Convolutional Neural Networks

Rodrigo Gonzalez, PhD

Hello!

I am Rodrigo Gonzalez, PhD

You can find me at rodralez@gmail.com

Summary

1. Main ideas
2. Basic CNN architecture
3. Padding and stride
4. CNN in Keras

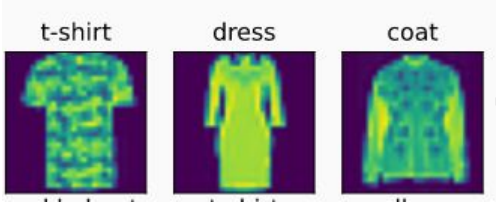
1.

Main ideas

The goals of a CNN

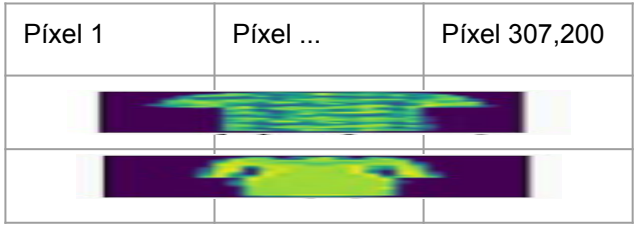
Common neural network approach

MATRIX



V

VECTOR



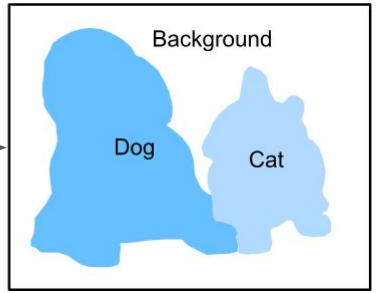
640 pixels x 480 pixels = 307,200 input neurons

CNN approach

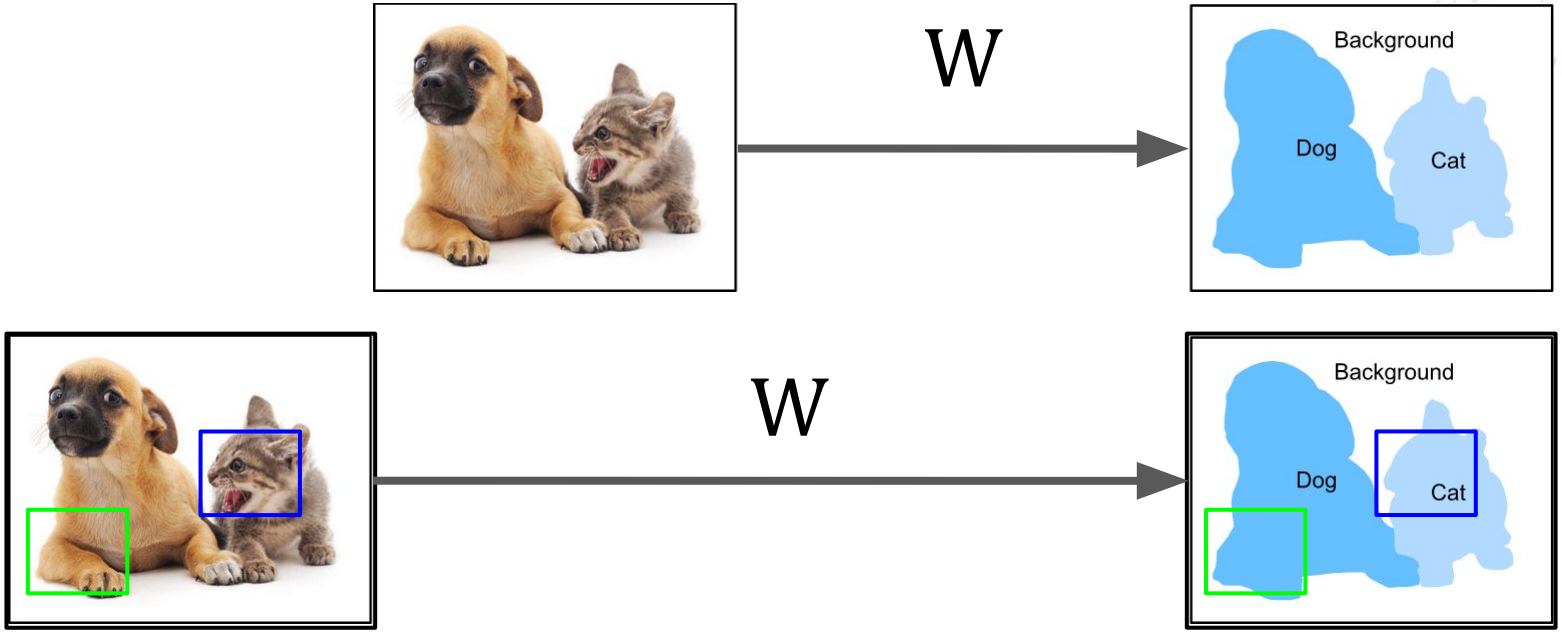
MATRIX

W

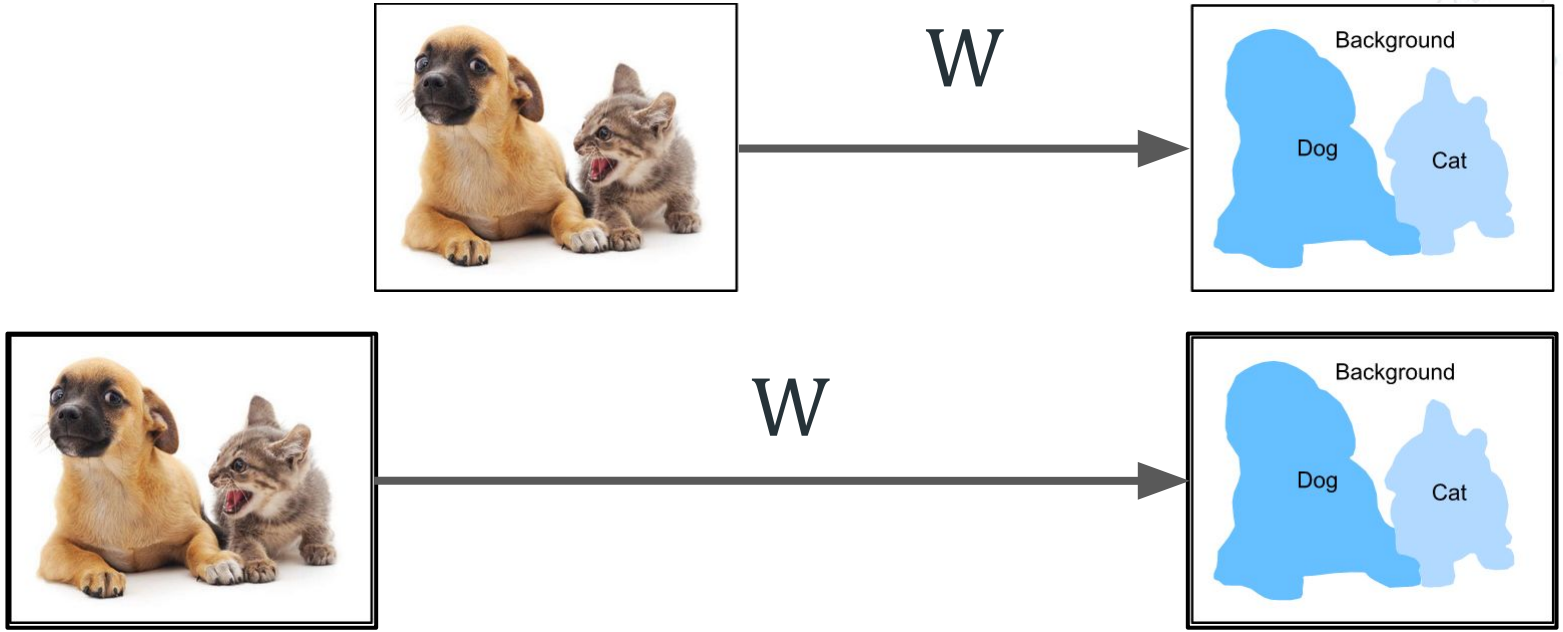
MATRIX



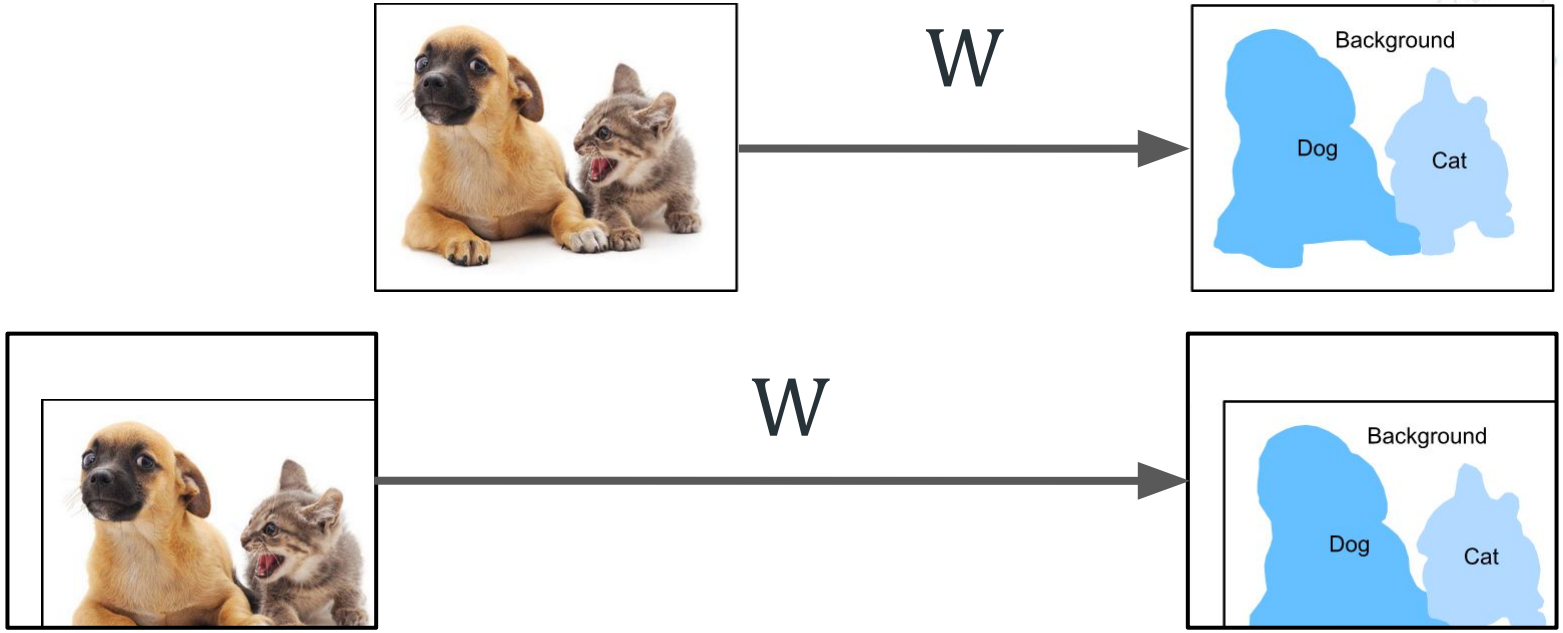
Locality



Translation invariance



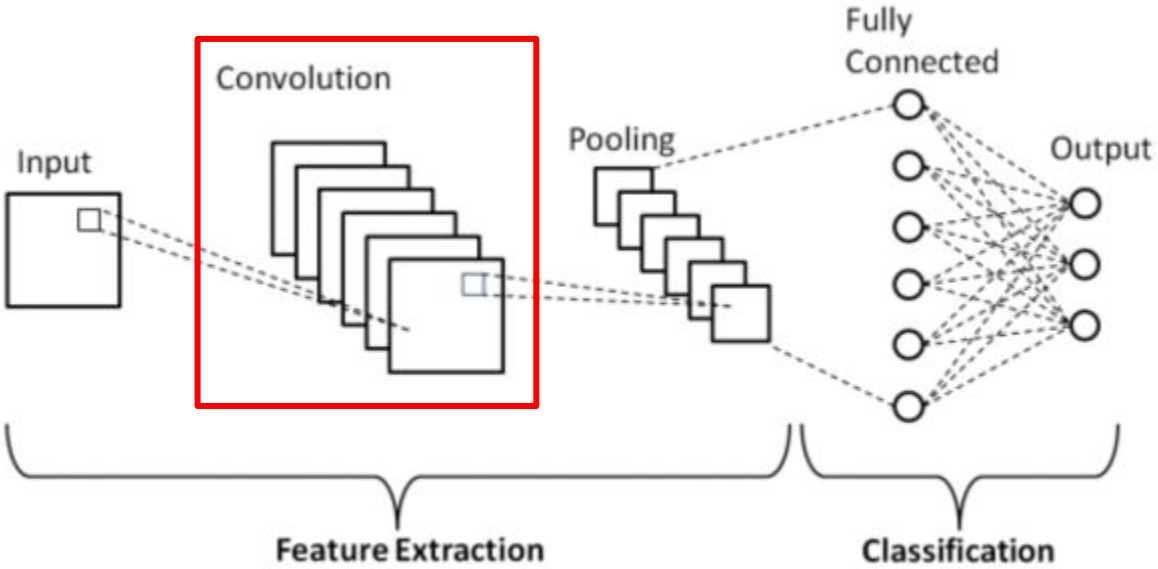
Translation invariance



2.
Basic CNN architecture

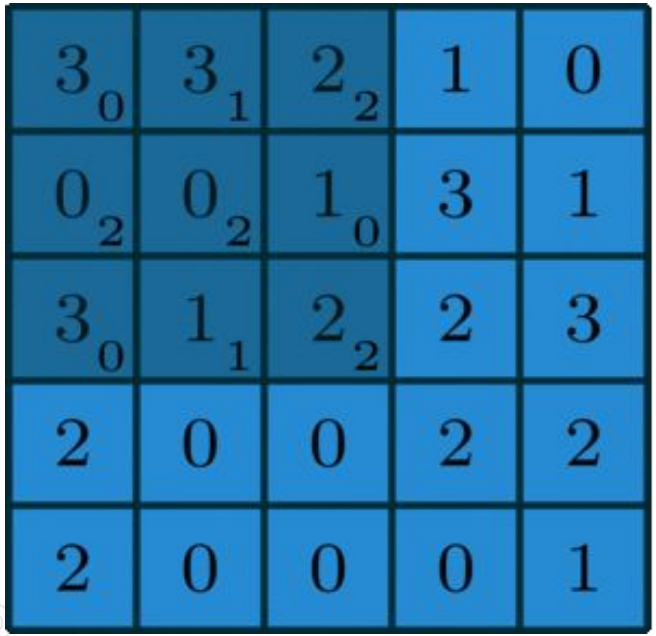
Kernel and pooling

Basic CNN architecture



Convolution

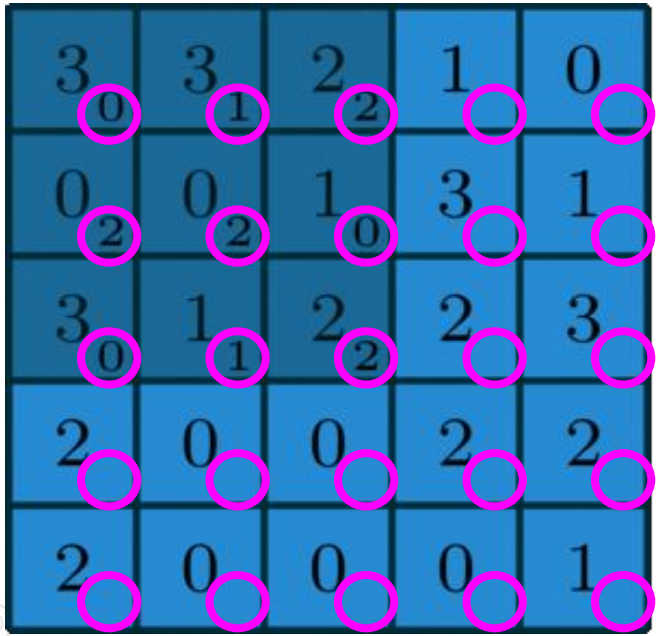
2D convolution is a **dot product** between an image (nxn matrix) and a **kernel** (3x3)



12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Convolution

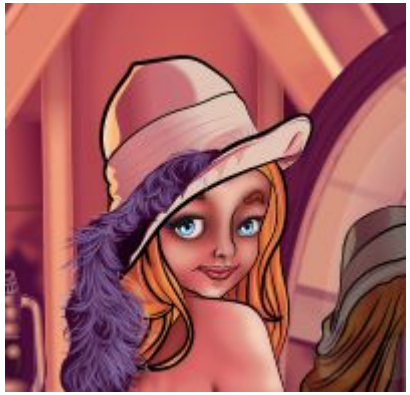
2D convolution is a **dot product** between an image (nxn matrix) and a **kernel** (3x3)



12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Common 2D convolution

Edge detector



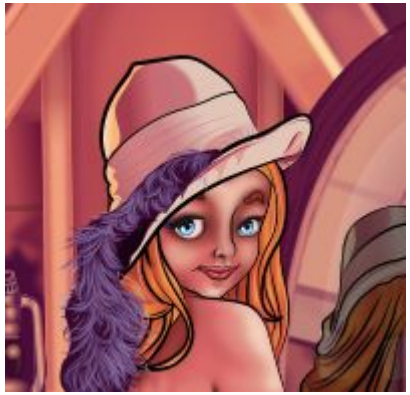
Kernel

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}$$

https://planetcalc.com/9313/

Common 2D convolution

Sharpening

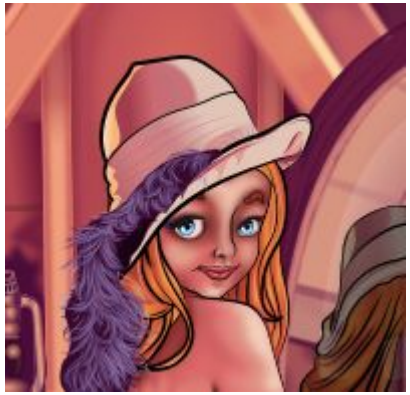


Kernel

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix}$$

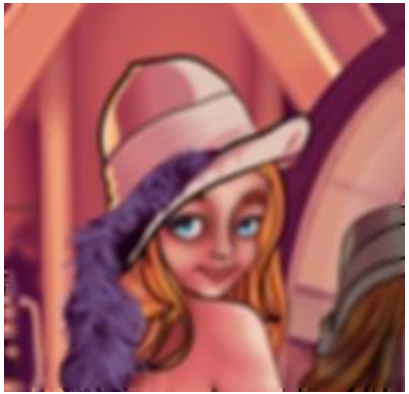
Common 2D convolution

Gaussian



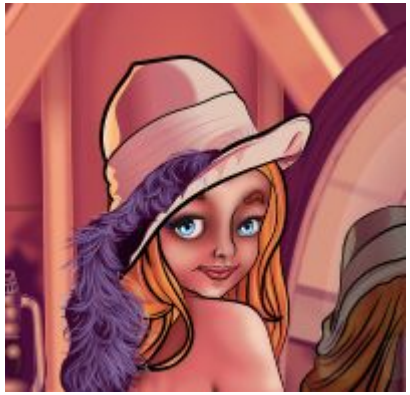
Kernel

1	2	3	2	1
2	4	5	4	2
3	5	6	5	3
2	4	5	4	2
1	2	3	2	1



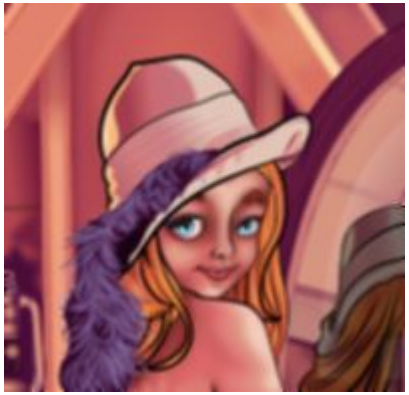
Common 2D convolution

Smoothing



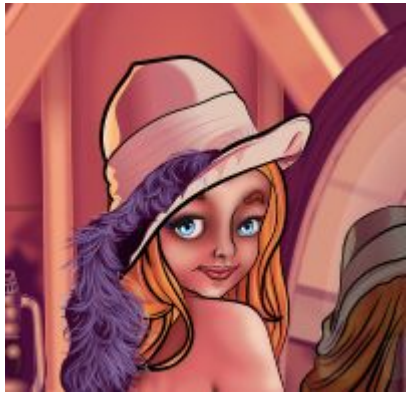
Kernel

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$



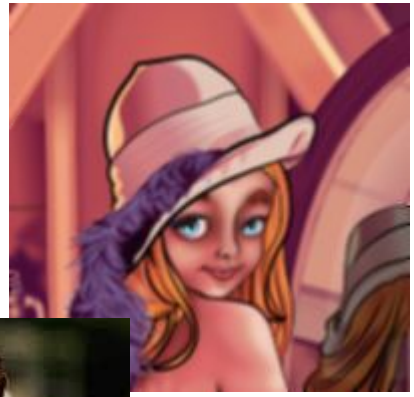
Common 2D convolution

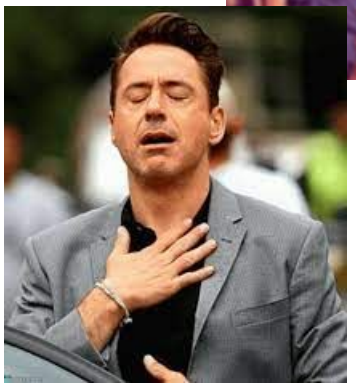
Smoothing



Kernel

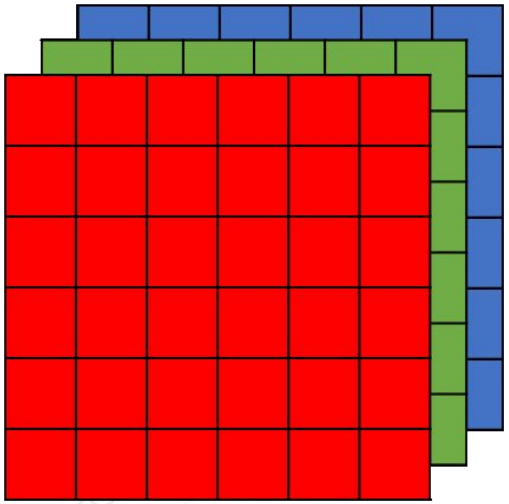
$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$





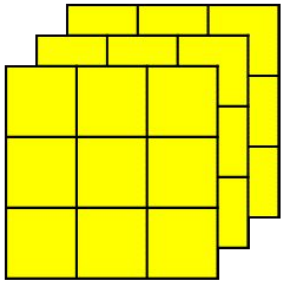
Don’t worry! The CNN will learn the kernels!

Convolutions on RGB image



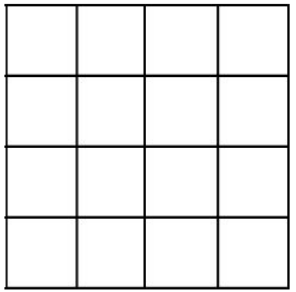
$6 \times 6 \times 3$

*



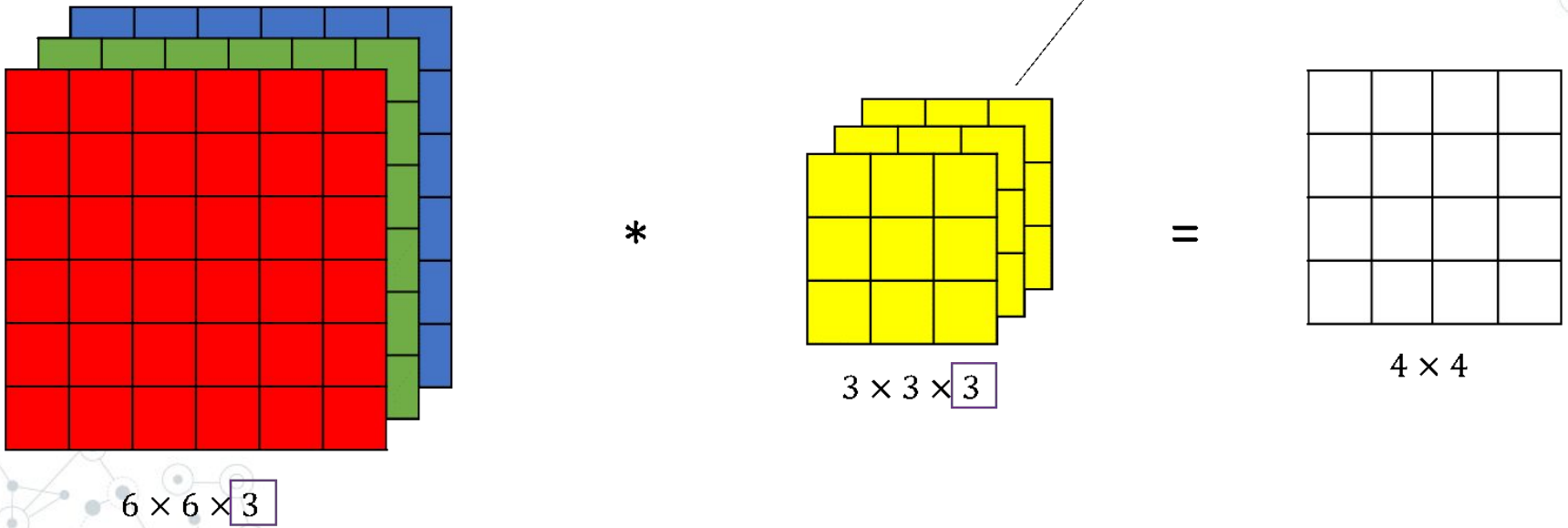
$3 \times 3 \times 3$

=

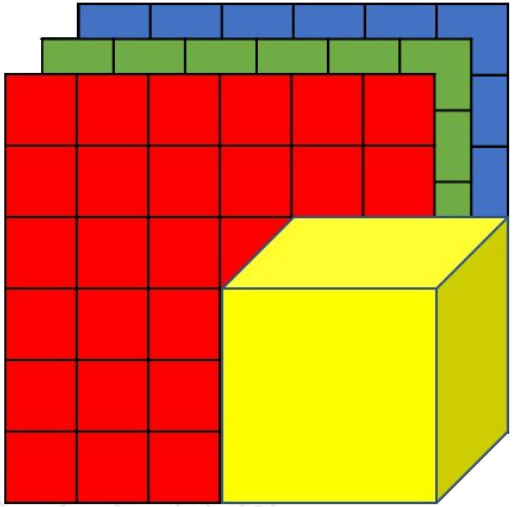


4×4

Convolutions on RGB image

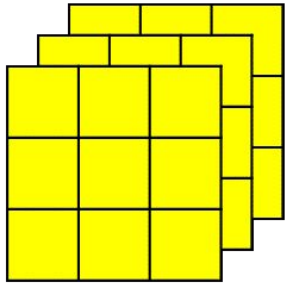


Convolutions on RGB image



$6 \times 6 \times 3$

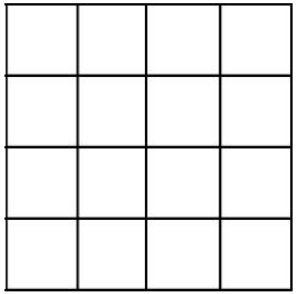
*



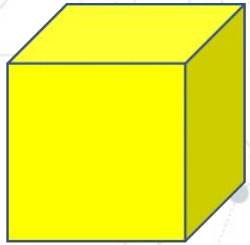
$3 \times 3 \times 3$

27 numbers

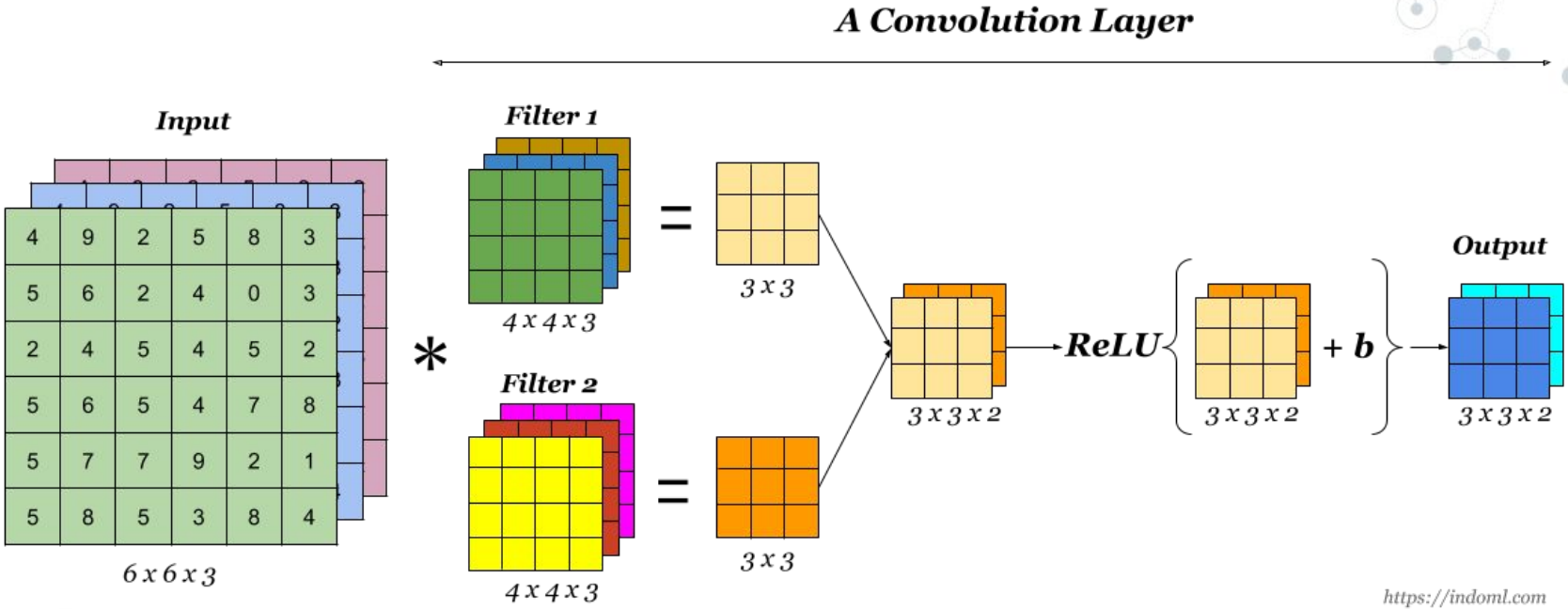
=



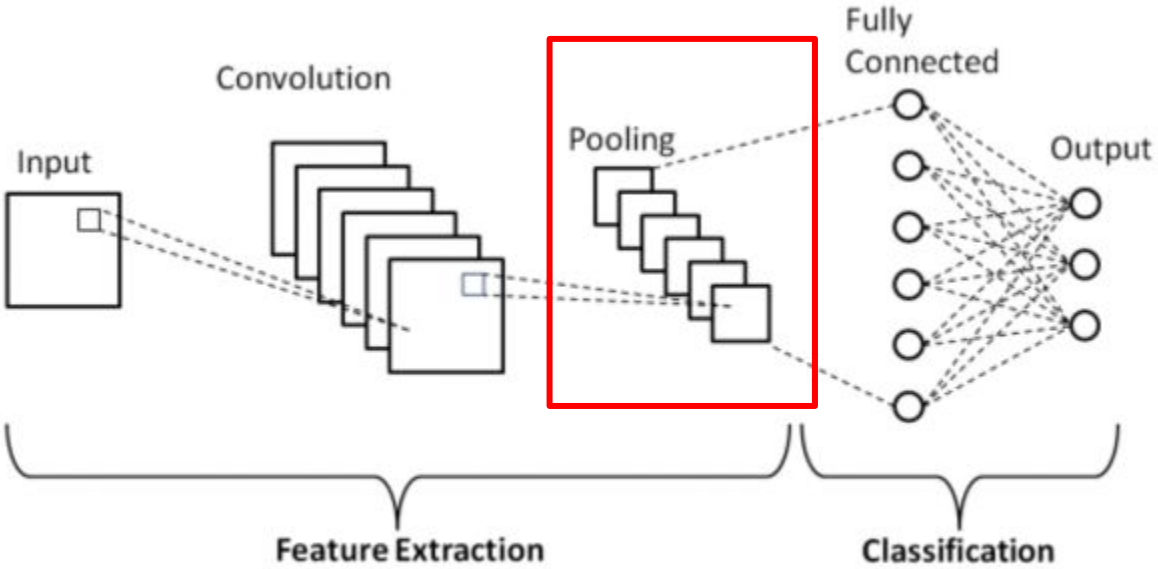
4×4



Convolutions on RGB image



Basic CNN architecture



Pooling

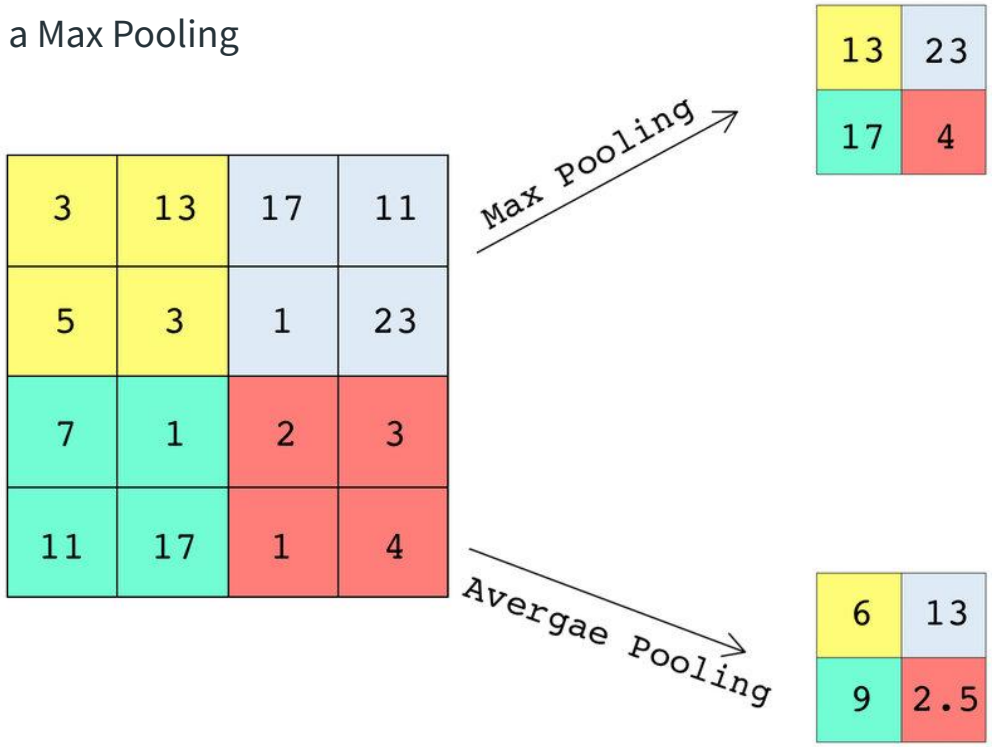
Pooling

Pooling

- Pooling is image compression
- Reduce computation

Max pooling and Average pooling

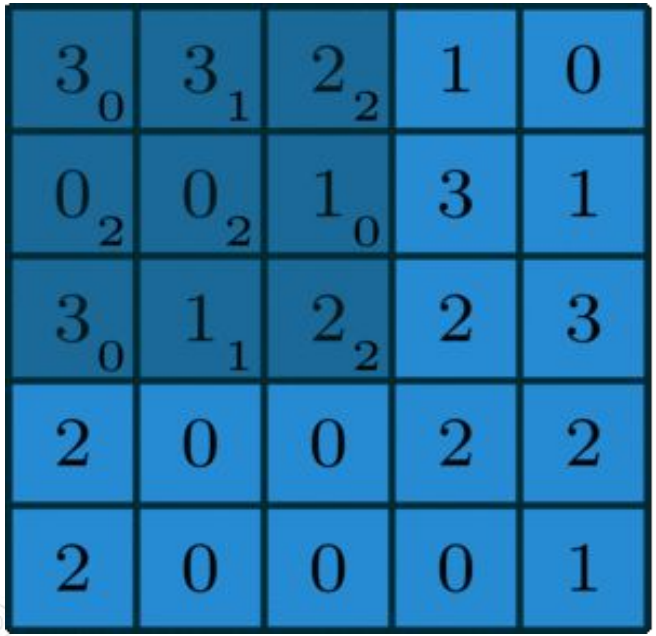
No need to learn a Max Pooling

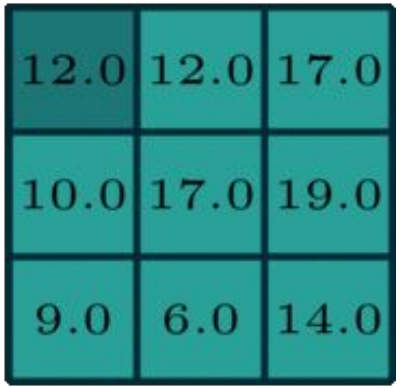


3.
Padding and stride

Padding

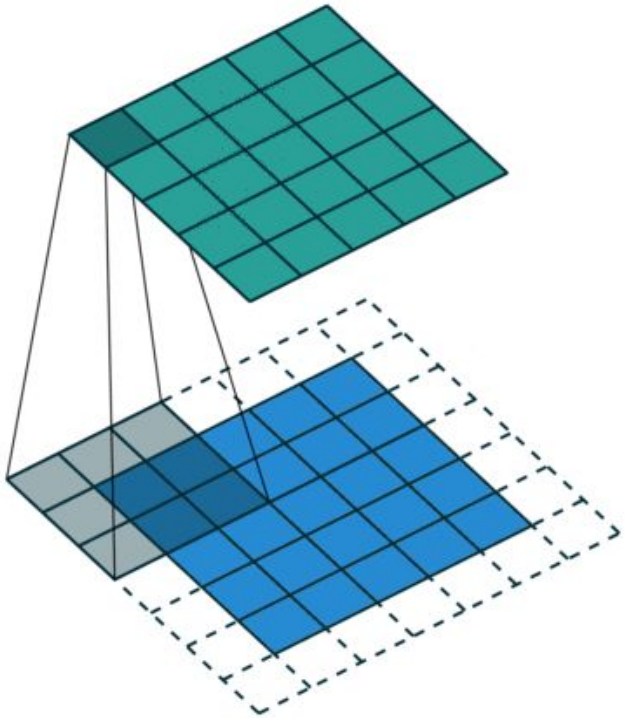
In general, a 2D convolution reduces the size of the image





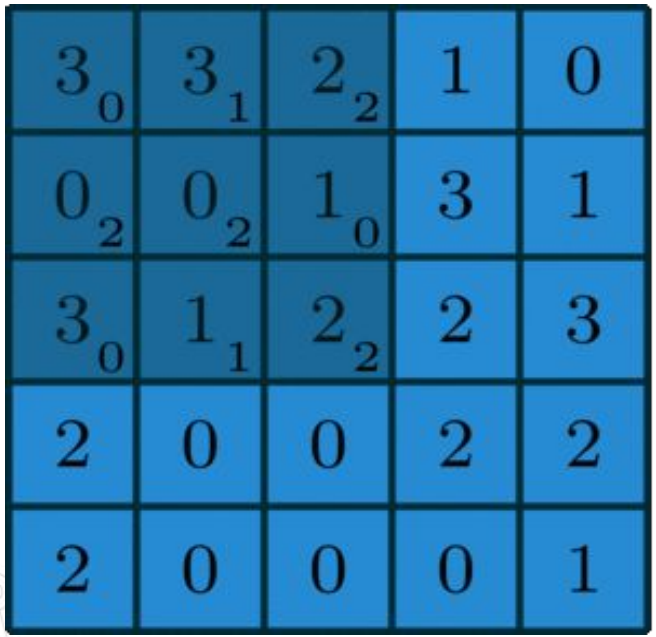
Padding

In general, a 2D convolution reduces the size of the image



Stride

Stride is the number of rows and columns traversed per slide. In general, a 2D convolution the kernel moves 1 row at a time in both directions.

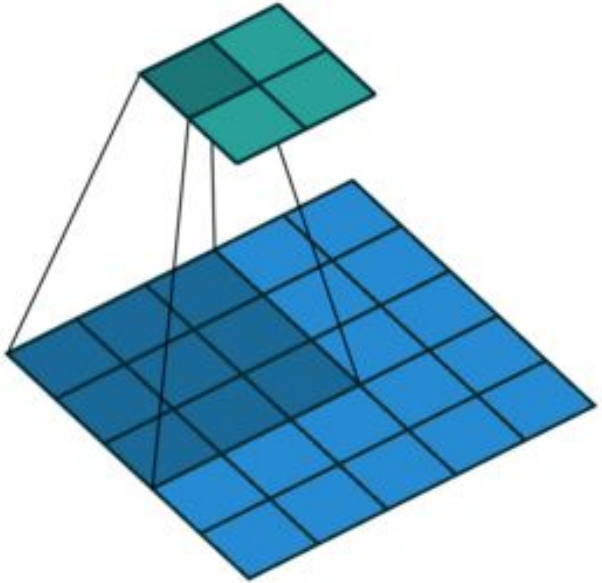


12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Stride

Stride of 2 vertically and 2 horizontally.

- Computational efficiency
- Downsampling



4.
CNN in Keras

Keras

Listing 8.1 Instantiating a small convnet

```
from tensorflow import keras
from tensorflow.keras import layers
inputs = keras.Input(shape=(28, 28, 1))
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(inputs)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.Flatten()(x)
outputs = layers.Dense(10, activation="softmax")(x)
model = keras.Model(inputs=inputs, outputs=outputs)
```

LeNet-5

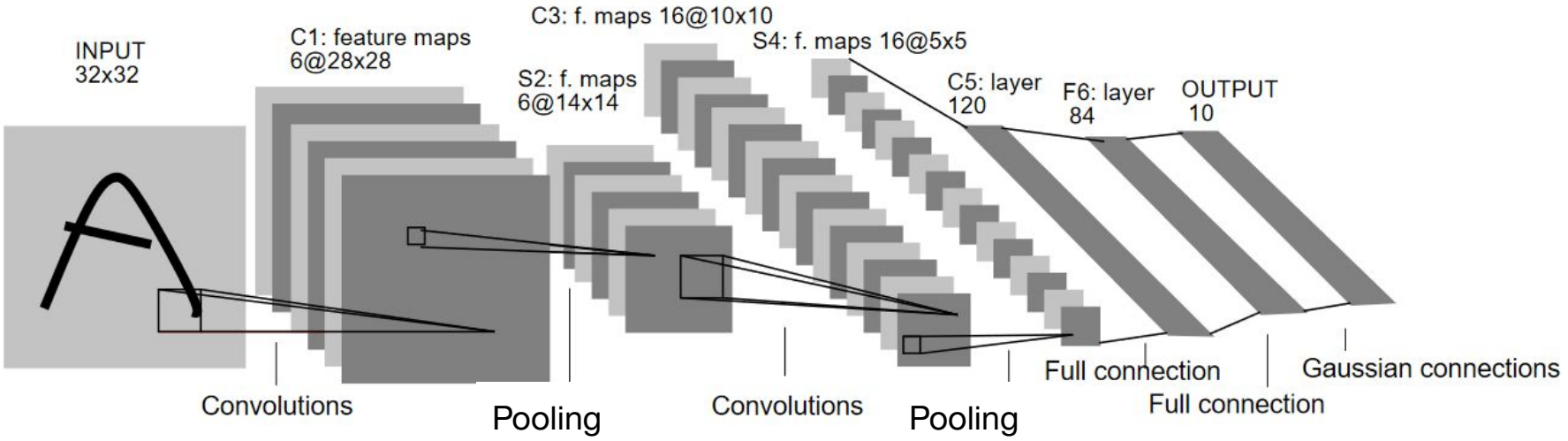
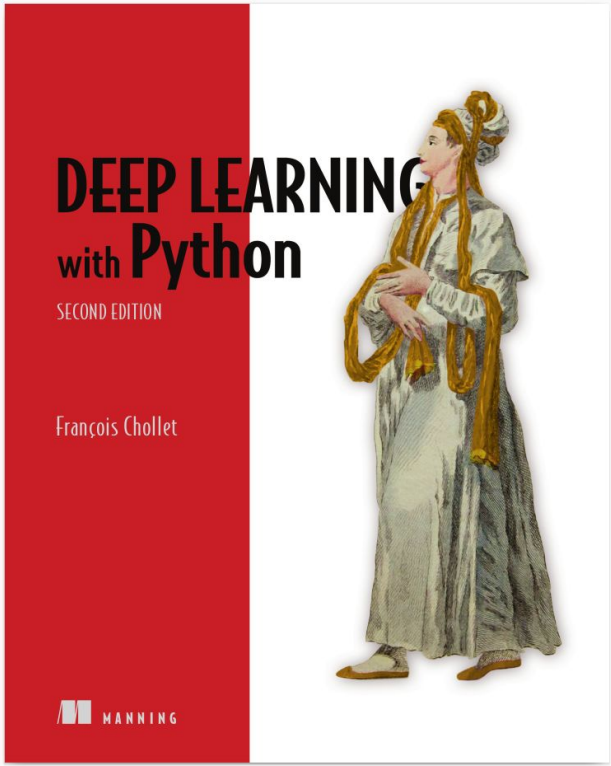


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: **Gradient-Based Learning Applied to Document Recognition**, *Proceedings of the IEEE, 86(11):2278-2324, November **1998**, \cite{lecun-98}*.

5.
Book

Deep Learning with Python, 2nd Ed. by Francois Chollet



- Chapter 8

Thanks!

Any questions?

You can find me at:

rodralez@gmail.com