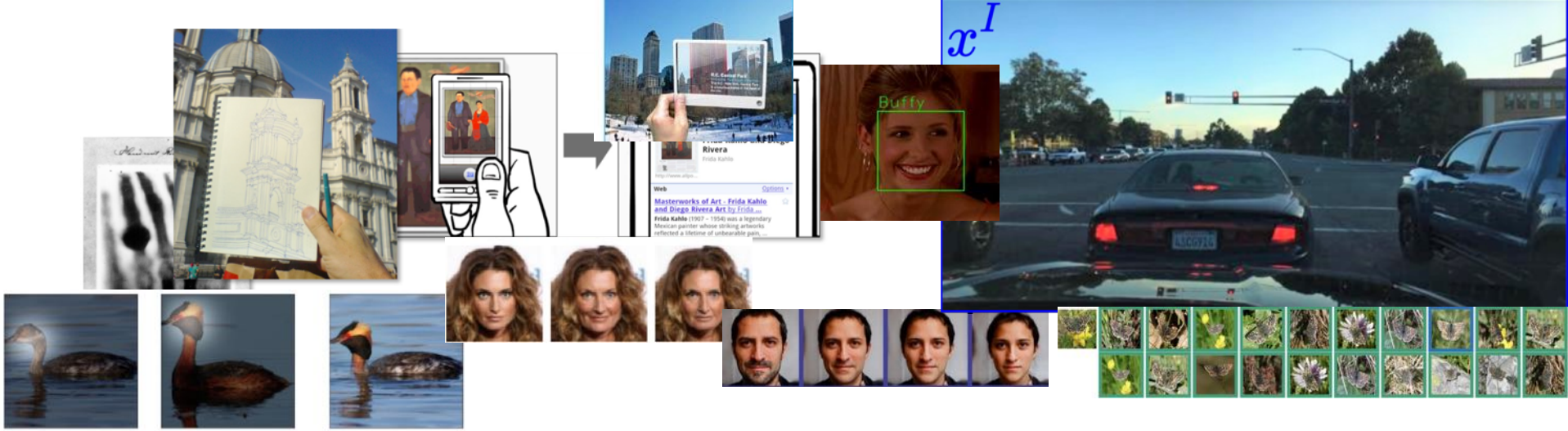
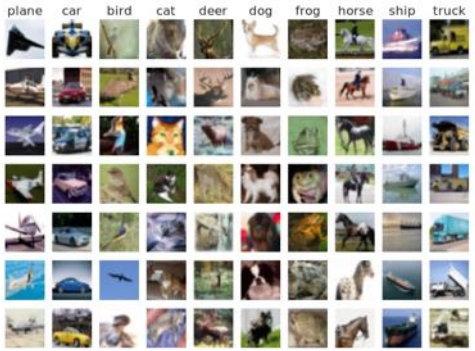
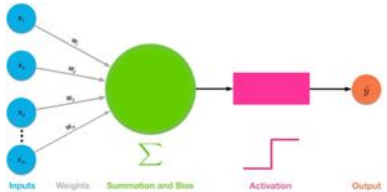


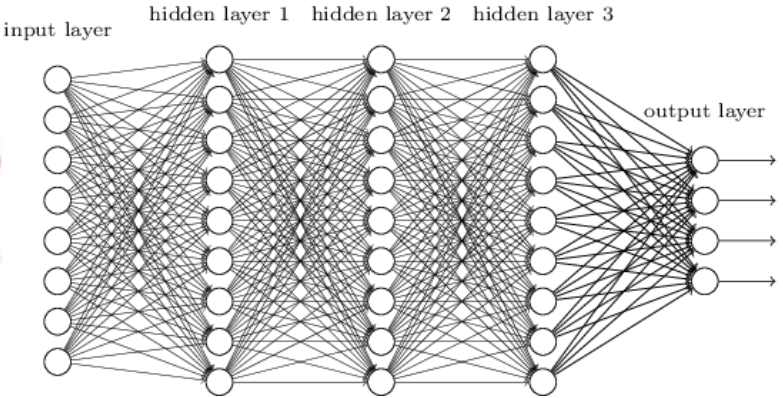
Vision Transformers

Matthieu Cord
SCAI center, ISIR lab
Sorbonne University, valeo.ai

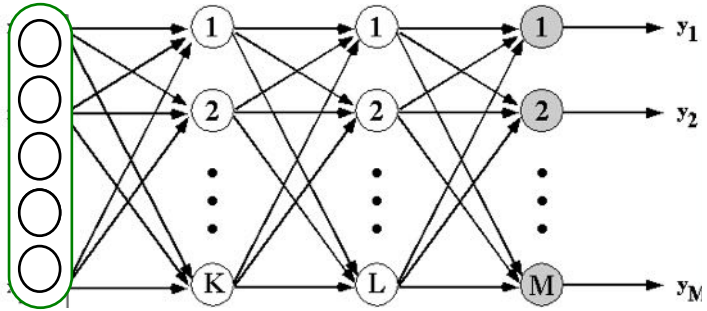
Preamble (before Transformers): Data, Training, Neural Nets







x



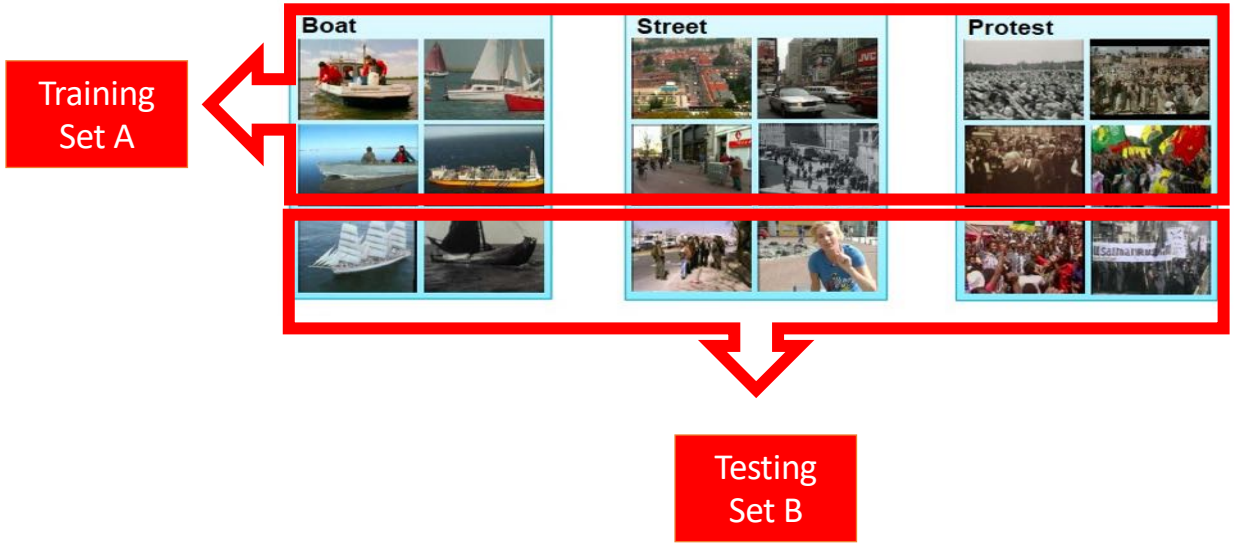
y(x)

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

Training: 1.2 Million images, 1000 classes

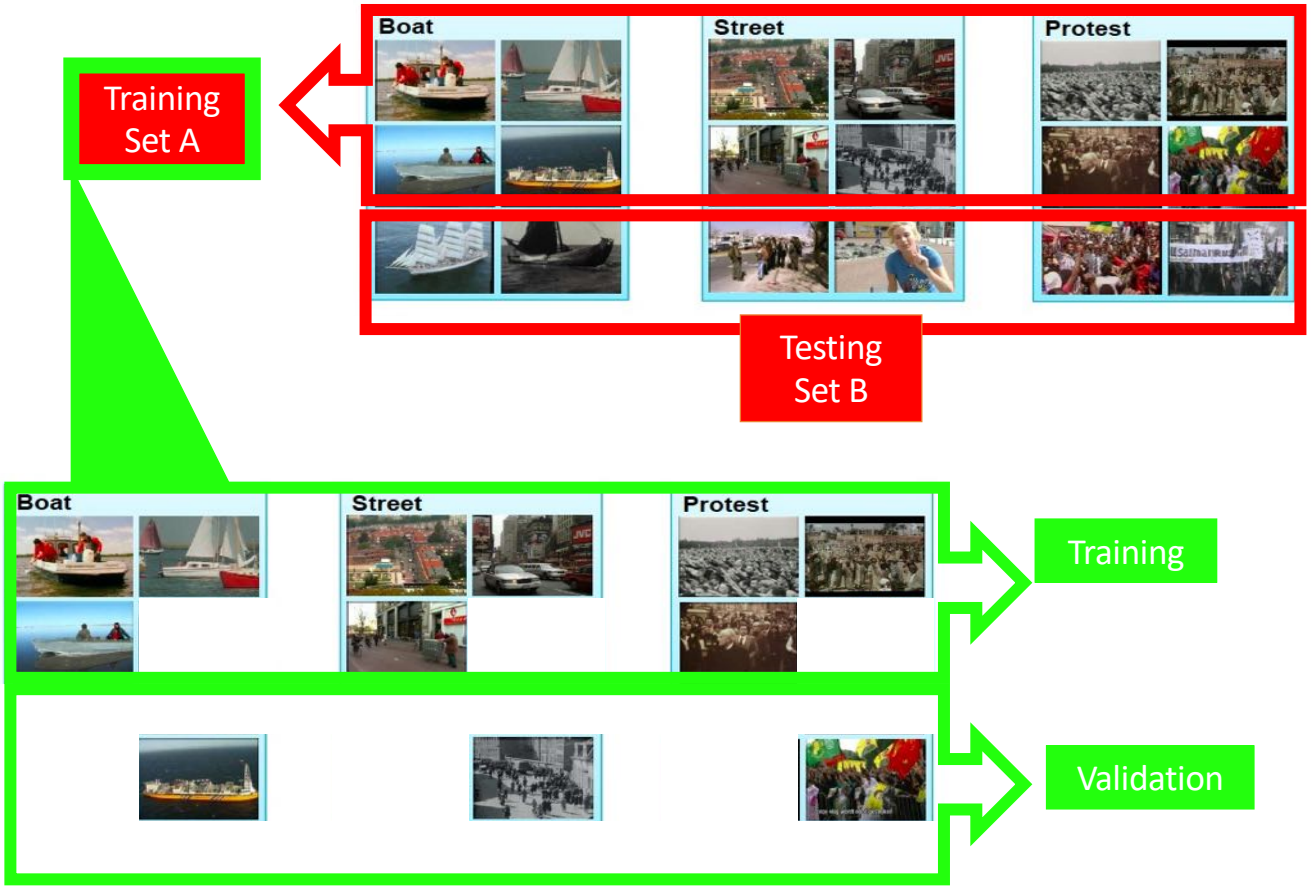
Paper: ImageNet: A Large-Scale Hierarchical Image Database, Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, **CVPR 2009**

Image/video datasets for training/testing



- Training classifiers on A
- Testing on B: error evaluation
- A and B disjoints!

Training: Cross-validation

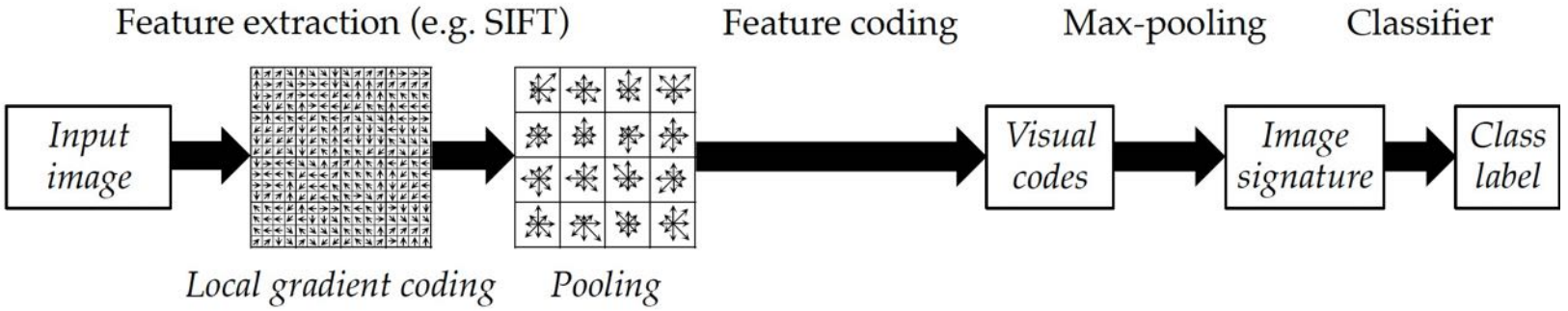


Context: Image classification **Before** ImageNet

The 2000s: *BoWs image modeling + SVMs* for Visual Classification

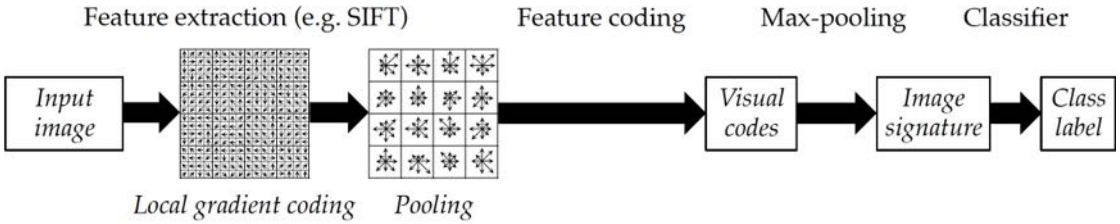
2 steps:

1. From input image to Vector representation
   - BoW = Bag of (Visual) Words
   - Visual features => SIFT descriptors
2. From Vector to classes: Classifier (SVM, NN, …)



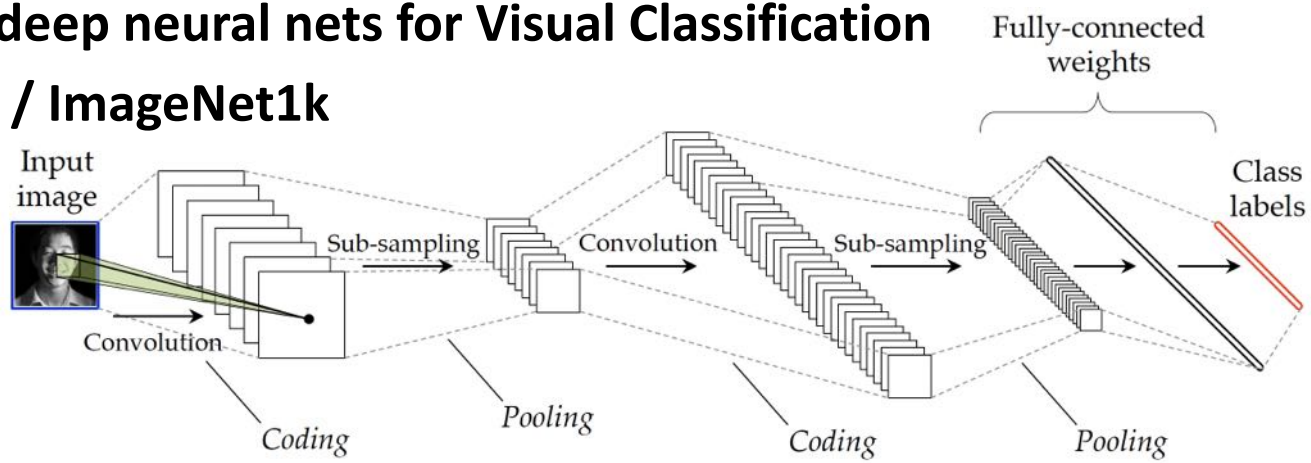
Context: Image classification **After** ImageNet (2009)

The 2000s: *BoWs image modeling + SVMs* for Visual Classification

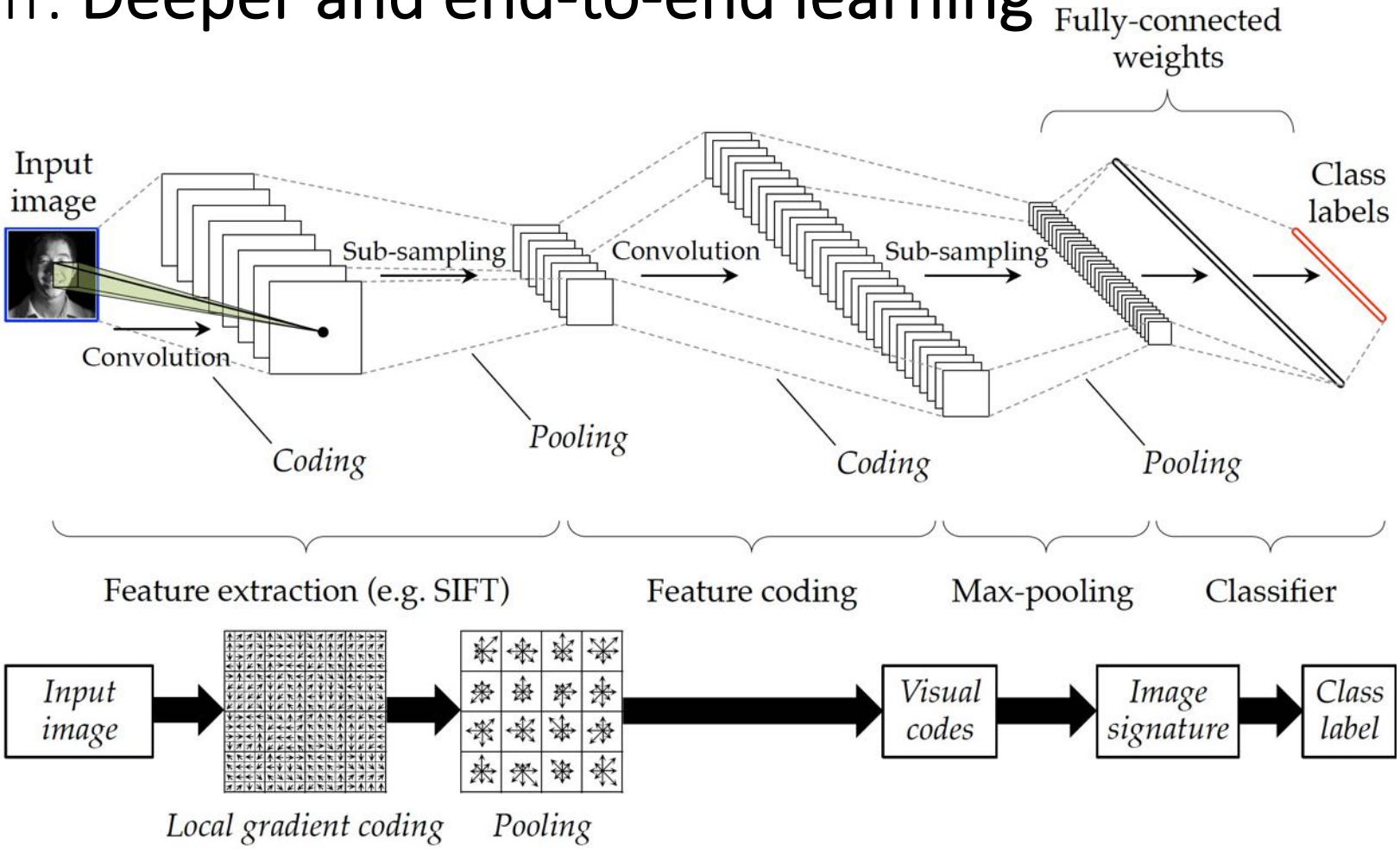


The 2010s: *Large* deep neural nets for Visual Classification

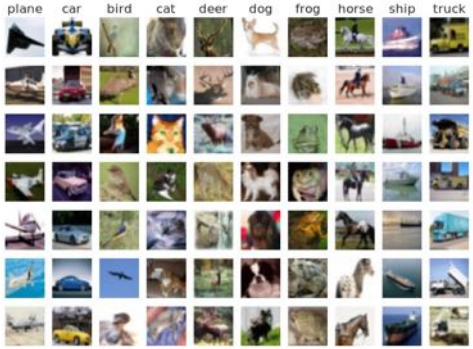
The star: **ConvNet / ImageNet1k**

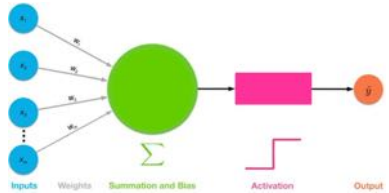


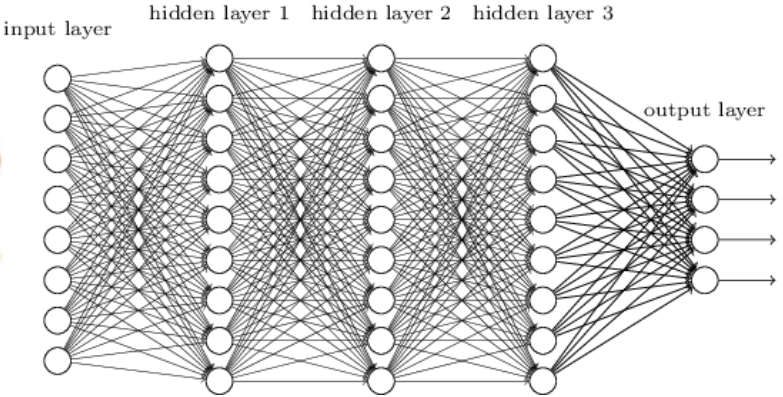
Diff: Deeper and end-to-end learning



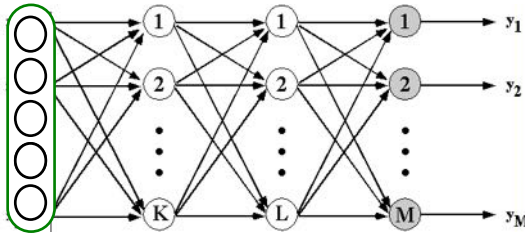
Preamble (before Transformers): Neural Nets







x



y(x)

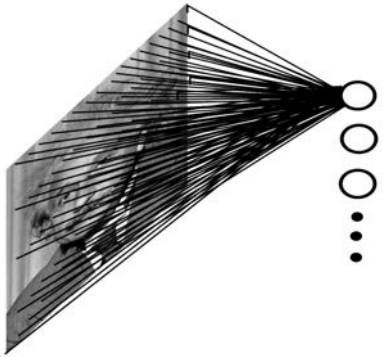
Pb: Scalability

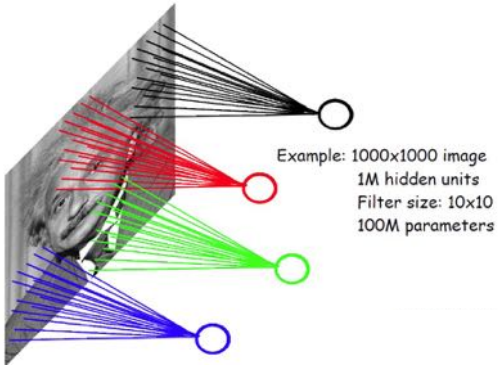
Large images => extremely large number of trainable parameters

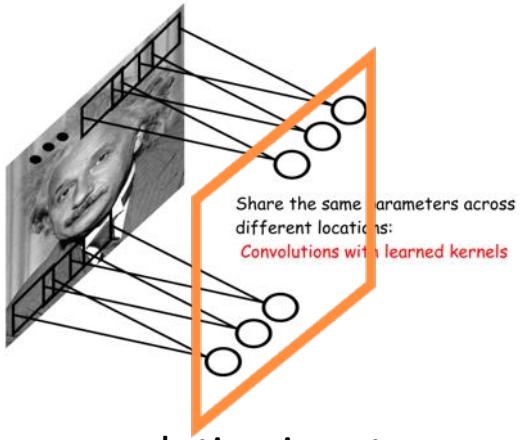
Exploit 2D image structure Knowledge

From fully connected to convolutional Neural Networks

- ~~Fully~~ **Locally** connected weights
- Keep spatial information in a 2D feature map (hidden layer map)
- Hidden nodes at different locations share the same weights
    - greatly reduces the number of parameters to learn







⇒ Computing responses at hidden nodes equivalent to convoluting input image with a linear filter (learned)

⇒ Learned filter == feature detector

Step aside: (1D/2D) convolution

1D discrete convolution of input signal x[n], with filter impulse response h[n], and output y[n]:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

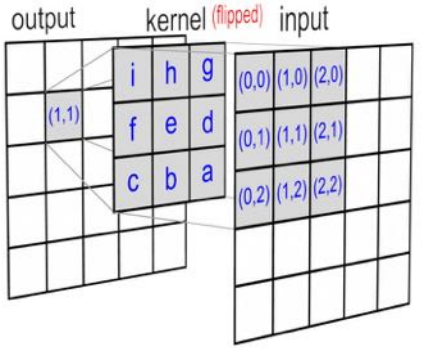
2D discrete convolution of input signal x[m,n], with filter impulse response h[m,n] (*kernel*), and output y[m,n]:

$$y[m,n] = x[m,n] * h[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[m-i, n-j]$$

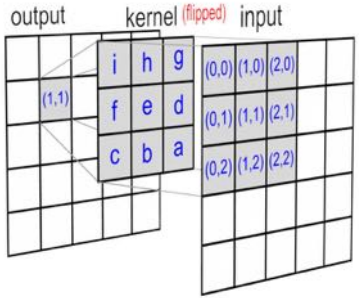
Example with impulse response (kernel) 3x3, and it's values are a, b, c, d,... :
(0,0) located in the center of the kernel

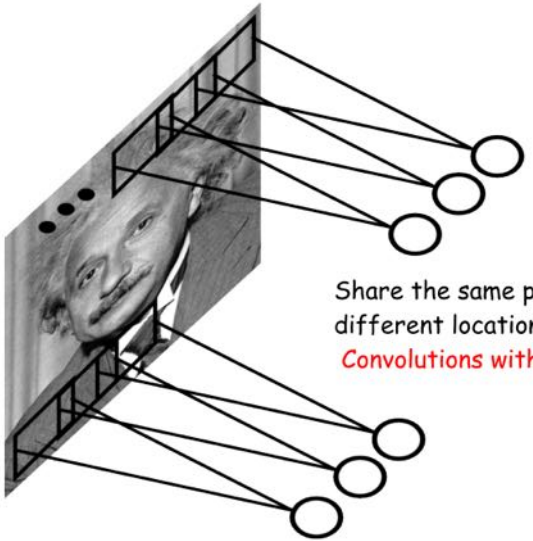
n \ m	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

$$\begin{aligned} y[1,1] &= \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[1-i, 1-j] \\ &= \quad x[0,0] \cdot h[1,1] \quad + x[1,0] \cdot h[0,1] \quad + x[2,0] \cdot h[-1,1] \\ &\quad + x[0,1] \cdot h[1,0] \quad + x[1,1] \cdot h[0,0] \quad + x[2,1] \cdot h[-1,0] \\ &\quad + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1] \end{aligned}$$

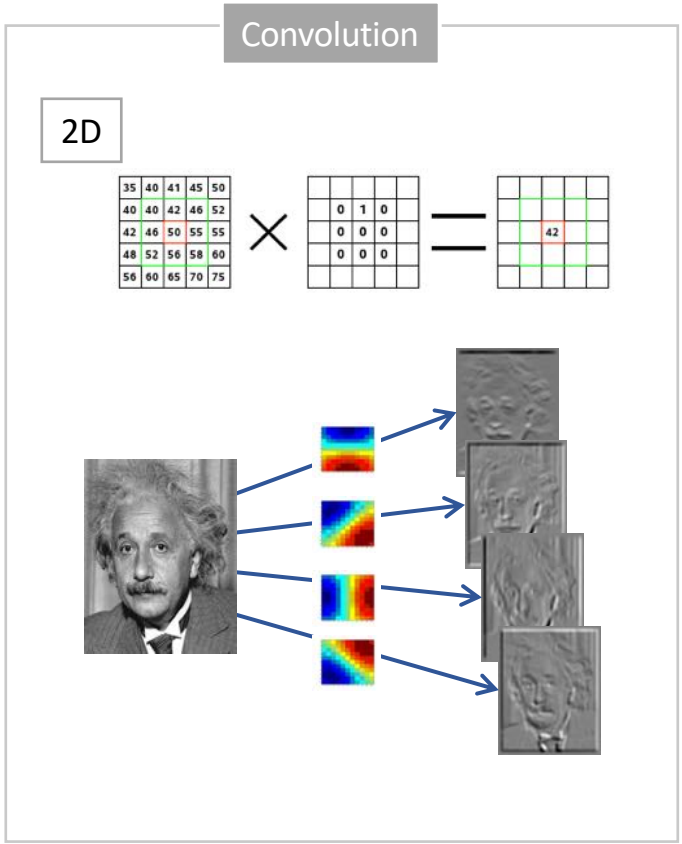


Step aside: convolution operator





Share the same parameters across different locations:
Convolutions with learned kernels

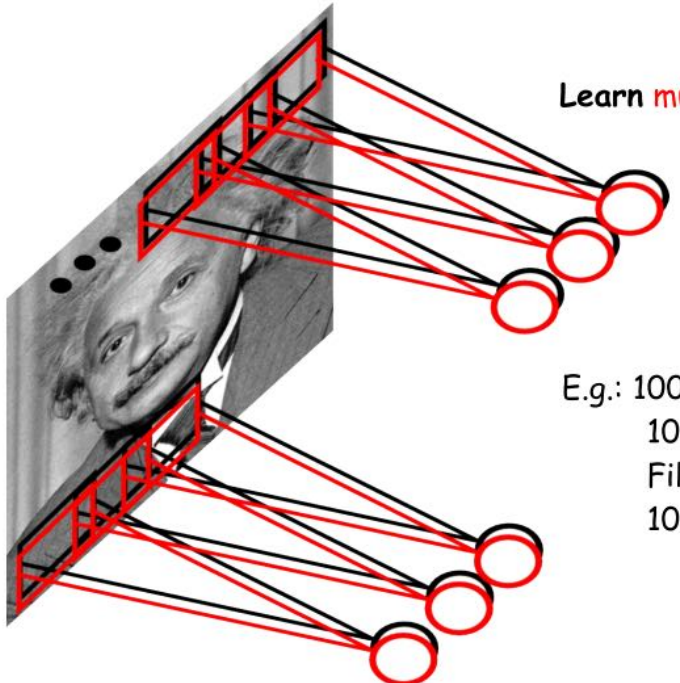


From one to many filters

1 filter => 1 feature map (corresponding to 1 visual pattern)

To detect spatial distributions of multiple visual patterns: Multiple filters

M filters => M feature maps! Get richer description



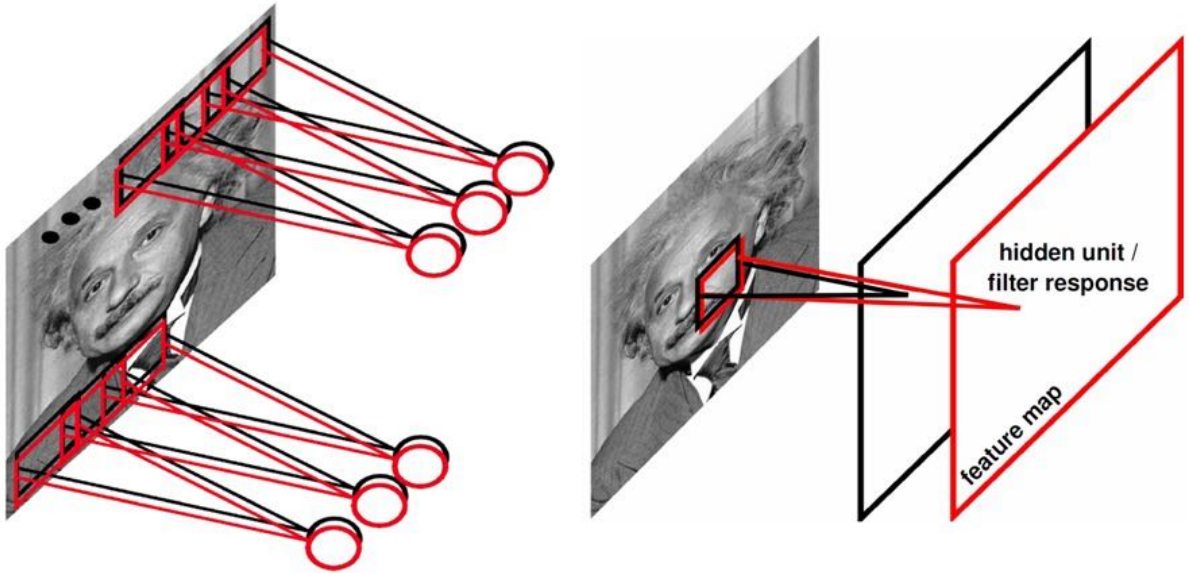
Learn multiple filters.

E.g.: 1000x1000 image
100 Filters
Filter size: 10x10
10K parameters

Not a big deal! Many filters => still few parameters

From one to many filters

M filters => M feature maps

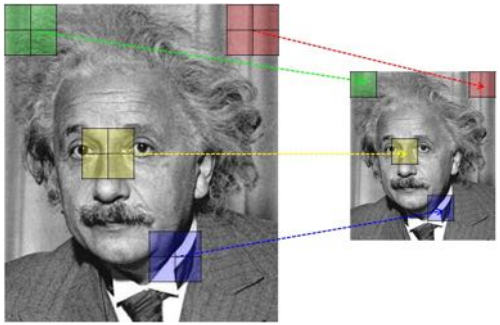


Rq: not many weights but many neurons! => memory issues will appear

Getting (more) local Invariance

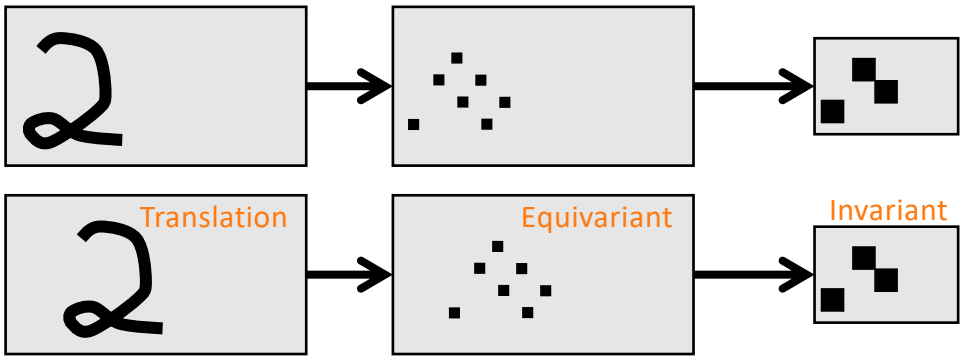
POOLING -- Sub Sampling:

- Averaging neighboring replicated detectors to give a single output to the next level
- Max pooling: Taking the maximum in a neighboring

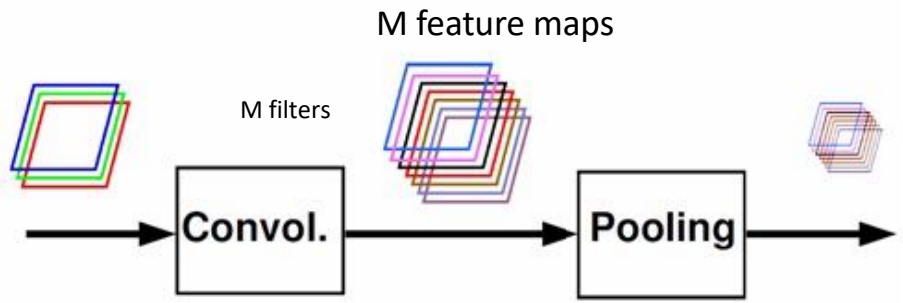


$$y_{ij} = \frac{1}{4}\left(x_{2i,2j} + x_{2i+1,2j} + x_{2i,2j+1} + x_{2i+1,2j+1}\right)$$

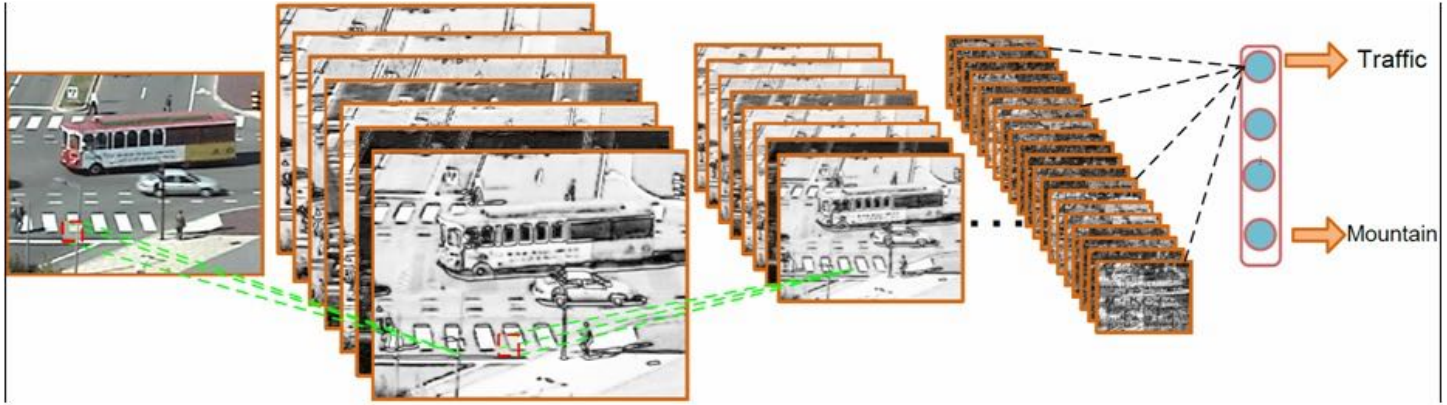
Get a small amount of translational invariance at each level

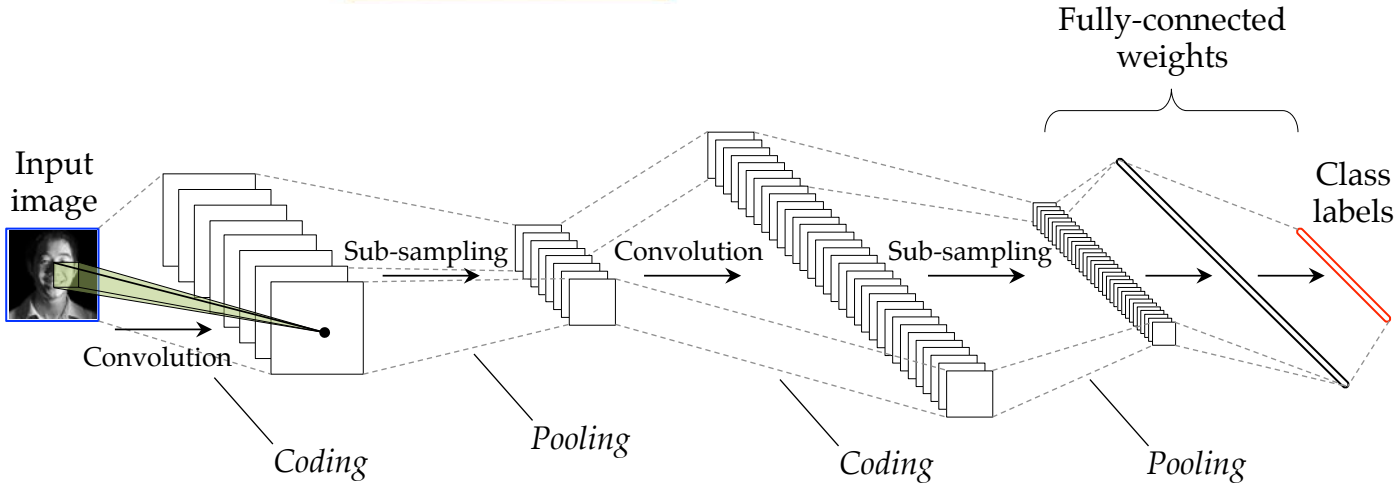


To sum up:



To sum up: Full ConvNet architecture

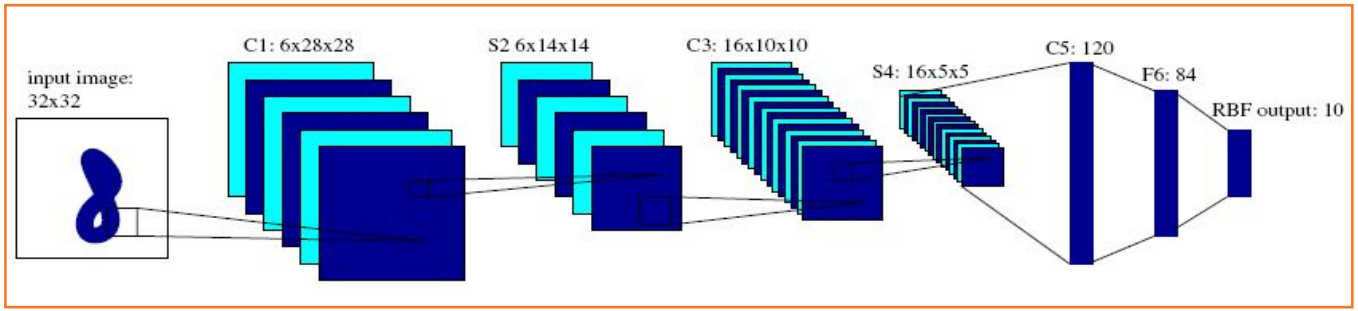




Example: LeNet5

Introduced by Y. LeCun

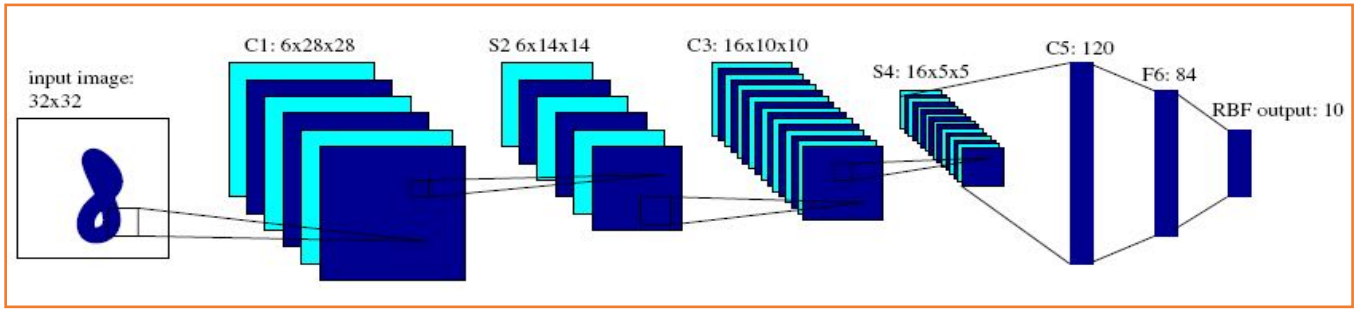
Raw image of 32 × 32 pixels as input



Example: LeNet5

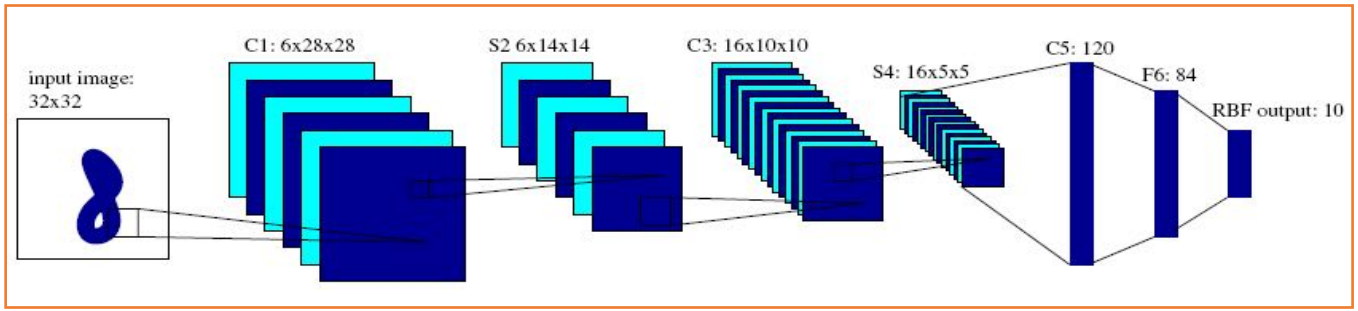
- C1,C3,C5 : Convolutional layer
- 5 × 5 Convolution matrix
- S2 , S4 : Subsampling layer = Pooling+stride s=2

  => Subsampling by factor 2
- F6 : Fully connected layer

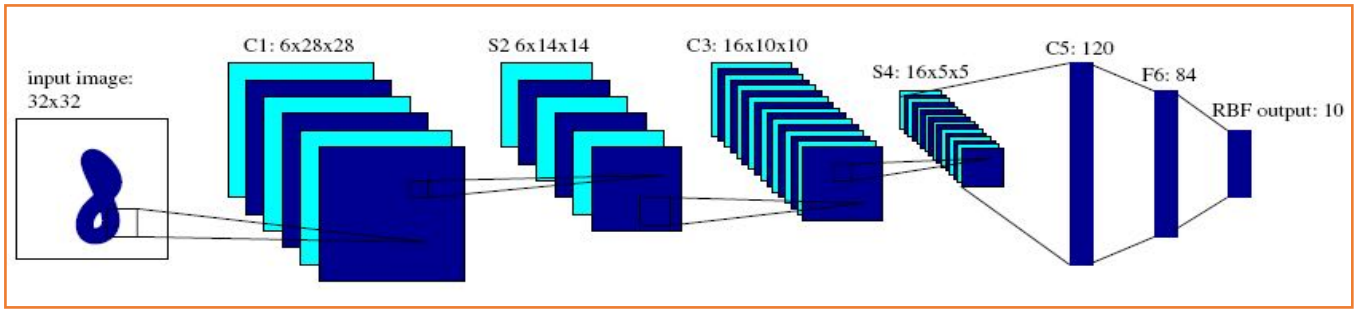


LeNet5

All the units of the layers up to F6 have a sigmoidal activation function

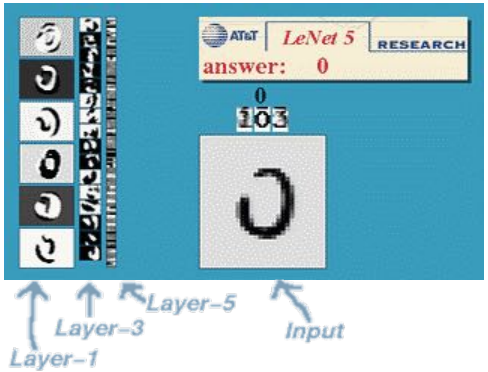


LeNet5

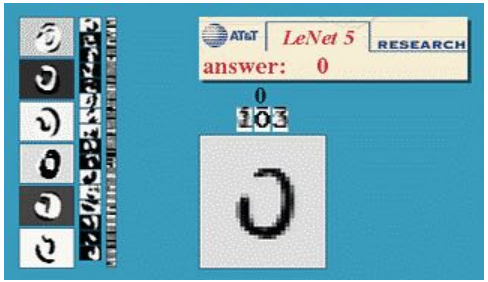


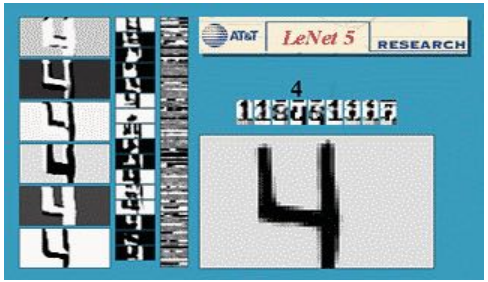
About 187,000 connections

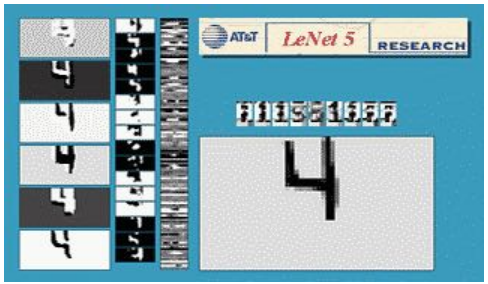
About 14,000 trainable weights

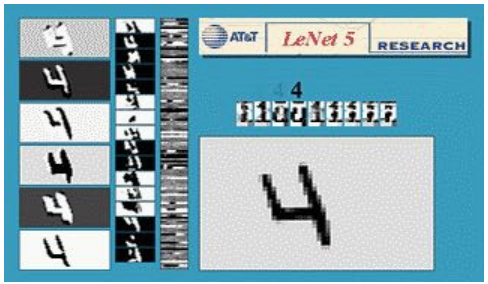


LeNet5 (@LeCun)

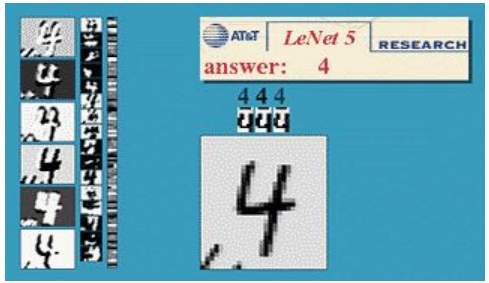


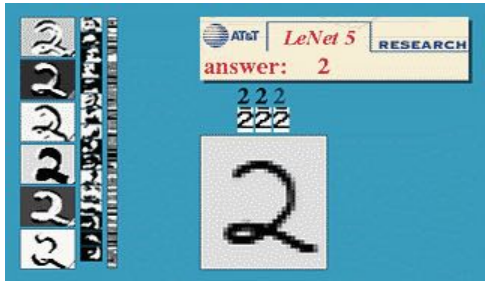


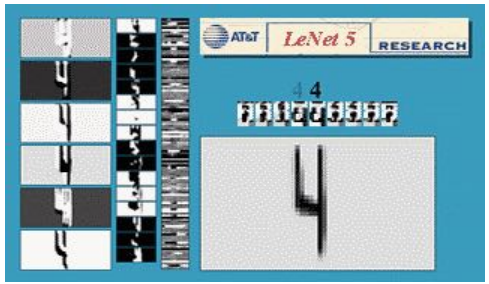


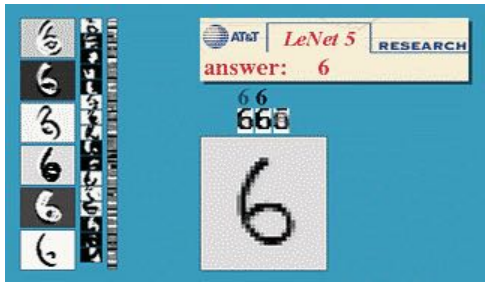


LeNet5 (@LeCun)





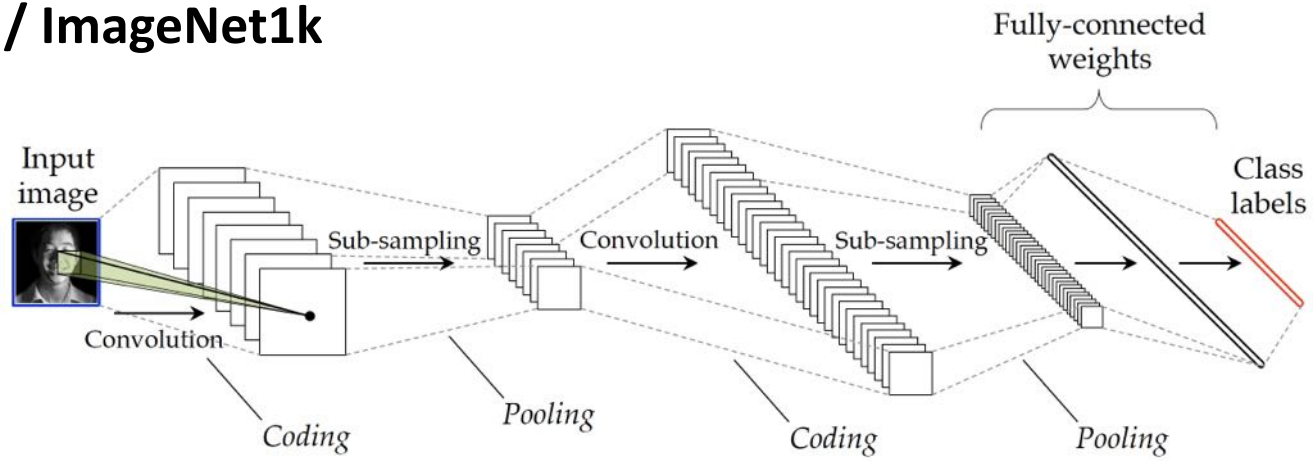




Context: Image classification **After** ImageNet (2009)

The 2010s: *Large* deep neural nets for Visual Classification

The star: **ConvNet / ImageNet1k**



AlexNet 2012

- Same model as LeCun'98 but:
  - Bigger model (8 layers)
  - More data (10^6 vs 10^3 images)
  - GPU implementation (50x speedup over CPU)
  - Better regularization (DropOut)

Post-2012 revolution

VGG Net

K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

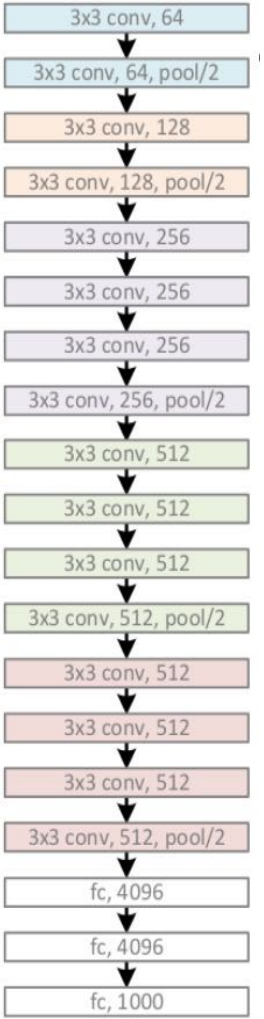
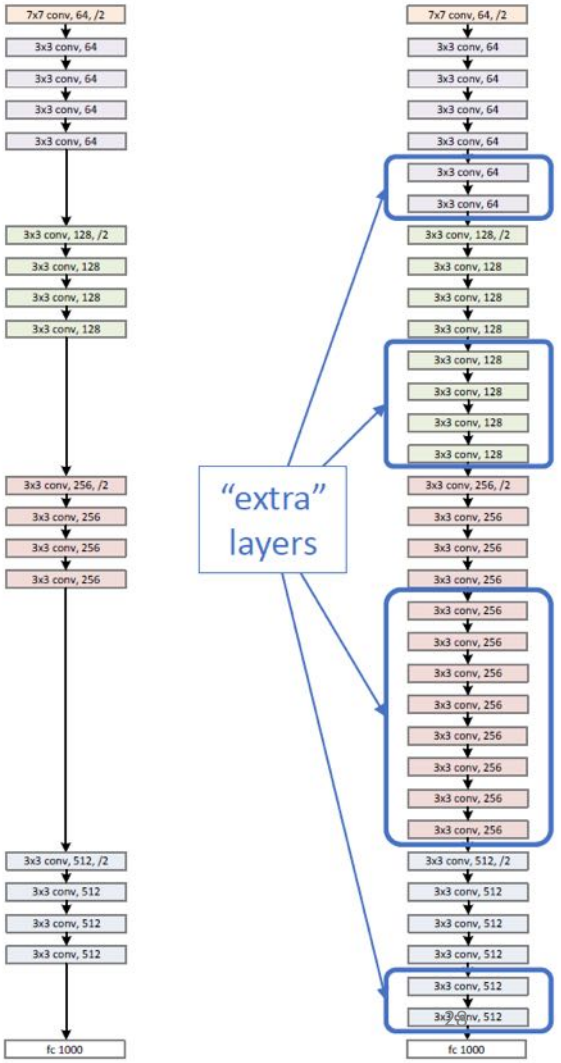
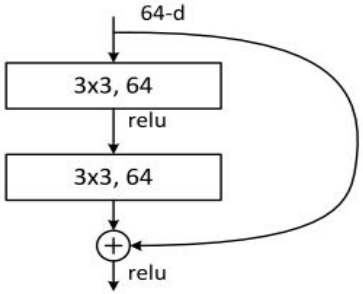


TABLE CREDIT:VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION, ICLR201

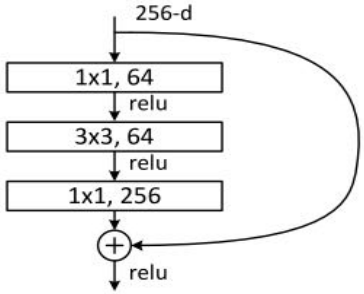
Deeper Network?



ResNet Architecture

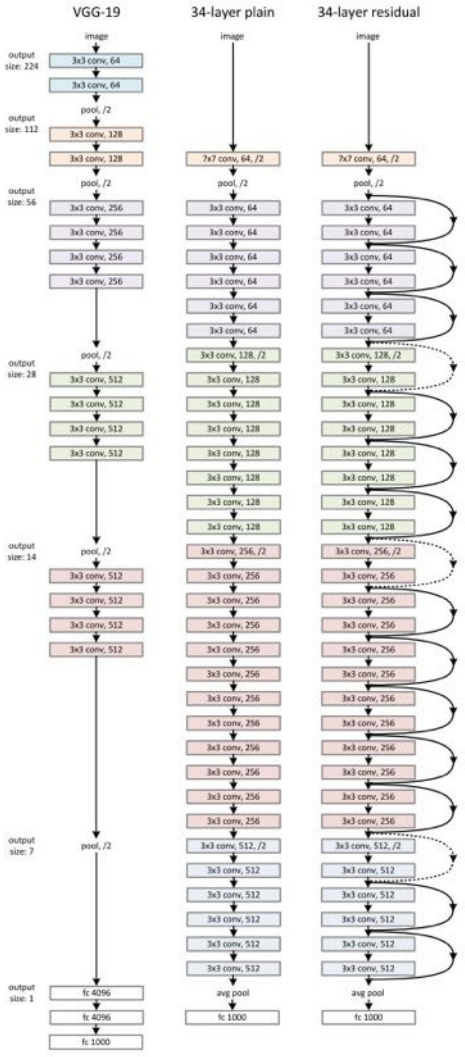


A naïve residual block



"bottleneck" residual block
(for ResNet-50/101/152)

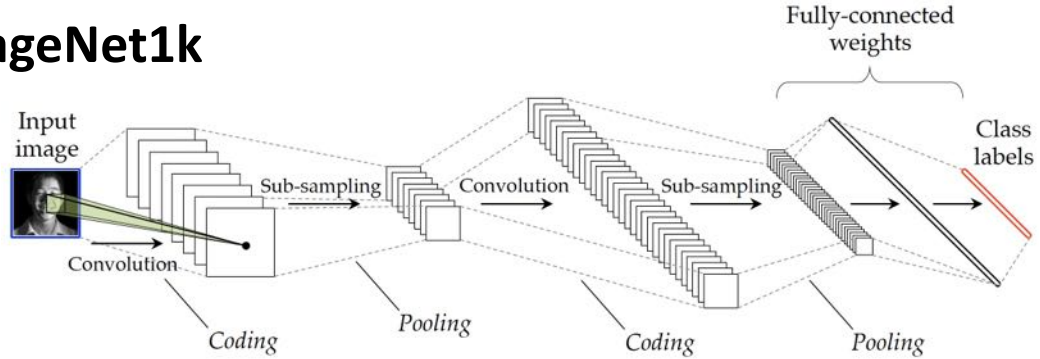
ConvNet Configuration			
B	C	D	E
13 weight layers	16 weight layers	16 weight layers	19 weight layers
ut (224 × 224 RGB image)			
conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool			
conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool			
conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool			
conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool			
conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool			
FC-4096			
FC-4096			
FC-1000			
soft-max			



Context: Beyond ImageNet?

The 2010s: *Large* deep neural nets for Visual Classification

The star: **ConvNet / ImageNet1k**



What is expected for the 2020s?

"Attention is all you need": **Transformers** for Vision !?

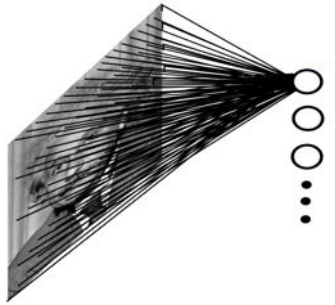
And **datasets? Internet…**

[Vaswani et al., Attention is all you need, NeurIPS 2017]

Outline

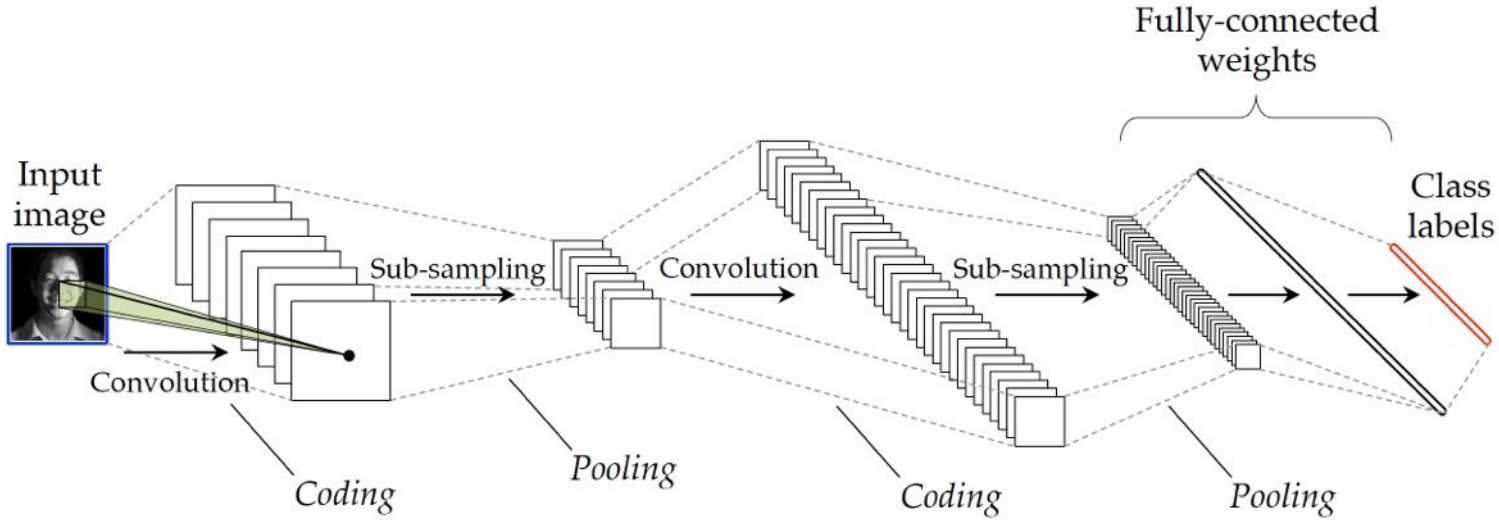
1. Attention and Vision Transformers
   - NLP: Attention is all you need

Attention process in ConvNets



In ConvNets, what information is shared between pixels (or features) in one block? => *2D spatial locality (typically 3x3) => attention is done locally*

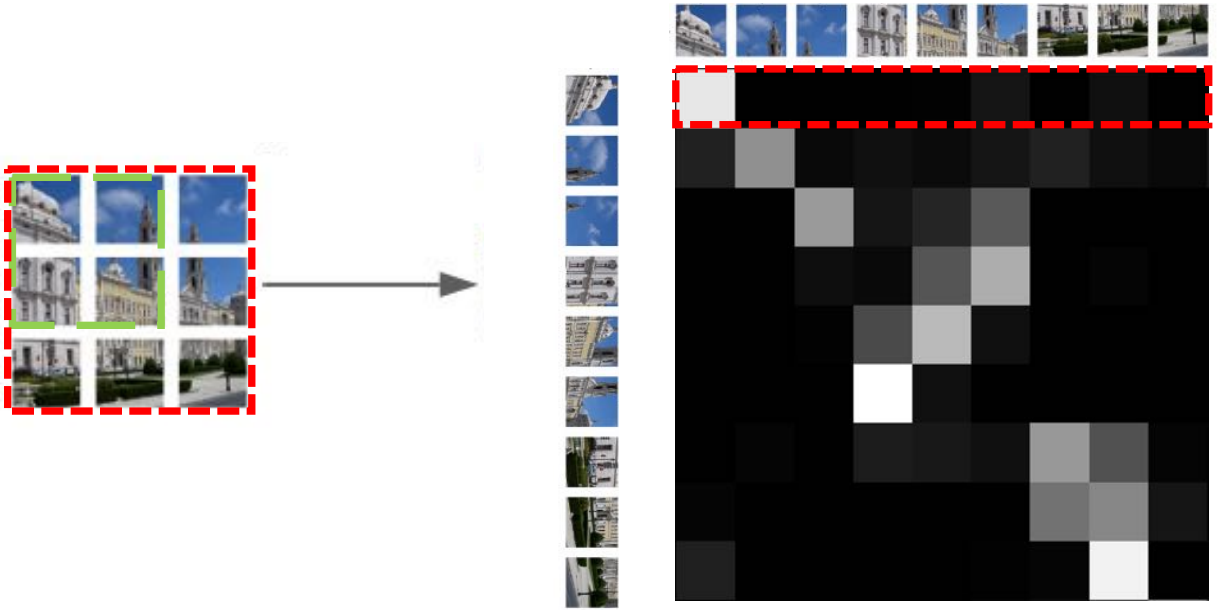
Rq: less local after many layers



Global (Self) attention

How to build a deep architecture with ~~local~~ global attention inside?
Meaning that one patch may interact with all others!

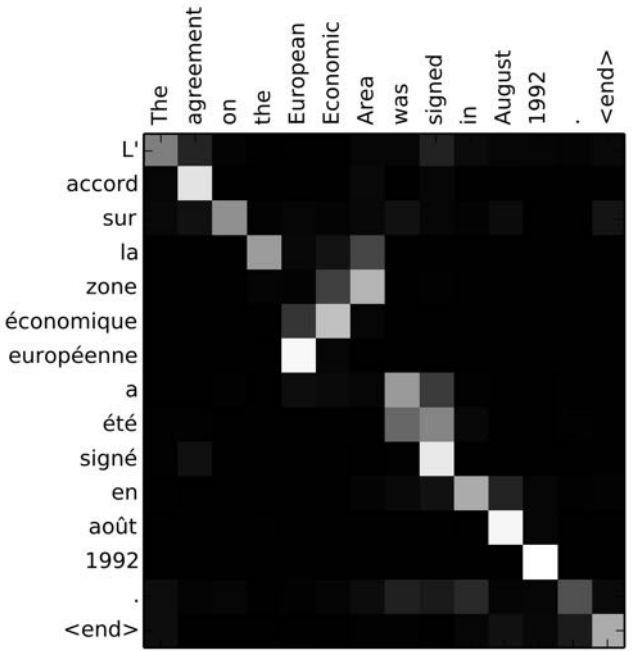
=> Different than convNet!



Let’s see what they do in Natural Language Processing (NLP):

Attention between words in Machine translation process:

1. Computing of weights
2. Use them to compute new features

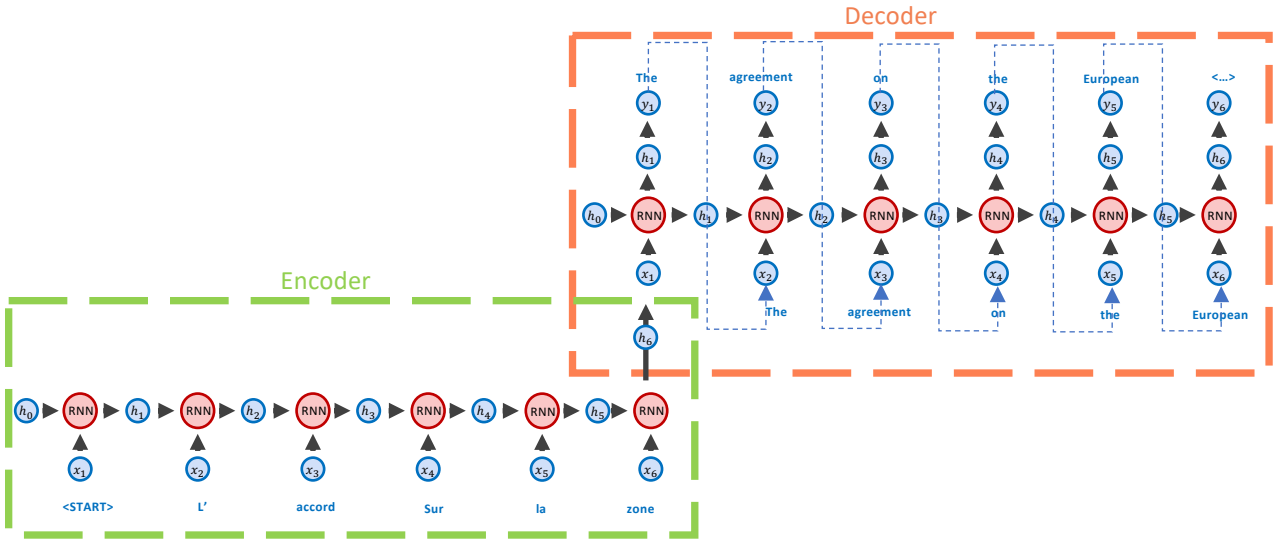


Attention process in NLP

Basic language translation models: Encoder/Decoder

Ex.: Seq2Seq -- RNNs2RNNs

Cross-attention for language translation in at the end of Encoder

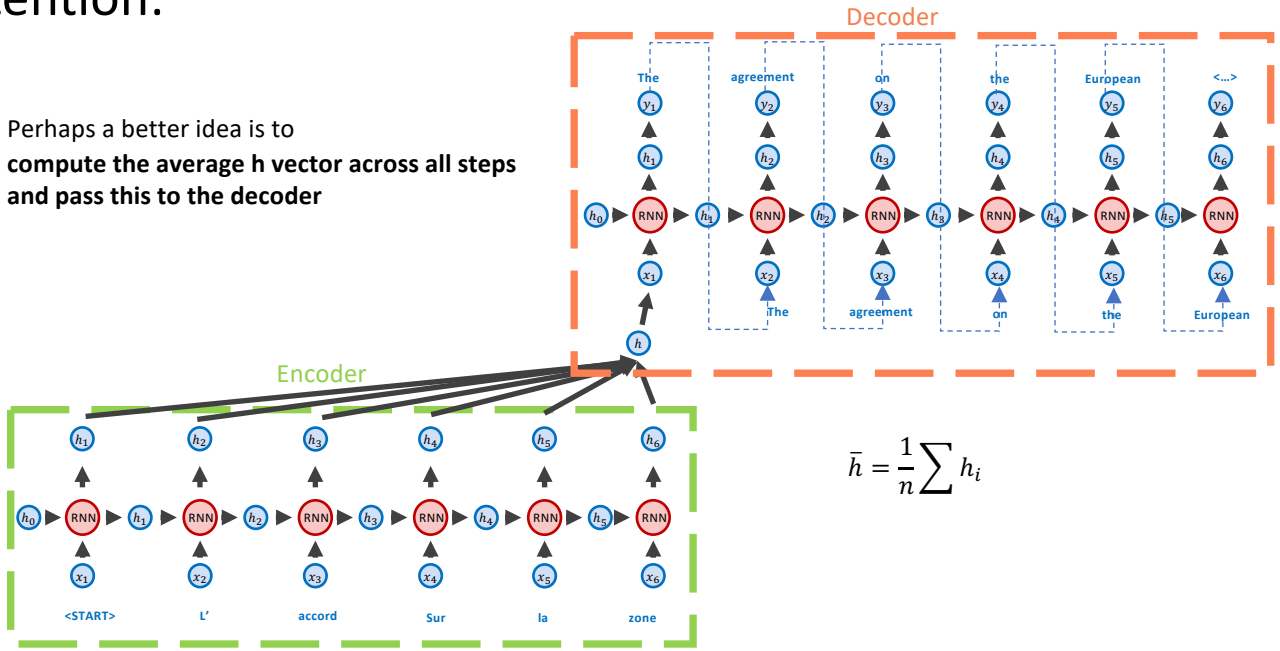


Attention process in NLP

Basic language translation models: Encoder/Decoder

Cross-attention:

Perhaps a better idea is to
compute the average h vector across all steps and pass this to the decoder



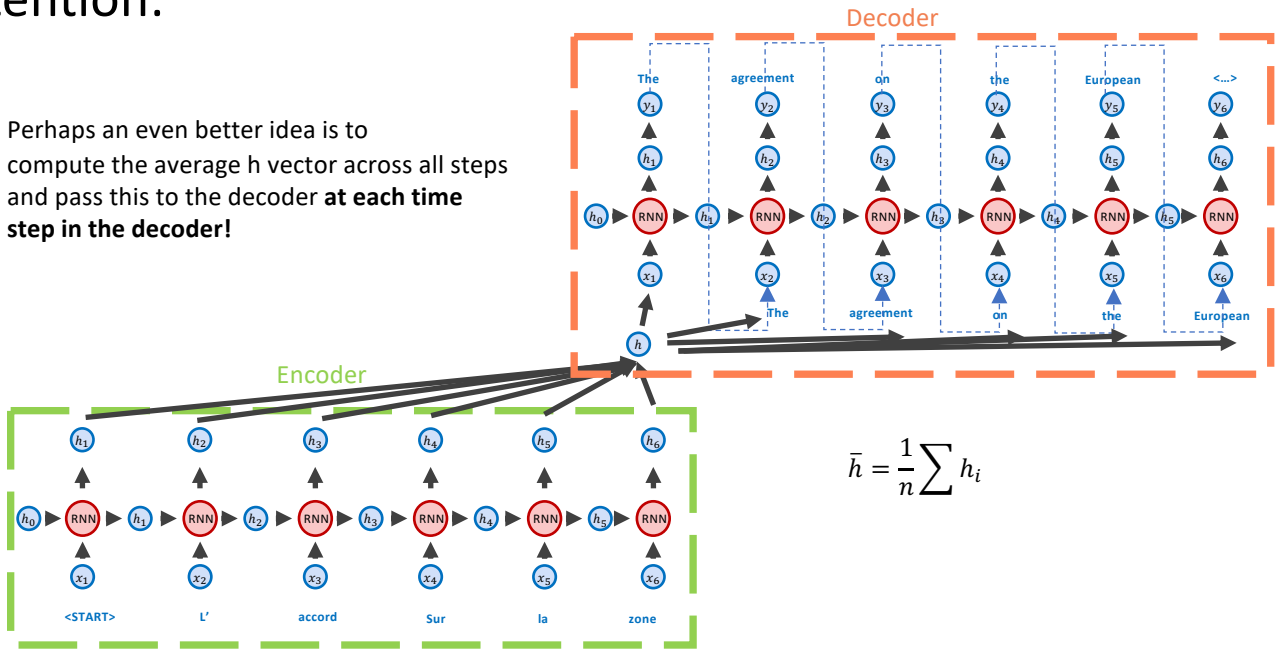
$$\bar{h} = \frac{1}{n}\sum h_i$$

Attention process in NLP

Basic language translation models: Encoder/Decoder

Cross-attention:

Perhaps an even better idea is to compute the average h vector across all steps and pass this to the decoder **at each time step in the decoder!**



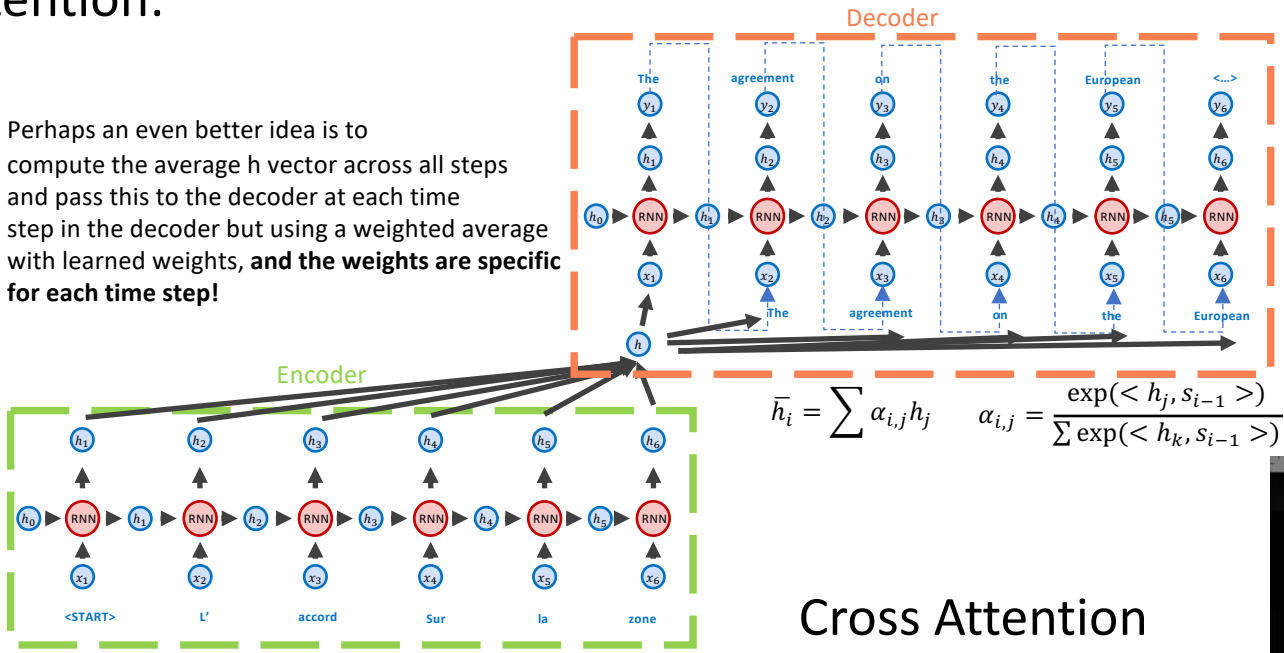
$$\bar{h} = \frac{1}{n}\sum h_i$$

Attention process in NLP

Basic language translation models: Encoder/Decoder

Cross-attention:

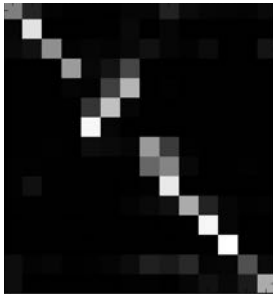
Perhaps an even better idea is to compute the average h vector across all steps and pass this to the decoder at each time step in the decoder but using a weighted average with learned weights, **and the weights are specific for each time step!**



$$\bar{h}_i = \sum \alpha_{i,j} h_j \qquad \alpha_{i,j} = \frac{\exp(< h_j, s_{i-1} >)}{\sum \exp(< h_k, s_{i-1} >)}$$

Cross Attention

Encoder/ Decoder

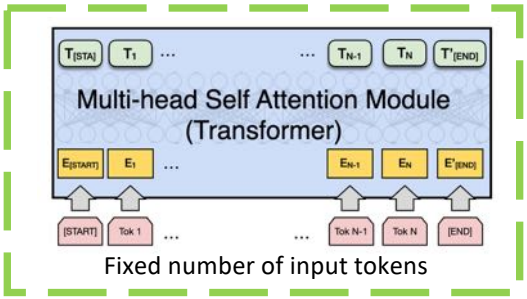


Attention process in NLP

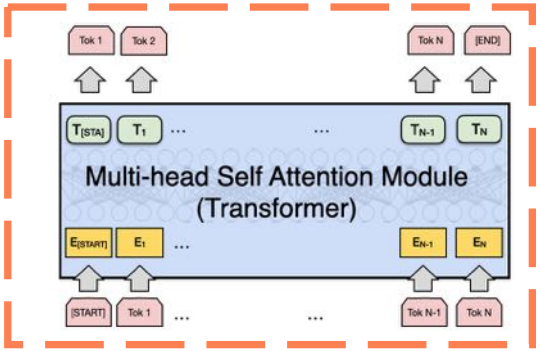
Basic language translation models: Encoder/Decoder

Transformer architecture (no RNNs)

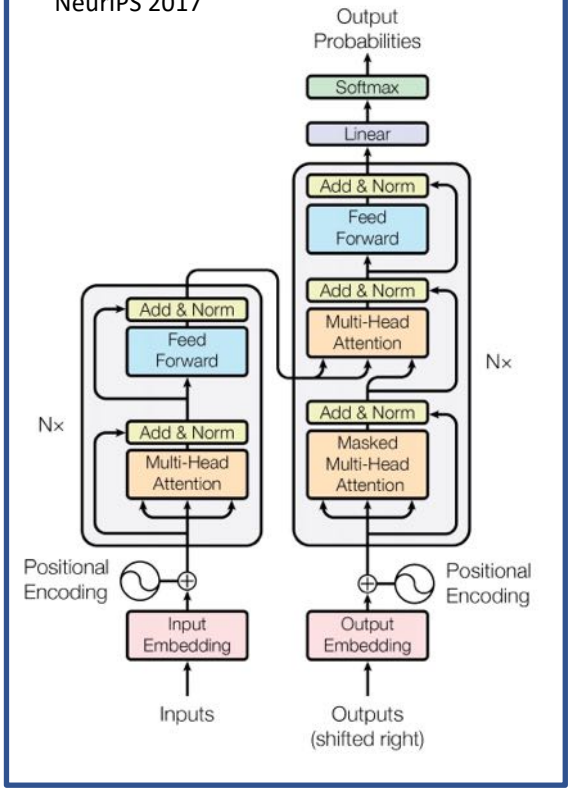
Encoder



Decoder



[Vaswani et al. Attention is all you need]
https://arxiv.org/abs/1706.03762
NeurIPS 2017

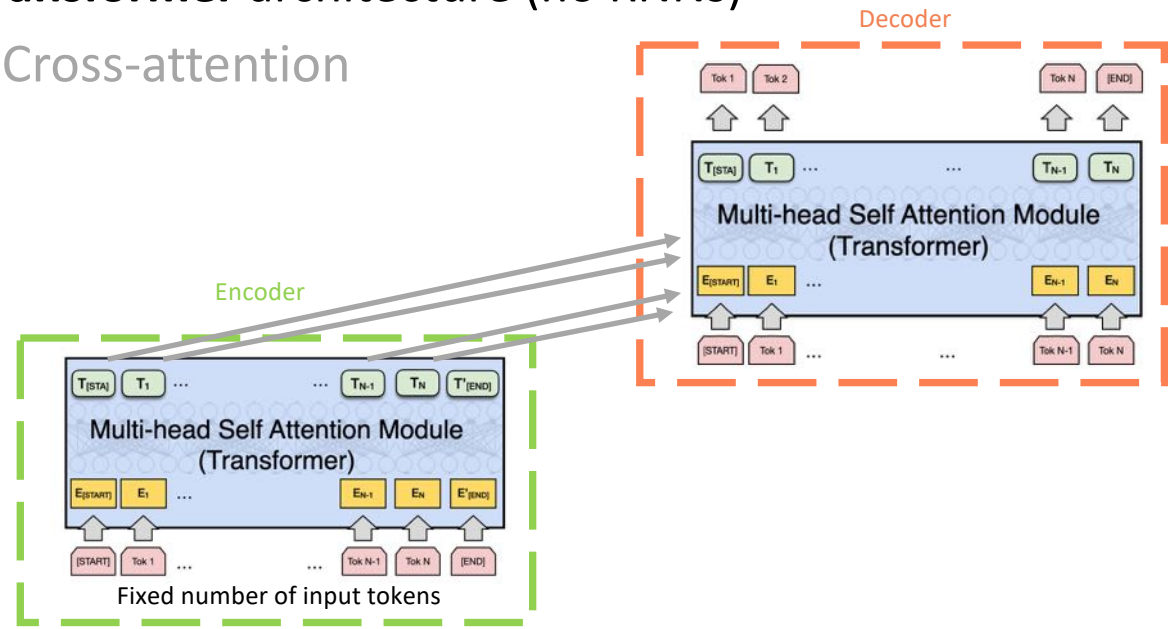


Attention process in NLP

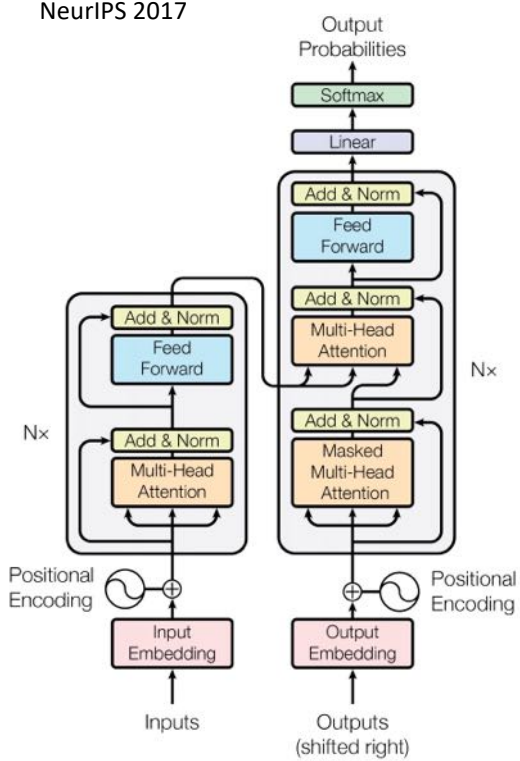
Basic language translation models: Encoder/Decoder

Transformer architecture (no RNNs)

- Cross-attention



[Vaswani et al. Attention is all you need]
https://arxiv.org/abs/1706.03762
NeurIPS 2017

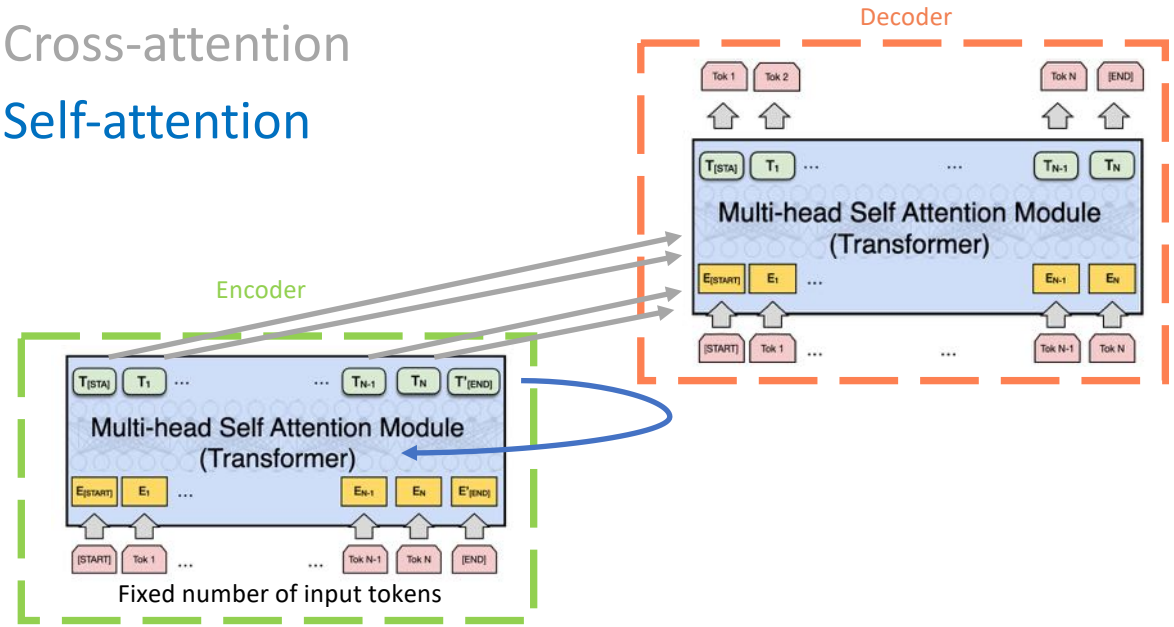


Attention process in NLP

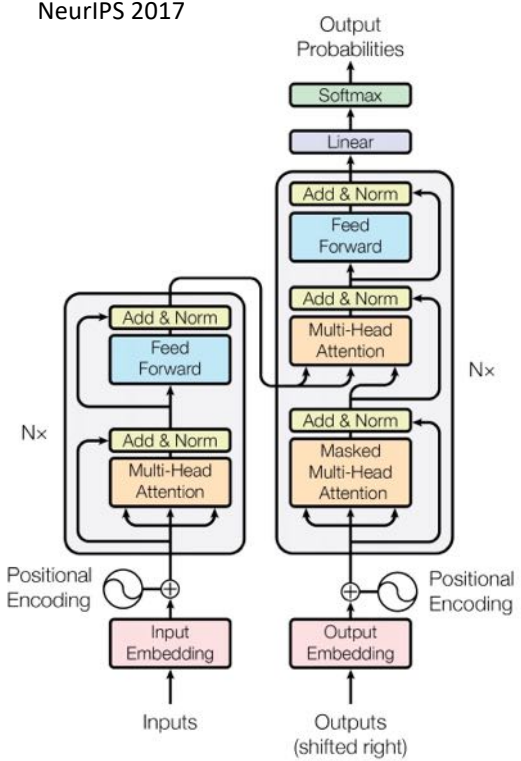
Basic language translation models: Encoder/Decoder

Transformer architecture (no RNNs)

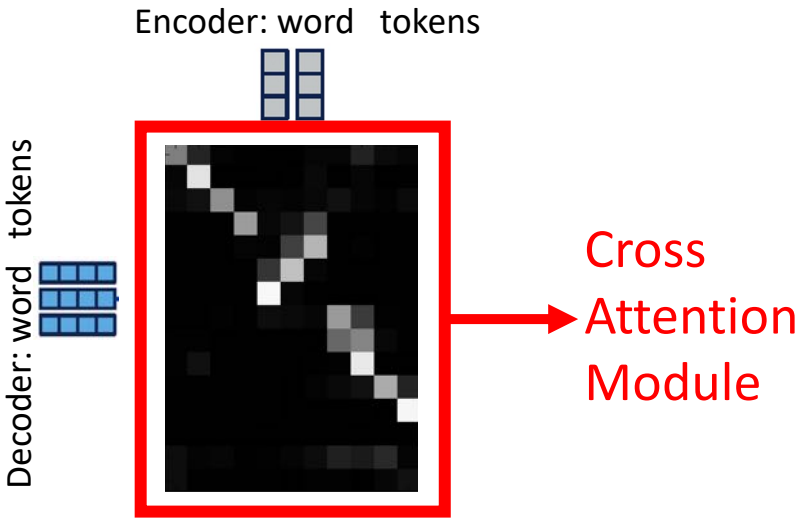
- Cross-attention
- Self-attention



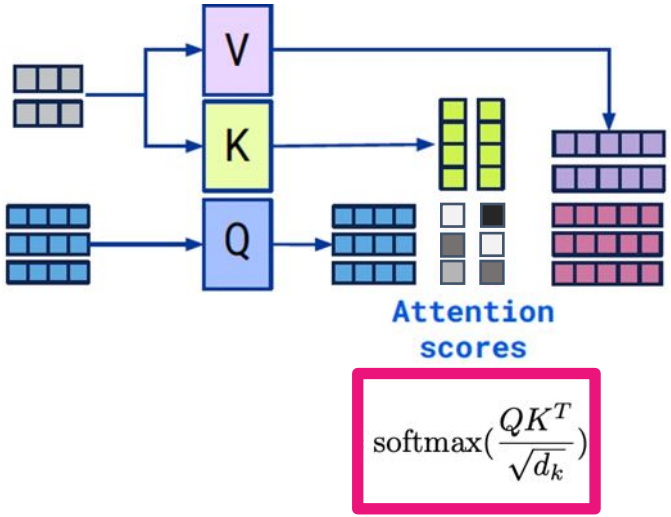
[Vaswani et al. Attention is all you need]
https://arxiv.org/abs/1706.03762
NeurIPS 2017



Attention process in NLP



Cross Attention Module

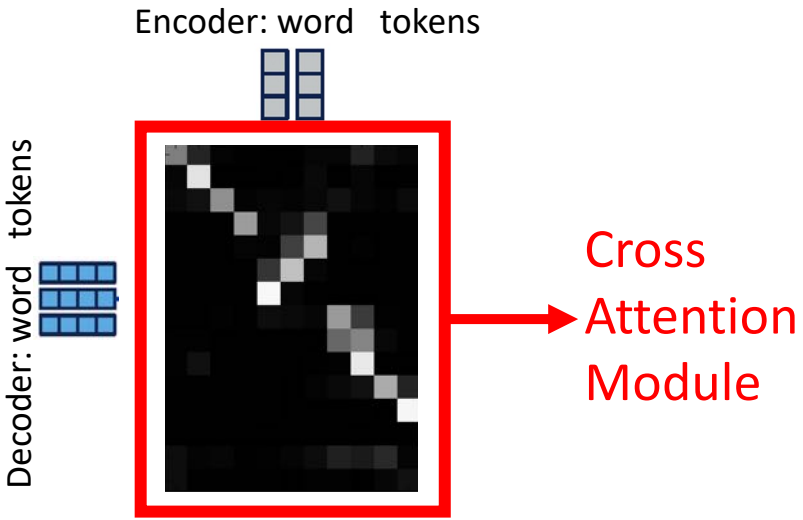


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

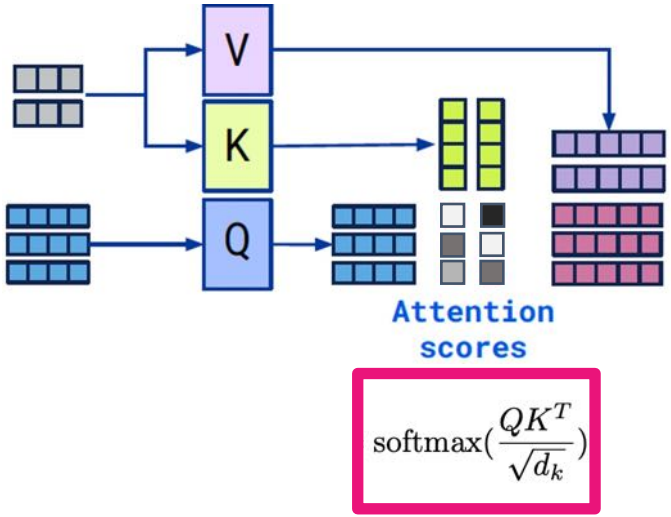
Outline

1. Attention and Vision Transformers (ViT)
   - NLP: Attention is all you need
   - **Transformer for image classification**

Attention process in NLP

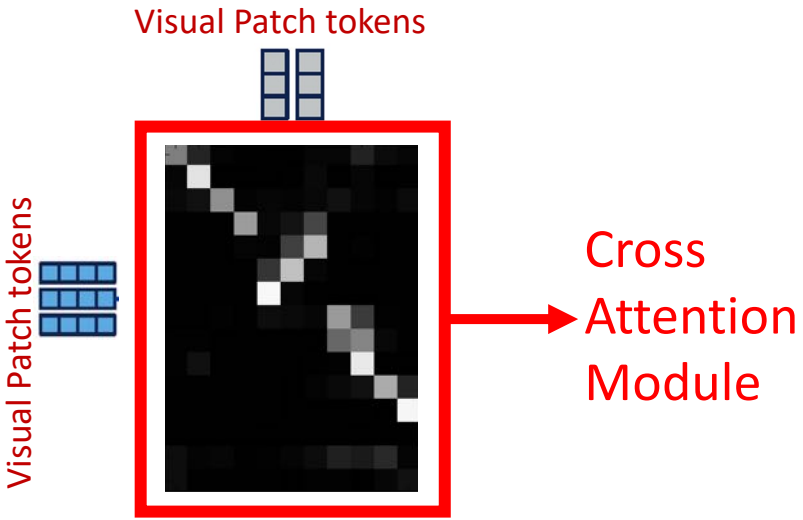


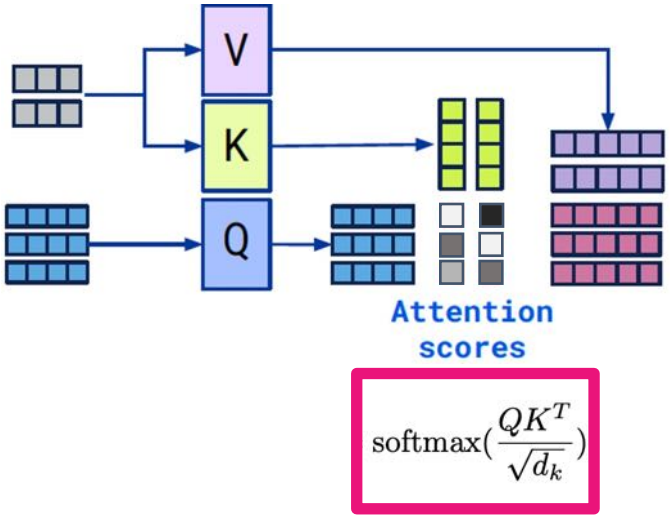
Cross Attention Module



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

Attention process in Vision





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

Very similar except that Visual token is definitively less natural than word for NLP

Attention process in Vision

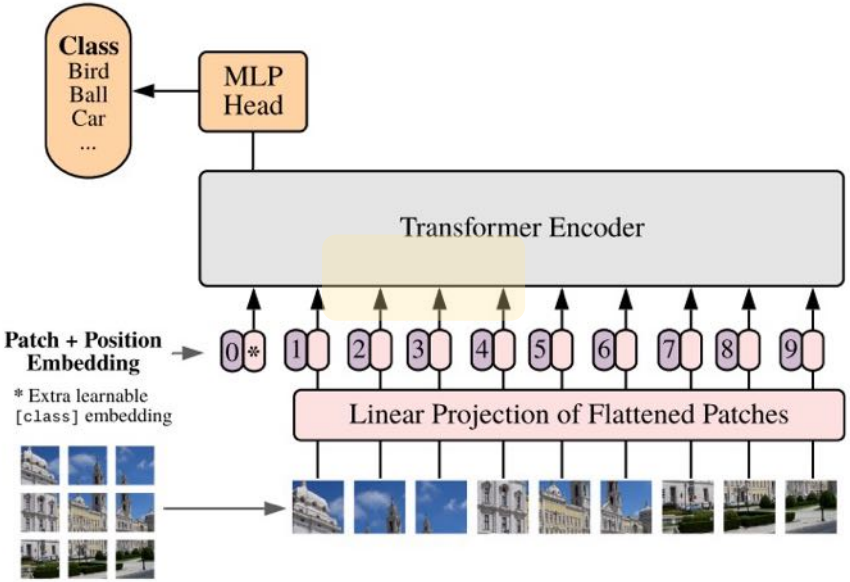
Is it possible to mimic this attention-based architecture for vision processing?

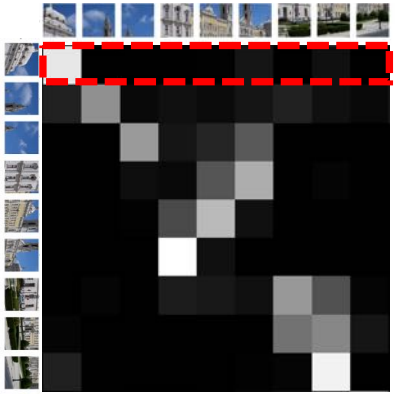
Yes! **ViT** (Vision image Transformers) architecture

Published as a conference paper at ICLR 2021

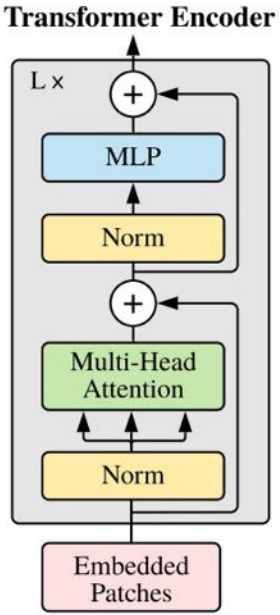
AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

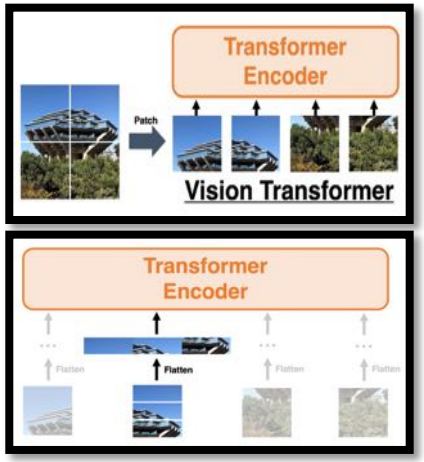
Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com





Attention process in Vision





$$x \in \mathbb{R}^{H \times W \times C}$$

$$x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$$

$$N = HW/P^2$$

CLS token

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D},\ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1 \ldots L$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0)$$

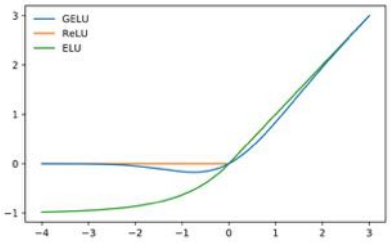
[class=CLS] token: a learnable embedding to the sequence of embedded patches

Layernorm (LN) before every block, and residual connections after every block

MSA: Multi Head Self Attention

MLP: two layers with a GELU non-linearity

Hybrid Architecture : Raw image patches --> Feature map of a CNN

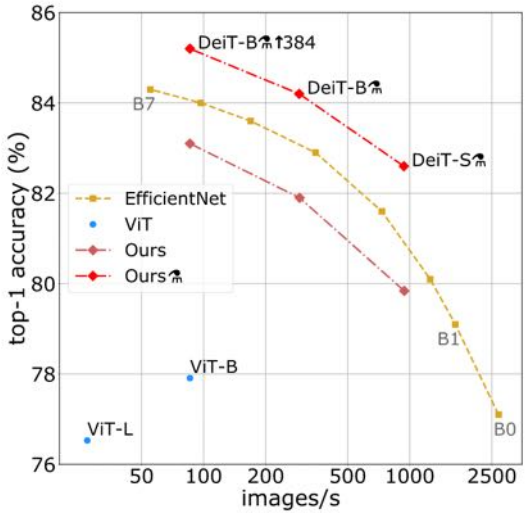


Attention process in Vision

Experiments with ViT (and variants DeiT, CaiT) transformers for image classification

State-of-the-art performance on ImageNet1k classification!

From ViT paper, **many tricks/discussions to simplify learning** in DeiT, CaiT, …



Published as a conference paper at ICML 2021

DeiT

Training data-efficient image transformers & distillation through attention

Hugo Touvron [1 2] Matthieu Cord [1 2] Matthijs Douze [1]
Francisco Massa [1] Alexandre Sablayrolles [1] Hervé Jégou [1]

Attention process in Vision

How to choose the image splitting?

Pb: quadratic complexity with the nb of patches

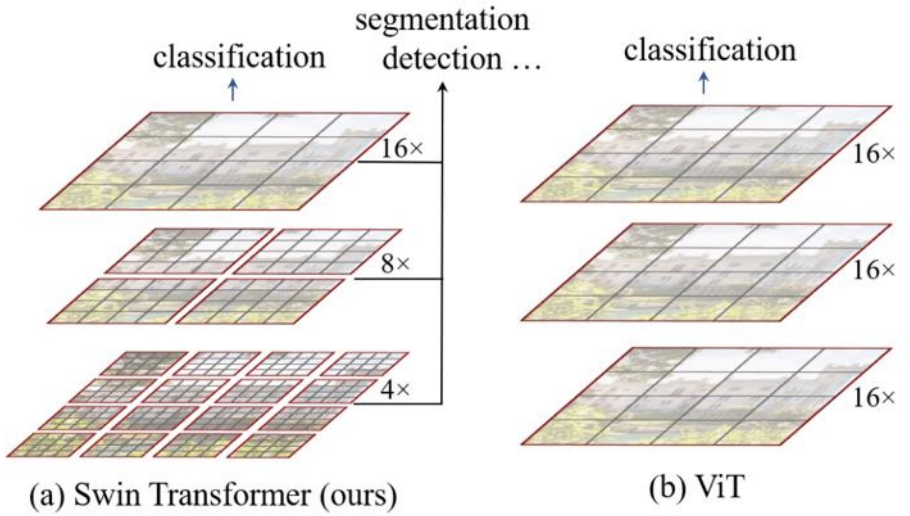
Many kinds of hybrid architectures with convnets/transformers

Ex: Swin Transformers

Published as a conference paper at ICCV 2021

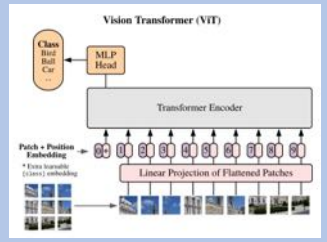
Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Ze Liu[1,2†*] Yutong Lin[1,3†*] Yue Cao[1*] Han Hu[1*‡] Yixuan Wei[1,4†]
Zheng Zhang[1] Stephen Lin[1] Baining Guo[1]
[1]Microsoft Research Asia [2]University of Science and Technology of China
[3]Xian Jiaotong University [4]Tsinghua University
{v-zeliu1,v-yutlin,yuecao,hanhu,v-yixwe,zhez,stevelin,bainguo}@microsoft.com



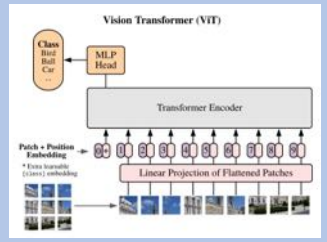
Outline

1. Attention and Vision Transformers (ViT)
   - NLP: Attention is all you need
   - Transformer Encoder ViT with Self Attention for image classification



Outline

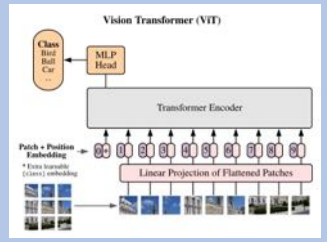
1. Attention and Vision Transformers (ViT)
   - NLP: Attention is all you need
   - Transformer Encoder ViT with Self Attention for image classification



2. **Transformer Decoder** for downstream tasks

Outline

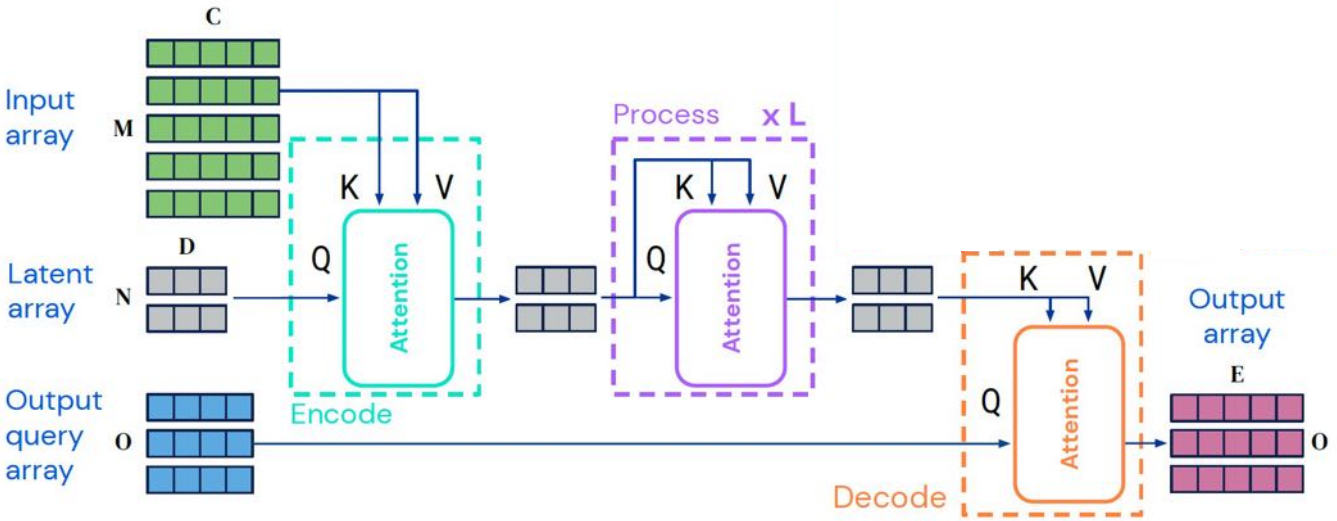
1. Attention and Vision Transformers (ViT)
   - NLP: Attention is all you need
   - Transformer Encoder ViT with Self Attention for image classification



2. **Transformer Decoder** for downstream tasks
   - Detection
   - Segmentation
   - Continual Learning, …

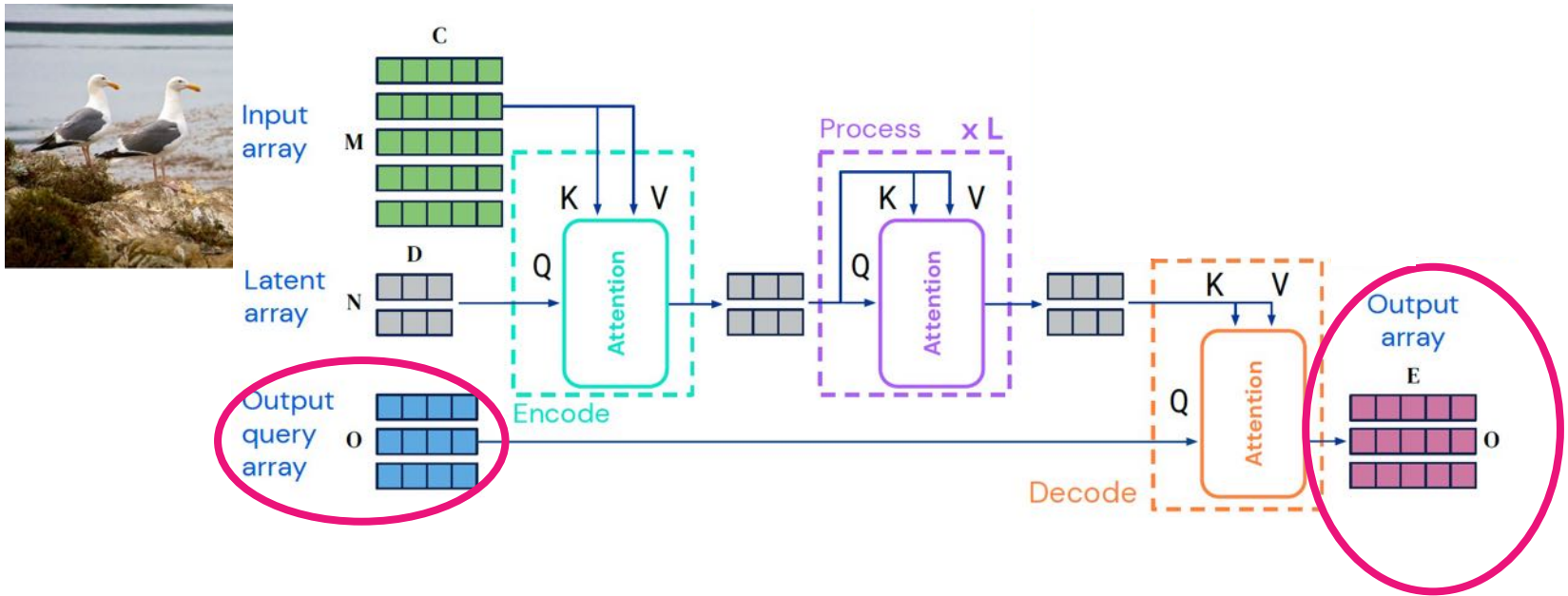
General Decoder

[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]



General Decoder

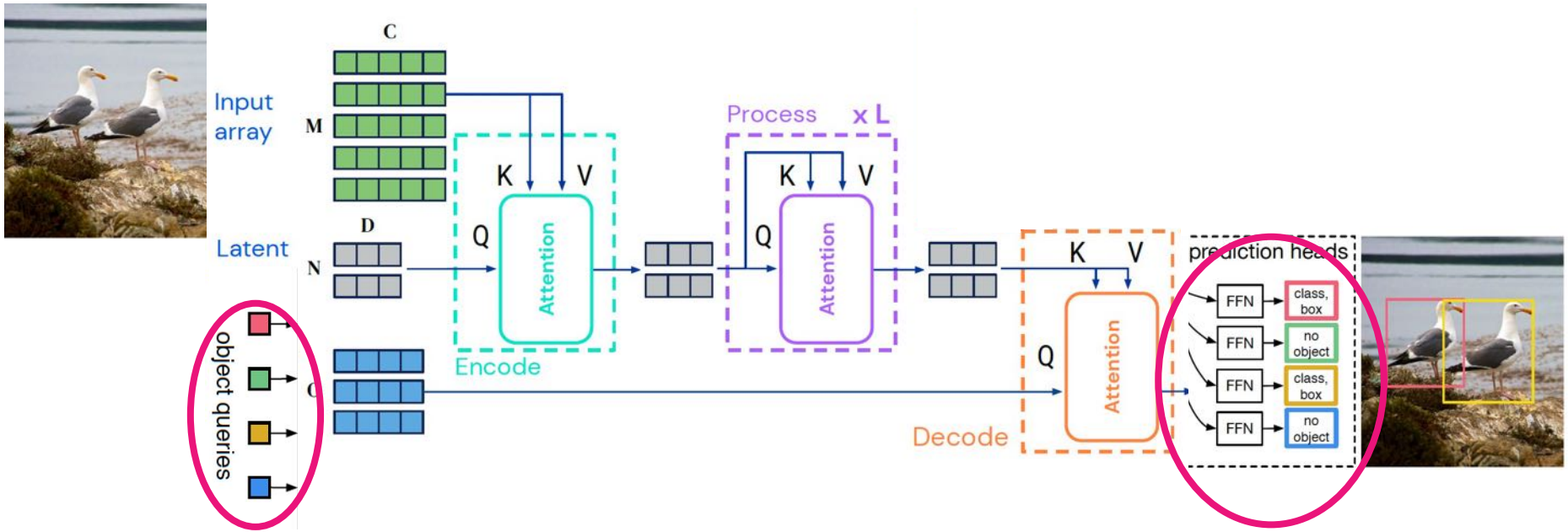
[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]



Output query array / Output array defines the downstream task: detection, segmentation …

General Decoder

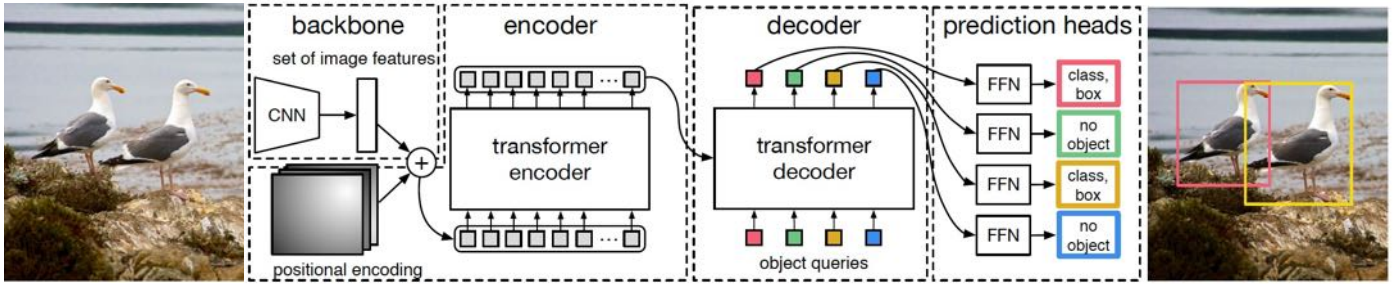
[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]

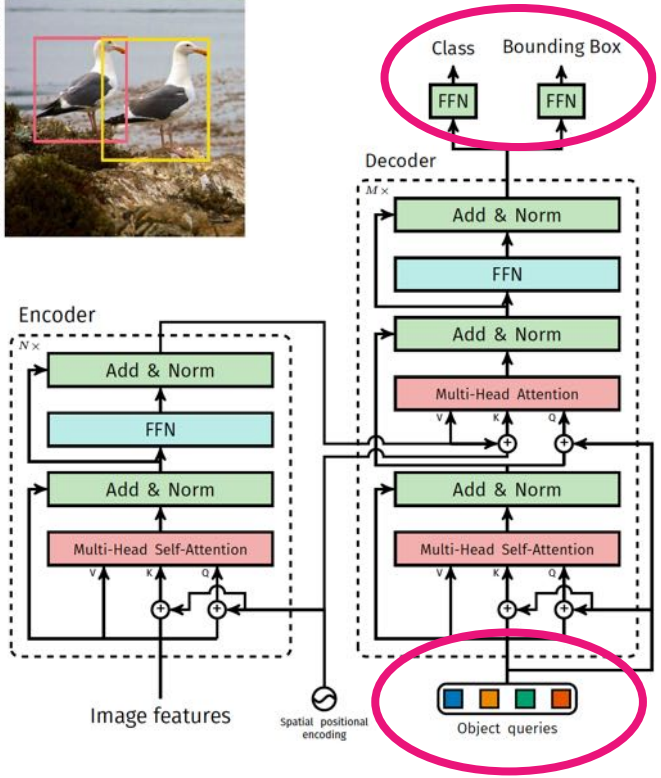


Output query array / Output array defines the downstream task: detection

Transformer Decoder for detection

Just another scheme for DETR model





Cornell University

arXiv > cs > arXiv:2005.12872

Computer Science > Computer Vision and Pattern Recognition

[Submitted on 26 May 2020 (v1), last revised 28 May 2020 (this version, v3)]

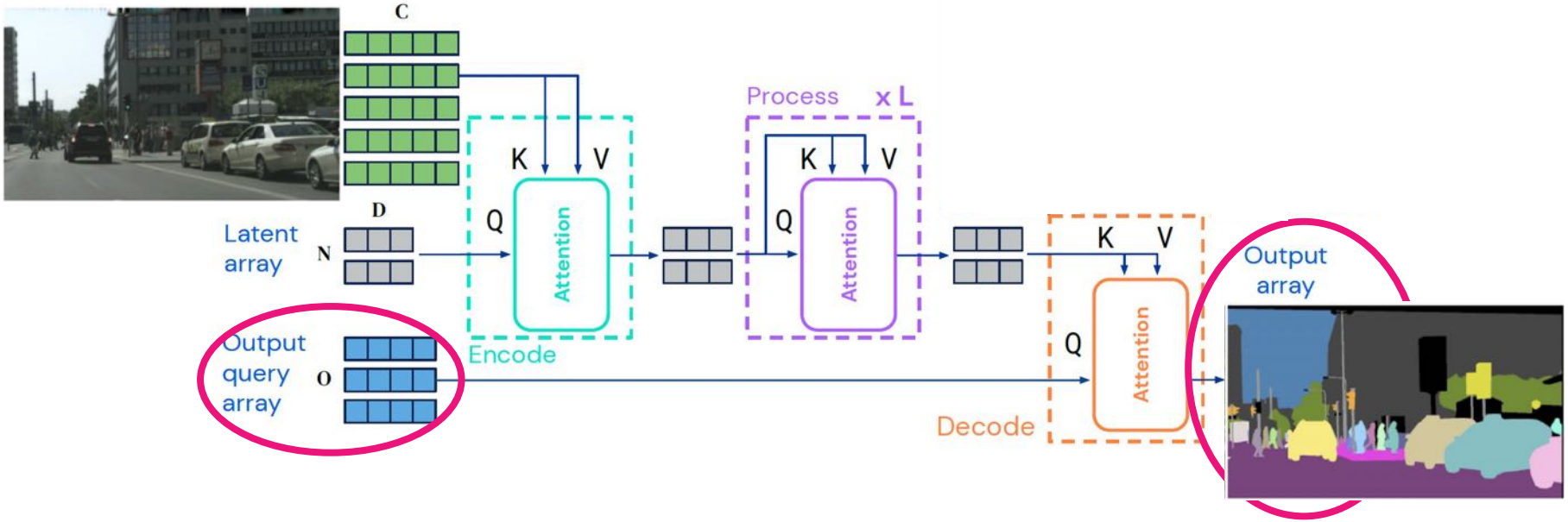
End-to-End Object Detection with Transformers

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko

We present a new method that views object detection as a direct set prediction problem. Our approach streamlines the detection pipe hand-designed components like a non-maximum suppression procedure or anchor generation that explicitly encode our prior knowl the new framework, called DEtection TRansformer or DETR, are a set-based global loss that forces unique predictions via bipartite ma

General Decoder

[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]



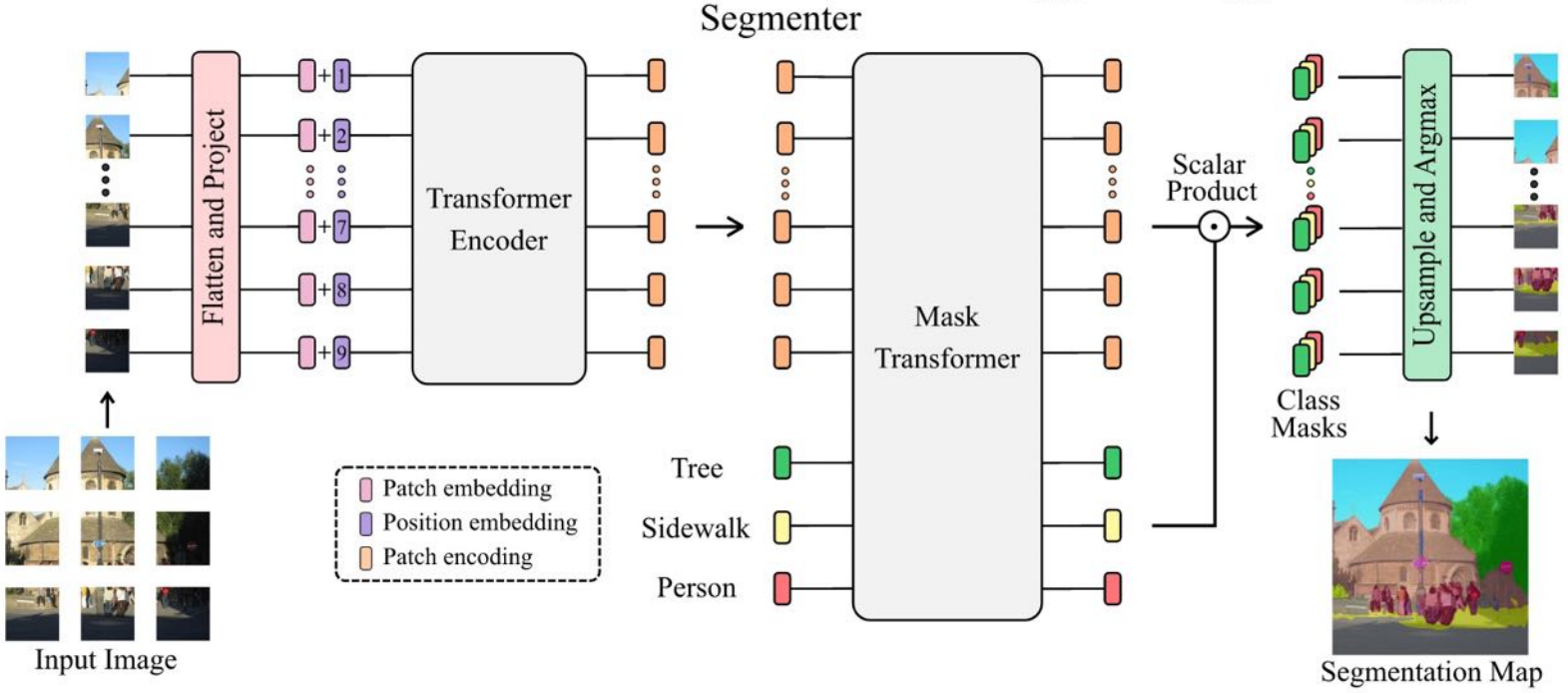
Output query array / Output array defines the downstream task: segmentation …

General Decoder: or not!

This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

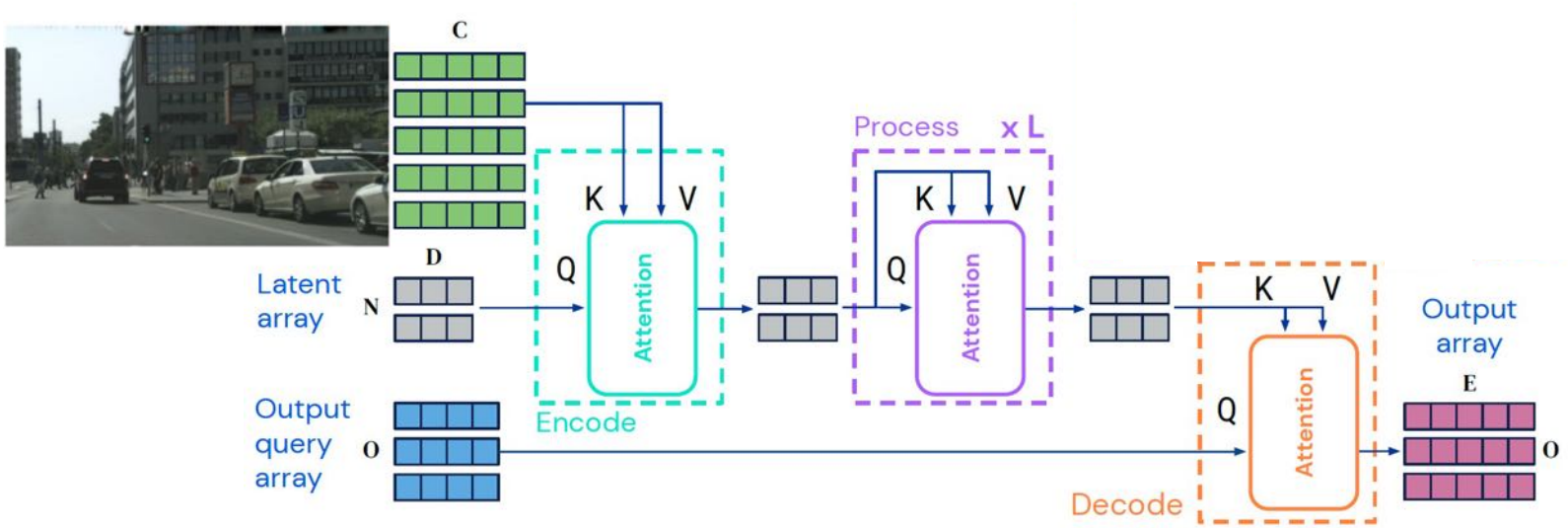
Segmenter: Transformer for Semantic Segmentation

Robin Strudel* Inria† Ricardo Garcia* Inria† Ivan Laptev Inria† Cordelia Schmid Inria†



General Decoder

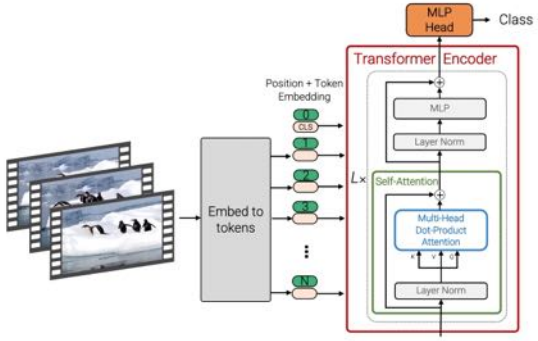
[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]

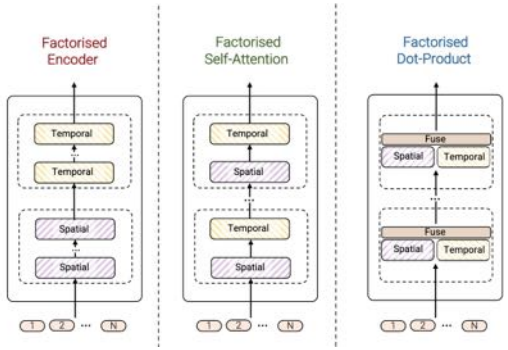


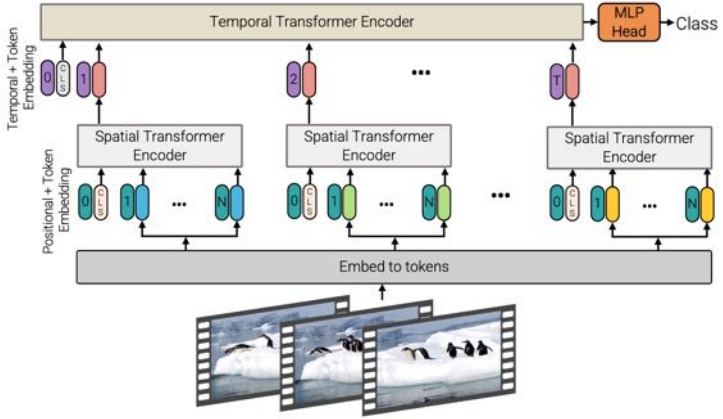
Output query array / Output array defines the downstream task: continual learning

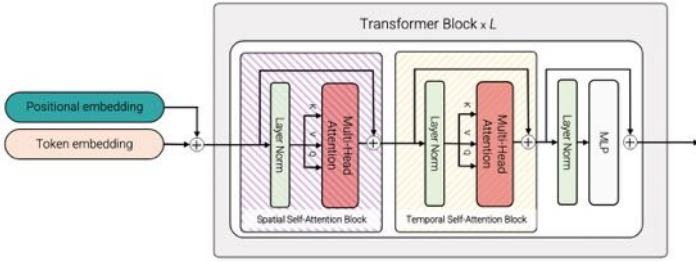
Video Transformer

[ViViT: A Video Vision Transformer ICCV 2021]







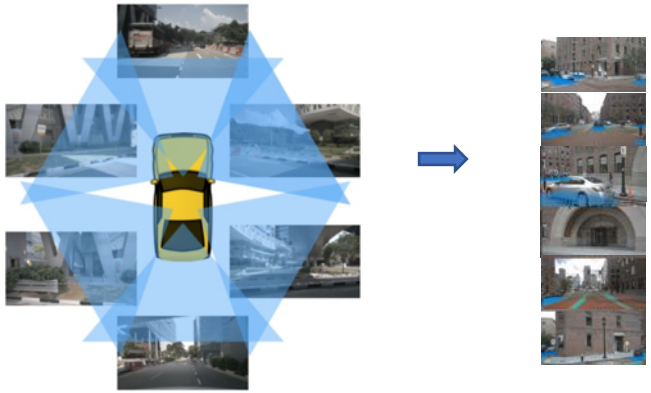


General Encoder / Decoder

Input array = N cameras

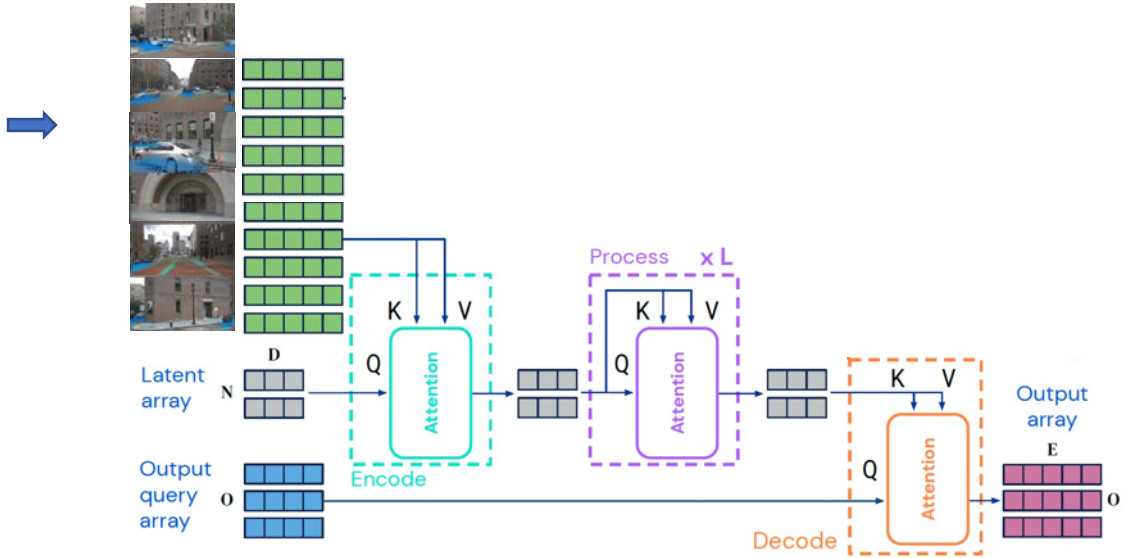
General Encoder / Decoder

Input array = N cameras



General Encoder / Decoder

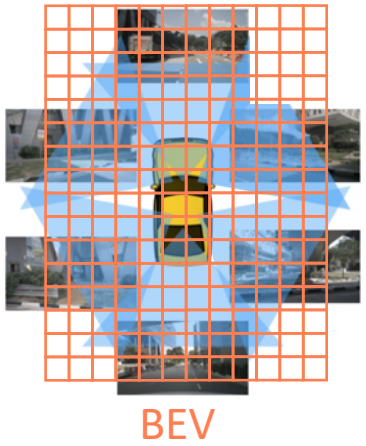
Input array = N cameras

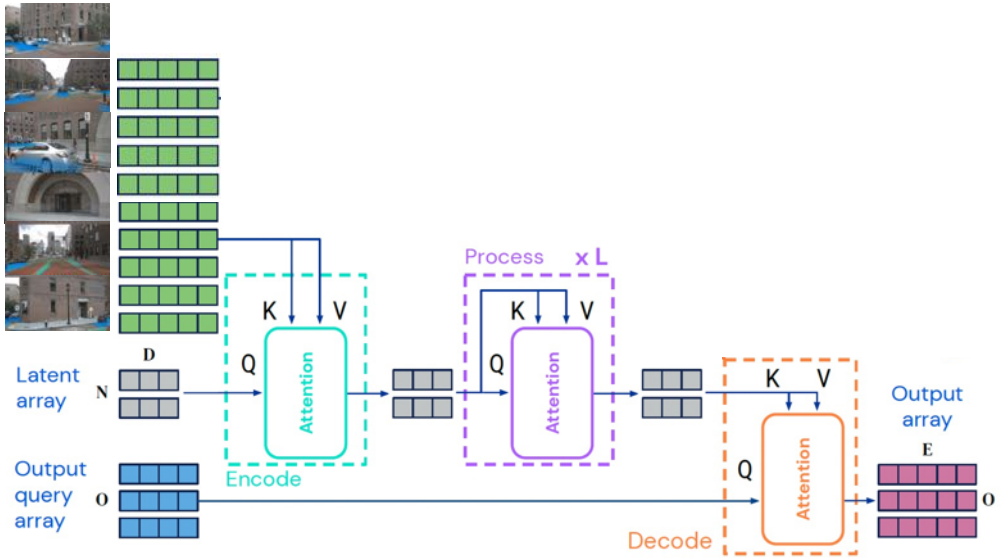


General Encoder / Decoder

Input array = N cameras

Output array = Bird Eye View (BEV) representation

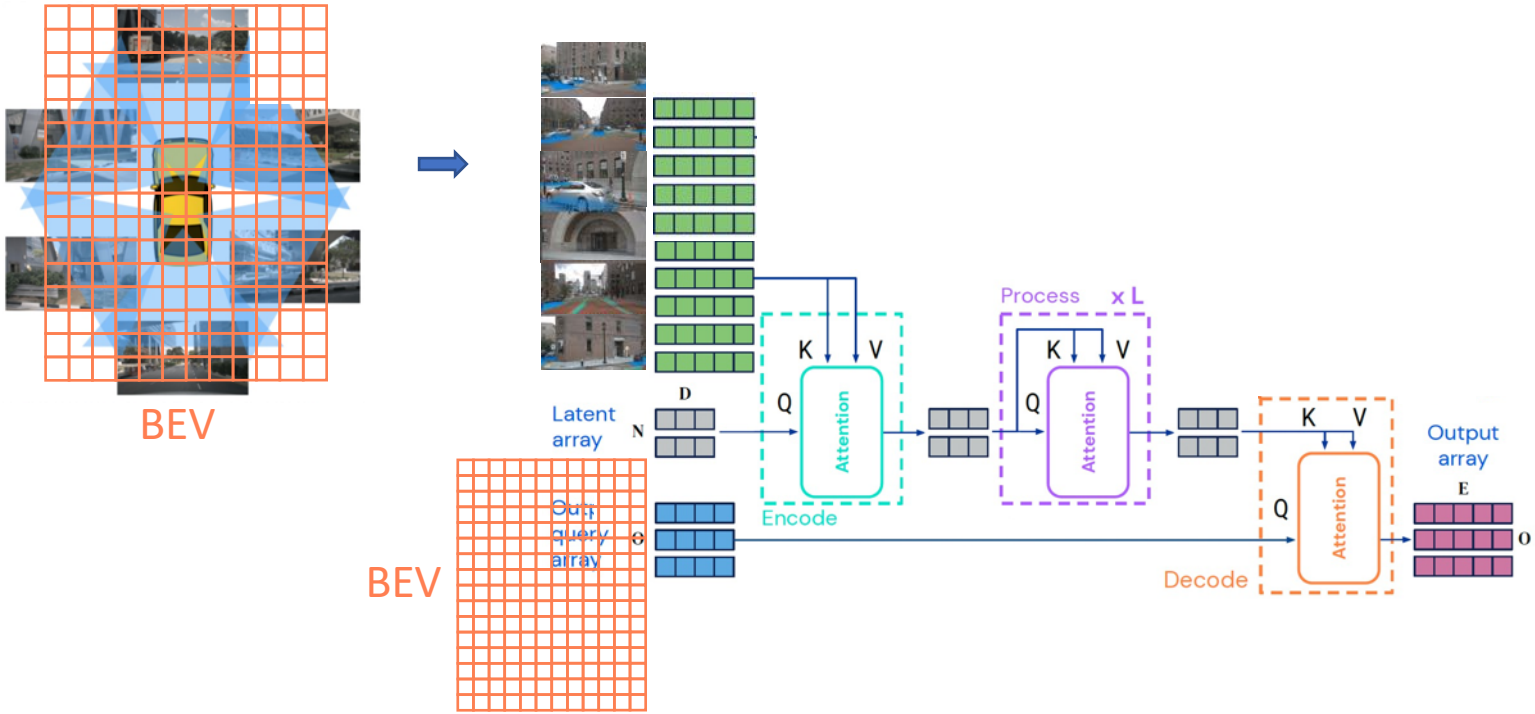




General Encoder / Decoder

Input array = N cameras

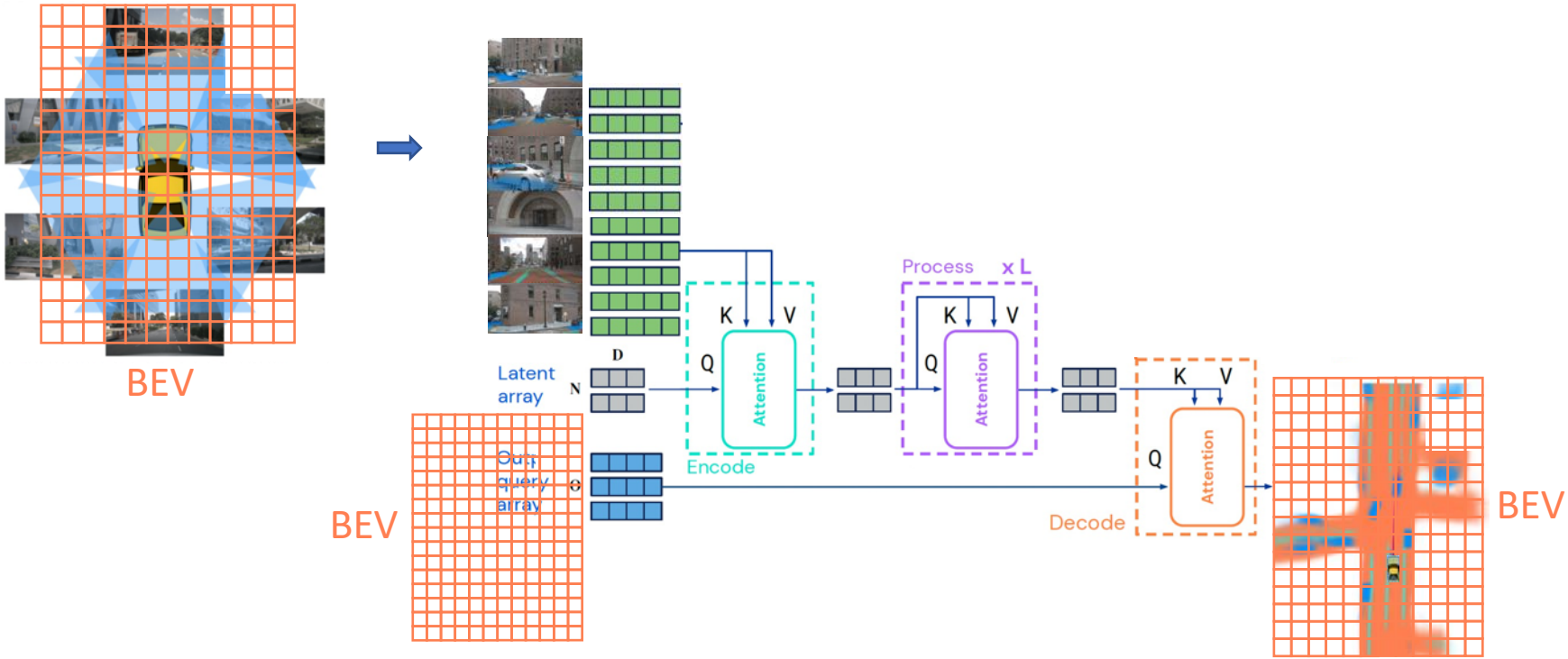
Output array = Bird Eye View (BEV) representation



General Encoder / Decoder

Input array = N cameras

Output array = Bird Eye View (BEV) representation



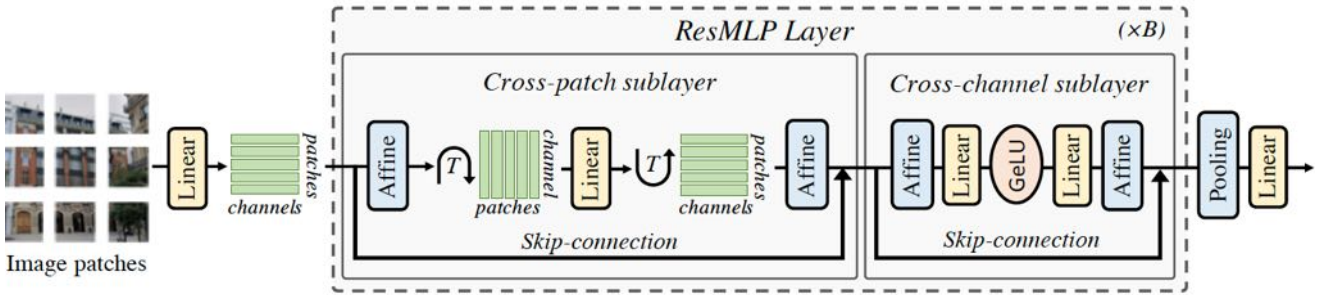
Vision Transformers

Global Attention mechanism at every layer of the deep archi

Very **competitive architectures** in image classification with the best Convnets

Fusion/Merging by mixing thanks to cross attention process

Somehow universal deep structure around encoding/decoding for many vision tasks as classification (1 class token), object detection, segmentation, …



Outline

1. Attention (from NLP) and Vision Transformers (ViT)
2. Transformer Decoder for downstream tasks
3. **Mixing Visual and Language Models**

CLIP: Vision + Language Models (VLM)

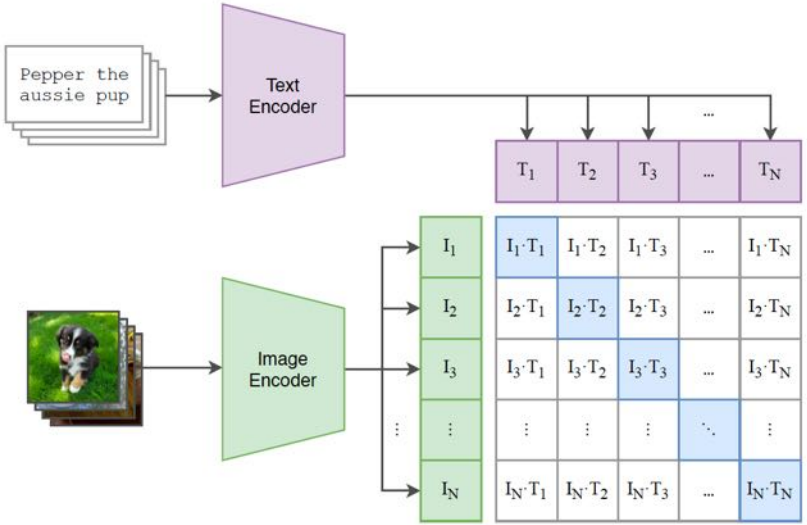
[Learning transferable visual models from natural language supervision. Radford/Sutskever ICML, 2021]

CLIP: Vision + Language Models (VLM)

[Learning transferable visual models from natural language supervision. Radford/Sutskever ICML, 2021]

Massive Text+Image =**500M pairs** pre-trained model (from Internet = no manual labeling)

Contrastive loss for pre-training

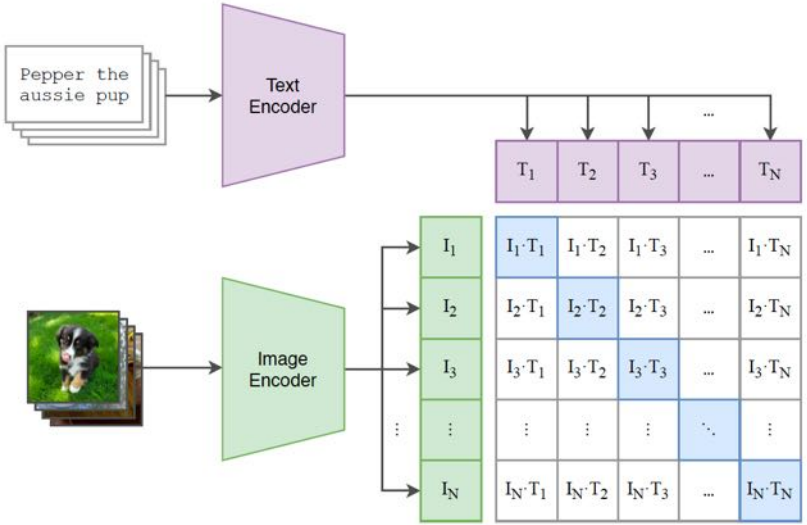


CLIP: Vision + Language Models (VLM)

[Learning transferable visual models from natural language supervision. Radford/Sutskever ICML, 2021]

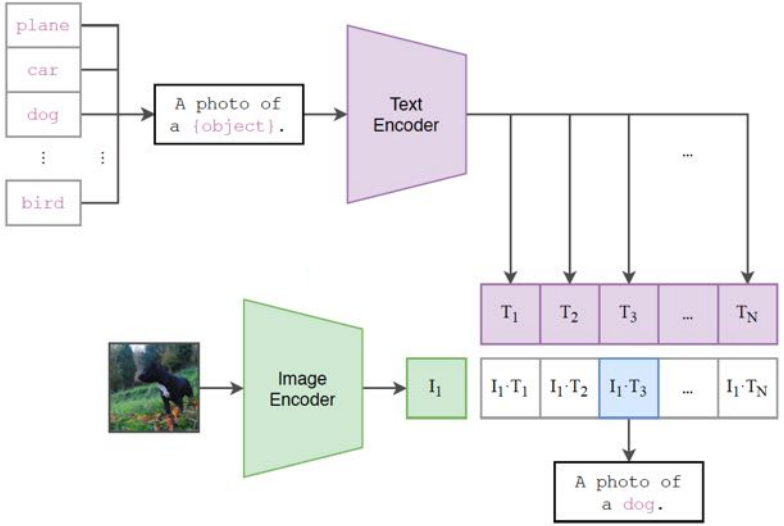
Massive Text+Image =**500M pairs** pre-trained model (from Internet = no manual labeling)

Contrastive loss for pre-training



Pre-trained encoders = **dual encoders** (Text/Image)

used for Zero-shot classifier, and other downstream tasks

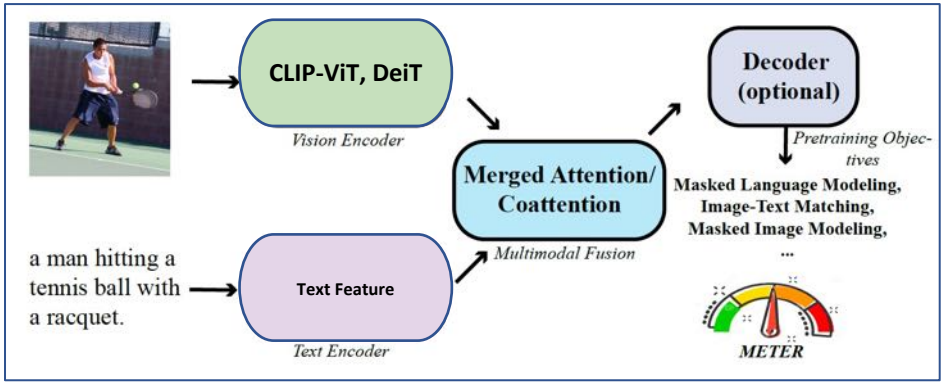


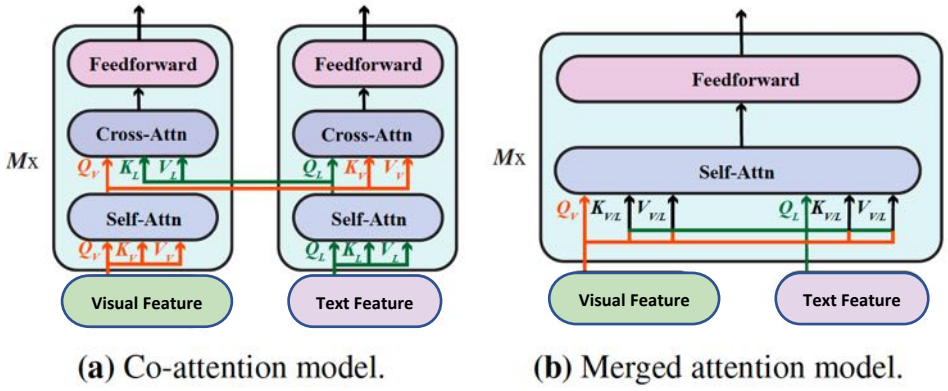
Training End-to-End Vision-and-Language Transformers

From **dual** encoders (Text/Image)

Training VLM Multimodal Fusion on specific datasets and pretext tasks

Commonly used datasets (4M images in total): COCO, Conceptual Captions, SBU Captions, and Visual Genome





(a) Co-attention model. (b) Merged attention model.

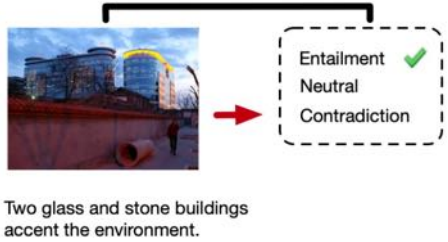
VLM experiments

Several Complex visual understanding tasks
VLMs give the **best** results

Visual reasoning (NLVR2)
Answer: yes or no

Visual Question Answering

Visual Entailment



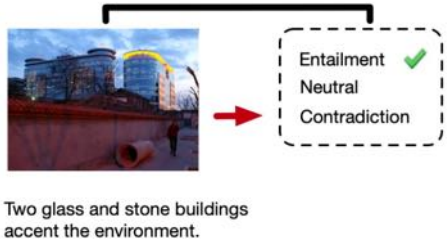
VLM experiments

Several Complex visual understanding tasks
VLMs give the **best** results

Visual reasoning (NLVR2)
Answer: yes or no

Visual Question Answering

Visual Entailment



VLM => Foundation models

What is a foundation model? " *In recent years, a new successful paradigm for building AI systems has emerged: Train one model on a huge amount of data and adapt it to many applications. We call such a model a foundation model*" (from Stanford Center for Research on Foundation Models).

Language/vision/etc models (BERT/GPT3+ViT…) with (huge) training data
From Web pages => Sequence of tokens (text,images, video,audio) no labeling required

To sum up

2010s: Convnets with imagenet (1.2M images 1000 class/words) per-sample labels

2020s?: Transformers => All is Token and transformers for all tokens

- Starting with 500M pairs of images+text => free language description

Many ongoing research challenges:

- Training foundation Models
- Building extra models on top of VLM that can be rapidly adapted to numerous tasks using only a handful of annotated examples (or even none)
- Better understanding the latent structure of these multimodal spaces and interaction functions

Q: Here it is about 1B images for learning, how it compares with our brain visual experience?

Thank you!

Few related publications:

- Deit iii: Revenge of the vit H Touvron, M Cord, H Jégou ECCV 2022
- Three things everyone should know about Vision Transformers H Touvron, M Cord, A El-Nouby, J Verbeek, H Jégou ECCV 2022
- CaiT: Going deeper with image transformers H Touvron, M Cord, A Sablayrolles, G Synnaeve, H Jégou ICCV 2021
- DeiT: Training data-efficient image transformers & distillation through attention H Touvron, M Cord, M Douze, F Massa, A Sablayrolles, H Jégou ICML 2021
- DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion A Douillard, A Ramé, G Couairon, M Cord CVPR 2022
- Dynamic Query Selection for Fast Visual Perceiver C Dancette, M Cord CVPR workshop 2022
- Transformer Decoders with MultiModal Regularization for Cross-Modal Food Retrieval M Shukor, G Couairon, A Grechka, M Cord CVPR workshop 2022
- Explainability of vision-based autonomous driving systems: Review and challenges É Zablocki, H Ben-Younes, P Pérez, M Cord IJCV 2022
- Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering, C Dancette, R Cadene, D Teney, M Cord, ICCV 2021
- RUBi: Reducing Unimodal Biases in Visual Question Answering, R Cadene, C Dancette, H Ben-younes, M Cord, D Parikh NeurIPS 2019