Deep Learning for Computer Vision Part III: Advanced Topics

Computer Vision

Outline

1. Introduction to neural networks – this week
   Basics of neural networks

2. Convolutional neural networks– 13.12
   Basic applications of deep learning to image analysis and computer vision

3. Advanced topics and applications – 20.12

Computer Vision

A bit more detailed outline

3. Advanced topics and applications – 20.12

a. Visualization and diagnostics
b. Localization and classification
c. Unsupervised learning

Visualization and diagnostics

Understanding how a network works

- Challenging task
  - Multivariate interactions, information in different areas of the image are used in interaction with each other
  - Nonlinear mapping between features and labels
  - Hierarchical mapping, information gathered in multiple layers
- Definition of ‘understanding’ is crucial
  - What do you exactly want to get out of the system?
  - There are different approaches with different definitions
  - We will see one particular example: Visualizing features

Visualizing features

- Discussion based on [Zeiler and Fergus 2013]

Visualizing and Understanding Convolutional Networks

Matthew D. Zeiler ZEILER@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

Rob Fergus FERGUS@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

- Visualizing the input that activates a neuron in any layer
- ``Deconvolutional’’ network

Computer Vision

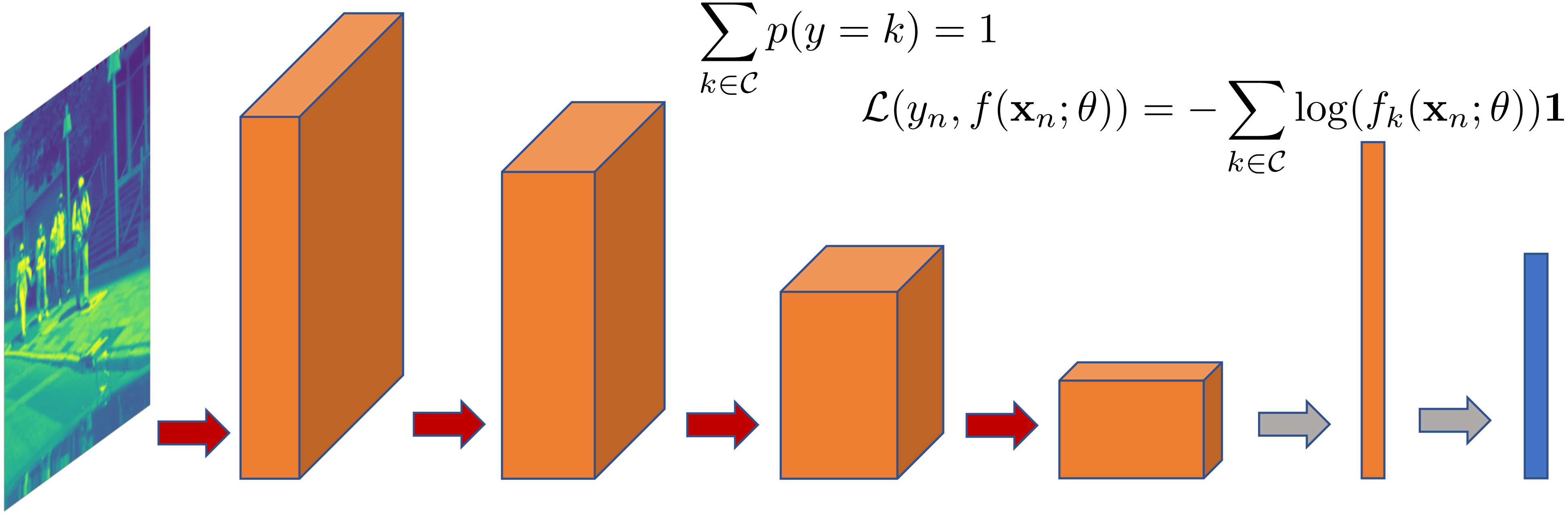
Visualization

General network architecture

$$a_L = \mathbf{W}_L h_{L-1} + b_L \in \mathbb{R}^K \qquad p(y=k) = f_k(\mathbf{x};\theta) = \frac{e^{a_{L,k}}}{\sum_{k' \in \mathcal{C}} e^{a_{L,k'}}}$$

$$\sum_{k \in \mathcal{C}} p(y=k) = 1$$

$$\mathcal{L}(y_n, f(\mathbf{x}_n;\theta)) = -\sum_{k \in \mathcal{C}} \log(f_k(\mathbf{x}_n;\theta))\mathbf{1}(y_n = k)$$



Output
Number of neurons
equal number of classes

Convolution followed by ReLu nonlinearity followed by max-pooling [optionally]

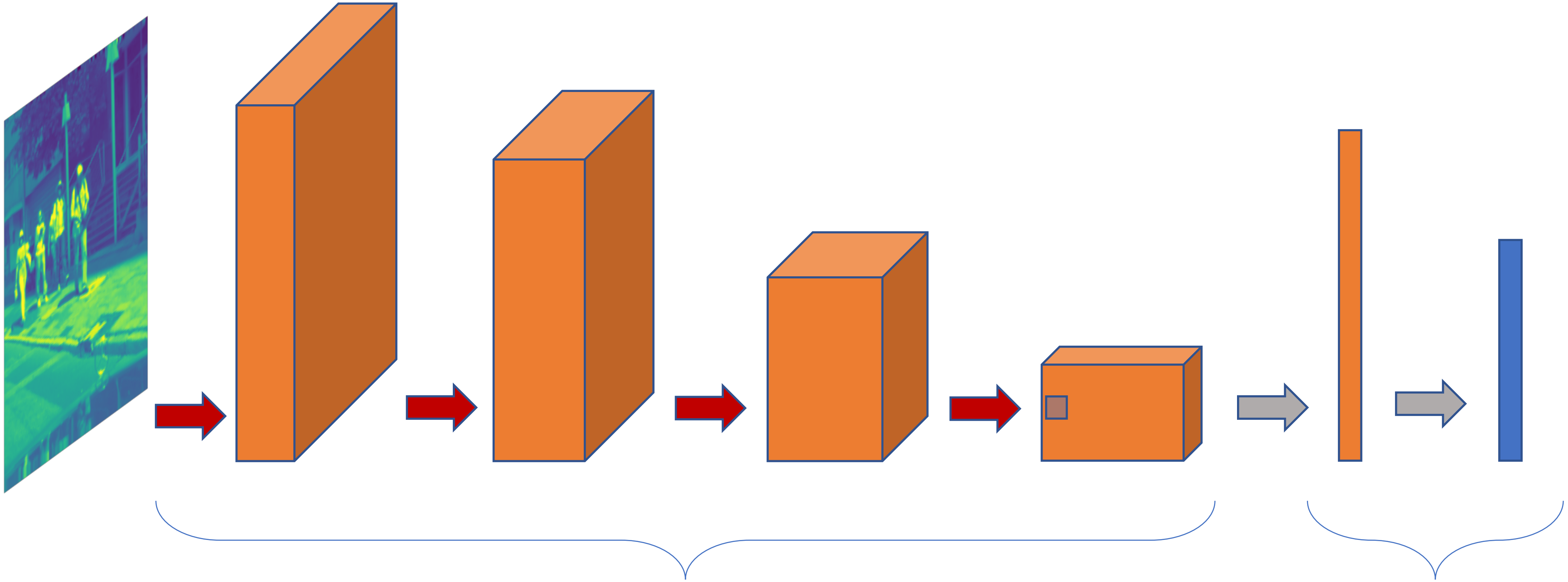
Fully connected layer: transformation followed by non-linearity

Similar to LeCun et al. 1998 and Krizhevsky et al. 2012

Computer Vision

Visualization

Interpreting internal features



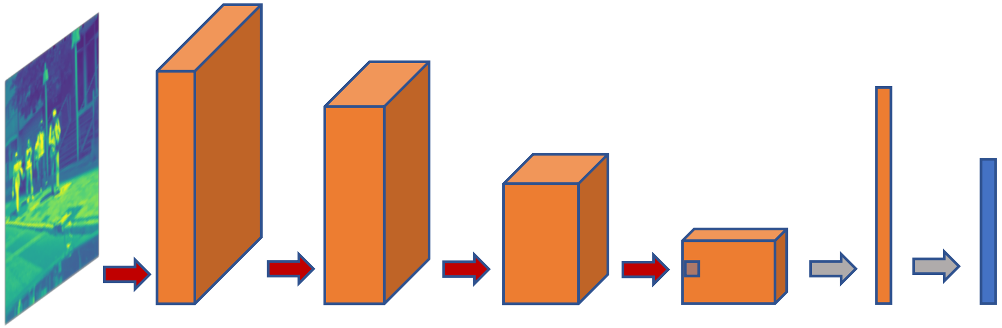
Convolutional layers
What input pattern activates a given neuron in an intermediate layer?

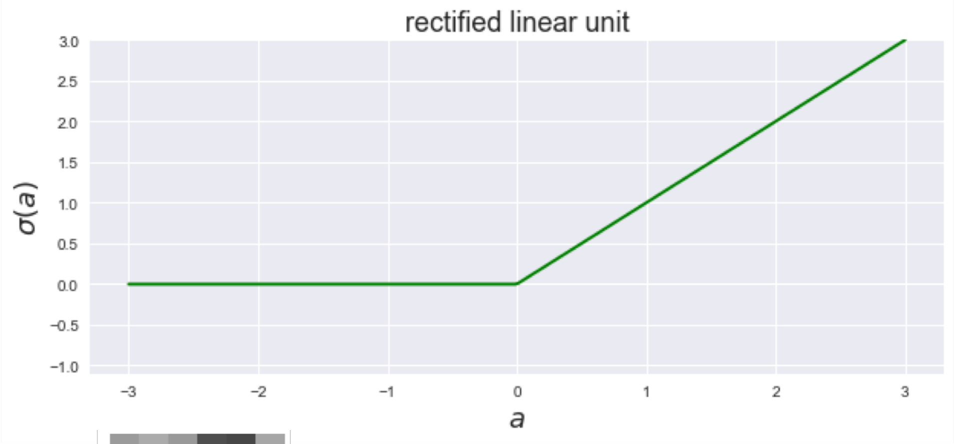
Fully connected layers

Computer Vision

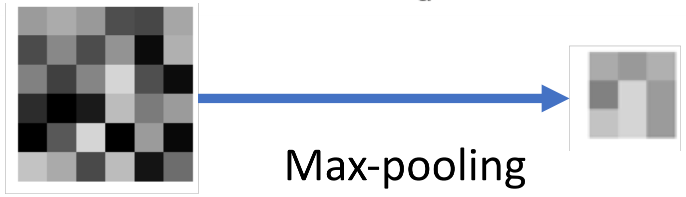
Visualization

Image dependent visualization





- The activation level in the neuron depends on the input image
- For different inputs it will be activated at different levels
- The difference is due to the non-linearity
- If it was linear, neuron’s activation would be based on the respective linear projection

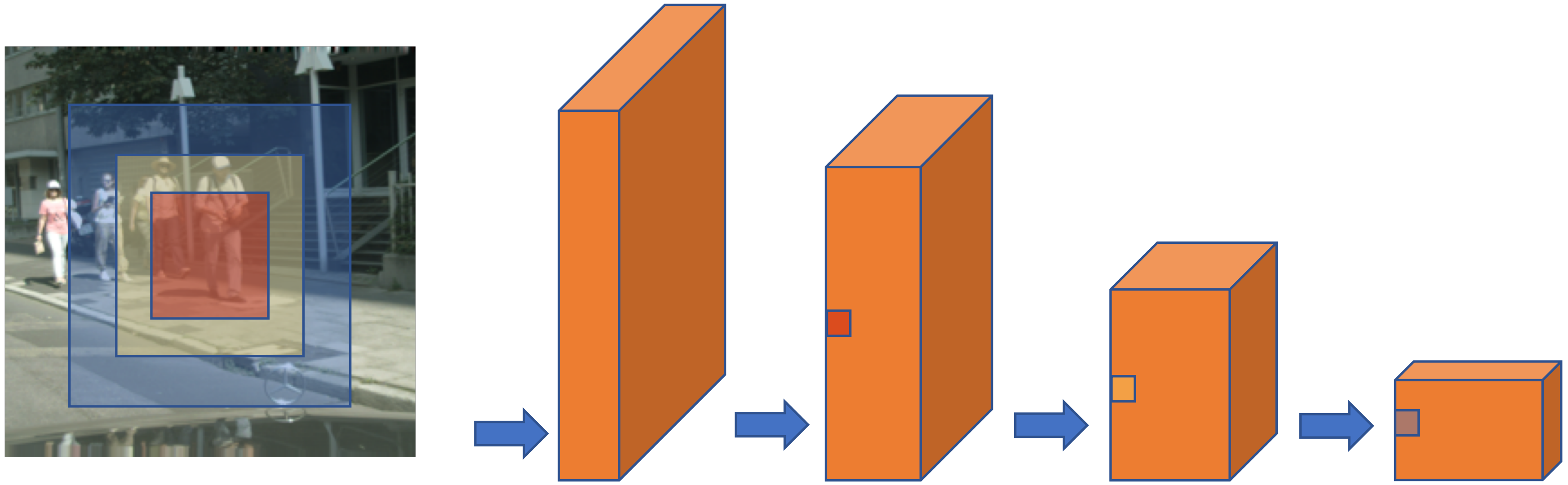


- Analysis should be based on the input image.
- The new question “In the input image which pattern caused the activation in a given neuron in an intermediate layer?”

Computer Vision

Visualization

Size of the pattern in the input image

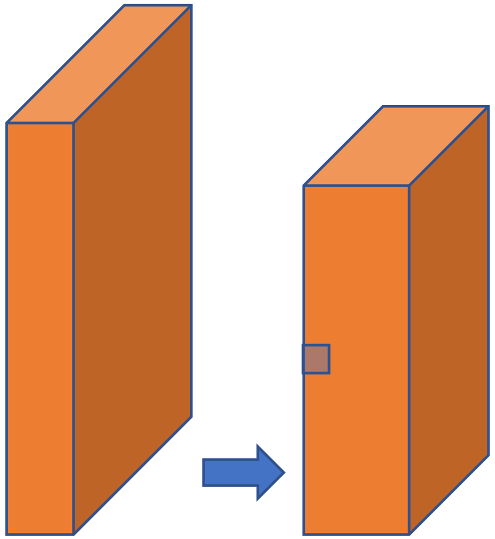


- Size of the input pattern changes with respect to the receptive field
- Depending on the layer the neuron sits, its receptive field changes
- The size of the input pattern changes as well.
- The image patch that activates the blue neuron is larger than the one that activates the red neuron

Computer Vision

Visualization

Operations in the forward pass



- Consider the operations we need to do in order to compute the activation in the blue neuron from the layer below
- Three operations
  - Convolutions with a set of filters

$$a_{l,k}^{(x,y)} = \left[\sum_j w_{l,kj} * h_{l-1,j} + b_{l,k} \right]_{(x,y)}$$

  - Non-linearity with ReLu function

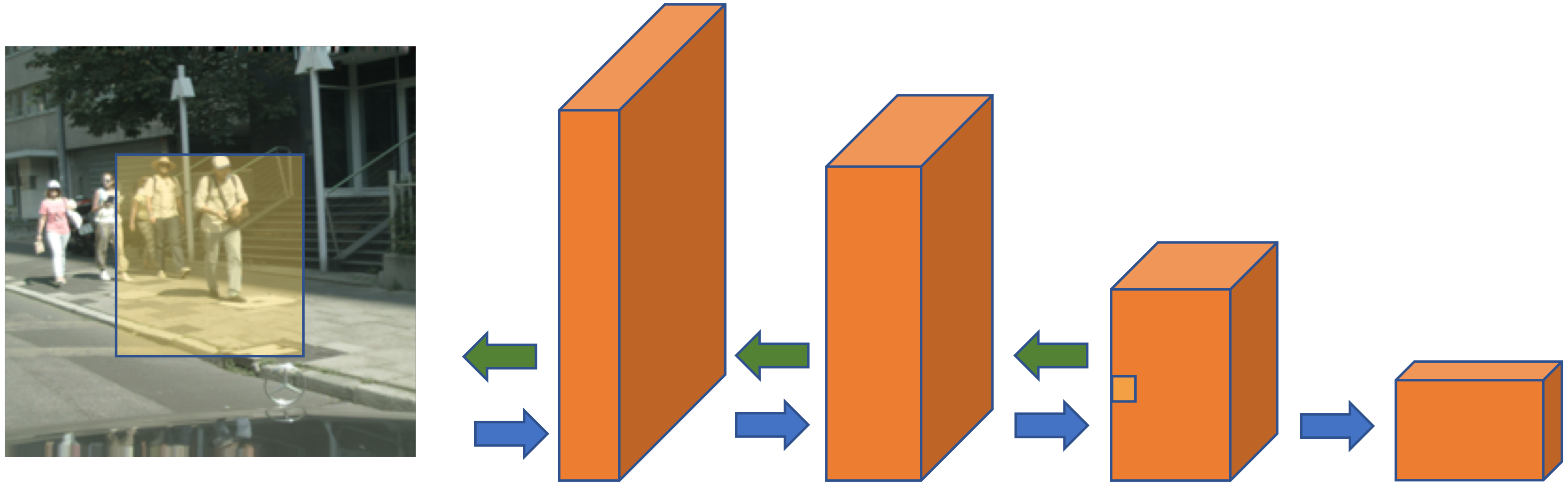
$$h_{l,k}^{(x,y)} = \sigma\left(a_{l,k}^{(x,y)}\right)$$

  - Max-pooling

Computer Vision

Visualization

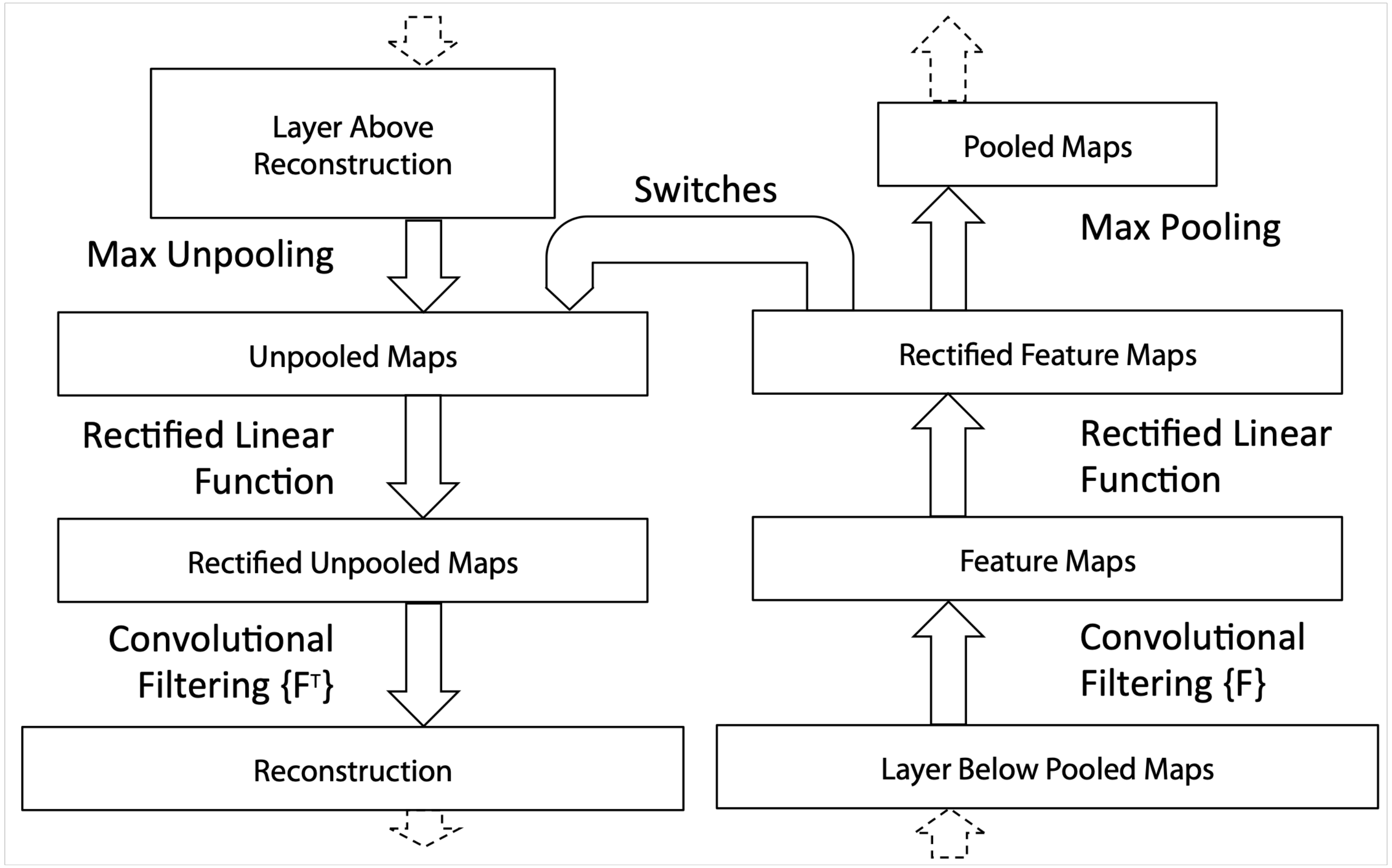
The idea



- Run an input image forward and compute all the features
- Keep the activation of the neuron you want and set the rest to 0.
- Starting from the same layer run the operations in reverse order.

- Unpool
- Rectify
- Transposed convolution
- A linked reverse ”deconvolutional” network
- Modified layers are the inputs

The idea

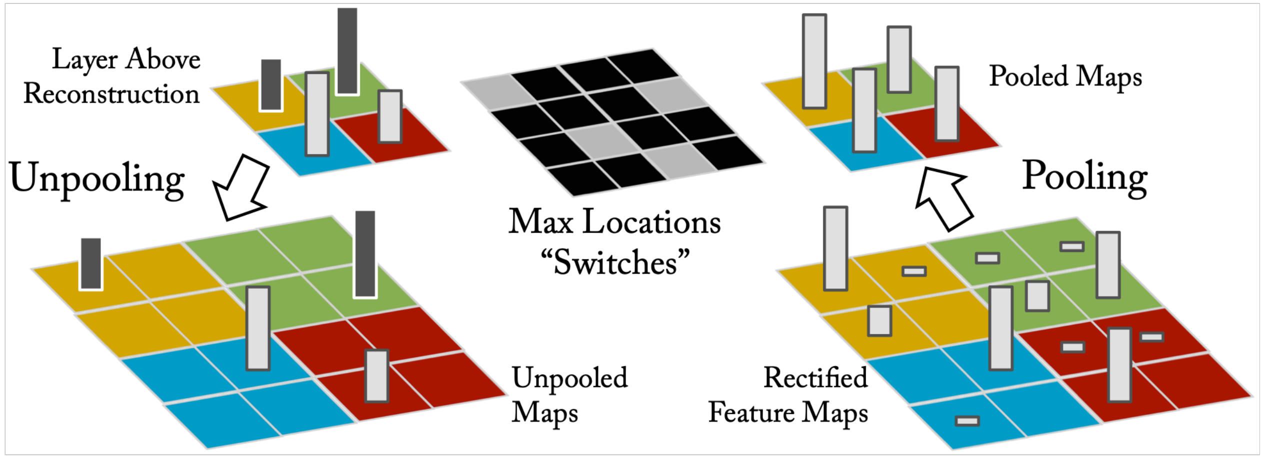


[Image from Zeiler and Fergus 2013]

Computer Vision

Visualization

Inverting the operations – max pooling



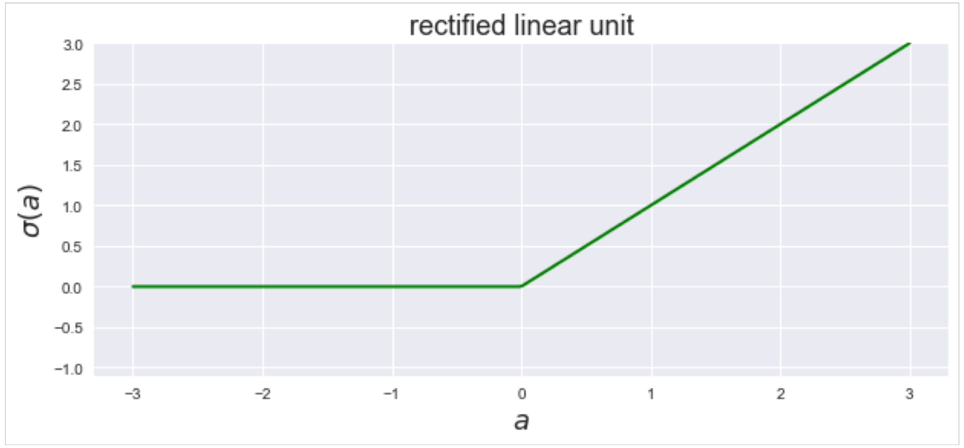
[Image from Zeiler and Fergus 2013]

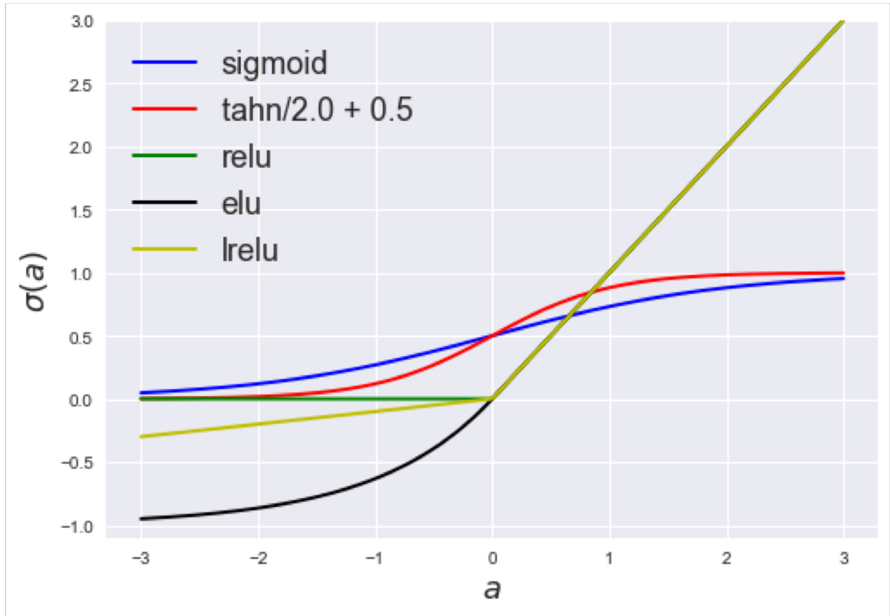
- Keep the max locations while forward passing the image
- While *unpooling* place the values to the respective positions
- Pooling leads to information loss
- It is not possible to regenerate this information
- Instead zero values are placed for the locations where activations are discarded during forward pass

Computer Vision

Visualization

Inverting the operations - ReLu



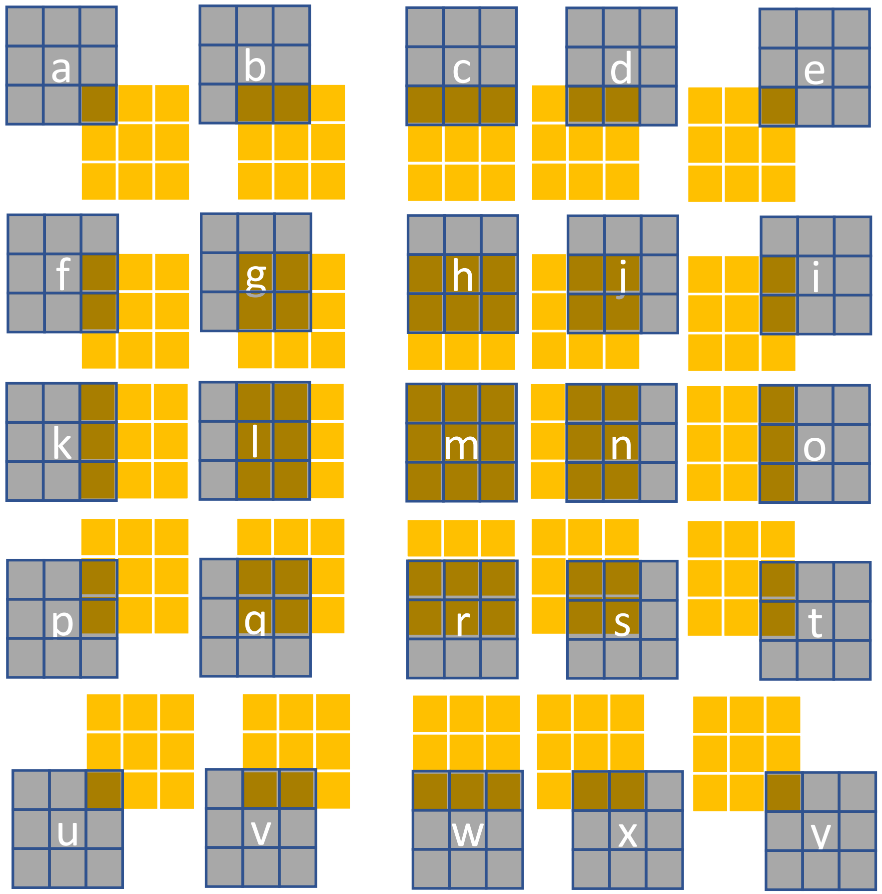


- Only keeps the positive layers
- As the reverse the same function is used
- ReLu yields only positive activation maps
- To keep the activation maps the same, ReLu is used again to keep the activations during reconstruction positive
- You can in theory, also use the inverse of a function

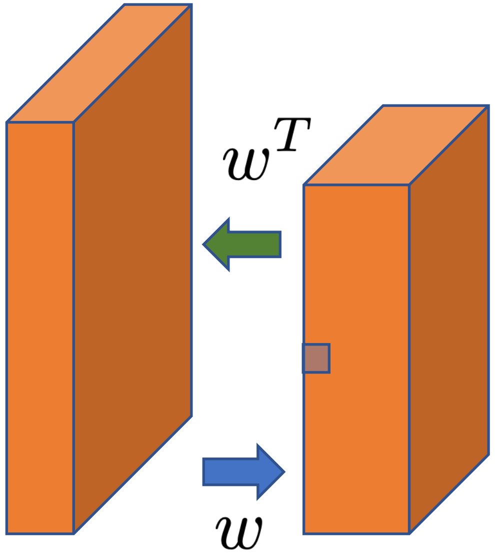
Computer Vision

Visualization

Inverting the operations - Convolution

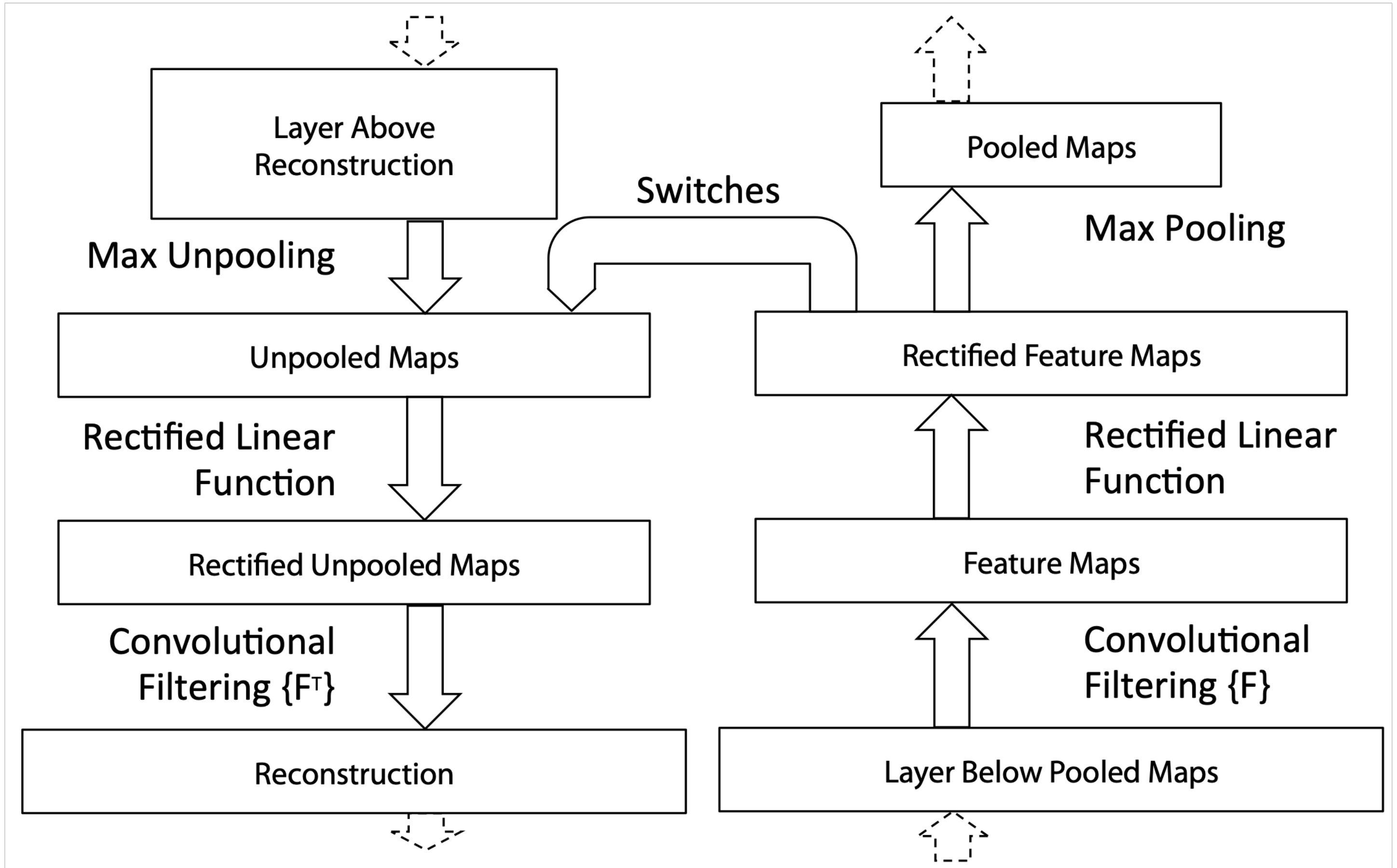


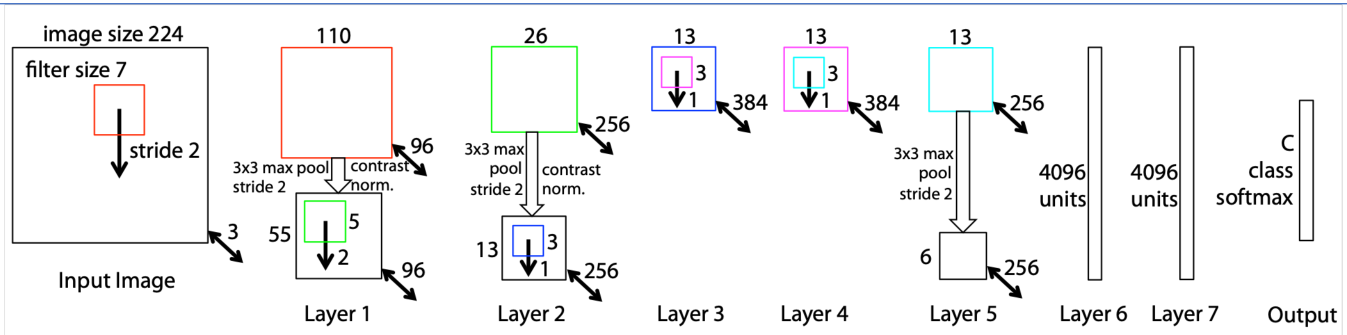
- Transposed convolution
- The kernel is the kernel used in the forward pass, flipped horizontally and vertically



Computer Vision

Visualization



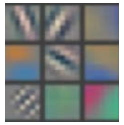


[Images from Zeiler and Fergus 2013]

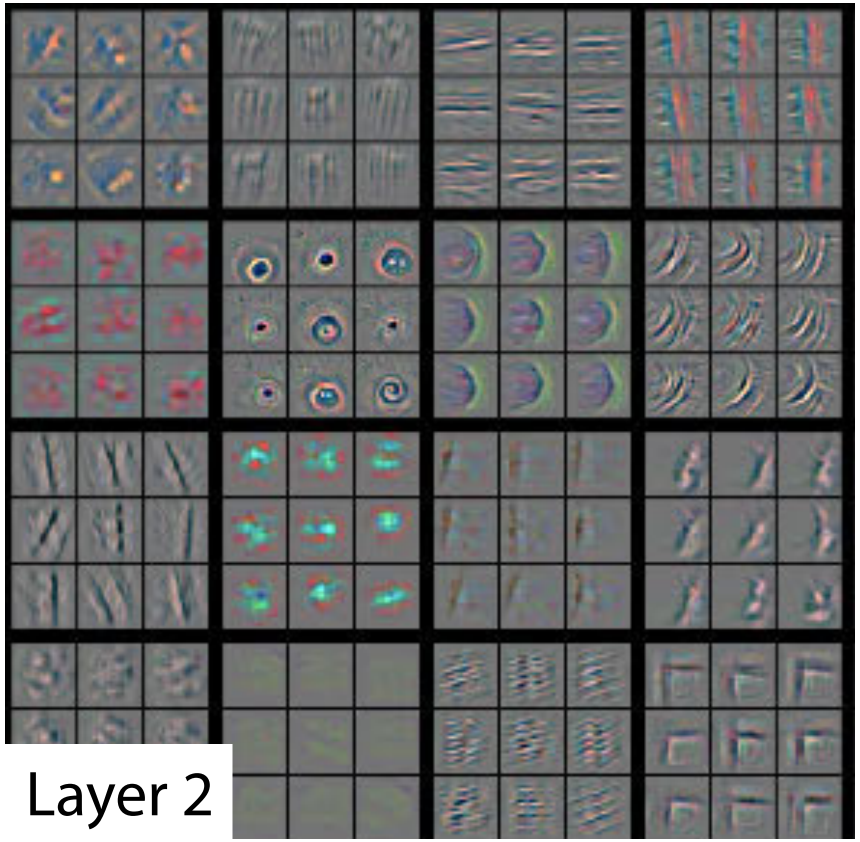
Computer Vision

Visualization

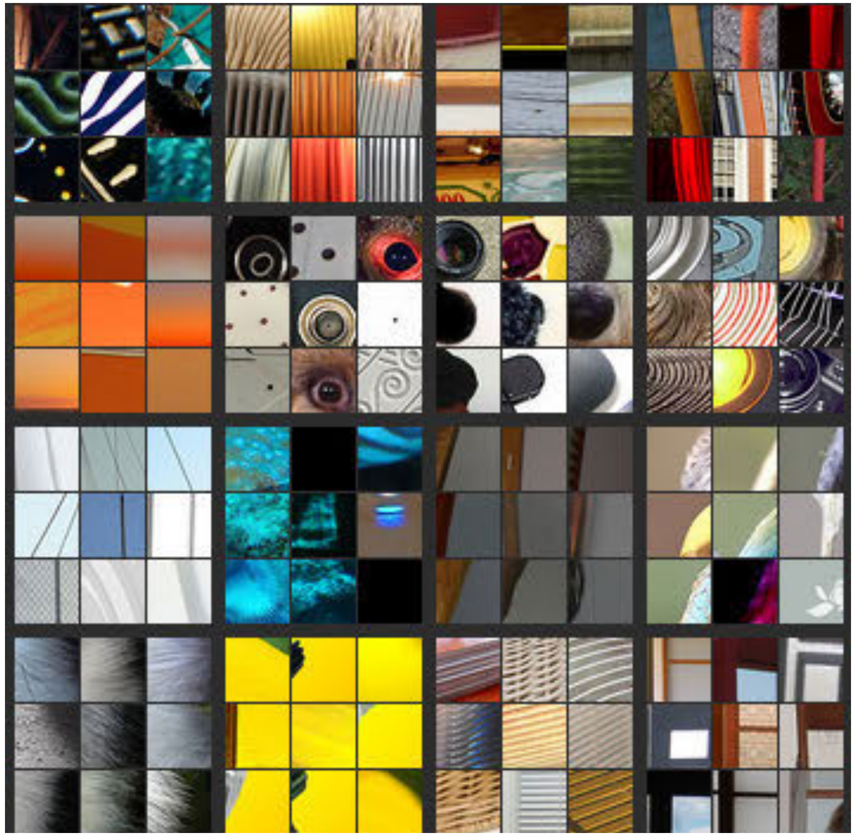
Feature maps



Layer 1

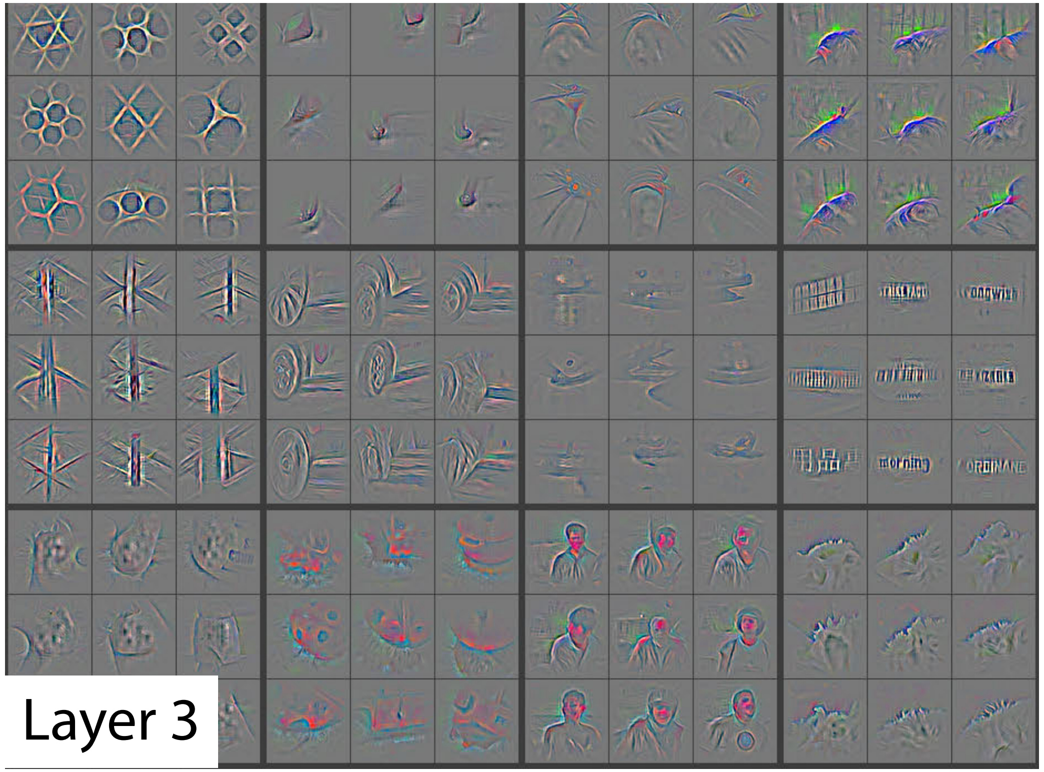


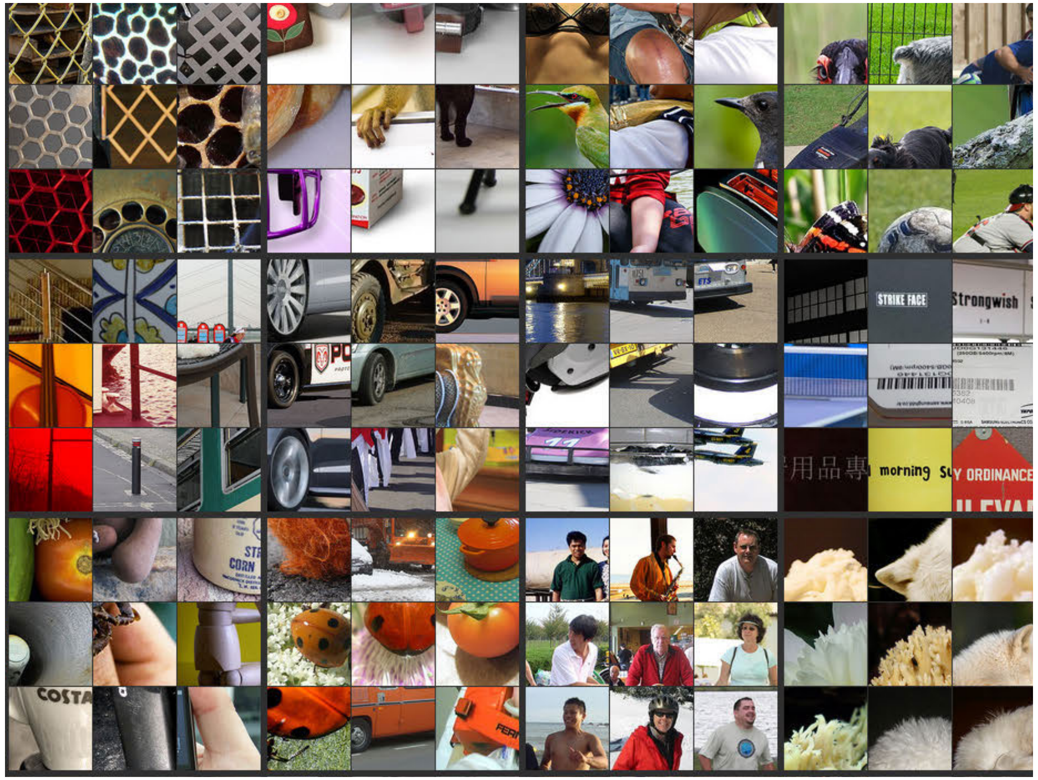
Layer 2



[Images from Zeiler and Fergus 2013]

Further deeper features





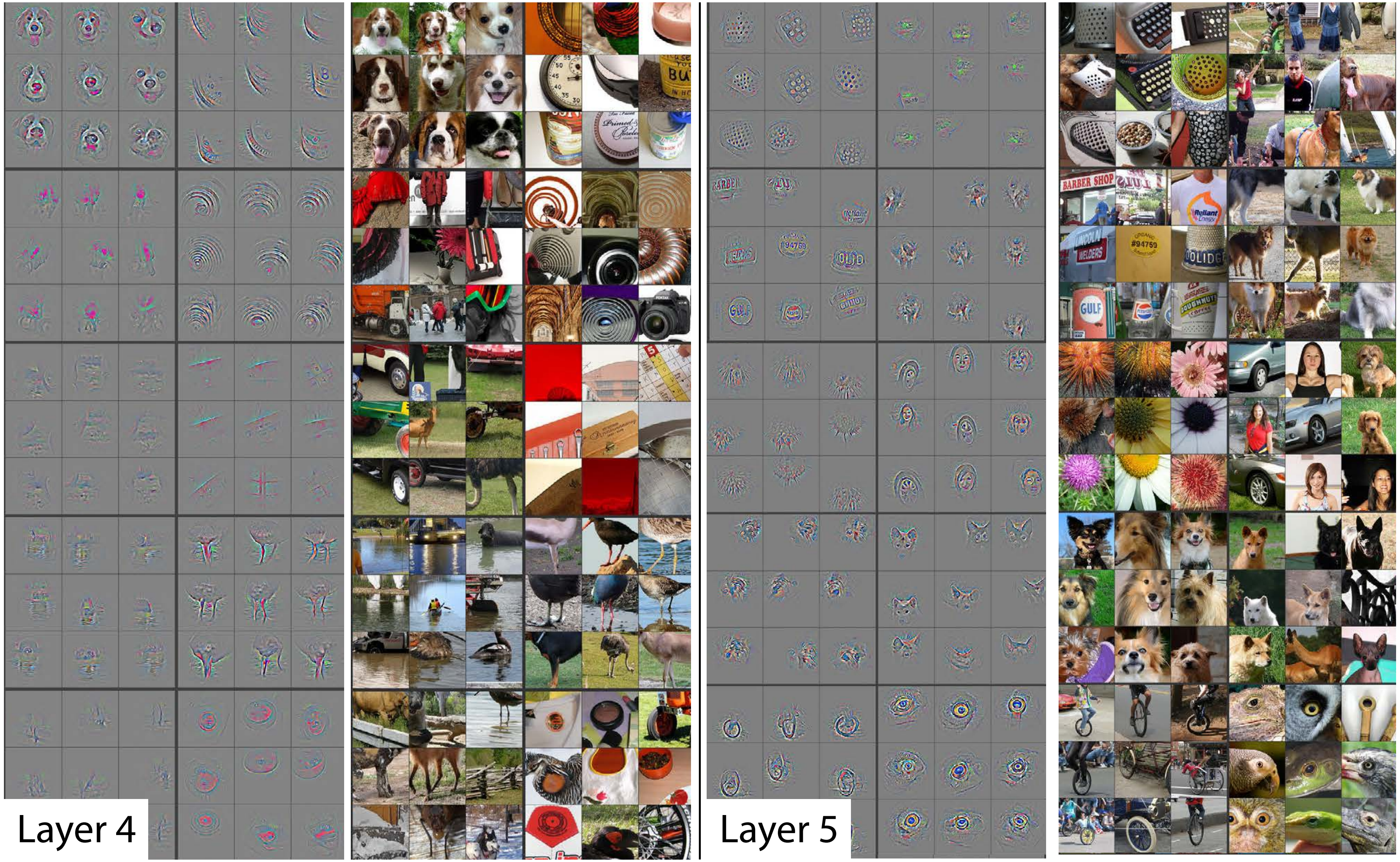
[Images from Zeiler and Fergus 2013]

Computer Vision

Visualization

[Images from Zeiler and Fergus 2013]

Even further deeper



Localization and classification

Computer Vision

Visualization

Combined Localization

Class activation maps

- So far for classification we were only interested in determining the class assignment
- We also had a separate localization network that relied on separate classification tasks at proposal regions
- With slight modifications classification networks can identify approximate locations
- Based on global average pooling idea
- Discussion based on [Zhou et al. 2016]

Learning Deep Features for Discriminative Localization

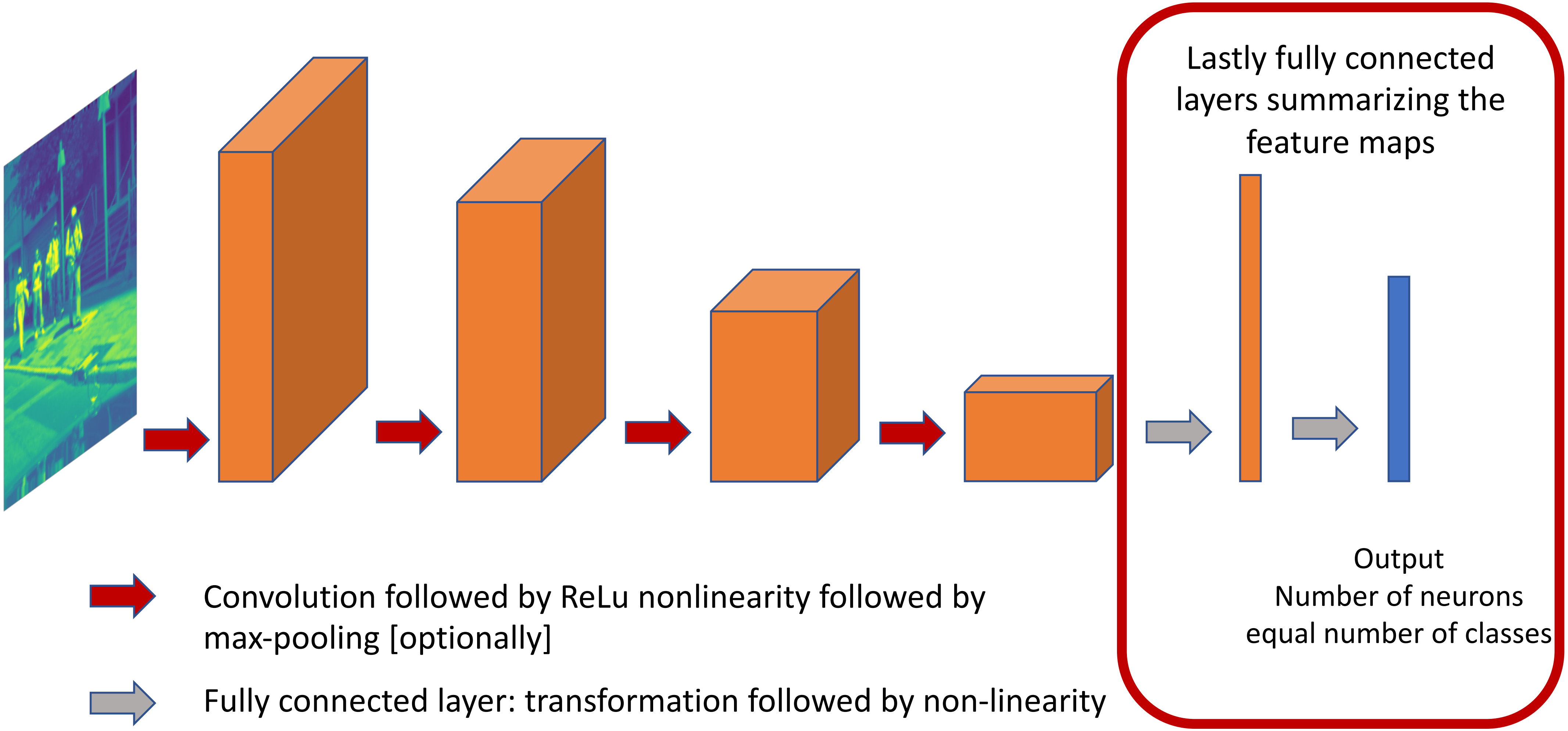
Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba
Computer Science and Artificial Intelligence Laboratory, MIT
{bzhou,khosla,agata,oliva,torralba}@csail.mit.edu

Computer Vision

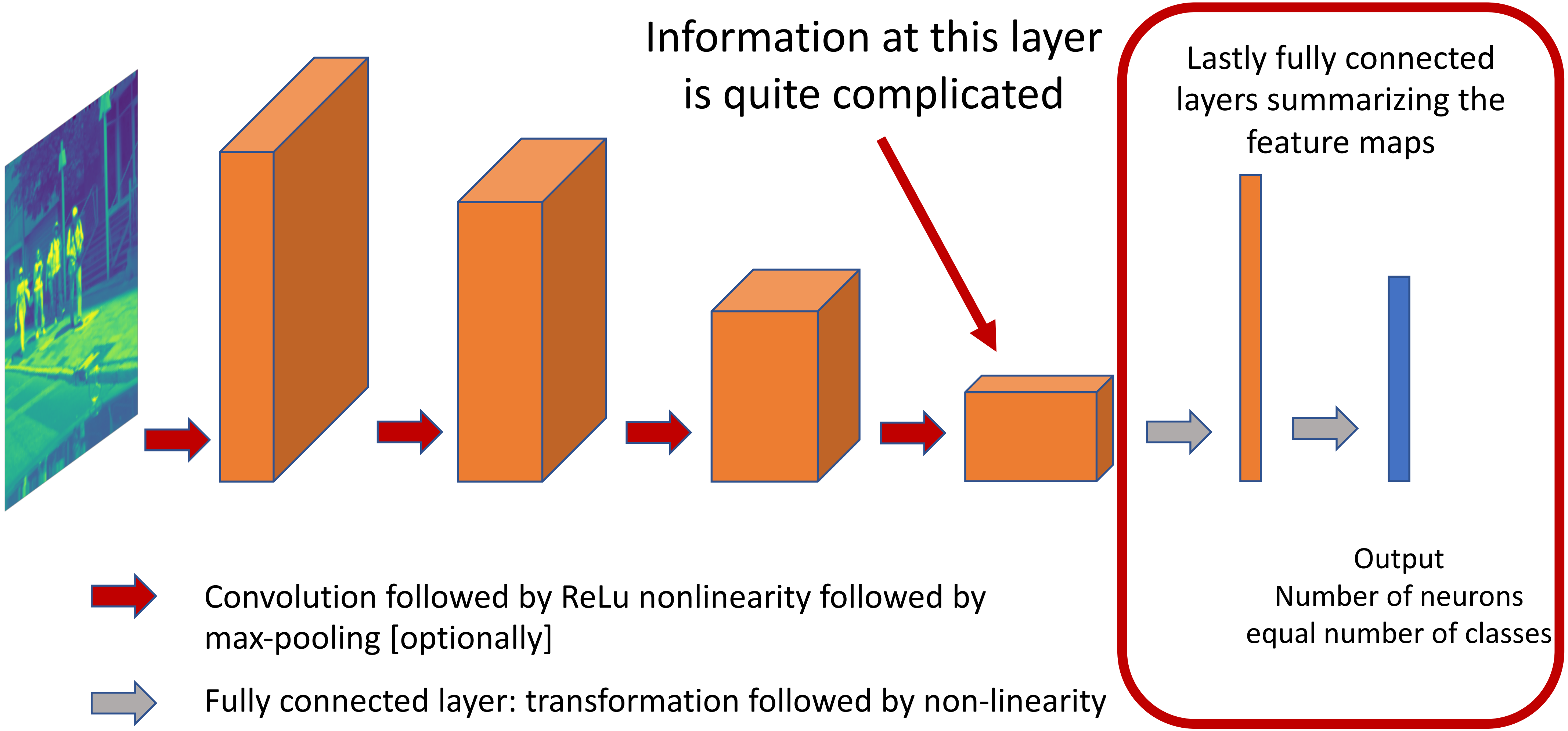
Visualization

Combined Localization

Normal classification network



Normal classification network

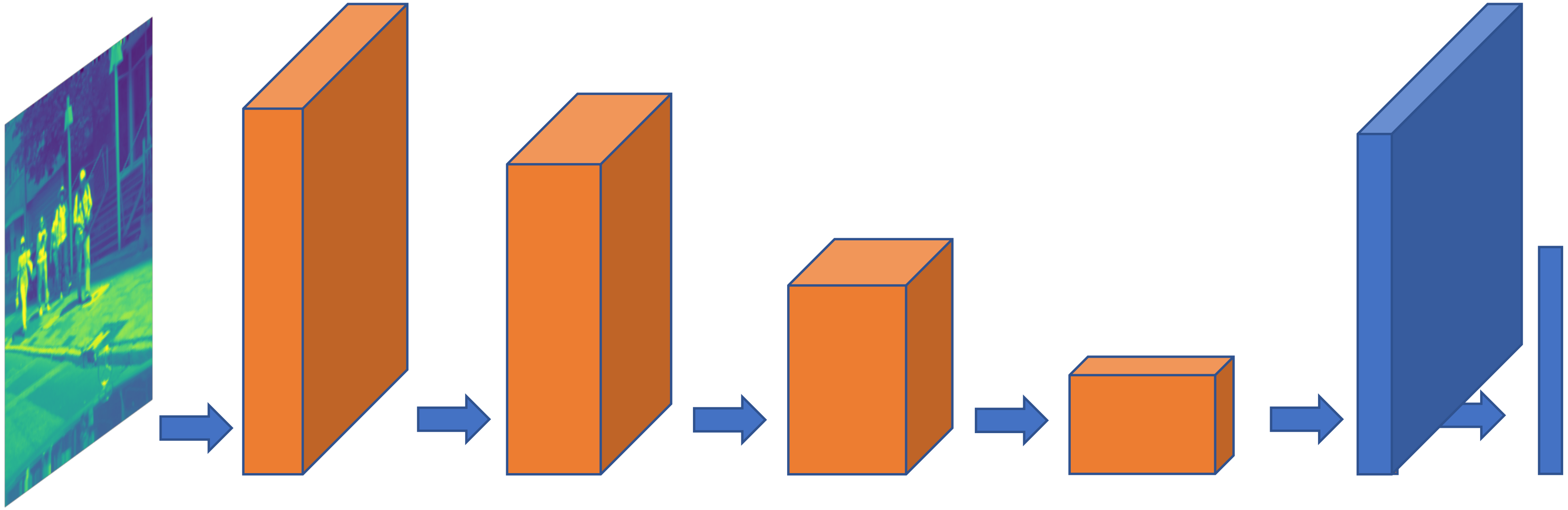


Computer Vision

Visualization

Combined Localization

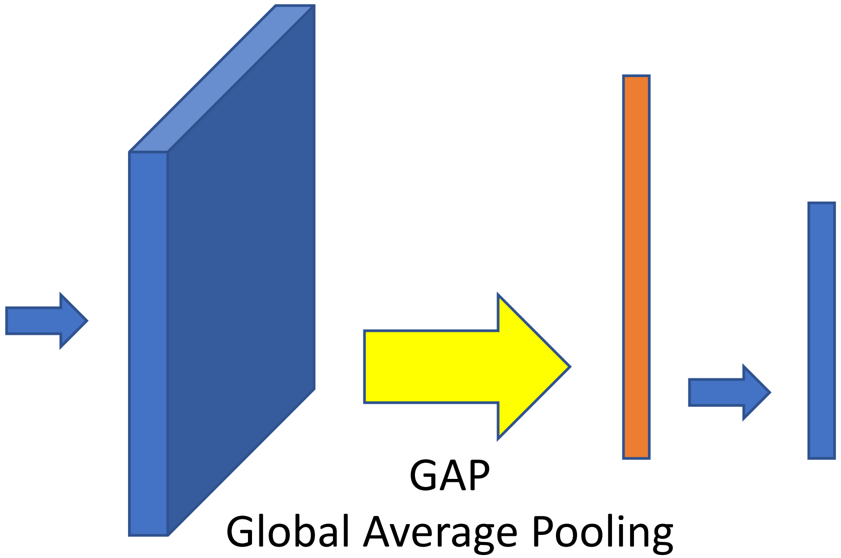
We have also seen fully convolutional networks for segmentation



- Image size outputs
- Replaced the final fully connected layers
- Upsampling using transpose convolutions or bilinear upsampling followed by convolutions

Combining these ideas

- Class activation maps combines these ideas
- Using Global Average Pooling



- Like normal pooling, applies to each channel in a layer separately

$$h_{l,k} = \frac{1}{M_{l-1}N_{l-1}} \sum_{x,y} h_{l-1,k}^{(x,y)}$$

- Averaging all the information to a single number!
- Then continue as usual

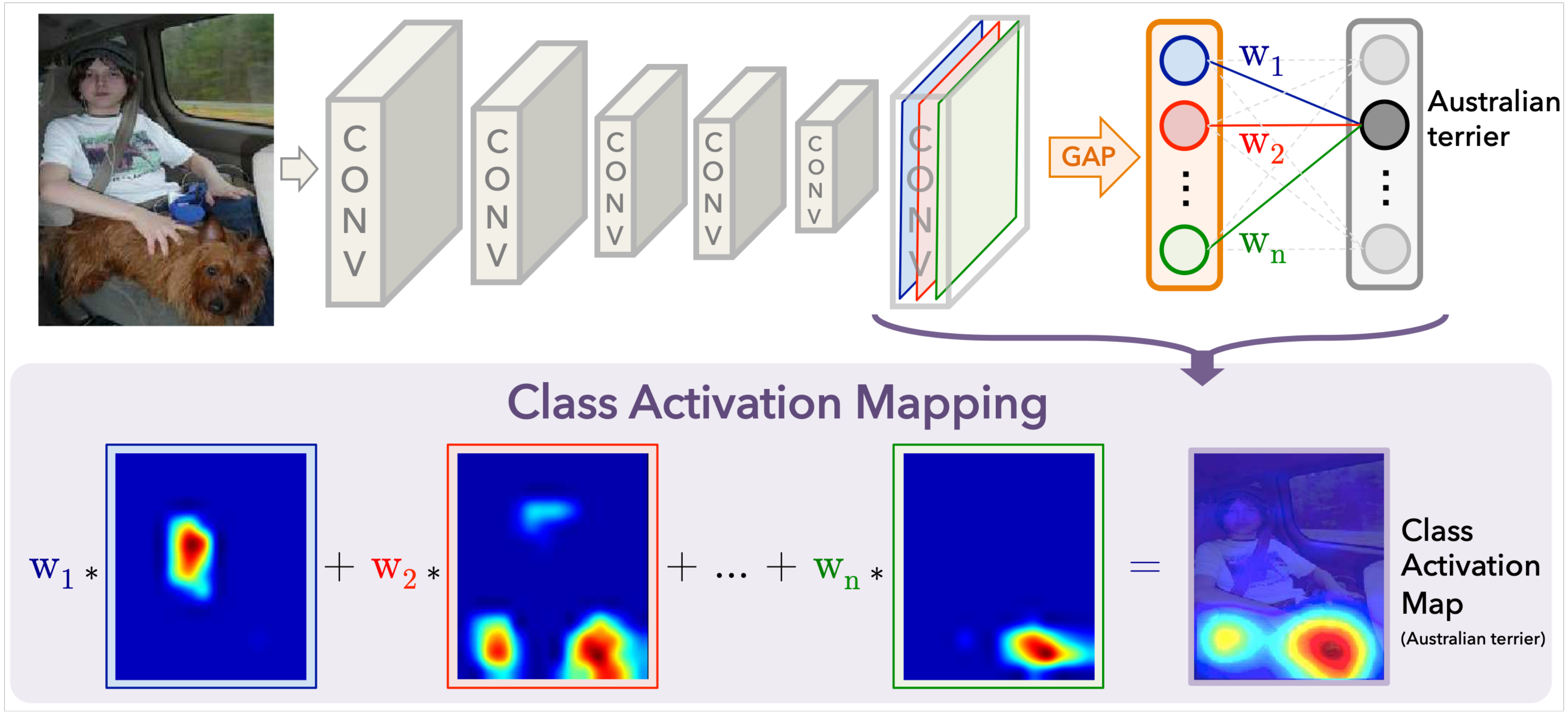
$$a_{L,j} = \sum_k w_{L,jk} h_{l,k}$$

Computer Vision

Visualization

Combined Localization

Activation Maps



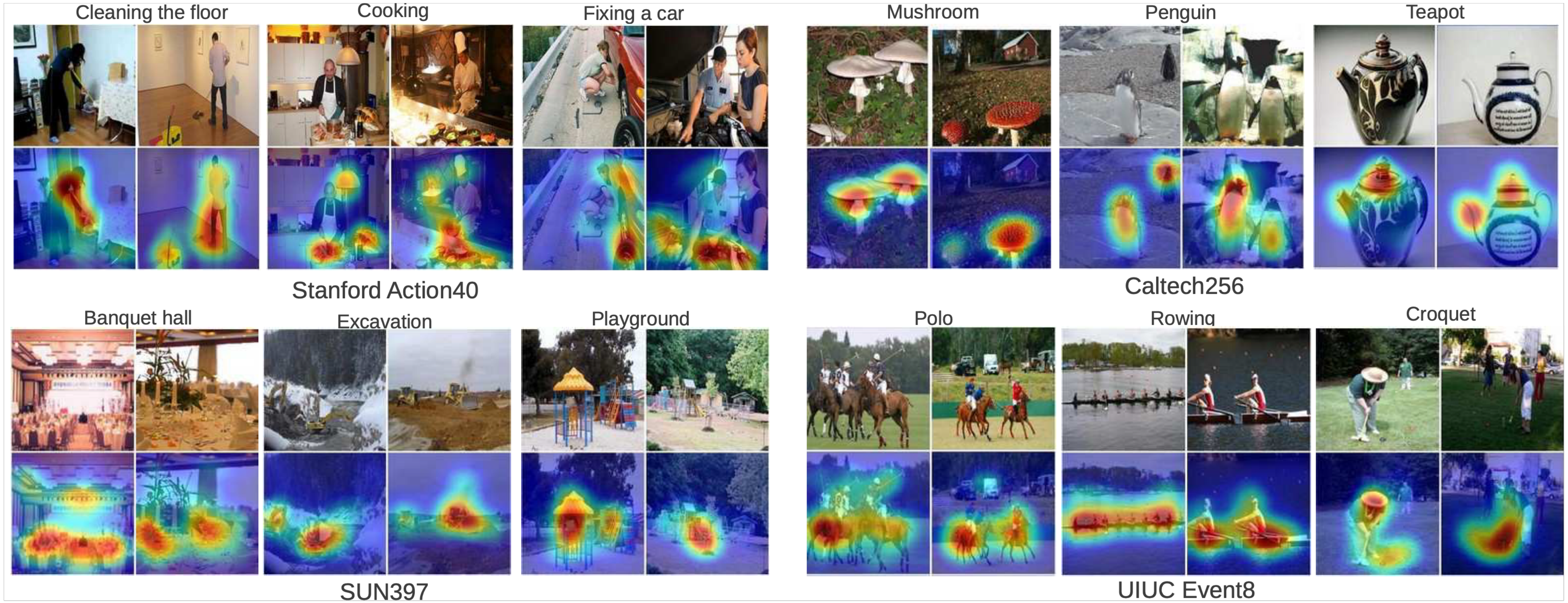
- Per-class weighted sum of all the channels before global average pooling yields the class-specific activation map

[Image taken from Zhou et al. 2016]

Computer Vision

Visualization

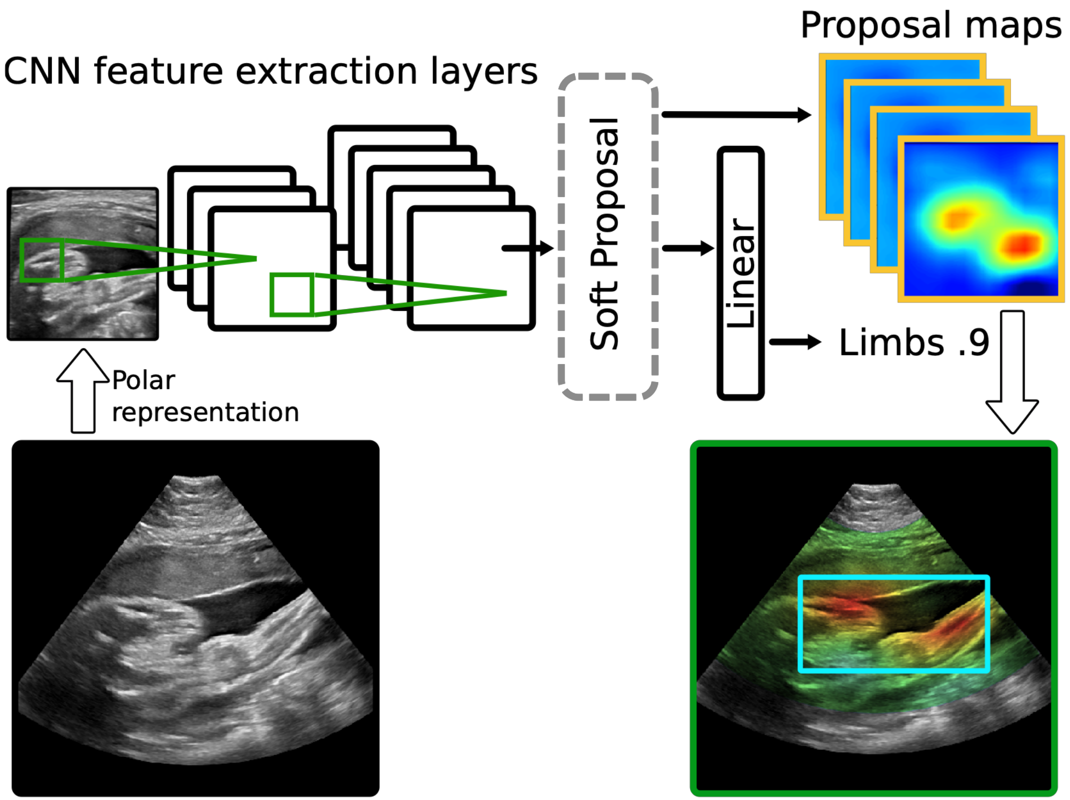
Combined Localization



- Network architecture preceding the GAP layer can change
- Form of weak-supervision for localization

[Image taken from Zhou et al. 2016]

Various applications



Weakly Supervised Localisation for Fetal Ultrasound Images

Nicolas Toussaint[1], Bishesh Khanal[1,2], Matthew Sinclair[2], Alberto Gomez[1], Emily Skelton[1], Jacqueline Matthew[1], and Julia A. Schnabel[1]

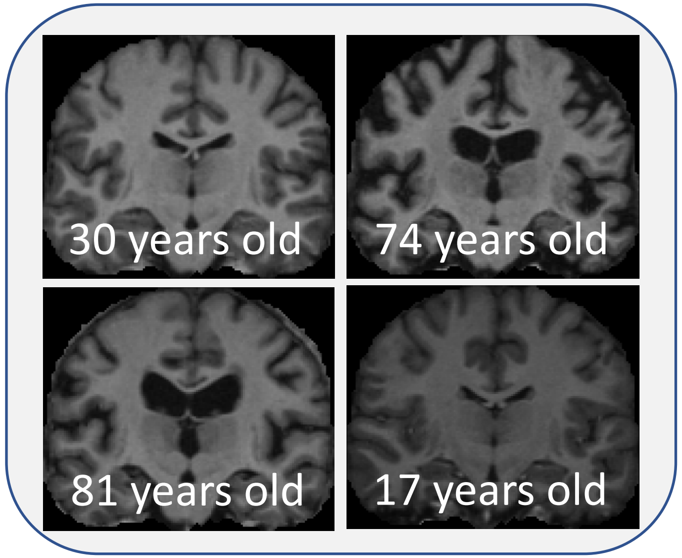
- Especially in medical imaging
- Labels are expensive and difficult to get.
- Approximate localization with CAM allow identifying areas of interest
- Also weak supervision to train stronger localization algorithms

Unsupervised learning

Very coarse view on supervised learning

Supervised learning

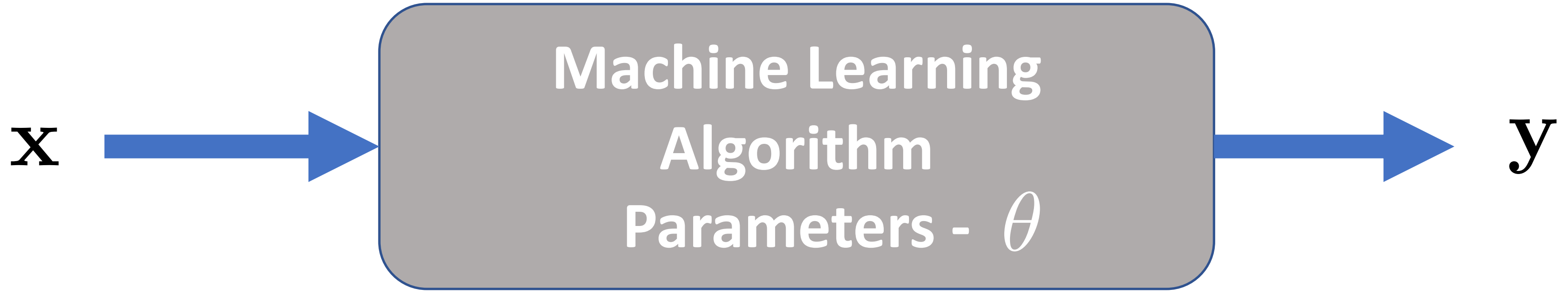
- Patterns between two types of data
- Goal: predicting one from the other
- Examples have both types of data
- At prediction only one exist



30 years old

General idea in the supervised approach

- Algorithms assume a mathematical model between features and labels



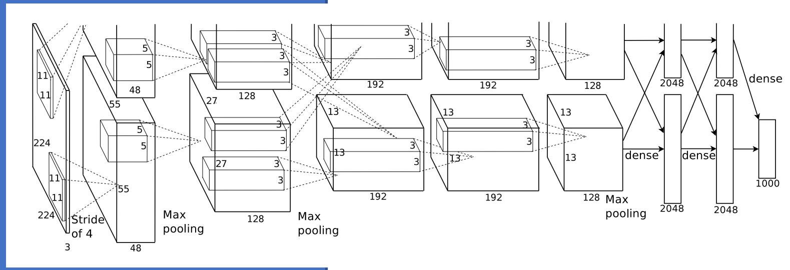
- Estimate the parameters of the model to best predict labels from features in the training examples

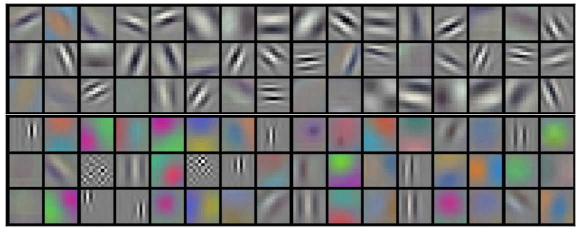
$$y = f(\mathbf{x}|\theta)$$

Unsupervised learning

Unsupervised learning of features

- Filters are important for performing image analysis tasks
- So far, we determine features in a supervised way, task-specific manner

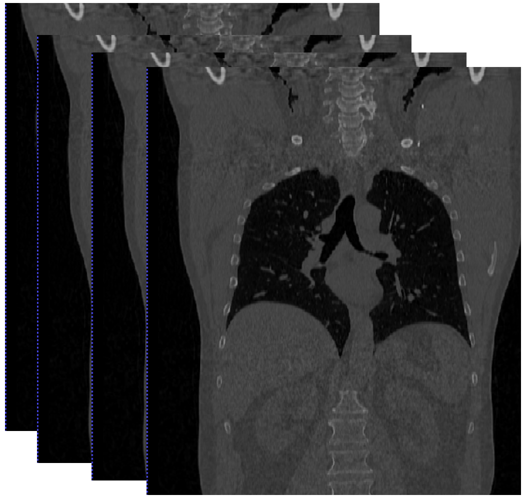




- Determine features in an unsupervised manner
- Examples have only features

Unsupervised learning of distributions

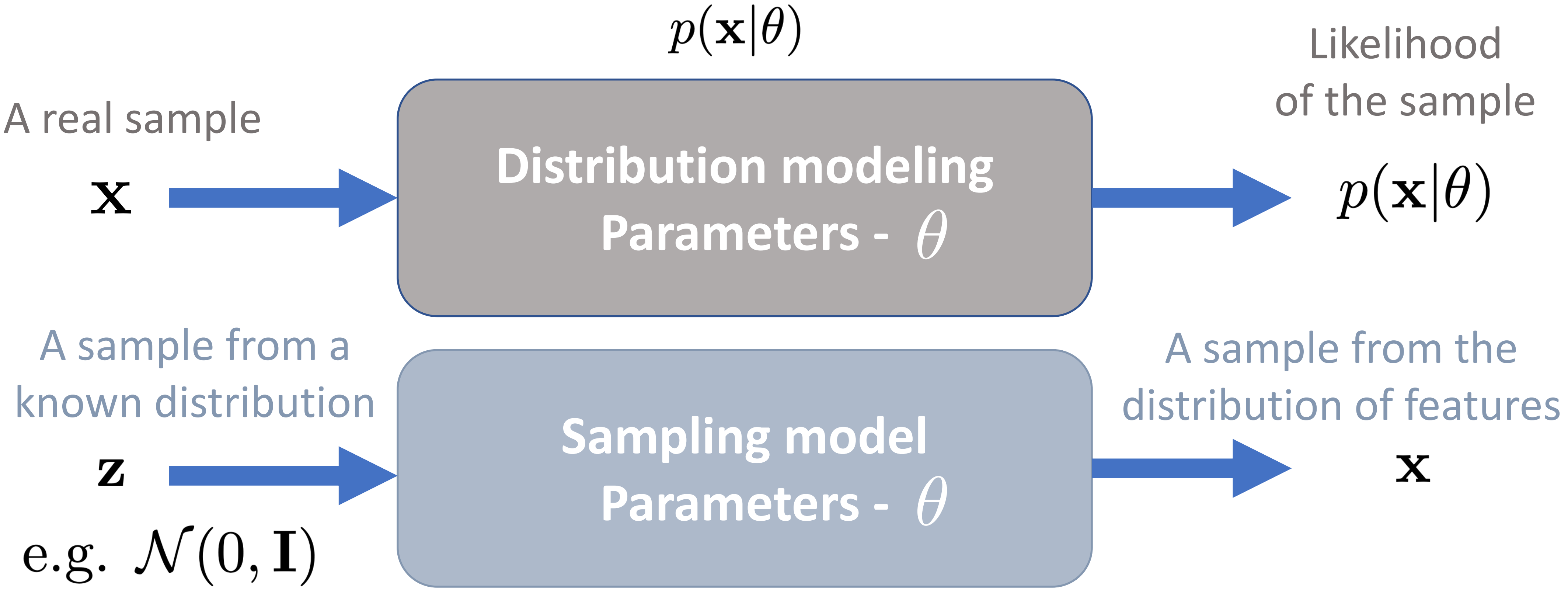
- Patterns within the data
- Goal: describe variability in the data
- Estimate the distribution of the data
- There is still a training dataset
- Examples have only features



Both are unsupervised in the sense that there are no labels!

General idea in unsupervised distribution learning

- Algorithm assumes a mathematical model for the features
- Ideally this is the probability distribution of features



Why is this useful?

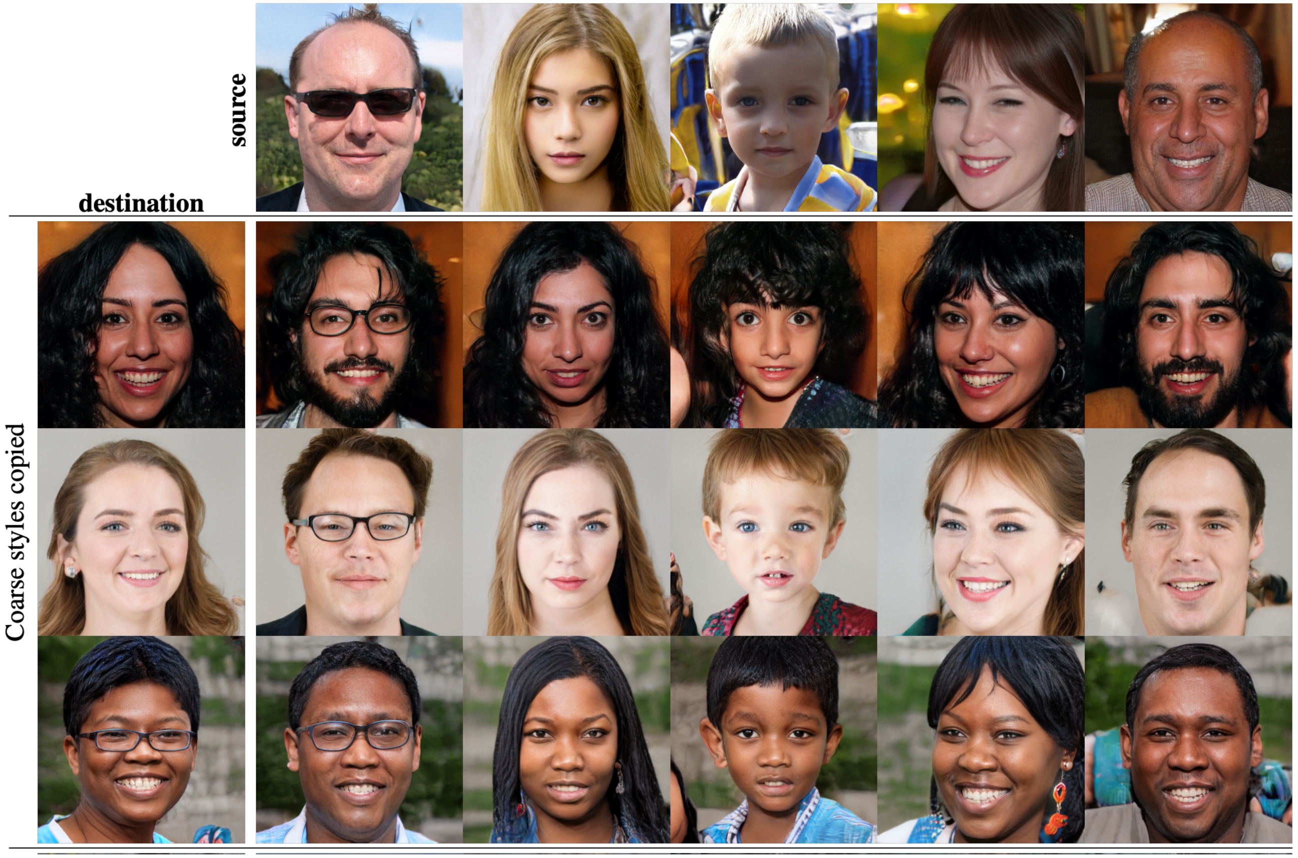
- Sample from the distribution of image to generate images

Figure 1: Class-conditional samples generated by our model.

[Figure from Brock, Donahue and Simonyan 2018 – Class conditional generation of images]

Why is this useful?

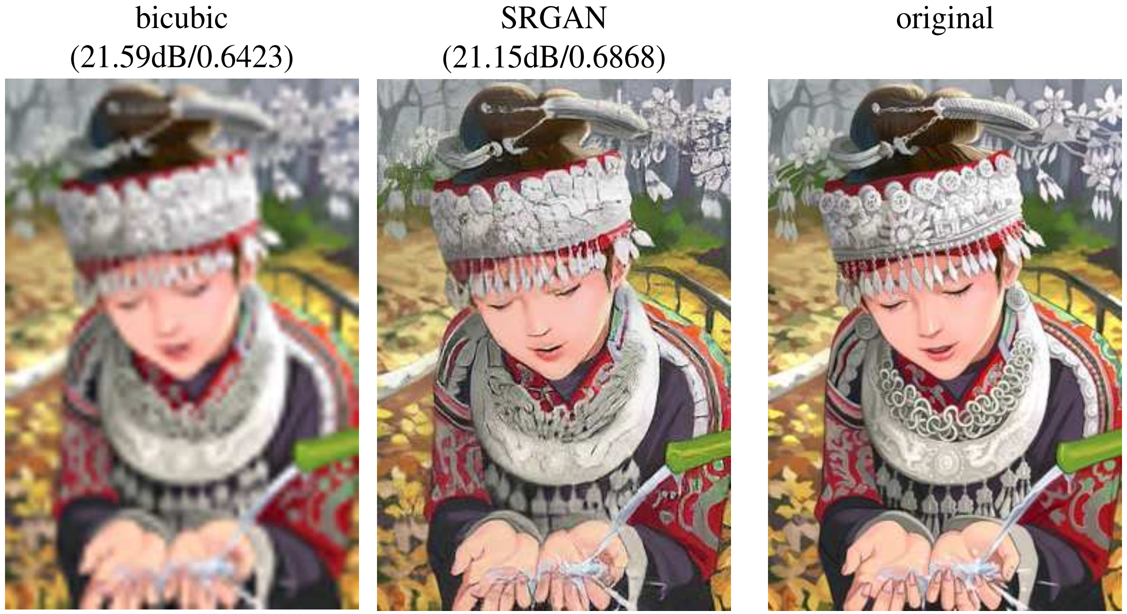
- Style transfer



[Figure from Karras, Laine and Aila, 2018]

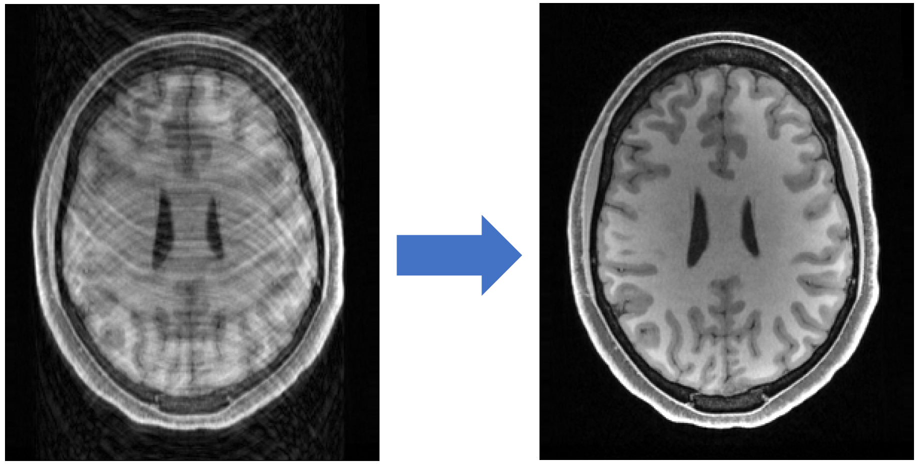
Why is this useful?

Improving resolution of an image



[Figure from Ledig et al. 2017]

Bayesian reconstruction of medical images



[Figure from Tezcan et al. 2018]

Why is this useful?

- Many more applications:
  - In-painting
  - Realistic video and image editing
  - Video frame prediction
  - Outlier detection
  - …
- Scientifically
  - Building a model of the visual world
  - Possibly an important component in human learning.
    - We do not see 100s of cups to understand what a cup is
    - We constantly observe around and get visual input to our brains.

Images are big

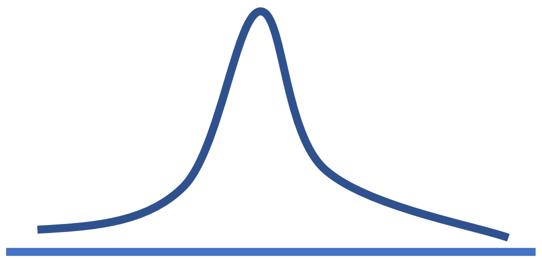
- Images are very high dimensional
- Consider a small image of 64x64
- Even that is 4096 dimensional!
- We need to keep that in mind when we think about unsupervised learning.

The most straightforward way

- Kernel density estimation (KDE)
- Given a sample set of images the naïve way is

$$p(x|\theta) = \frac{1}{N}\sum_n K_\theta(x, x_n)$$

$$\int K_\theta(x, x_n)dx = 1$$



- Place a ”kernel” around each training sample
- Determine the likelihood of a new sample based on these kernels
- If kernels depends on Euclidean distance, e.g. Gaussian kernel, then likelihood is related to the distance in Euclidean space.

Computer Vision

Visualization

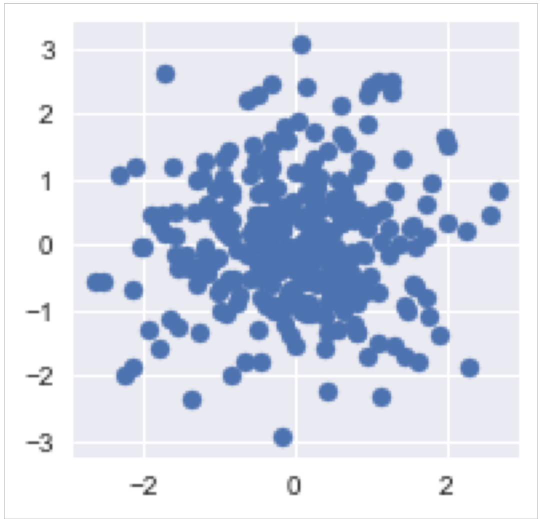
Combined Localization

Unsupervised Learning

Distribution Learning

Bad idea due to the dimensions

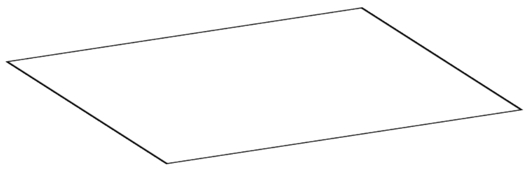
- For the KDE to work, roughly speaking, you need to somehow “fill” the space, e.g.



- To fill a space of 4096 dimensions, you need a lot of samples, we need to find a better solution.

Latent variable models

- Assume that images live in a lower dimensional sub-space
- We build a mapping between them



latent space

$z \in \mathbb{R}^d$

$\xrightarrow{\quad}$ $p(x|z)$

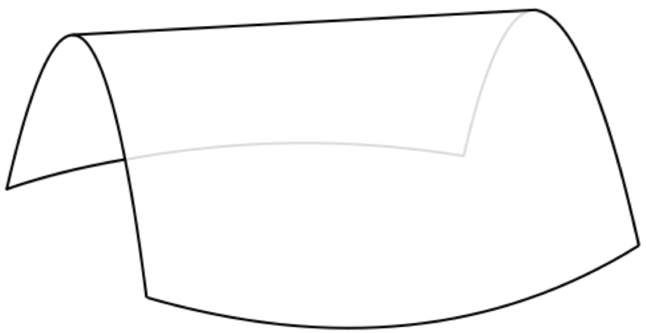


image space

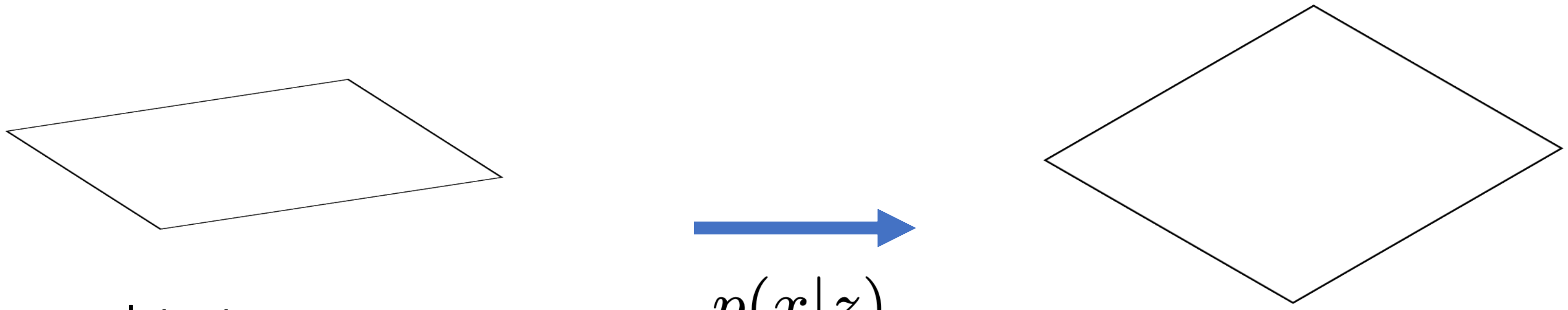
$x \in \mathcal{M} \subset \mathbb{R}^D$

$$p(x) = \int p(x|z)p(z)dz$$

$$x \in \mathbb{R}^D,\ z \in \mathbb{R}^d,\ d << D$$

Probabilistic principal component analysis

- Assumes the mapping is a linear one
- Probabilistic principal component analysis [Tipping & Bishop 1999]



latent space

$z \in \mathbb{R}^d$

$p(x|z)$

image space

$x \in \mathcal{M} \subset \mathbb{R}^D$

$$p(x) = \int p(x|z)p(z)dz$$

$$x \in \mathbb{R}^D,\ z \in \mathbb{R}^d,\ d << D$$

$$p(x|z) = \mathcal{N}\left(Wz + \mu_x, \sigma^2\right)$$

$$p(z) = \mathcal{N}(0, I)$$

$$W \in \mathbb{R}^{D\times d}$$

μ_x : mean image

σ^2 : noise

Link to PCA

- Maximum likelihood estimate of the parameters yield the PCA
- Eigenvalues and eigenvectors of the sample covariance matrix
- Derivation in [Tipping and Bishop 1999]

$$W_{ML} = U_d \left(\Lambda_d - \sigma^2 I\right)^{1/2} R$$

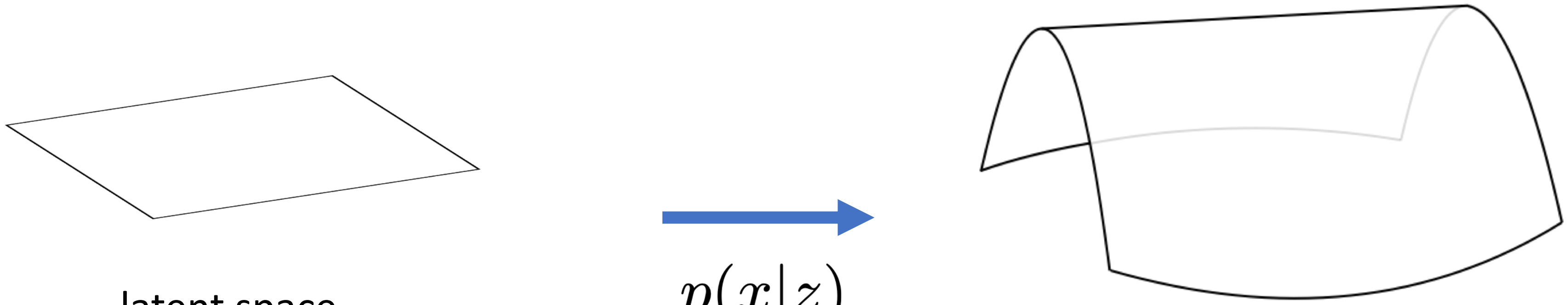
$$W_{ML} = \underbrace{U_d}_{\text{d eigenvectors}} \left(\underbrace{\Lambda_d}_{\text{d eigenvalues}} - \sigma^2 I \right)^{1/2} \underbrace{R}_{\text{arbitrary rotation}}$$

$$\mu_x : \text{ sample mean image}$$

$$\sigma^2_{ML} = \frac{1}{D-d} \sum_{q=d+1}^{D} \lambda_q$$

Non-linear maps

- In supervised learning linear maps were not enough
- The same idea applies here



latent space

$z \in \mathbb{R}^d$

$p(x|z)$

image space

$x \in \mathcal{M} \subset \mathbb{R}^D$

$$p(x) = \int p(x|z)p(z)dz$$

$$x \in \mathbb{R}^D, \; z \in \mathbb{R}^d, \; d << D$$

$$\boxed{p(x|z) = \mathcal{N}\left(f(z|\theta) + \mu_x, \sigma^2\right)}$$

$$p(z) = \mathcal{N}(0, I)$$

μ_x : mean image

σ^2 : noise

Computer Vision

Visualization

Combined Localization

Unsupervised Learning

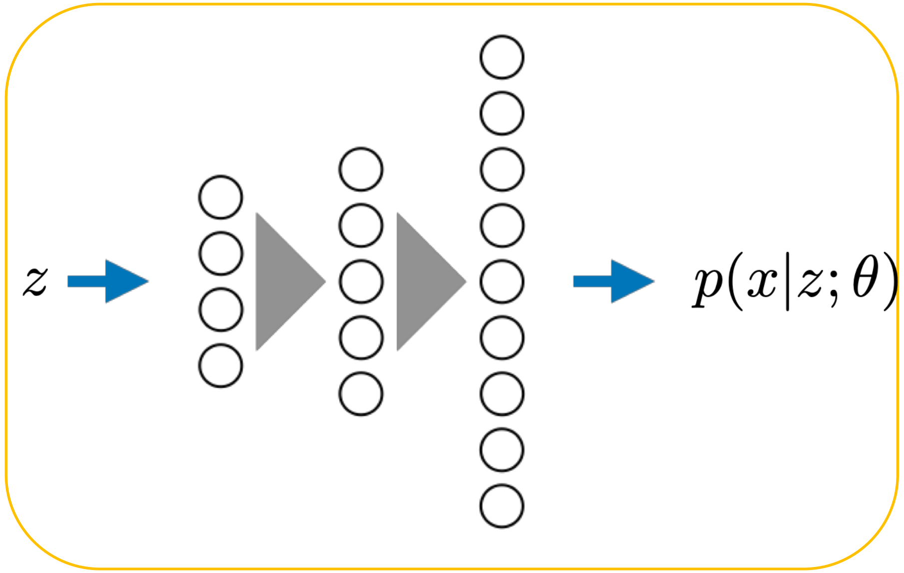
Distribution Learning

Density networks

[MacKay, Nucl. Inst. Met. In Physics Research 1995]

$p(x|z;\theta)$: Parameterize with a network with parameters $\boldsymbol{\theta}$

$$p(x;\theta) = \int p(x|z;\theta)p(z)dz$$



For the given samples, maximize with respect to $\boldsymbol{\theta}$

$$\prod_n p(x_n;\theta)$$

$$= \prod_n \int p(x_n|z;\theta)p(z)dz$$

using Monte Carlo integration

$$\sum_n \ln \frac{1}{R} \sum_r p(x_n|z_r;\theta), \;\; z_r \sim p(z)$$

Sampling was not efficient for very large dimensional problems, need too many samples

MacKay hinted importance sampling

Computer Vision

Visualization

Combined Localization

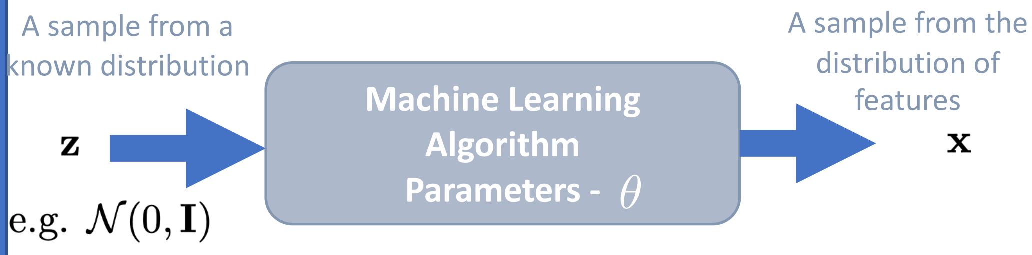
Unsupervised Learning

Distribution Learning

Two avenues – both end of 2013

Generative Adversarial Network

Sampler

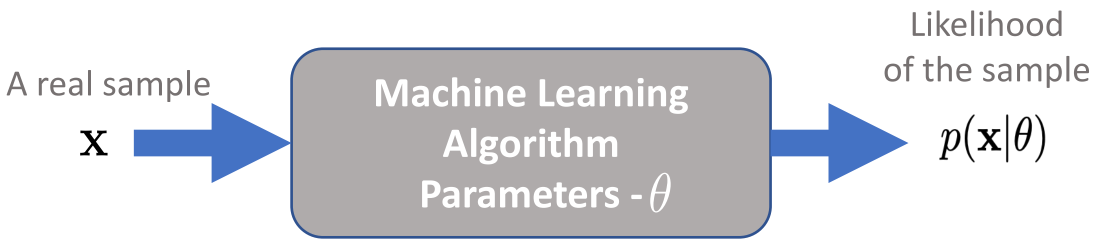


Generative Adversarial Nets

Ian J. Goodfellow,* Jean Pouget-Abadie,† Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,‡ Aaron Courville, Yoshua Bengio§
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Variational Auto-encoders

Distributional model



Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Danilo J. Rezende, Shakir Mohamed, Daan Wierstra
{danilor, shakir, daanw}@google.com
Google DeepMind, London

Computer Vision

Visualization

Combined Localization

Unsupervised Learning

Distribution Learning

Variational auto-encoders

Builds on density networks concept but instead of Monte-Carlo uses variational inference with a network parameterized sampling (approximate) distribution

$$
\begin{aligned}
\ln p(x;\theta) &= \ln \int p(x|z;\theta)p(z)dz \\
&= \ln \int p(x|z;\theta)p(z)\frac{q(z|x;\phi)}{q(z|x;\phi)}dz \\
&\geq \int q(z|x;\phi)\ln p(x|z;\theta)\frac{p(z)}{q(z|x;\phi)}dz \\
&= \mathbb{E}\left[\ln p(x|z;\theta)\right] - D_{KL}\left[q(z|x;\phi)\|p(z)\right]
\end{aligned}
$$

Let’s find a distribution more focused so I will sample for less for the same approximation

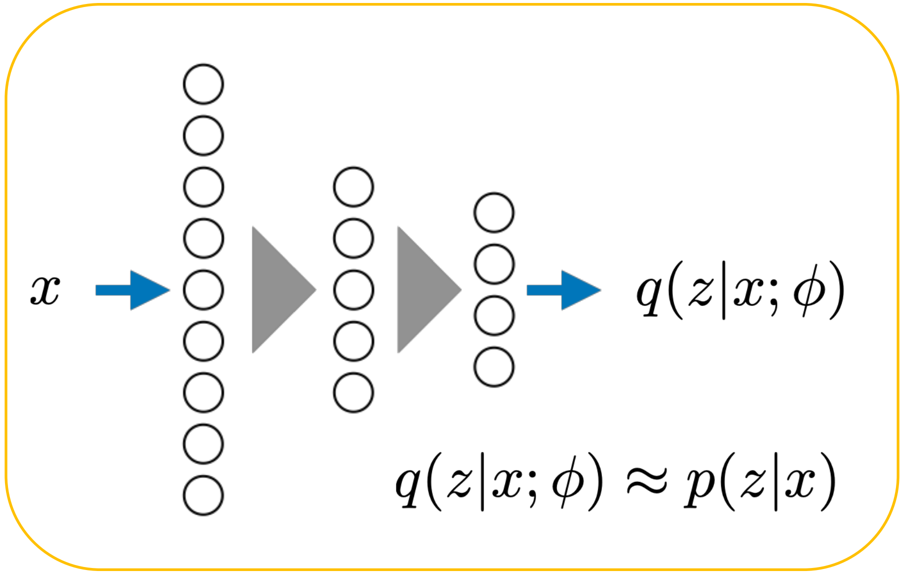
Importance Sampling

Best option is the posterior p(z|x)

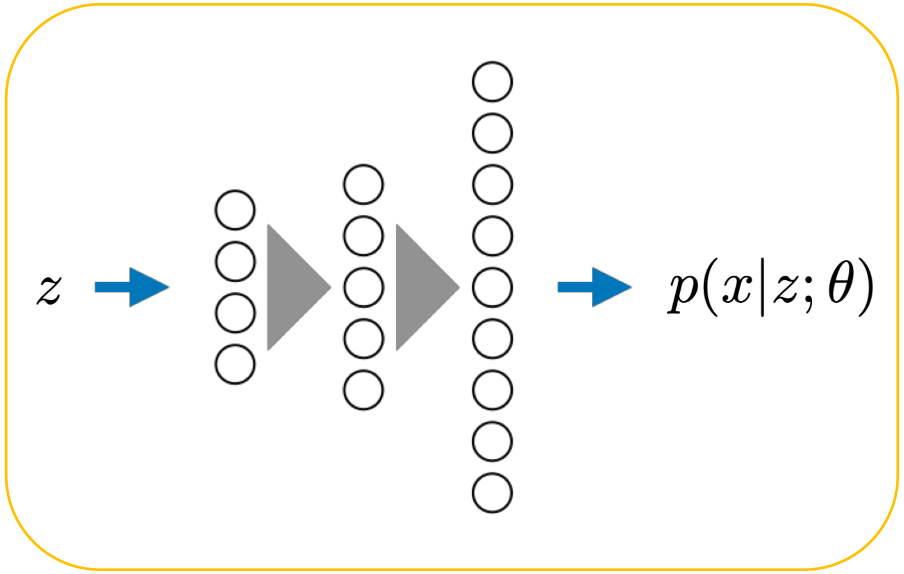
Jensen’s Inequality

$$\ln p(x;\theta) \geq \underbrace{\mathbb{E}_{q(z|x;\phi)}\left[\ln p(x|z;\theta)\right] - D_{KL}\left[q(z|x;\phi)\|p(z)\right]}_{\text{Evidence lower-bound : Maximize this instead of real likelihood}}$$

Variational auto-encoders



Encoding Model

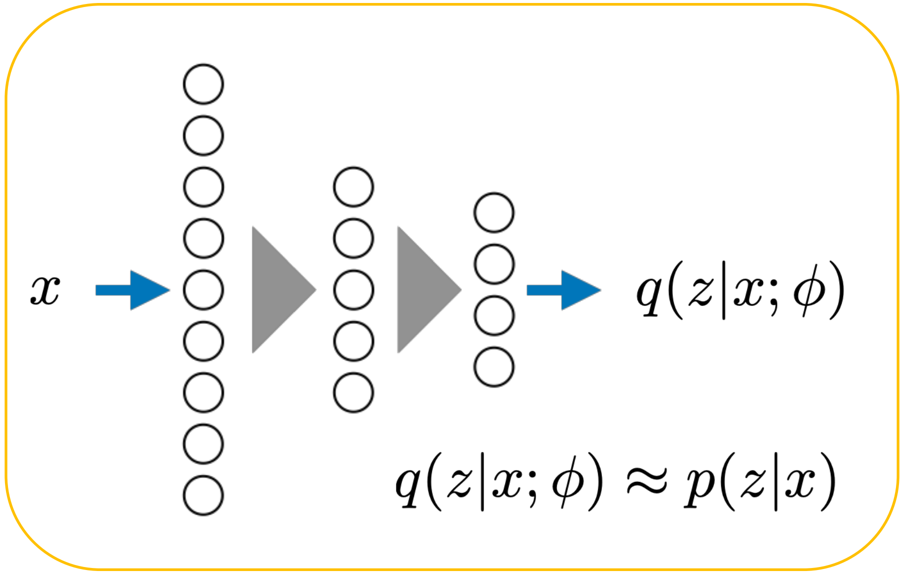


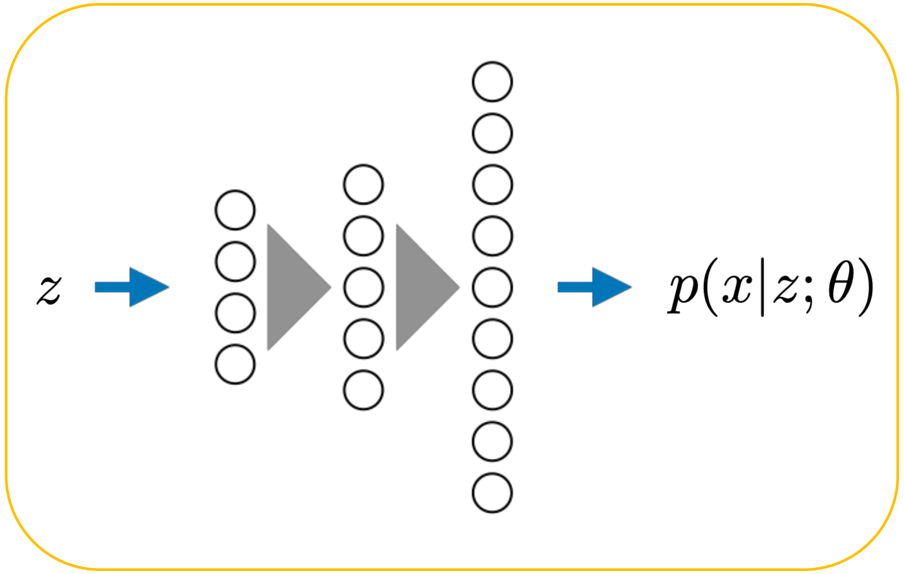
Decoding Model

Takes an image and maps it to the posterior distribution in the latent space.
Encodes to the lower dimensional space

Takes the lower dimensional representation and maps to an image.
Can be used as a sampler.
Can be used as a reconstruction tool

Variational auto-encoders





Encoding Model

Decoding Model

$$q(z|x;\phi) = \mathcal{N}(z; \mu_z(x;\phi), \Sigma_z(x;\phi))$$

$$p(x|z;\theta) = \mathcal{N}(x; \mu_x(z;\theta), \Sigma_x(z;\theta))$$

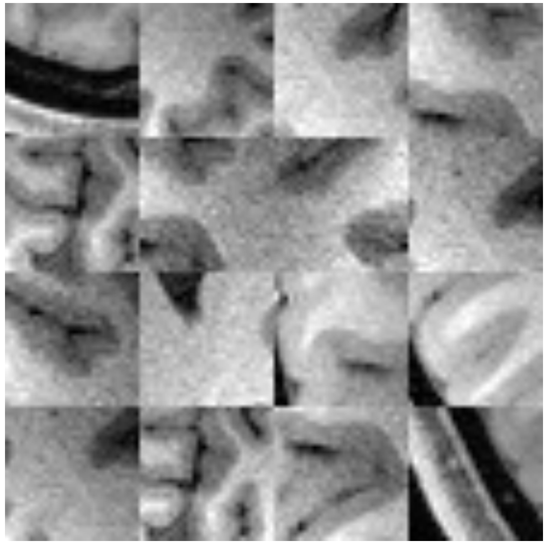
Both Gaussian Models

Homework: Can you determine the link with the probabilistic PCA model?

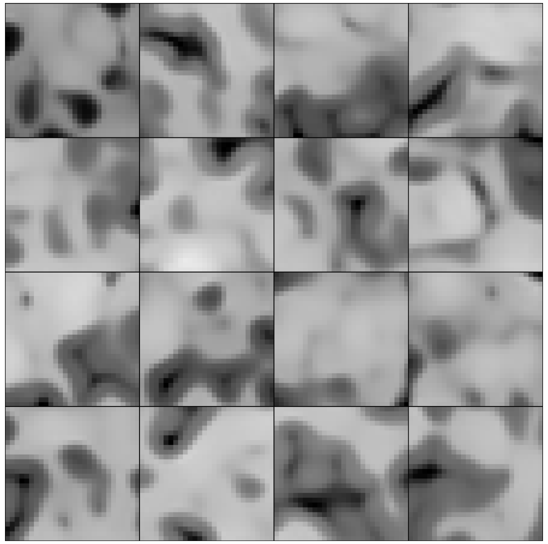
Difference with PCA

Image patches from Magnetic Resonance Images of the brain

Real patches of 28x28

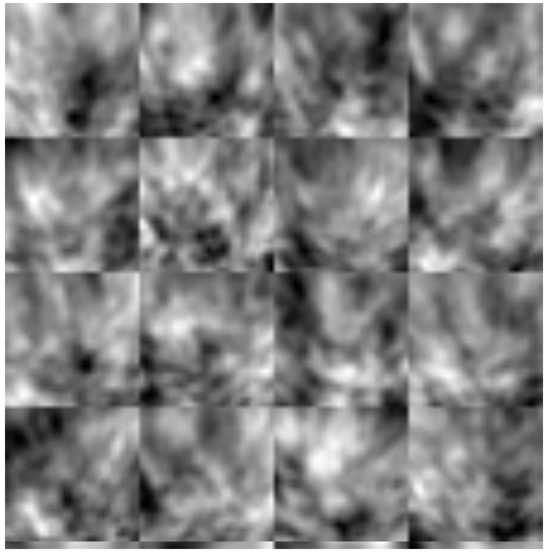


VAE Generated



60 components

PCA



250 components

60 components

[Tezcan et al. 2018]

Computer Vision

Visualization

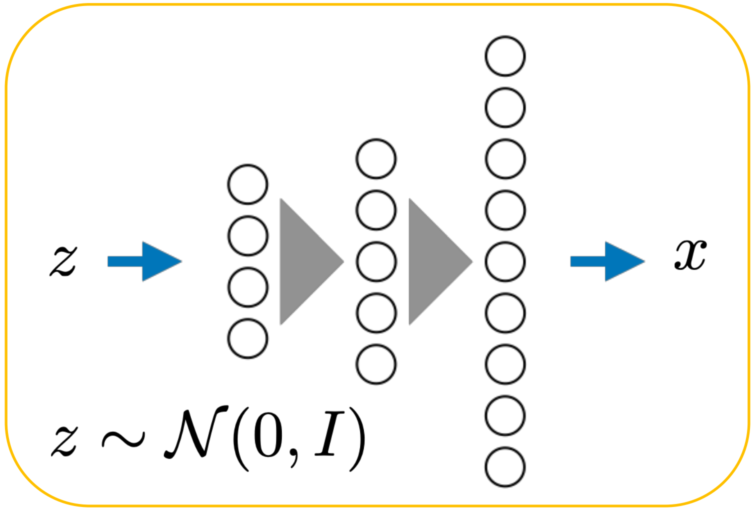
Combined Localization

Unsupervised Learning

Distribution Learning

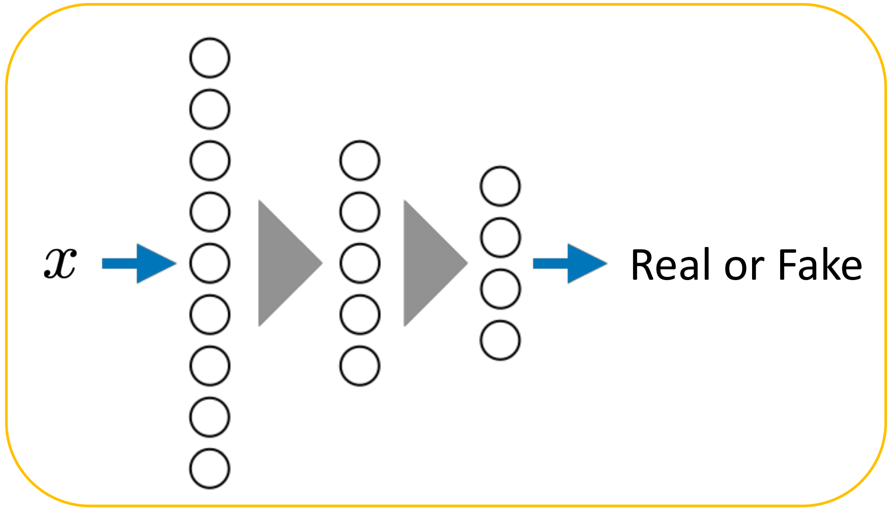
Generative adversarial networks

Instead of an explicit probabilistic model, a GAN is a sampling tool that generates samples from the data distribution



Generator

Generates realistic looking images from random samples in the latent space.



Discriminator

Tries to classify images into two categories: Real or generated (Fake)

Computer Vision

Visualization

Combined Localization

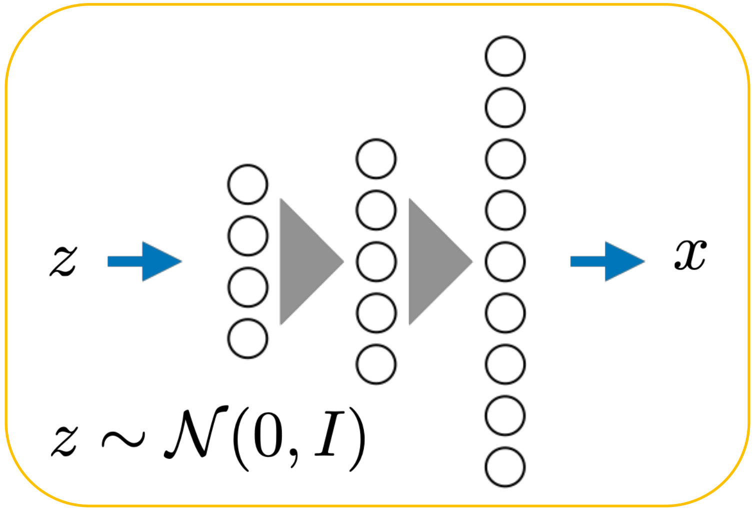
Unsupervised Learning

Distribution Learning

During training they compete

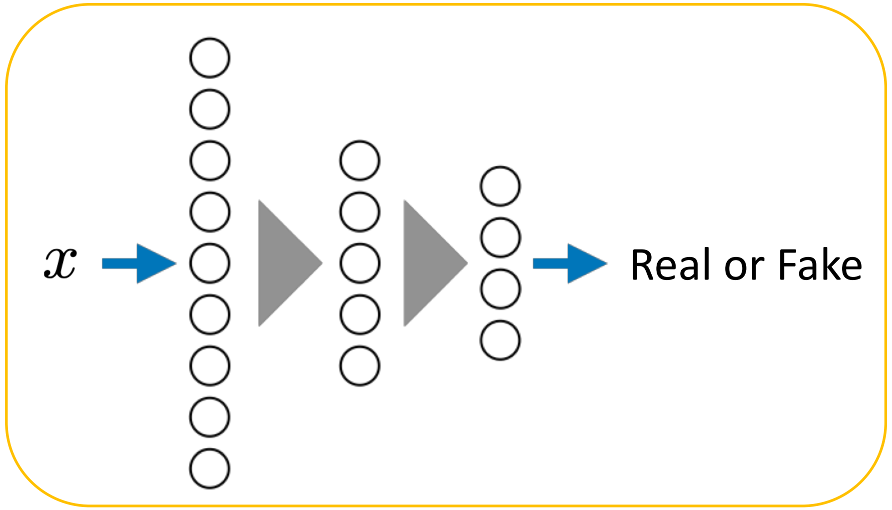
Generator - G

Tries to create samples that can fool the discriminator



Discriminator - D

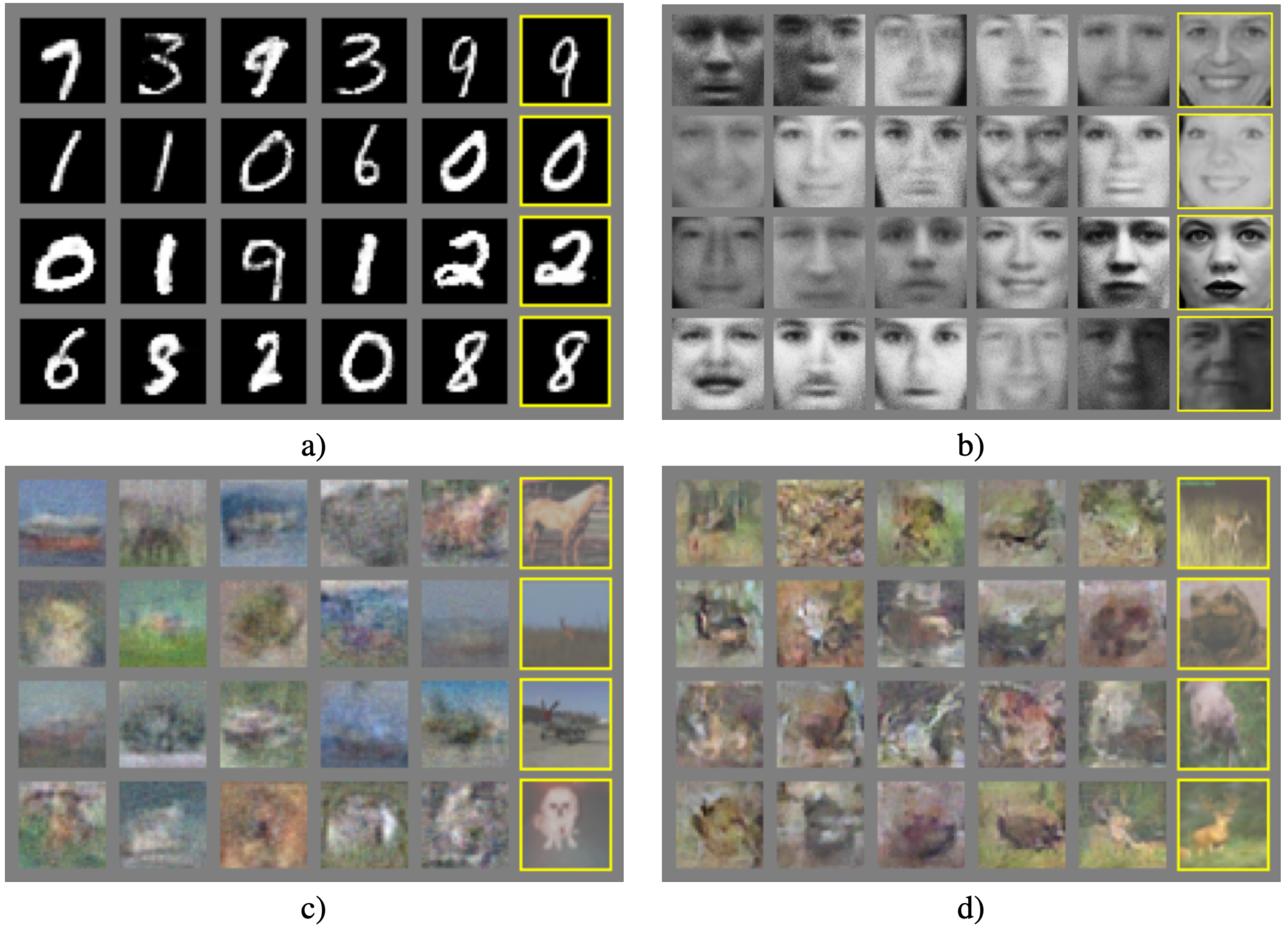
Tries to identifies the images the generator creates



Solve this problem: Optimize the network weights with a two-player game

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p_{\text{real}}}(x) \left[\ln D(x;\phi) \right] + \mathbb{E}_{z \sim p(z)} \left[\ln(1 - D(G(z;\theta)) \right]$$

Random samples



[Images from Goodfellow et al. 2014]

Computer Vision

Visualization

Combined Localization

Unsupervised Learning

Distribution Learning

Very active area of research

A Style-Based Generator Architecture for Generative Adversarial Networks

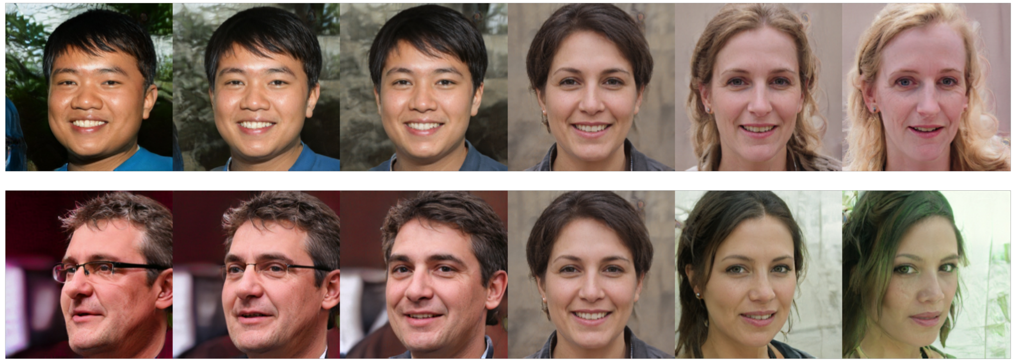
Tero Karras
NVIDIA
tkarras@nvidia.com

Samuli Laine
NVIDIA
slaine@nvidia.com

Timo Aila
NVIDIA
taila@nvidia.com

December 12, 2018

- **The model is not yet peer-reviewed**
- However, the samples they claim to generate are remarkable.



$\psi = 1$ $\psi = 0.7$ $\psi = 0.5$ $\psi = 0$ $\psi = -0.5$ $\psi = -1$

Interpolation between images

Computer Vision

Visualization

Combined Localization

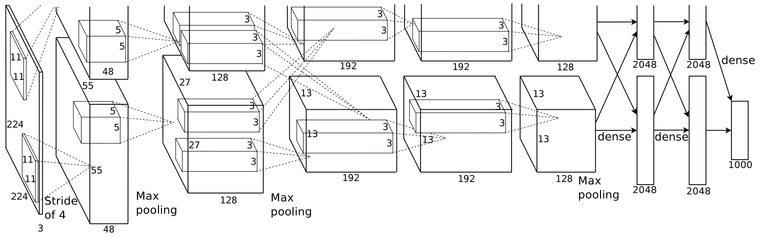
Unsupervised Learning

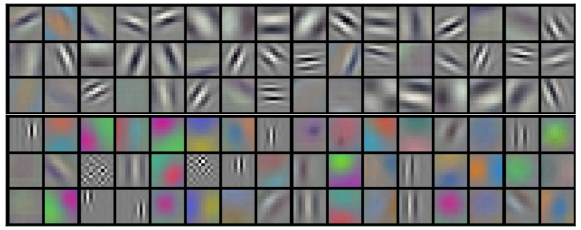
Distribution Learning

Unsupervised learning

Unsupervised learning of features

- Filters are important for performing image analysis tasks
- So far, we determine features in a supervised way, task-specific manner

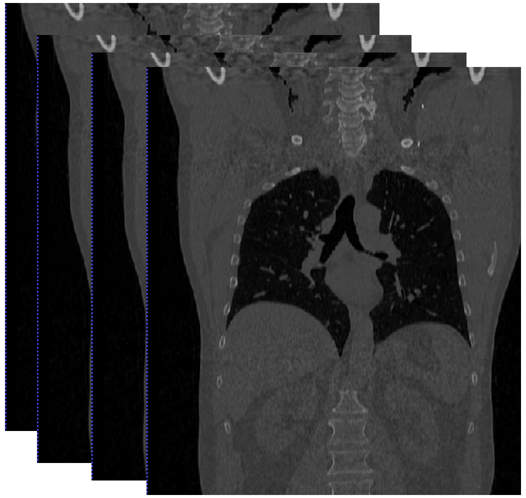




- Determine features in an unsupervised manner
- Examples have only features

Unsupervised learning of distributions

- Patterns within the data
- Goal: describe variability in the data
- Estimate the distribution of the data
- Only features
- Examples have only features



Unsupervised learning of features

- Features are important, they are the essential building blocks
- For any task it is important to get the right features
- It requires large number of labelled images to do this

1. It would be wonderful if we could do it with only few images
2. Humans do not seem to require lots of labelled images for good features, assuming humans do have good features
3. Are there features that can be used for any visual task?

Computer Vision

Visualization

Combined Localization

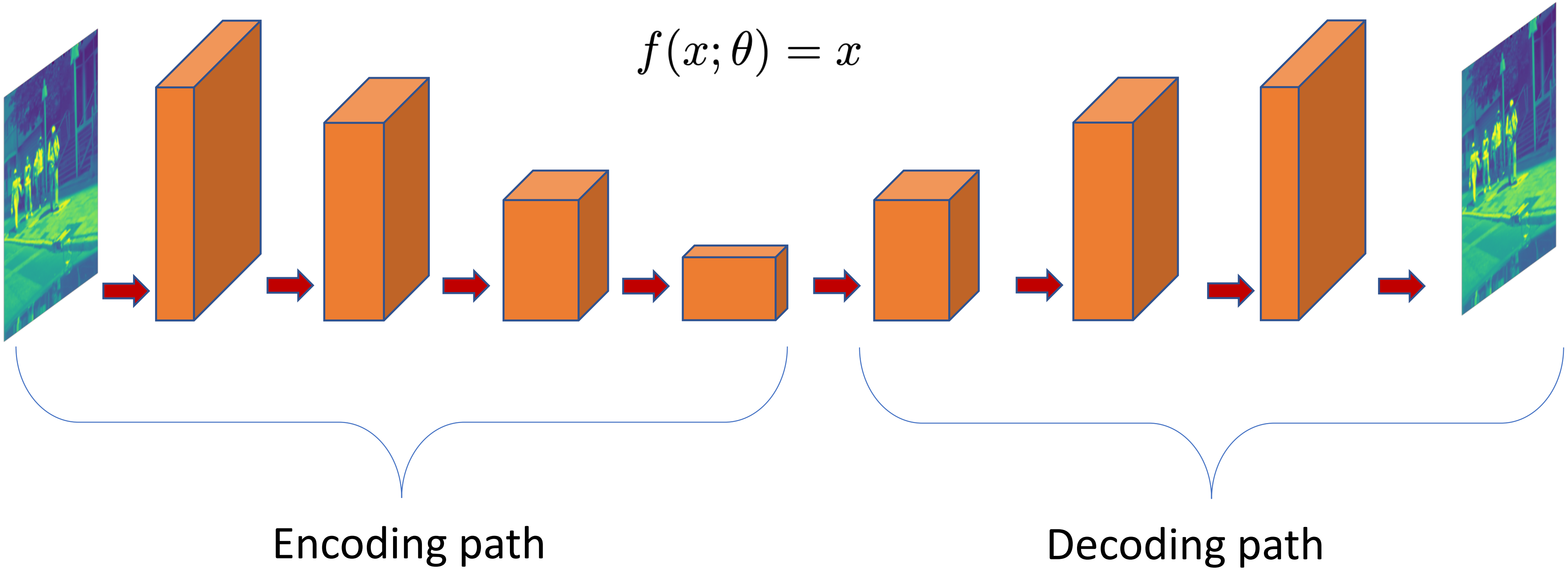
Unsupervised Learning

Distribution Learning

Unsupervised Feature Learning

Auto-encoding models

$$f(x;\theta) = x$$

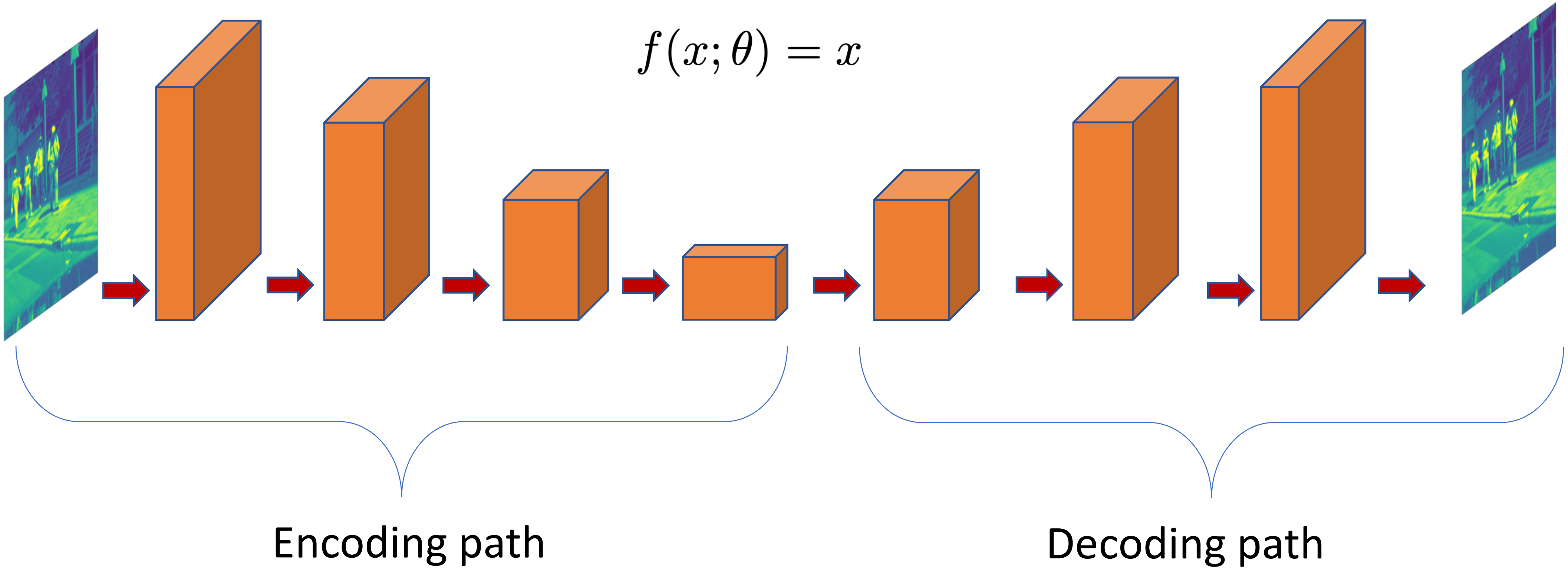


$$\min_{\theta} \|x - f(x;\theta)\|_2^2$$

The bottleneck layer does not allow the network to learn an identity map

It learns to summarize the most important information for reconstruction

Auto-encoding models



$$\min_{\theta} \|x - f(x;\theta)\|_2^2$$

Minimization only requires the images. The goal is to be able to reconstruct the image with high fidelity.

Computer Vision

Visualization

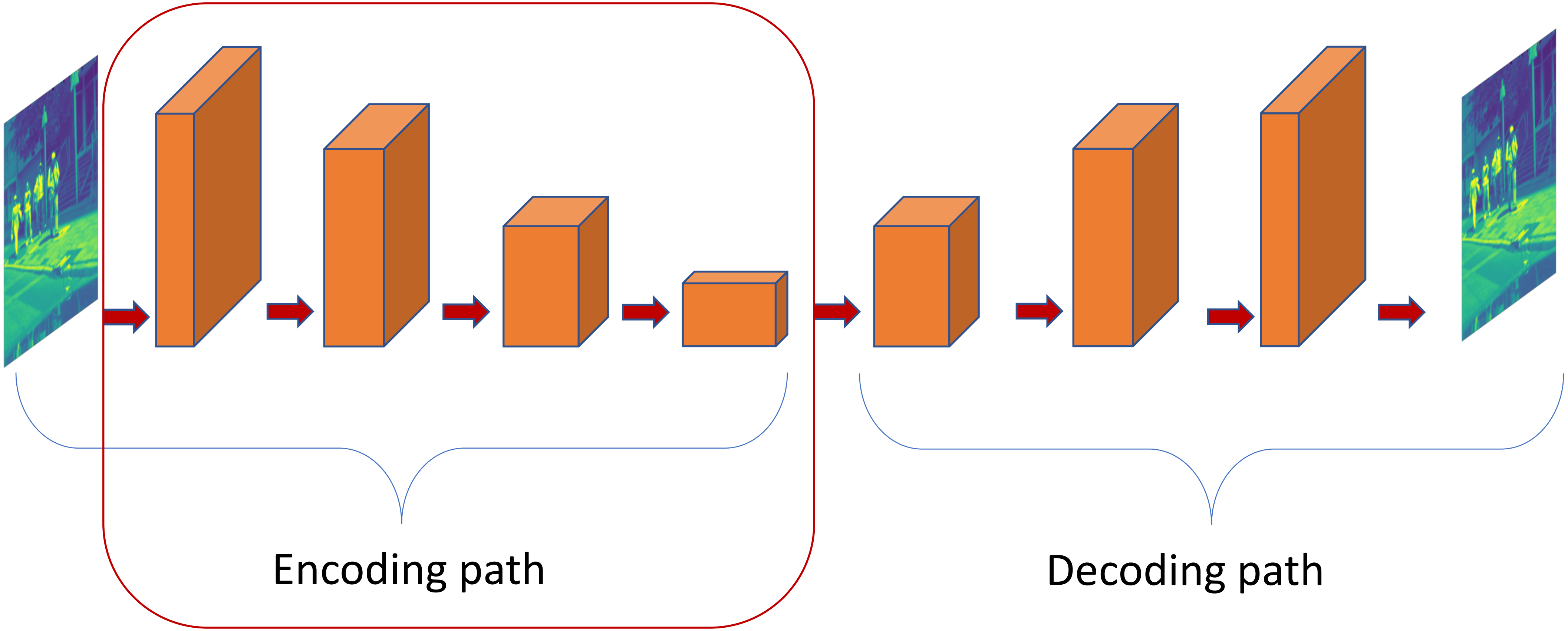
Combined Localization

Unsupervised Learning

Distribution Learning

Unsupervised Feature Learning

Auto-encoding models



Computer Vision

Visualization

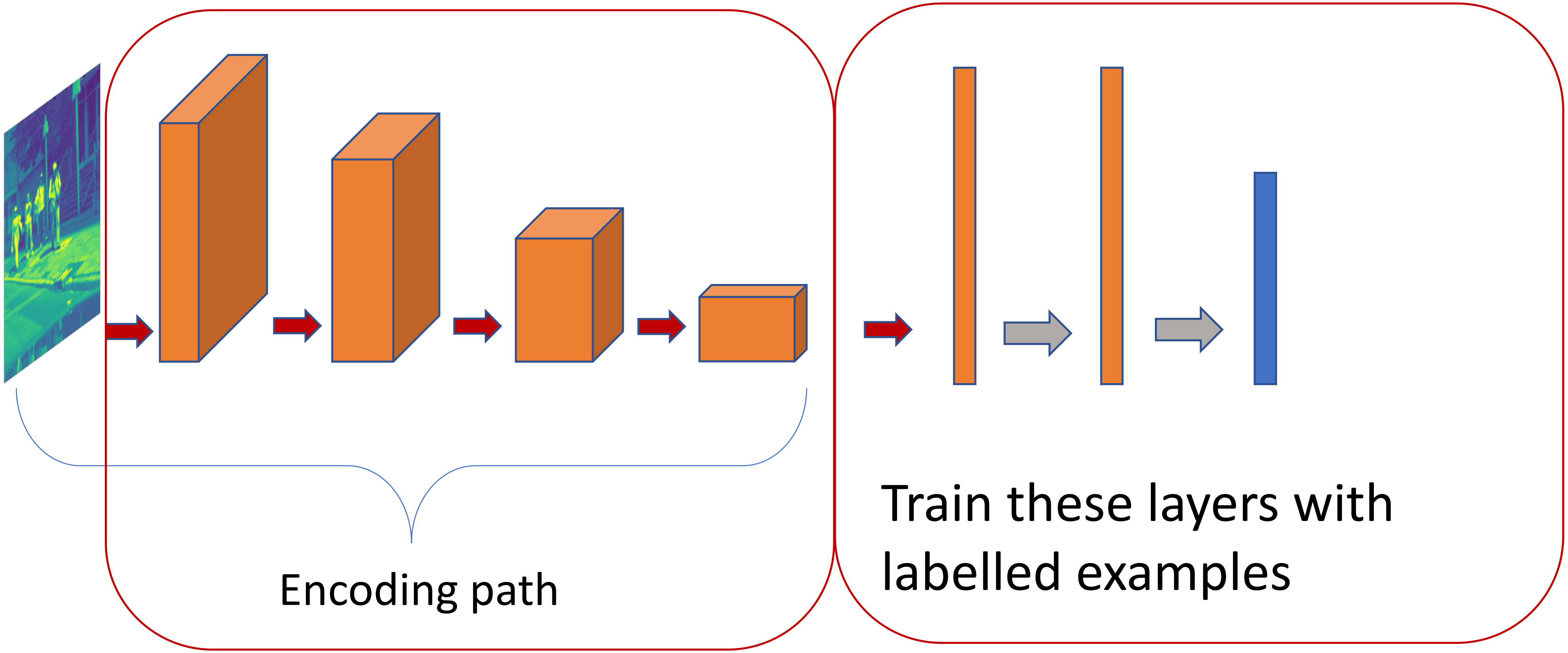
Combined Localization

Unsupervised Learning

Distribution Learning

Unsupervised Feature Learning

Auto-encoding models



Features learnt here can then be translated
to another task either directly or by fine-tuning, i.e. starting the optimization
from the pre-learnt weights

In practice

- The features learnt from a simple auto-encoder can be very helpful
- They are not however, extremely useful
- In the end, you may still need large number of labelled examples
- Not as large as training from scratch though

Computer Vision

Visualization

Combined Localization

Unsupervised Learning

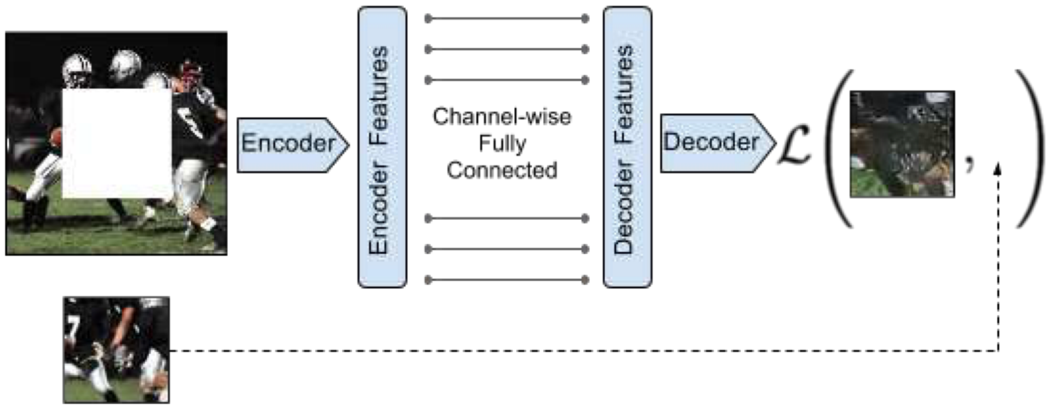
Distribution Learning

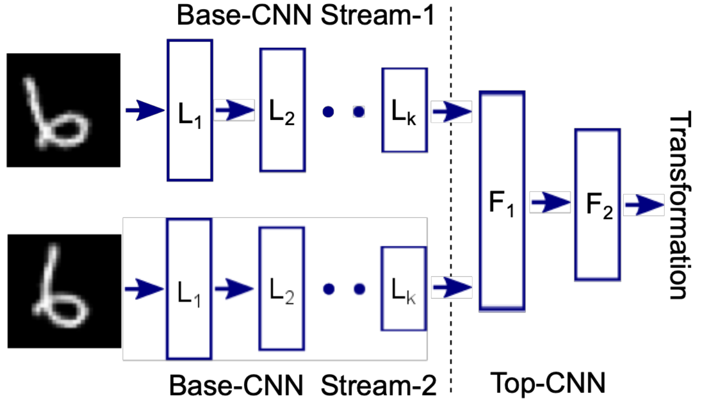
Unsupervised Feature Learning

An example from more recent works – Context-Encoder

Context Encoders: Feature Learning by Inpainting

Deepak Pathak Philipp Krähenbühl Jeff Donahue Trevor Darrell Alexei A. Efros
University of California, Berkeley
{pathak,philkr,jdonahue,trevor,efros}@cs.berkeley.edu





Learning to See by Moving

Pulkit Agrawal
UC Berkeley
pulkitag@eecs.berkeley.edu

João Carreira
UC Berkeley
carreira@eecs.berkeley.edu

Jitendra Malik
UC Berkeley
malik@eecs.berkeley.edu