Deep Generative Models: Transformers for Vision

Fall Semester 2025

René Vidal

Director of the Center for Innovation in Data Engineering and Science (IDEAS)

Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

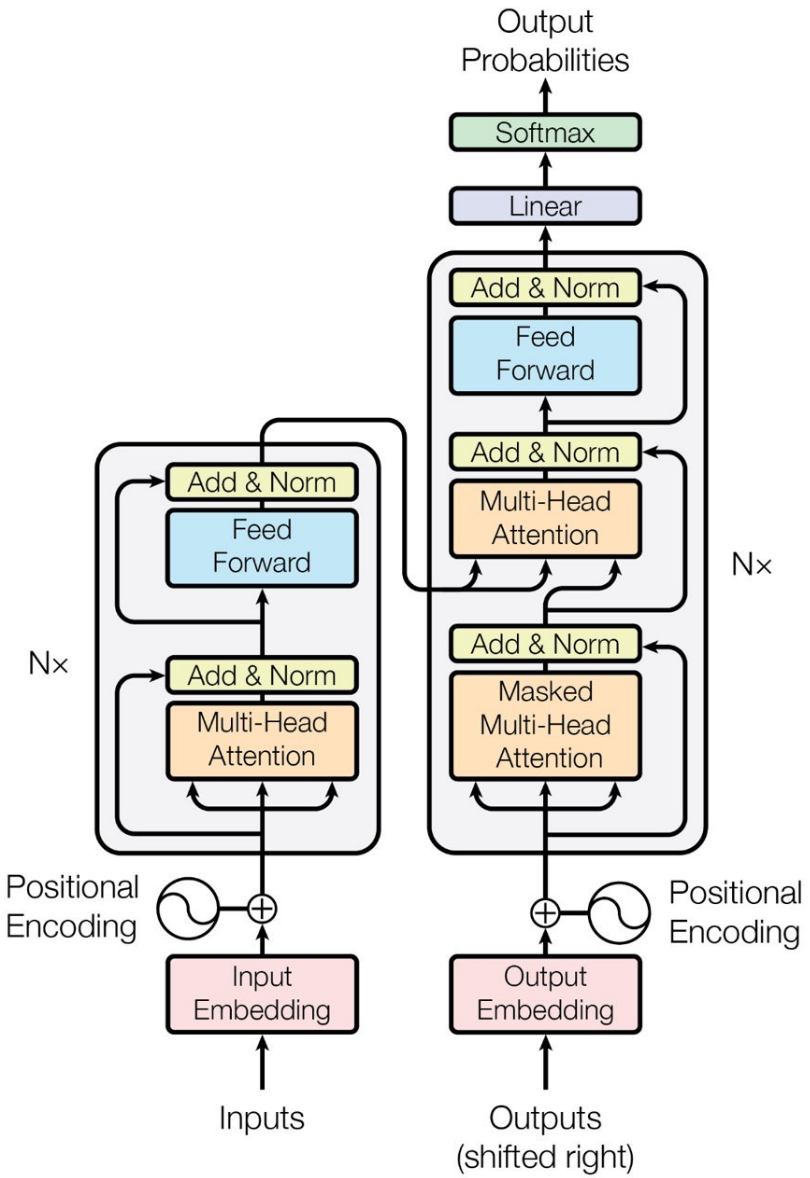
Last Two Lectures -> Today's Lecture

Natural Language Processing:

- Attention is all you need: Enc-Dec Transformer
- BERT (Bidirectional Encoder Representations from Transformers)
- GPT (Generative Pre-trained Transformer)
- RoBERTa (Robustly Optimized Bert Pre-training)
- T5 (Text-to-Text Transfer Transformer)

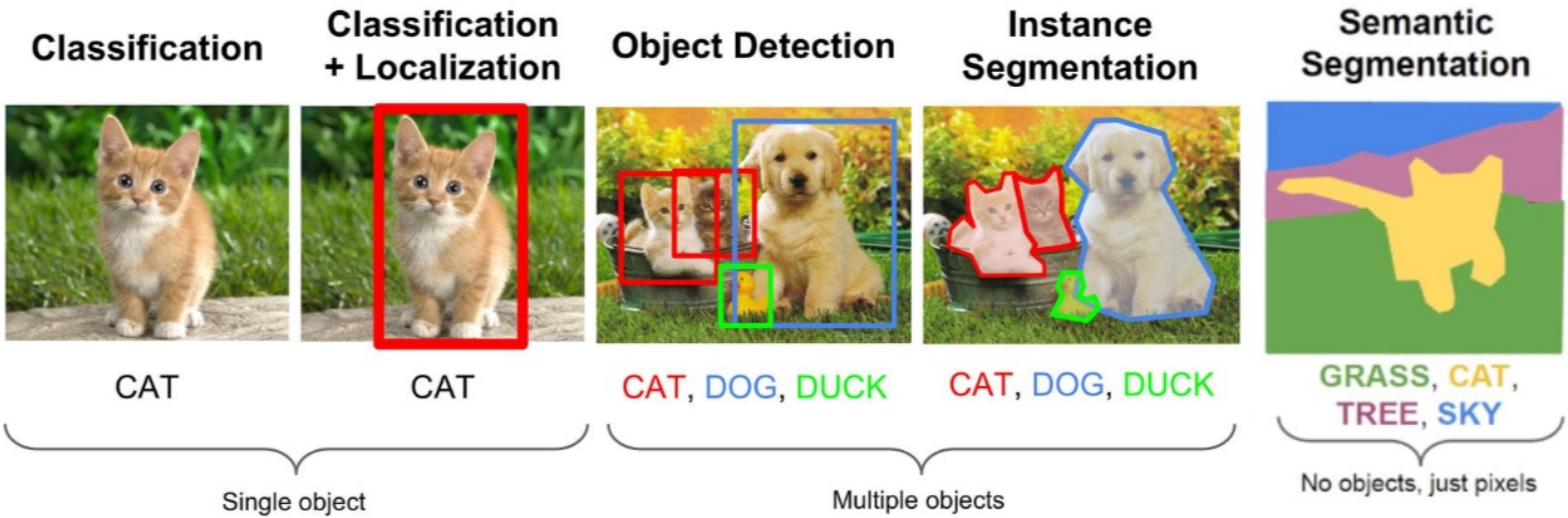
Computer Vision:

- Generative Pretraining from Pixels
- Vision Transformer
- Swin Transformer, Pyramid Vision Transformer



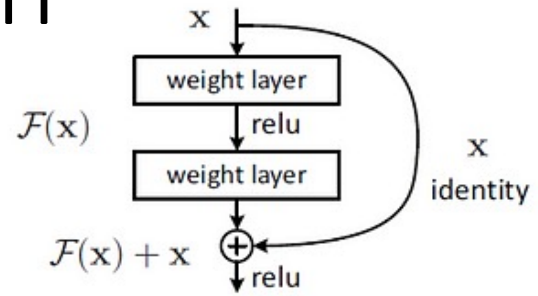
Computer Vision Tasks

- Computer Vision is the field of AI that enables machines to interpret and make decisions based on visual data. It uses a variety of algorithms to recognize, classify, and understand images or videos.
- Some key tasks in Computer Vision are:



Neural Networks in Vision

Building block of ResNet



1998
LeNet

ImageNet
2009

2010
Pascal VOC

AlexNet
(deepCNN)
2012

2014
VGG
R-CNN
GoogLeNet
(Inception)

Fast R-CNN
ResNet
2015

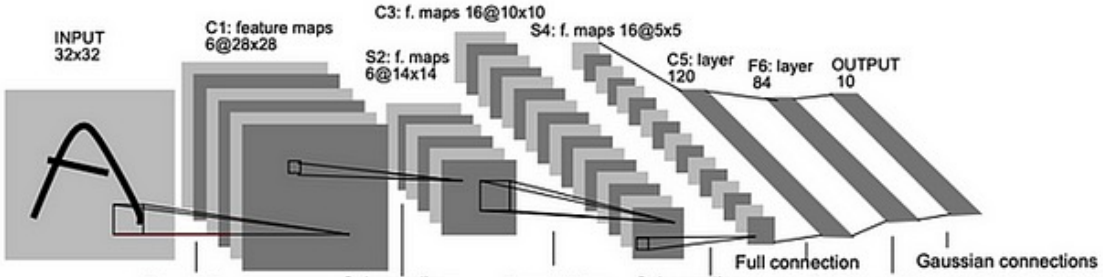
2016
SSD
YOLO
Inception-v4

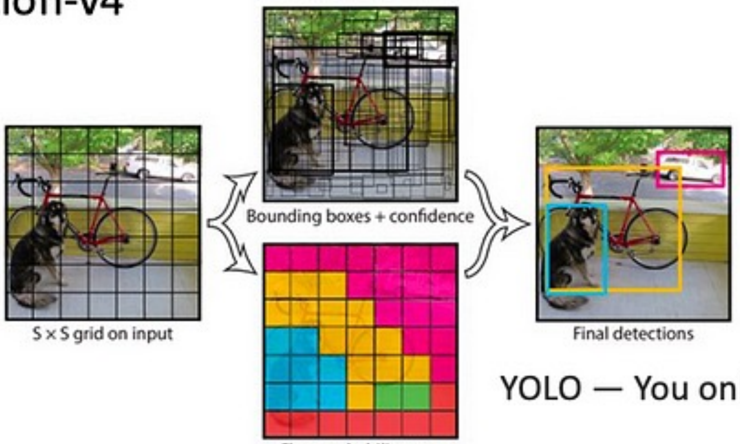
RevNet
YOLOv2
Transformer
2017

2018
YOLOv3
BERT

2020
Image GPT

Vision Transformer (ViT)
2021

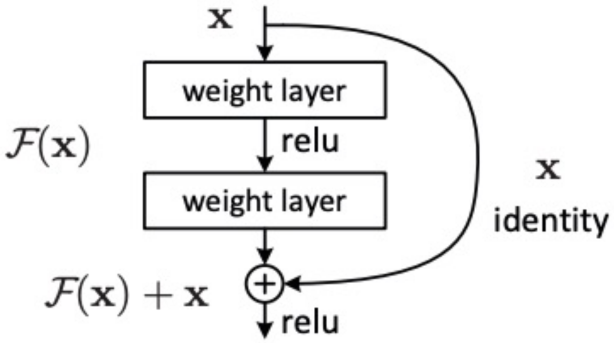


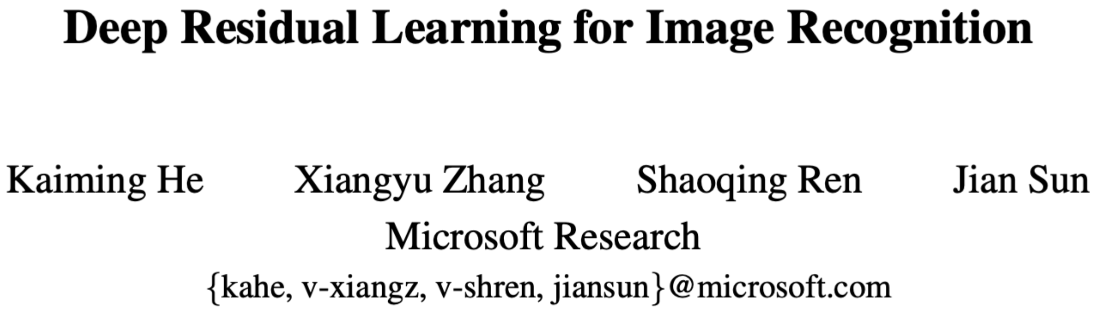


YOLO — You only look once

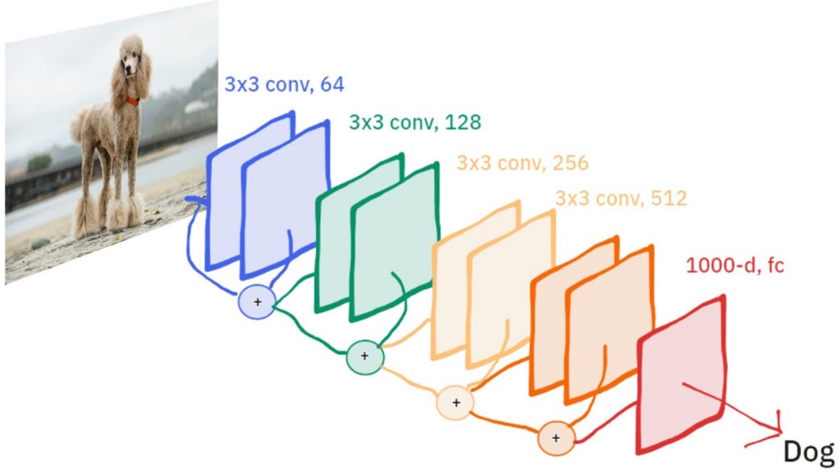
Pre-2019: Convolution Structures Dominated

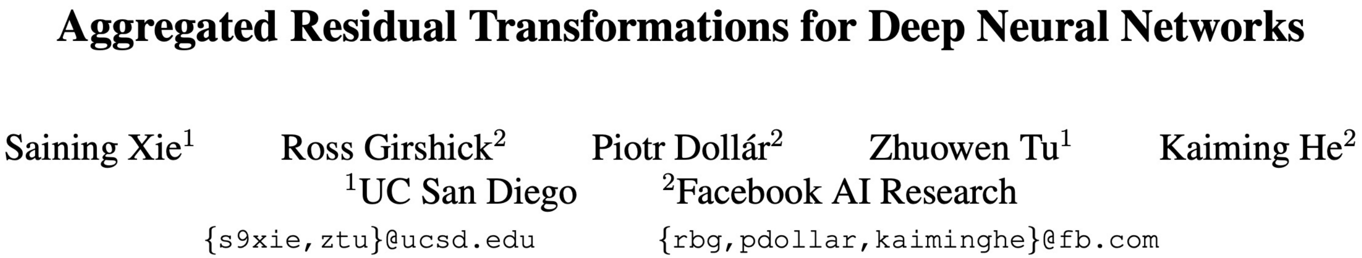
- In large-scale image recognition (e.g., ImageNet competitions), convolutional residual learning (e.g., ResNet and ResNeXt) architectures were still state of the art up to 2019.




Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

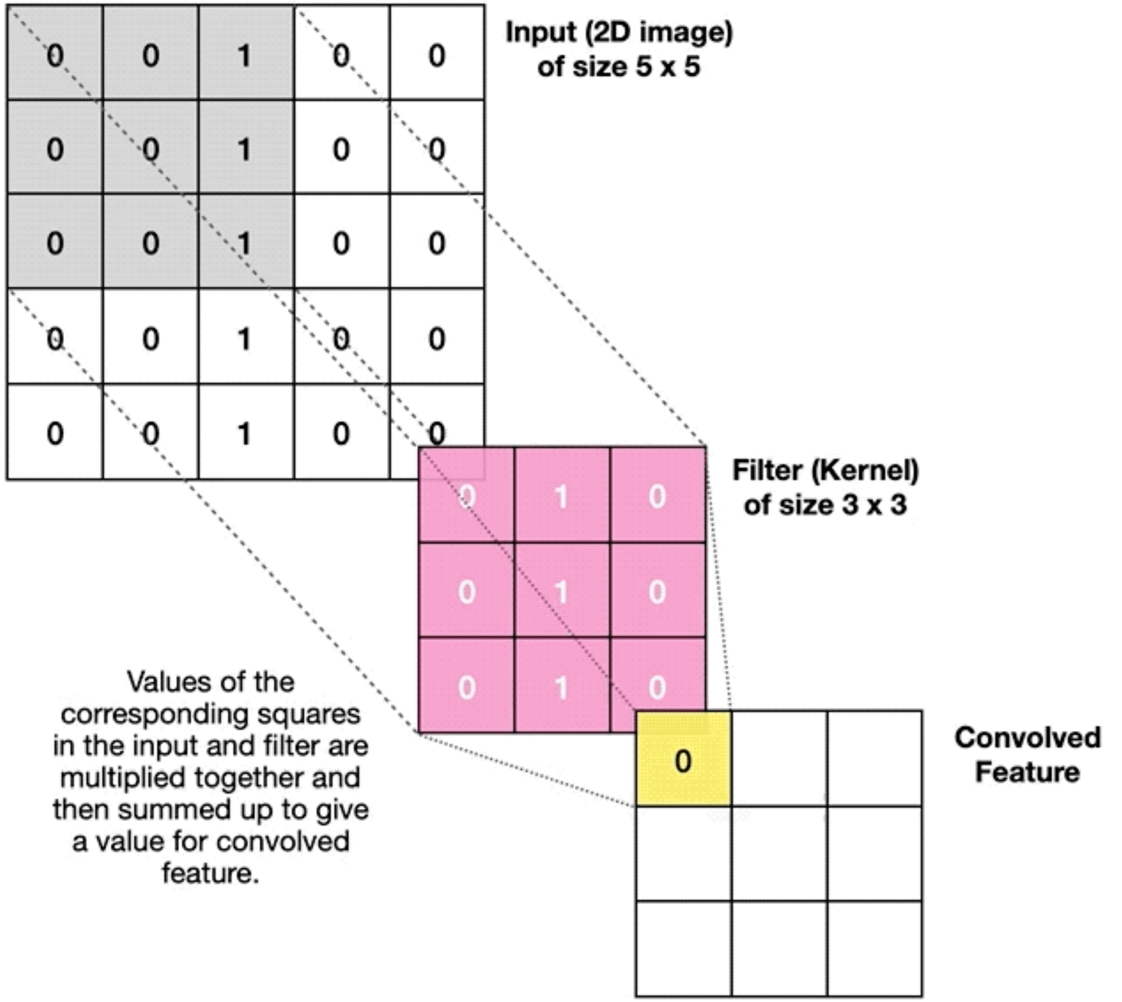


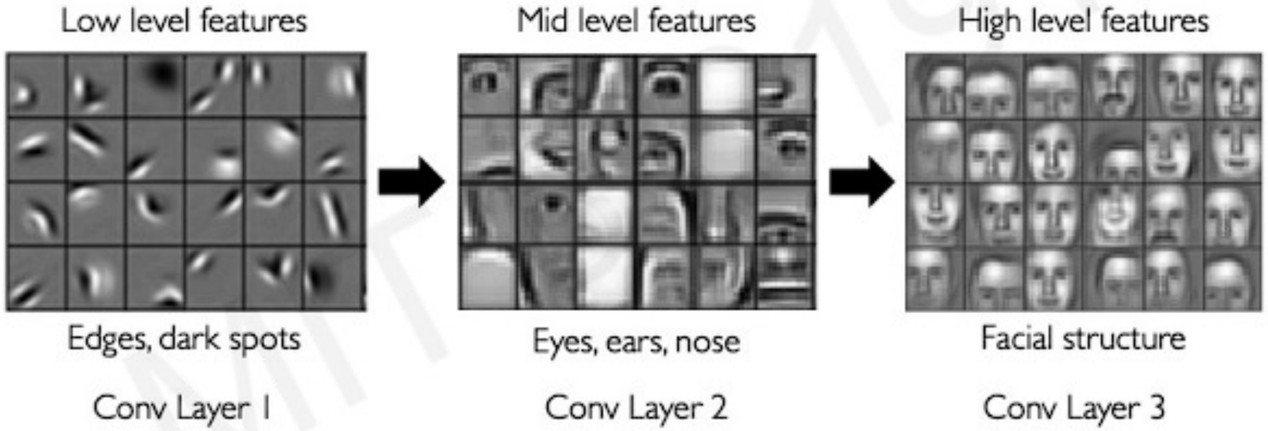

Aggregated Residual Transformations for Deep Neural Networks

Saining Xie[1] Ross Girshick[2] Piotr Dollár[2] Zhuowen Tu[1] Kaiming He[2]
[1]UC San Diego [2]Facebook AI Research
{s9xie,ztu}@ucsd.edu {rbg,pdollar,kaiminghe}@fb.com

Pre-2019: Convolution Structures Dominated

- In large-scale image recognition (e.g., ImageNet competitions), convolutional residual learning (e.g., ResNet and ResNeXt) architectures were still state of the art up to 2019.
  - A convolution operation involves sliding a filter or kernel across the image. Each position results in a weighted sum of the pixel values covered by the filter, producing a convolved feature map, highlighting various features such as edges, textures, and patterns.


Input (2D image) of size 5 x 5
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
Filter (Kernel) of size 3 x 3
0 1 0
0 1 0
0 1 0
Convolved Feature
0
Values of the corresponding squares in the input and filter are multiplied together and then summed up to give a value for convolved feature.



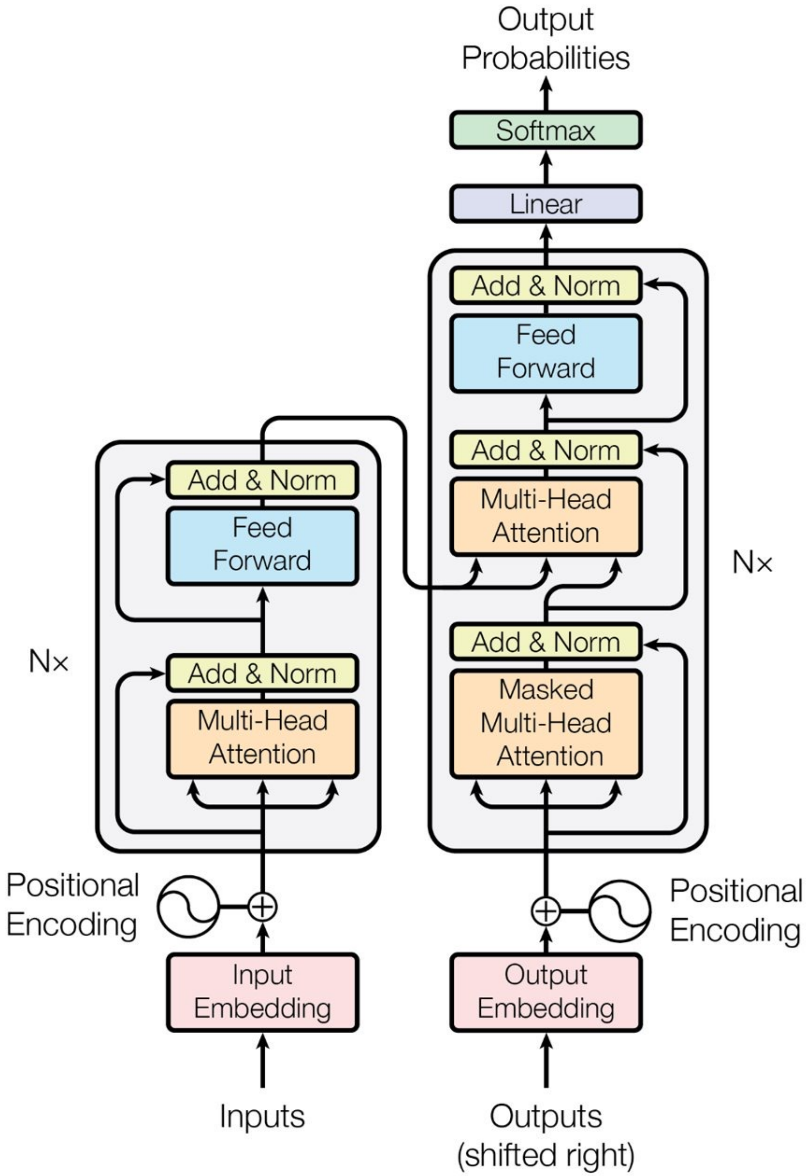
Vision Transformer: Transformer for the CV domain

- Self-attention-based architectures, in particular Transformers, have become the model of choice in natural language processing (NLP).
- The learning paradigm with Foundation Models emerges: Researchers now get the **pretraining** on a large text corpus and then **fine-tune/inference** on a smaller task-specific dataset.
- Transformers' **computational efficiency and scalability** make it a suitable choice for such pretraining. We can now train NLP models of unprecedented size, with over 100B parameters (e.g., GPT-4, LLaMA).

- But how about the computer vision (CV) community? Can we apply the same success story in the CV domain?

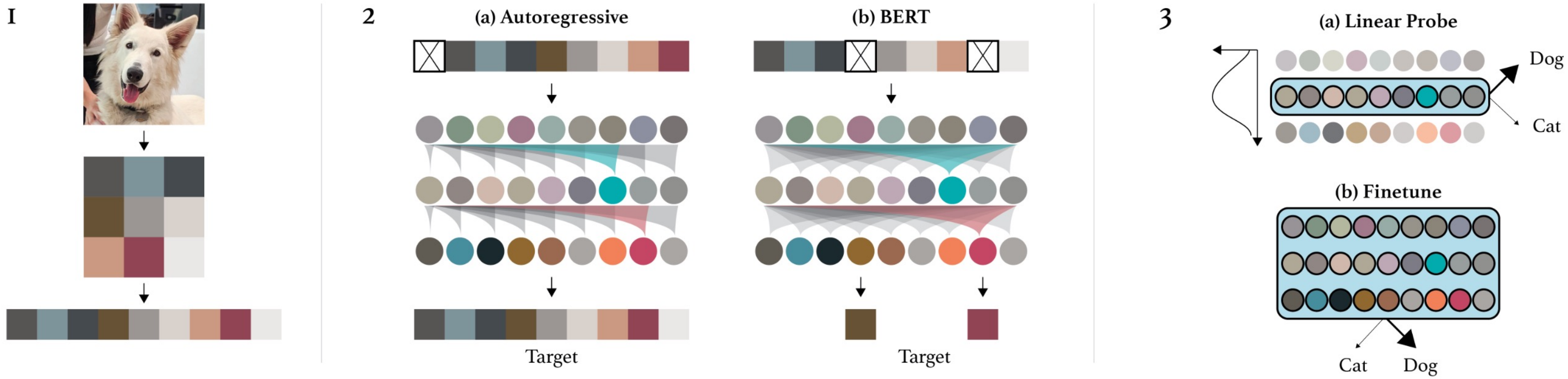
Transformers for CV: Transformer Overview

- Input Tokens
  - How do we tokenize an image in a manner similar to text tokenization?
    - In language tasks, we use words or subwords as tokens. What could be the equivalent for images?
- Input Embedding
  - How do we embed these image tokens?
    - In NLP, embedding tables work for discrete word tokens, but pixels are continuous. How can we effectively embed image pixels?
- Positional encoding
- Multi-head attention
- Add & Norm
- Feed Forward Net



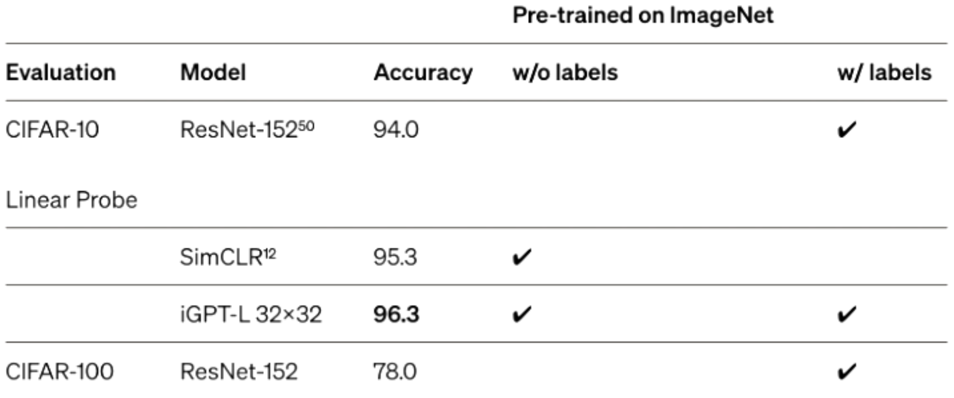
ImageGPT: Generative Pretraining from Pixels [2020]

- Treat color value from each pixel as a discrete token!
  - Typically represented as a 24-bit value ([0-255] per color channel) (vocab size of ~16.7M).
  - Reduction: We may not need to store that many colors?
    - A 9-bit representation ([0-8] per color channel) reduces vocabulary size to 512.
- However, Transformers have quadratic complexity $O(n^2)$ w.r.t. token length.
  - For a 256x256 image, we would have 65,536 tokens (BERT max length was 512).
- Solution: just use lower resolution images (maximum size of 64x64).
- Trained on a similar objective to BERT (predict the next/masked pixels).

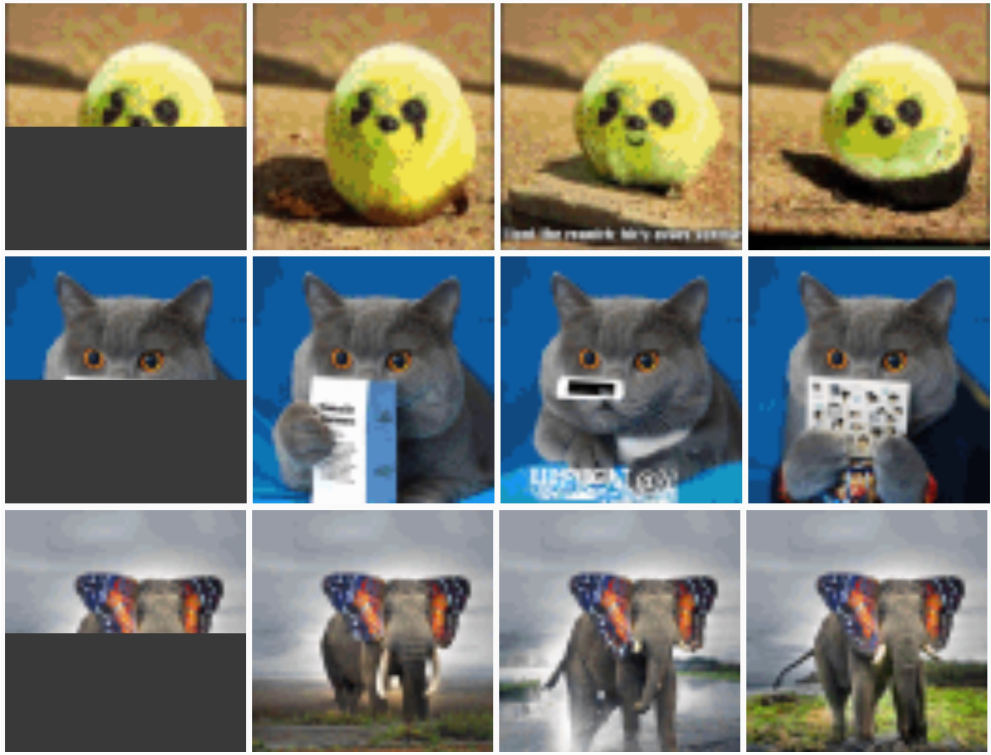


ImageGPT: Generative Pretraining from Pixels [2020]

- **Model Variants:**
  - iGPT-S, iGPT-M, iGPT-L:
    - Parameters: 76M, 455M, 1.4B respectively.
    - Trained on ImageNet.
  - iGPT-XL:
    - 6.8B parameters, trained on ImageNet + additional web images.
- **Key Outcomes:**
  - Good image representations.
    - Was SOTA on semi-supervised classification.
  - Good image generations.
    - Shown to be effective at modeling visual information.
- **Training complexity:**
  - **iGPT-L** was trained for roughly **2500** V100-days.
  - ResNet equivalent model trained in **70** V100-days.
  - And this is just for **64x64** resolution images!

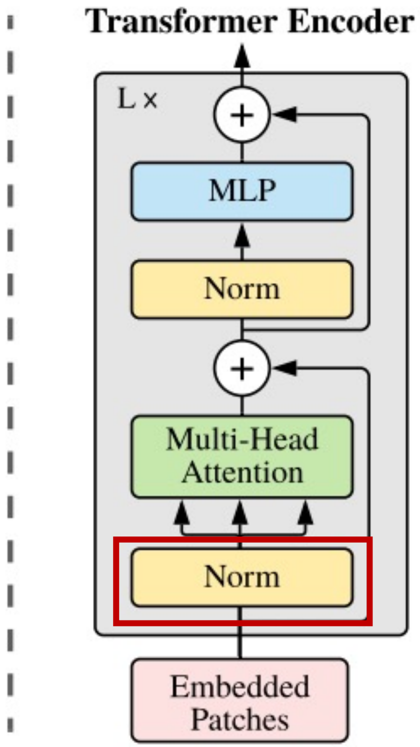


			Pre-trained on ImageNet	
Evaluation	Model	Accuracy	w/o labels	w/ labels
CIFAR-10	ResNet-152[50]	94.0		✓
Linear Probe				
	SimCLR[12]	95.3	✓	
	iGPT-L 32×32	**96.3**	✓	✓
CIFAR-100	ResNet-152	78.0		✓



Vision Transformer: An Image is Worth 16x16 Words

- Rather than quantizing pixels, **Vision Transformer** splits an image into **patches (16x16 pixels)**, which are flattened and linearly embedded to form tokens.
- Adds learnable positional embeddings to retain spatial information of patches.
- Adds a [CLS] token as an additional input to the transformer encoder.
  - After processing, the representation of the [class] token is used for image classification.



Vision Transformer (ViT)

- To handle 2D images, ViT reshapes the image of shape (H, W, C) into a sequence of flattened 2D patches of shape (P^2, C).
- (H, W) is the resolution of the original image, C is the number of channels, P is the width of each image patch.
- We then get $N = \frac{H \times W}{P^2}$: the number of patches (as the input sequence length).

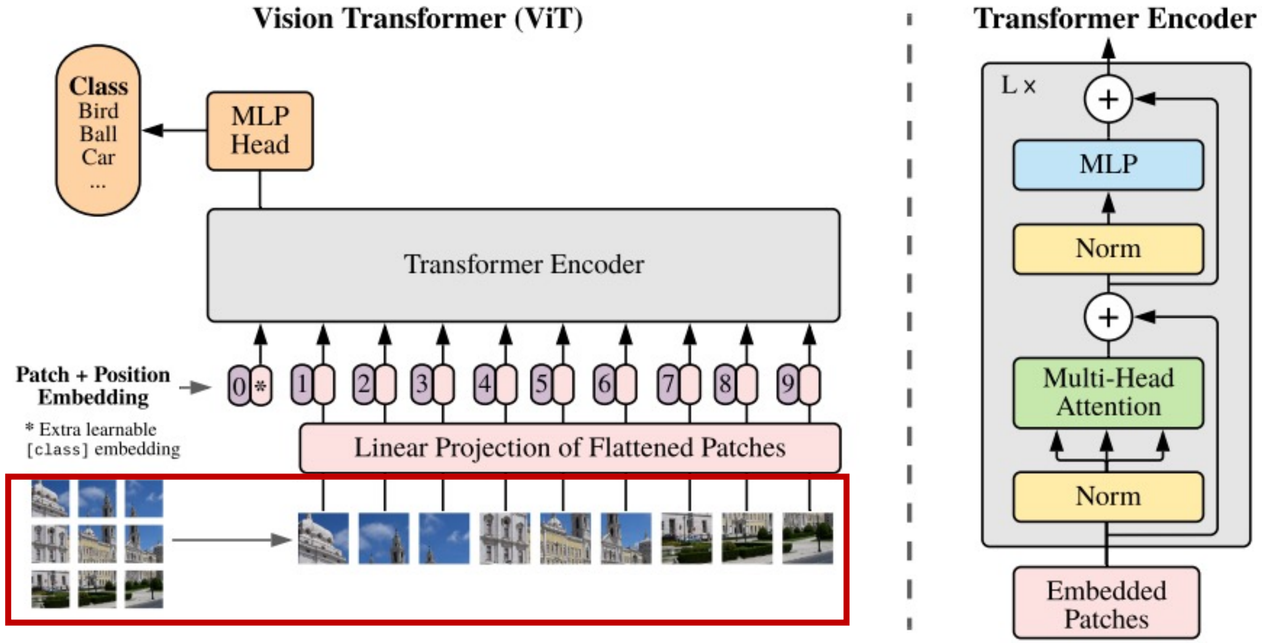


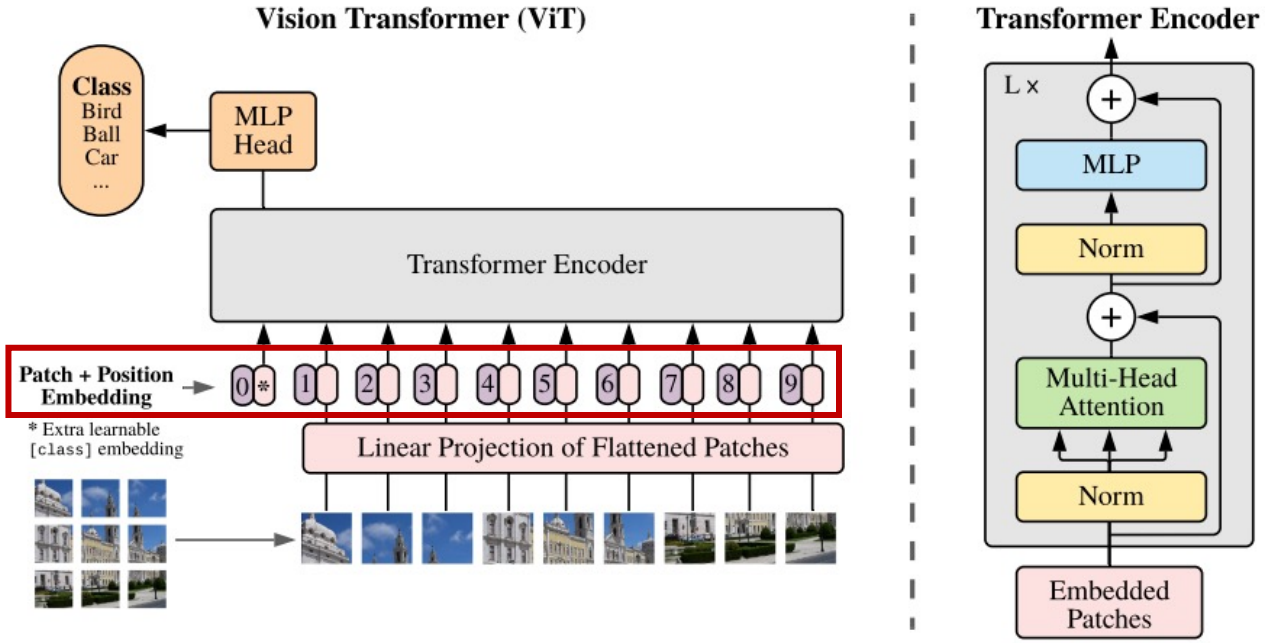
image to patch sequence

Vision Transformer (ViT)

- **Position embeddings (E_{pos})** are added to the **patch embeddings ($x_p E$)** to retain positional information. ViT uses learnable 1D position embeddings.

- Similar to BERT's [class] token, ViT prepends a learnable embedding for image class to the beginning of the embedding sequence, whose state serves as the task representation for image classification.

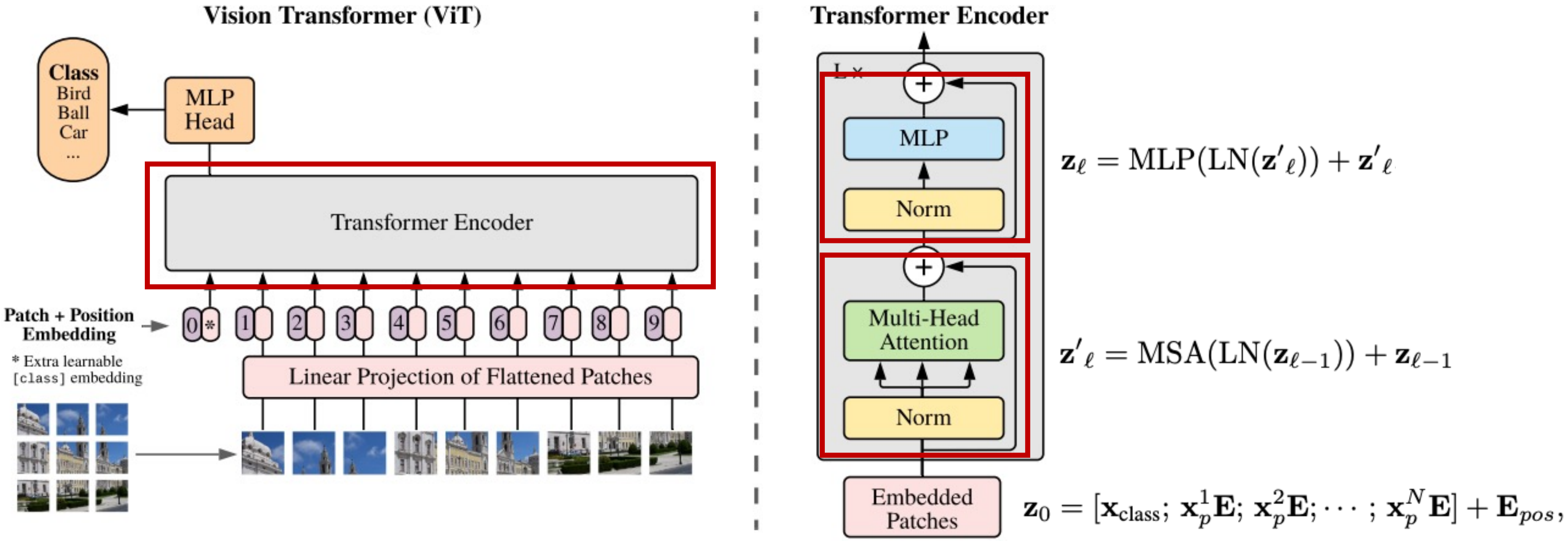
positional and image class embedding



Vision Transformer (ViT)

- Transformer encoder consists of alternating layers of Multi-Head Self-Attention (MSA) and MLP blocks.
- LayerNorm (LN) is applied before every block, and residual connections after every block.
  - Pre-Norm configurations tend to help improve the gradient flow during training.
  - This approach has been adopted in updated official implementation of the Transformer.

Recall the Transformer architecture from last week.



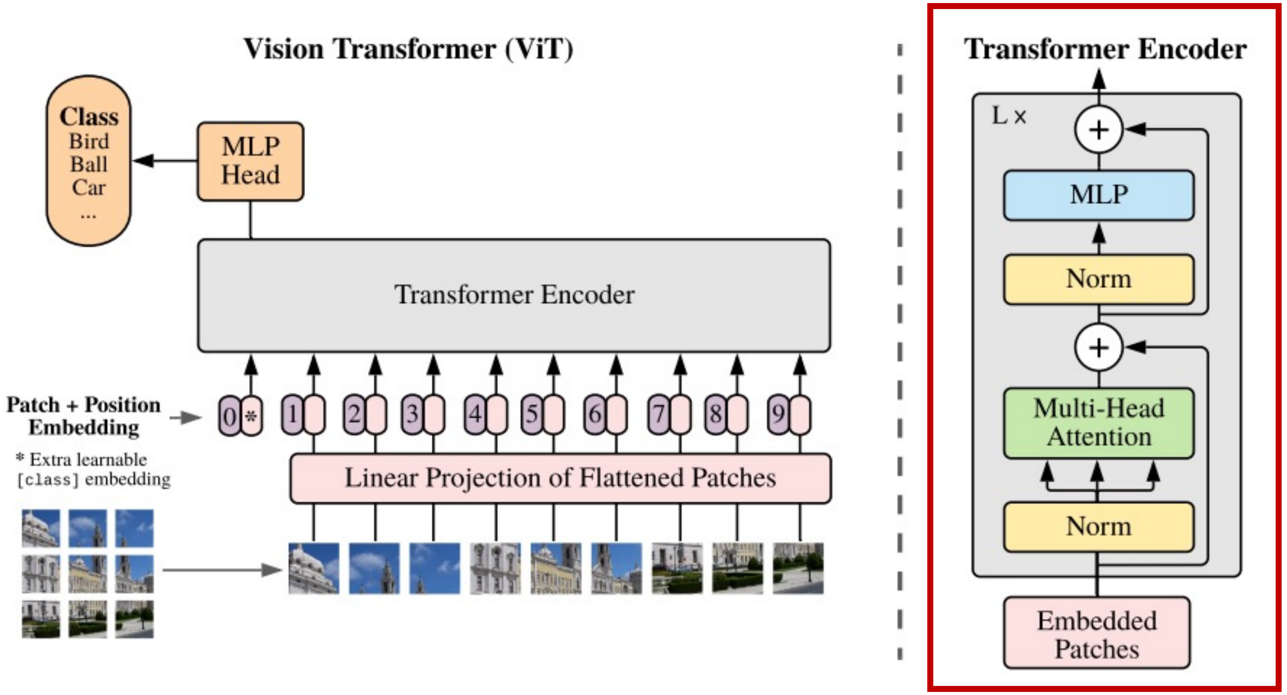
$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell$$

$$\mathbf{z}'_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}$$

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\, \cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos},$$

Vision Transformer (ViT)

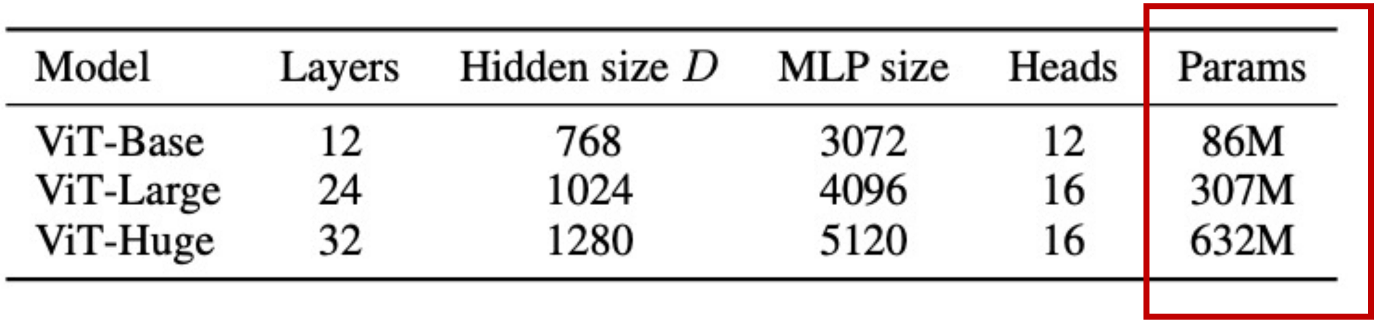
- ViT has less built-in image-specific assumptions compared to CNNs.
  - CNNs use local receptive fields and shared weights, making them better suited for capturing spatial patterns.
  - ViT operates on image patches without assuming local structure or spatial hierarchies.
    - It relies on self-attention to capture relationships, making it more flexible but less biased towards spatial locality. Each patch can attend to every other patch, providing a global context from the start.
    - However, ViT requires more data or pre-training to learn spatial relationships effectively due to the lack of locality bias.



Less image-specific inductive bias

ViT Scalability and Attention Visualization

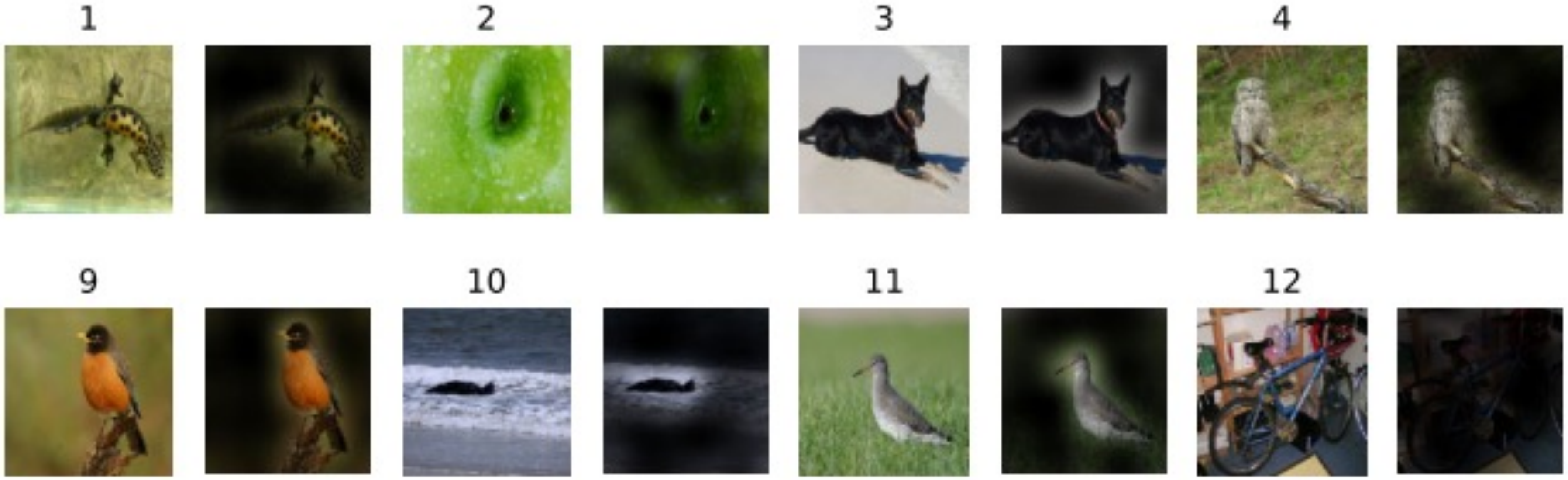
- A popular choice of the transformer encoder is Bidirectional Encoder Representations from Transformers (BERT). The “Base” and “Large” models are adopted from BERT.



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

Scaled-up compared to previous CNNs

- Visualization of attention values from the output token to the input space.

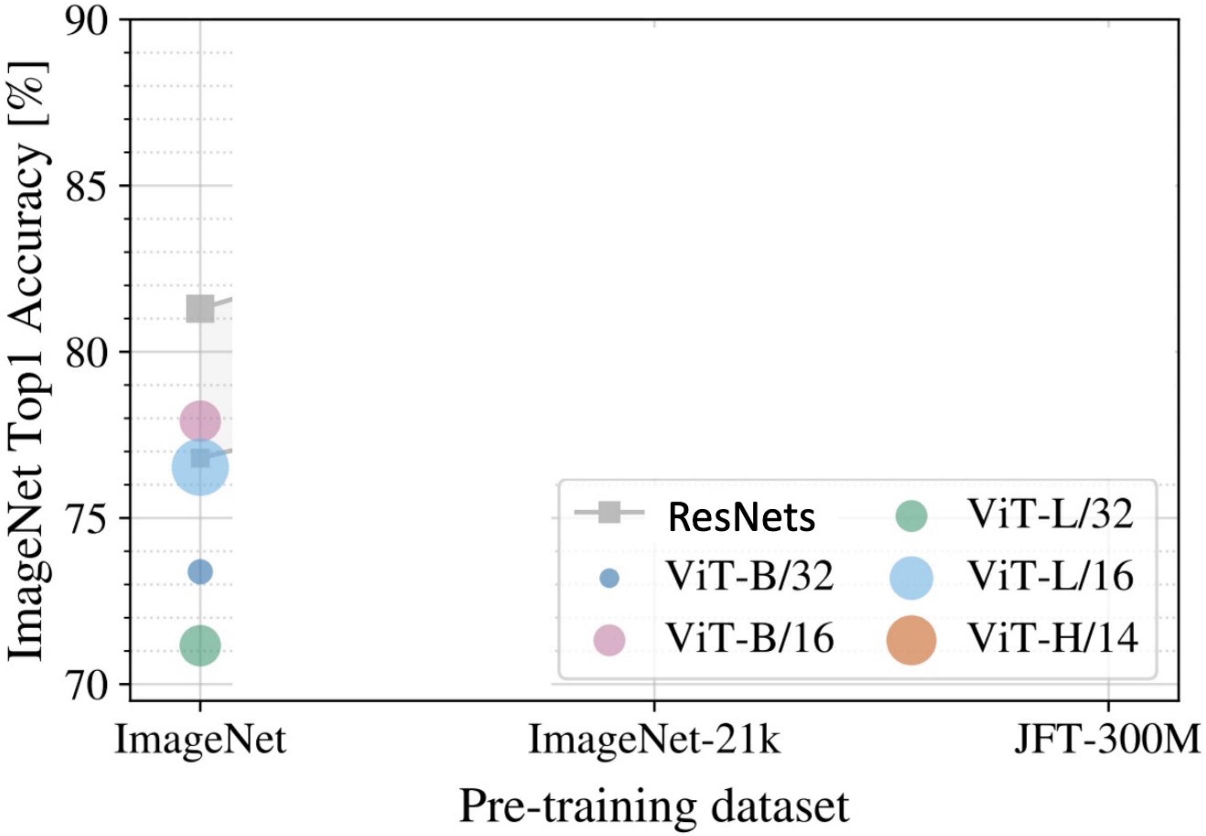


ViT Results on ImageNet

- A popular choice of the transformer encoder is Bidirectional Encoder Representations from Transformers (BERT). The “Base” and “Large” models are adopted from BERT.

ImageNet dataset has 1k categories, 1.2M Images.

When trained on ImageNet, ViT models perform worse than ResNets



B = Base
L = Large
H = Huge

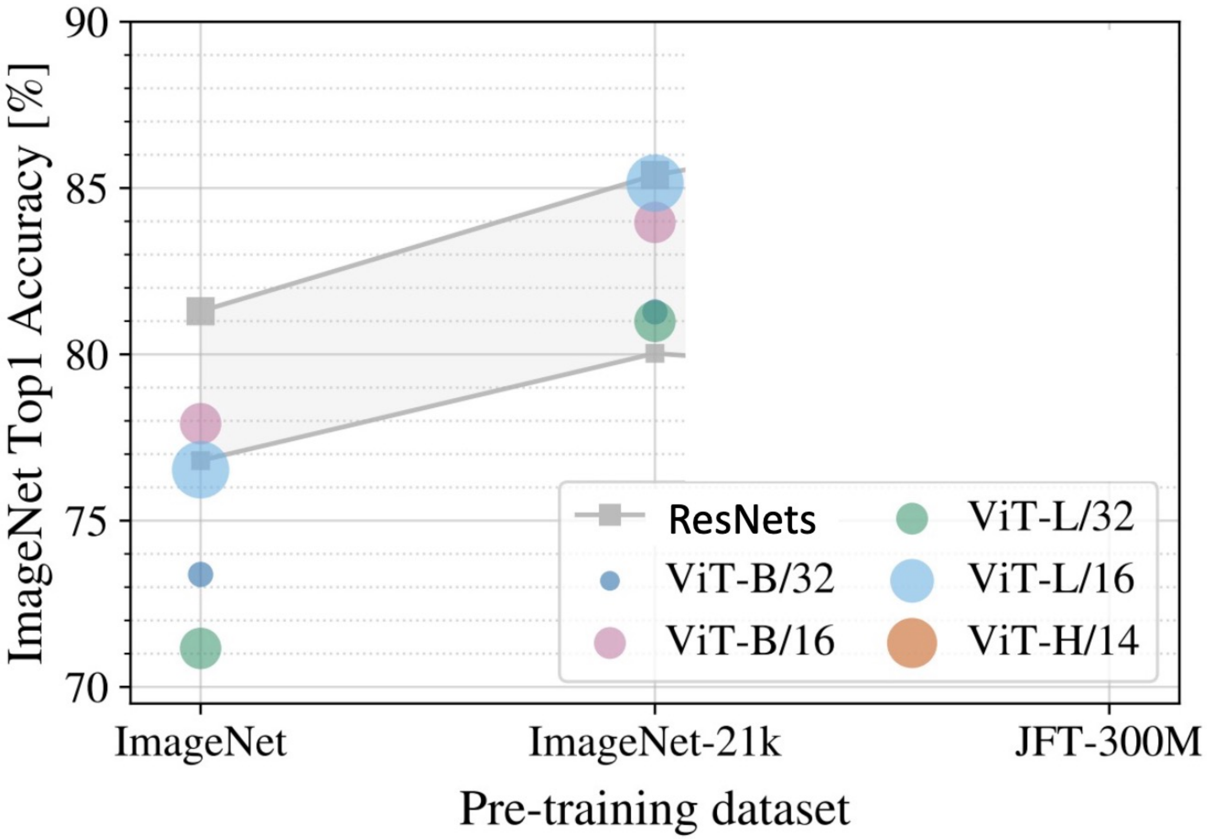
/32, /16, /14 is patch size; smaller patch size is a bigger model (more patches)

ViT Results on ImageNet

- A popular choice of the transformer encoder is Bidirectional Encoder Representations from Transformers (BERT). The “Base” and “Large” models are adopted from BERT.

ImageNet-21k has 14M images with 21k categories.

If you pretrain on ImageNet-21k and finetune on ImageNet, ViT does better: big ViTs match big ResNets



B = Base
L = Large
H = Huge

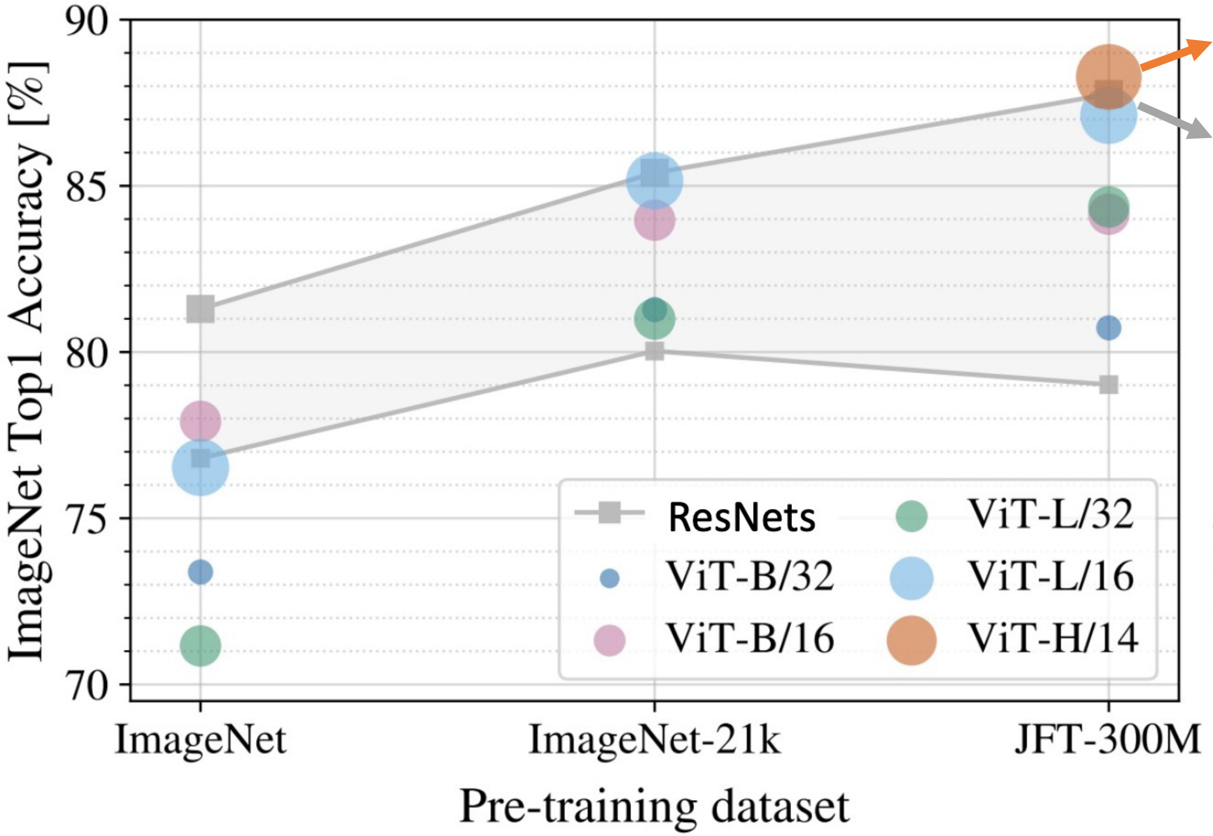
/32, /16, /14 is patch size; smaller patch size is a bigger model (more patches)

ViT Results on ImageNet

- A popular choice of the transformer encoder is Bidirectional Encoder Representations from Transformers (BERT). The “Base” and “Large” models are adopted from BERT.

JFT-300M is an internal Google dataset with 300M labeled images.

If you pretrain on JFT and finetune on ImageNet, large ViTs outperform large ResNets.



ViT: 2.5k TPU-v3 core days of training (~$33K)

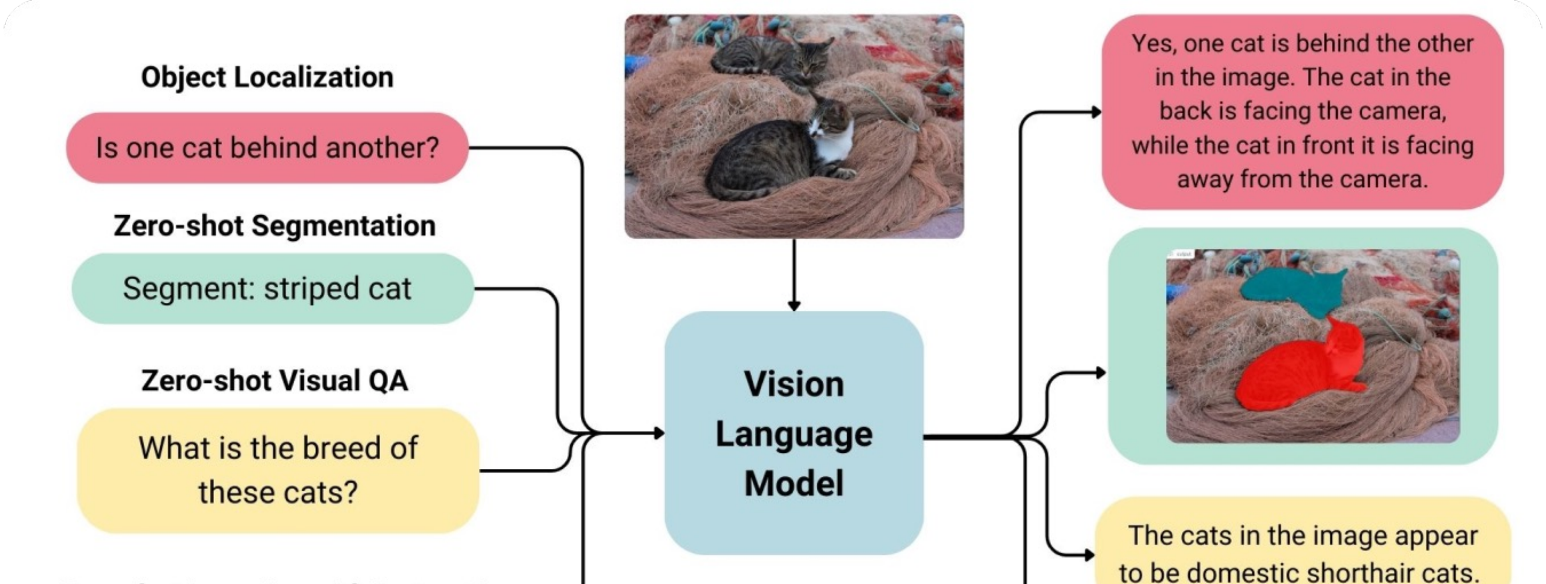
ResNet: 9.9k TPU-v3 core days of training (~$133k)

B = Base
L = Large
H = Huge

/32, /16, /14 is patch size; smaller patch size is a bigger model (more patches)

ViT: Conclusion

- Introduced a paradigm shift in computer vision by applying pure Transformer architectures to image recognition, replacing classical convolutional designs.
  - Demonstrated scalability: performance improves with model and data size, aligning vision with NLP trends.
  - Established patch-based tokenization as a viable representation of visual data, supporting unified modeling of images and text (next lecture).
    - Influenced multimodal models (e.g., CLIP) by bridging vision and language through shared transformer backbones.
  - Inspired extensive follow-up research, including DeiT, Swin Transformer, ViViT, and ViLT.



Object Localization

Is one cat behind another?

Zero-shot Segmentation

Segment: striped cat

Zero-shot Visual QA

What is the breed of these cats?

Vision Language Model

Yes, one cat is behind the other in the image. The cat in the back is facing the camera, while the cat in front it is facing away from the camera.

The cats in the image appear to be domestic shorthair cats.