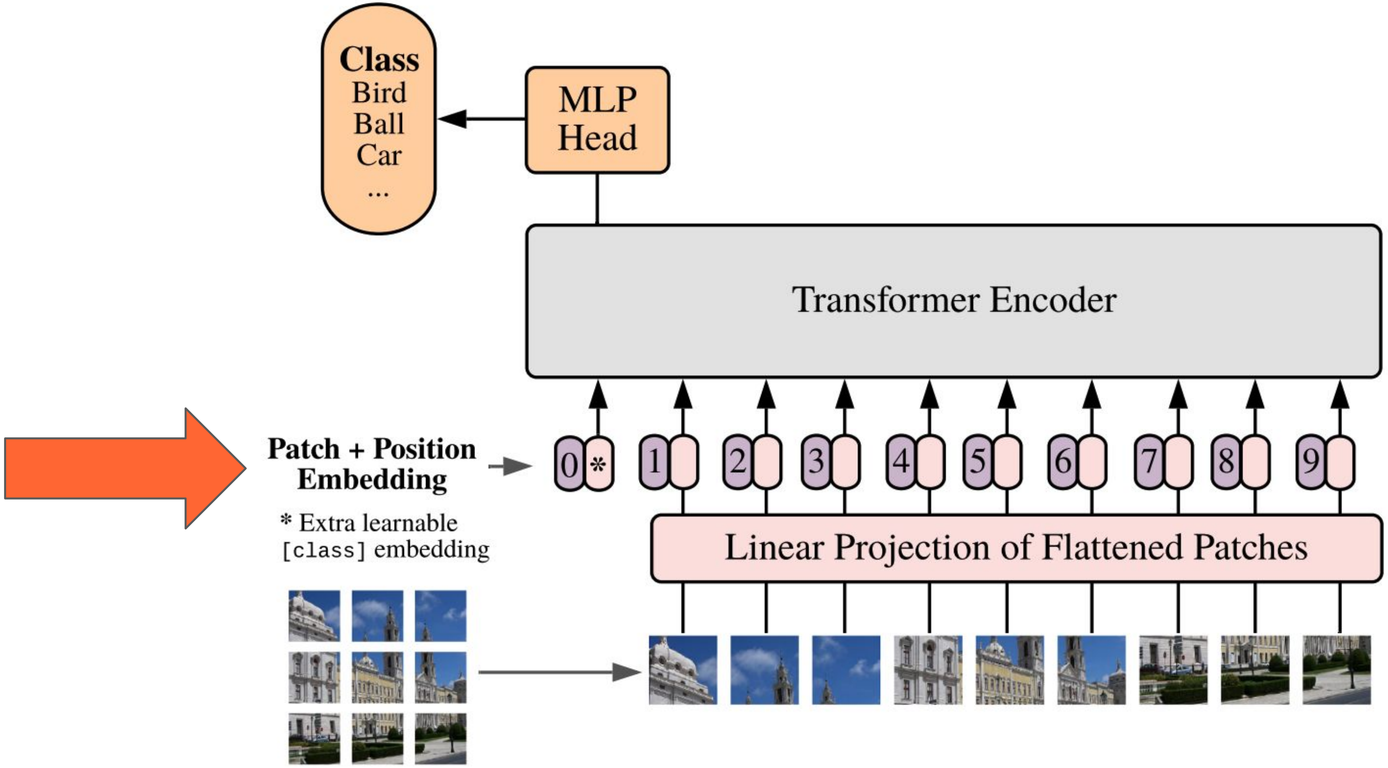
Observation: Fully connected neurons could be implemented as convolutional ones.



Outline

1. Vision Transformer (ViT)
    a. Tokenization
    b. Position embeddings
    c. Class embedding
    d. Receptive field
    e. Performance

Position Embeddings

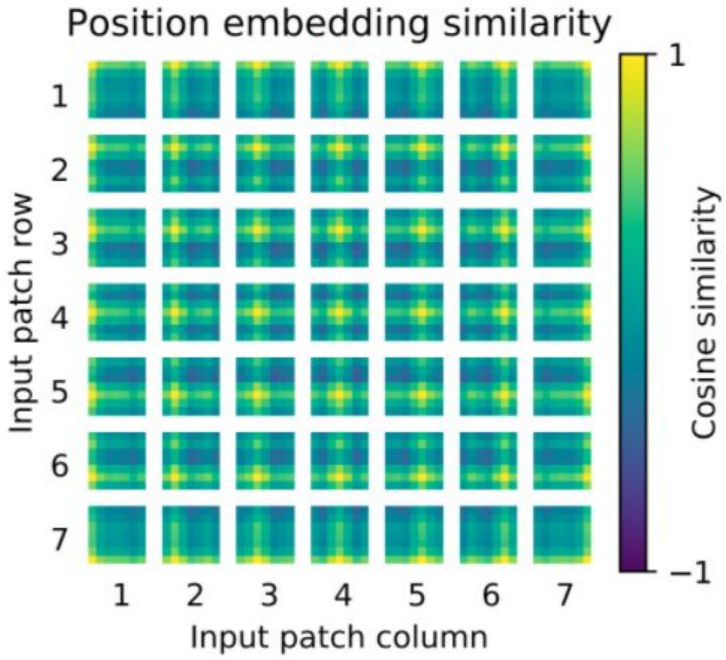


#ViT Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." ICLR 2021. [blog] [code] [video by Yannic Kilcher]

Position embeddings

The model **learns** to encode the relative position between patches.

Each position embedding is most similar to others in the same row and column, indicating that the model has recovered the grid structure of the original images.



#ViT Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." ICLR 2021. [blog] [code] [video by Yannic Kilcher]

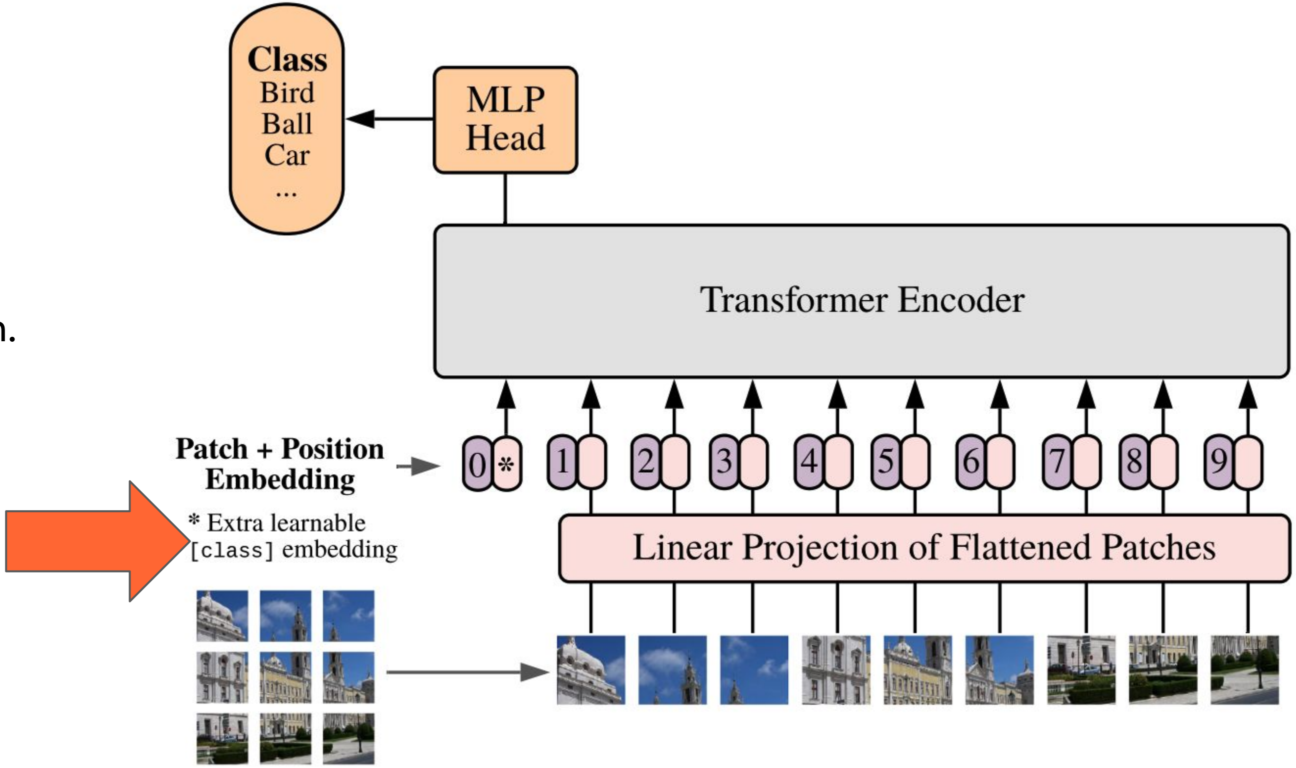
Outline

1. Vision Transformer (ViT)
    a. Tokenization
    b. Position embeddings
    c. **Class embedding**
    d. Receptive field
    e. Performance

Class embedding

[class] is a special <u>learnable</u> embedding added in front of every input example.

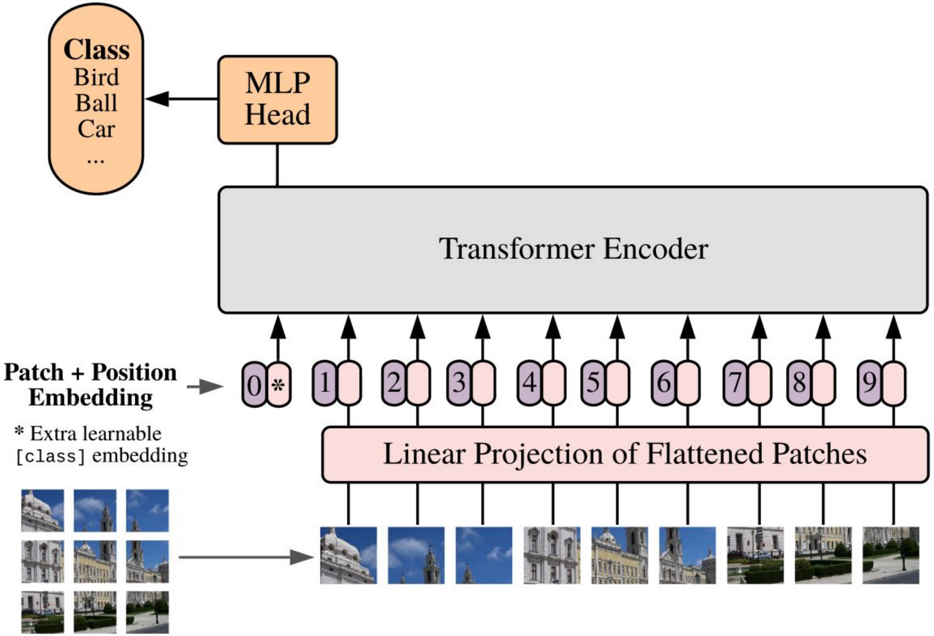
It triggers the class prediction.



#BERT Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. ["Bert: Pre-training of deep bidirectional transformers for language understanding."](#) NAACL 2019.

Class embedding

Why does the ViT not have a decoder in its architecture ?

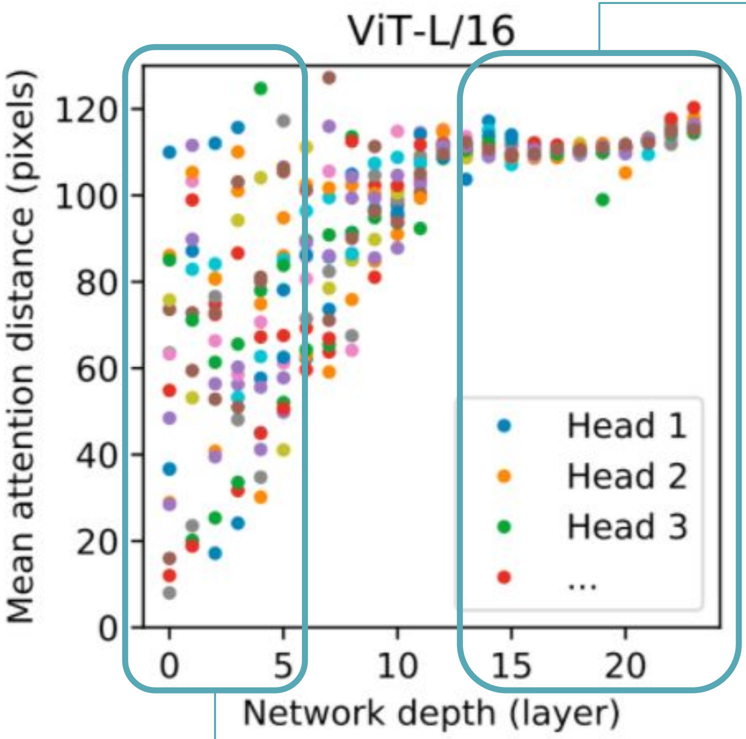


Outline

1. Vision Transformer (ViT)
    a. Tokenization
    b. Position embeddings
    c. Class embedding
    d. Receptive field
    e. Performance

Receptive field

Average spatial distance between one element attending to another for each transformer block:



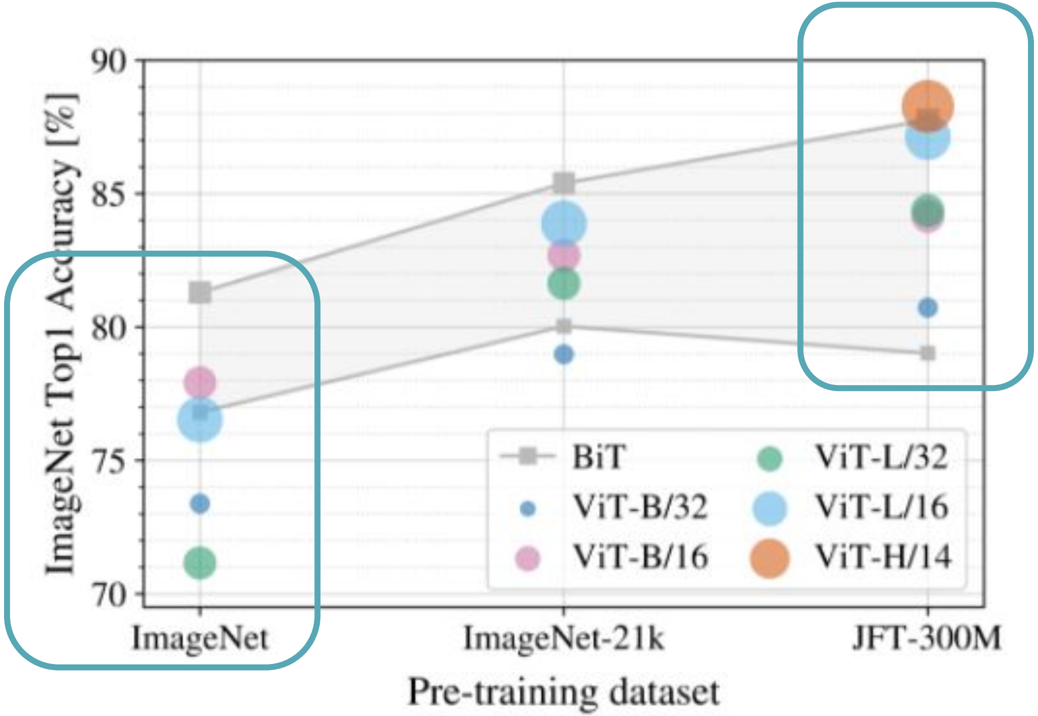
Deeper layers attend all over the image.

Both short & wide attention ranges in early layers (CNN can only learn short ranges)

Outline

1. Vision Transformer (ViT)
    a. Tokenization
    b. Position embeddings
    c. Class embedding
    d. Receptive field
    e. Performance

Performance: Accuracy

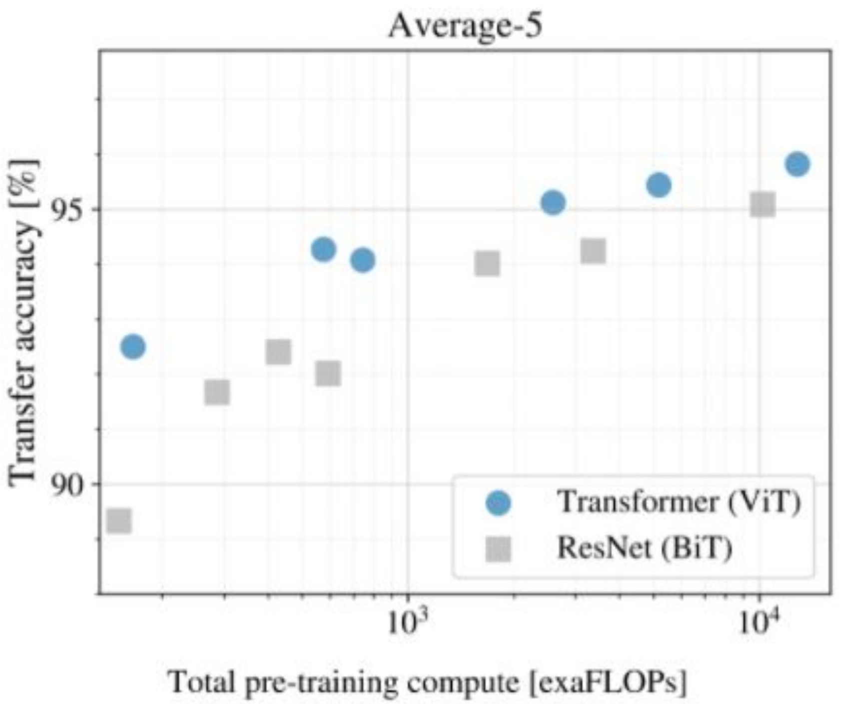


Worse performance than CNN (BiT) with ImageNet data only.

Slight improvement over CNN (BiT) when very large amounts of training data available.

#BiT Kolesnikov, Alexander, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. "Big transfer (bit): General visual representation learning." ECCV 2020.
#ViT Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." ICLR 2021. [blog] [code] [video by Yannic Kilcher]

Performance: Computation



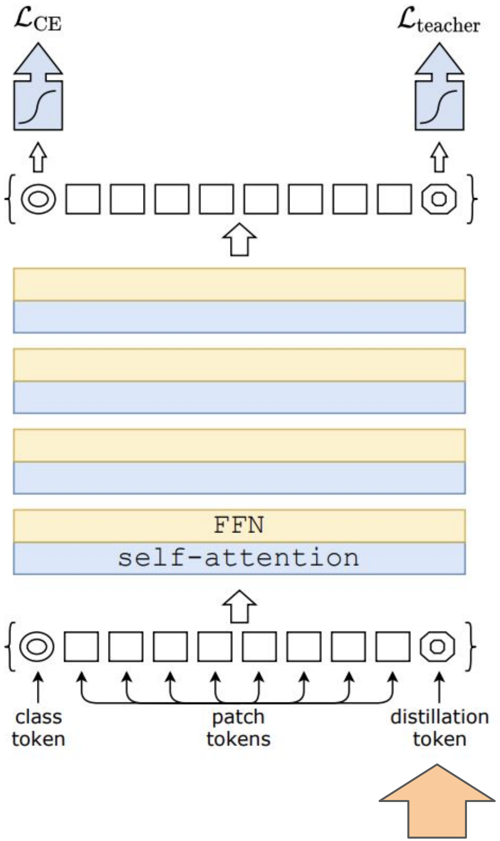
Requires less training computation than comparable CNN (BiT).

#BiT Kolesnikov, Alexander, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. "Big transfer (bit): General visual representation learning." ECCV 2020.
#ViT Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." ICLR 2021. [blog] [code] [video by Yannic Kilcher]

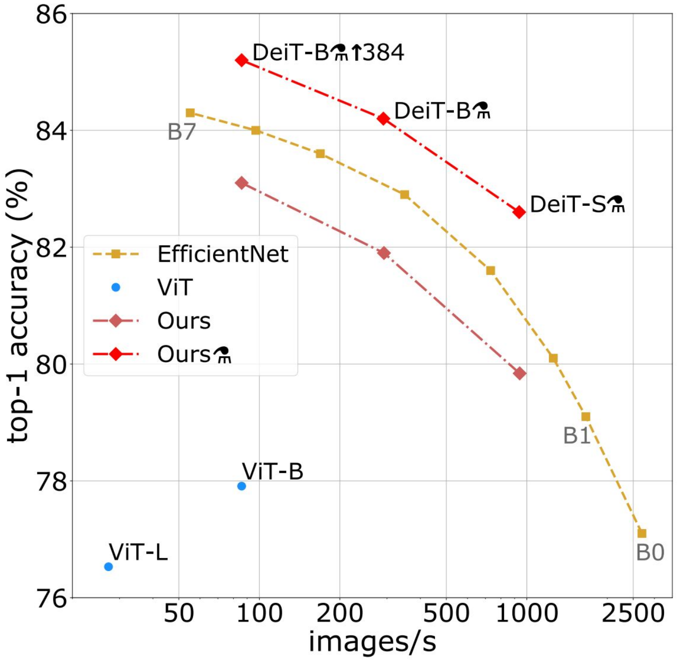
Outline

1. Vision Transformer (ViT)
    a. Tokenization
    b. Position embeddings
    c. Class embedding
    d. Receptive field
    e. Performance
2. **Beyond ViT**

Data-efficient Transformer (DeiT)



Distillation token that aims at predicting the label estimated by a teacher CNN. This allows introducing the convolutional bias in ViT.

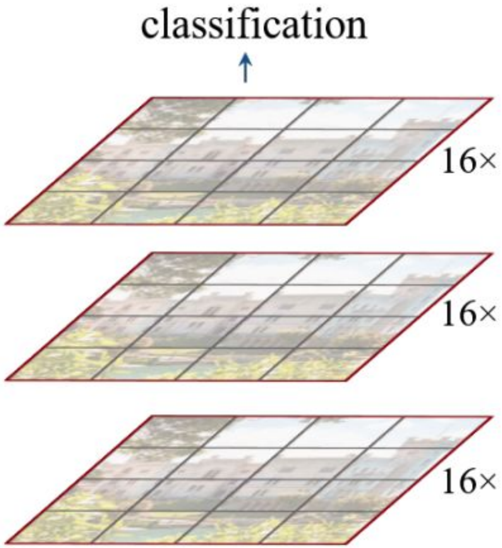


#DeIT Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. [Training data-efficient image transformers & distillation through attention](). ICML 2021.

Shifted WINdow (SWIN) Self-Attention (SA)

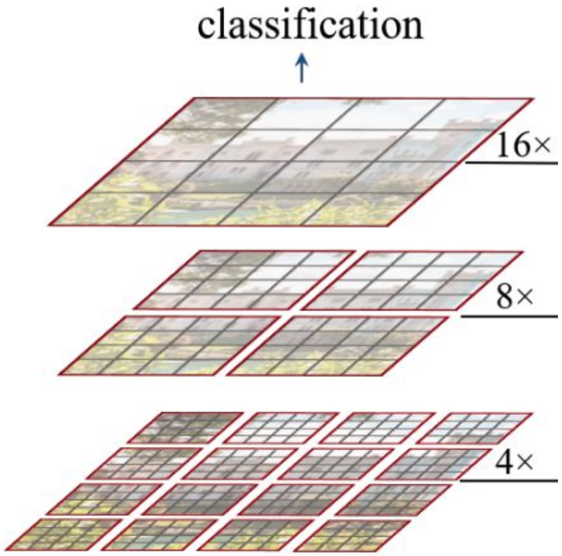
Less computation by self-attenting only in local windows (in grey).

Global SA



ViT

Global SA



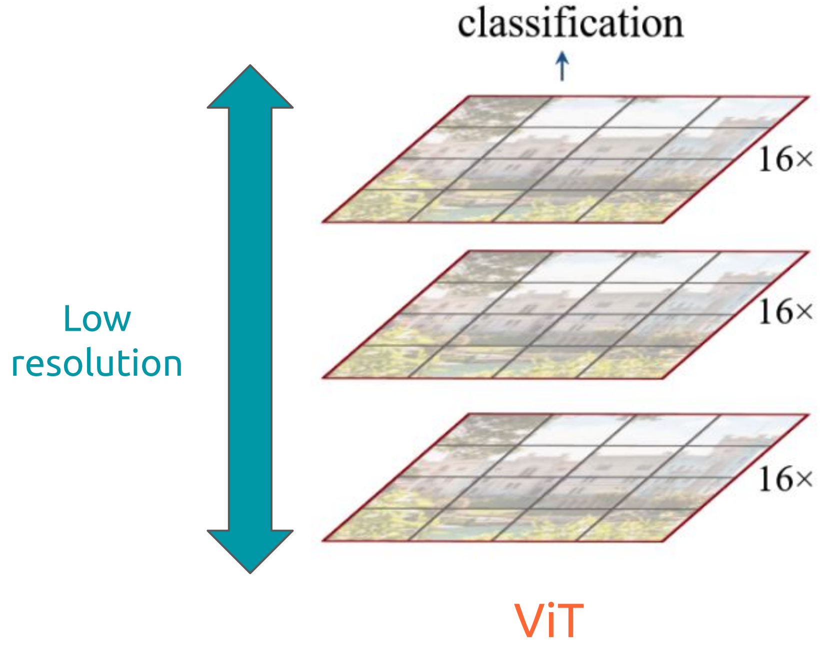
Localized SA

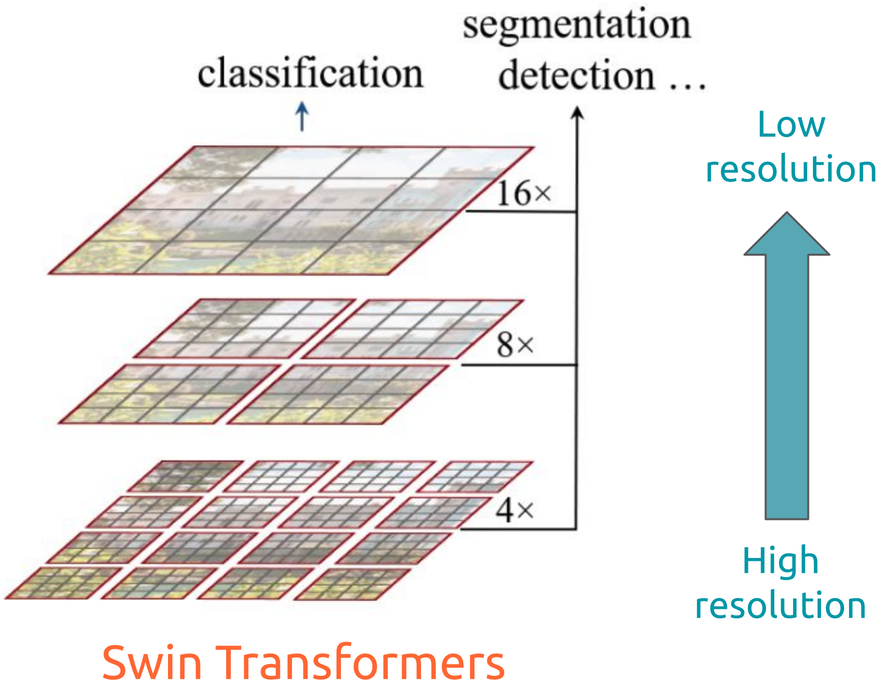
Swin Transformers

#SWIN Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. ICCV 2021 [Best paper award].

Hierarchical ViT Backbone

Hierarchical features maps by merging image patches (in red) across layers.

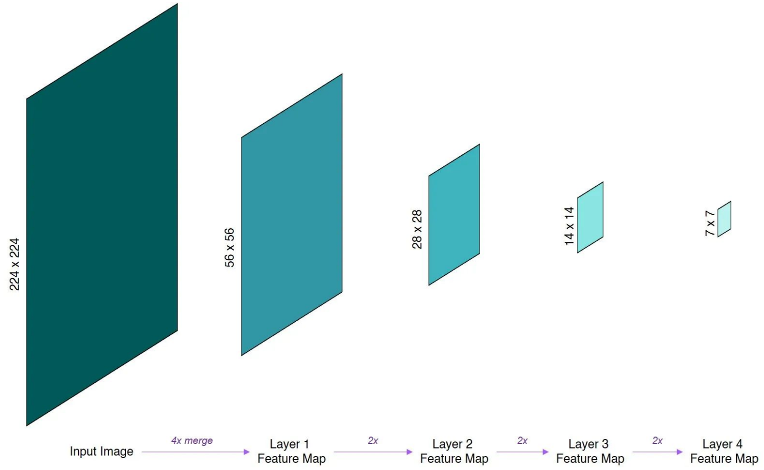


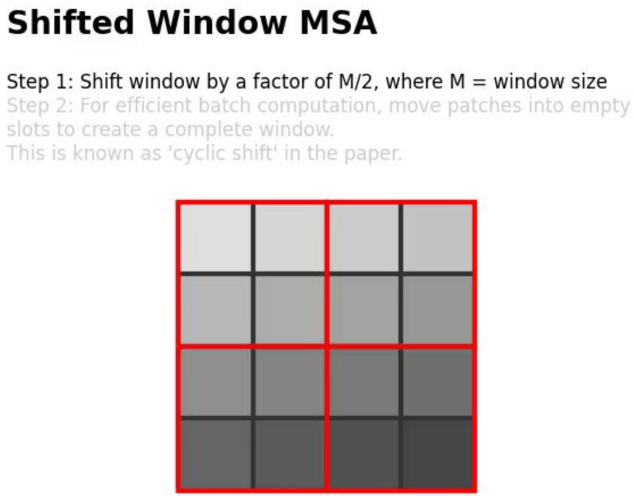


#SWIN Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. ICCV 2021 [Best paper award].

Hierarchical ViT Backbone

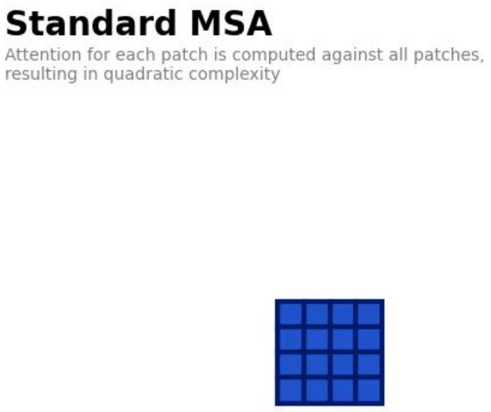
Hierarchical features maps across layers.

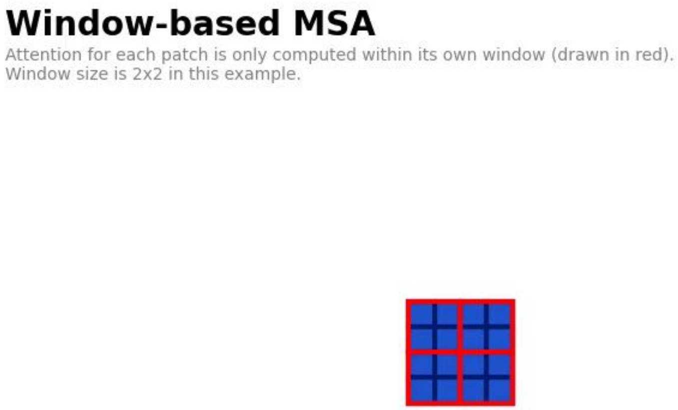




Figures: James Loy, A Comprehensive Guide to Microsoft's Swin Transformer, towardsdatascience.com

Hierarchical ViT Backbone

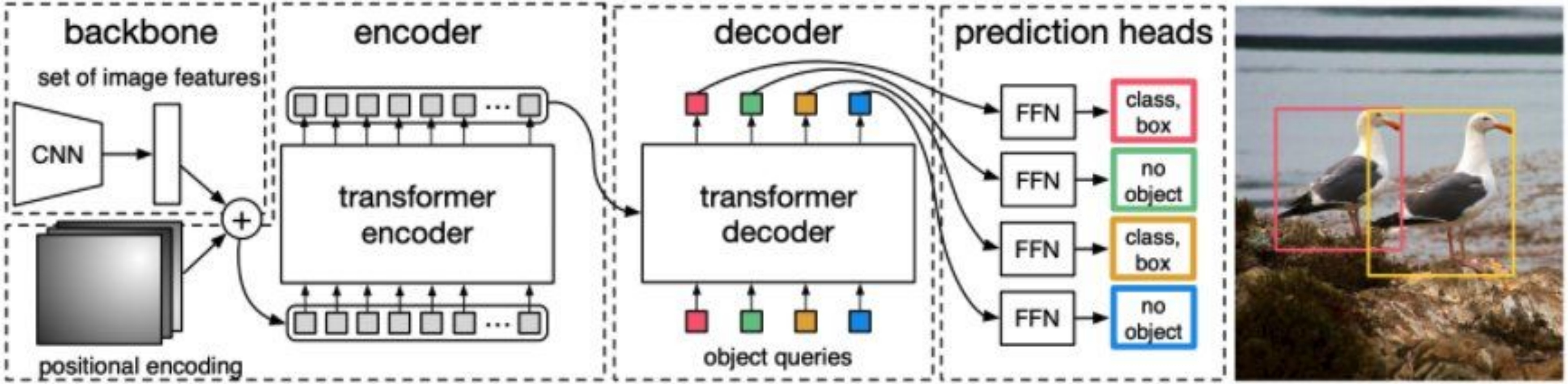




Figures: James Loy, [A Comprehensive Guide to Microsoft's Swin Transformer](), towardsdatascience.com

Object Detection

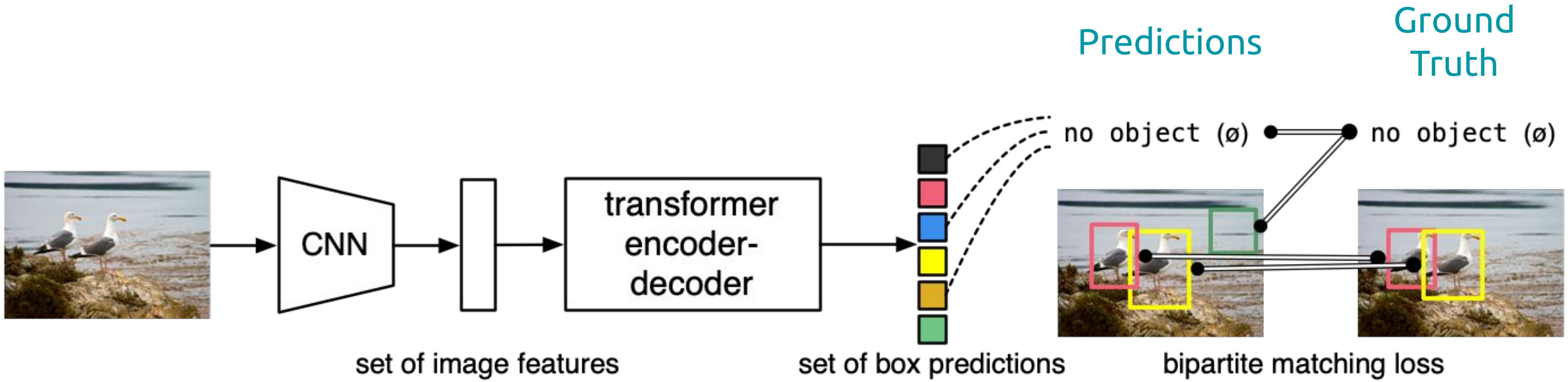
- Object detection formulated as a set prediction problem.
- DETR infers a fixed-size amount of predictions.
- Comparable performance to Faster R-CNN.



#DETR Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-End Object Detection with Transformers." ECCV 2020. [code] [colab]

Object Detection

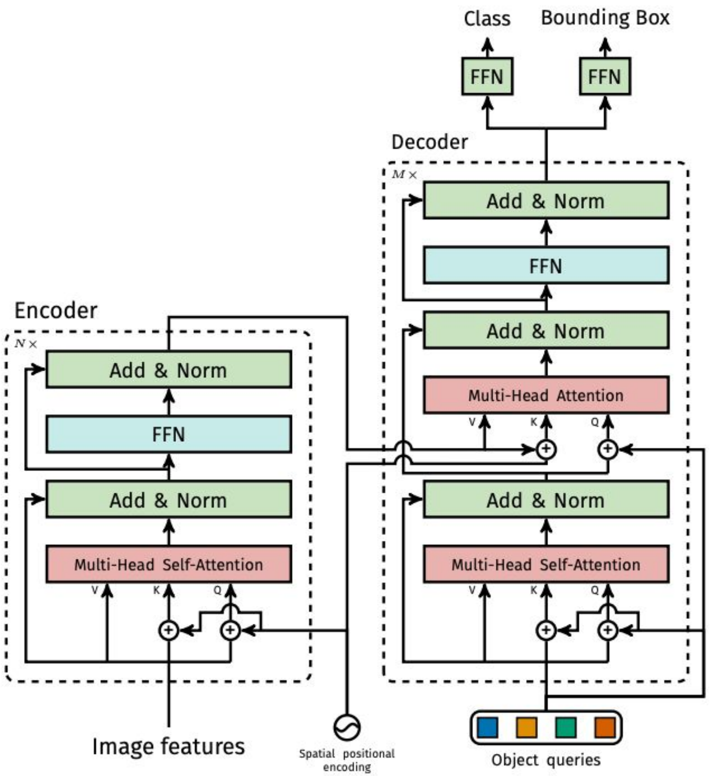
- During training, bipartite matching uniquely assigns predictions with ground truth boxes.
- Prediction with no match should yield a “no object” (∅) class prediction.



#DETR Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-End Object Detection with Transformers." ECCV 2020. [code] [colab]

Object Detection

- Architecture of DETR.



#DETR Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. ["End-to-End Object Detection with Transformers."]() ECCV 2020. [[code]]() [[colab]]()

Questions?