Deep Learning

Convolutional Neural Networks

Edgar F. Roman-Rangel.
edgar.roman@itam.mx

Digital Systems Department.
Instituto Tecnológico Autónomo de México, ITAM.

February 19th, 2021.

Outline

Definition

Sub-field of Artificial Intelligence, which addresses several tasks around enabling computers with the ability to gain high-level understanding from visual sensors.

Teach machines how to see.

Definition

Extract relevant patterns from images, then make sense of them.

- What/who is there in the image?
- Where are they within the image?
- How are they?
- What are they doing?
- ...

Vision process

- Acquiring images (signal sensing).
- Processing data from pixels (image processing)..
- Interpret information (machine learning).

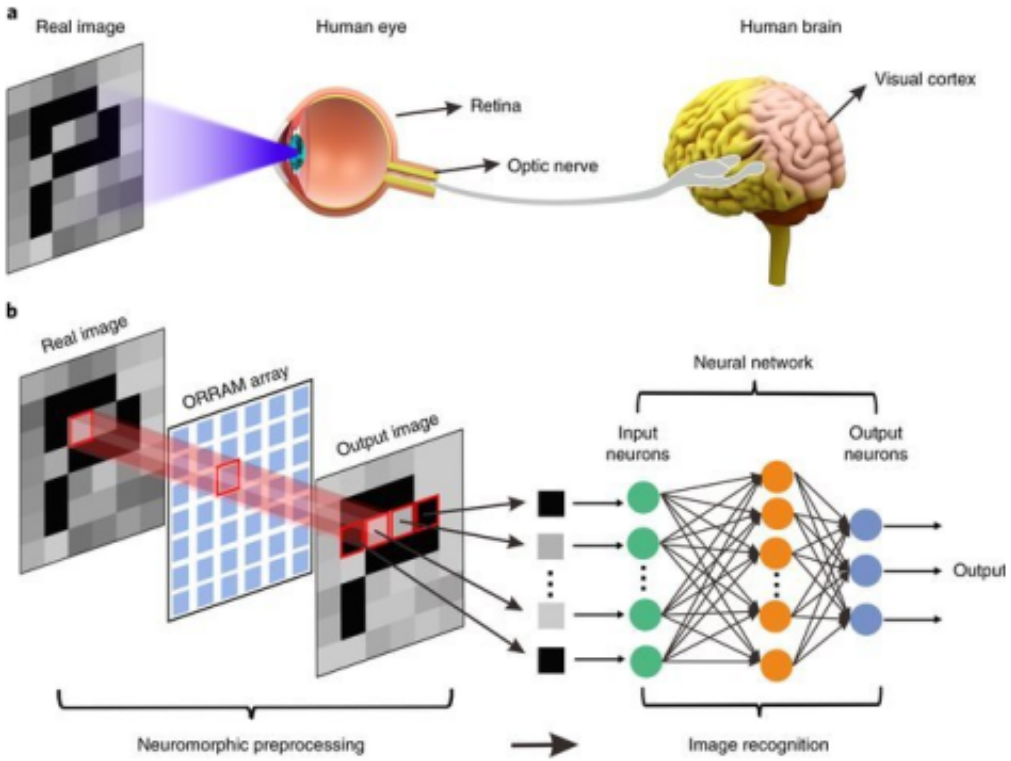
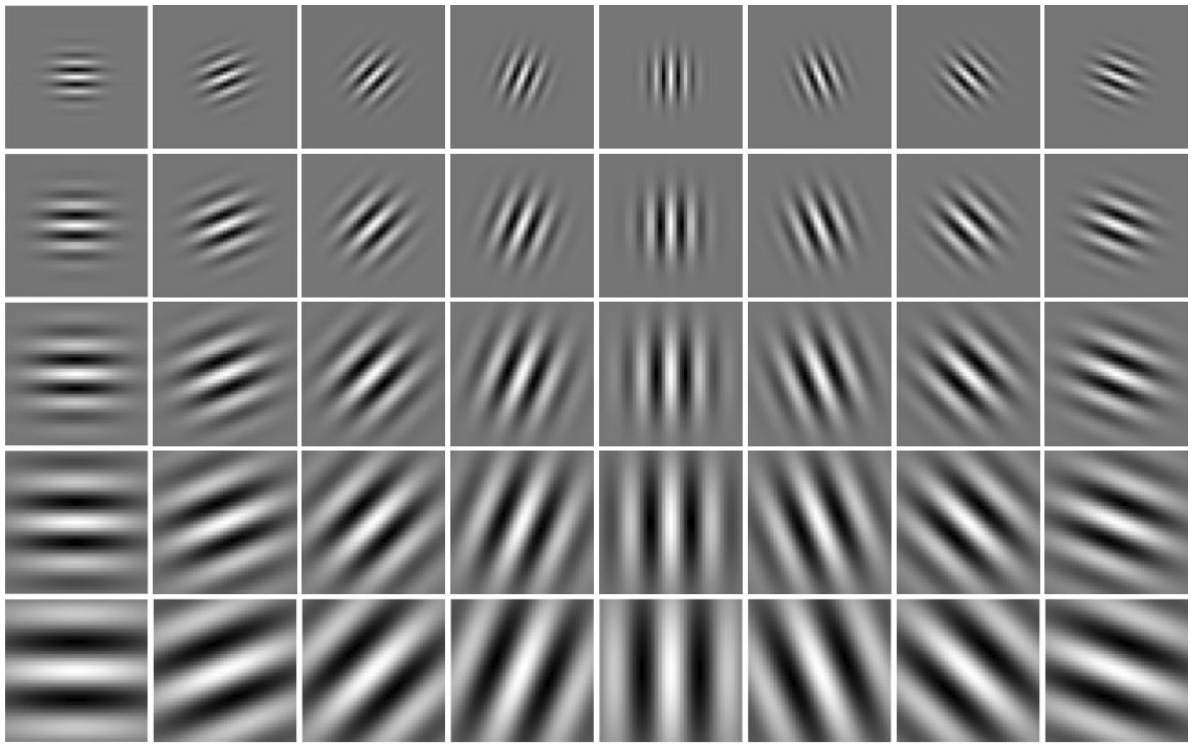


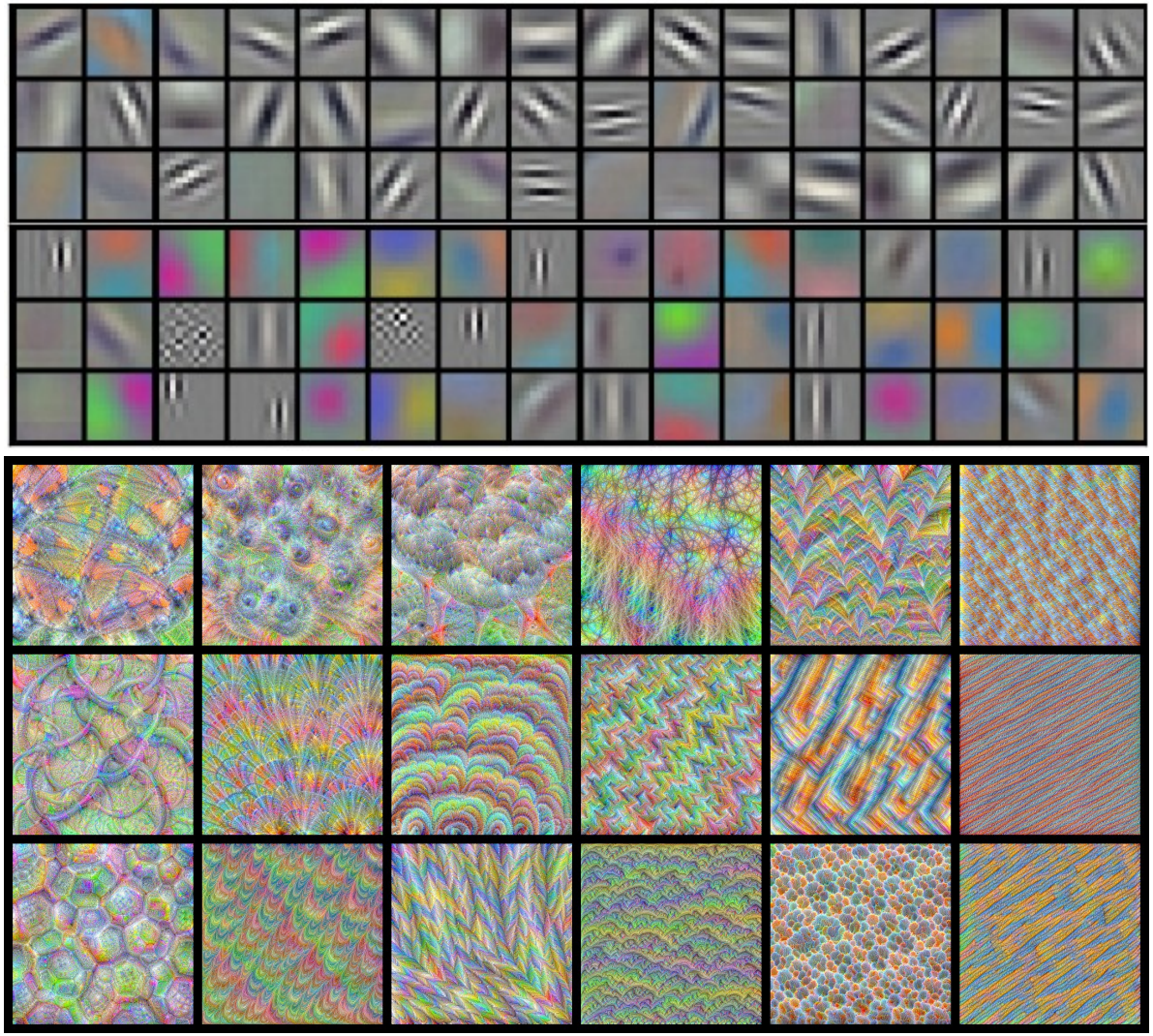
Image filters, I

- Design digital filters to extract relevant information.
- Generate numerical representation (vector) of images.



Filters are commonly applied by **convolution** operations.

Image filters, II



Common tasks

Classification:

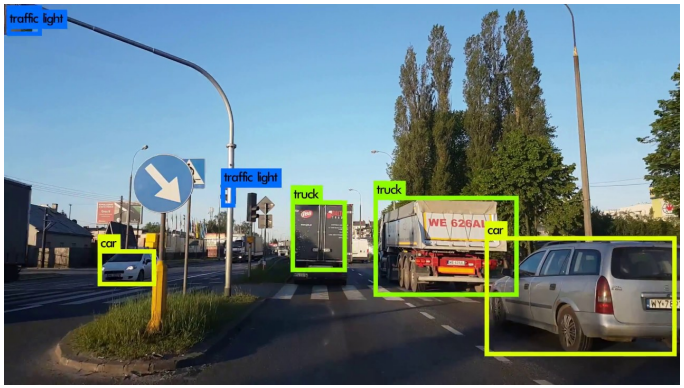
assign a label.

Segmentation:

pixels of an object?

Detection and localization:

is there certain object? where?



Generation:

synthetic (a.k.a., fake)

Popular examples

- e-commerce.
- Beauty/Quality assessment.
- Information retrieval.
- Video surveillance.
- Autonomous navigation.
- Medical applications.
- Game Playing.
- Style Transfer.
- Weather forecast.
- etc.

Outline

Definition

Etymology

lat. Convolvere: *volvere* (roll), *com* (together).

Meaning

Roll together. Entwine. Merged shapes.
Combine one function (image or sound) with another (filter).

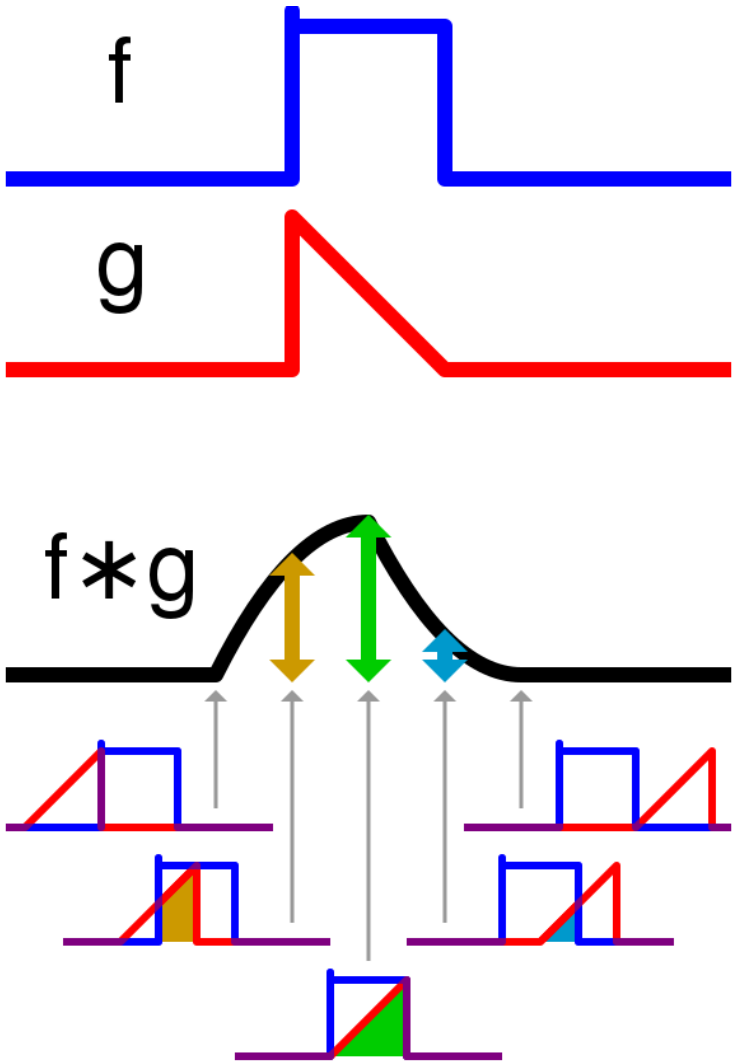
Mathematically $(*)$

Discrete time, $y[n] = x[n] * h[n]$:

$$(x * h)[n] = \sum_{k=-\infty}^{\infty} x[n-k]h[n],$$
$$= \sum_{k=-\infty}^{\infty} x[k]h[n-k].$$

Often, h is zero-centered, and of odd length, e.g., for $h[n]$, $n = \{-N, -N+1, \ldots, 0, \ldots, N-1, N\}$.

1D Visual example



1D convolution

Procedure:

1. Mirror one signal (often the kernel).
2. Slide it over the other signal (input), one sample at the time.
3. Multiply overlaps and record partial resulting signals.
4. Sum up partial signals to get the final output.

Example, I

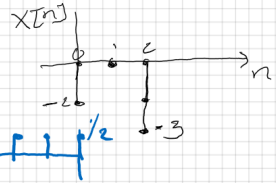
$$x = [-2, 0, -3], \qquad h = \frac{1}{3}[1, 1, 1].$$

$$y[n] = x[n] * h[n] = [-0.667, -0.667, -1.667, -1, -1].$$

Notice the length of $y[n]$. It is $M + N - 1$, where M and N are the lengths of x and h, respectively.

Example, II

$X = [-2, 0, -3]$



$h = \frac{1}{3}[1, 1, 1]$

$n = 0$ ✗ 0 0 −2 0 −3
1/3 1/3 1/3 — —

$y[0] = -2/3 + 0/3 + 0/3 = -2/3 = -0.667$

$n = 1$ ✗ 0 −2 0 −3
1/3 1/3 1/3 —

$y[1] = 0 - 2/3 + 0 = -0.667$

$n = 2$ ✗ −2 0 −3
1/3 1/3 1/3

$y[2] = -3/3 + 0 - 2/3 = -1.667$

$n = 3$ ✗ −2 0 −3 0
— 1/3 1/3 1/3

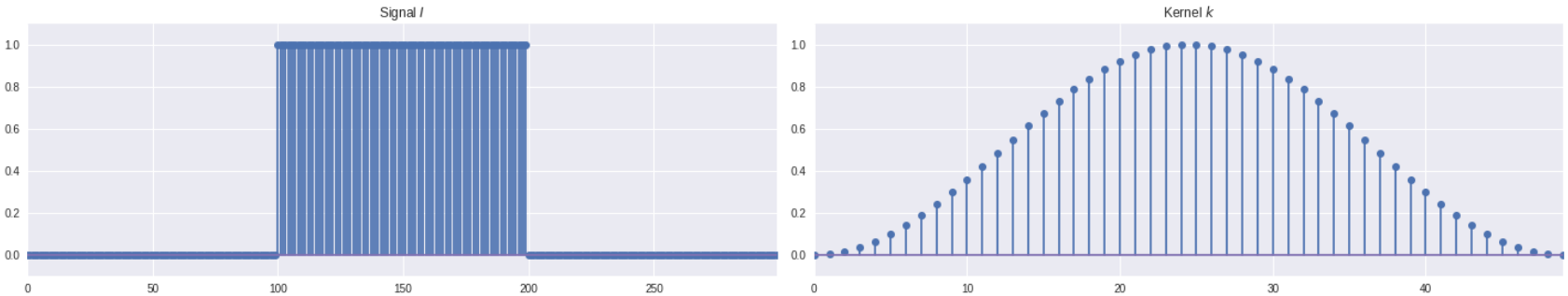
$y[3] = 0 - 3/3 + 0 = -1$

$n = 4$ ✗ −2 0 −3 0 0
— — 1/3 1/3 1/3

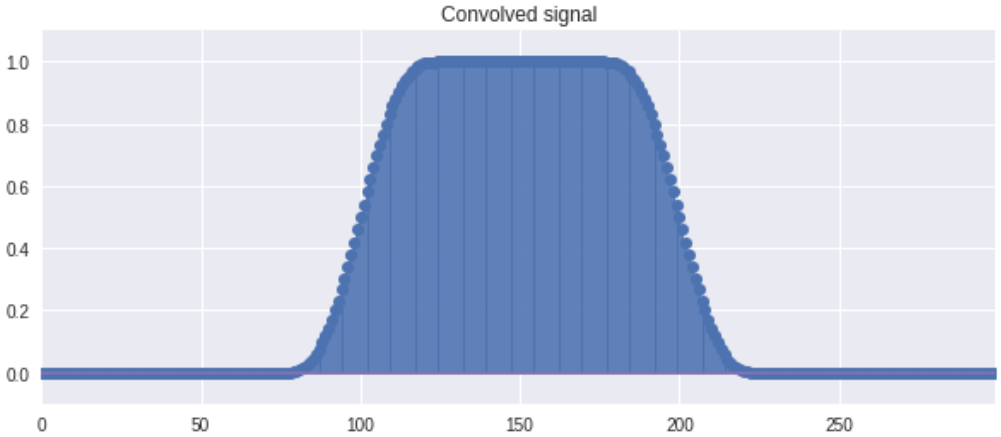
$y[4] = 0 + 0 - 3/3 = -1$

$y = [-0.667, -0.667, -1.667, -1, -1]$

Example, III



Q: What would it be the result of convolving these two signals?



2D signal

Images are a function of light with respect to space.

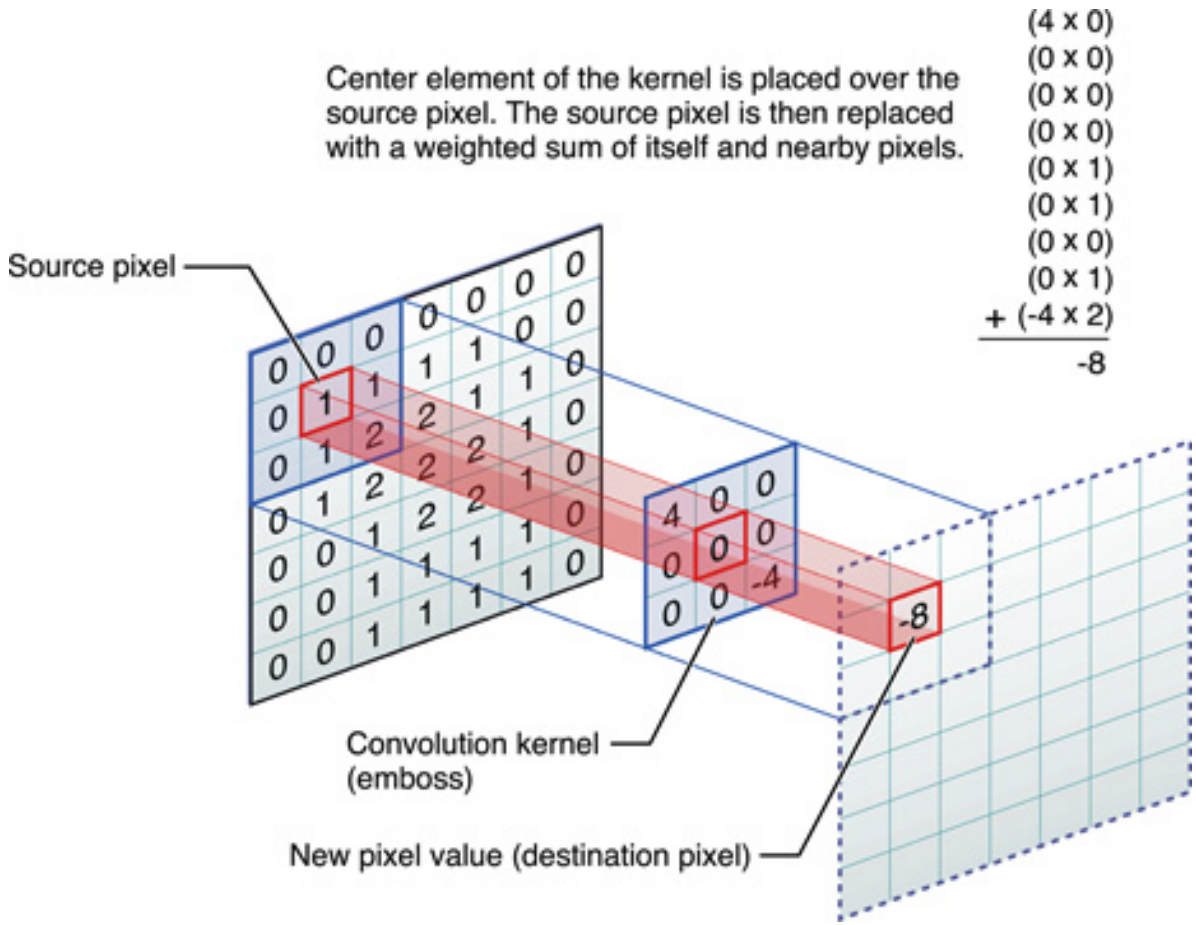
Common notation for images.

- I: image (signal).
- k: kernel (filter).
- $[x, y]$: indices in the horizontal and vertical directions.

Example: $I[x, y]$ refers to a pixel in the image.

$$(I * k)[x, y] = \sum_{i,j} I[x - i, y - j]k[i, j],$$

2D Images


Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
Convolution kernel (emboss)
New pixel value (destination pixel)

Padding

- Zeros: Infinite long signal with zeros everywhere it is not defined.
  $[5, 4, 2, 3, 7] \implies [\ldots, 0, 0, 5, 4, 2, 3, 7, 0, 0, \ldots]$
- Extended: Infinite long repeating image, only visible where defined.
  $[5, 4, 2, 3, 7] \implies [\ldots, 5, 5, 5, 4, 2, 3, 7, 7, 7, \ldots]$
- Cyclic: The sequence repeats itself over and over.
  $[5, 4, 2, 3, 7] \implies [\ldots, 3, 7, 5, 4, 2, 3, 7, 5, 4, \ldots]$
- Undefined: No computation is performed, or smaller result.
  $[5, 4, 2, 3, 7] \implies [?, ?, 5, 4, 2, 3, 7, ?, ?]$

Common way: Zero-padding of length equal to the kernel size.

Padding and resulting size

$$I[M \times N] * k[m \times n] \implies O[M - m + 1 + P \times N - n + 1 + P],$$

where P denotes the padding size.

Properties

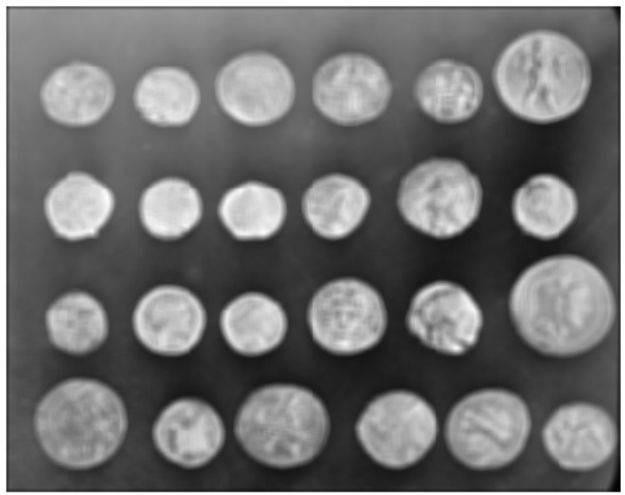
- Shift invariant.
- Linear.
- Associative.
- Commutative.
- No difference for symmetric filters.
- The Fourier transform of two convolved signals is the product of their individual Fourier transforms.

Properties

- Commutative: $f * g = g * f$.
- Associative: $f * (g * h) = (f * g) * h$.
- Distributive: $f * (g + h) = (f * g) + (f * h)$.
- Scaling: $a \cdot (f * g) = (a \cdot f) * g = f * (a \cdot g)$.
- Identity: $\delta * f = f$, where δ is the Dirac impulse.
- Convolution theorem: $\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$.

Smoothing (average)

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



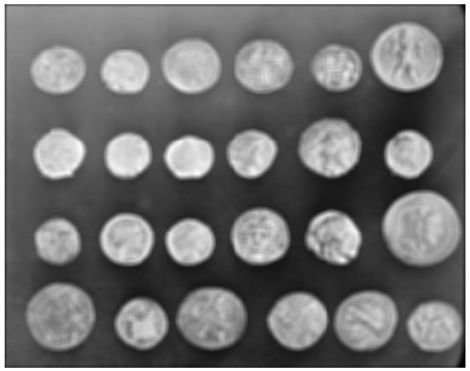
Identity

$$k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Sharpening

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

Original.

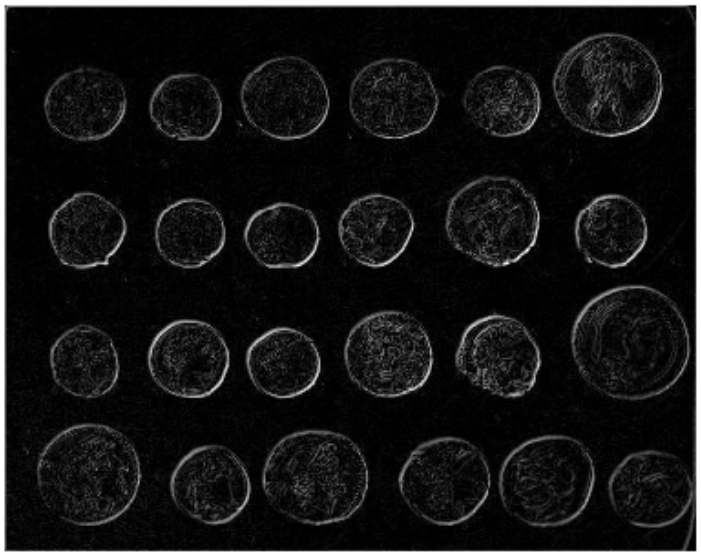


Smoothed out.

Sharpened

Edge detection

$$k = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



More applications

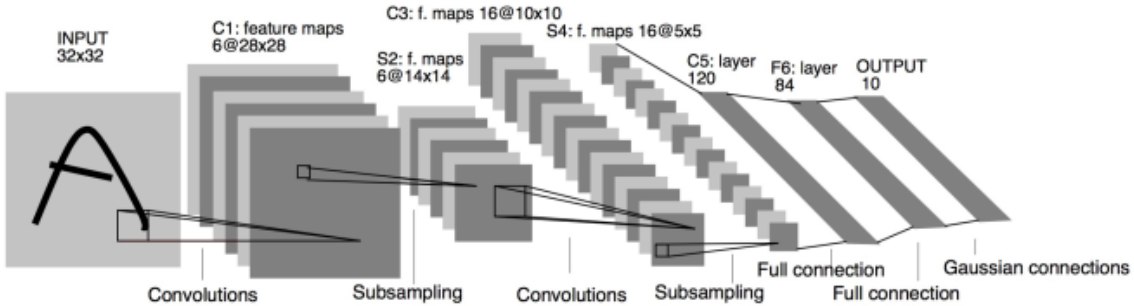
- **Statistics**: the probability distribution of the sum of two random variables is the convolution of each of their probability distributions.
- **Optics(1)**: a shadow is the convolution of a light source and the shape of an object whose shadow is being cast.
- **Optics(2)**: an out-of-focus photograph is the convolution of the sharp image and a blurring circle.
- **Acoustics**: the echo is the convolution of a sound source with a function representing objects reflecting that sound.
- **Signal processing**: the output of a stationary (time- or space-invariant) linear system is the convolution of the input with train of Dirac functions.

Outline

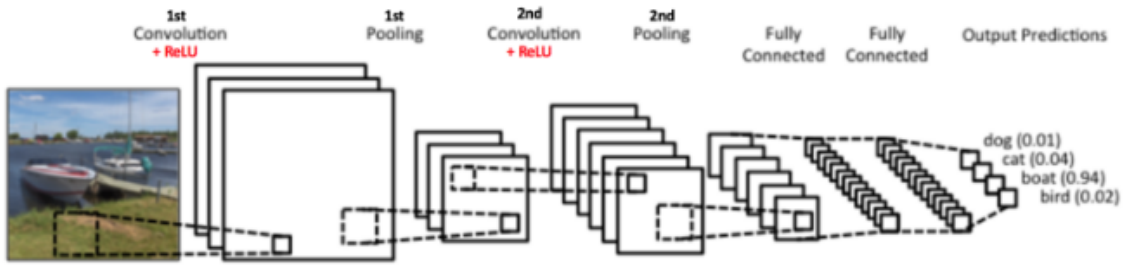
Deep Learning for images

Change hidden units from perceptrons to convolutional filters.

- Design no filter.
- Use backprop to learn these filters.
- Design the network.



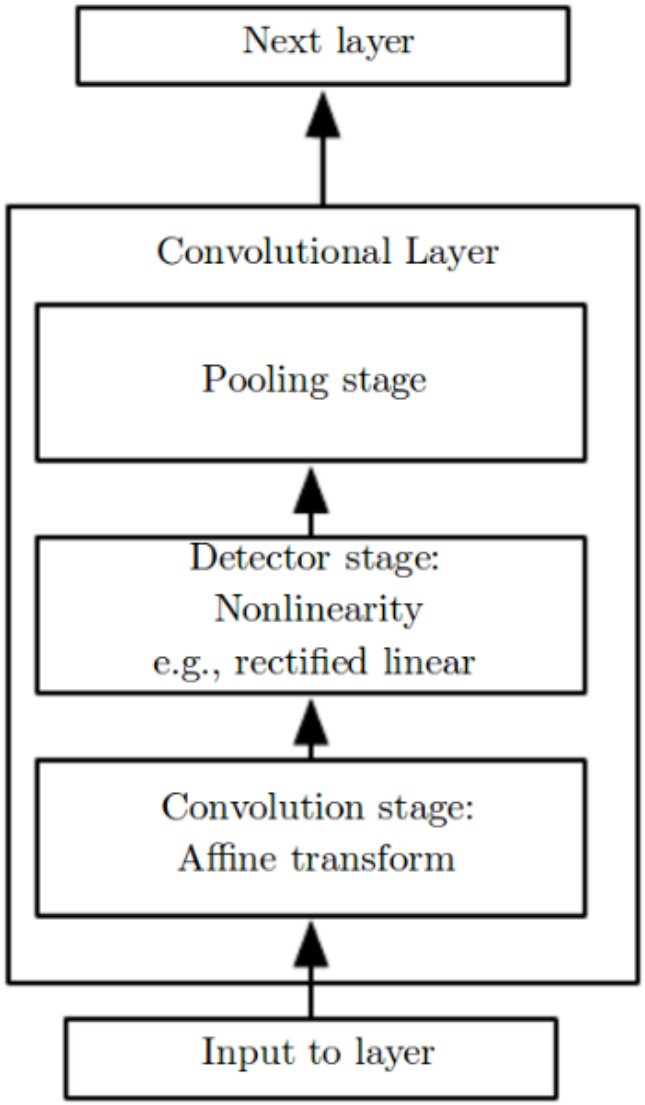
LeNet



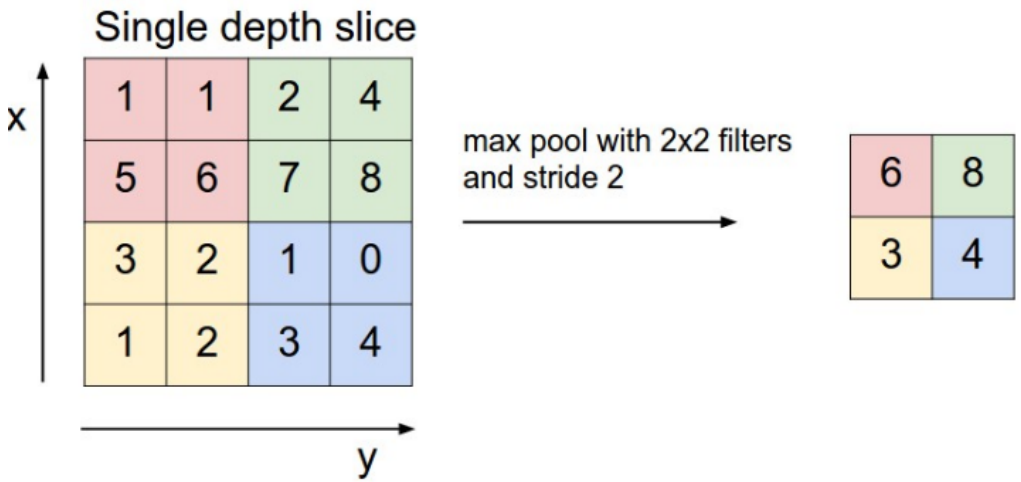
Convolutional unit

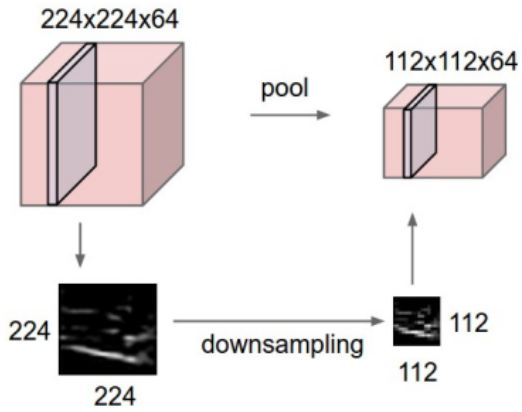
Using convolutions instead of perceptrons, we can exploit the spatial arrangement of visual information (pixels) in images.

Conv layer


Next layer
Convolutional Layer
Pooling stage
Detector stage:
Nonlinearity
e.g., rectified linear
Convolution stage:
Affine transform
Input to layer

Pooling


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4


224x224x64
pool
112x112x64
224
224
downsampling
112
112

Conv layer in numbers

Output shape

$$[W, H, C],$$

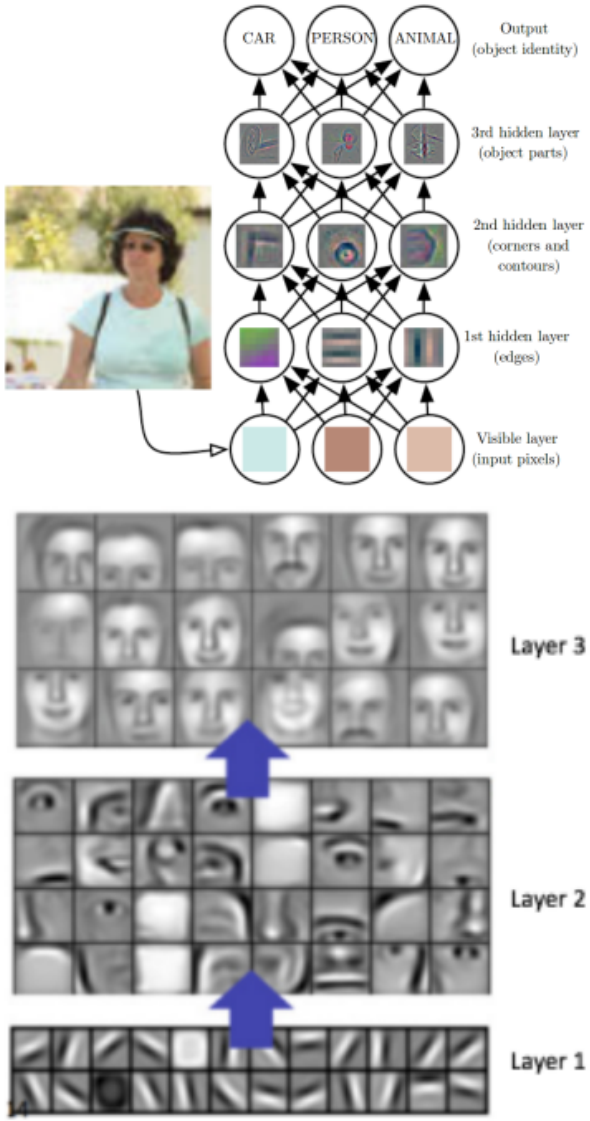
W : width,
H : height,
C : number of channels (number of filters in previous layer).

Number of parameters

$$n = N_f \times k_W \times k_H \times k_C + N_f,$$

N_f : number of convolutional filters (kernels)
(there is one bias per filter),
k_W, k_H, k_C : size of each filter (width, height, and channels).

Learning


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)
Layer 3
Layer 2
Layer 1

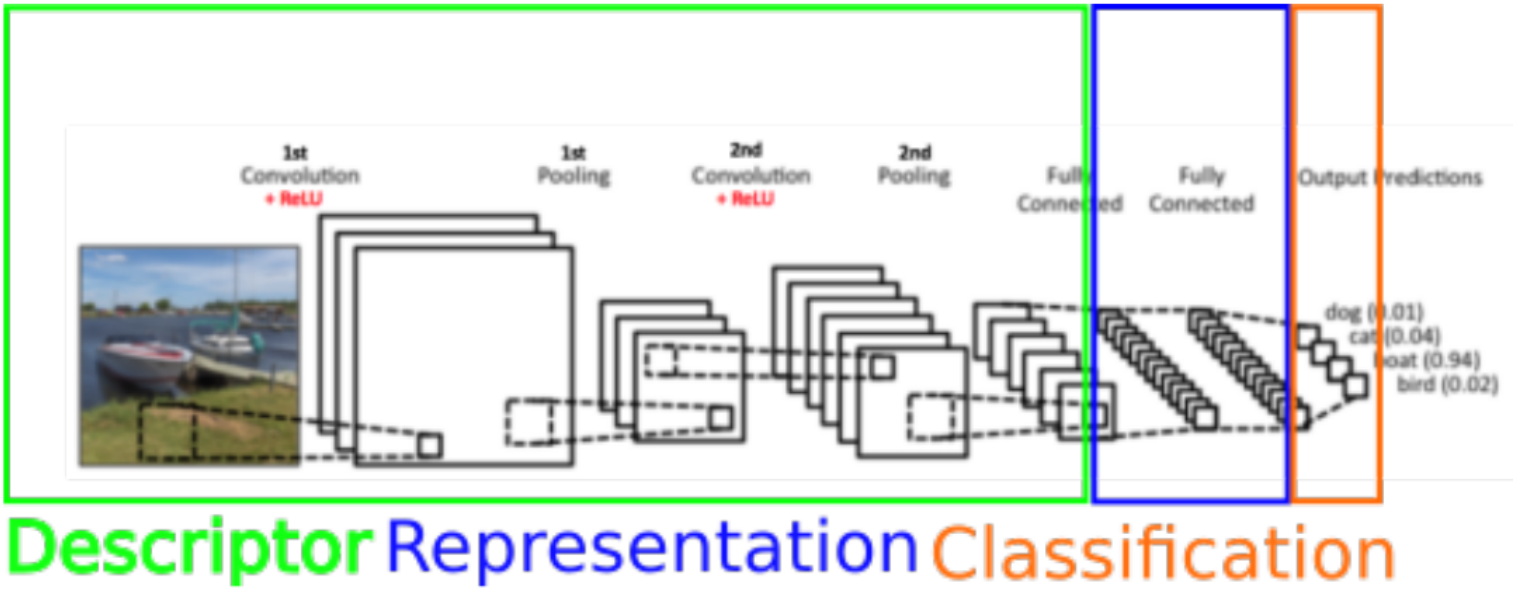
Common practices

- Input layer divisible by 2.
- Small filters, with odd size, e.g., 3×3 to 9×9.
- Zero padding.
- Stride $= 1$.
- Max pooling (2×2) vs stride $= 2$.
- ReLU transfer function.
- BatchNorm and Dropout as regularization.
- Increasing number of convolutional filters.
- Blocks.

Common blocks

- Conv + ReLU + MaxPool.
- Conv + BatchNorm + ReLU + MaxPool.
- Conv + ReLU + Conv + ReLU + MaxPool + Dropout.

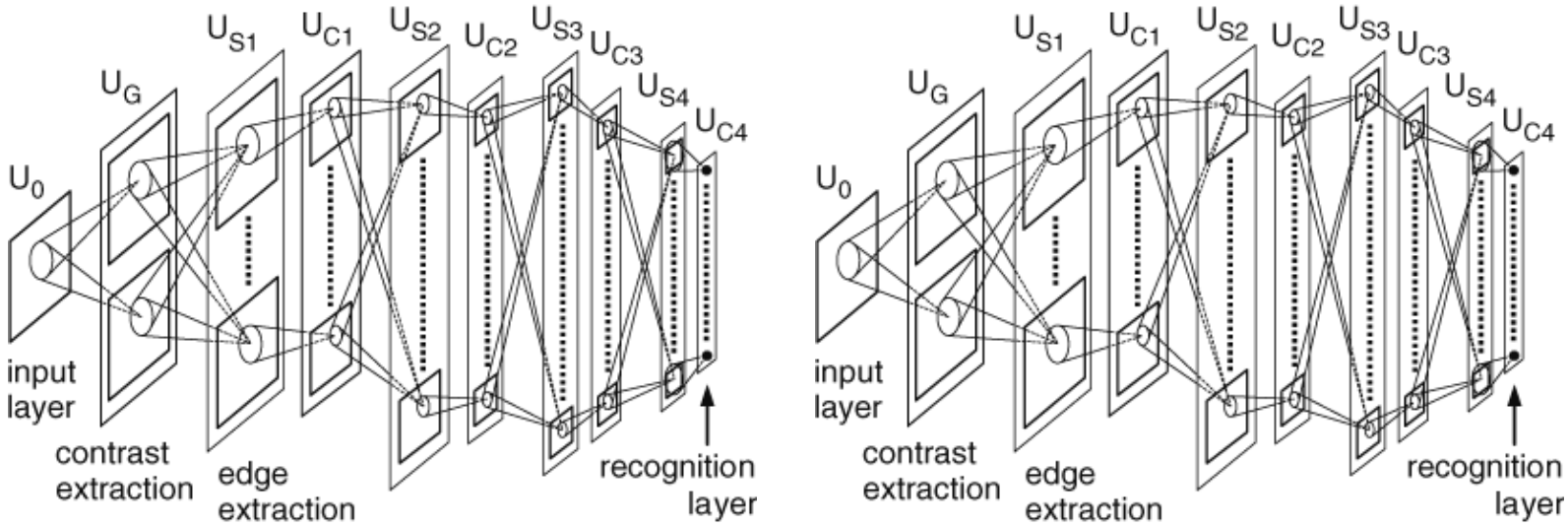
Data science point of view



Outline

Neocognitron

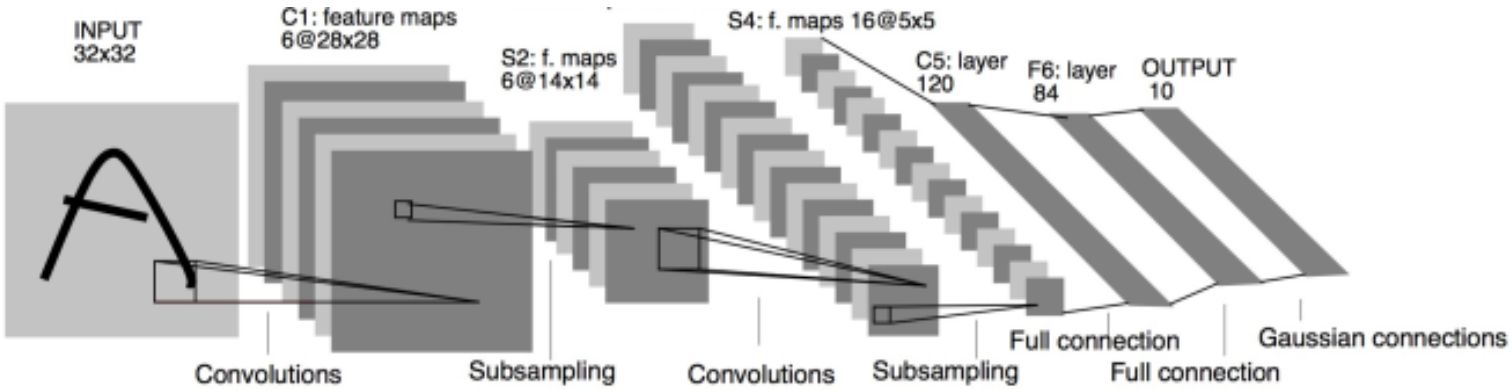
Fukushima, 1980.



- S-cell and C-cells.
- Sublayers, (cell-planes, i.e., same filter - different locations).
- Recognition layers.

LeNet5

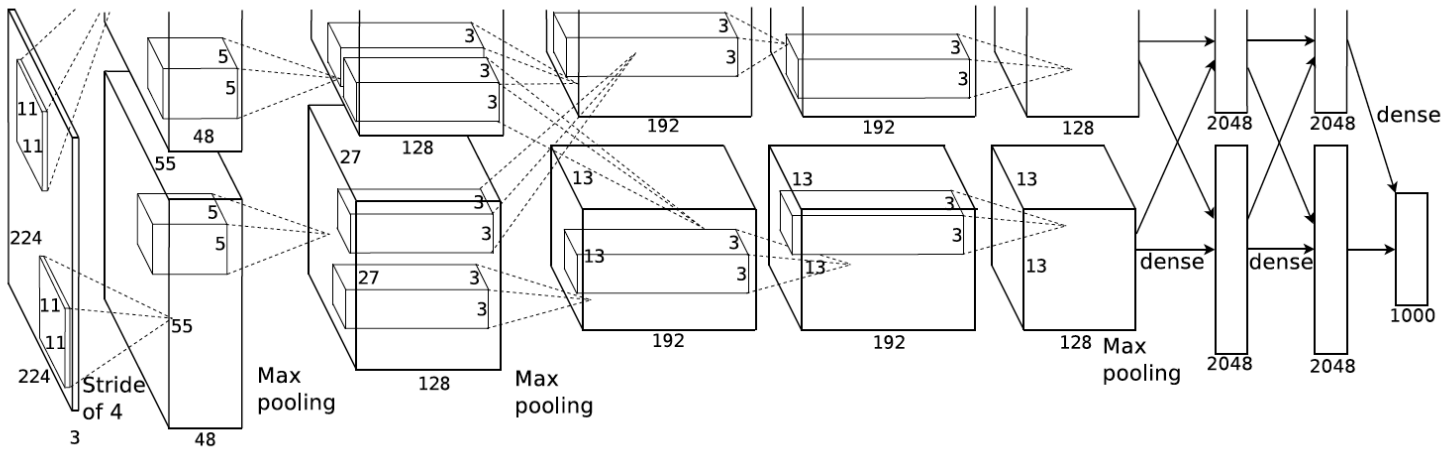
Yan LeCun et al., 1998.



- Convolution to exploit local correlations.
- Nonlinearity tanh or sigmoid.
- Multilayer perceptron for classification.
- Sparse connectivity between layers.
- Trained con CPU.

AlexNet

Alex Krizhevsky et al., 2012.

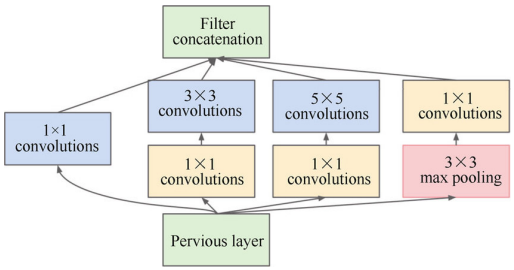


- Expanded LeNet.
- ReLU.
- Dropout.
- Max-pooling.
- Trained con GPUs.

Variants: ZFNet (ILSVRC 2013 winner), VGGNet (Depth analysis).

GoogLeNet (Inception Net)

Szegedy et al., 2014.

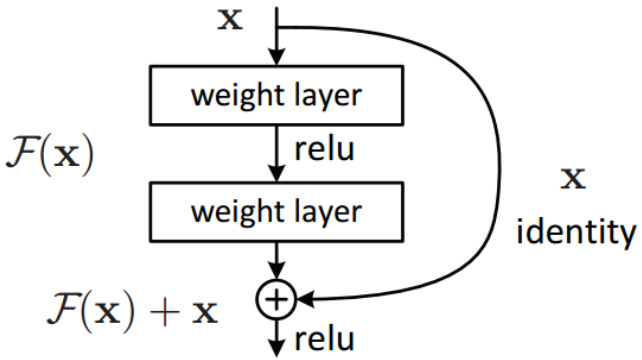


- ILSVRC 2014 winner.
- Inception module: reduce parameters (60M to 4M).
- Average Pooling instead of fully-connected layer.

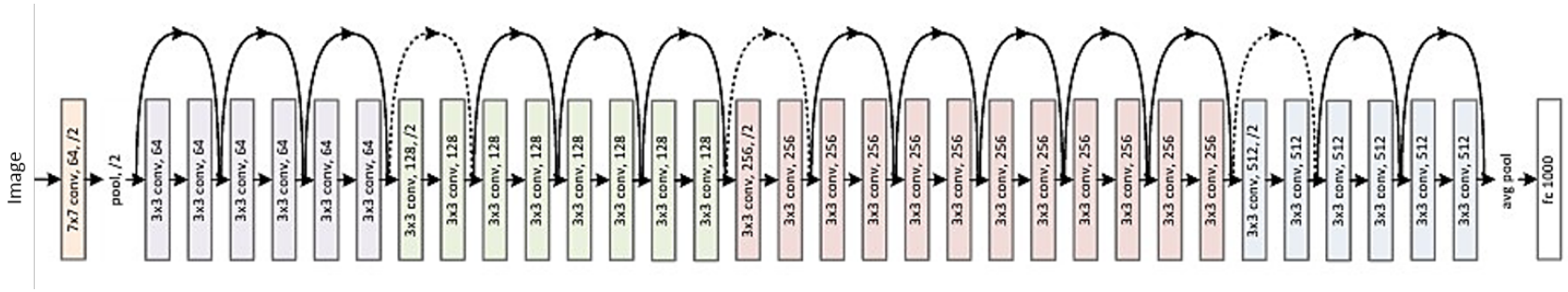
Variants of inception definition.

ResNet

He et al., 2015.

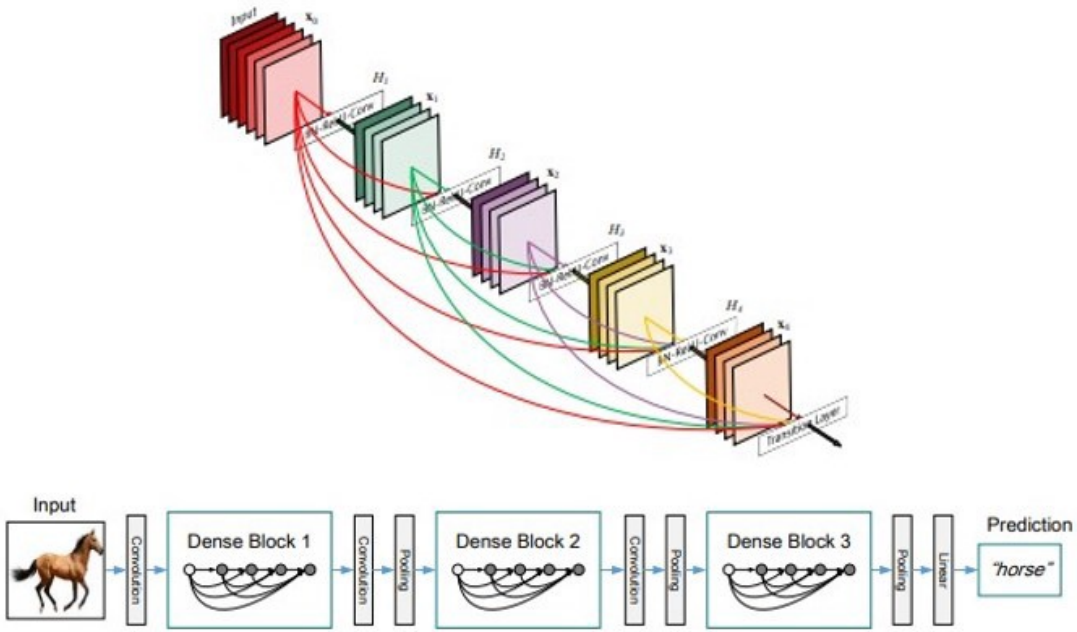


- ILSVRC 2015 winner.
- Residual learning (bypassing info).
- 1st network with $100+$ layers.



DenseNet

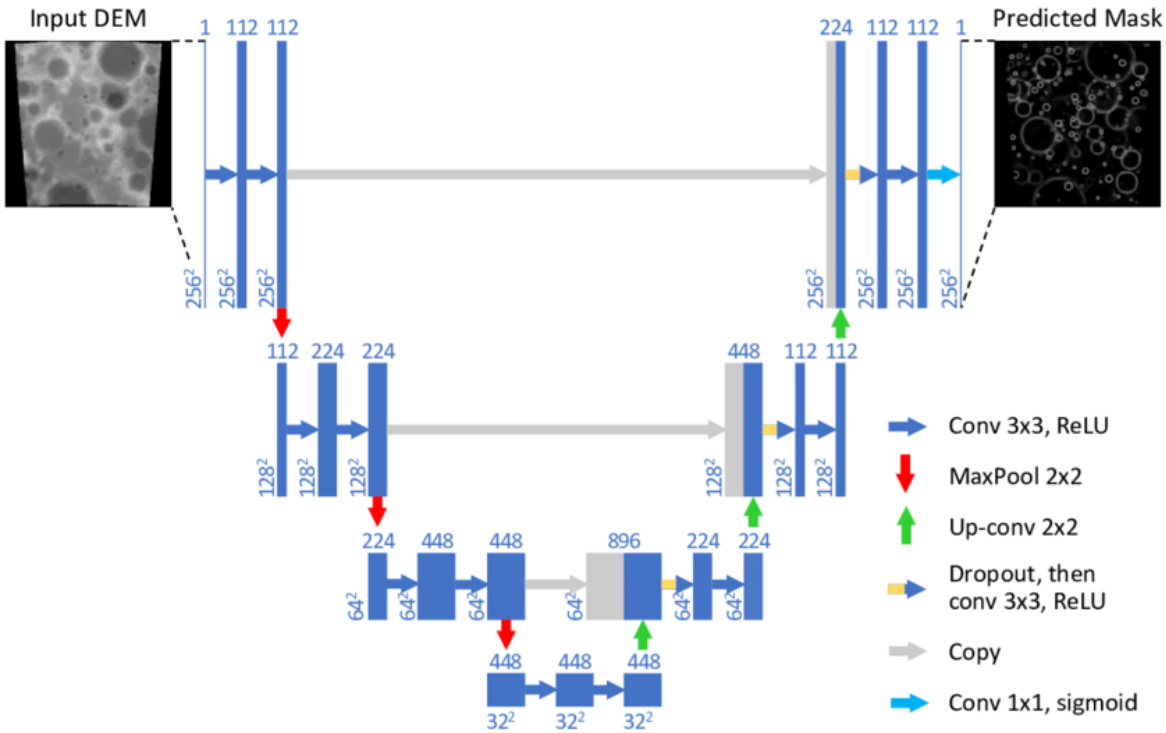
Huang et al., 2016.



- 2017 CVPR Best Paper Award.
- Improves residual connectivity with concatenation.
- Reduces vanishing gradients.

UNet

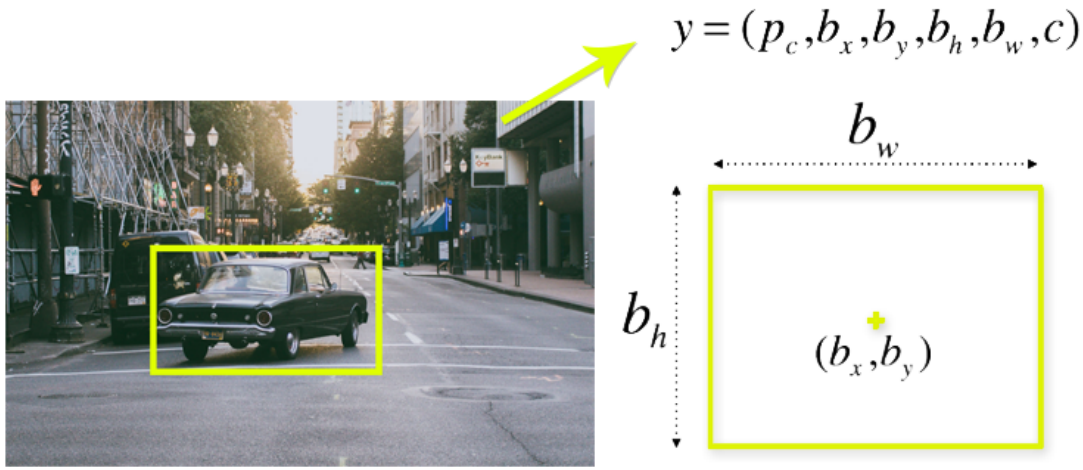
Ronneberger et al., 2015.



- For biomedical image segmentation.
- Fully convolutional.
- Contracting and expansive paths.

YOLO

Redmon et al., 2015 "You Only Look Once".



- Object detection.
- 45 fps in GPU.
- Avoids sliding windows.

YOLO v5, current version.

Q&A

Thank you!

edgar.roman@itam.mx