Convolutional neural network

Huy V. Vo

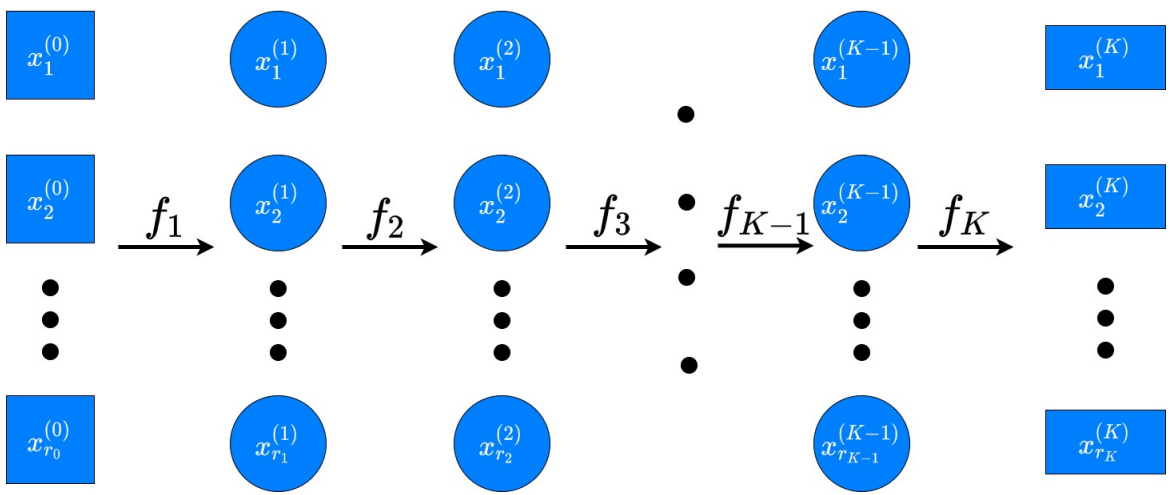
MASSP

Outline

- Introduction
  - Neural network reminder.
  - Convolutional neural network and its applications.
- Convolution in image processing.
- Convolutional neural networks.

Introduction

- Neural network reminder:
  - A model which consists of multiple layers, noted x_0, x_1, ..., x_K where x_0 is the input layer, x_K is the output layers and x_1, ... x_{K-1} are hidden layers.
  - Layers are vector of real numbers. Their entries are called neurons.
  - Layer k is obtained by applying a function f_k on layer $k-1$: $x_k = f_k(x_{k-1})$. The neural network represents the function $f = f_K \circ f_{K-1} \circ ... \circ f_1$.

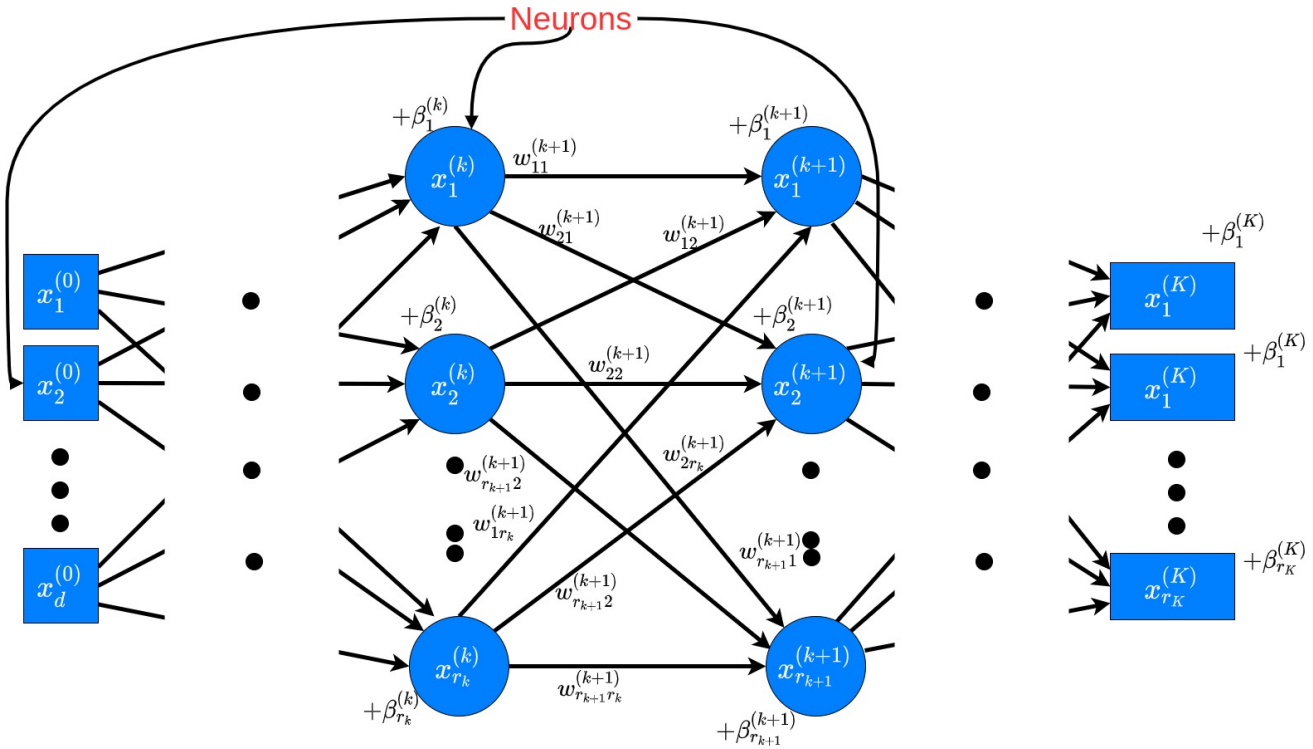


Introduction

- Neural network reminder:
  - Usually $f_k(x) = \sigma_k(W^{(k)}x + \beta^{(k)})$ where $W^{(k)} \in \mathbb{R}^{r_k \times c_k}$ and $\beta^{(k)} \in \mathbb{R}^{r_k}$ and σ_k **is a point-wise non-linear activation function**. So:

$$x^{(k)} = \sigma(W^{(k)}x^{(k-1)} + \beta^{(k)}).$$

  - $W^{(k)}$ and $\beta^{(k)}$, $1 \leq k \leq K$, are the parameters of the neural network. K and r_k, $1 \leq k \leq K$ are hyper-parameters.

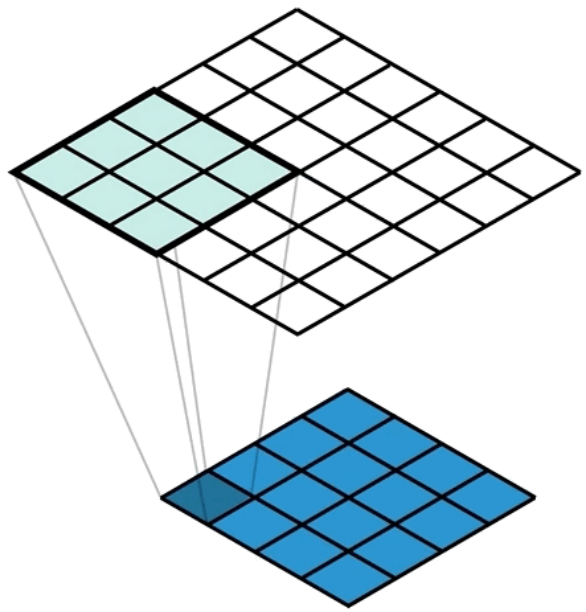


Introduction

- Neural network reminder: When the weight matrices $W^{(k)}$ are dense, we have a fully connected neural network.
  - The spatial relation between neurons is not accounted $\rightarrow$ not optimal for data with spatial structure like images, speech, ...
  - Not all neurons are correlated. For example, pixels in images often correlate only to nearby pixels.
  - Too many parameters $\rightarrow$ heavy computation.

Introduction

- Convolutional neural network:
  - A special type of neural network with *convolution* as the linear operator.
  - Work on data where spatial information is important.
  - Each neuron in the $k+1$-th layer is connected to only a few neurons in the k-th layer.



Introduction

- Applications: Image classification.
  - Input: Images as $H \times W \times C$ tensors (RGB, gray scale, ...).
  - Output: Class of the input images (dogs, cats, horses, ...).
  - ImageNet: 1000 image classes, 1.2 millions images. ILSVRC challenge from 2010 to 2017.

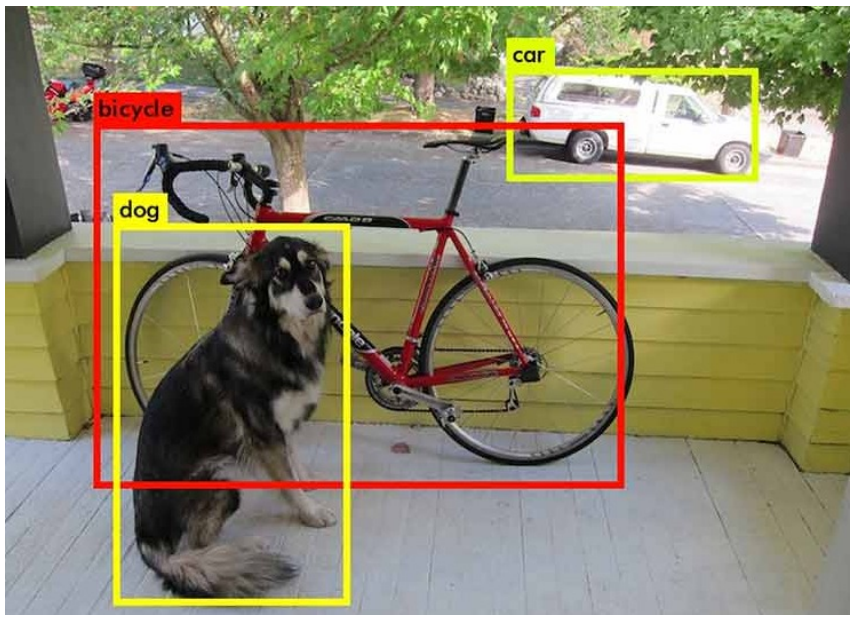
Team name	Filename	Error (5 guesses)	Description
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.

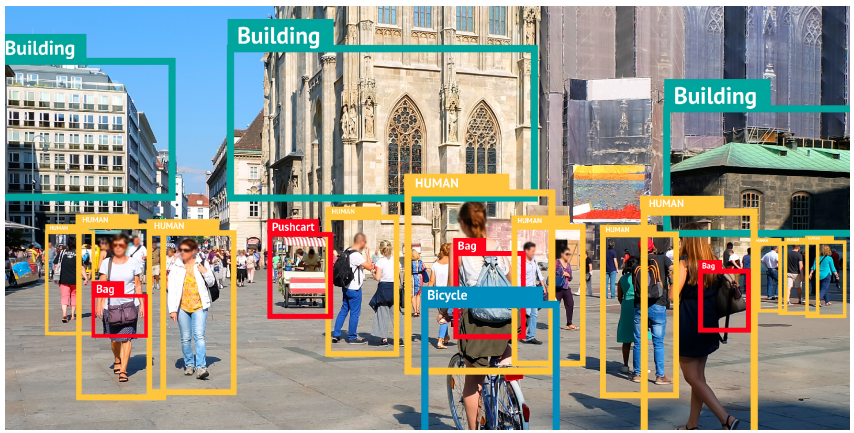
	Top1 Acc.
ResNet-152 (He et al., 2016)	77.8%
EfficientNet-B1	**79.1%**
ResNeXt-101 (Xie et al., 2017)	80.9%
EfficientNet-B3	**81.6%**
SENet (Hu et al., 2018)	82.7%
NASNet-A (Zoph et al., 2018)	82.7%
EfficientNet-B4	**82.9%**
GPipe (Huang et al., 2018) †	84.3%
EfficientNet-B7	**84.3%**

On par with expert human performance.

Introduction

- Applications: Object detection.
  - Input: images and a set of predefined object classes.
  - Output: Rectangles around the objects in the images and their classes.
  - Important for autonomous driving, surveillance, robotics,...





RCNN, Fast RCNN, Faster RCNN, YOLO, Mask RCNN, ...

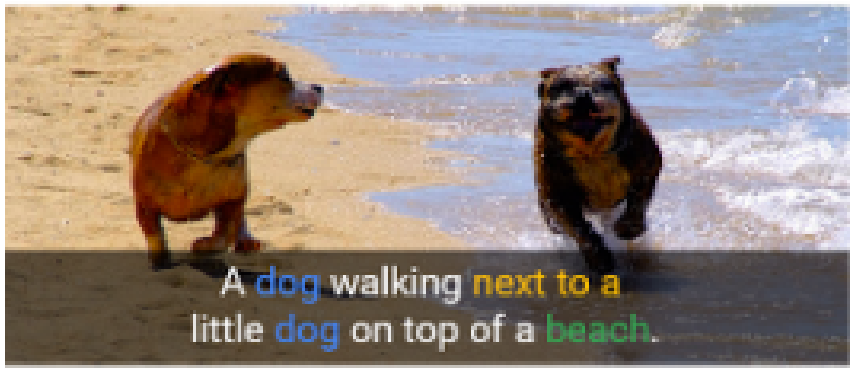
Introduction

- Applications: Semantic Segmentation.
  - Input: Images and a predefined set of object classes.
  - Output: Pixel assignment to the classes and possibly their instances.

FCN, DeepLab, UNet,...

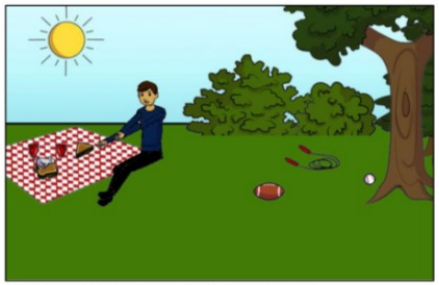
Introduction

- Applications: Image Captioning, Visual Question answering.
  - Input: An image or an image and a question.
  - Output: A summary of the image content (caption) or an answer to the question based on the image content.




What color are her eyes?
What is the mustache made of?

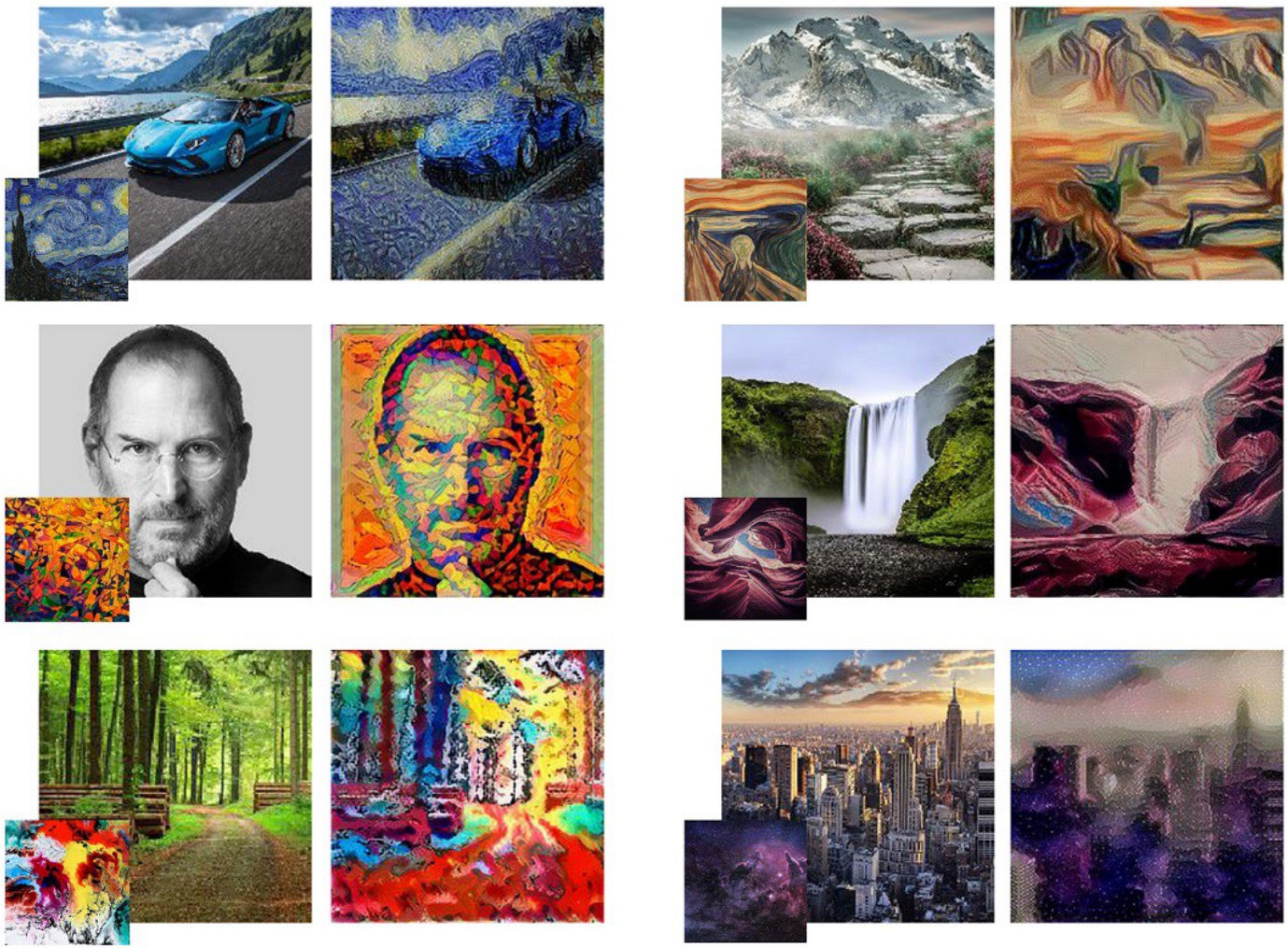

How many slices of pizza are there?
Is this a vegetarian pizza?


Is this person expecting company?
What is just under the tree?


Does it appear to be rainy?
Does this person have 20/20 vision?

Introduction

- Applications: Style transfer.
  - Create deep art from a *content* image and a *style* image. https://reiinakano.com/arbitrary-image-stylization-tfjs/



Introduction

- Applications: Video Style transfer, DeepFake.
  https://www.youtube.com/watch?v=cQ54GDm1eL0

Convolution in Image Processing

- Image filtering: Common image processing technique.
  - Eliminate information that is redundant in later processing steps and extract only useful information from the image.
  - Image editing.
  - Common filters: blur (Box, Gaussian), sharpen, edge detection (Sobel, Laplacian).

Original

Box Blur

Gaussian Blur

Sharpen

Laplacian

Sobel x

Sobel y

Convolution in Image Processing

- Local linear filtering:
  - Applying a linear operator on a small neighborhood of each pixel. The most common form is the multiplication with a *kernel* K.

$$\tilde{I}(i,j) = \sum_{m,n} I(i+m, j+n)K(m,n) = I \otimes K,$$

    where the values of (m,n) define the neighborhood of pixel (i,j). This can be seen as the *correlation* of the kernel K and (i,j)'s neighborhood.
  - A variant of the above formula:

$$\tilde{I}(i,j) = \sum_{m,n} I(i-m, j-n)K(m,n) = I * K.$$

    This is called the *convolution* operator.
  - Correlation and convolution are almost identical but convolution is more popular in image pre-processing because it allows fast computation with Fourier transform.

Convolution in Image Processing

- Convolution: A closer look.

I

13	22	56	57	100
17	23	45	42	37
78	10	0	5	33
89	72	81	23	19
17	1	46	55	99

K

0 (-1,-1)	1 (-1,0)	0 (-1,1)
0 (0,-1)	0 (0,0)	-1 (0,1)
0 (1,-1)	0 (1,0)	0 (1,1)

$\tilde{I}$

	-7	-23	-40	
	-6	71	23	
	-88	-26	-26	

Convolution in Image Processing

- Convolution: A closer look.

I

13	22	56	57	100
17	23	45	42	37
78	10	0	5	33
89	72	81	23	19
17	1	46	55	99

K

0 (-1,-1)	1 (-1,0)	0 (-1,1)
0 (0,-1)	0 (0,0)	-1 (0,1)
0 (1,-1)	0 (1,0)	0 (1,1)

$\tilde{I}$

	-7	-23	-40	
	-6	71	23	
	-88	-26	-26	

Convolution in Image Processing

- Convolution: Padding.

I

0	0	0	0	0	0	0
0	13	22	56	57	100	0
0	17	23	45	42	37	0
0	78	10	0	5	33	0
0	89	72	81	23	19	0
0	17	1	46	55	99	0
0	0	0	0	0	0	0

0 (-1,-1)	1 (-1,0)	0 (-1,1)
0 (0,-1)	0 (0,0)	-1 (0,1)
0 (1,-1)	0 (1,0)	0 (1,1)

$\tilde{I}$

17	10	23	-14	-20
78	-7	-23	-40	-9
89	-6	71	23	14
17	-88	-26	-26	76
0	-17	-1	-55	-55

Convolution in Image Processing

- Convolution: Stride.

I

0	0	0	0	0	0	0
0	13	22	56	57	100	0
0	17	23	45	42	37	0
0	78	10	0	5	33	0
0	89	72	81	23	19	0
0	17	1	46	55	99	0
0	0	0	0	0	0	0

I

0	0	0	0	0	0	0
0	13	22	56	57	100	0
0	17	23	45	42	37	0
0	78	10	0	5	33	0
0	89	72	81	23	19	0
0	17	1	46	55	99	0
0	0	0	0	0	0	0

I

0	0	0	0	0	0	0
0	13	22	56	57	100	0
0	17	23	45	42	37	0
0	78	10	0	5	33	0
0	89	72	81	23	19	0
0	17	1	46	55	99	0
0	0	0	0	0	0	0

0 (-1,-1)	1 (-1,0)	0 (-1,1)
0 (0,-1)	0 (0,0)	-1 (0,1)
0 (1,-1)	0 (1,0)	0 (1,1)

$\tilde{I}$

17	23	-20
89	71	14
0	-1	-55

$\tilde{I}$

17	23	-20
89	71	14
0	-1	-55

$\tilde{I}$

17	23	-20
89	71	14
0	-1	-55

Convolution in Image Processing

- Convolution: Popular kernels.

Box Blur

$$\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Gaussian Blur

$$\frac{1}{25}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 1 \\ 6 & 24 & 36 & 24 & 1 \\ 4 & 16 & 24 & 16 & 1 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Sharpen

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Convolution in Image Processing

- Convolution: Popular kernels.

Laplacian

0	1	0
1	-1	1
0	1	0

Sobel X

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

Sobel Y

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

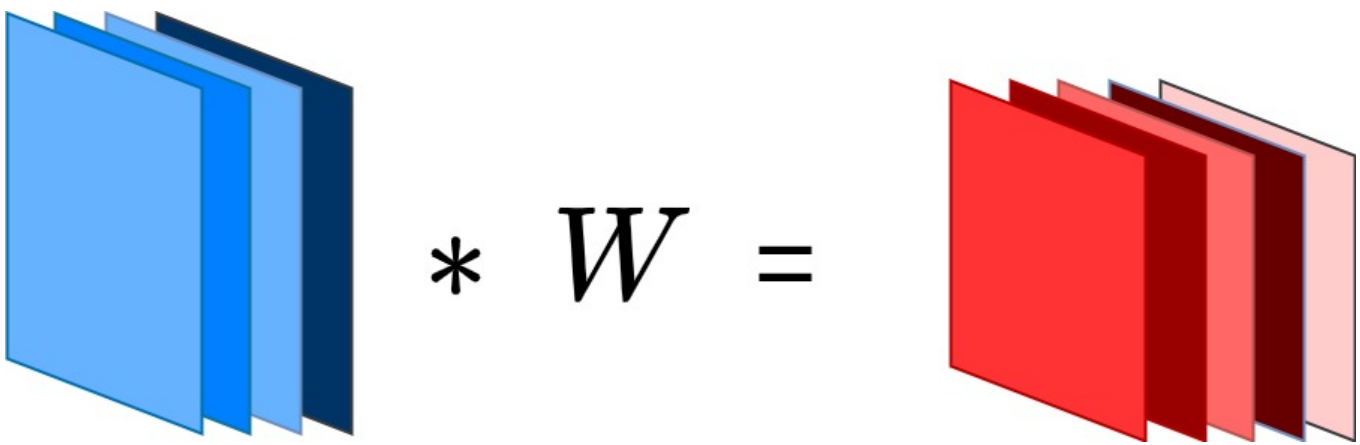
Machine learning on imaging data before deep learning

- Pre-process the images with appropriate filters to detect local information (edges, corners, ...). Multiple filters are used successively.
- Optimize a classifier/regressor on the pre-processed features (Linear regression, logistic regression, SVM, ...).
- Drawbacks:
  - Filters are manually designed, not adaptive to the input data.
  - It is hard to design a large number of meaningful filters.
  - It is hard to choose the order in which the filters are applied.

$\Rightarrow$ Learn the filters and a classifier/regressor simultaneously.

Convolutional neural network

- Neural networks where linear operators are linear convolution operators.
- A typical convolutional layer in models for computer vision:
  - Input: $T^{in} \in \mathbb{R}^{H^{in} \times W^{in} \times C^{in}}$ tensor. Think of it as an image with C^{in} color channels.
  - Output: $T^{out} \in \mathbb{R}^{H^{out} \times W^{out} \times C^{out}}$ tensor.
  - Convolutional kernels $W \in \mathbb{R}^{m \times n \times C_{in} \times C_{out}}$. W is a collection of $C_{in} \times C_{out}$ convolutional kernels of size $m \times n$.
  - Convolution is actually implemented in popular frameworks (Pytorch, Tensorflow, Keras, ...) as correlation.

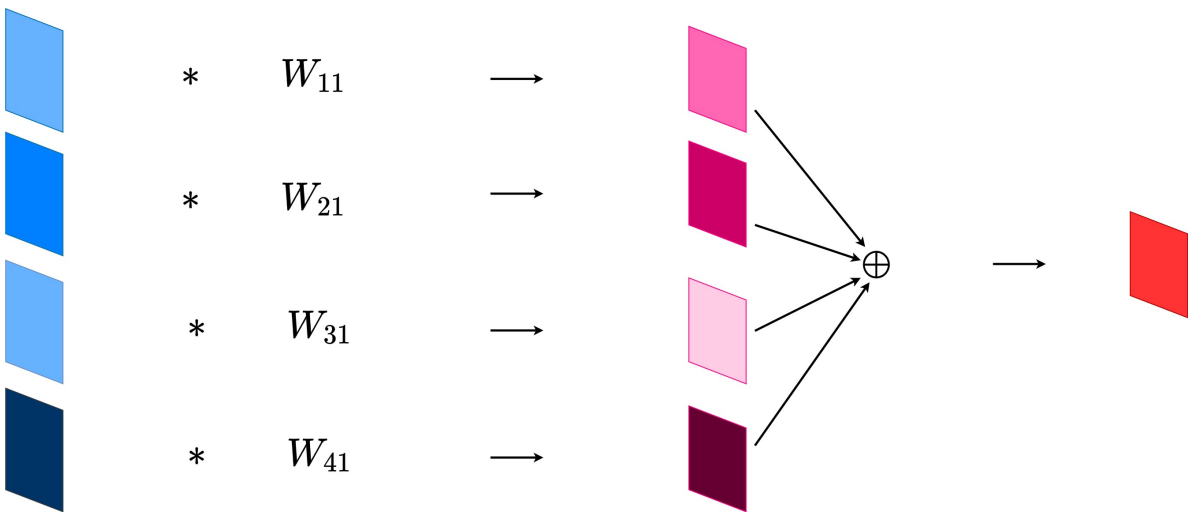


Convolutional Neural Network

- A typical convolutional layer in models for computer vision:
  - Denote channel i of tensor T as F_i. F_i^{out} is obtained by applying a convolution on each channel of T^{in} and taking the sum of these convolutions.

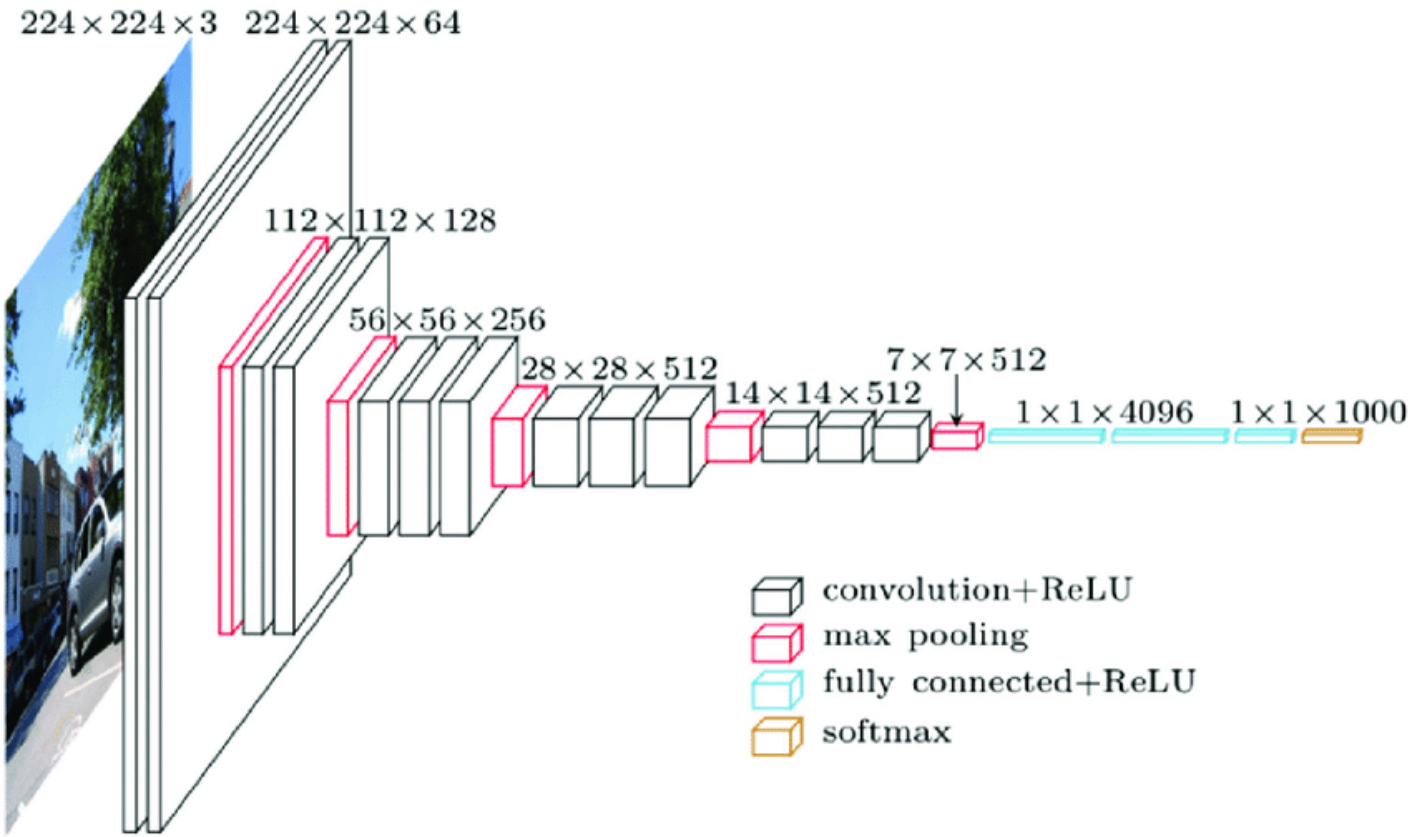
$$F_i^{out} = \sum_{j=1}^{C^{in}} F_i^{in} * W_{ji},$$

where W_{ji} is a convolutional kernel. W_{ji} are chosen to have the same size and can be stacked into a weight tensor $W \in \mathbb{R}^{m \times n \times C^{in} \times C^{out}}$ where $m \times n$ is the size of the convolutional kernels.



Convolutional neural networks

- VGG16: A typical convolutional neural network.
  - 13 convolutional layers, 3 fully connected layers.
  - Filters of size 3×3.
- Other CNNs: AlexNet, LeNet, ResNet, WideResNet, DenseNet, EfficientNet, ...



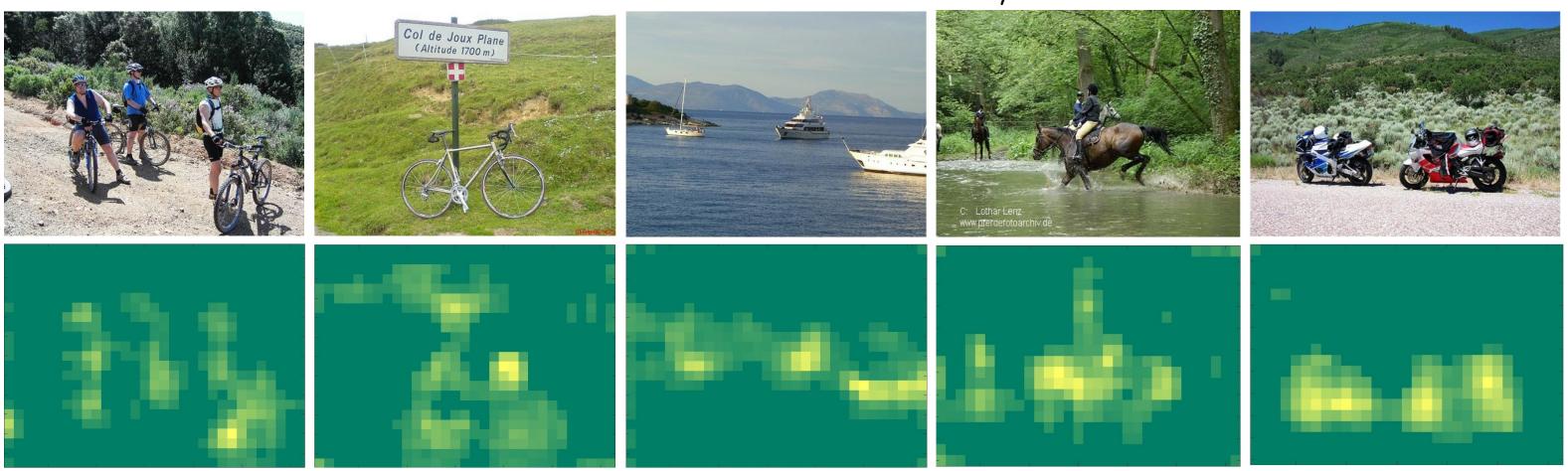
Convolutional Neural Network

- VGG16: Pytorch.

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Convolutional Neural Network

- Filters in the first layer of CNNs trained for image classification resemble to edge detectors.
- CNNs produce features that highlight important objects in images.
- CNN pretrained for classification on large data sets can be used to extract features for other tasks / data sets.



Conclusions

- Convolutional neural network (CNN) is a special type of neural network suitable for structured data (images, videos, text, . . .).
- CNN has much fewer parameters than fully connected neural network, thus cheaper in memory and faster to train.
- CNN yields state-of-the-art performance in many computer vision tasks.