Computer Vision

Digital Image Basics

Dr. Aurélien Quillet

Data Scientist & AI dev

Professional experience

- PhD: Molecular Biologist (4 years)
- Data Science Project Leader (3 years)
- Teacher / Mentor (3 years)

Teaching experience

Topics:

- Data / IA
- Code / programming
- Algorithmes
- Mathématiques
- Molecular Biology

My training

Computer Vision

Course description

The Syllabus

Session 1 : intro

Session 2 : CNN architecture

Session 3 : Object detection

Session 4 : Image segmentation

Session 5 : Image generation

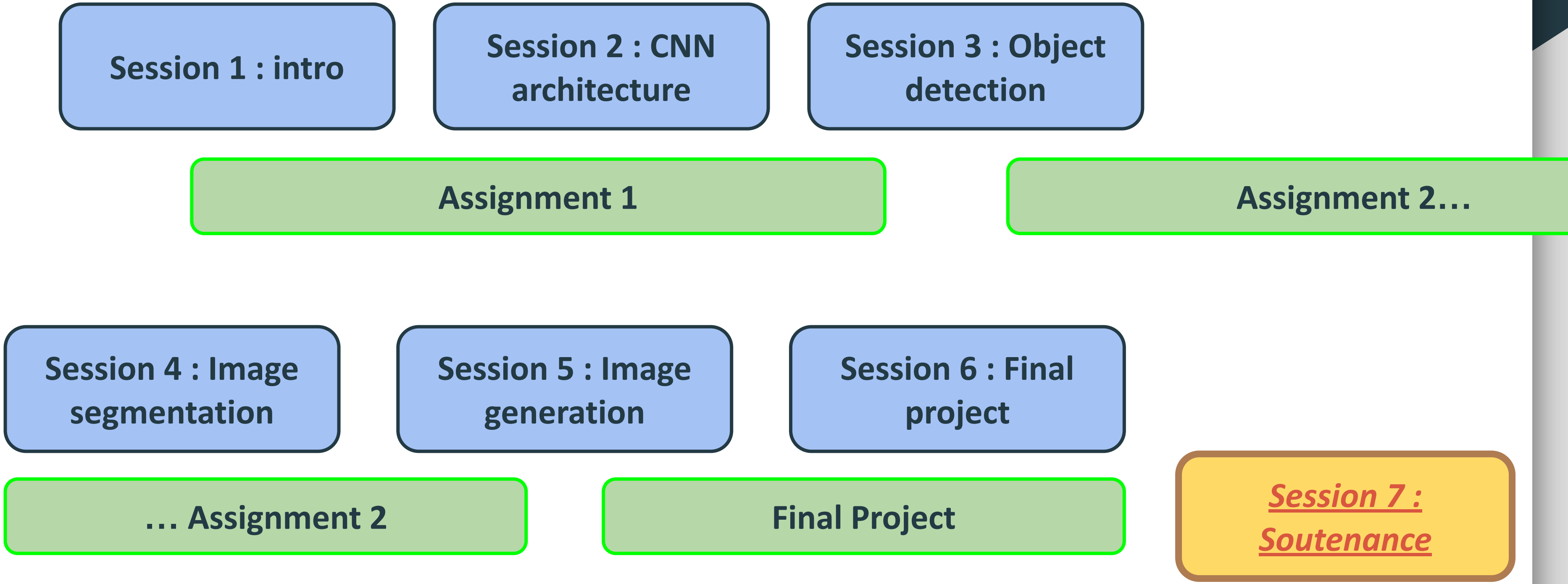
Session 6 : Final project

Session 7 : Soutenance

Computer Vision

Course description

The Syllabus



Session 1 : intro

Session 2 : CNN architecture

Session 3 : Object detection

Assignment 1

Assignment 2…

Session 4 : Image segmentation

Session 5 : Image generation

Session 6 : Final project

… Assignment 2

Final Project

Session 7 : Soutenance

Computer Vision

Course description

The Syllabus

Assignment	Given out	Turned in	Points	Coef
1	27/05/2024	17/06/2024	100	1
2	17/06/2024	4/07/2024	100	2
3 (final)	04/07/2024	27/07/2024	100	3

Computer Vision

Course description

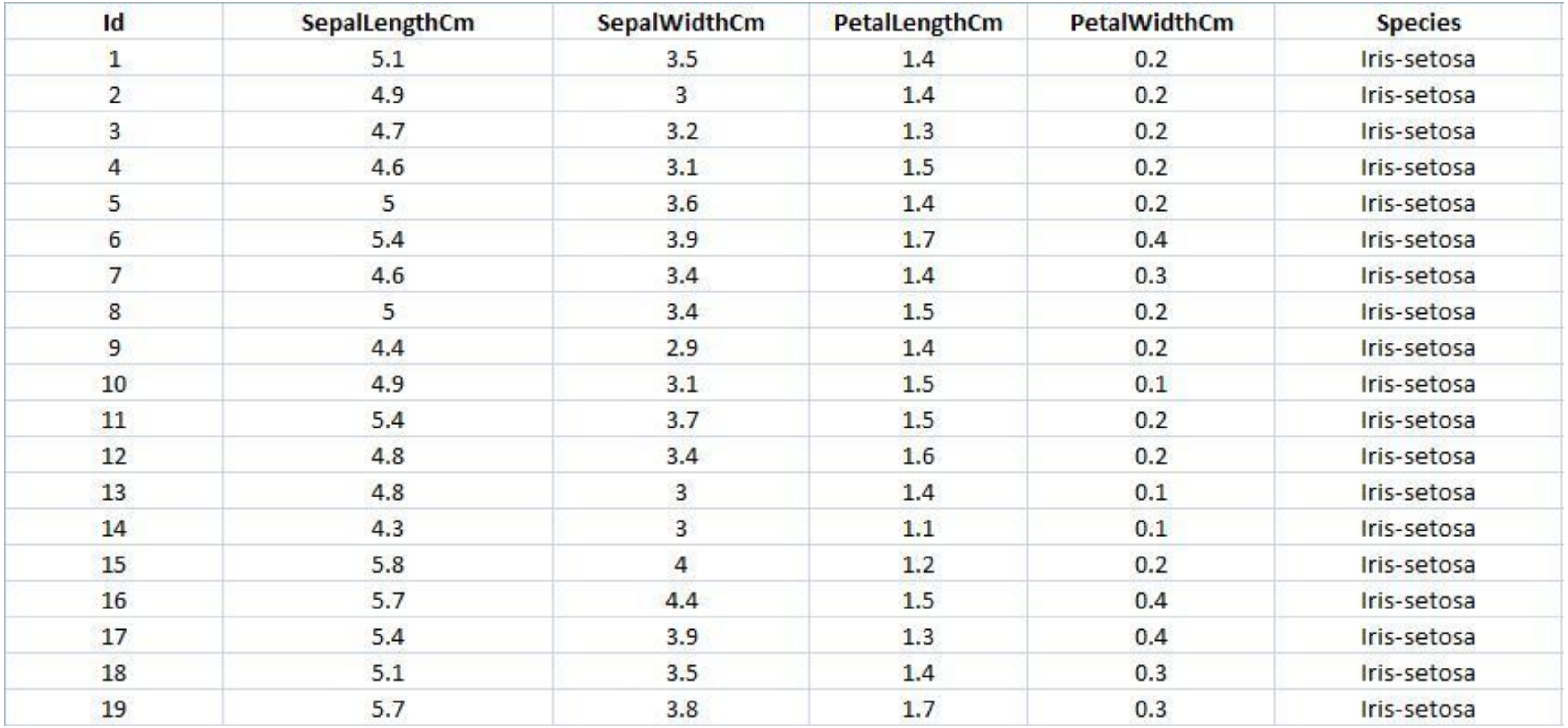
Grading Grid

Skill	Description	Points
Documentation (markdown)	● Using at least 3 different resources (kaggle notebooks, blogs, youtube videos or else…), explain your strategy and why you think this is going to work. ● Your code is commented when needed. ● The model and hyperparameters selection is explained. ● The performances are commented on. ● Bibliographical references are present.	50
Code (python)	● All blocks necessary to implement your strategy are present. ● Specialized libraries have been used. ● All notebook cells have been executed successfully sequentially.	30
Performances	● A baseline is defined. ● All necessary comparisons are done. ● Figures are readable and legends are present. ● A proper evaluation metric was selected.	20
Application (bonus)	● The script is functional. ● The script respects the required parameters.	10

Plan

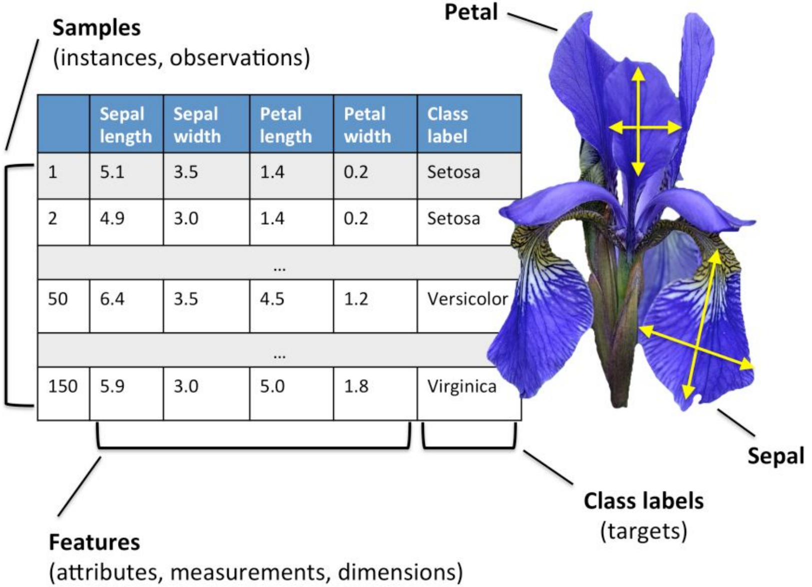
- Image analysis
  - Pixel notions
  - Colors
  - Image histogram
- Image filtering
  - Convolution filtering

Usual dataset



Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa

Usual features / classification

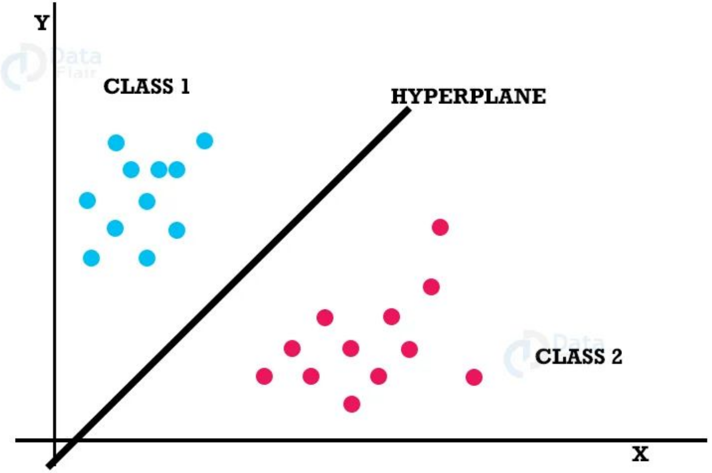


Samples
(instances, observations)

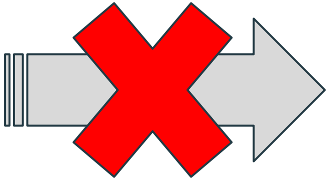
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)



The problem



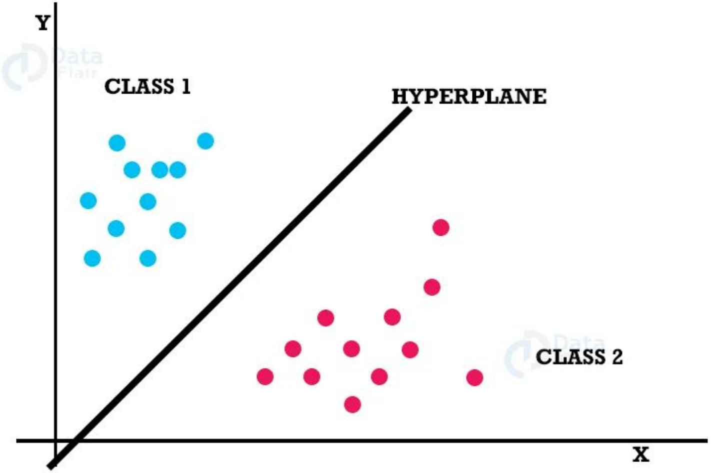


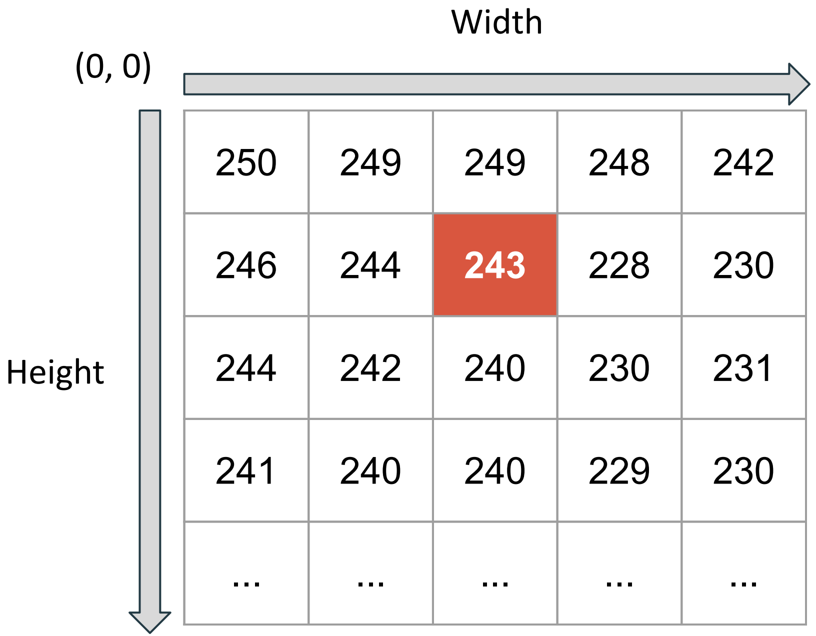
Image Analysis

Pixel notions & histograms

Pixel notions

- Image sampling : **divide** image in **small areas** (pixels) containing a **value** (or a list of values)
- Coordinates : **Width** & **Height** position of a pixel
- Quantification : number of **possible values**
- Definition : W X H
- Resolution : Pixels / Length unit

Grayscale



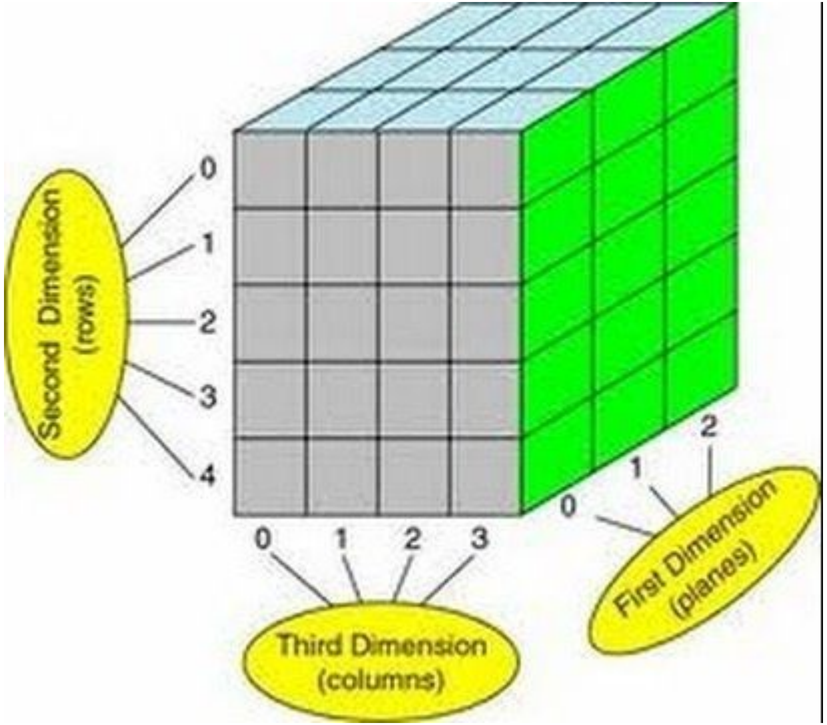
250	249	249	248	242
246	244	243	228	230
244	242	240	230	231
241	240	240	229	230
...	...	...	...	...

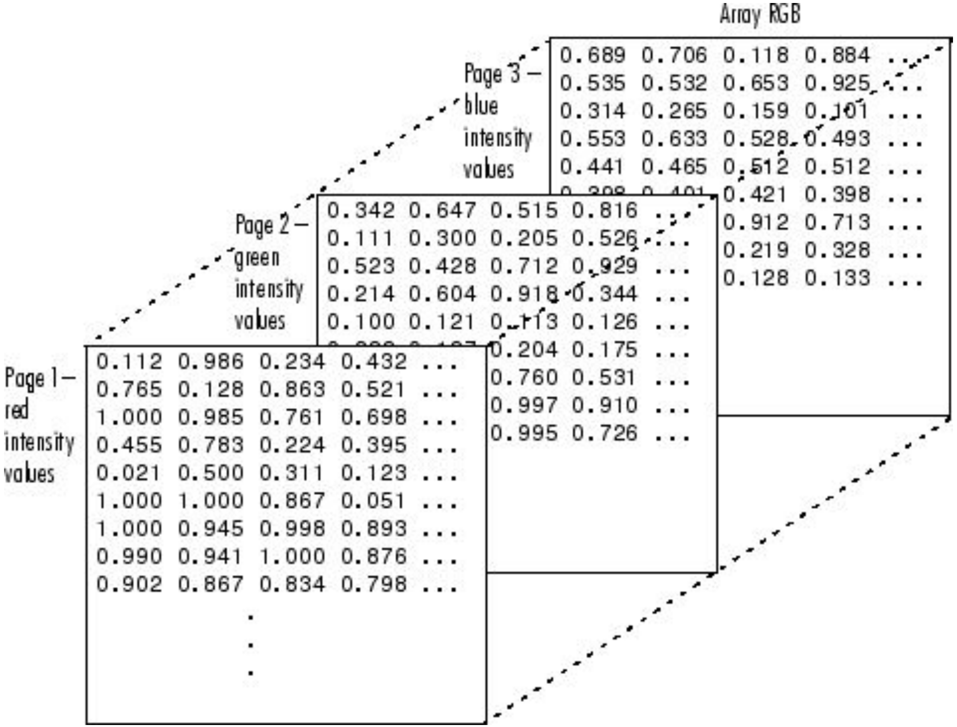
An image / 2D array

	Col1	Col2	Col3	Col4	····
Row1	Arr[0][0]	Arr[0][1]	Arr[0][2]	Arr[0][3]	
Row2	Arr[1][0]	Arr[1][1]	Arr[1][2]	Arr[1][3]	
Row3	Arr[2][0]	Arr[2][1]	Arr[2][2]	Arr[2][3]	
Row4	Arr[3][0]	Arr[3][1]	Arr[3][2]	Arr[3][3]	
⋮					

250	249	249	248	242
246	244	**243**	228	230
244	242	240	230	231
241	240	240	229	230
...	...	...	...	...

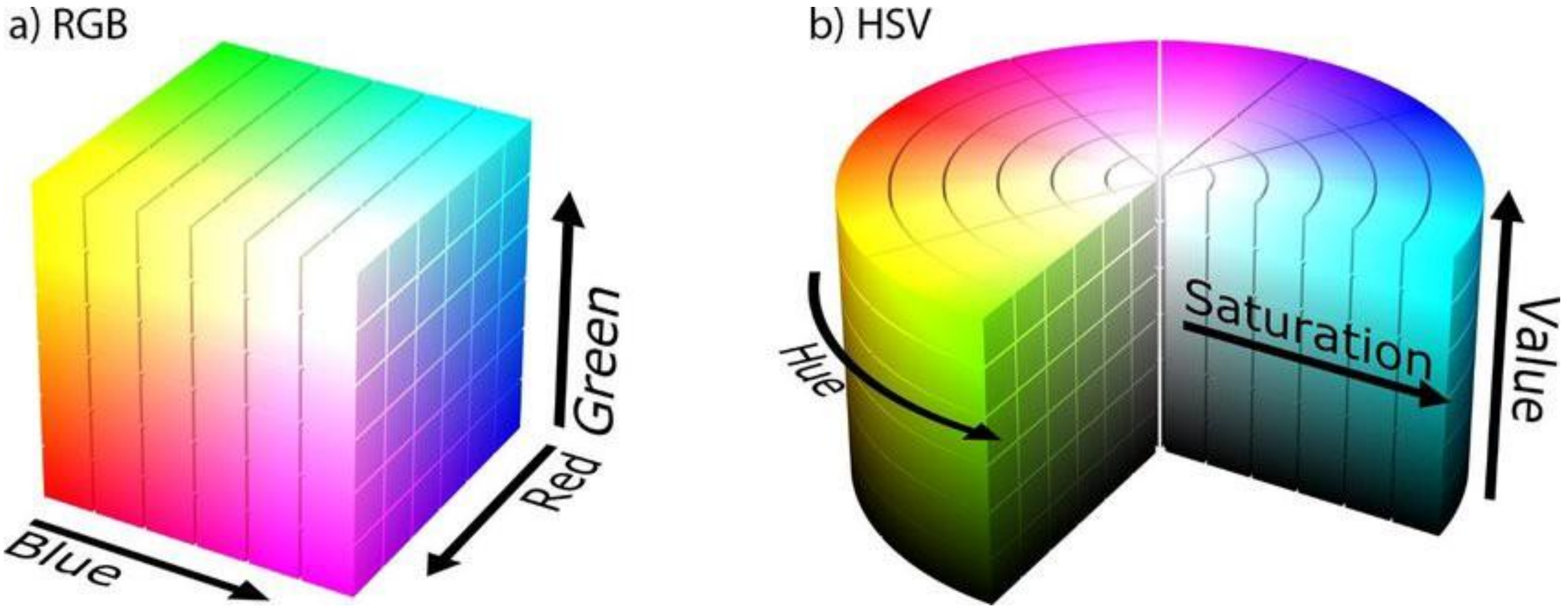
An image / 3D array




Array RGB
Page 3 – blue intensity values
Page 2 – green intensity values
Page 1 – red intensity values

```
Page 3
0.689 0.706 0.118 0.884 ...
0.535 0.532 0.653 0.925 ...
0.314 0.265 0.159 0.101 ...
0.553 0.633 0.528 0.493 ...
0.441 0.465 0.512 0.512 ...
            0.421 0.398 ...
            0.912 0.713 ...
            0.219 0.328 ...
            0.128 0.133 ...

Page 2
0.342 0.647 0.515 0.816 ...
0.111 0.300 0.205 0.526 ...
0.523 0.428 0.712 0.929 ...
0.214 0.604 0.918 0.344 ...
0.100 0.121 0.113 0.126 ...
            0.204 0.175 ...
            0.760 0.531 ...
            0.997 0.910 ...
            0.995 0.726 ...

Page 1
0.112 0.986 0.234 0.432 ...
0.765 0.128 0.863 0.521 ...
1.000 0.985 0.761 0.698 ...
0.455 0.783 0.224 0.395 ...
0.021 0.500 0.311 0.123 ...
1.000 1.000 0.867 0.051 ...
1.000 0.945 0.998 0.893 ...
0.990 0.941 1.000 0.876 ...
0.902 0.867 0.834 0.798 ...
            .
            .
            .
```

Colors

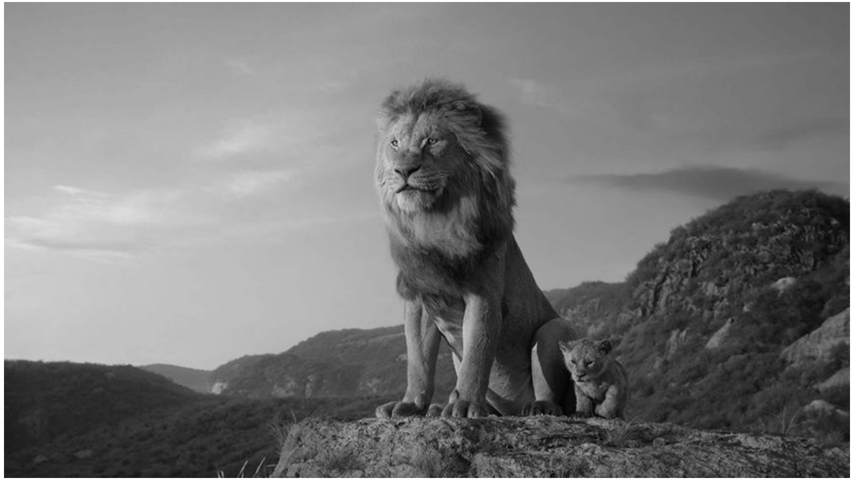


a) RGB

b) HSV

Image Histogram

Original



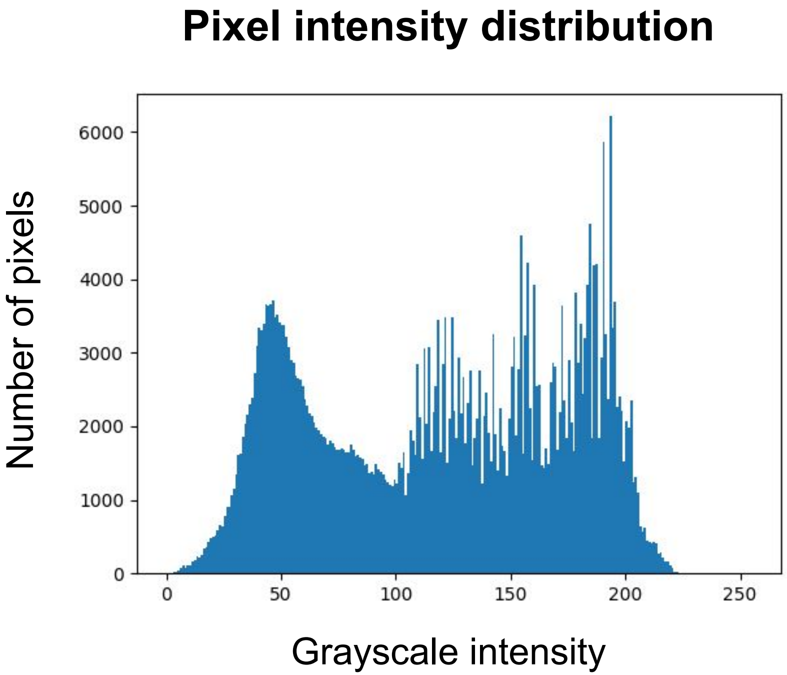
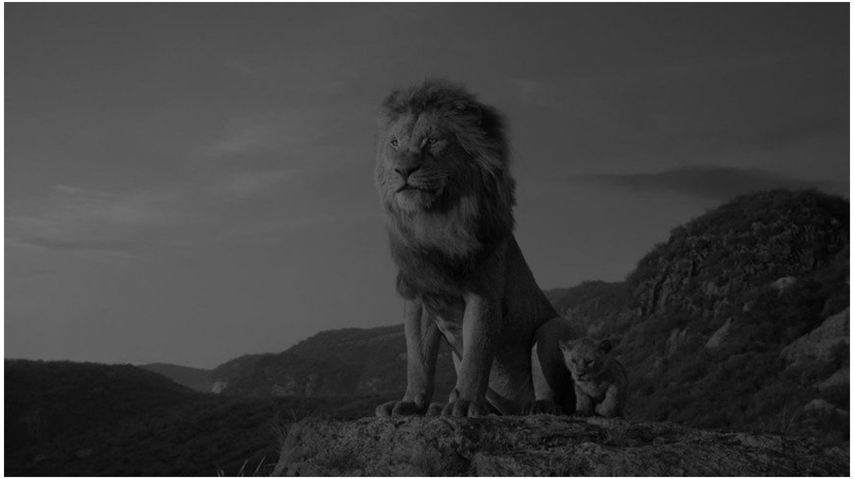


Image Histogram

Under exposed



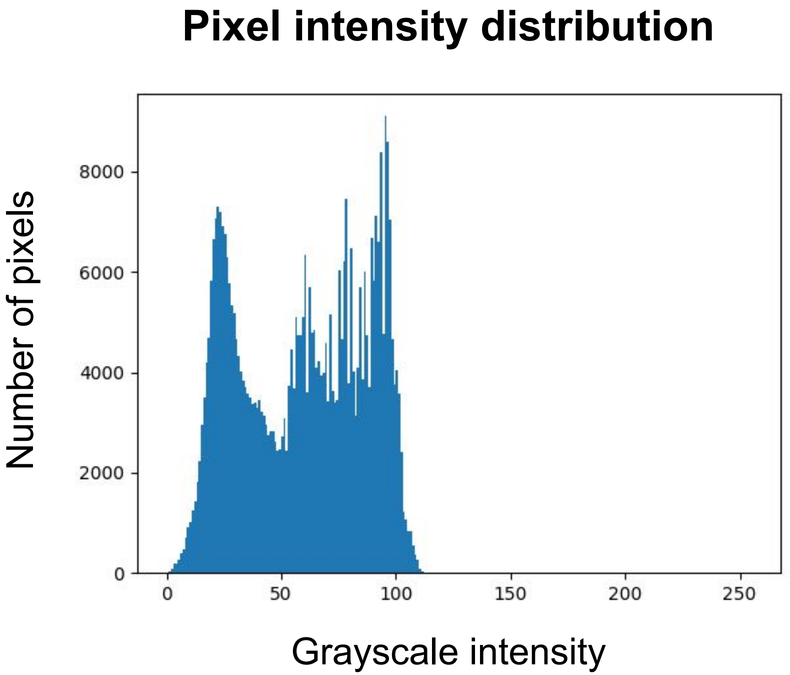
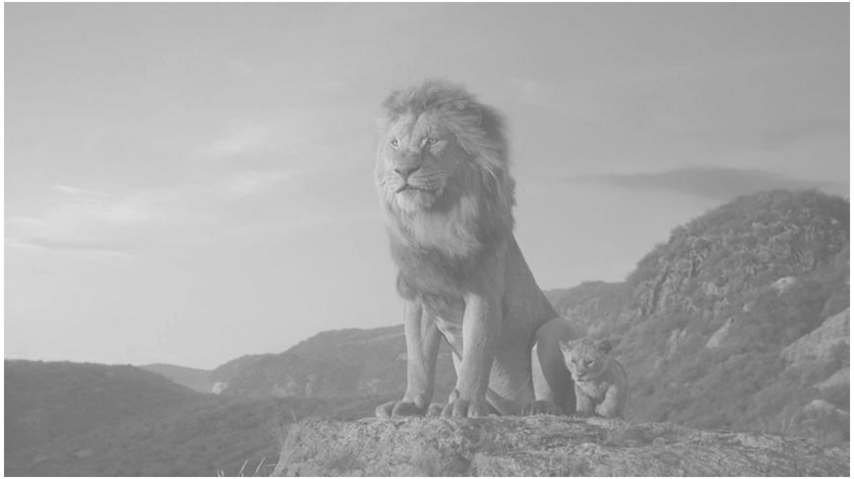
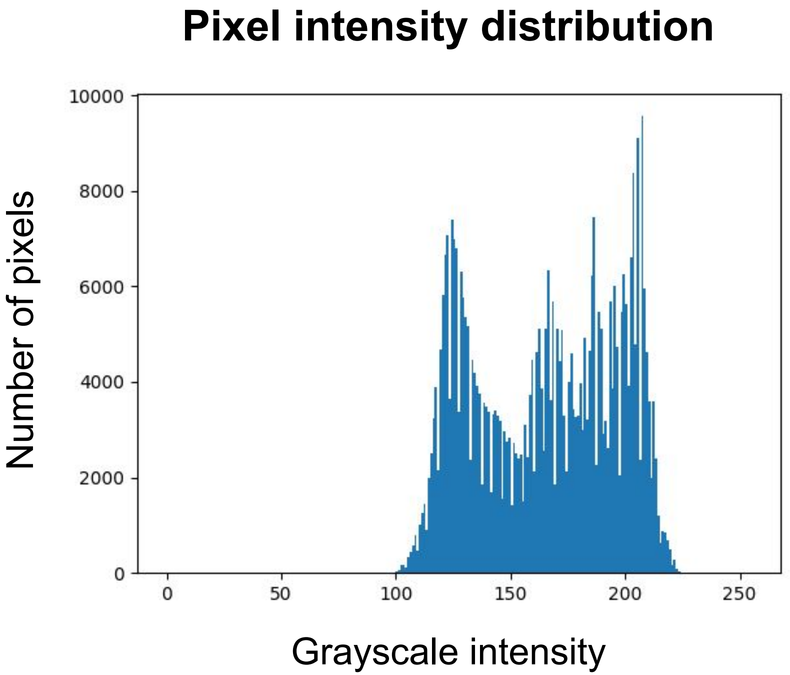


Image Histogram

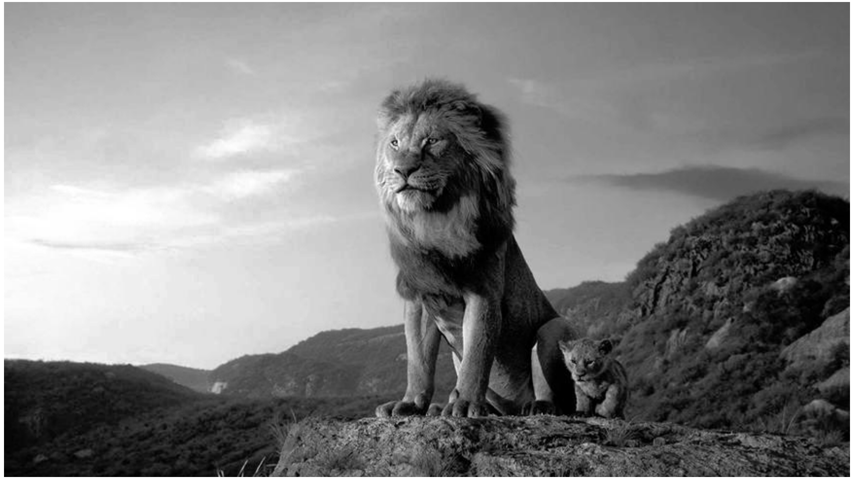
Over exposed



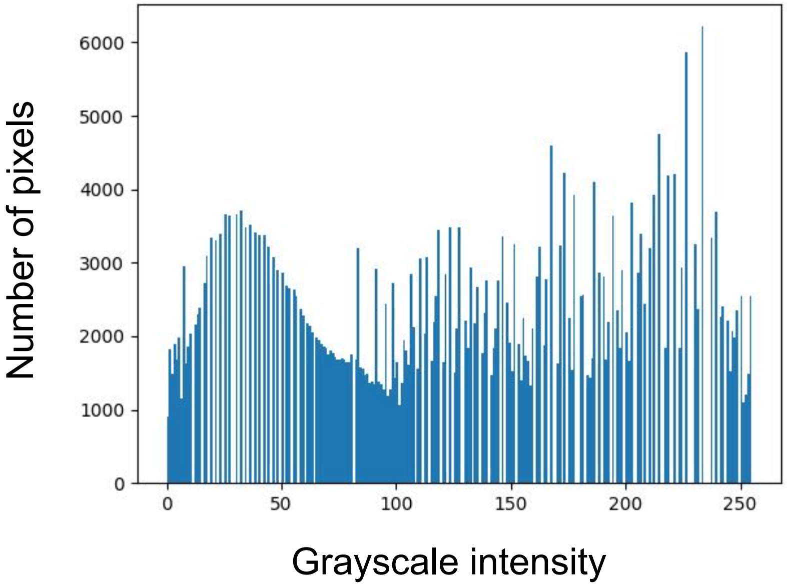


Histogram Equalization

Contrast adjusted

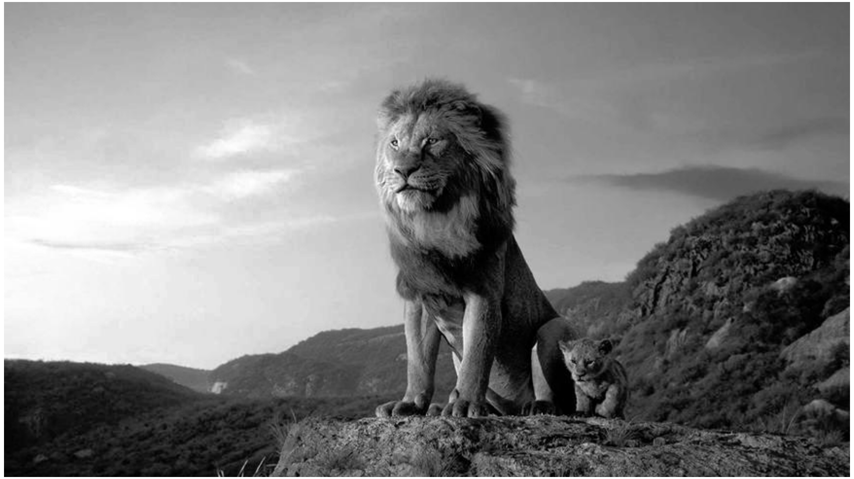


Pixel intensity distribution

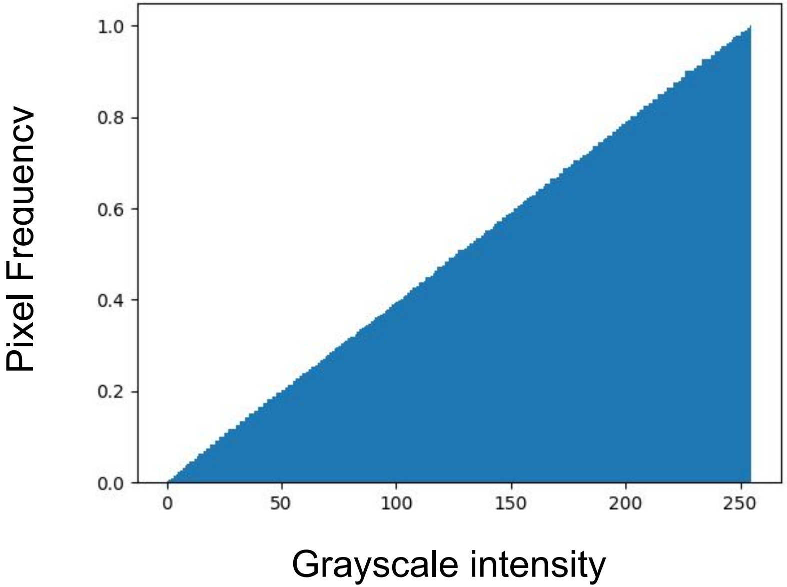


Histogram Equalization

Contrast adjusted



Cumulative pixel intensity frequency



Librairies

scikit-image
image processing in python

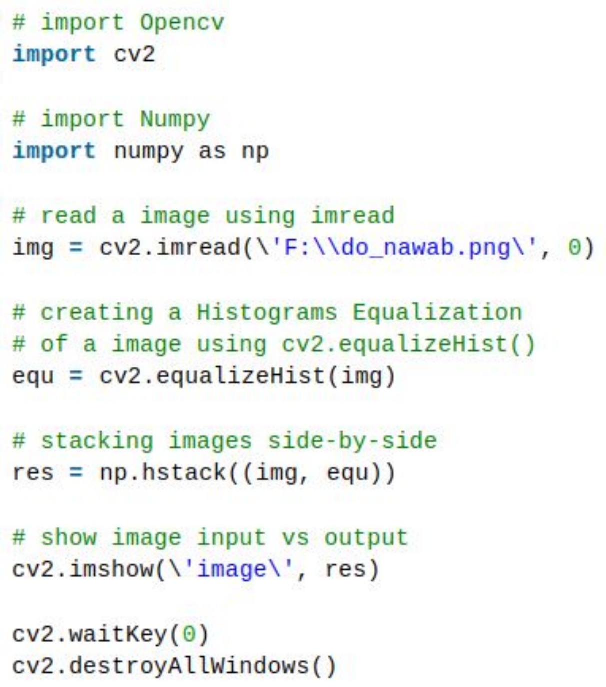
OpenCV

matplotlib

pillow

Practice

Before you ask…



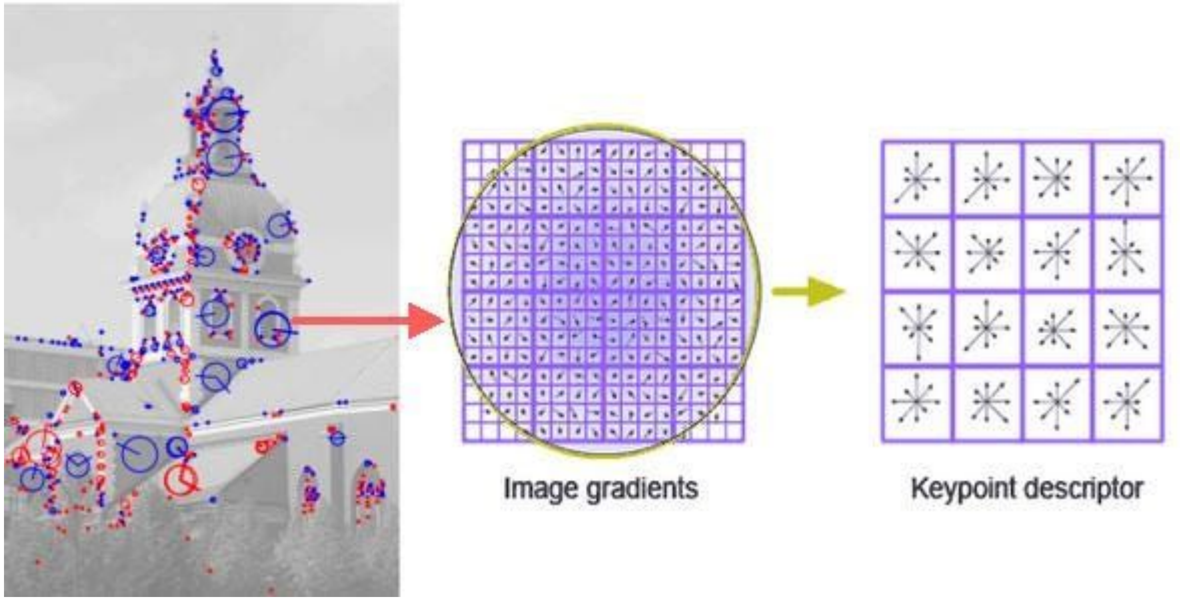
```
# import Opencv
import cv2

# import Numpy
import numpy as np

# read a image using imread
img = cv2.imread(\'F:\\do_nawab.png\', 0)

# creating a Histograms Equalization
# of a image using cv2.equalizeHist()
equ = cv2.equalizeHist(img)

# stacking images side-by-side
res = np.hstack((img, equ))

# show image input vs output
cv2.imshow(\'image\', res)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Source: geeksforgeeks.org

Image Filtering

Feature extraction & Convolution

Feature extraction



SIFT

SURF

FAST

BRIEF

ORB

Feature extraction

ID	Desc1.1	Desc1.2	Desc1.3	Desc1.4	Desc2.1	Desc2.2
1	0.256	20.64	5.588	8.461	3.5123	0.4561
2	620.64	53.5123	83.5123	78.461	10.4561	50.4561
3	58.461	60.256	820.64	28.461	220.64	320.64
4	50.4561	40.256	28.461	0.2566	80.256	50.4561
5	60.256	40.256	80.4561	20.4561	38.461	20.4561
6	63.5123	30.4561	520.64	83.5123	90.256	70.4561

SIFT

SURF

FAST

BRIEF

ORB

Mais... ça ne marche pas top !

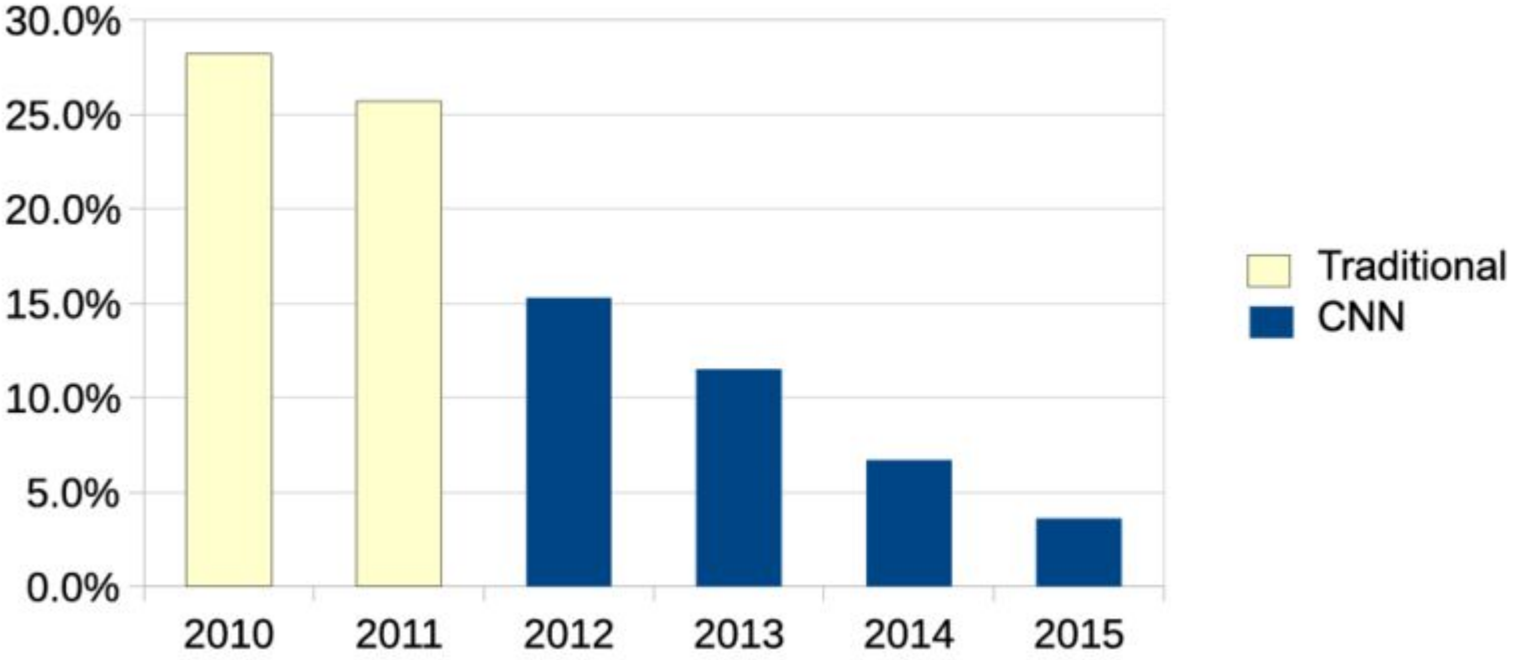
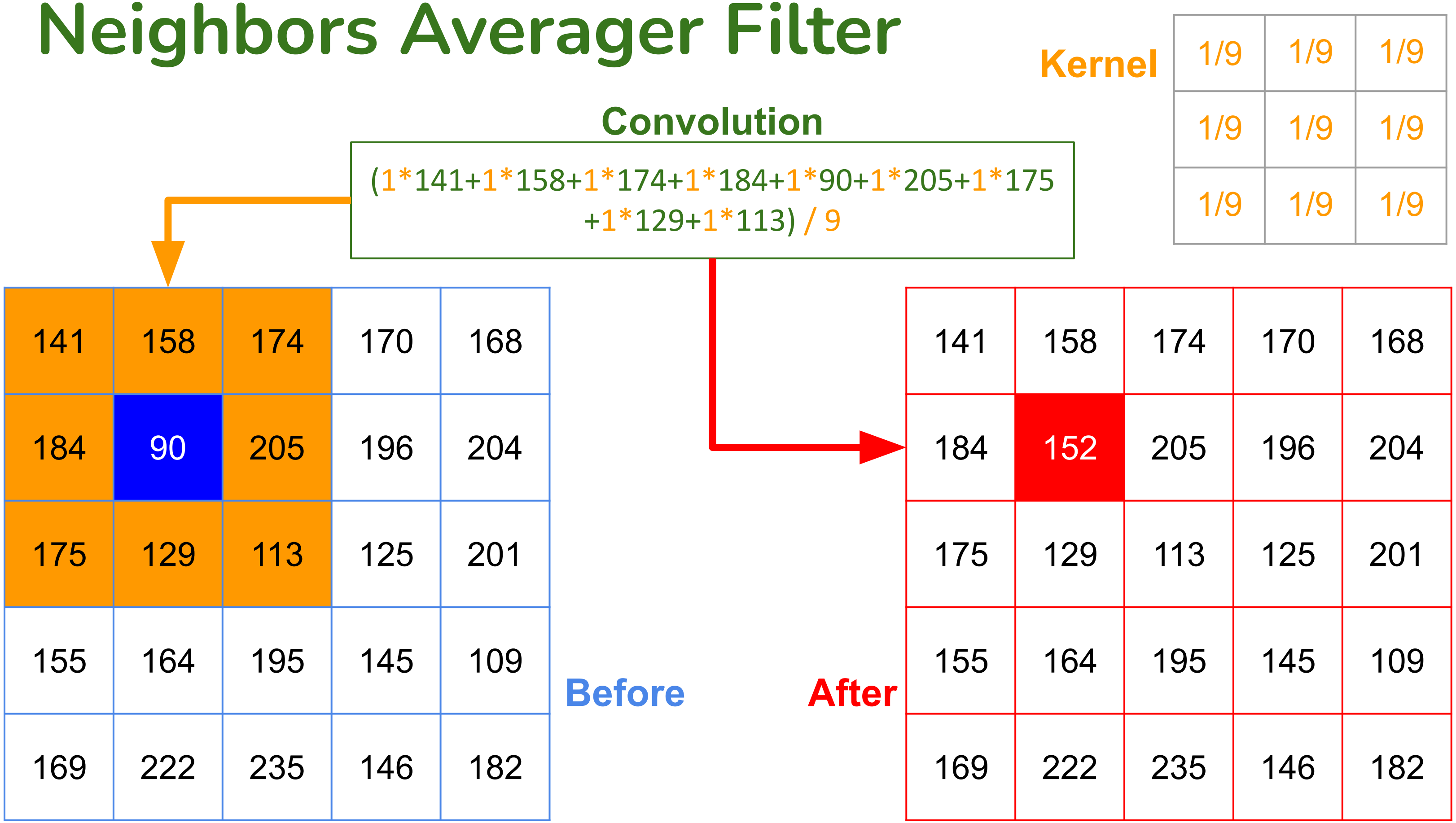


Image Filtering

Feature extraction & Convolution

Neighbors Averager Filter



Kernel

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Convolution

(1*141+1*158+1*174+1*184+1*90+1*205+1*175
+1*129+1*113) / 9

Before

141	158	174	170	168
184	90	205	196	204
175	129	113	125	201
155	164	195	145	109
169	222	235	146	182

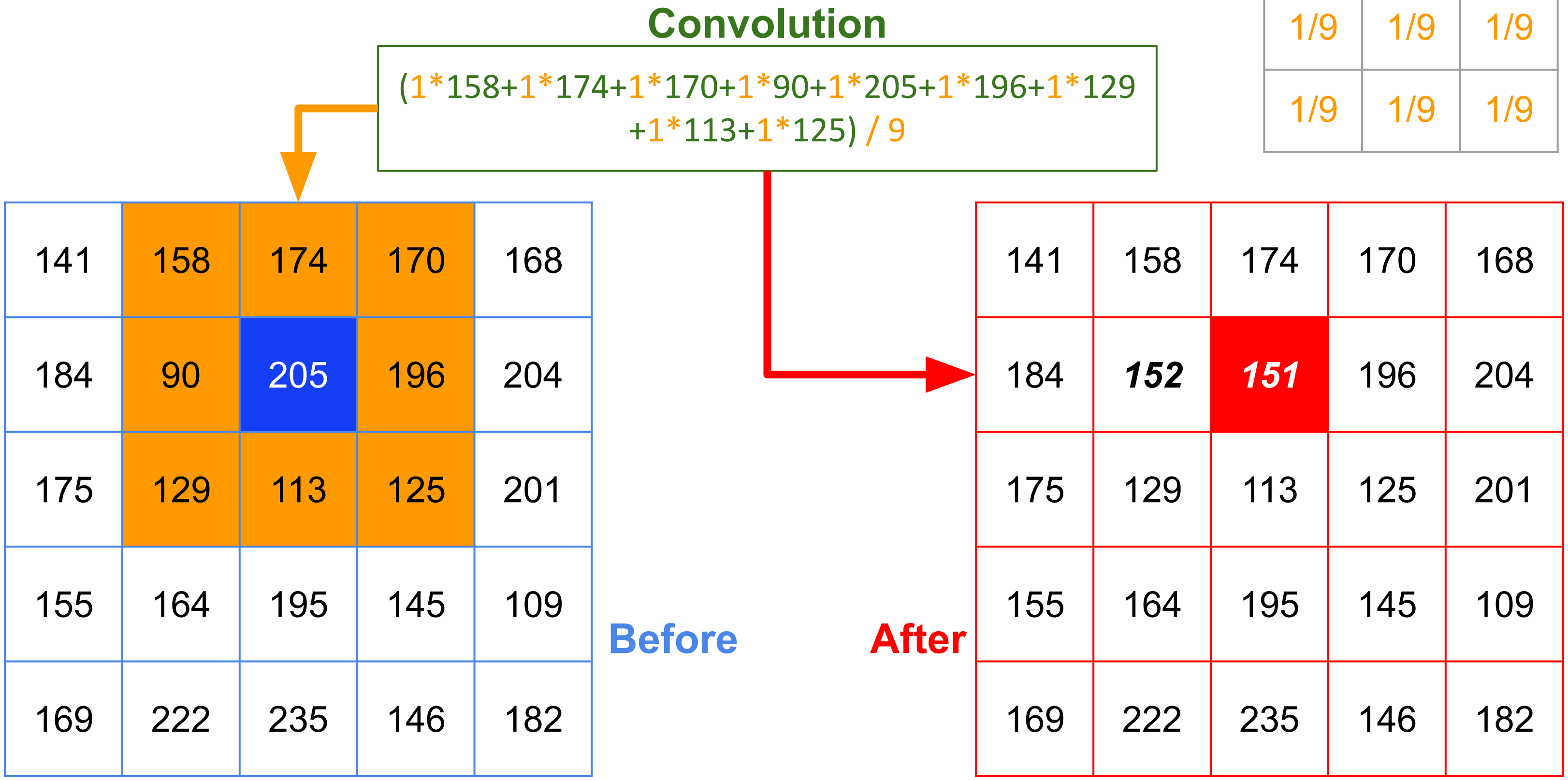
After

141	158	174	170	168
184	152	205	196	204
175	129	113	125	201
155	164	195	145	109
169	222	235	146	182

Neighbors Averager Filter

Kernel

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



Convolution

(1*158+1*174+1*170+1*90+1*205+1*196+1*129
+1*113+1*125) / 9

Before

141	158	174	170	168
184	90	205	196	204
175	129	113	125	201
155	164	195	145	109
169	222	235	146	182

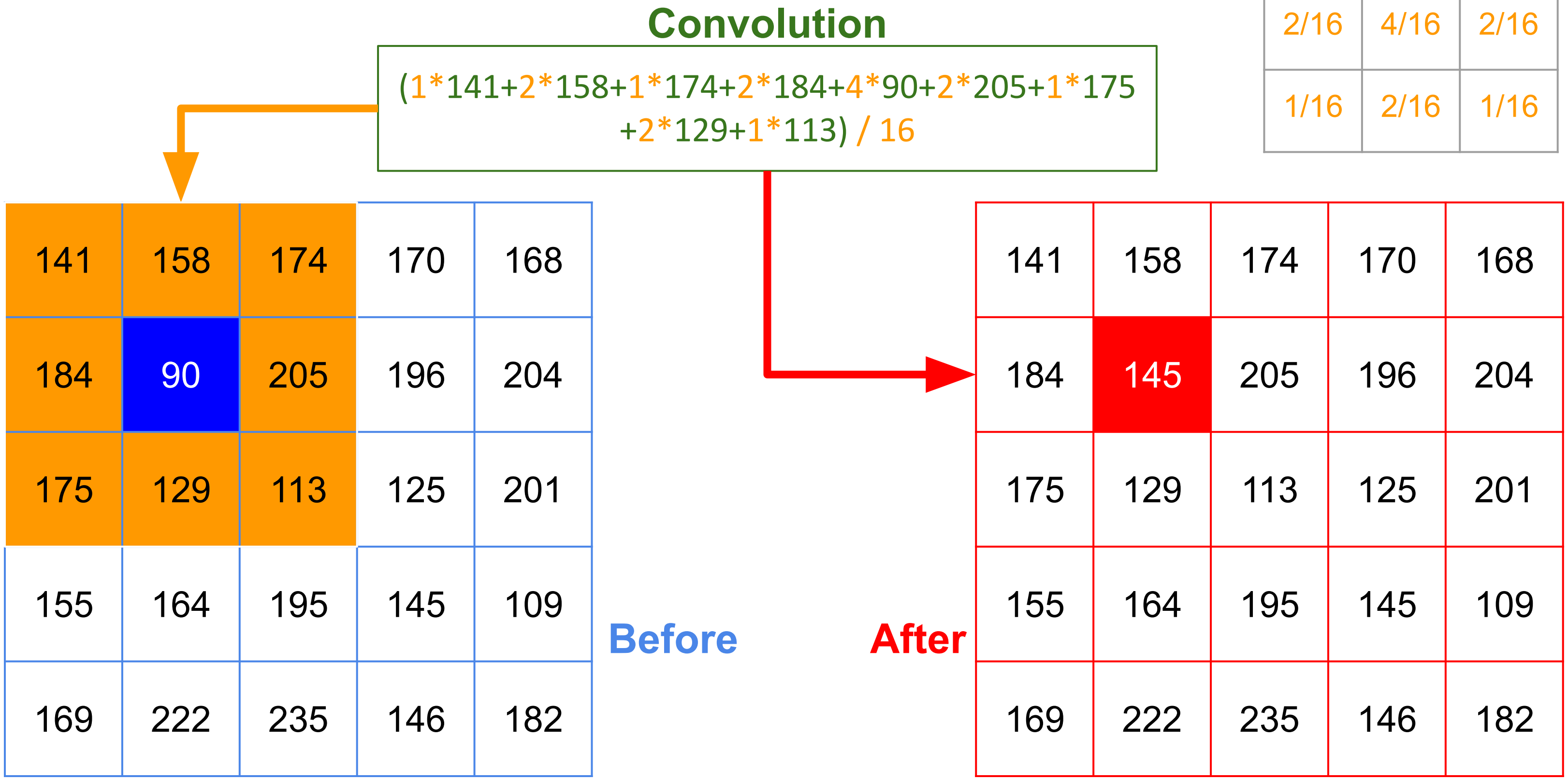
After

141	158	174	170	168
184	***152***	***151***	196	204
175	129	113	125	201
155	164	195	145	109
169	222	235	146	182

Gaussian Filter

Kernel

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16



Convolution

(1*141+2*158+1*174+2*184+4*90+2*205+1*175
+2*129+1*113) / 16

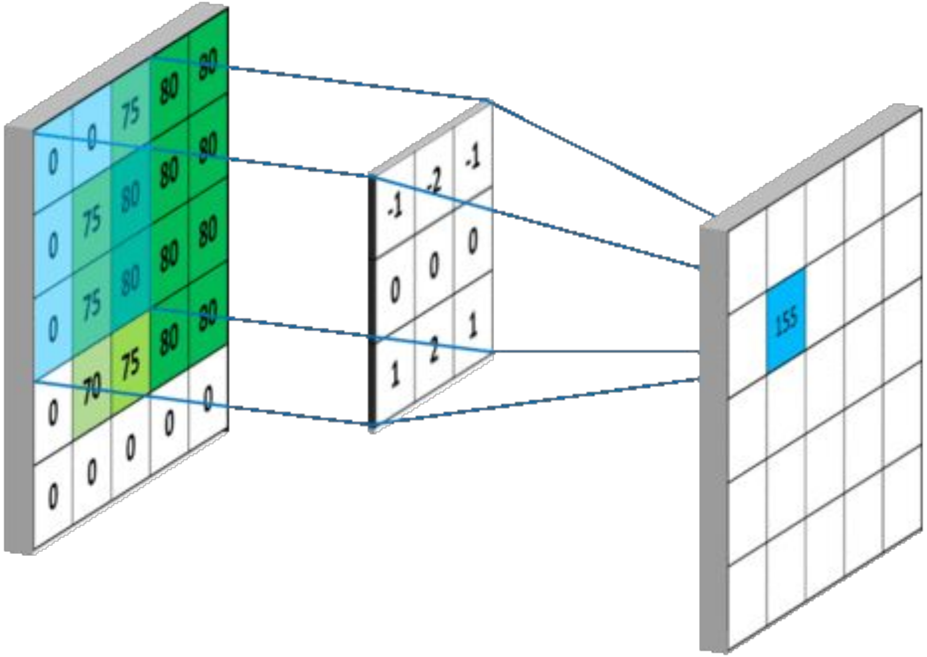
Before

141	158	174	170	168
184	90	205	196	204
175	129	113	125	201
155	164	195	145	109
169	222	235	146	182

After

141	158	174	170	168
184	145	205	196	204
175	129	113	125	201
155	164	195	145	109
169	222	235	146	182

Animated Convolution

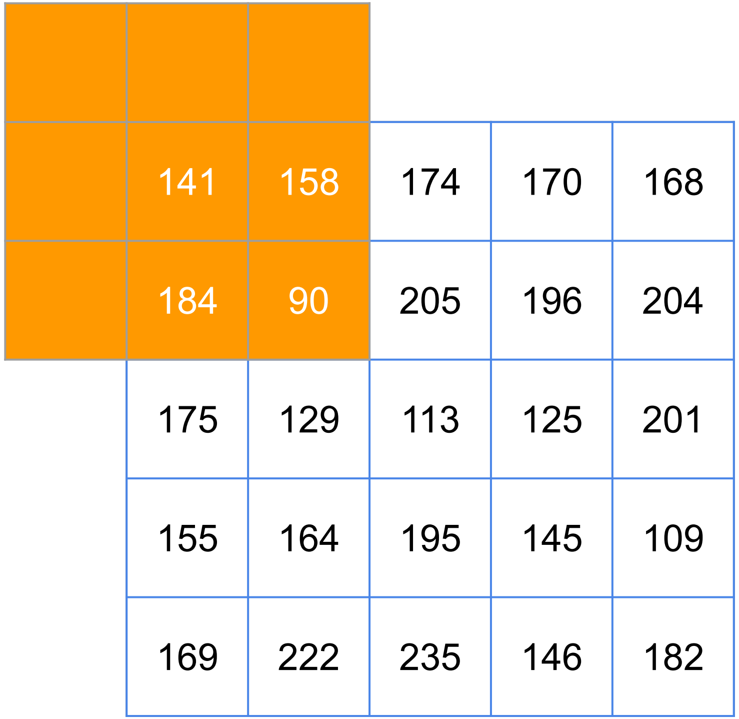


0	0	75	80	80
0	75	80	80	80
0	75	80	80	80
0	70	75	80	80
0	0	0	0	0

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

Borders management

- Zero padding
- Duplication
- Partial convolution

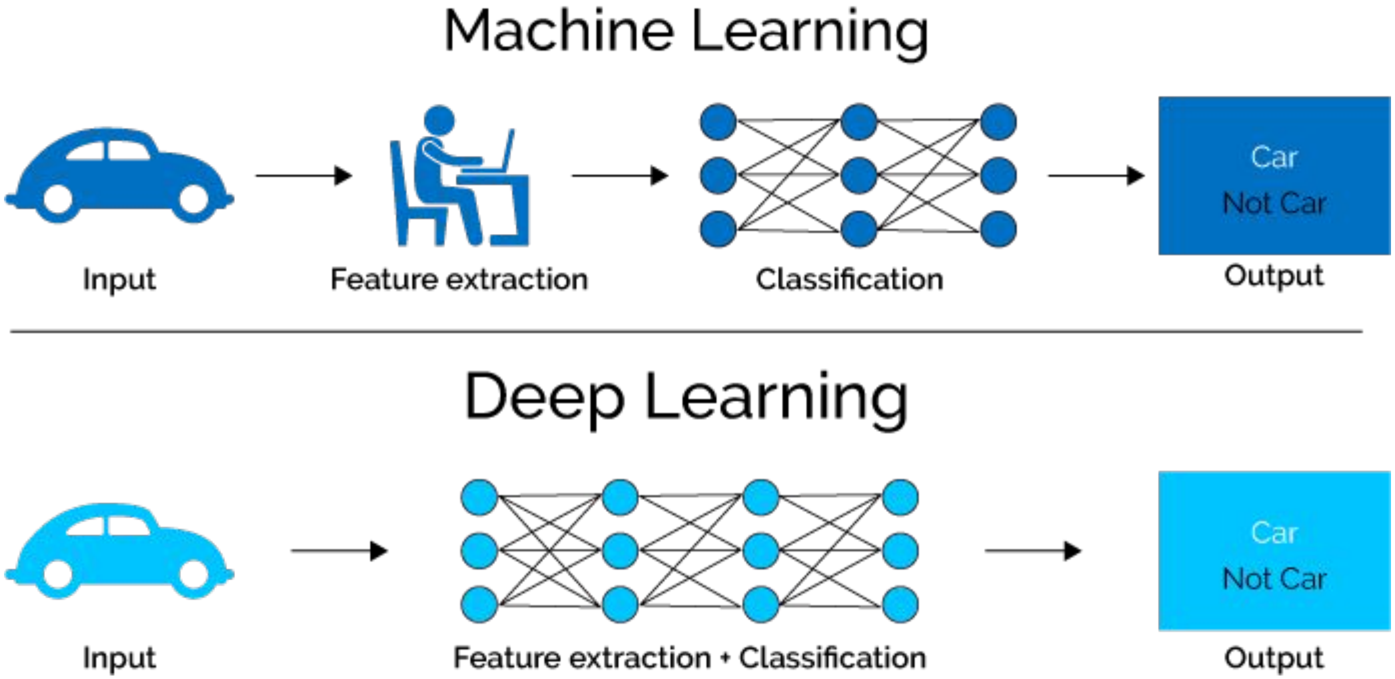


	141	158	174	170	168
	184	90	205	196	204
	175	129	113	125	201
	155	164	195	145	109
	169	222	235	146	182

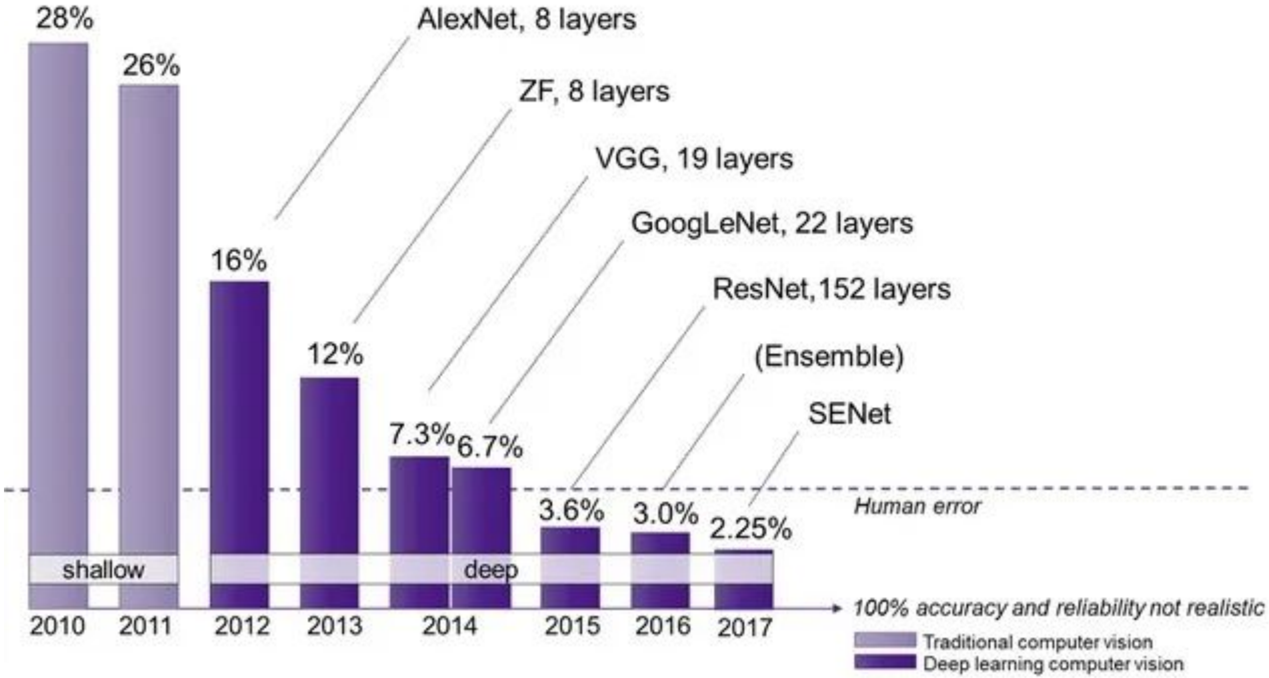
What’s next ?

Suspens… !!!

Teaser...


Machine Learning
Car
Not Car
Input
Feature extraction
Classification
Output
Deep Learning
Car
Not Car
Input
Feature extraction + Classification
Output

Teaser...


28%
26%
16%
12%
7.3%
6.7%
3.6%
3.0%
2.25%
AlexNet, 8 layers
ZF, 8 layers
VGG, 19 layers
GoogLeNet, 22 layers
ResNet,152 layers
(Ensemble)
SENet
Human error
shallow
deep
2010
2011
2012
2013
2014
2015
2016
2017
100% accuracy and reliability not realistic
Traditional computer vision
Deep learning computer vision

Exercices

Coding games & Image classification

Bonus points (10) !
Let’s play some Codingame !

STEP UP YOUR CODING GAME
The new way to improve your programming skills while having fun and getting noticed

- easy:
  - flip the sign
  - Reverse minesweeper
  - sudoku validator
  - lumen
  - pirate’s treasure
- medium:
  - forest fire
  - battleship