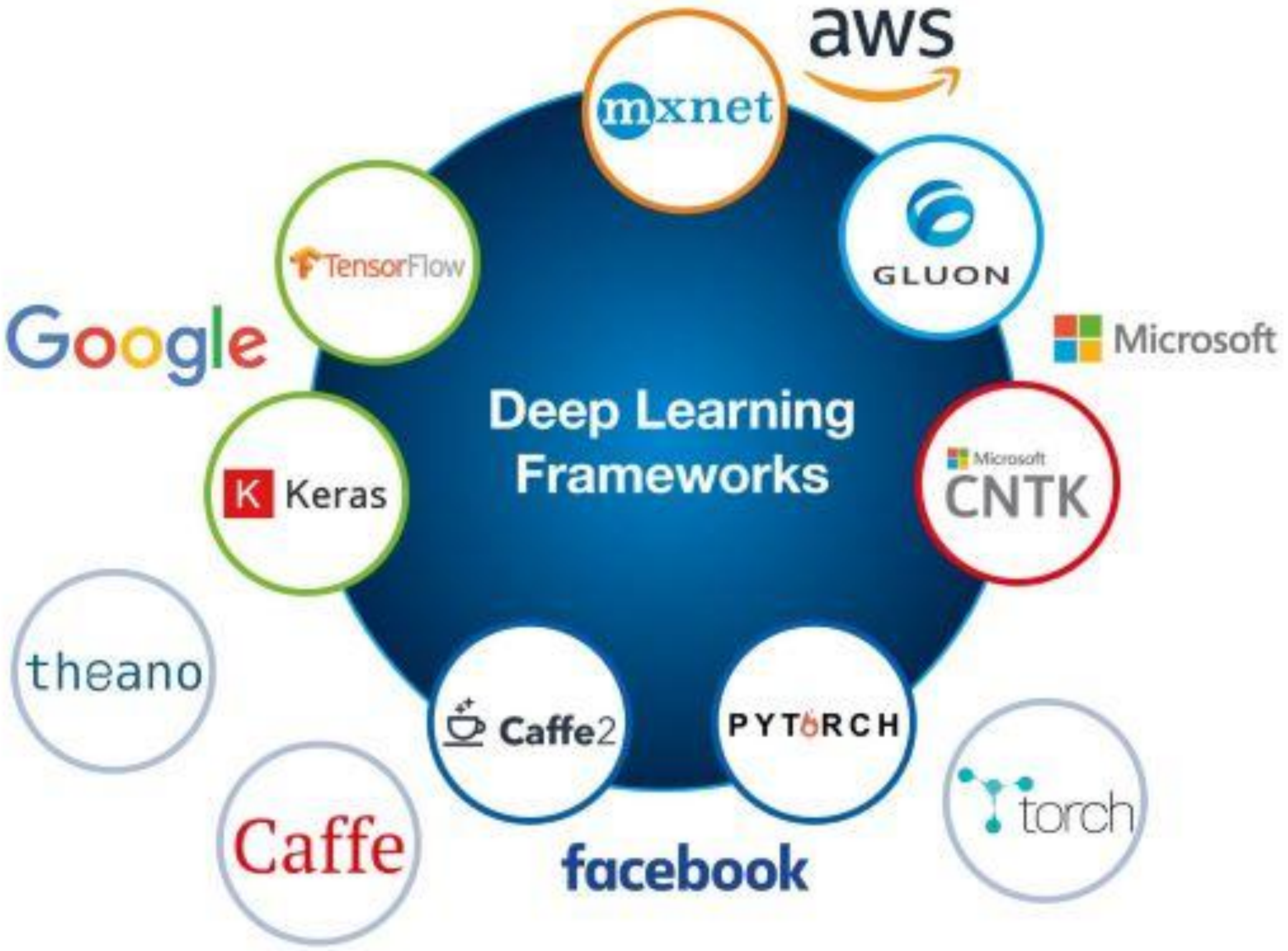
Deep Learning:
Keras and Tensorflow

Dr. Giuseppina Andresini

Deep Learning frameworks



Tensorflow

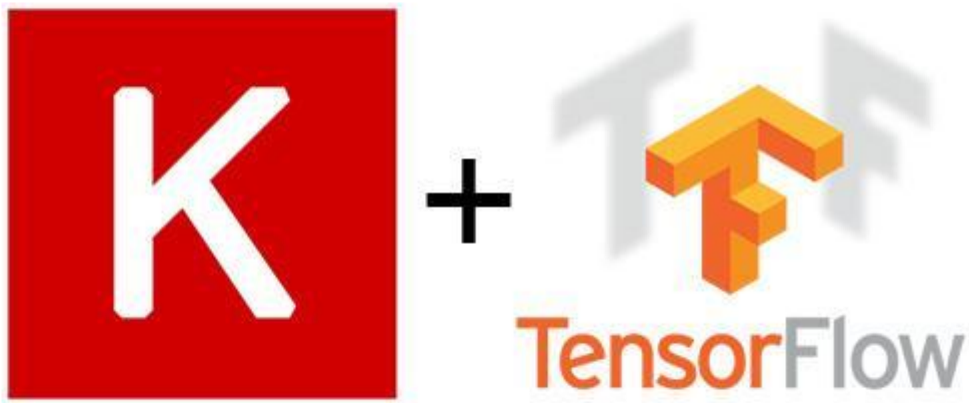
- Open-source library for Deep Neural Network
- Developed by **Google Brain team**
- First version developed in 2015; version 2.0 in 2018
- Can be used in a wide variety of programming languages:
  - **Python**, Javascript, C++, and Java
- Supports running computations on a variety of types of devices:
  - including CPU and GPU
- Basic type:
  - Tensor: multi-dimensional arrays with a uniform type, (kind of) like ***np.arrays***

https://www.tensorflow.org/

Keras

- Open-source library that provide an interface for DL architectures
- High-level API written in Python:
  - Running on top of **TensorFlow** platform
- Provides modular abstractions and building blocks for developing artificial neural networks:
  - contains implementations of commonly neural-network building blocks (e.g., Convolutional layer, Dense layer), activation functions, loss function.

https://keras.io/



Keras the functional API

- The Keras *functional API* is a way to create models
  - more flexible than the tf.keras.Sequential API.
  - The functional API can handle models with non-linear topology, shared layers, and even multiple inputs or outputs.
- The main idea is that a deep learning model is usually a **directed acyclic graph (DAG) of layers.**
  - the functional API builds *graphs of layers*.

```
model = Sequential()

model.add(Dense(4,activation='relu'))

model.add(Dense(4,activation='relu'))
```

```
input_layer = Input(shape=(3,))
Layer_1 = Dense(4, activation="relu")(input_layer)
Layer_2 = Dense(4, activation="relu")(Layer_1)
```

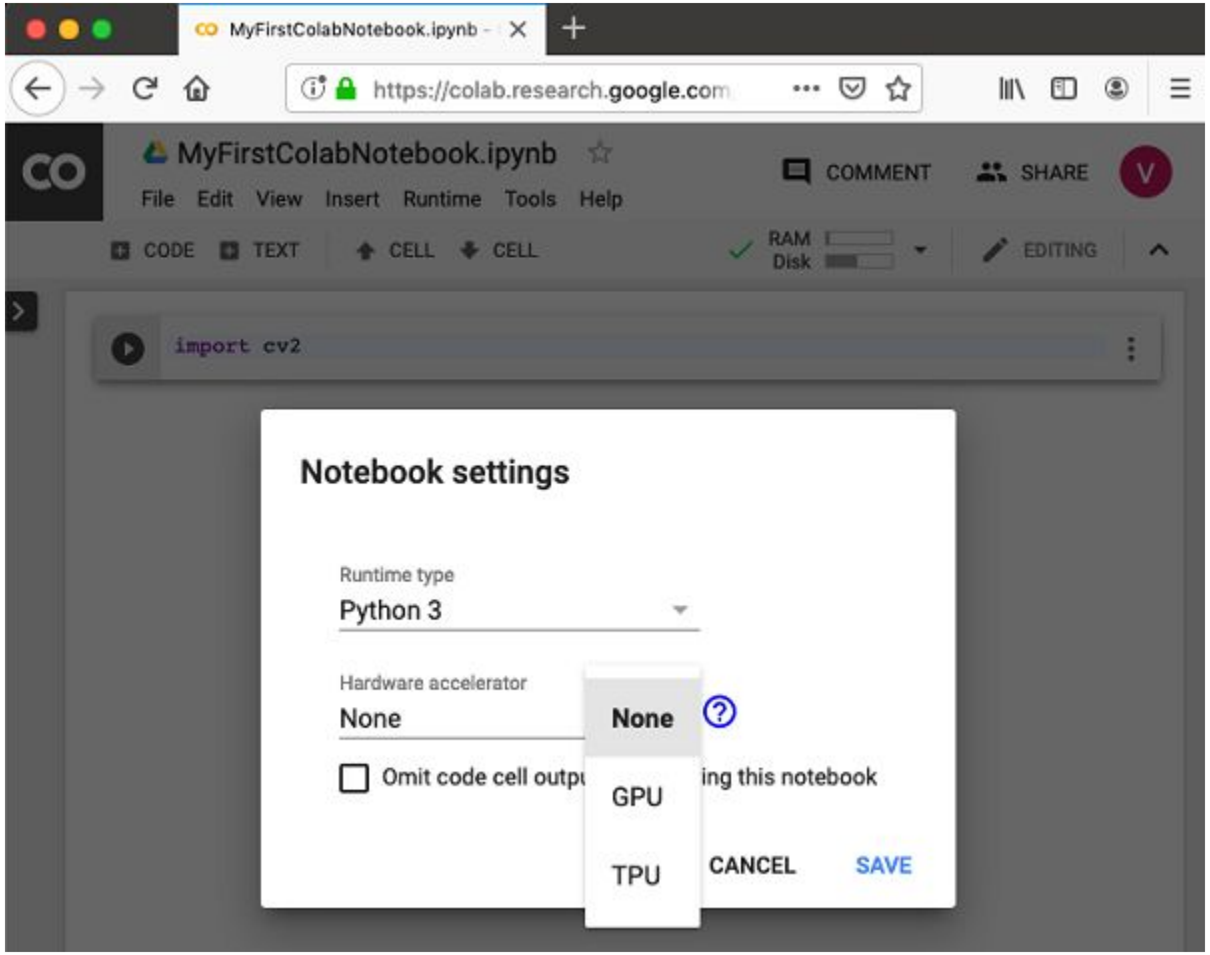
Google Colab

- Free platform from Google that allows users to code in Python.
- Colab is essentially the Google version of a Jupyter Notebook.
- Advantages:
    - zero configuration
    - free access to GPUs & CPUs
    - sharing of code.

https://colab.research.google.com/

Google Colab:GPU

- Edit
- Notebook settings



MyFirstColabNotebook.ipynb
https://colab.research.google.com
CO
MyFirstColabNotebook.ipynb
File Edit View Insert Runtime Tools Help
COMMENT
SHARE
V
CODE TEXT
CELL CELL
RAM
Disk
EDITING

```
import cv2
```

Notebook settings

Runtime type
Python 3

Hardware accelerator
None

None
GPU
TPU

□ Omit code cell outp ing this notebook

CANCEL SAVE

https://colab.research.google.com/

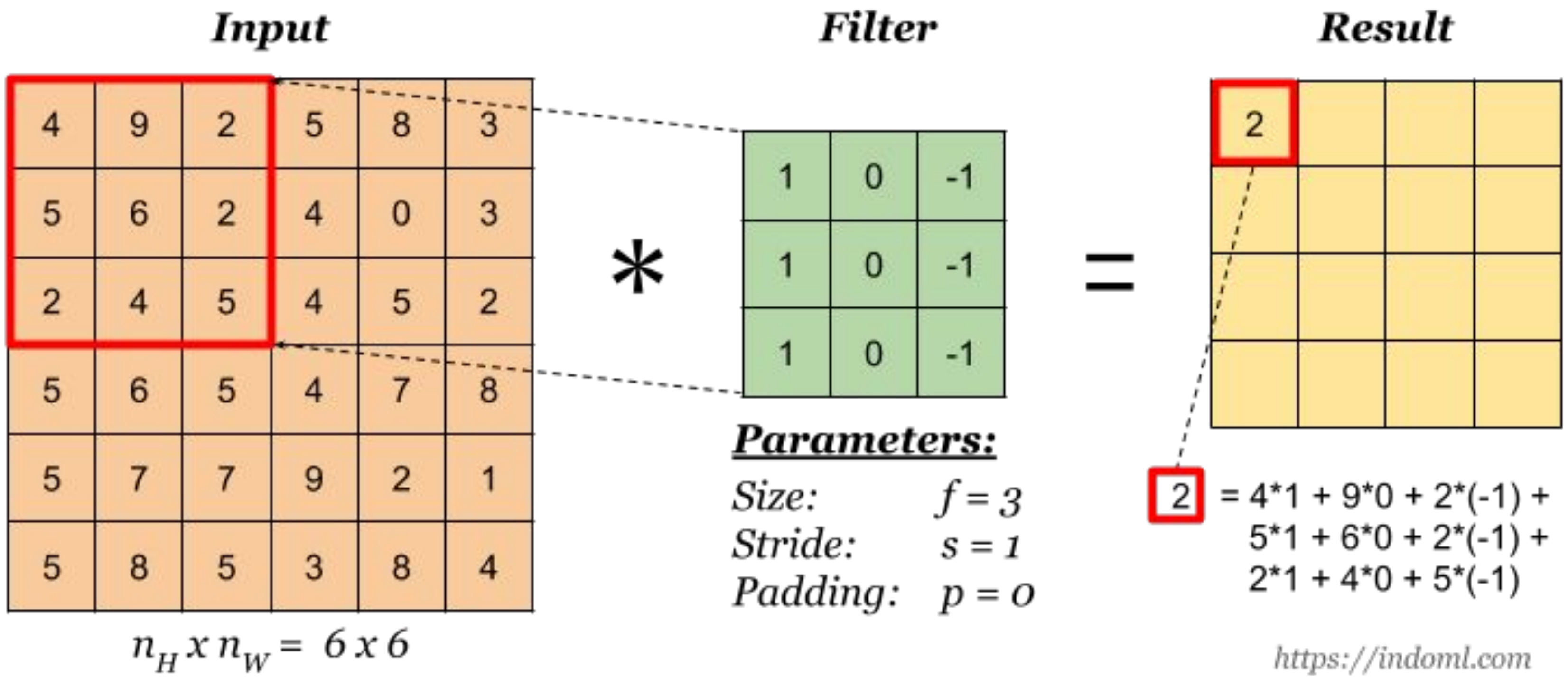
Exercise 1: CNN

Classification of MNIST dataset:

- 1 CNN
  - 3 Convolutional layers with padding = 0 and stride =1
  - 16,32,64 neurons
  - A MaxPooling layer
  - A Dropout layer
  - A Dense layer with 128 neurons
- Creation of adversarial examples using Adversarial Robustness Toolbox library
  - https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/evasion.html

CNNs: filters

- ***A transformation filter***:
  - used to learn the shared information in image pixels
  - the position define a spatial filter (feature)



Input

4	9	2	5	8	3
5	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

$n_H \, x \, n_W = 6 \, x \, 6$

$*$

Filter

1	0	-1
1	0	-1
1	0	-1

Parameters:

Size:	$f = 3$
Stride:	$s = 1$
Padding:	$p = 0$

$=$

Result

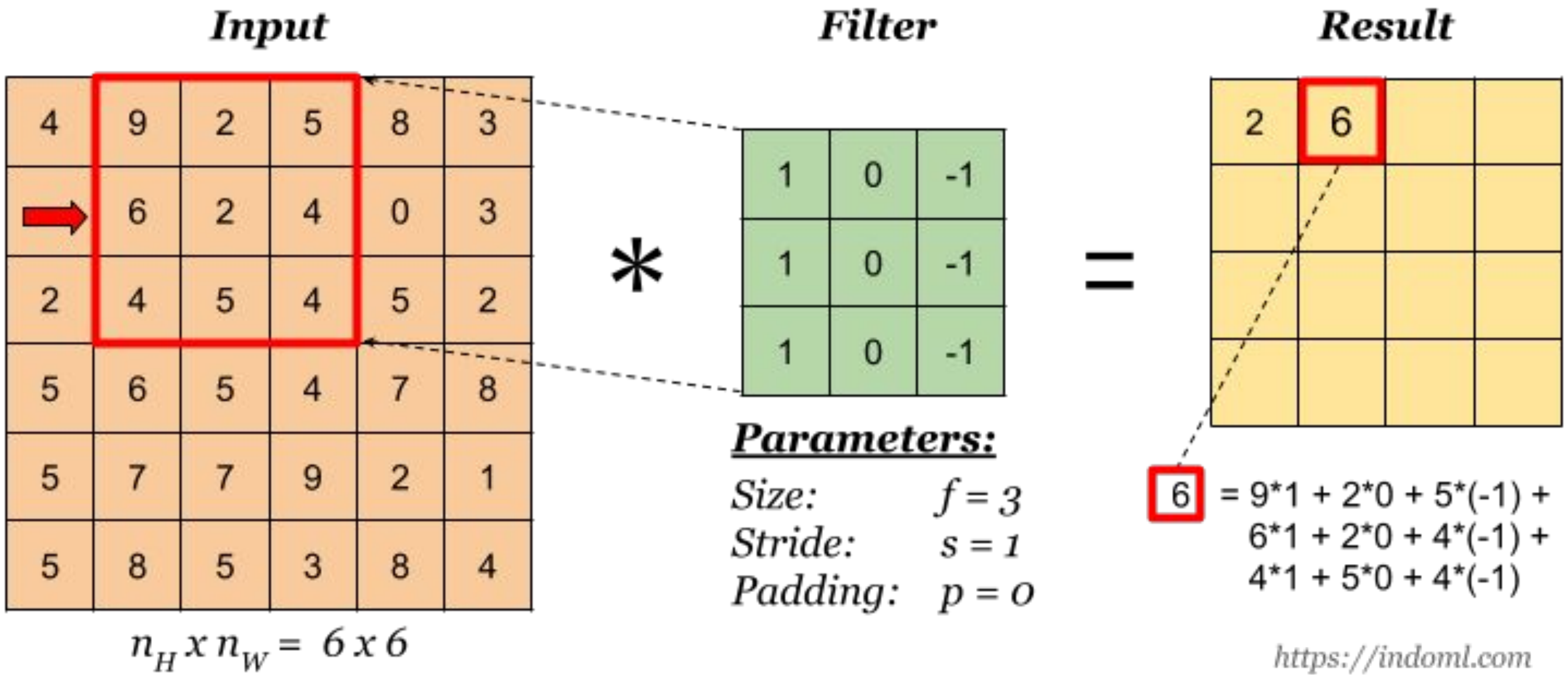
2			

2 = 4*1 + 9*0 + 2*(-1) +
5*1 + 6*0 + 2*(-1) +
2*1 + 4*0 + 5*(-1)

https://indoml.com

CNNs: stride

- **Stride:**
  - how many cells filter is moved next in one step.



Input

4	9	2	5	8	3
6	2	4	0	3	
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

$n_H \, x \, n_W = 6 \, x \, 6$

Filter

1	0	-1
1	0	-1
1	0	-1

Result

2	6		

Parameters:

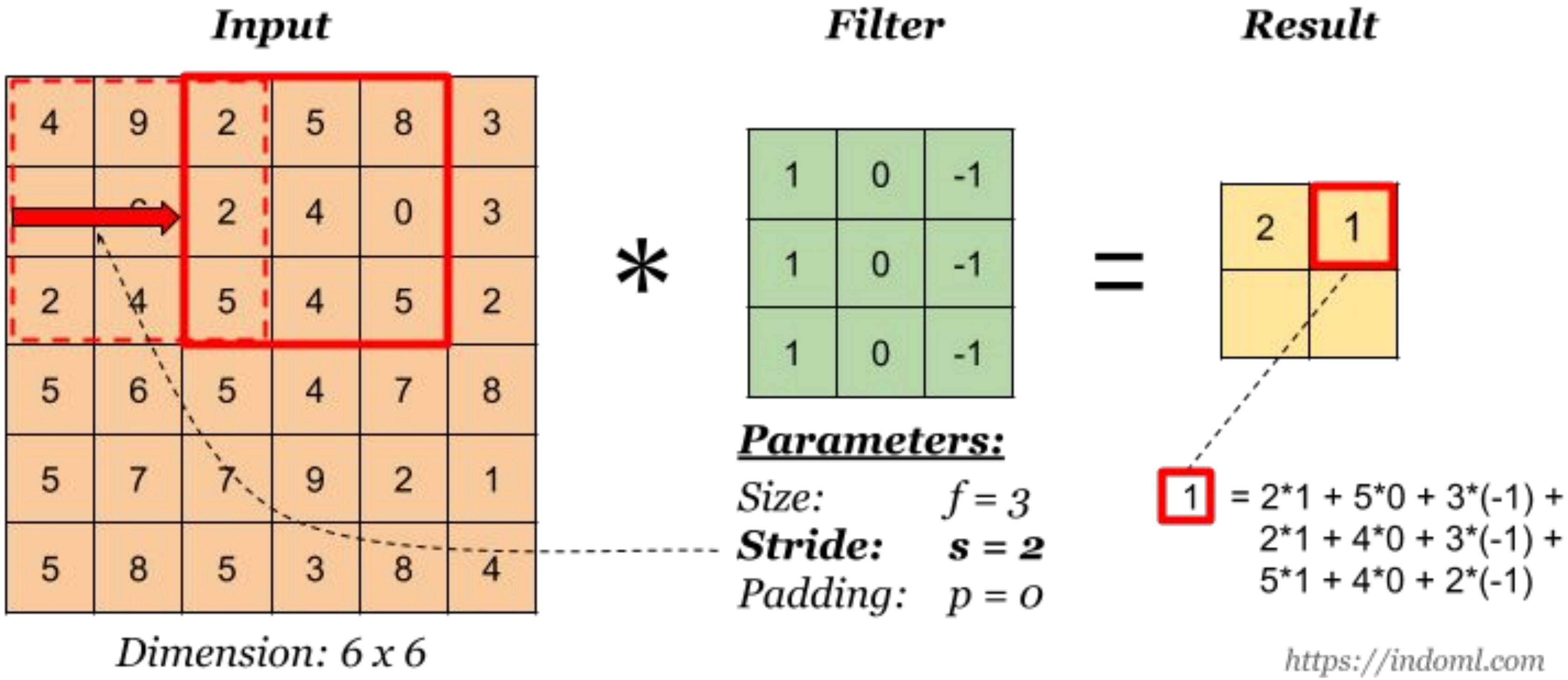
Size: $f = 3$
Stride: $s = 1$
Padding: $p = 0$

$\boxed{6}$ = 9*1 + 2*0 + 5*(-1) +
6*1 + 2*0 + 4*(-1) +
4*1 + 5*0 + 4*(-1)

https://indoml.com

CNNs: stride

- **Stride:**
    - how many cells filter is moved next in one step.



Input

4	9	2	5	8	3
[illegible]	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

Dimension: 6 x 6

Filter

1	0	-1
1	0	-1
1	0	-1

Result

2	1

Parameters:

Size: $f = 3$

Stride: $\boldsymbol{s = 2}$

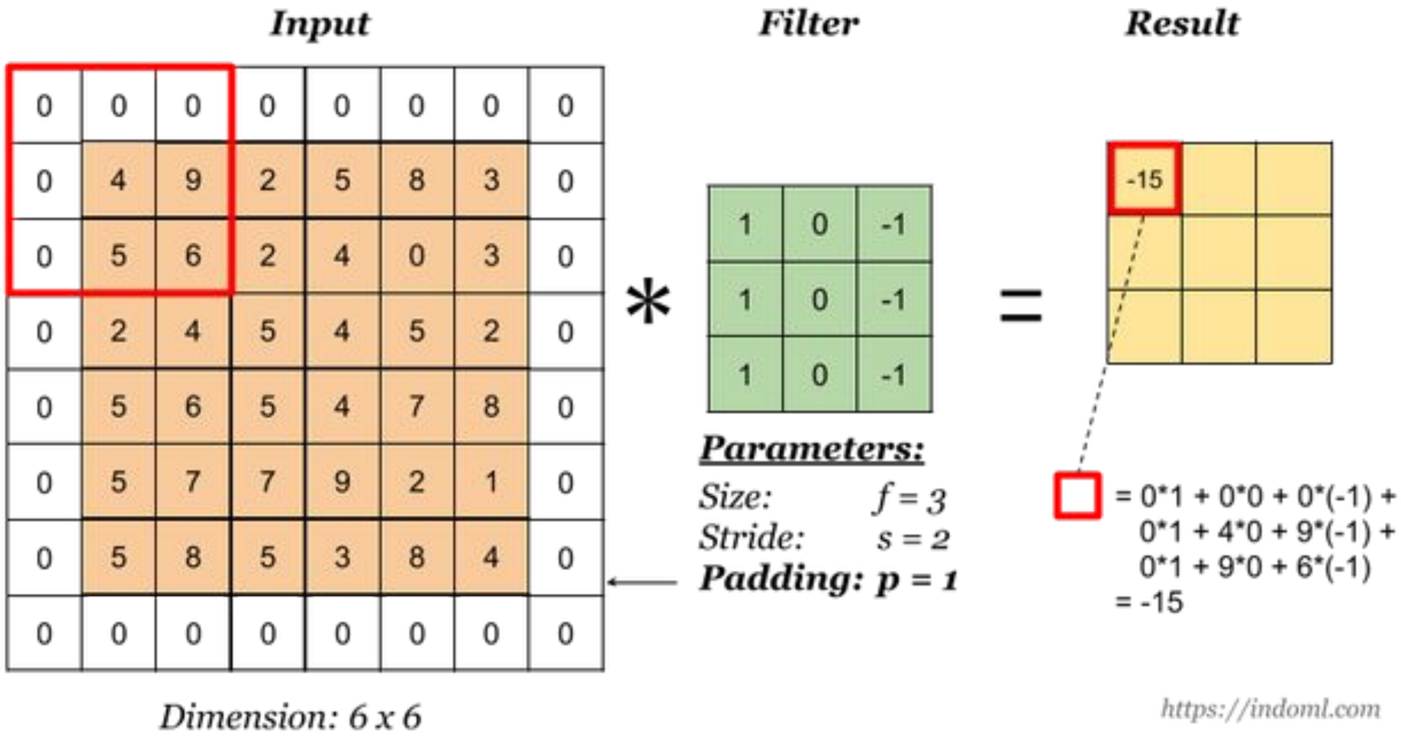
Padding: $p = 0$

$\boxed{1}$ = 2*1 + 5*0 + 3*(-1) +
2*1 + 4*0 + 3*(-1) +
5*1 + 4*0 + 2*(-1)

https://indoml.com

CNNs: padding

- Add the pixels around the border of feature map in order to maintain the spatial footprint
- The value of the padded features is 0



Input

0	0	0	0	0	0	0	0
0	4	9	2	5	8	3	0
0	5	6	2	4	0	3	0
0	2	4	5	4	5	2	0
0	5	6	5	4	7	8	0
0	5	7	7	9	2	1	0
0	5	8	5	3	8	4	0
0	0	0	0	0	0	0	0

Dimension: 6 x 6

* **Filter**

1	0	-1
1	0	-1
1	0	-1

Parameters:

Size: $f = 3$
Stride: $s = 2$
Padding: $p = 1$

= **Result**

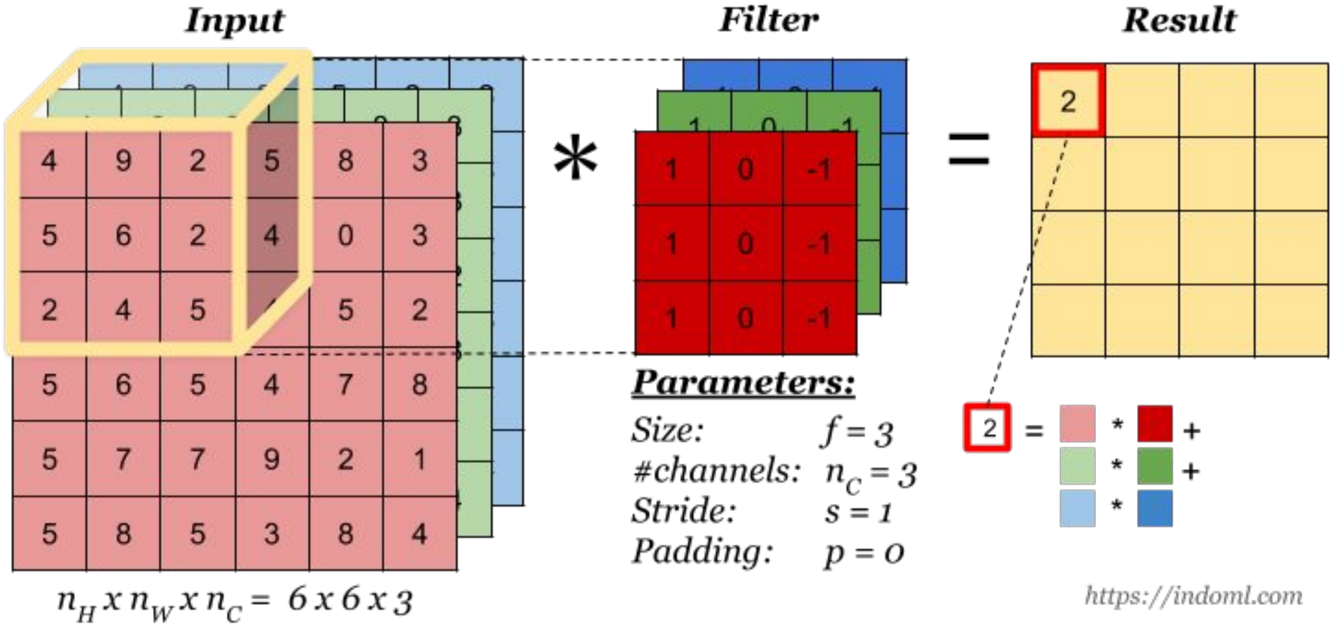
-15		

□ = 0*1 + 0*0 + 0*(-1) +
0*1 + 4*0 + 9*(-1) +
0*1 + 9*0 + 6*(-1)
= -15

https://indoml.com

CNNs: convolution on 3 channels

- Channels
  - RGB input: 3 **channels** red, green and blue
  - A convolution layer:
    - receives the image (h×w×c) as input
    - generates as output an activation map of dimensions h’×w’×c’.
  - The number of channels is the **depth** of the matrices involved in the convolutions



4	9	2	5	8	3
5	6	2	4	0	3
2	4	5	[illegible]	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

1	0	-1
1	0	-1
1	0	-1

2			

Parameters:

Size:	$f = 3$
#channels:	$n_C = 3$
Stride:	$s = 1$
Padding:	$p = 0$

https://indoml.com

Exercise 2

- Goal: create an Adversarial Autoencoder
  - Using MINST Clothes dataset

Exercise 3

- Goal: create an unsupervised GAN
  - Using MINST Clothes dataset