Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 2.2: Unimodal Representations (Part 2)

Louis-Philippe Morency

** Original course co-developed with Tadas Baltrusaitis. Major revision co-developed with Paul Liang. Co-lecturers included Yonatan Bisk, Daniel Fried and Carlos Busso.*

Administrative Stuff

Team Matching Event – Today!

Today around 10:30am ET

(later part of the lecture)

- Detailed instructions will be shared during lecture
- Event optional for students who already have a full team

Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 2.2: Unimodal Representations (Part 2)

Louis-Philippe Morency

** Co-lecturer: Paul Liang. Original course co-developed with Tadas Baltrusaitis. Spring 2021 and 2022 editions taught by Yonatan Bisk*

Lecture Objectives

- Image representations
  - Image gradients, edges, kernels
- Convolution neural network (CNN)
  - Convolution and pooling layers
  - Visualizing CNNs
- Vision Transformers
  - CNN with self-attention
  - Masked auto-encoder

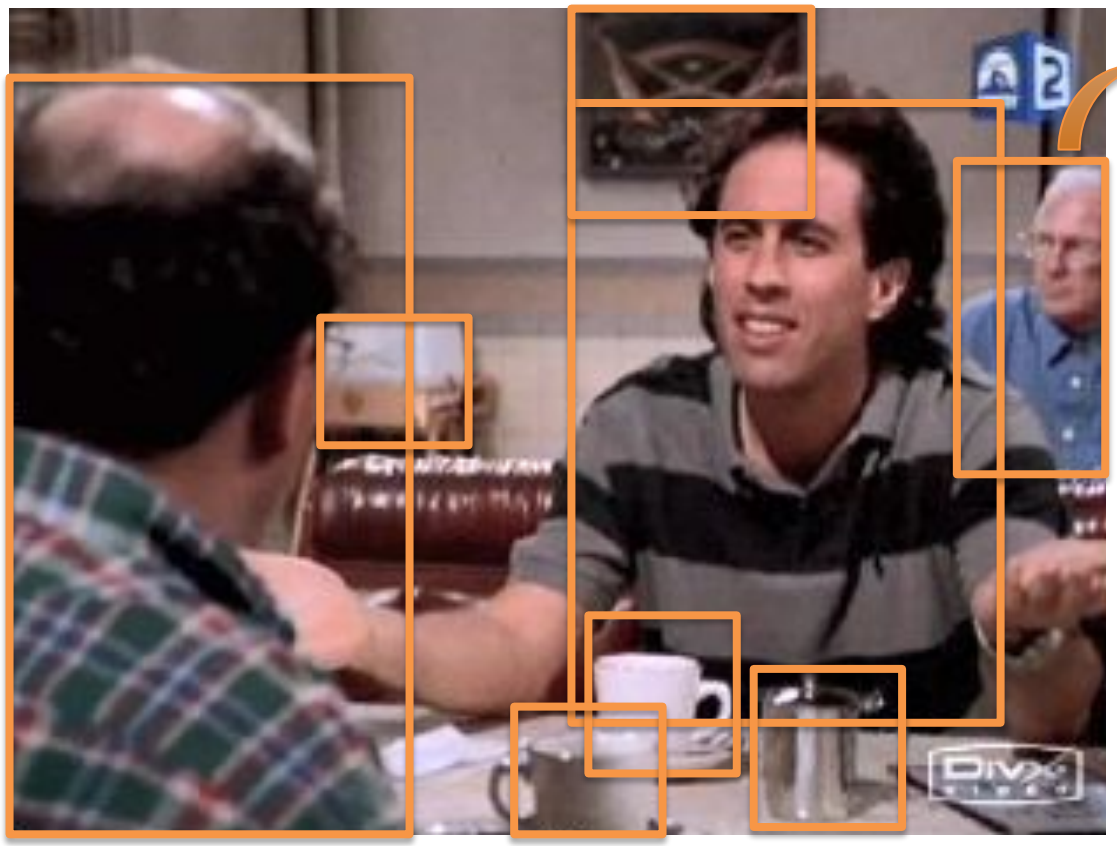
Image Representations

How to Better Describe This Image?

?

88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80
...

Object-Based Visual Representation



"person" label +

Appearance descriptor

- ❑ Age
- ❑ Expression
- ❑ Clothes
- ❑ …

Feature vector

Object Descriptors

How to represent and detect an object?

Many approaches over the years…

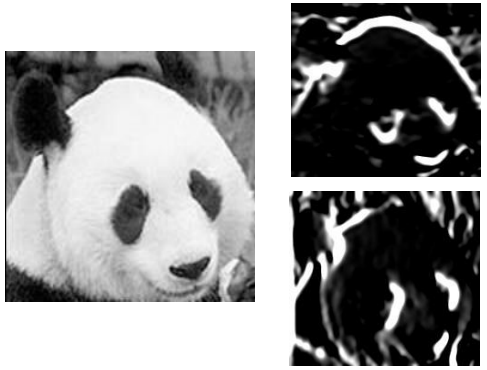
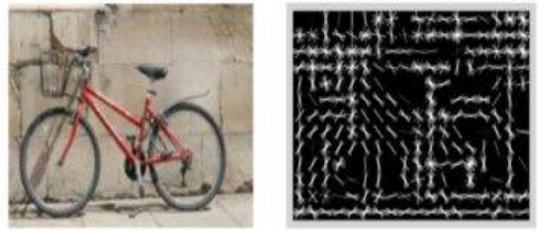


Image gradient

Edge detection



Histograms of Oriented Gradients

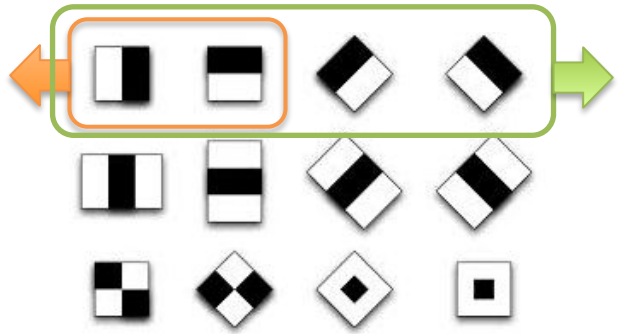
Optical Flow

Object Descriptors

How to represent and detect an object?

Many approaches over the years…

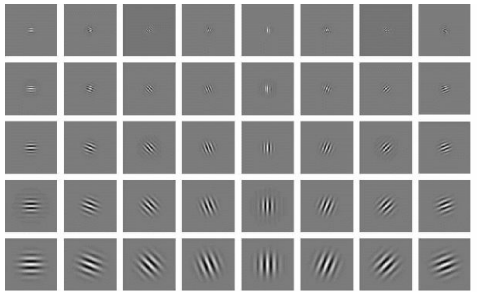
Horizontal and vertical gradients



Oriented gradients

Haar Wavelets

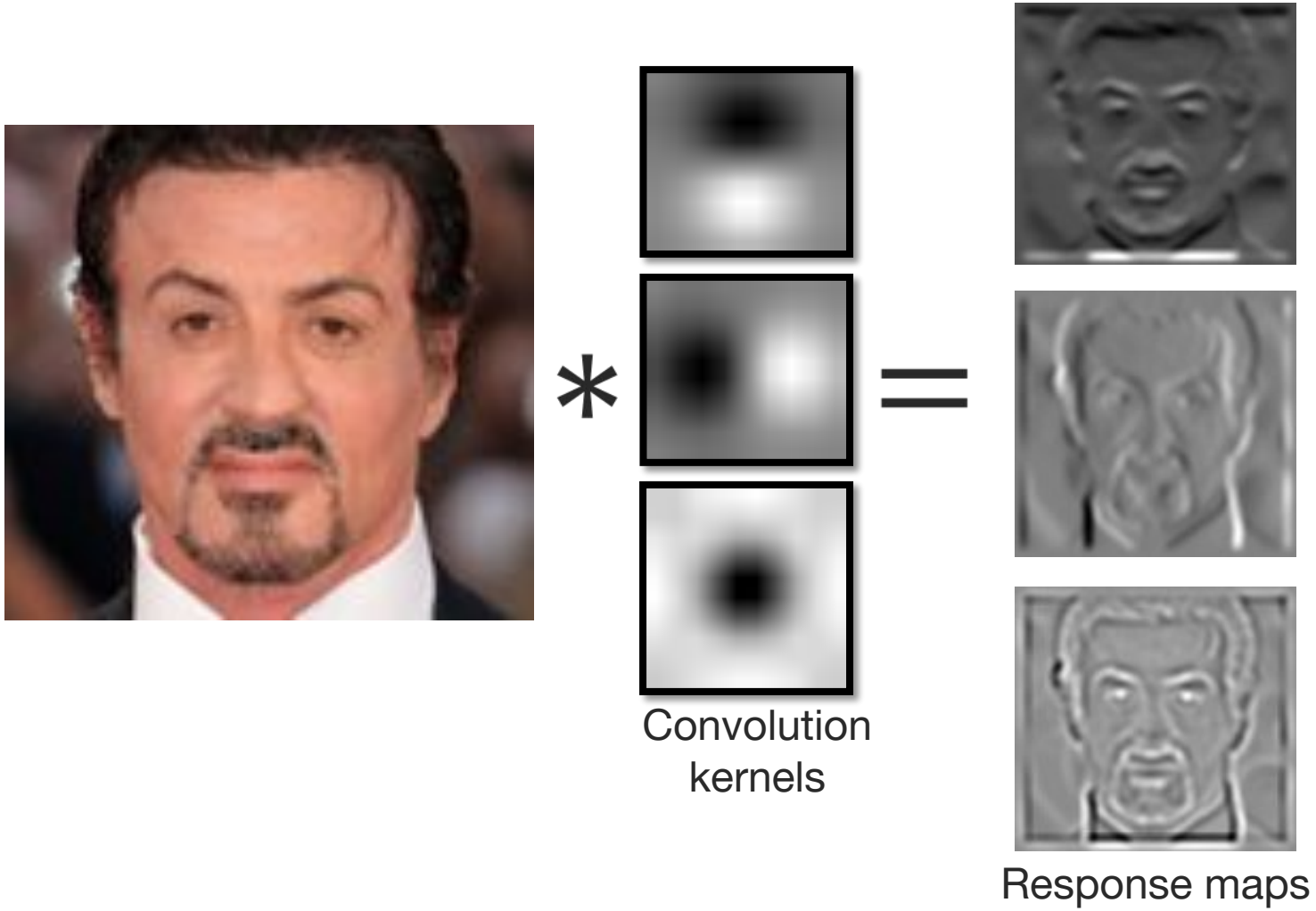
Templates tested on the image *(i.e., convolution kernels)*



Gabor filters

Inspired by visual cortex

Convolution Kernels



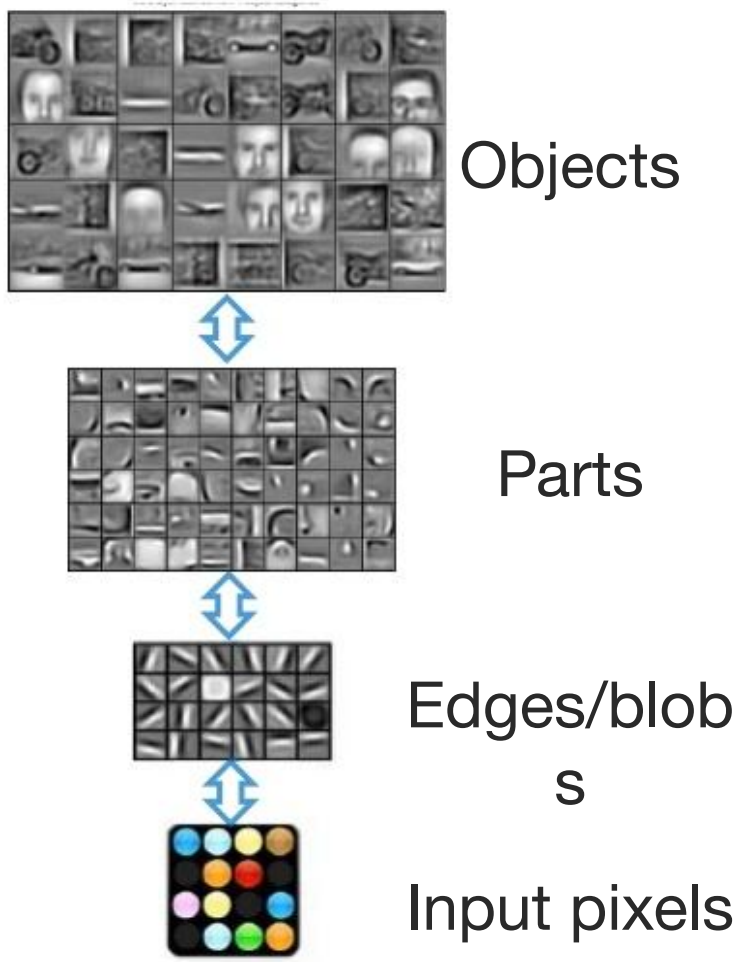
Convolutional Neural Networks

Why using Convolutional Neural Networks?

Goal: building more abstract, hierarchical visual representations

Key advantages:

1) Inspired from visual cortex
2) Encourages visual abstraction
3) Exploits *translation invariance*
4) Kernels/templates are learned
5) Fewer parameters than MLP



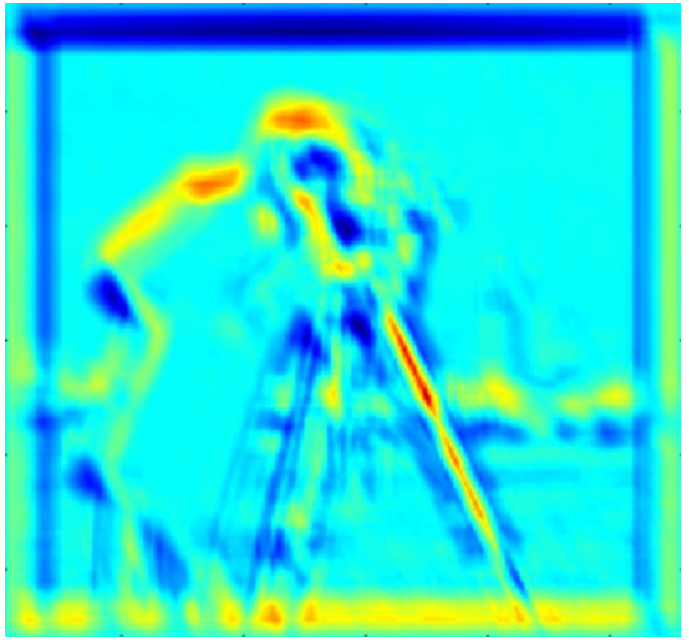
Convolution in 2D – Example

Input image

*

Convolution kernel

=



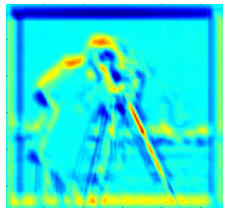
Response map

Convolution as a Fully-Connected Network

Input

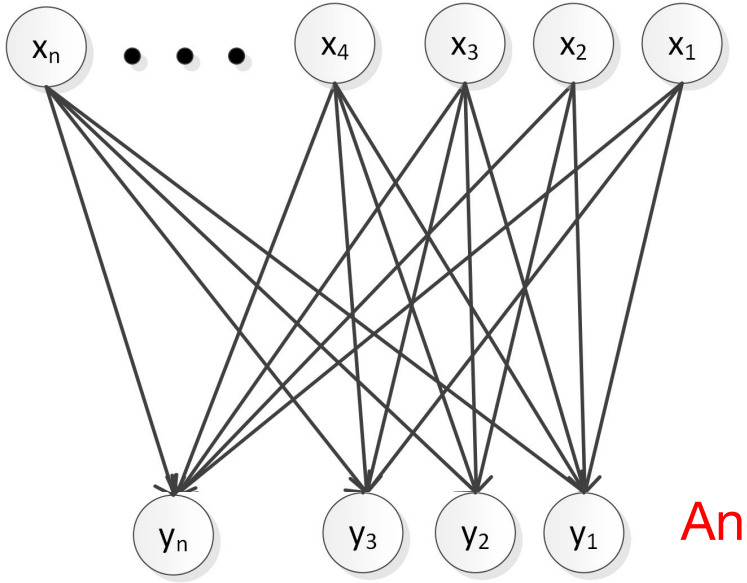
(image)

Output



(response map)

Input: all pixels



Output: kernel responses

Not efficient!

200×200 image
requires
$40{,}000 \times n$ parameters

(where n is size of kernel)

And it may learn different kernels for different pixel positions

➡ Not translation invariant

Convolutional Neural Layer

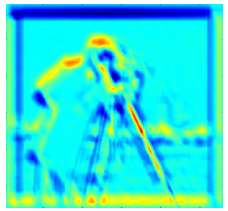
Input

(image)

Weighted sum

Wx

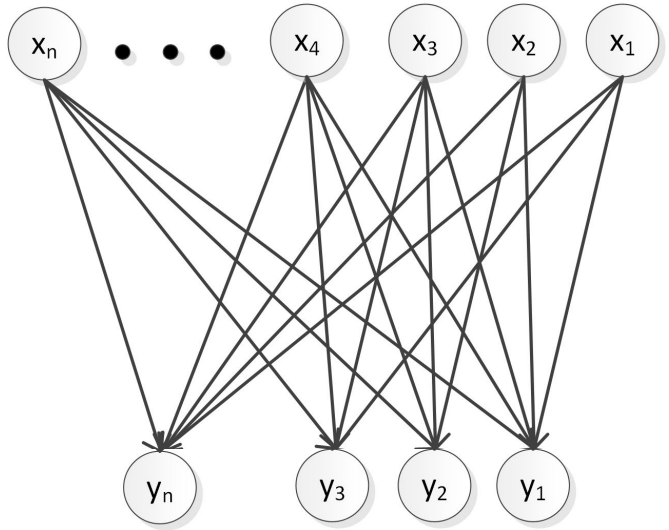
Output



(response map)

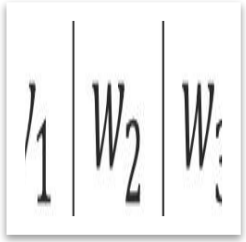
$y = Wx$

Input: all pixels



Output: kernel responses

Example with 1D kernel:



Convolution kernel

Convolutional Neural Layer

Modification 1: Sliding window – Only apply the kernel to a small region

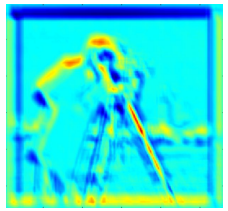
Input

(image)

Weighted sum

Wx

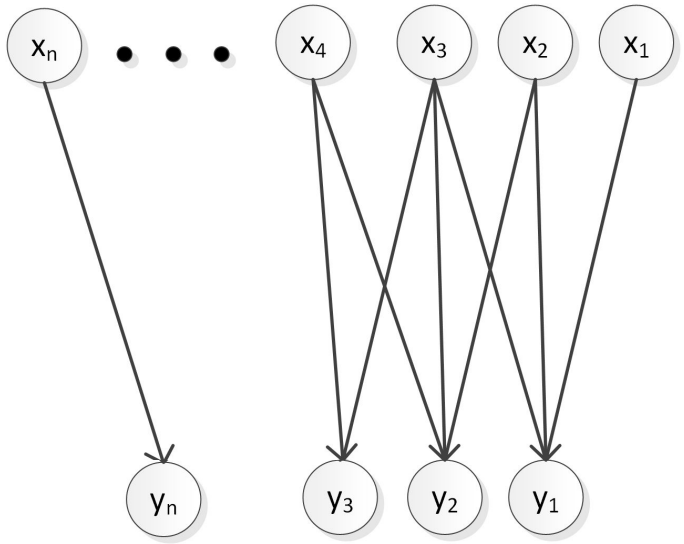
Output



(response map)

$y = Wx$

Input: all pixels



Output: kernel responses

Example with 1D kernel:

Convolutional Neural Layer

Modification 2: Same kernel applied to all sliding windows

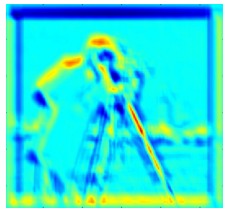
Input

(image)

Weighted sum

Wx

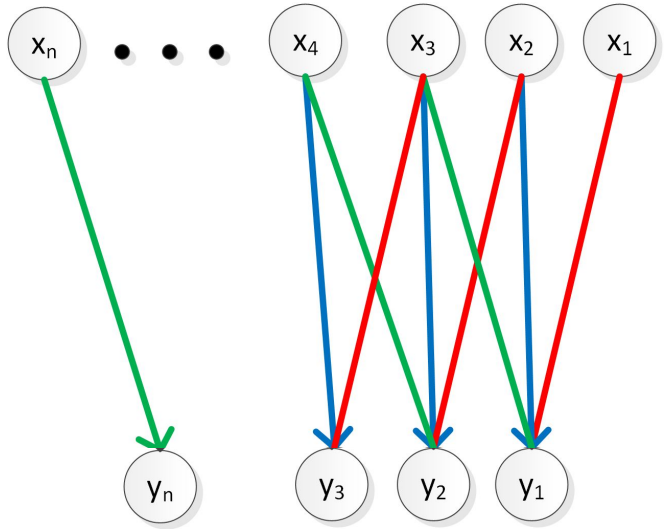
Output



(response map)

$y = Wx$

Input: all pixels



Output: kernel responses

Example with 1D kernel:

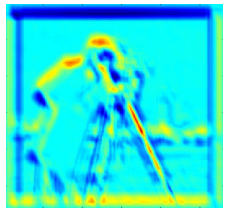
Convolutional Neural Layer

Modification 2: Same kernel applied to all sliding windows

Input

(image)

Output



(response map)

$$y = Wx$$

$$W = \begin{pmatrix} w_1 & w_2 & w_3 & & 0 & 0 & 0 \\ 0 & w_1 & w_2 & \cdots & 0 & 0 & 0 \\ 0 & 0 & w_1 & & 0 & 0 & 0 \\ \vdots & & & \ddots & \vdots & & \\ 0 & 0 & 0 & & w_3 & 0 & 0 \\ 0 & 0 & 0 & \cdots & w_2 & w_3 & 0 \\ 0 & 0 & 0 & & w_1 & w_2 & w_3 \end{pmatrix}$$

Example with 1D kernel:

- Can be implemented efficiently on GPUs
- *W* will be 3D: 3rd dimension allows for multiple kernels

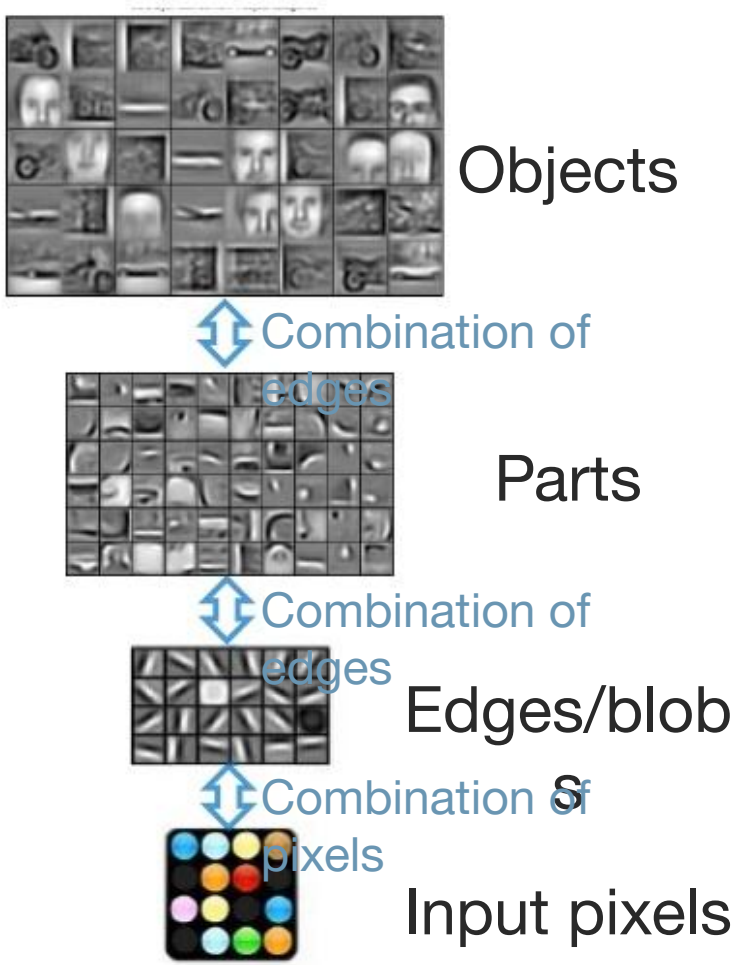
Convolutional Neural Network

Multiple convolutional layers

➡ Allows the network to learn combinations of sub-parts, to increase complexity

but how to encourage abstraction and summarization?

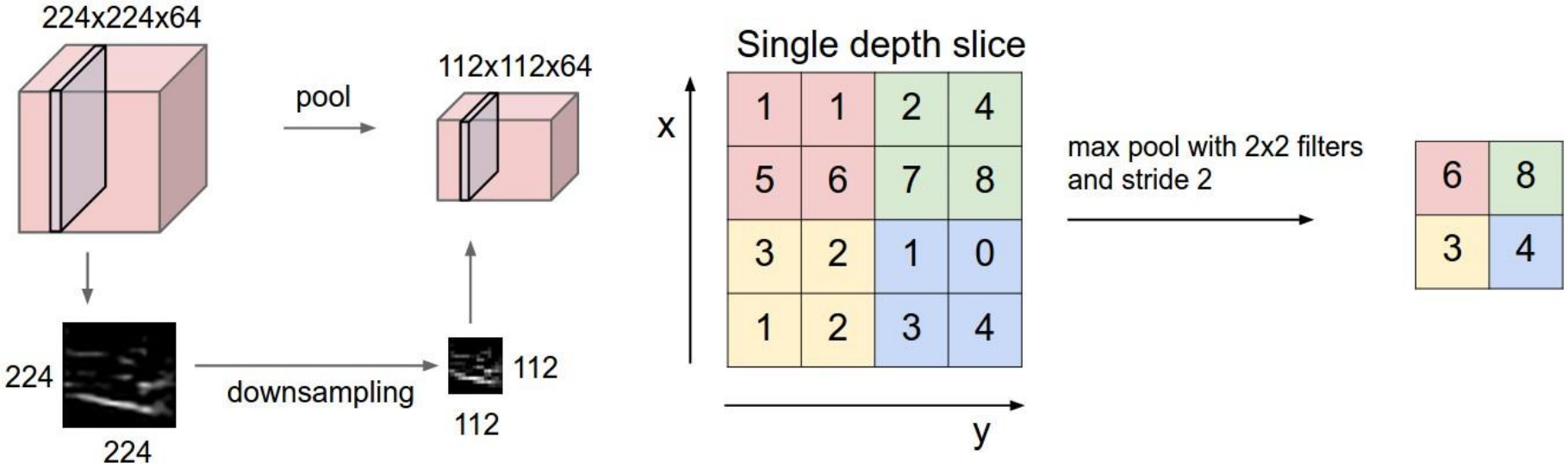
Answer: Pooling layers



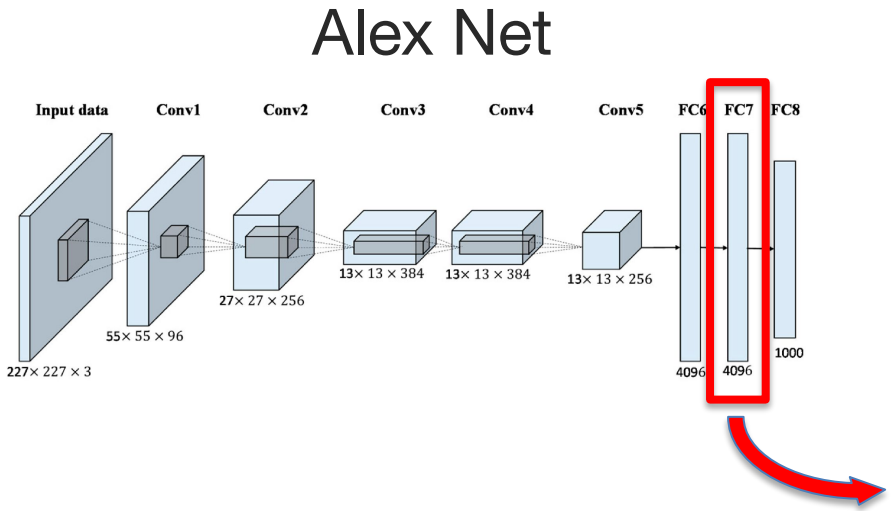
Pooling Layer

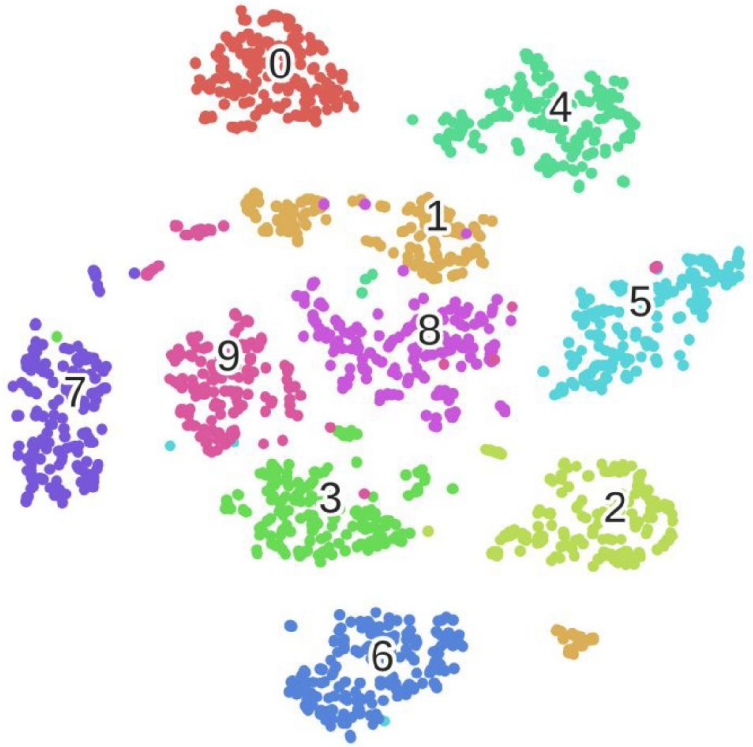
Response map subsampling:

Allows summarization of the responses



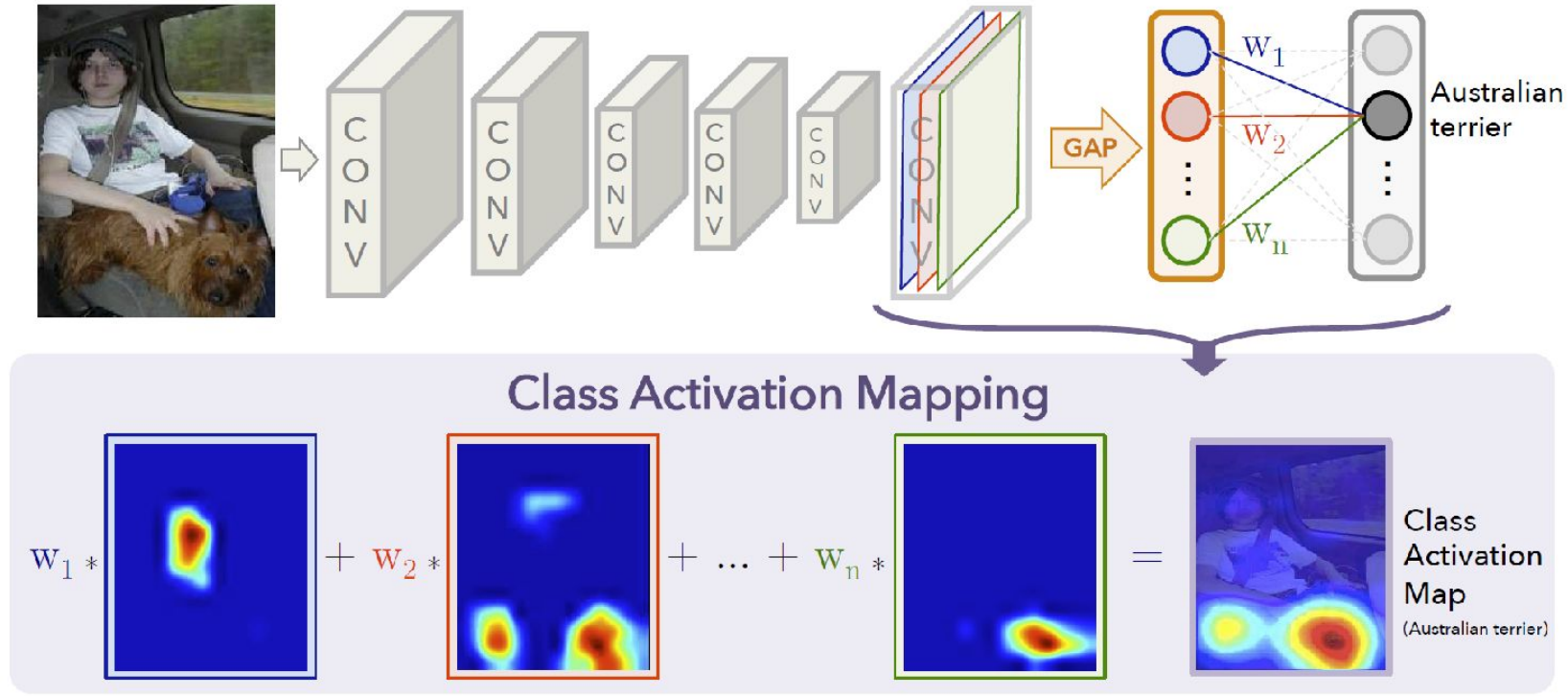
Visualizing the Last CNN Layer: t-sne





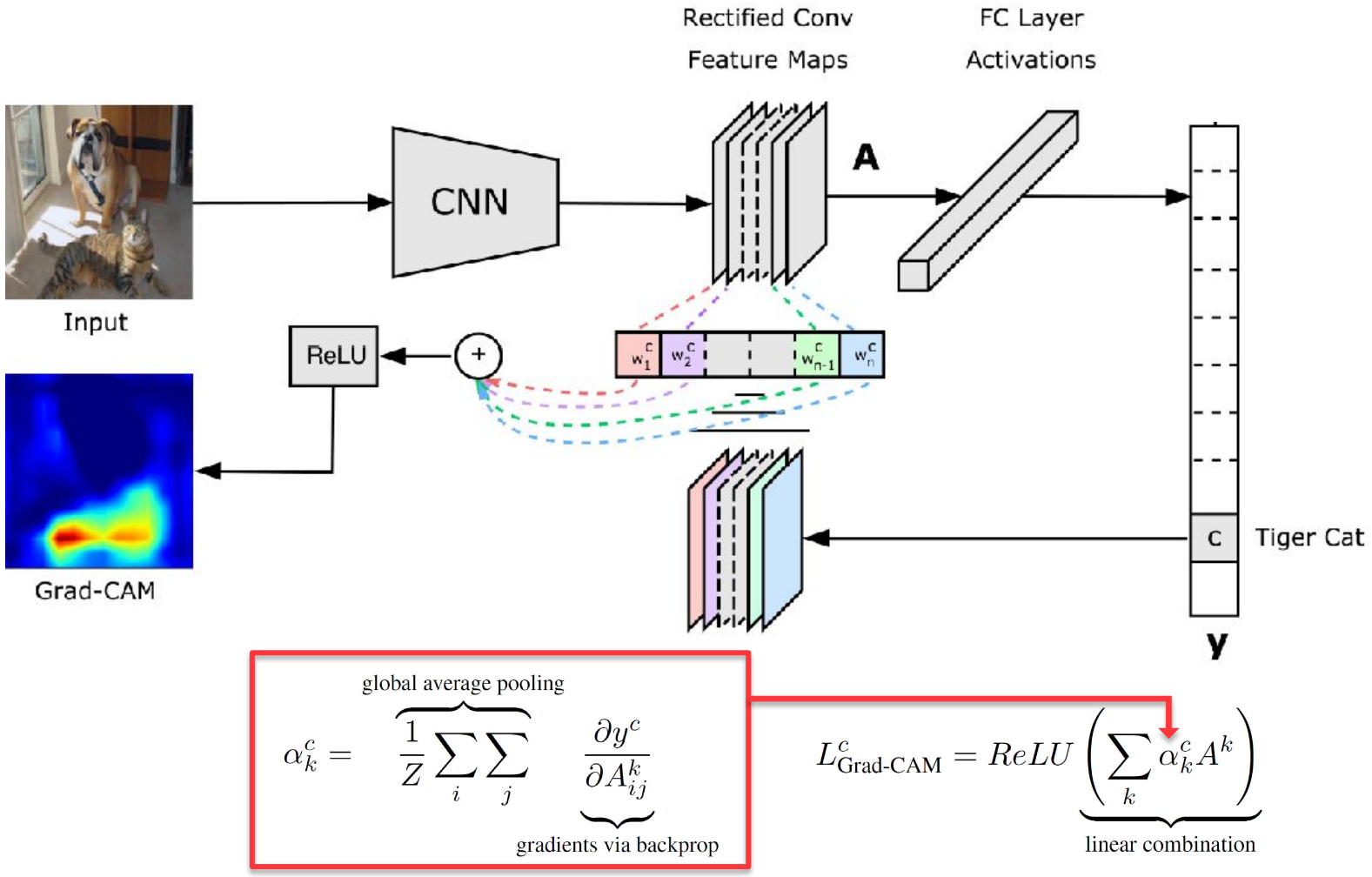
Embed high dimensional data points (i.e. feature codes) so that pairwise distances are conserved in local neighborhoods.

CAM: Class Activation Mapping [CVPR 2016]



$$L^c_{\text{CAM}} = \underbrace{\sum_k w^c_k A^k}_{\text{linear combination}}$$

Grad-CAM [ICCV 2017]



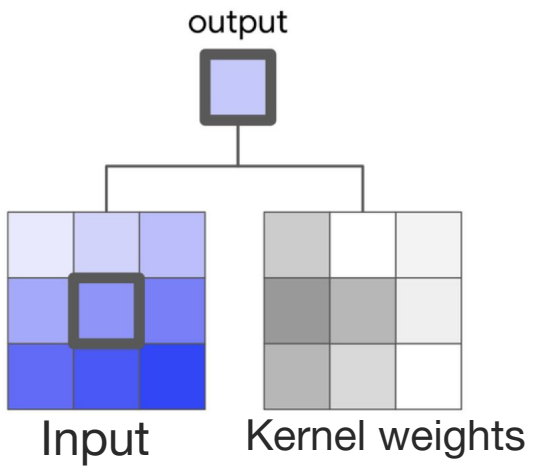
$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Vision Transformers

CNN

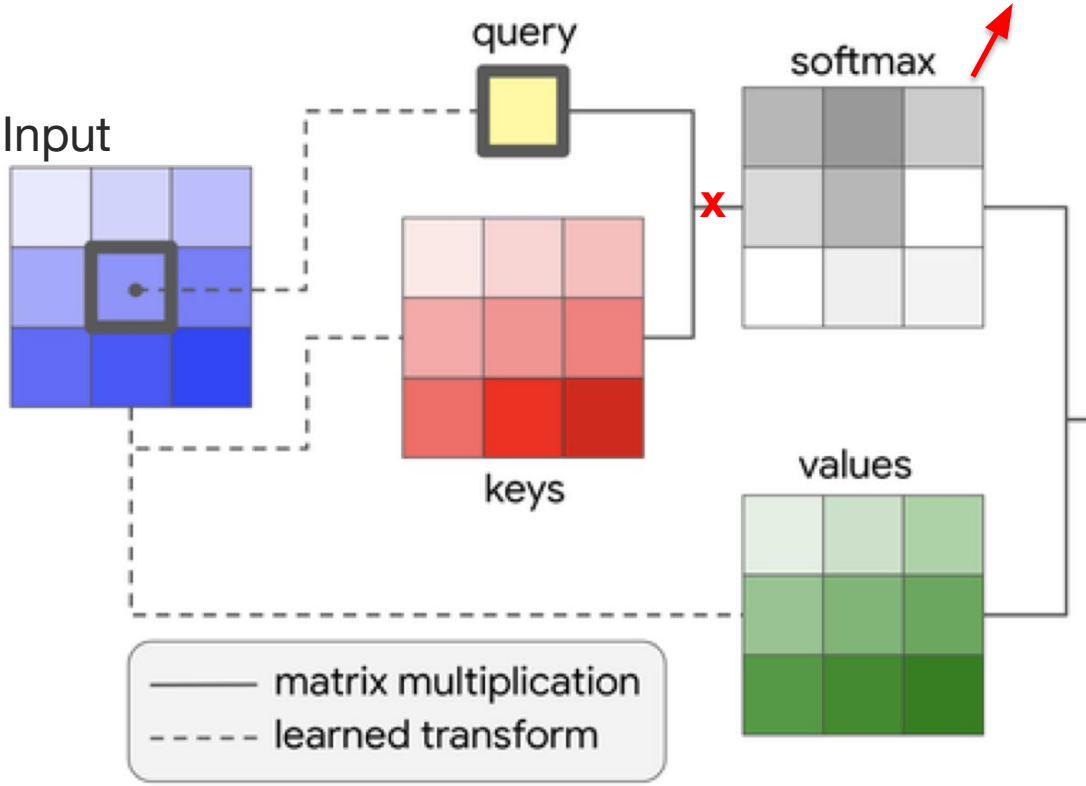
Replacing a CNN w/ Self-Attention



Convolution

How well the query matches the keys?
(How well the pixel matches its neighbors?)

aka "memory block"



Pixel contextualized by its neighbors

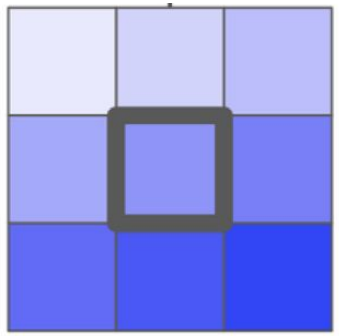
Self-attention

But how to encode position?

https://arxiv.org/abs/1906.05909

Replacing a CNN w/ Self-Attention

Image patch



2D relative position embedding

-1, -1	-1, 0	-1, 1	-1, 2
0, -1	0, 0	0, 1	0, 2
1, -1	1, 0	1, 1	1, 2
2, -1	2, 0	2, 1	2, 2

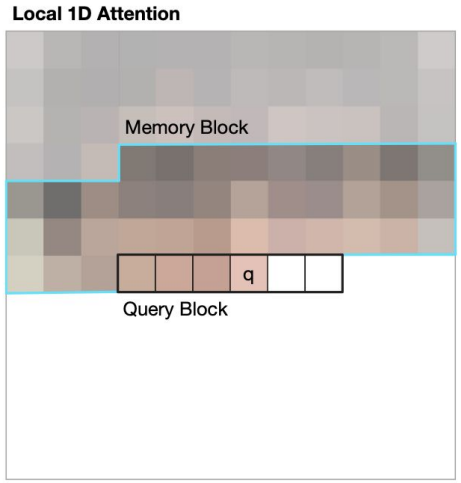
Position embedding is added to the key:

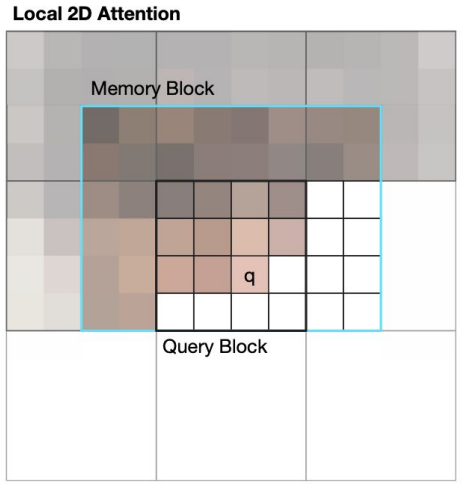
$$y_{ij} = \sum_{a,b \in \mathcal{N}_k(i,j)} \texttt{softmax}_{ab} \left(q_{ij}^\top k_{ab} + q_{ij}^\top r_{a-i,b-j} \right) v_{ab}$$

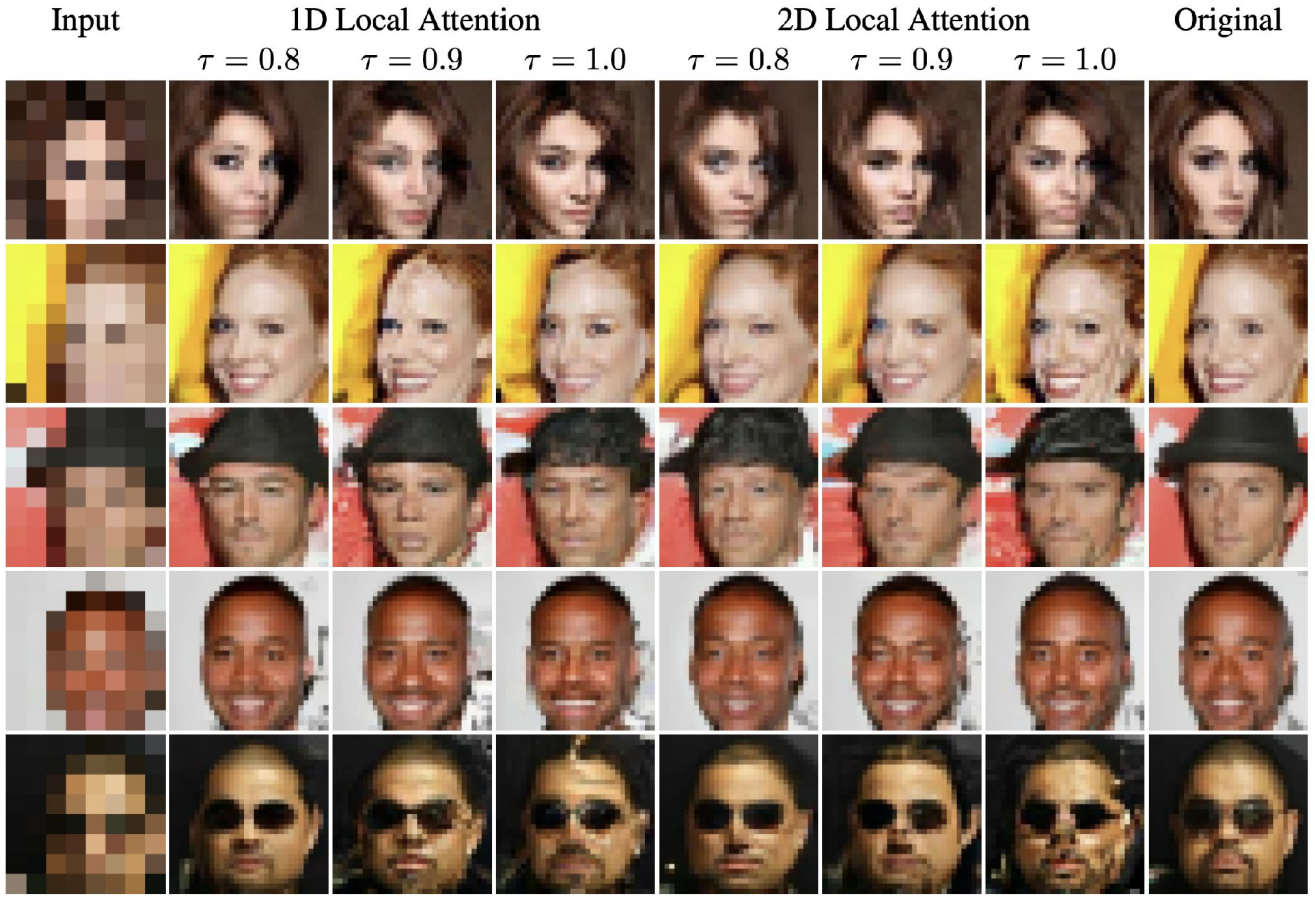
https://arxiv.org/abs/1906.05909

Pixel-Based Image Generation via Transformers

Produce 32x32 images, one channel of each pixel at a time. 3 x 32 x 32 = 3072 positions







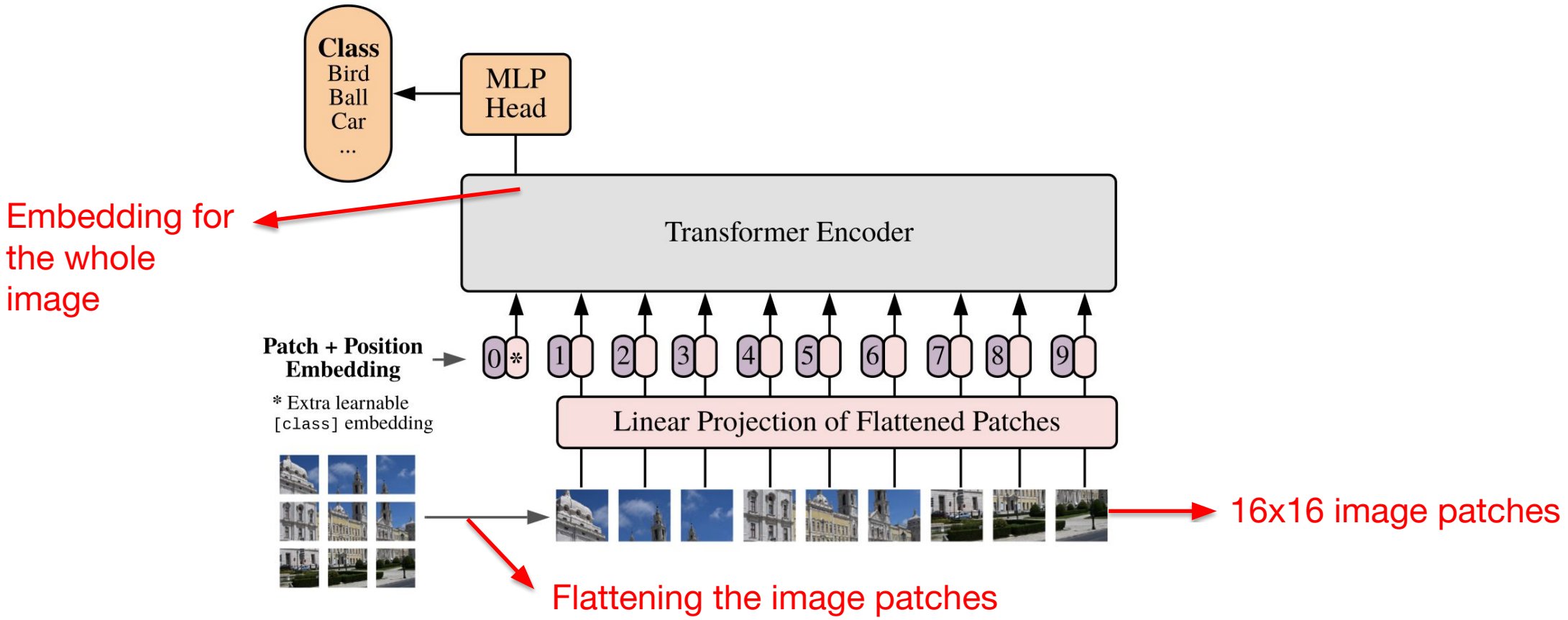
https://arxiv.org/abs/1802.05751

Vision Transformer (ViT)

Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv* (2020).

https://arxiv.org/abs/2010.1192

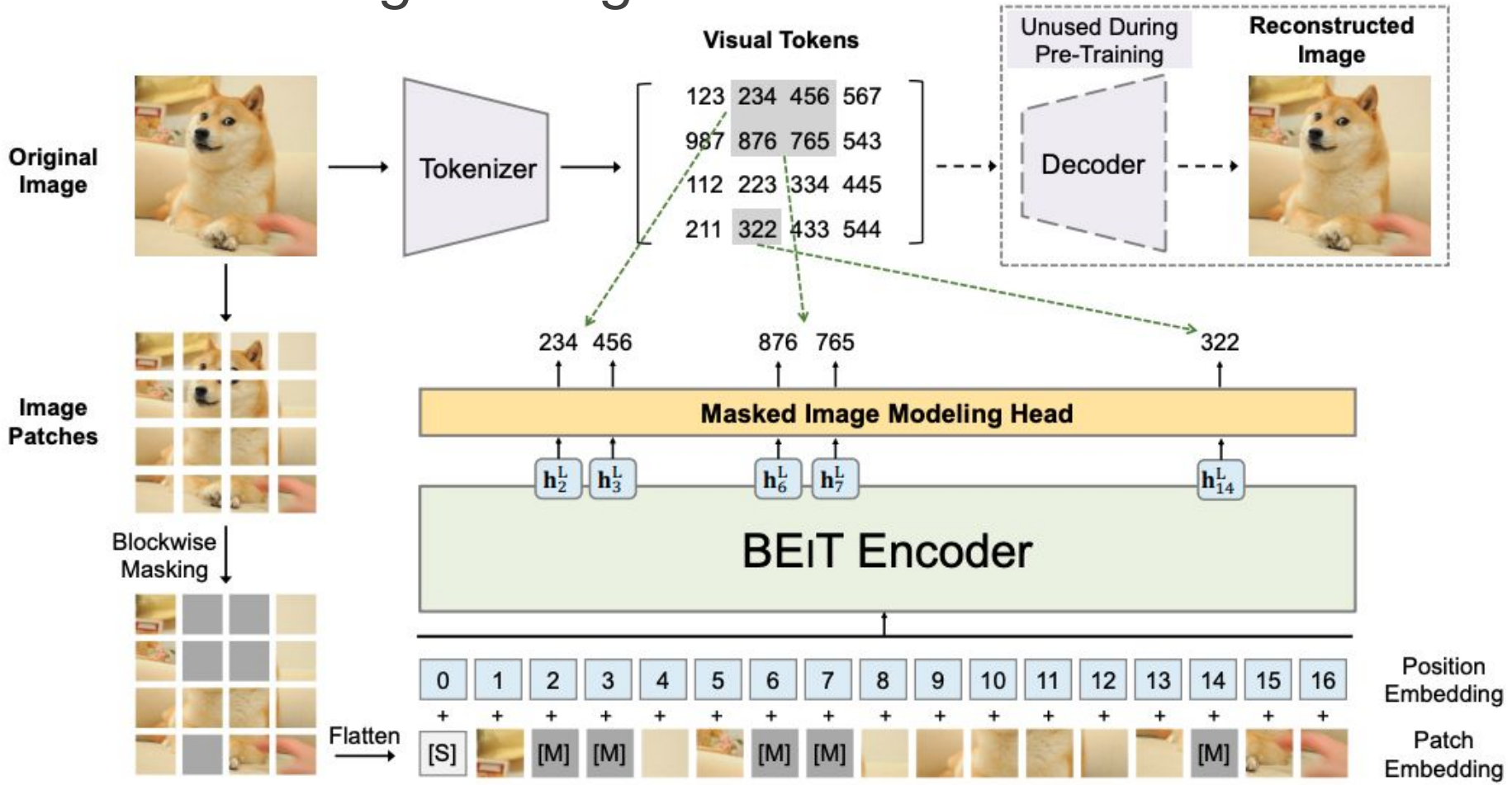
Vision Transformer (ViT)



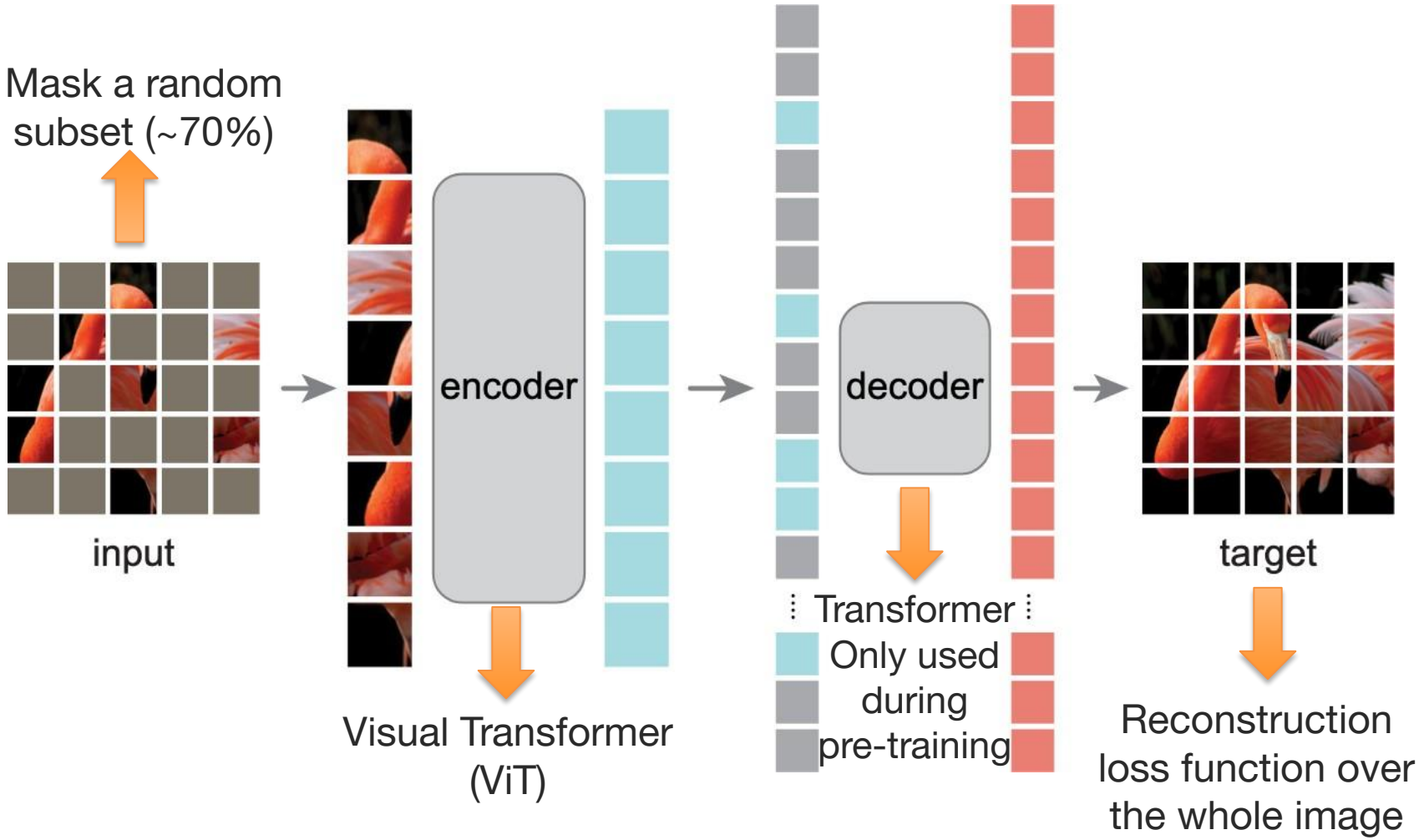
Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv* (2020).

Visual Tokens

BeIT: BERT Pre-Training of Image Transformers

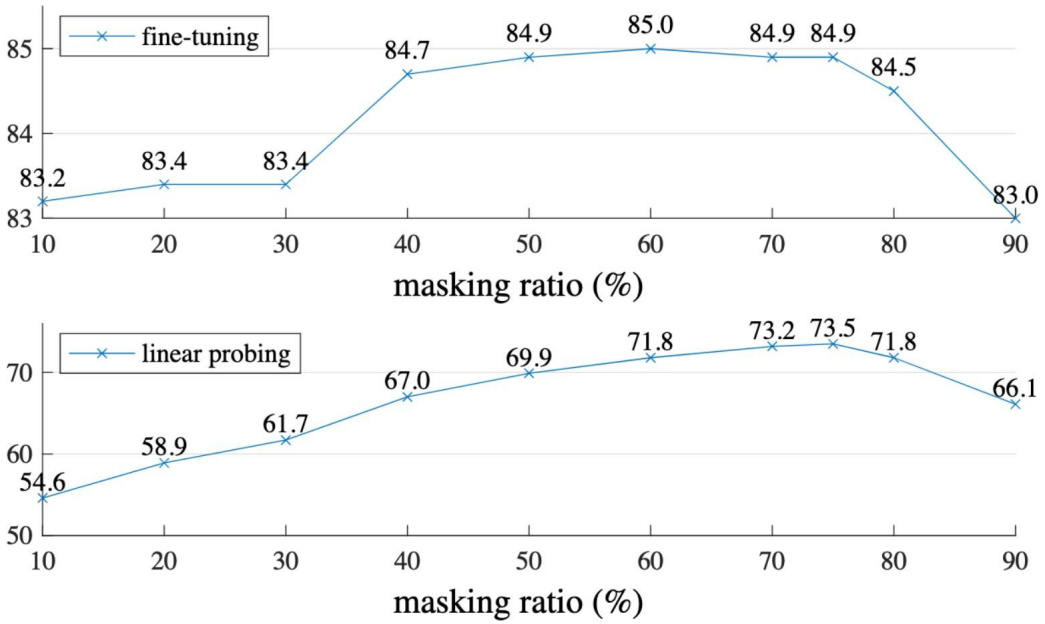


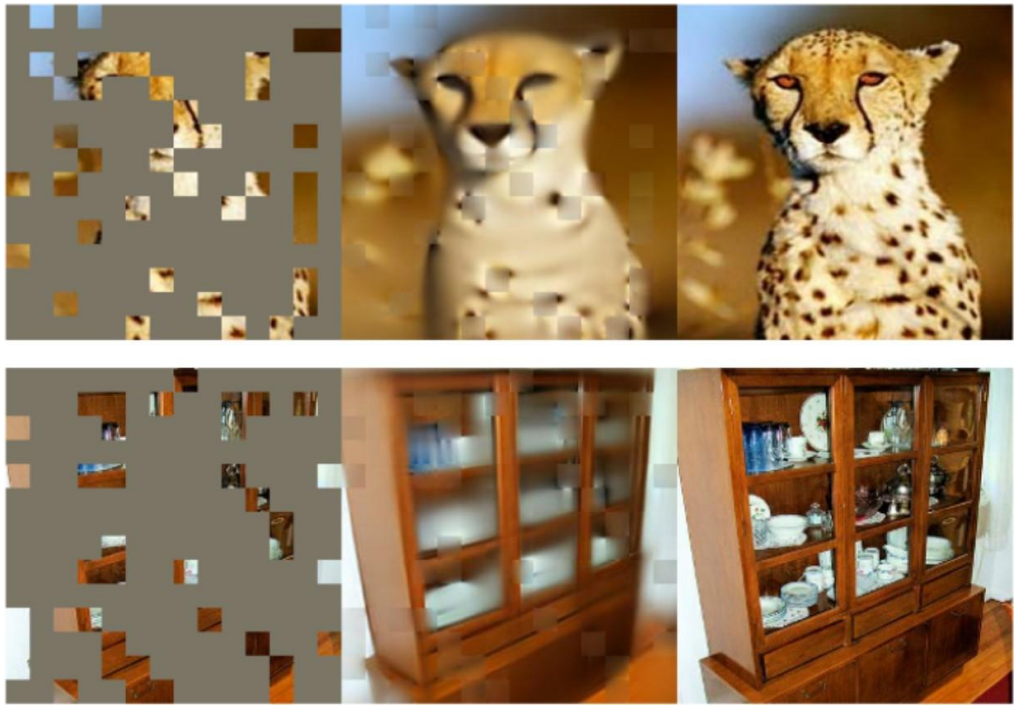
Masked Auto-Encoder (MAE)



He et al., Masked Autoencoders Are Scalable Vision Learners, CVPR 2022

Masked Auto-Encoder (MAE)





He et al., Masked Autoencoders Are Scalable Vision Learners, CVPR 2022