Représentation des données Images et Vidéos

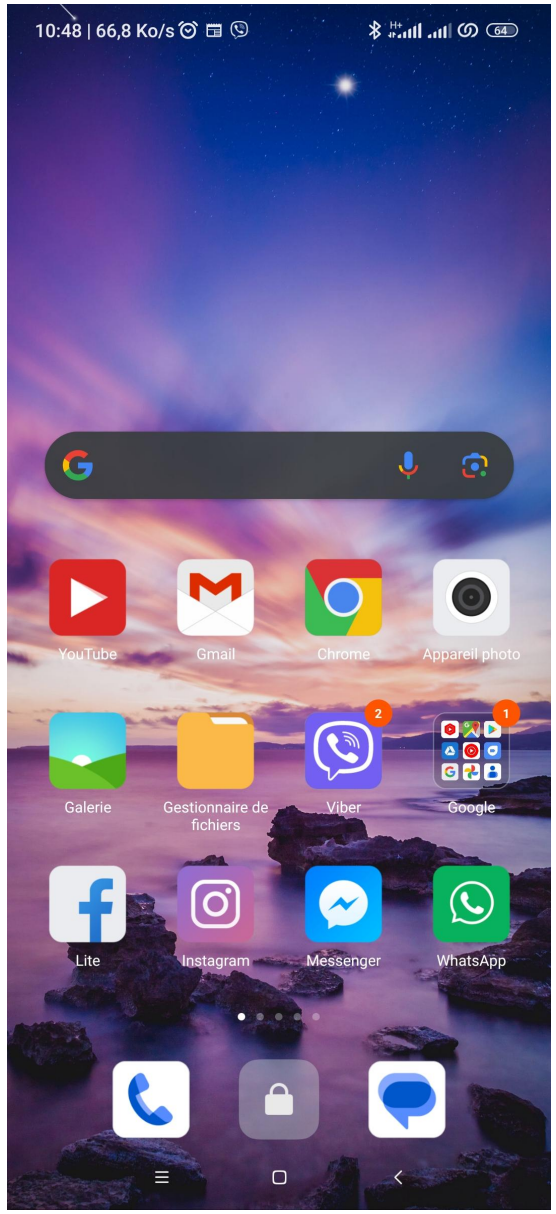
Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Introduction

Les images et les vidéos sont omniprésentes !!!

- Photos, Vidéos personnelles
- Réseaux sociaux : Insta, FB, discord, emoji, youtube, tiktok
- Plateforme de messagerie et d'appel vidéos : whatsapp, messenger, viber, skype
- Plateforme de visio-conférence : meet, zoom, teams, etc.
- Navigation : maps

Support de communication à l'instar du langage



Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Introduction

Application infinies !!!

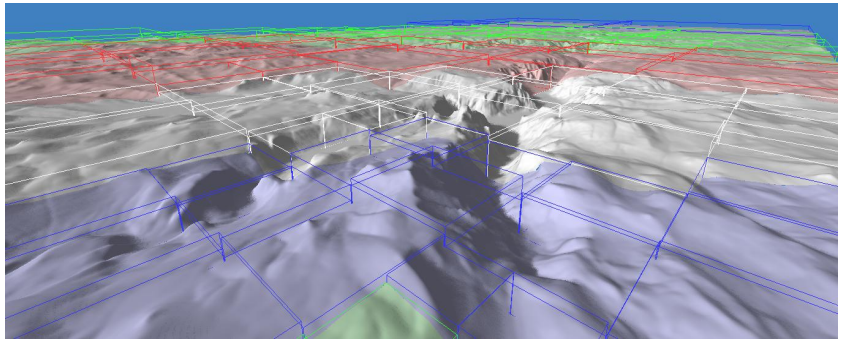
Sécurité

Réalité virtuelle

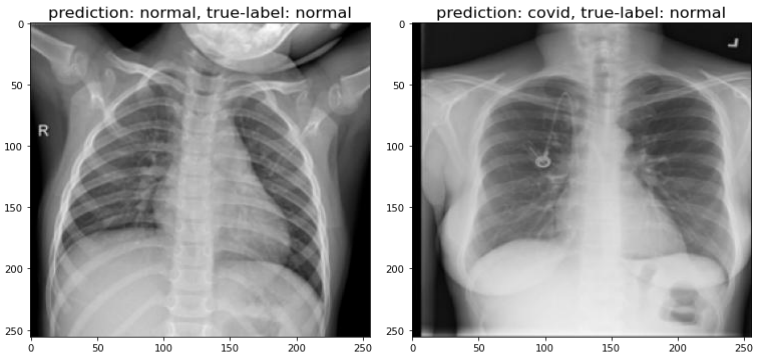
Robotique

Industrie

Guidage automatique et poursuite d'engins,

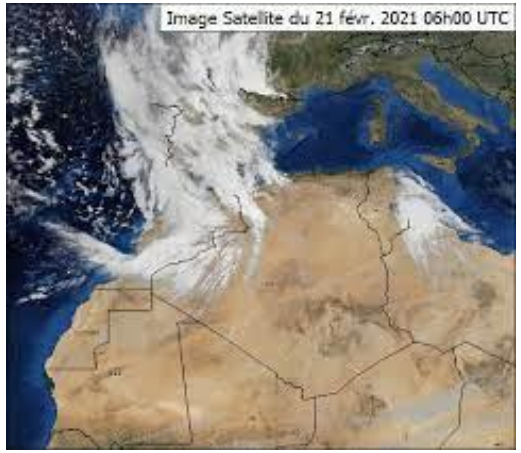


Médecine (Cytologie, Tomographie, Echographie. etc.

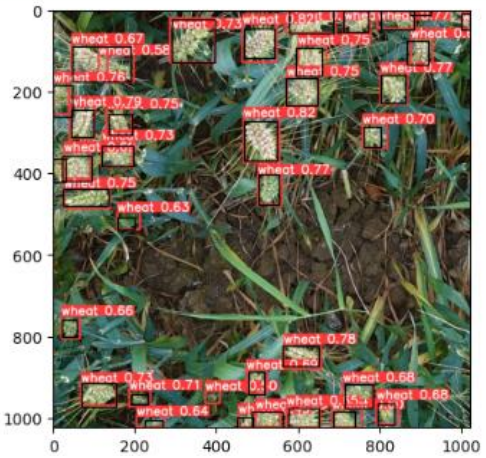


Biométrie

Imagerie aérienne et spatiale



Agriculture



Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Introduction

Utilisations vidéos

Apprendre

Bébé Requin Danse | Chante et danse! les enfants

14 Md de vues • il y a 7 ans

Pinkfong Baby Shark - Kids' Songs & Stories

PINKFONG! non. 1 app enfants choisi par 100 millions d'e

مطبخ ام وليد كيكة الكرامال و القهوة

7,5 M de vues • il y a 7 ans

Oum Walid

بحلية بالكرمال و القهوة

How To Fix a Water Damaged Laptop

61 M de vues • il y a 3 ans

HowToBasic

Today I show you how to fix a water damaged laptop. Acc

Learn Python - Full Course for Beginners [Tutorial]

43 M de vues • il y a 5 ans

freeCodeCamp.org

This course will give you a full introduction into all of the core concepts in python

Sous-titres

Introduction | Installing Python & PyCharm | Setup & Hello World | D

HAVE BEEN / HAS BEEN / HAD BEEN - Examples

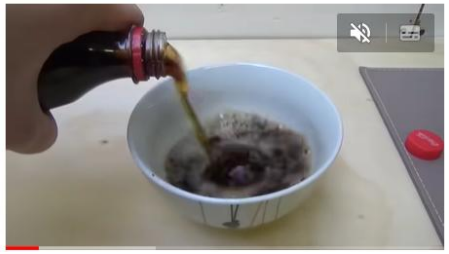
39 k vues • il y a 21 heures

English with Lucy

How do we use HAVE BEEN, HAS BEEN and HAD BEEN? L

Nouveau 4K

Introduction | Subject-Verb Agreement | Posi



How To Fix a Cracked iPhone Screen

60 M de vues • il y a 8 ans

HowToBasic

Today I show you how to easily fix any cracked smartphor

Abdos en 2 SEMAINES | Get Abs in 2 WEEKS |

531 M de vues • il y a 4 ans

Chloe Ting

Abdos ! Abdos ! Abdos ! Tout le monde m'a demandé un calendrier cc

Sous-titres

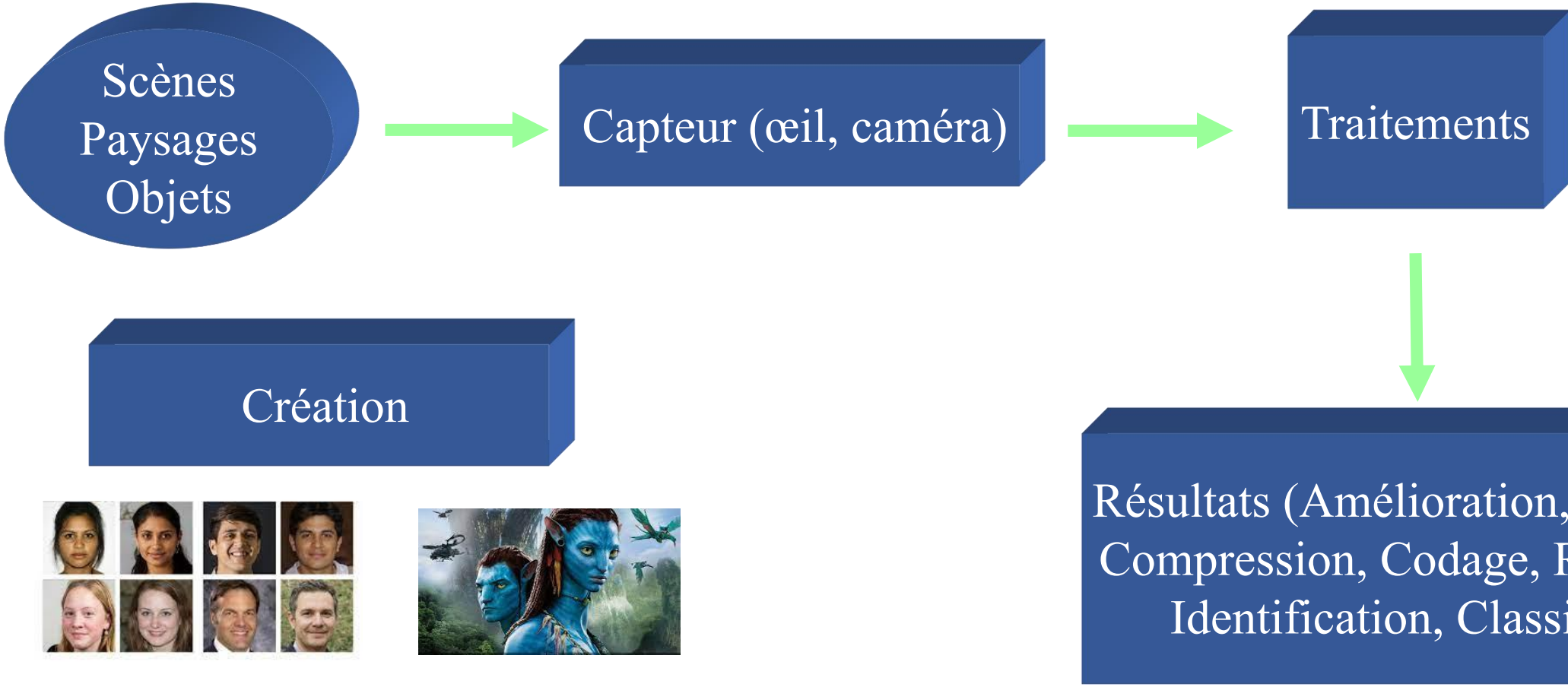
Intro | SPIDER-MAN PLANK | CROSSBODY MOUNTAIN C

Loisirs/Commercial : Films, Séries,Documentaires, publicité, promotion (vlog)

News (parfois fake → Manipulation)

Vidéo Surveillances : Rue, Routes, Aéroport, Magasin, etc.

Introduction



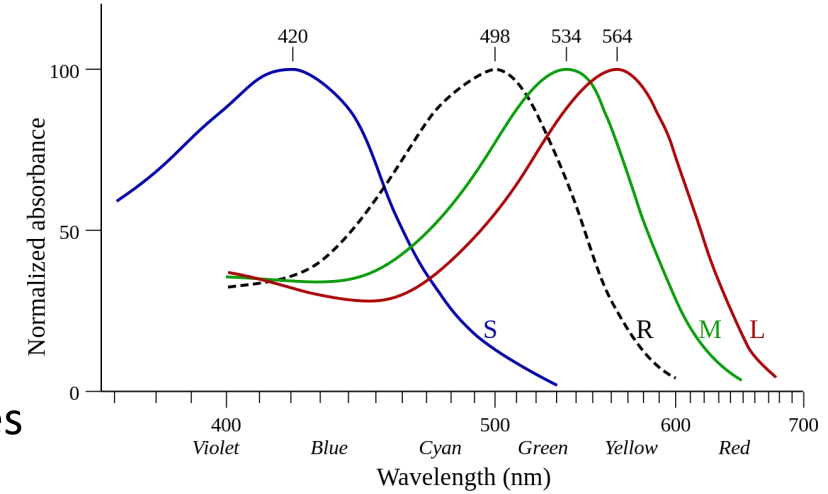
Introduction

Lumière du soleil : ondes électromagnétiques.

Les longueurs d'onde de la lumière visible vont de 380 à 780 nm.

- Rétine détecte les caractéristiques de la lumière à partir de ses photorécepteurs :

- les bâtonnets (R): insensibles à la couleur et sensibles au mouvement,
- les cônes qui sont sensibles aux couleurs. Les cônes sont de trois types appelés conventionnellement : bleu (S), vert (L) et rouge (M)

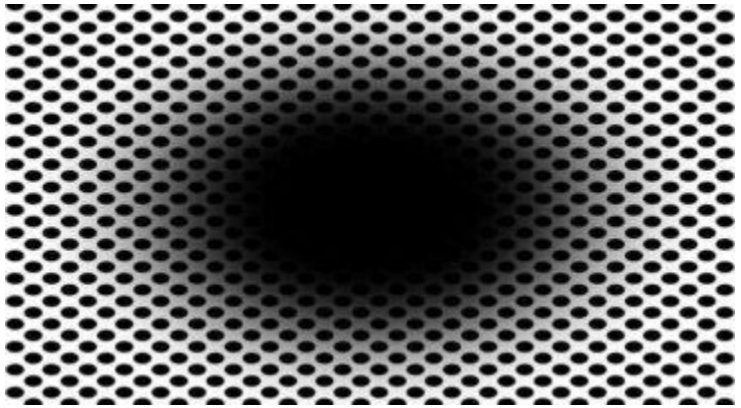


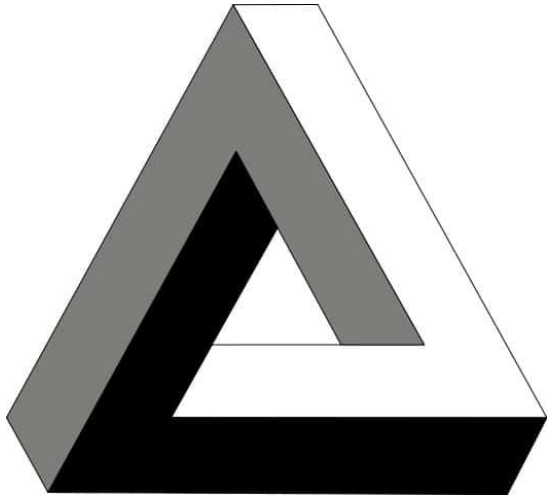
Par Vectorized version of the GFDL image Cone-response.png uploaded by User:Maxim Razin based on work by User:DrBob and User:Zeimusu. — After Bowmaker and Dartnall (1980). "Visual pigments of rods and cones in a human retina". J. Physiol. 298: 501-11. DOI:10.1113/jphysiol.1980.sp013097. PMID 7359434., CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2447660

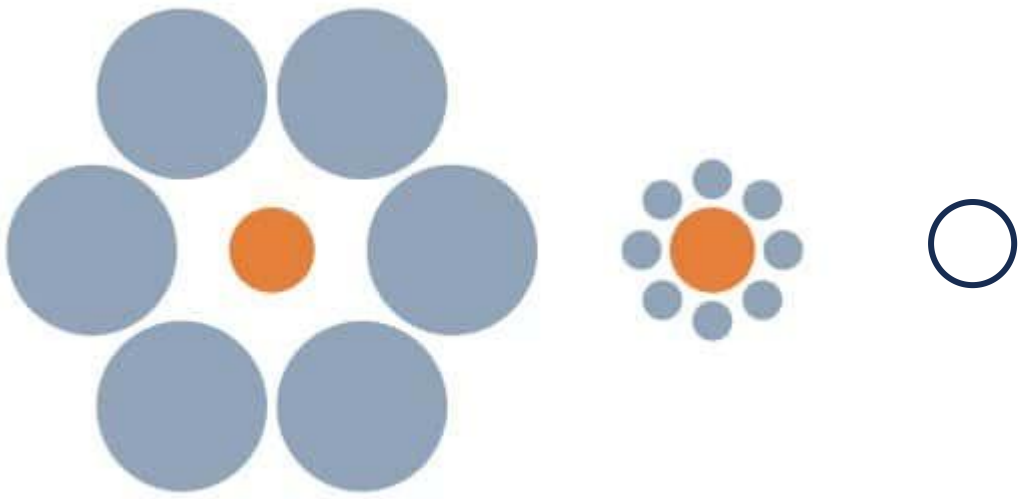
- Cerveau reçoit deux images plates, inversées, peu colorées, floues en bonne partie.
- L'aire V1 (cortex visuel primaire) extrait l'information visuelle : couleur, orientation, mouvement, contraste, fréquence spatiale, forme)
- Extraction et Traitement par des aires cérébrales séparées plus spécialisées : V3 (forme), V4 (couleur+orientation), V5 (mouvement)
- Informations regroupées et utilisées pour la reconnaissance de ce qui est vu et/ou pour l'action.

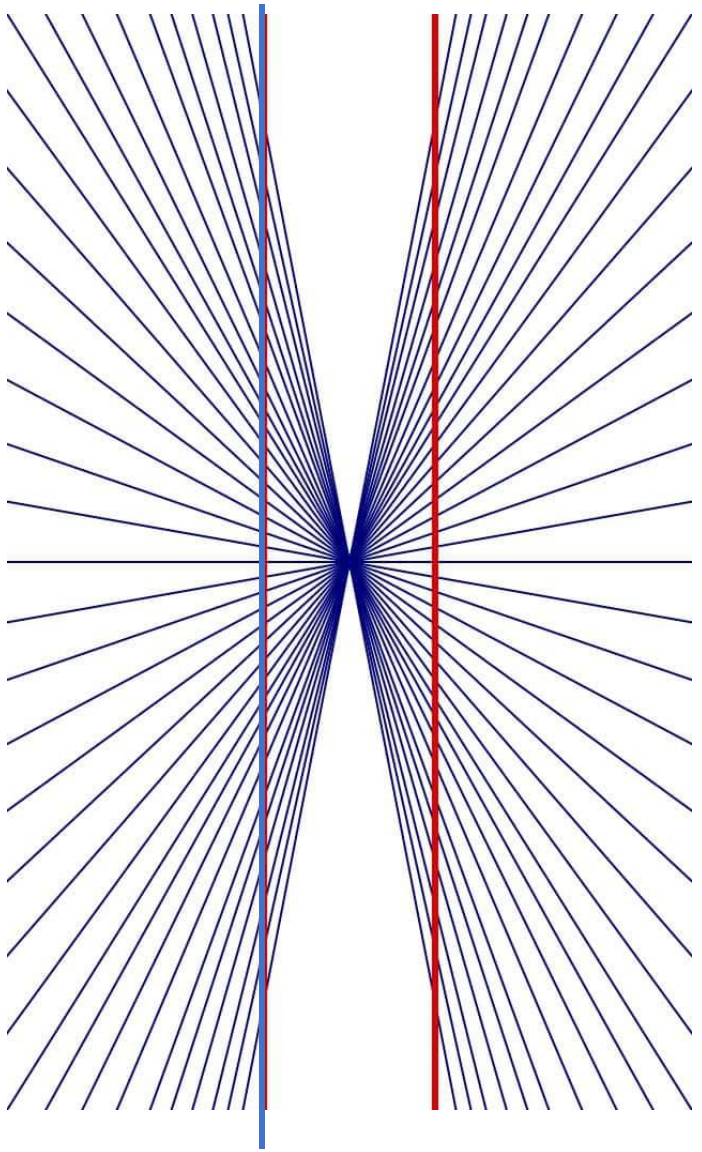
Triangle de Kaniza

Introduction





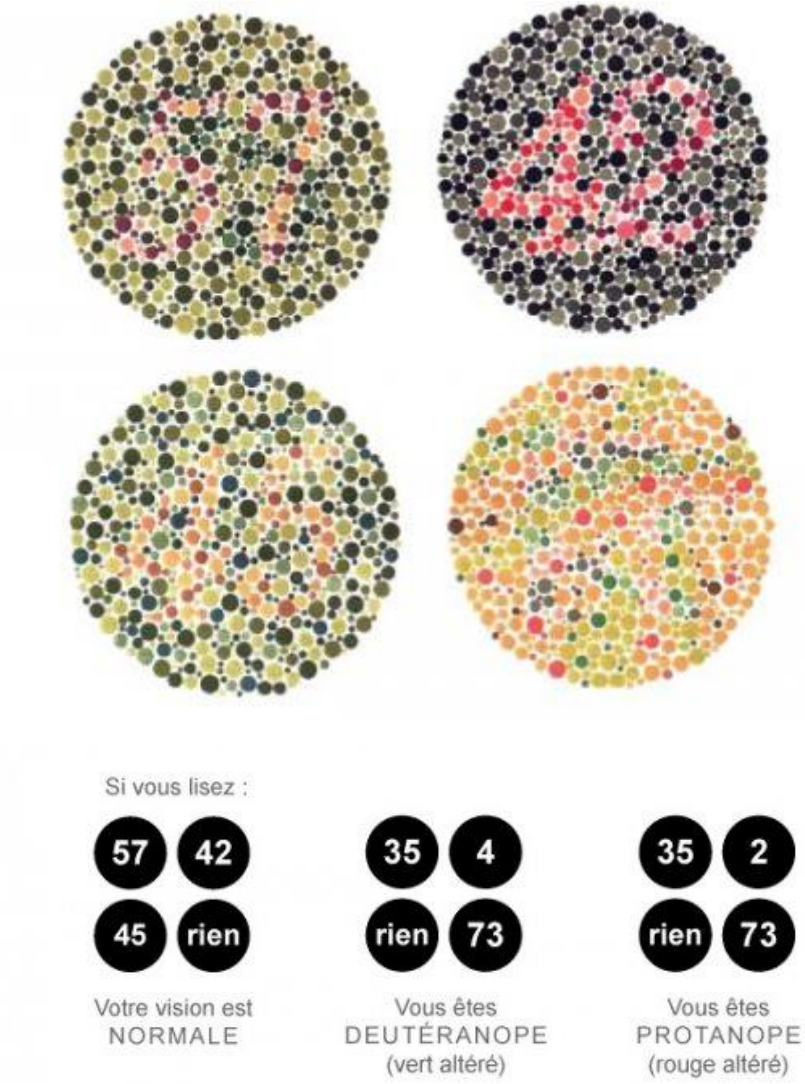




https://www.futura-sciences.com/sciences/photos/photos-top-15-illusions-optique-plus-surprenantes-691/

Introduction

https://illusionoftheyear.com/2009/05/the-illusion-of-sex/

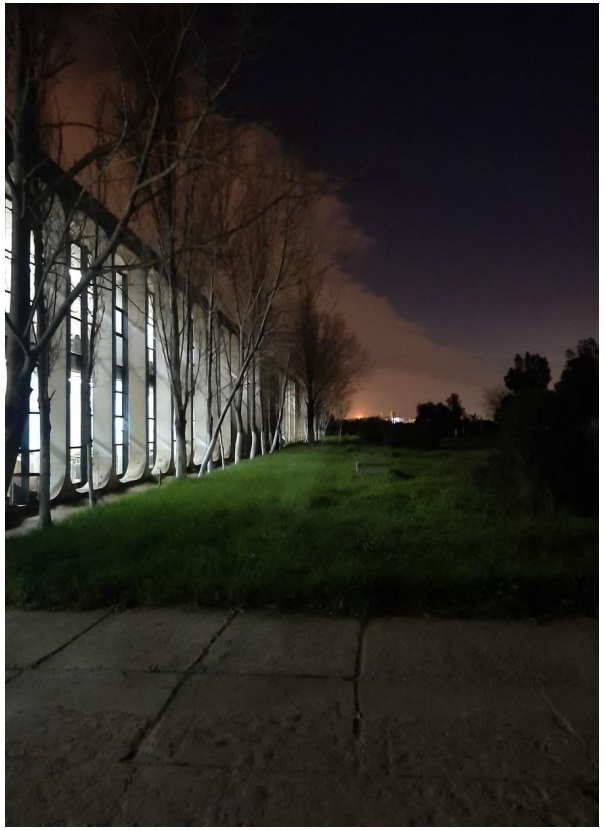


https://www.register-design.com/__trashed-2/

Définitions

Image numérique : forme discrète d'un phénomène continu.

- Créée, traitée, stockée sous forme binaire (suite de *0* et de *1*).
- Bidimensionnelle : L lignes et C colonnes
- L'information : caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).



Définitions

Image numérique : forme discrète d'un phénomène continu.

- Créée, traitée, stockée sous forme binaire (suite de *0* et de *1*).
- Bidimensionnelle : L lignes et C colonnes
- L'information : caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).
- Pixel : "picture element", unité de base de l'image correspondant à un pas de discrétisation.

Position et valeur (niveaux de gris ou couleur).

176	173	172	174	175	174	175
179	185	187	181	174	173	165
197	181	168	167	171	169	170
161	170	180	183	180	174	175

$$2^3 = 256$$

Définitions

Image numérique : forme discrète d'un phénomène continu.

- Créée, traitée, stockée sous forme binaire (suite de *0* et de *1*).
- Bidimensionnelle : L lignes et C colonnes
- L'information : caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).
- Pixel : "picture element", unité de base de l'image correspondant à un pas de discré

Position et valeur (niveaux de gris ou couleur).

- Représentation des couleurs : RGB, HSV, etc.

Canal bleu

51	51	53	54	60	57
15	30	44	53	55	58
89	91	95	99	102	98
53	70	86	95	97	99
101	103	107	110	114	107
65	82	98	107	109	108
73	86	101	113	111	110
113	111	115	116	115	113

60
65
101
104

Canal vert

Canal rouge

Définitions

Image numérique : forme discrète d'un phénomène continu.

- Créée, traitée, stockée sous forme binaire (suite de *0* et de *1*).
- Bidimensionnelle : L lignes et C colonnes
- L'information : caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).
- Pixel : "picture element", unité de base de l'image correspondant à un pas de discrétisation.

Position et valeur (niveaux de gris ou couleur).

- Représentation des couleurs : RGB, HSV, etc.

$$2^{8^3} \approx 16.\ 10^6$$

0.299 Rouge + 0.587 Vert + 0.114 Bleu → Niveaux de

couleur	R	G	B	Octet 1	Octet 2	Octet 3
	0	0	0	00000000	00000000	00000000
	255	255	255	11111111	11111111	11111111
	255	0	0	11111111	00000000	00000000
	0	255	0	00000000	11111111	00000000
	0	0	255	00000000	00000000	11111111
	132	122	191	10000100	01111010	10111111

Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

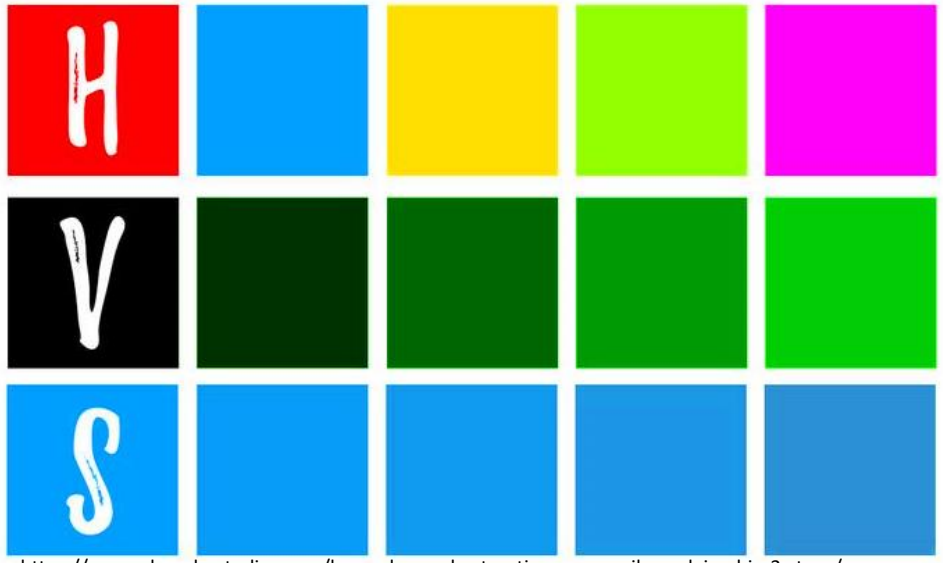
Définitions

Image numérique : forme discrète d'un phénomène continu.

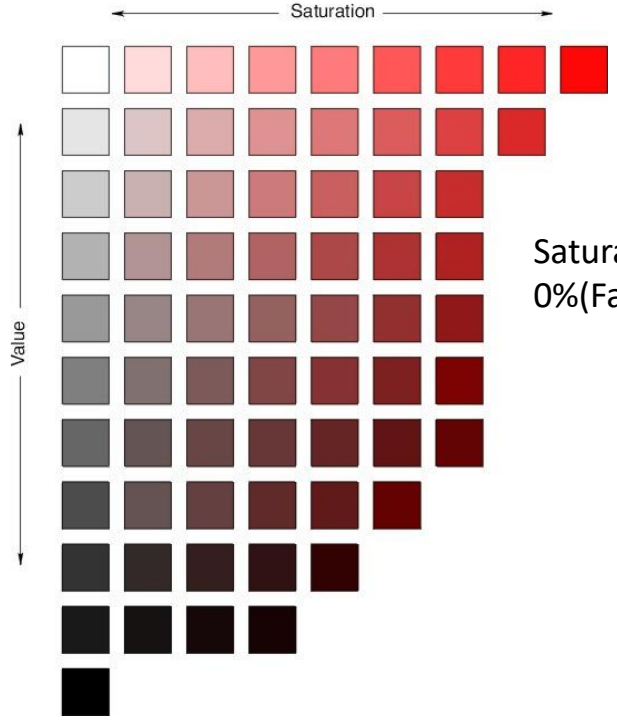
- Créée, traitée, stockée sous forme binaire (suite de *0* et de *1*).
- Bidimensionnelle : L lignes et C colonnes
- L'information : caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).
- Pixel : "picture element", unité de base de l'image correspondant à un pas de discrétisation.

Position et valeur (niveaux de gris ou couleur).

- Représentation des couleurs : RGB, HSV, etc.



https://www.oleanderstudios.com/hue-value-and-saturation-very-easily-explained-in-3-steps/



Saturation (Pureté, Intensité)
0%(Fade,Gris) → 100%(Intense)

Value (Brightness)
0% (Sombre→ 100%(Lumineuse)

https://learn.leighcotnoir.com/artspeak/elements-color/hue-value-saturation/cone_slice/

Définitions

Image numérique : forme discrète d'un phénomène continu.

- Créée, traitée, stockée sous forme binaire (suite de *0* et de *1*).
- Bidimensionnelle : L lignes et C colonnes
- L'information : caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).
- Pixel : "picture element", unité de base de l'image correspondant à un pas de discrétisation.

Position et valeur (niveaux de gris ou couleur).

- Représentation des couleurs : RGB, HSV, etc.

Remarque : Images Vectorielles SEED4NA-logo.pdf SEED4NA-logo.bmp

Les courbes et figures mathématiques de l'image sont stockées

Si une image vectorielle est agrandie (ou réduite), la distance entre les tracés est recalculée, puis ajustée dans les bonnes proportions

Définitions

Image numérique : forme discrète d'un phénomène continu.

- Créée, traitée, stockée sous forme binaire (suite de *0* et de *1*).
- Bidimensionnelle : L lignes et C colonnes (***Définition***)
- L'information : caractéristique de l'intensité lumineuse (couleur ou niveaux de gris).
- Pixel : "picture element", unité de base de l'image correspondant à un pas de discrétisation.

Position et valeur (niveaux de gris ou couleur).

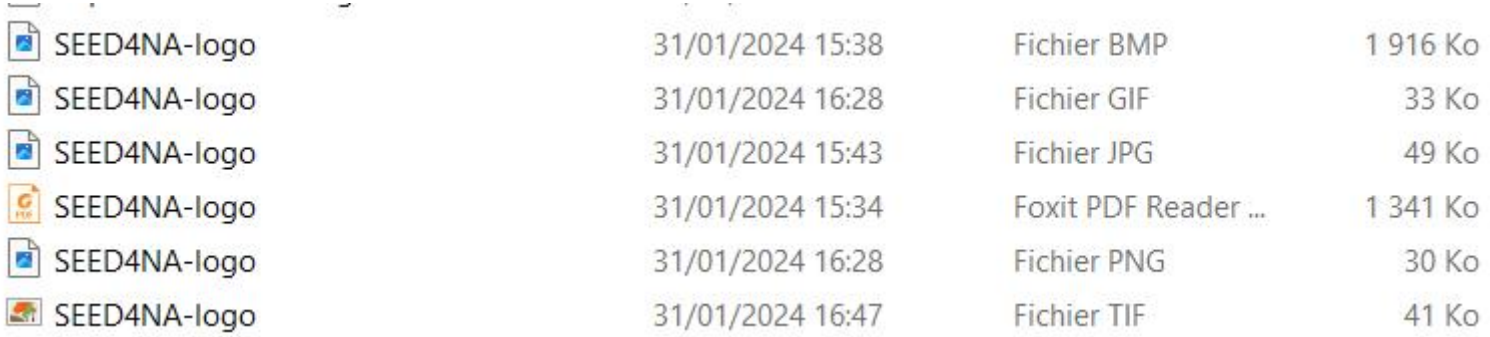
- Représentation des couleurs : RGB, HSV, etc.

- ***Résolution*** d'une image : le nombre de pixels contenus dans l'image par unité de longueur. Exprimée en **ppp** (point par pouces) ou en **dpi** (dots per inch). Elle définit la netteté et la qualité d'une image.

Définitions

Quelques Formats d'Images et Vidéos :

Formats	Avantages	Inconvénients
BMP *(Bitmap) :* Image graphique stockant les pixels sous forme de tableau de points.	Excellente qualité	Fichiers volumineux
GIF (*Graphics Interchange Format*) : Petites images, icônes, boutons des pages Web, etc.	Bonne compression, possibilité d'animation	Limité à 256 couleurs
.JPEG .JPG (*Joint Photography Experts Group*) : Photos et images texturées.	Fichiers compressés très compacts (>90%)	Perte de qualité
PNG (Portable Network Graphics) Destinés à remplacer GIF	Excellente compression sans perte	Moins efficace sur grandes photos
TIFF (Tagged Image File Format)	Compression sans perte. Couche de transparence.	Fichiers non compressés volumineux



Définitions

Quelques Formats d'Images et Vidéos :





SEED4NA-logo	31/01/2024 15:38	Fichier BMP	1 916 Ko
SEED4NA-logo	31/01/2024 16:28	Fichier GIF	33 Ko
SEED4NA-logo	31/01/2024 15:43	Fichier JPG	49 Ko
SEED4NA-logo	31/01/2024 15:34	Foxit PDF Reader ...	1 341 Ko
SEED4NA-logo	31/01/2024 16:28	Fichier PNG	30 Ko
SEED4NA-logo	31/01/2024 16:47	Fichier TIF	41 Ko

Définitions

Quelques Formats d'Images et Vidéos :

Formats	Avantages	Inconvénients
AVI (*Audio Video Interleave*)	Excellente qualité vidéo et audio, accepté par beaucoup de systèmes	Grande taille de fichier. incapacité à prendre en compte plusieurs pistes audio,
MP4 (*Moving Picture Expert Group*) Standard pour le Web	vidéos MP4 bon compromis qualité et tailles des fichiers	Processus d'encodage et de décodage nécessitent beaucoup de ressources.
MKV (*Matroska*)	Intègre en plus des fichiers vidéo, des pistes audio, des menus	Nécessité codec approprié disponible sur le net
MOV Quicktime Apple montage vidéo	Qualité supérieure à celle des fichiers MP4	Compresser la vidéo et la formater pour le Web,
WebM de Google	Taille extrêmement petite pour une qualité acceptable	Non pris en charge par d'autres navigateurs
WMV (*Windows Media Video*)	Excellente qualité tout en étant de petite taille	Non compatible avec Apple et Linux

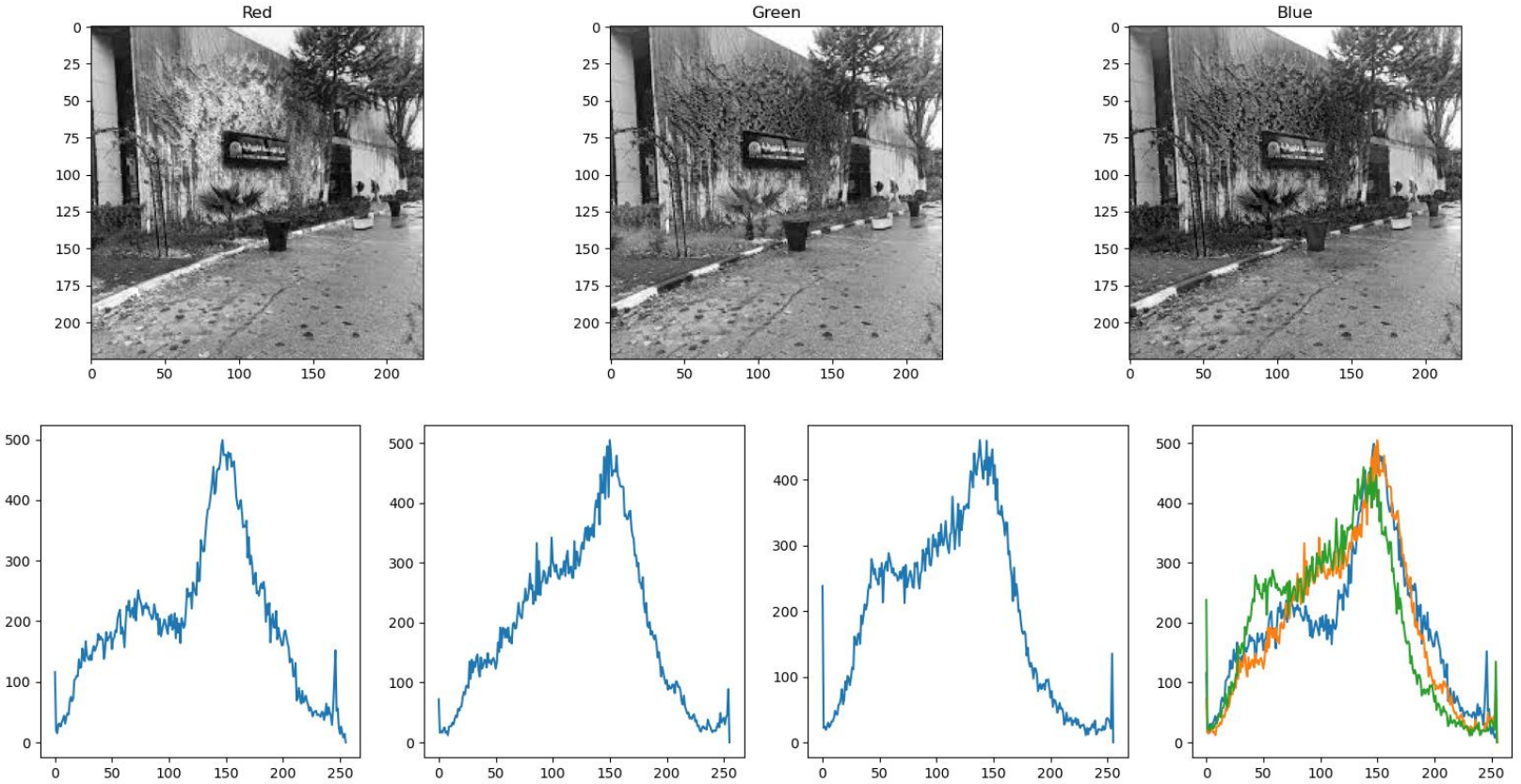
..\TImages\USTHB.mp4

Opérations de base

- ✓ Notions d'histogramme
- ✓ Correction de la dynamique de l'image par les transformations affines sur l'histogramme
- ✓ Filtrage spatial et Convolution 2D : notions de masque (moyenneur, gaussien, binomial, etc.)
- ✓ Lissage linéaire puis non linéaire de l'image (médian, …)
- ✓ Détection de contours (Objectifs, Types de contours, Masque de Roberts, Prewitt, Sobel, Opérateurs Laplacien, Filtre de Marr-Hildreth,…)
- ✓ Filtrage fréquentiel : (FFT 2D et propriété de séparabilité, filtre passe-bas, passe-haut, …)
- ✓ Opérations morphologiques (dilatation, érosion, ouverture, fermeture, …)
- ✓ Seuillage et Binarisation

Opérations de base : Histogramme

- ✓ Notions d'histogramme : une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels ayant cette valeur



Opérations de base : Egalisation d'histogramme

- ✓ Notions d'histogramme
- ✓ Correction de la dynamique de l'image par les transformations affines sur l'histogramme

L'égalisation d'histogramme est une méthode d'ajustement du contraste.

Cette transformation est construite à partir de l'histogramme cumulé de l'image de départ en ***étalant*** l'histogramme. Une transformation est appliquée sur chaque pixel de l'image, et donc d'obtenir une nouvelle image à partir d'une opération indépendante sur chacun des pixels.

$$T(x_k) = (L-1)\sum_{j=0}^{k} p(x_j) = \frac{(L-1)}{n}\sum_{j=0}^{k} n_j$$

Opérations de base : Egalisation d'histogramme

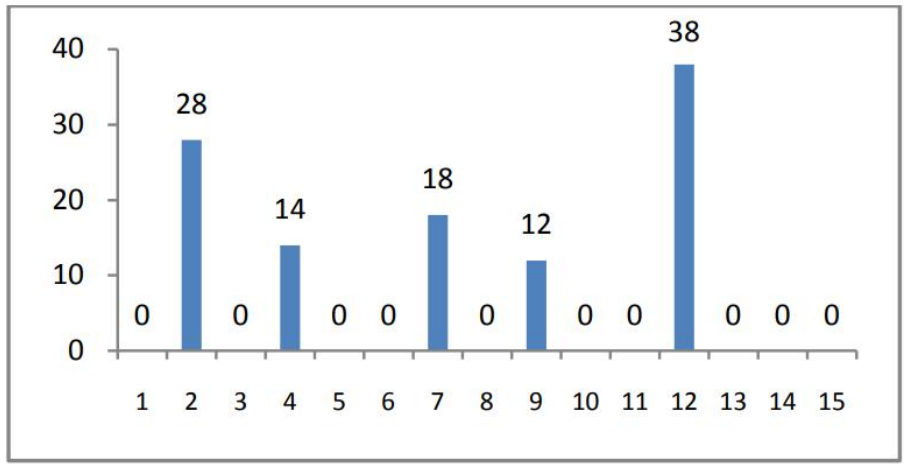
12	12	12	12	12	12	12	12	12	12
12	9	9	2	2	2	2	9	9	12
12	9	2	7	7	7	7	2	9	12
12	2	7	4	4	4	4	7	2	12
12	2	7	2	4	4	2	7	2	12
12	2	7	4	4	4	4	7	2	12
12	2	7	2	4	4	2	7	2	12
12	2	7	4	2	2	4	7	2	12
12	9	2	7	7	7	7	2	9	12
12	9	9	2	2	2	2	9	9	12
12	12	12	12	12	12	12	12	12	12

est une méthode d'ajustement du contraste.

nstruite à partir de l'histogramme cumulé de l'image de départ en transformation est appliquée sur chaque pixel de l'image, et donc à partir d'une opération indépendante sur chacun des pixels.

$$T(x_k) = (L-1)\sum_{j=0}^{k} p(x_j) = \frac{(L-1)}{N}\sum_{j=0}^{k} n_j$$

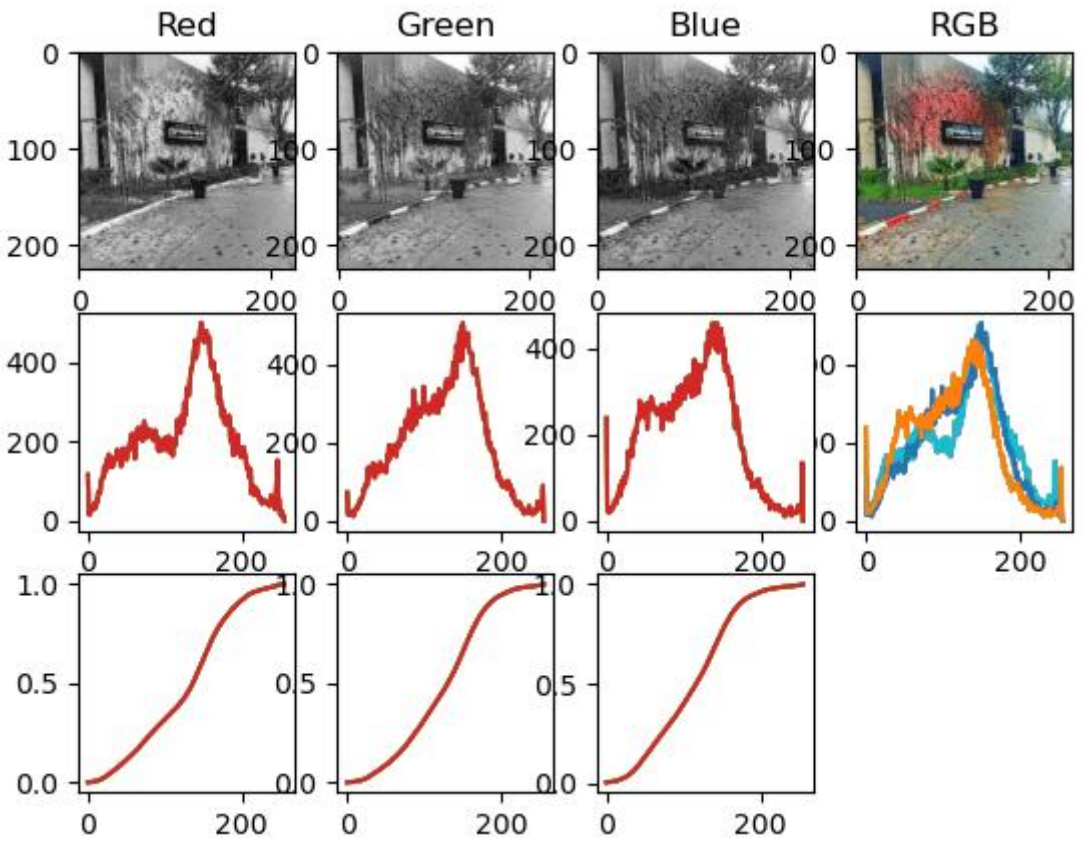
http://virtuelcampus.univ-msila.dz/factech/wp-content/uploads/2020/01/M2-ESEM-Vision-Artif.pdf

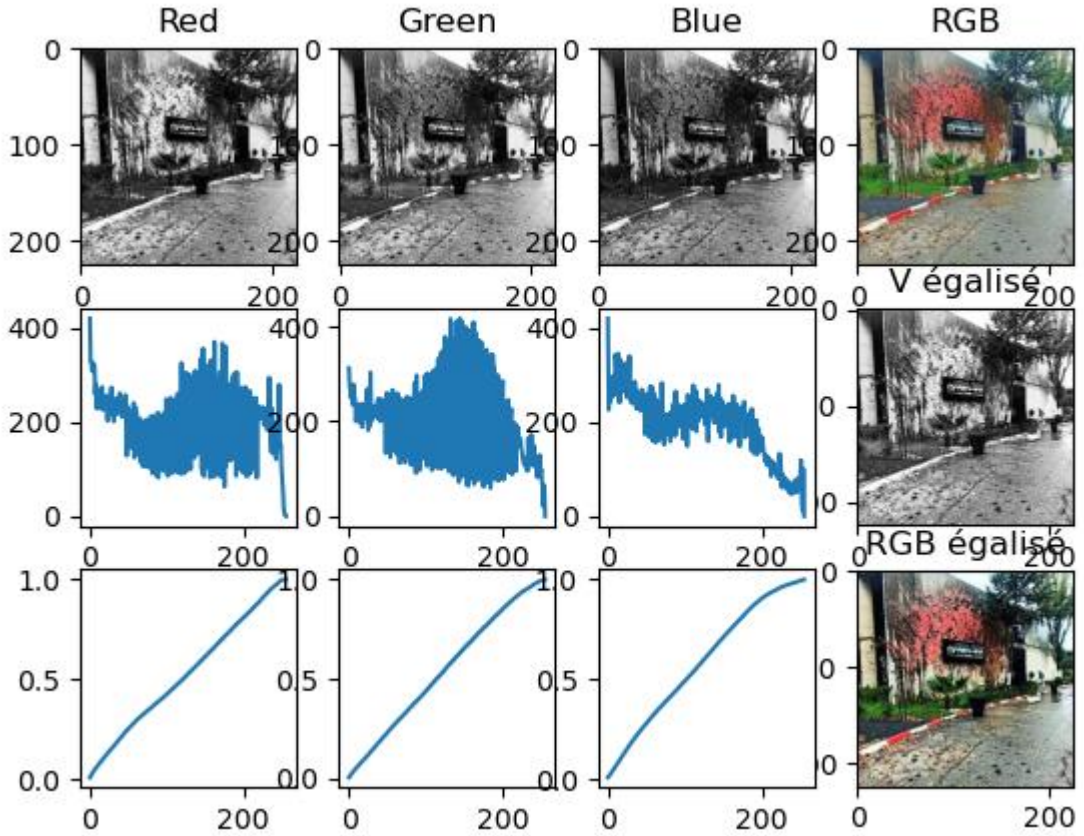


2	4
4	6
7	8
9	10
12	15

Opérations de base : Egalisation d'histogramme

✓ Notions d'histogramme





Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : notions de masque (moyenneur, gaussien, binomial, etc.)

0pération transformant une image en une autre image ayant des propriétés spatiales et fréquentielles différentes :

- Linéaire : Remplacer la valeur de chaque pixel par une moyenne pondérée calculée avec les pixels voisins. Le masque contient les coefficients de pondérations de chacun des pixels.

$$y(m,n) = \sum_i \sum_j h(i,j)x\,(m-i, n-j)$$

Exemple

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

$$y(m,n) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} h(i,j)x(m-i, n-j)$$

x(0,0)	x(0,1)	x(0,2)	x(0,3)	x(0,4)
x(1,0)	x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,0)	x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,0)	x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,0)	x(4,1)	x(4,2)	x(4,3)	x(4,4)

Appliquer le filtre h(i,j) revient à balayer l'image avec le masque g(i,j) = h(-i,-j)

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : notions de masque (moyenneur, gaussien, binomial, etc.)

0pération transformant une image en une autre image ayant des propriétés spatiales et fréquentielles différentes :

- Linéaire : Remplacer la valeur de chaque pixel par une moyenne pondérée calculée avec les pixels voisins. Le masque contient les coefficients de pondérations de chacun des pixels.

$$y(m,n) = \sum_i \sum_j h(i,j)x\,(m-i, n-j)$$

Exemple

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

$$y(m,n) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} h(i,j)x(m-i, n-j)$$

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

x(0,0)	x(0,1)	x(0,2)	x(0,3)	x(0,4)
x(1,0)	x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,0)	x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,0)	x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,0)	x(4,1)	x(4,2)	x(4,3)	x(4,4)

Appliquer le filtre h(i,j) revient à balayer l'image avec le masque g(i,j) = h(-i,-j)

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j h(i,j)x\,(m-i,n-j)$$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i,n+j)$$

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9								

Opérations de base : Filtrage Spatial

h : filtre

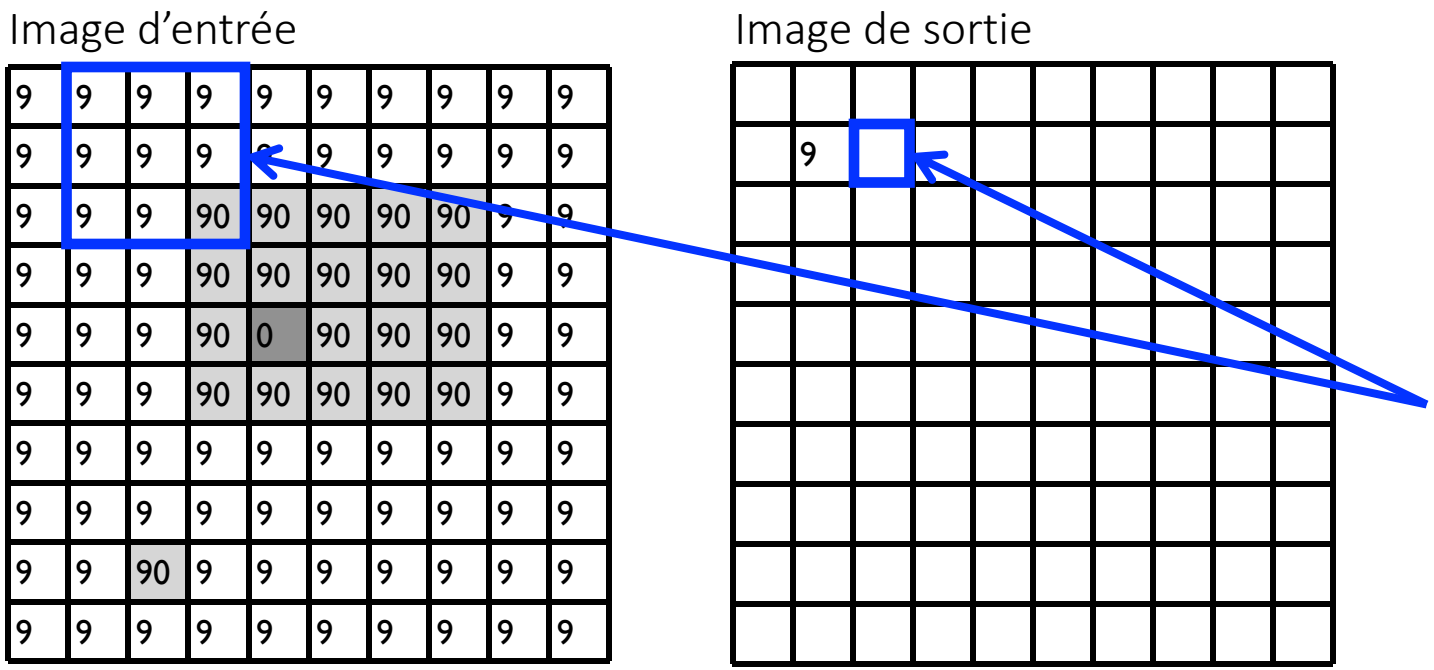
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18							

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18							

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27						

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27						

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36					

Opérations de base : Filtrage Spatial

h : filtre

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

g : masque

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36				

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27		

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9								

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9	27							

Opérations de base : Filtrage Spatial

h : filtre

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

g : masque

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9	27	45						

Opérations de base : Filtrage Spatial

h : filtre

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

g : masque

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9	27	45	63					

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9	27	45	63	63	63	45	27	

Opérations de base : Filtrage Spatial

h : filtre

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

g : masque

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9	27	45	63	63	63	45	27	
	9								

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9	27	45	63	63	63	45	27	
	9	36	53	80	80	90	63	36	
	9	36	53	80	80	90	63	36	
	9	27	35	53	53	63	45	27	
	9	18	27	36	36	36	27	18	
	18	18	18	9	9	9	9	9	
	18								

Opérations de base : Filtrage Spatial

h : filtre

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

g : masque

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Medium

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	18	27	36	36	36	27	18	
	9	27	45	63	63	63	45	27	
	9	36	53	80	80	90	63	36	
	9	36	53	80	80	90	63	36	
	9	27	35	53	53	63	45	27	
	9	18	27	36	36	36	27	18	
	18	18	18	9	9	9	9	9	
	18	18	18	9	9	9	9	9	

Opérations de base : Filtrage Spatial

h : filtre

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Medium

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Gaussien

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening ‘Rehaussement »

Remarque : Filtres séparables

$$\begin{bmatrix} a & b & c \end{bmatrix} \otimes \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} a\alpha & b\alpha & c\alpha \\ a\beta & b\beta & c\beta \\ a\gamma & b\gamma & c\gamma \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \times \begin{bmatrix} a & b & c \end{bmatrix}.$$

$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \otimes \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

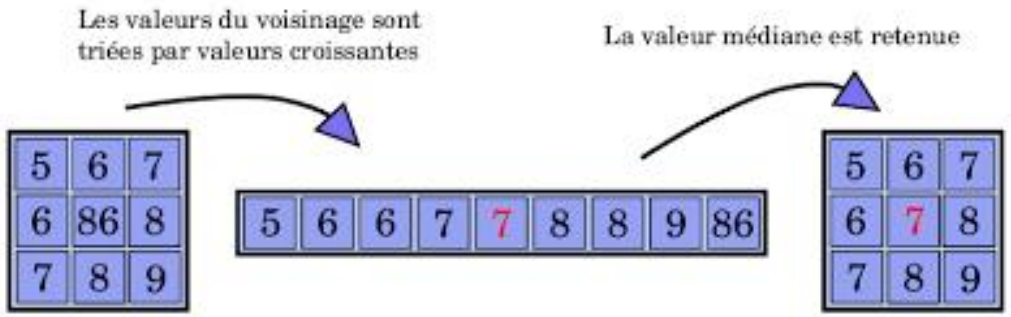
Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : notions de masque (moyenneur, gaussien, binomial, etc.)

0pération transformant une image en une autre image ayant des propriétés spatiales et fréquentielles différentes :

- Linéaire : Remplacer la valeur de chaque pixel par une moyenne pondérée calculée avec les pixels voisins. Le masque contient les coefficients de pondérations de chacun des pixels.

- Non linéaire : *remplacer la valeur de chaque pixel à partir des pixels voisins* (Median, Bilatéral)
-



Opérations de base : Filtrage Spatial

✓ Filtrage spatial

- Non linéaire : *remplacer la valeur de chaque pixel à partir des pixels voisins* (Median, Bilatéral)
-

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9								

Opérations de base : Filtrage Spatial

✓ Filtrage spatial

- Non linéaire : *remplacer la valeur de chaque pixel à partir des pixels voisins* (Median, Bilatéral)
-

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	9	9	9	9	9	9	9	

Opérations de base : Filtrage Spatial

✓ Filtrage spatial

- Non linéaire : *remplacer la valeur de chaque pixel à partir des pixels voisins* (Median, Bilatéral)
-

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	9	9	9	9	9	9	9	
	9	9	9	90	90	90	9	9	

Opérations de base : Filtrage Spatial

✓ Filtrage spatial

- Non linéaire : *remplacer la valeur de chaque pixel à partir des pixels voisins* (Median, Bilatéral)
-

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	9	9	9	9	9	9	9	9	
	9	9	9	90	90	90	9	9	
	9	9	90	90	90	90	9	9	
	9	9	90	90	90	90	9	9	
	9	9	9	90	90	90	9	9	
	9	9	9	9	9	9	9	9	
	9	9	9	9	9	9	9	9	
	9	9	9	9	9	9	9	9	

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradients

Dérivée première continue : $f'(t) = \lim_{u \to 0} \frac{f(t+u) - f(t)}{u}$ → Discret : Différences finies $f'(n) = \frac{f(n+k) - f(n)}{k} = \frac{f(n) - f(n-k)}{k}$

Image : 2 Dimensions $d_x f(n,m) = \frac{f(n+k,m) - f(n,m)}{k}$ ou $d_y f(n,m) = \frac{f(n,m+k) - f(n,m)}{k}$

k = 1 → Décalage d'1/2 pixel → $h = [-1 \;\; 1]$

x(0,0)	x(0,1)	x(0,2)	x(0,3)	x(0,4)
x(1,0)	x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,0)	x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,0)	x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,0)	x(4,1)	x(4,2)	x(4,3)	x(4,4)

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradients

Dérivée première continue : $f'(t) = \lim_{u \to 0} \frac{f(t+u) - f(t)}{u}$ → Discret : Différences finies $f'(n) = \frac{f(n+k) - f(n)}{k} = \frac{f(n) - f(n-k)}{k}$

Image : 2 Dimensions $d_x f(n,m) = \frac{f(n+k,m) - f(n,m)}{k}$ ou $d_y f(n,m) = \frac{f(n,m+k) - f(n,m)}{k}$

k = 1 → Décalage d'1/2 pixel → $h = [-1 \;\; 1]$

k = 2 → $h = [-1 \;\; 0 \;\; 1]$ → $g = [1 \;\; 0 \;\; -1]$

x(0,0)	x(0,1)	x(0,2)	x(0,3)	x(0,4)
x(1,0)	x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,0)	x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,0)	x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,0)	x(4,1)	x(4,2)	x(4,3)	x(4,4)

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradients

Dérivée première continue : $f'(t) = \lim_{u \to 0} \frac{f(t+u)-f(t)}{u}$ → Discret : Différences finies $f'(n) = \frac{f(n+k)-f(n)}{k} = \frac{f(n)-f(n-k)}{k}$

Image : 2 Dimensions $d_x f(n,m) = \frac{f(n+k,m)-f(n,m)}{k}$ ou $d_y f(n,m) = \frac{f(n,m+k)-f(n,m)}{k}$

k = 1 → Décalage d'1/2 pixel → $h = [-1 \;\; 1]$

k = 2 → $h = [-1 \;\; 0 \;\; 1]$ → $g = [1 \;\; 0 \;\; -1]$

x(0,0)	x(0,1)	x(0,2)	x(0,3)	x(0,4)
x(1,0)	x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,0)	x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,0)	x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,0)	x(4,1)	x(4,2)	x(4,3)	x(4,4)

g : Filtre

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

Filtre Prewitt horizontal

=

1
1
1

Lissage

·

1	0	-1

Dérivation

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

Filtre Prewitt vertical

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0								

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27							

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27	-27						

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27	-27	0					

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27	-27	0	0	0	27	27	

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

-I	-I	-I
-0	0	0
I	I	I

g : masque

$\frac{1}{3}$

I	I	I
0	0	0
-I	-I	-I

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0								

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x(m+i,n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27							

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x(m+i,n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27	-54						

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27	-27	-81					

Balayer l'image avec le masque

$g(i,j) = h(-i,-j)$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial et Convolution 2D : Gradients

h : filtre

$\frac{1}{3}$

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

g : masque

$\frac{1}{3}$

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

Image d'entrée

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

Image de sortie

	0	-27	-27	-54	-81	-81	-54	-27	

Balayer l'image avec le masque

$$g(i,j) = h(-i,-j)$$

$$y(m,n) = \sum_i \sum_j g(i,j)x\,(m+i, n+j)$$

Opérations de base : Filtrage Spatial

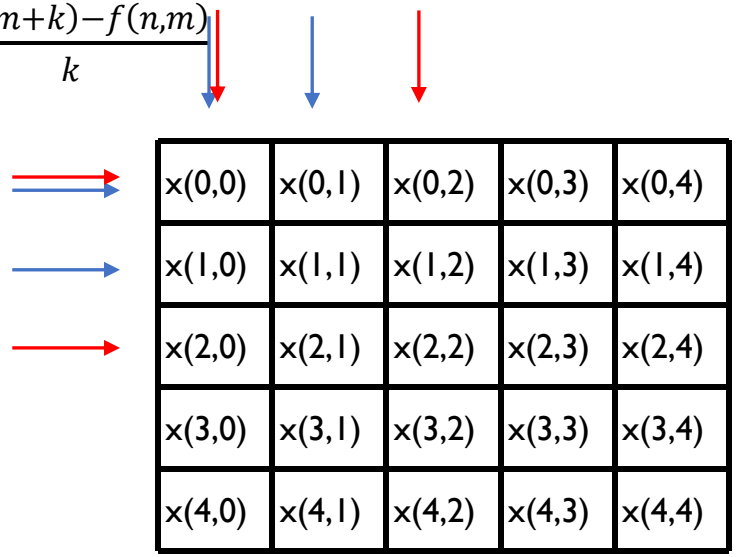
✓ Filtrage spatial : Gradients

Dérivée première continue : $f'(t) = \lim_{u \to 0} \frac{f(t+u)-f(t)}{u}$ → Discret : Différences finies $f'(n) = \frac{f(n+k)-f(n)}{k}$

Image : 2 Dimensions $d_x f(n,m) = \frac{f(n+k,m)-f(n,m)}{k}$ ou $d_y f(n,m) = \frac{f(n,m+k)-f(n,m)}{k}$

k = 1 → Décalage d'1/2 pixel → $h = [-1 \;\; 1]$

k = 2 → $h = [-1 \;\; 0 \;\; 1]$ → $g = [1 \;\; 0 \;\; -1]$



g : Filtre

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

Filtre Prewitt horizontal

=

1
1
1

Lissage

·

1	0	-1

Dérivation

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

Filtre Prewitt vertical

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradients

Dérivée première continue : $f'(t) = \lim_{u \to 0} \frac{f(t+u)-f(t)}{u}$ → Discret : Différences finies $f'(n) = \frac{f(n+k)-f(n)}{k}$

Image : 2 Dimensions $d_x f(n,m) = \frac{f(n+k,m)-f(n,m)}{k}$ ou $d_y f(n,m) = \frac{f(n,m+k)-f(n,m)}{k}$

k = 1 → Décalage d'1/2 pixel → $h = [-1 \;\; 1]$

k = 2 → $h = [-1 \;\; 0 \;\; 1]$ → $g = [1 \;\; 0 \;\; -1]$

x(0,0)	x(0,1)	x(0,2)	x(0,3)	x(0,4)
x(1,0)	x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,0)	x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,0)	x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,0)	x(4,1)	x(4,2)	x(4,3)	x(4,4)

g : Filtre

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

Filtre Sobel horizontal

=

1
2
1

Lissage

·

1	0	-1

Dérivation

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

Filtre Sobel vertical

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradients

Dérivée première continue : $f'(t) = \lim_{u \to 0} \frac{f(t+u)-f(t)}{u}$ → Discret : Différences finies $f'(n) = \frac{f(n+k)-f(n)}{k} = \frac{f(n)-f(n-k)}{k}$

Image : 2 Dimensions $d_x f(n,m) = \frac{f(n+k,m)-f(n,m)}{k} = \frac{f(n,m)-f(n-k,m)}{k}$ ou $d_y f(n,m) = \frac{f(n,m+k)-f(n,m)}{k} = \frac{f(n,m)-f(n,m-k)}{k}$

Dérivée seconde :

Horizontal → $\Delta_x f(n,m) = d_x(d_x f(n,m)) = d_x\left(\frac{f(n,m)-f(n-k,m)}{k}\right) = \frac{\left(f(n+k,m)-f(n,m)\right)-\left(f(n,m)-f(n-k,m)\right)}{k^2} = \frac{f(n+k,m)-2f(n,m)+f(n-k,m)}{k^2}$

Vertical → $\Delta_y f(n,m) = d_y(d_y f(n,m)) = d_y\left(\frac{f(n,m)-f(n,m-k)}{k}\right) = \frac{\left(f(n,m+k)-f(n,m)\right)-\left(f(n,m)-f(n,m-k)\right)}{k^2} = \frac{f(n,m+k)-2f(n,m)+f(n,m-k)}{k^2}$

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradients

Dérivée seconde :

Horizontal → $\Delta_x f(n,m) = d_x(d_x f(n,m)) = d_x\left(\frac{f(n,m)-f(n-k,m)}{k}\right) = \frac{\left(f(n+k,m)-f(n,m)\right)-\left(f(n,m)-f(n-k,m)\right)}{k^2} = \frac{f(n+k,m)-2f(n,m)+f(n-k,m)}{k^2}$

k=1 → $\Delta_x f(n,m) = f(n+1,m) - 2f(n,m) + f(n-1,m)$ → $h = [1 \quad -2 \quad 1]$

k=1 → $\Delta_x f(n,m) = f(n,m+1) - 2f(n,m) + f(n,m-1)$ → $h = [1 \quad -2 \quad 1]^t$

$\frac{1}{4}$

0	1	0
1	-4	1
0	1	0

$\frac{1}{4}$

Filtre Laplacien

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradients

Dérivée seconde :

Horizontal → $\Delta_x f(n,m) = d_x(d_x f(n,m)) = d_x\left(\frac{f(n,m)-f(n-k,m)}{k}\right) = \frac{(f(n+k,m)-f(n,m))-(f(n,m)-f(n-k,m))}{k^2} = \frac{f(n+k,m)-2f(n,m)+f(n-k,m)}{k^2}$

k=1 → $\Delta_x f(n,m) = f(n+1,m) - 2f(n,m) + f(n-1,m)$ → $h = [1 \quad -2 \quad 1]$

k=1 → $\Delta_x f(n,m) = f(n,m+1) - 2f(n,m) + f(n,m-1)$ → $h = [1 \quad -2 \quad 1]^t$

$\frac{1}{4}$

0	-1	0
-1	4	-1
0	-1	0

$\frac{1}{8}$

-1	-1	-1
-1	8	-1
-1	-1	-1

Filtre Laplacien

$\frac{1}{4}$

1	-2	1
-2	4	-2
1	-2	1

Filtre Gaussien Laplacien

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradient et Laplacien

Dérivée première et dérivée seconde :

k = 2 → $h = [-1 \;\; 0 \;\; 1]$ → $g = [1 \;\; 0 \;\; -1]$

k=1 → $h = g = [1 \;\; -2 \;\; 1]$

9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

	0	-81	-81	0	0	0	81	81	

Opérations de base : Filtrage Spatial

✓ Filtrage spatial : Gradient et Laplacien

Dérivée première et dérivée seconde :

k = 2 → $h = [-1 \;\; 0 \;\; 1]$ → $g = [1 \;\; 0 \;\; -1]$

k=1 → $h = g = [1 \;\; -2 \;\; 1]$

	0	-81	-81	0	0	0	81	81	

→

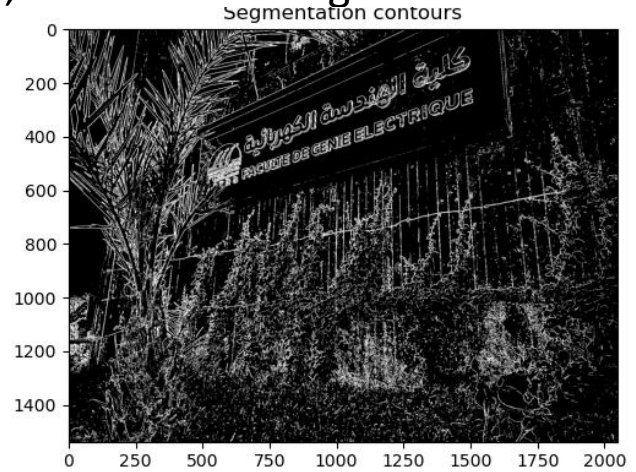
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	90	0	90	90	90	9	9
9	9	9	90	90	90	90	90	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	90	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9

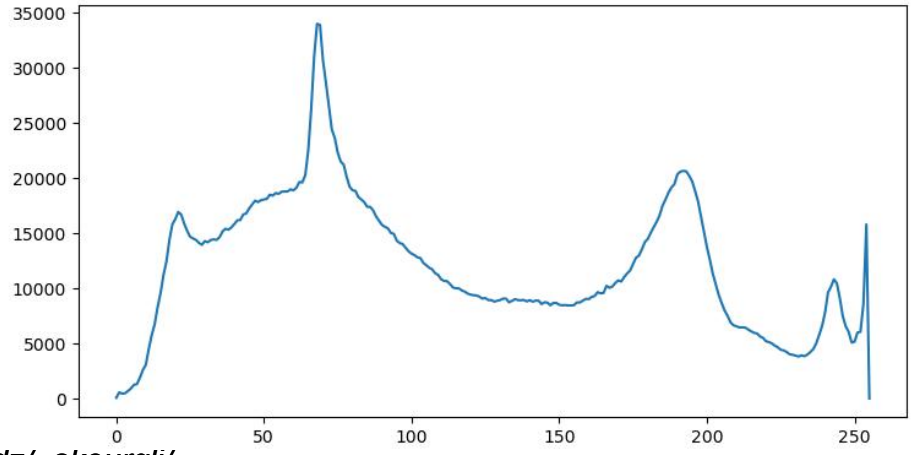
	0	81	-81	0	0	0	-81	63	

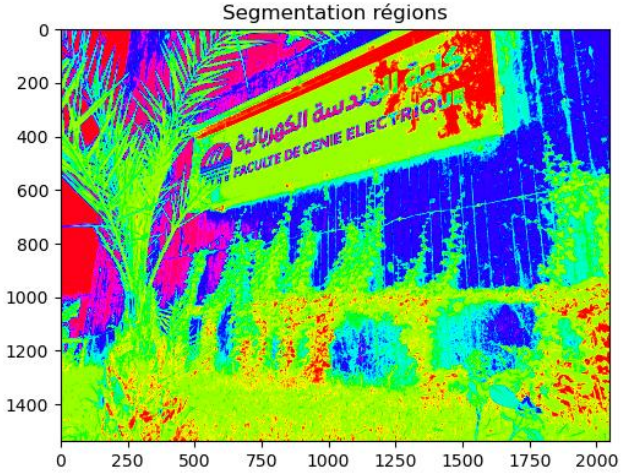
Segmentation

La segmentation d'image est le processus de partitionnement d'une image numérique en plusieurs régions d'image ou objets d'image (ensembles de pixels). Elle est généralement utilisé pour localiser des objets et des frontières (lignes, courbes, etc.) dans les images. Une étiquette est attribuée aux pixels partageant des caractéristiques communes.

Approches : classiques : seuillage, périphérie, clustering/classification, basées sur la région et l'IA



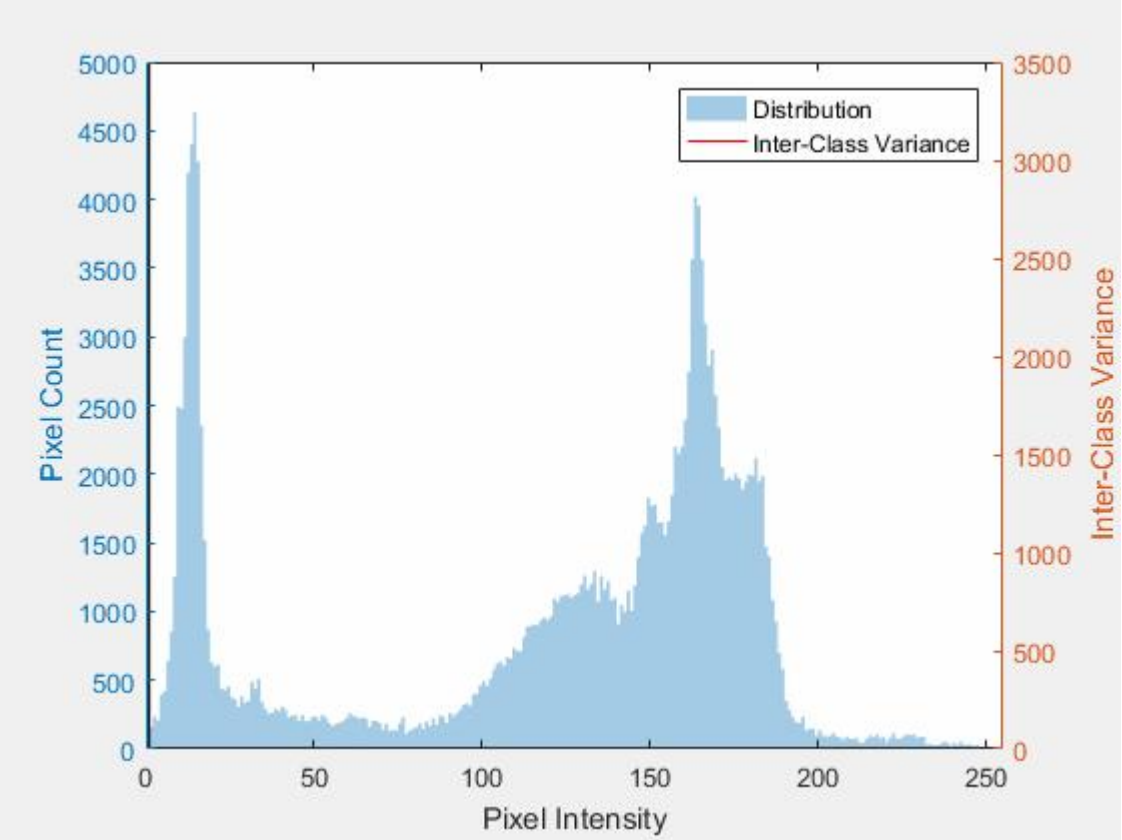




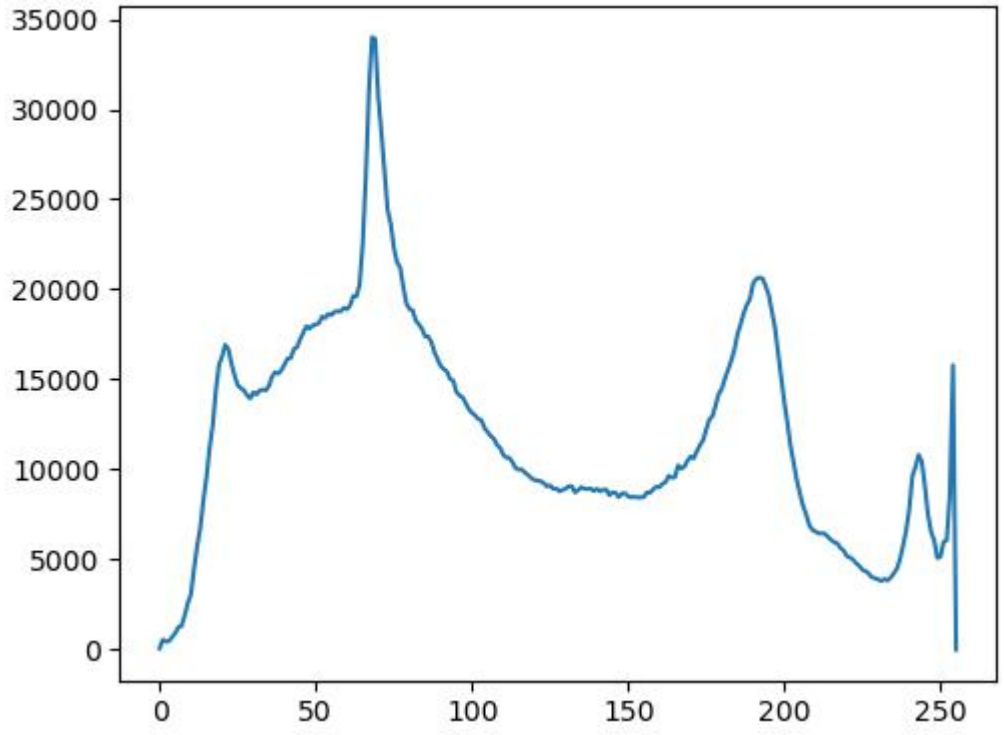
Segmentation using color

Le seuillage multi-Otsu est une extension de la méthode de seuillage d'Otsu, qui est utilisée pour segmenter une image en plusieurs classes ou régions en fonction des valeurs d'intensité des pixels.

L'image est divisée en classes K. L'algorithme recherche de manière itérative les seuils qui maximisent la variance entre les classes, conduisant à la meilleure séparation des niveaux d'intensité.



https://en.wikipedia.org/wiki/Otsu%27s_method#/media/File:Otsu's_Method_Visualization.gif



85
160

55
101
155
211

Segmentation

KMeans est un algorithme itératif qui tente de partitionner l'ensemble de données en K sous-groupes (clusters) distincts et prédéfinis qui ne se chevauchent pas, où chaque point de données appartient à un seul groupe **.**

Entrée:

Spécifier le nombre K de clusters

Ensemble de formation (Matrice de données)

Début

Choisir K points au hasard (une ligne de la matrice de données). Ces points sont les centres des clusters (appelés centroïde).

RÉPÉTITION

Calculer la somme de la distance au carré entre les points de données et tous les centroïdes.

Attribuer chaque point (élément de la matrice de données) au cluster le plus proche (centroïde)

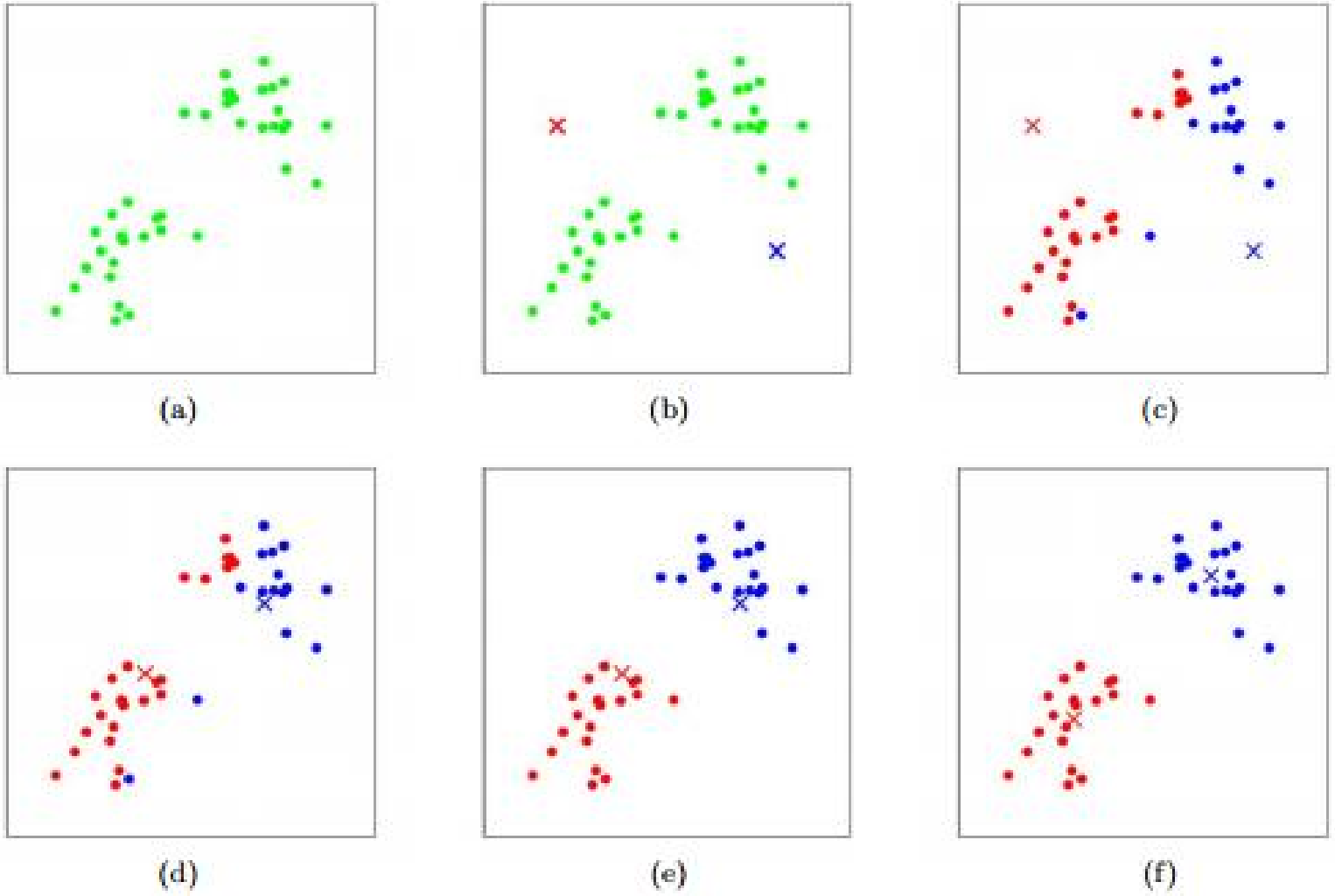
Recalculer les centroïdes des clusters en prenant la moyenne de tous les points de données appartenant à chaque cluster.

JUSQU'À ce qu'il n'y ait aucun changement dans les centroïdes. c'est-à-dire que l'affectation des points de données aux clusters ne change pas.

Fin

Segmentation

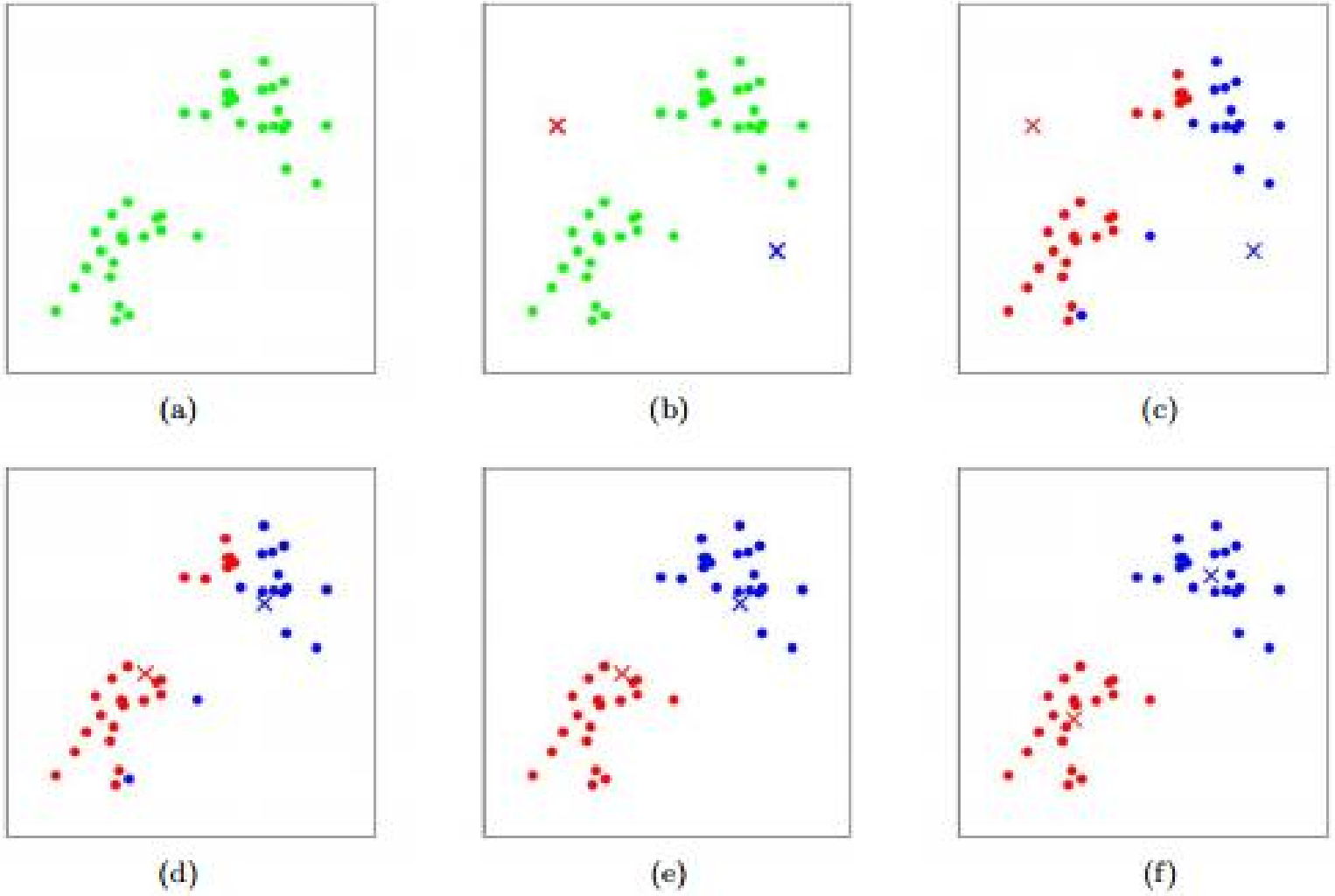
KMeans est un algorithme itératif qui tente de partitionner l'ensemble de données en K sous-groupes (clusters) distincts et prédéfinis qui ne se chevauchent pas, où chaque point de données appartient à un seul groupe .



https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

Segmentation

KMeans est un algorithme itératif qui tente de partitionner l'ensemble de données en K sous-groupes (clusters) distincts et prédéfinis qui ne se chevauchent pas, où chaque point de données appartient à un seul groupe .



Applications :

Segmentation,

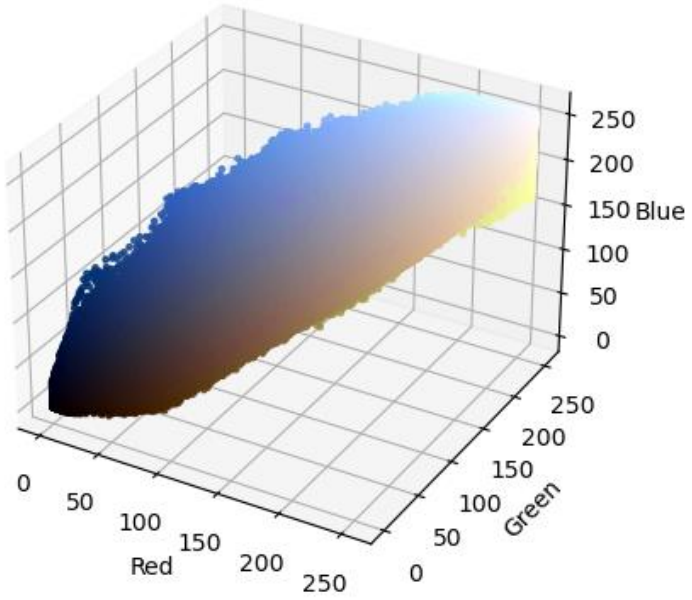
Quantification

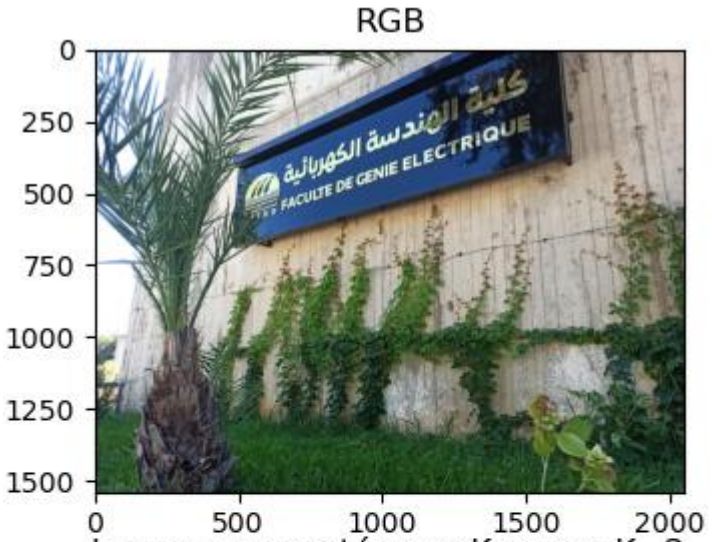
Compression

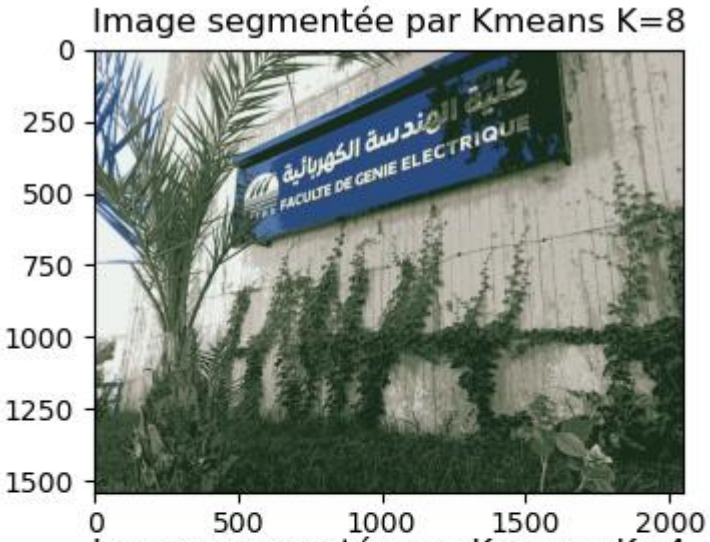
https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

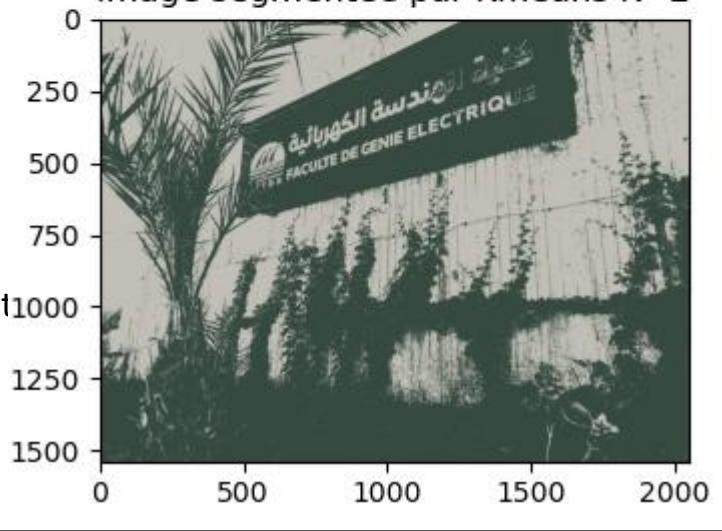
Segmentation

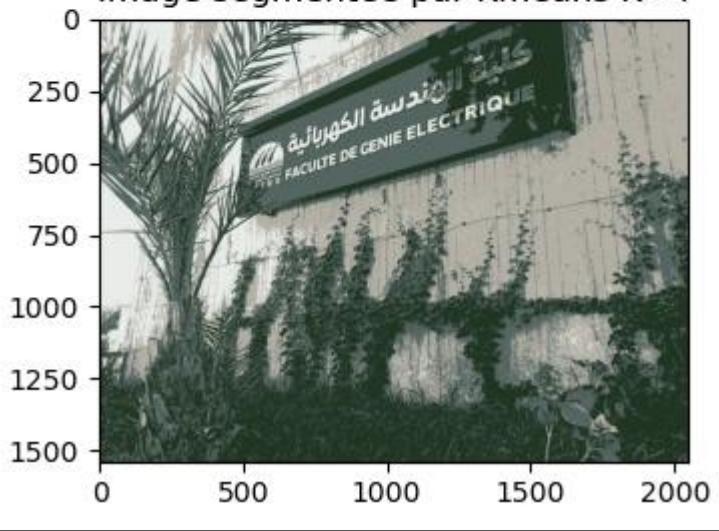
KMeans est un algorithme itératif qui tente de partitionner l'ensemble de données en K sous-groupes (clusters) distincts et prédéfinis qui ne se chevauchent pas, où chaque point de données appartient à un seul groupe **.**











Applications :

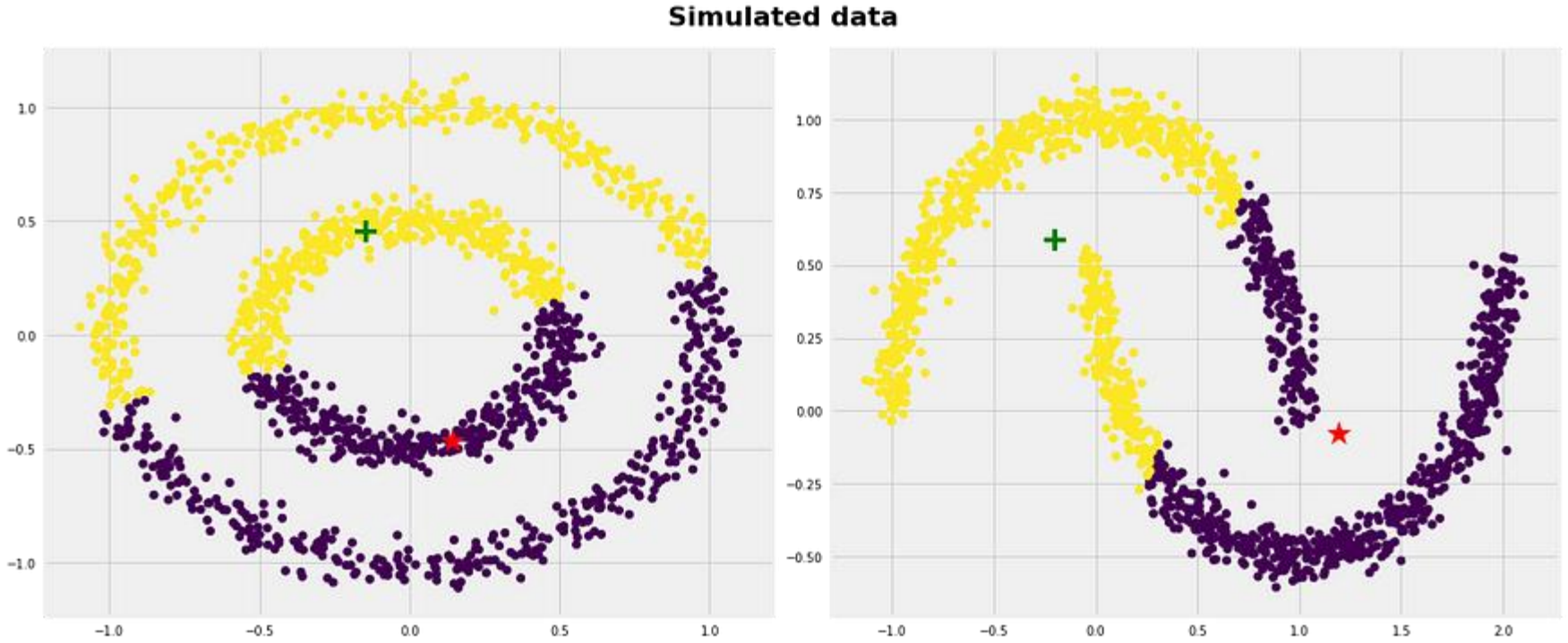
Segmentation, Quantification , Compression

Amélioration des ions : Utilisation d'un noyau, autre dist

Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Segmentation

KMeans est un algorithme itératif qui tente de partitionner l'ensemble de données en K sous-groupes (clusters) distincts et prédéfinis qui ne se chevauchent pas, où chaque point de données appartient à un seul groupe .



https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

Cas d'utilisation

Segmentation de la clientèle basée sur certains critères tels que les habitudes d'achat ou la démographie.

Dans l'exploration de données, le clustering est utilisé lors de l'exploration de données pour identifier des individus similaires.

Segmentation

Mean-Shift est un algorithme non paramétrique pour partitionner des données multidimensionnelles. C'est un algorithme itératif qui vise à faire converger un point vers le maximum local le plus proche.

Il peut être utilisé pour la segmentation, en immergeant l'image dans un espace à 5 dimensions, où chaque pixel est représenté par un point ayant pour coordonnées sa position en x, en y, et ses valeurs R, V, B.

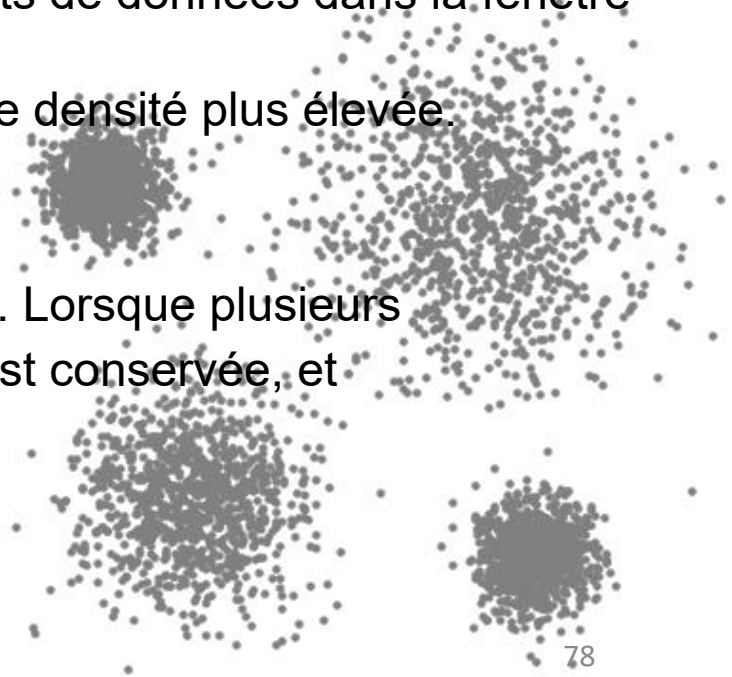
Algorithme :

1. Créer une fenêtre/un cluster coulissant pour chaque point de données dans l'espace des fonctionnalités
2. Chacune des fenêtres glissantes est décalée vers des régions de densité plus élevée en déplaçant leur centre de gravité (centre de la fenêtre glissante) vers la moyenne des points de données dans la fenêtre glissante.
3. Cette étape sera répétée jusqu'à ce qu'aucun changement ne produise une densité plus élevée.

(nombre de points dans la fenêtre glissante)

4. Sélection de fenêtres coulissantes en supprimant les fenêtres superposées. Lorsque plusieurs fenêtres coulissantes se chevauchent, la fenêtre contenant le plus de points est conservée, et les autres sont supprimés.
5. Attribution des points de données à la fenêtre à laquelle ils appartiennent.

https://www.chioka.in/meanshift-algorithm-for-the-rest-of-us-python/



Segmentation

Mean-Shift est un algorithme non paramétrique
algorithme itératif qui vise à faire converger un p

Il peut être utilisé pour la segmentation, en imr
pixel est représenté par un point ayant pour coo

Algorithme :

1. Créer une fenêtre/un cluster coulissant pour
2. Chacune des fenêtres glissantes est décalée
   centre de gravité (centre de la fenêtre glissar
   glissante.
3. Cette étape sera répétée jusqu'à ce qu'aucu

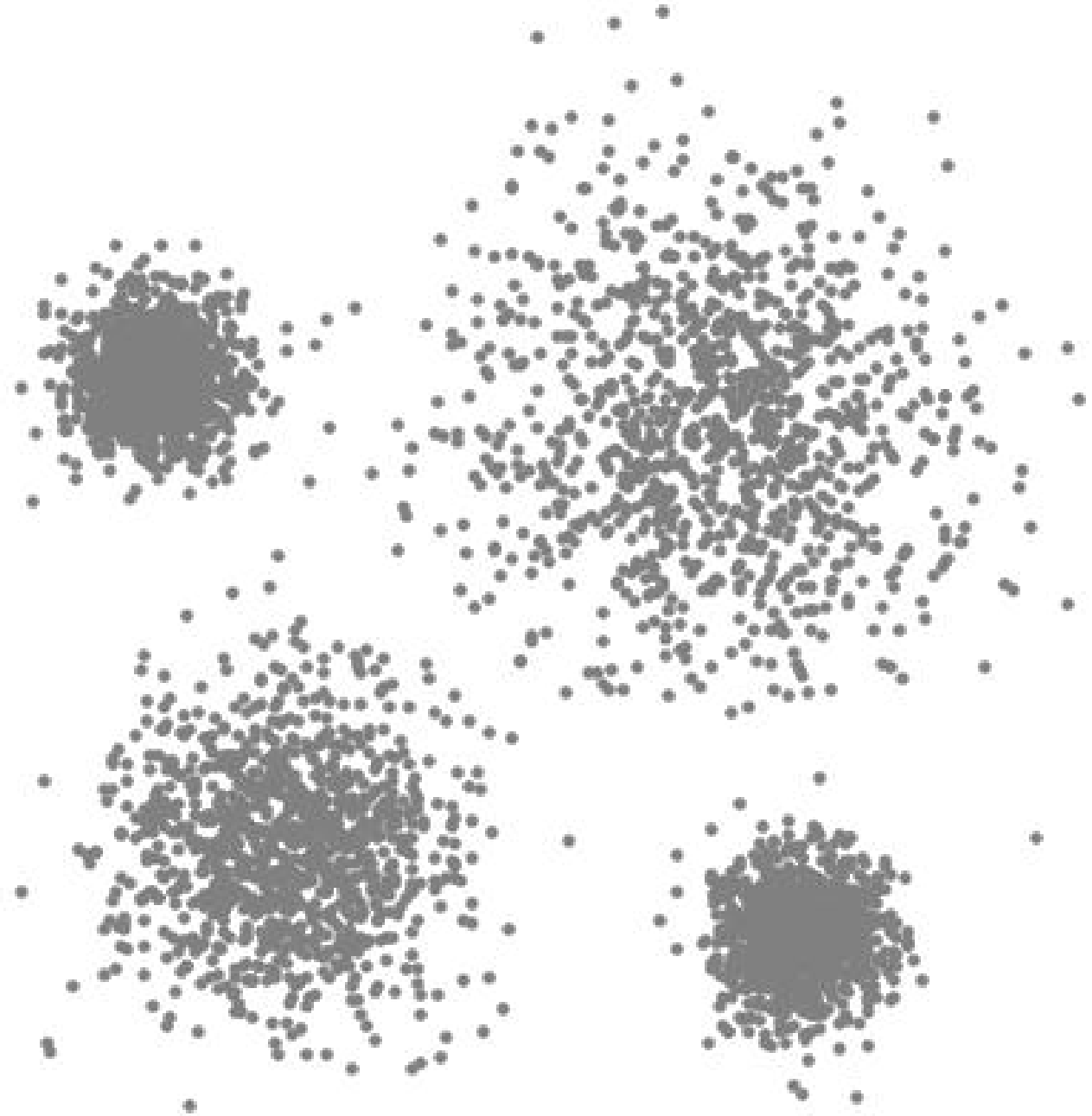
(nombre de points dans la fenêtre glissante)

4. Sélection de fenêtres coulissantes en supprir

les fenêtres coulissantes se chevauchent, la fen
les autres sont supprimés.

5. Attribution des points de données à la fenêtre

https://www.chioka.in/meanshift-algorithm-for-the



https://sbme-tutorials.github.io/2018/cv/notes/8_week8.html

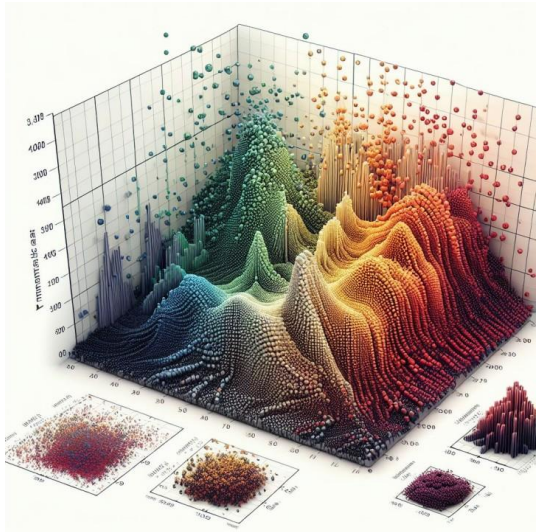
Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Segmentation

Mean Shift déplace les fenêtres vers une région de densité plus élevée en déplaçant leur centroïde (centre de la fenêtre glissante) vers la moyenne des points de données à l'intérieur de la fenêtre glissante. Nous pouvons également examiner cela en considérant nos points de données comme une fonction de densité de probabilité.

Les régions de densité plus élevée correspondent aux régions avec plus de points de données, et les régions de densité plus faible correspondent aux régions avec moins de points. Mean Shift essaie de trouver les régions à haute densité en déplaçant continuellement la fenêtre glissante plus près du sommet de la région. Ceci est également connu sous le nom d'escalade.

La zone la plus dense des données est déterminée par la fonction noyau, qui est une fonction qui attribue des pondérations aux points de données en fonction de leur distance par rapport à la moyenne. La fonction noyau utilisée dans le clustering Mean-Shift est généralement une fonction gaussienne.



Choisir la bonne bande passante/rayon

En fonction de la bande passante, les clusters résultants peuvent être très différents.

Bande passante extrêmement réduite → Chaque point ayant son propre cluster.

Bande passante énorme → un cluster contenant tous les points de données.

Segmentation

Mean Shift déplace les fenêtres vers u
de la fenêtre glissante) vers la moyen
pouvons également examiner cela en c
probabilité.

Les régions de densité plus élevée c
régions de densité plus faible correspor
les régions à haute densité en déplaç
région. Ceci est également connu sous

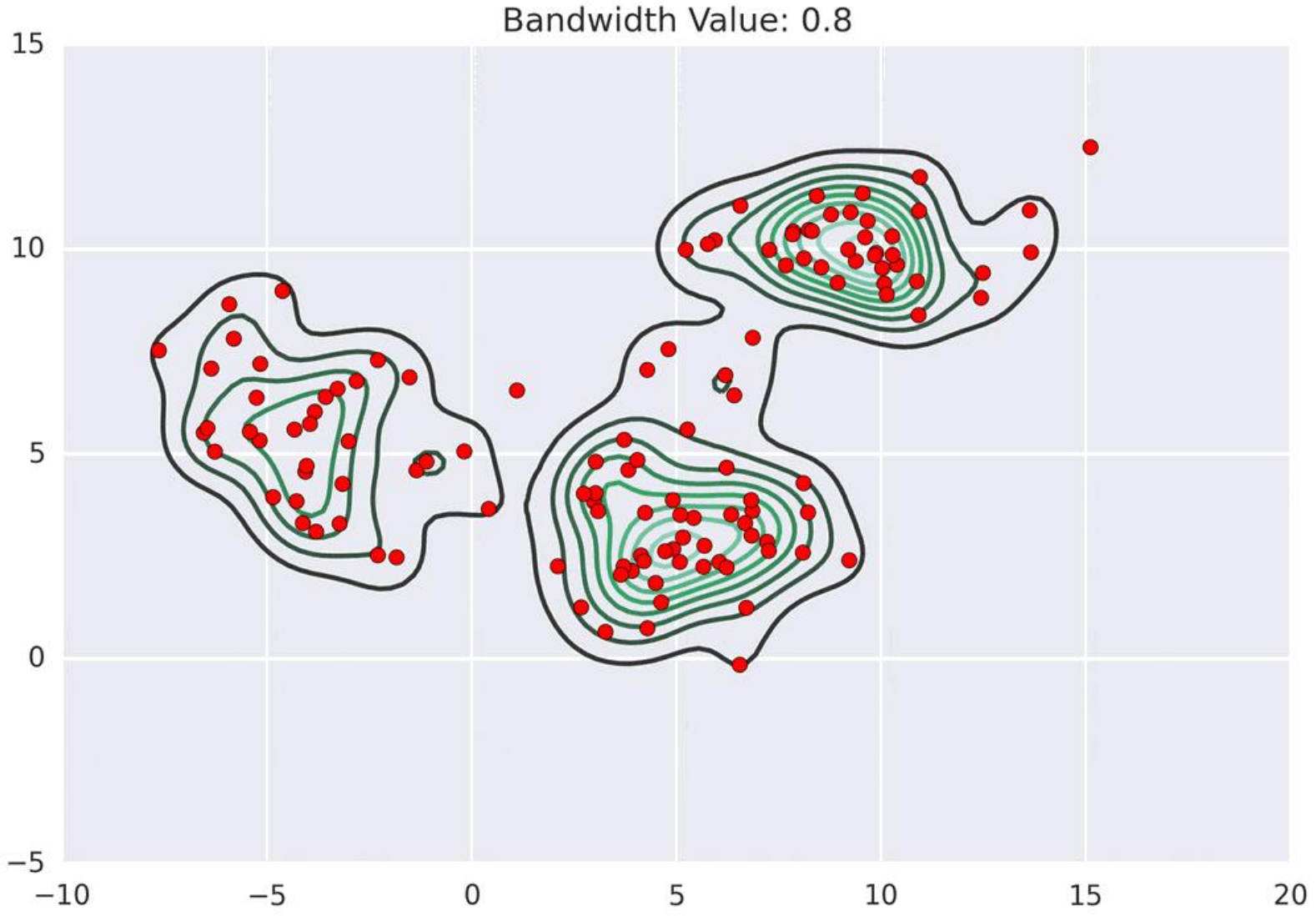
La zone la plus dense des données es
des pondérations aux points de donné
noyau utilisée dans le clustering Mean-

Choisir la bonne bande passante/ray

En fonction de la bande passante, les c

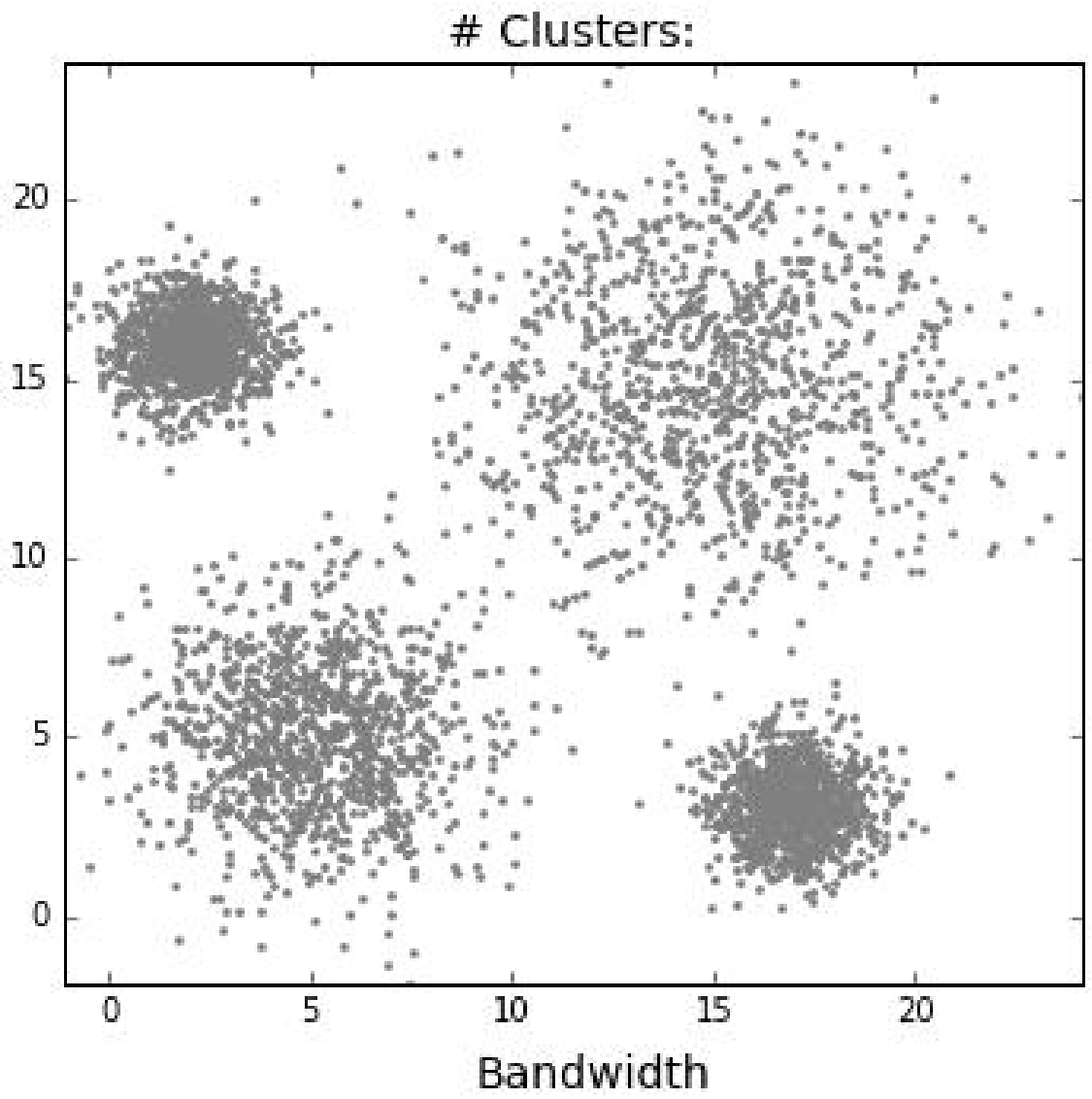
Bande passante extrêmement réduite -

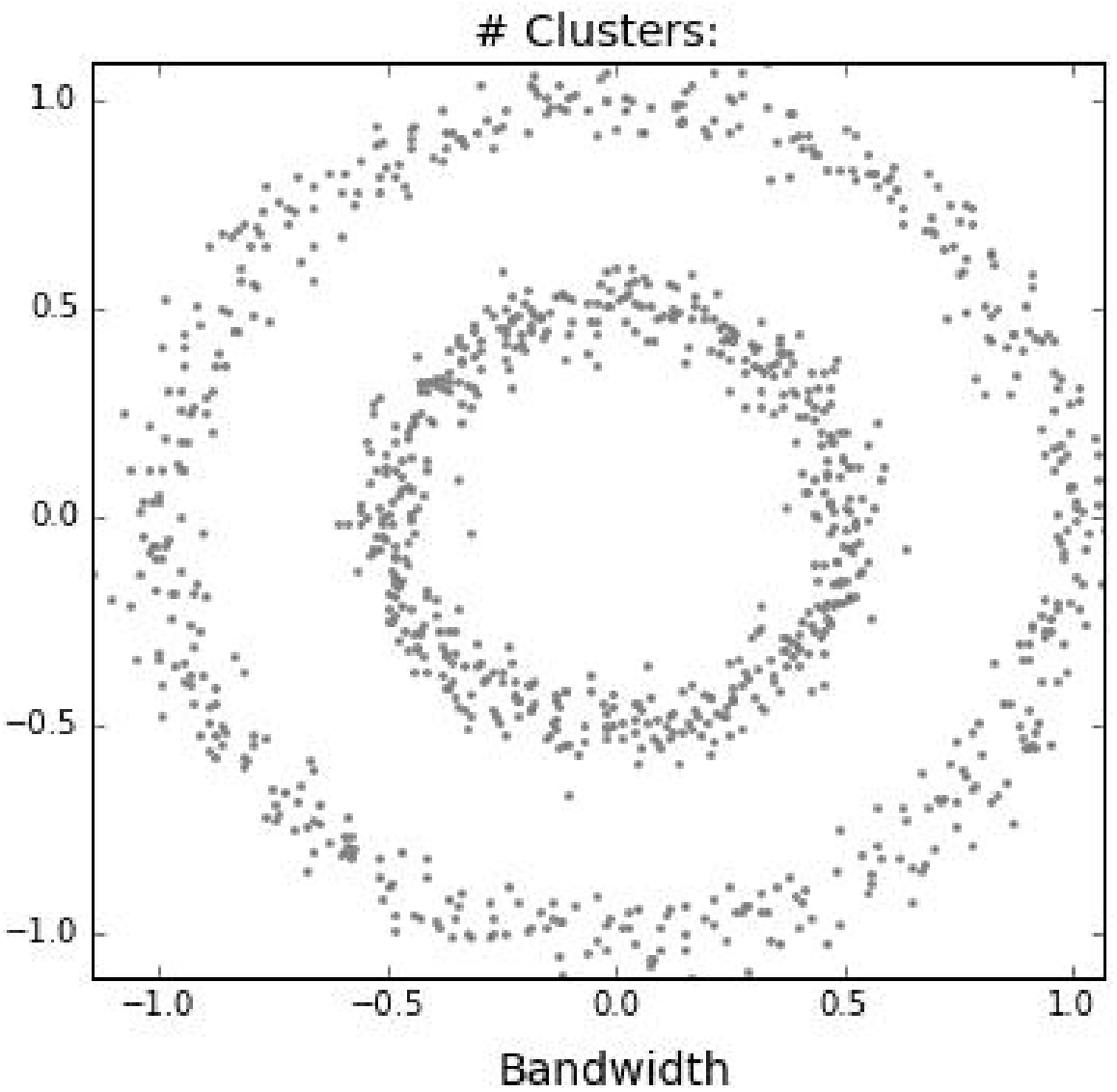
Bande passante énorme → un cluster c



https://spin.atomicobject.com/mean-shift-clustering/

Segmentation





Segmentation

Mean-Shift est un algorithme non paramétrique pour partitionner des données multidimensionnelles. C'est un algorithme itératif qui vise à faire converger un point vers le maximum local le plus proche.

Il peut être utilisé pour la segmentation, en immergeant l'image dans un espace à 5 dimensions, où chaque pixel est représenté par un point ayant pour coordonnées sa position en x, en y, et ses valeurs R, V, B.

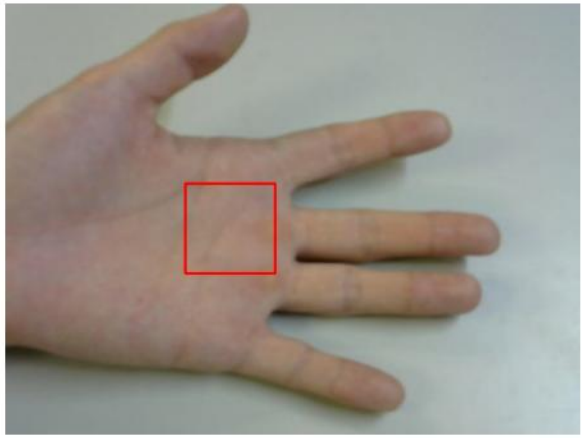
Segmentation à l'aide de la couleur

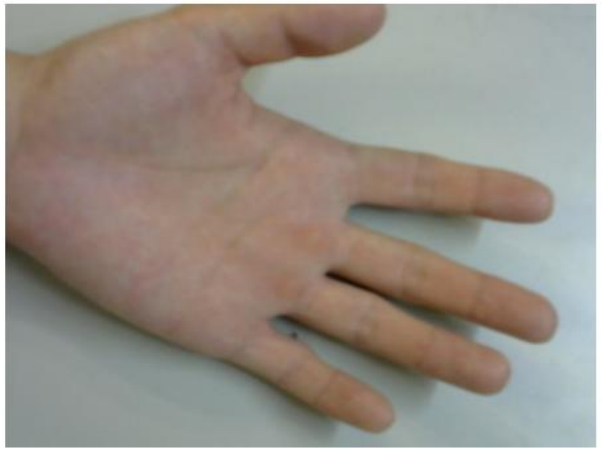
Le suivi d'objets avec un algorithme Mean-Shift peut être divisé en trois étapes :

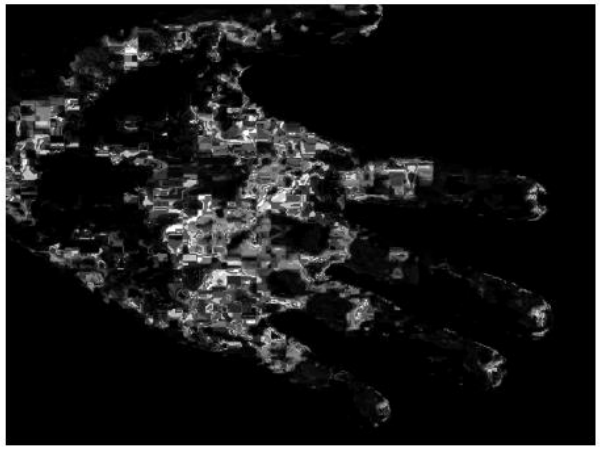
Le modèle cible est représenté par un histogramme de couleurs. Lorsque l'objet bouge, ce mouvement sera reflété dans l'histogramme.

Créer une carte de confiance dans la nouvelle image basée sur l'histogramme des couleurs de l'objet dans l'image précédente et utiliser le décalage moyen pour trouver le pic d'une carte de confiance près de l'ancienne position de l'objet.

La carte de confiance est une fonction de densité de probabilité sur la nouvelle image, attribuant à chaque pixel de la nouvelle image une probabilité, qui est la probabilité que la couleur du pixel apparaisse dans l'objet de l'image précédente.







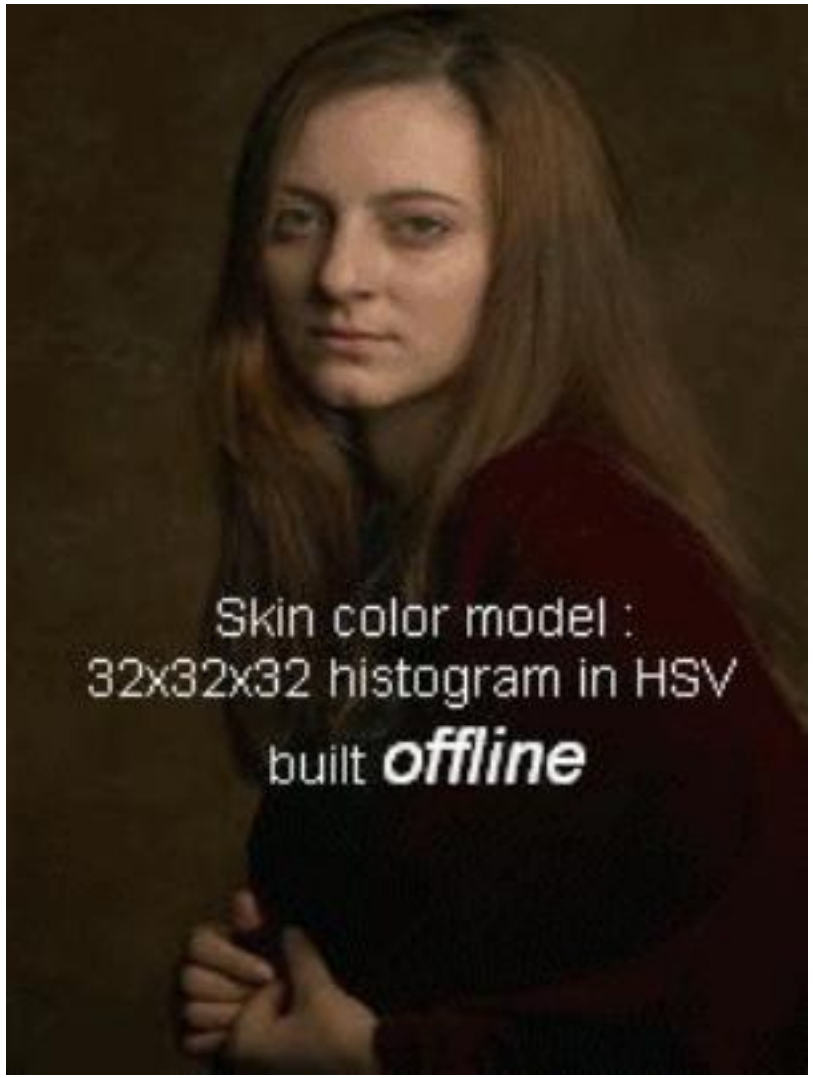
Segmentation à l'aide dela couleur

Le suivi d'objets avec un algorithme Mean-Shift peut être divisé en trois étapes :

Le modèle cible est représenté par un histogramme de couleurs. Lorsque l'objet bouge, ce mouvement sera reflété dans l'histogramme.

Créer une carte de confiance dans la nouvelle image basée sur l'histogramme des couleurs de l'objet dans l'image précédente et utiliser le décalage moyen pour trouver le pic d'une carte de confiance près de l'ancienne position de l'objet.

La carte de confiance est une fonction de densité de probabilité sur la nouvelle image, attribuant à chaque pixel de la nouvelle image une probabilité, qui est la probabilité que la couleur du pixel apparaisse dans l'objet de l'image précédente.



Segmentation à l'aide de la couleur

Le suivi d'objets avec un algorithme Mean-Shift peut être divisé en trois étapes :

- Cibler l'objet : Choisissez dans la première image l'emplacement initial de l'objet à suivre.

Le modèle cible est représenté par un histogramme de couleurs. Lorsque l'objet bouge, ce mouvement sera reflété dans l'histogramme.

- Recherche du nouvel emplacement : Dans l'image suivante, l'algorithme Mean-Shift déplace la fenêtre vers le nouvel emplacement avec une densité de pixels maximale.

L'histogramme actuel est utilisé pour rechercher le meilleur candidat de correspondance cible en maximisant la fonction de similarité.

-Mise à jour de la localisation : L'histogramme et la localisation de l'objet cible. sont mis à jour

Camshift (Continuously Adaptive Mean Shift) : Une fois le décalage moyen converge, l'algorithme Camshift met à jour la taille de la fenêtre de sorte que la fenêtre de suivi puisse changer de taille ou même pivoter pour mieux correspondre aux mouvements de l'objet suivi.

Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Segmentation à l'aide de la couleu

Le suivi d'objets avec un algorithme **Cam -Shift** peut être divisé en trois étapes :

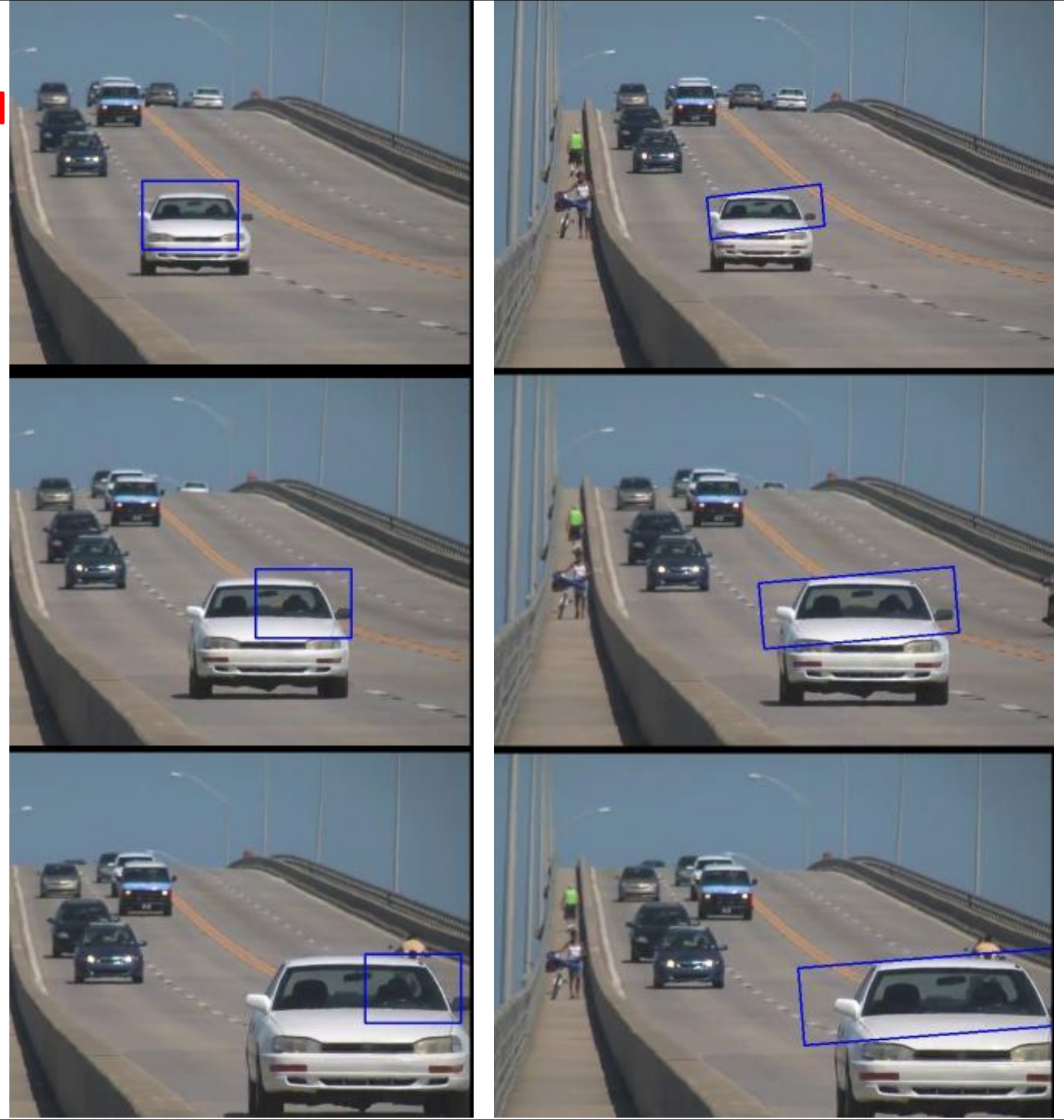
- Cibler l'objet : Choisissez dans la première image l'emplacement initial de l'objet à suivre.

Le modèle cible est représenté par un histogramme de couleurs. Lorsque l'objet bouge, ce mouvement sera reflété dans l'histogramme.

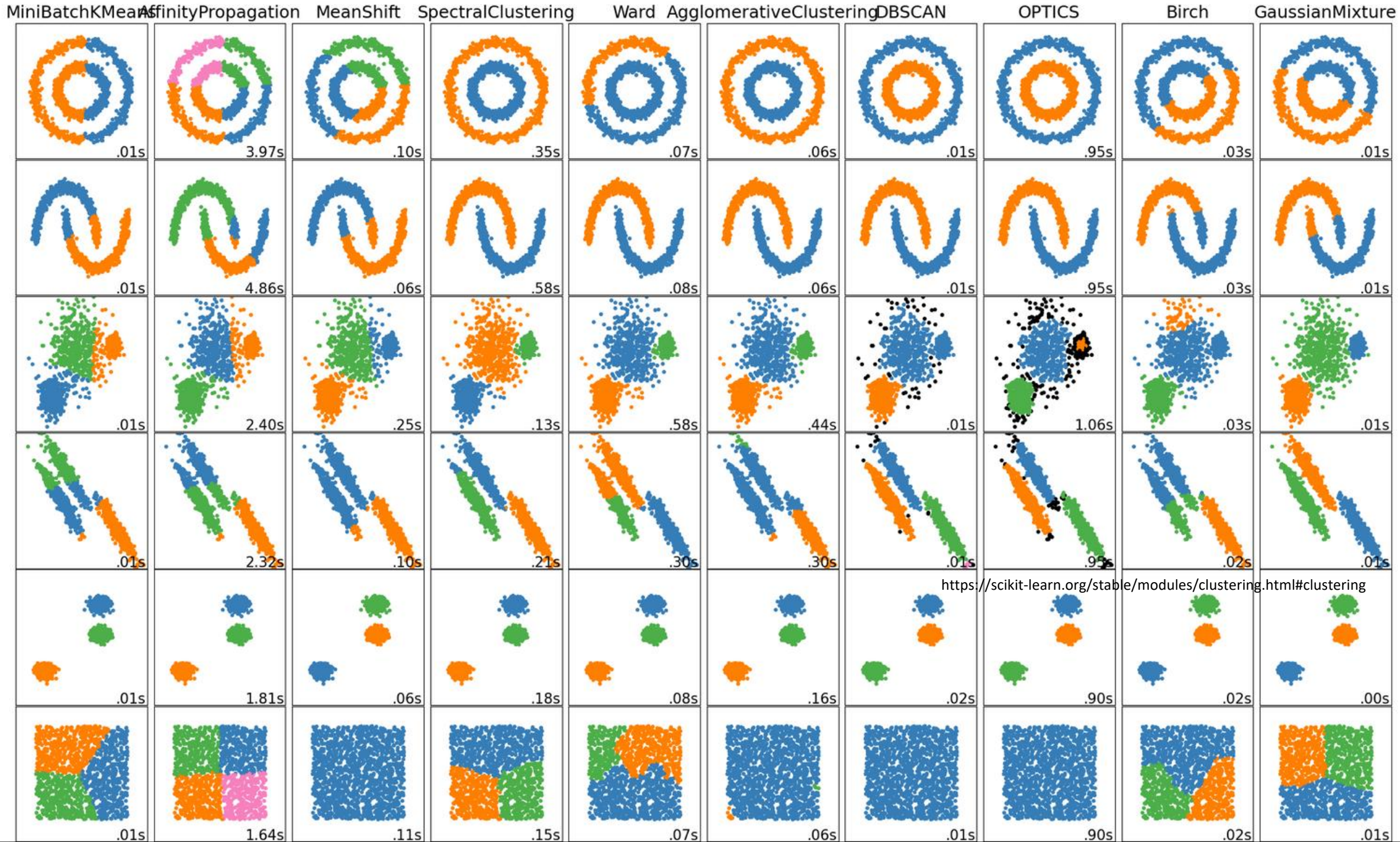
- Recherche du nouvel emplacement : Dans l'image suivante, l'algorithme Mean-Shift déplace la fenêtre vers le nouvel emplacement avec une densité de pixels maximale.

L'histogramme actuel est utilisé pour rechercher le meilleur candidat de correspondance cible en maximisant la fonction de similarité.

-Mise à jour de la localisation : L'histogramme et la localisation de l'objet cible sont mis à jour



Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

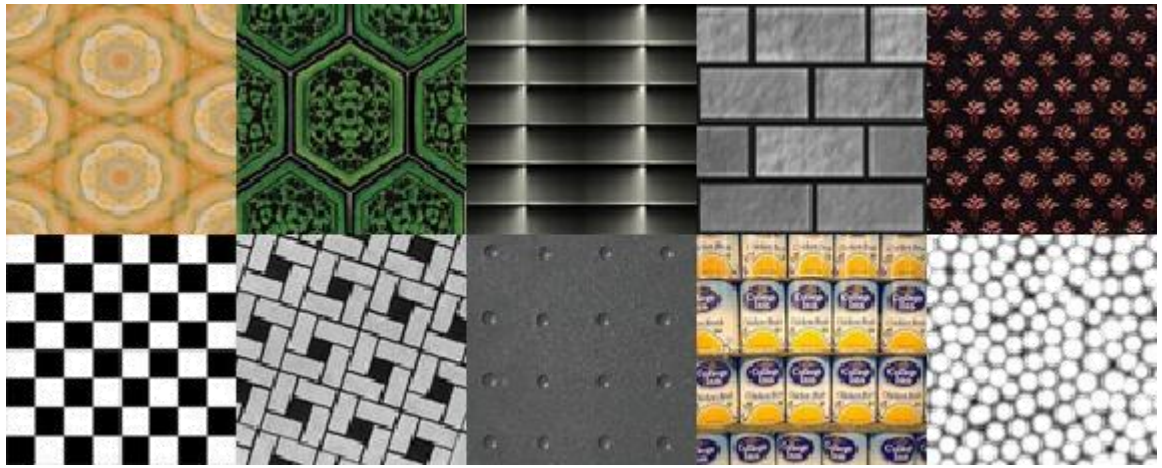


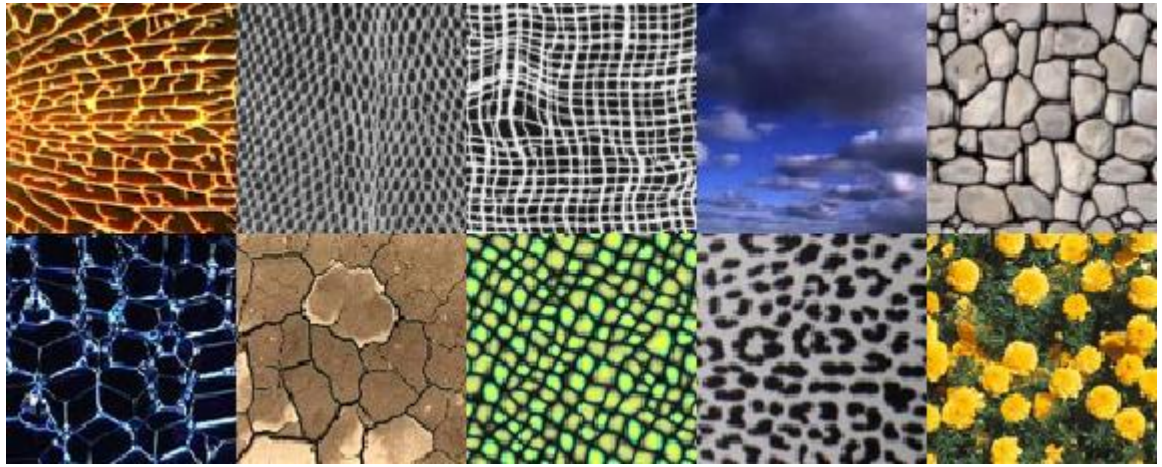
https://scikit-learn.org/stable/modules/clustering.html#clustering

Descripteurs de texture

La texture constitue une riche source d'informations sur la scène naturelle. Elle peut être définie en fonction de la variation spatiale des couleurs (intensité) des pixels dans un voisinage.

La surface de tout objet visible est texturée à une certaine échelle.





http://perso.usthb.dz/~akourgli/

https://www.cs.auckland.ac.nz/~georgy/research/texture/thesis-html/node5.html

Descripteurs de texture

La texture constitue une riche source d'informations sur la scène naturelle. Elle peut être définie en fonction de la variation **spatiale** des couleurs (intensité) des pixels dans **un voisinage.**

La surface de tout objet visible est texturée à une certaine échelle.

Descripteurs de texture

Pour les designers, une texture ajoute de la richesse à un design.

À mesure que la technologie progresse, les capacités de texturation évoluent également pour créer les personnages, les créatures et les paysages des mêmes jeux vidéo que nous connaissons et aimons tous.

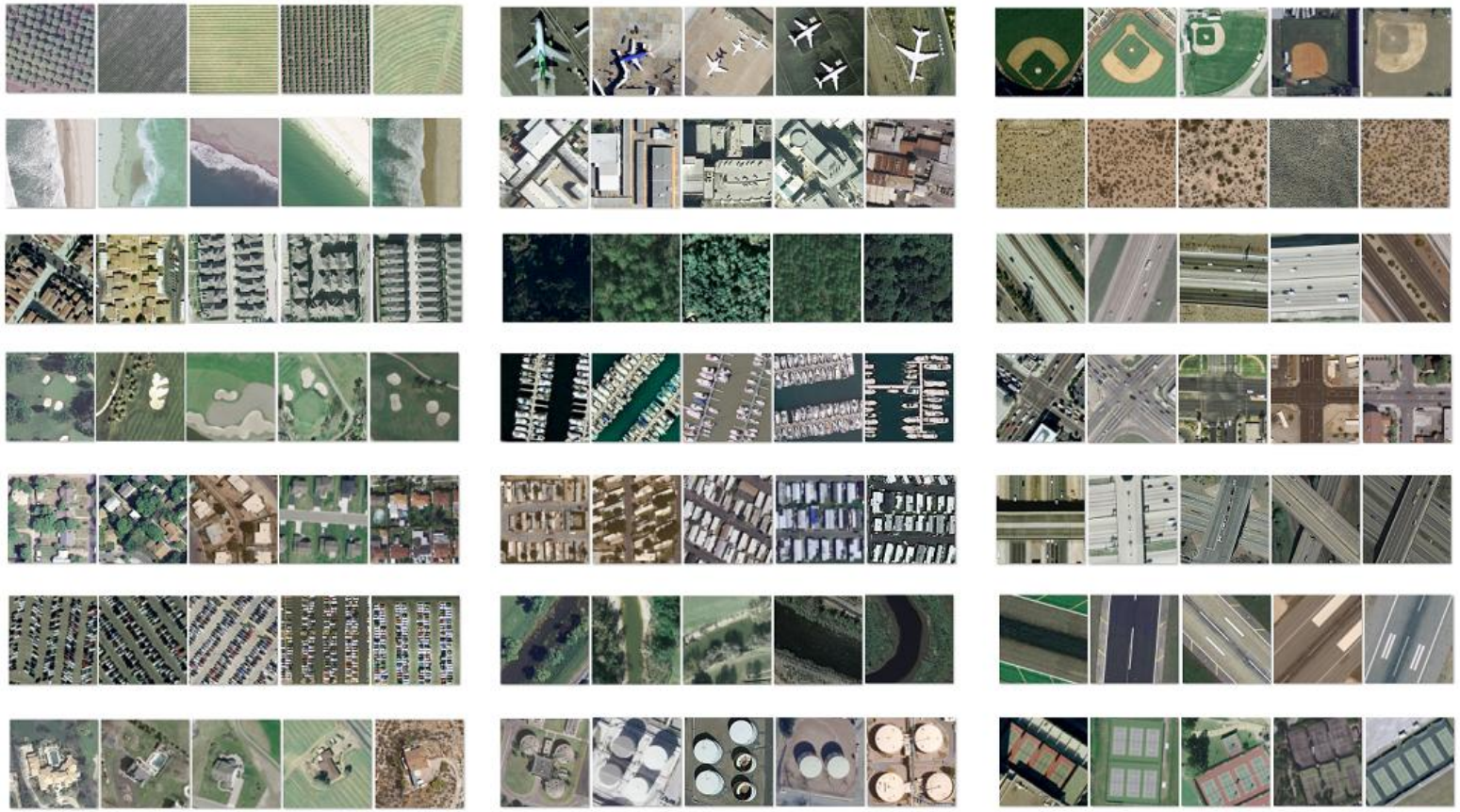
Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Descripteurs de texture

Un descripteur de texture nous donne des informations sur la disposition spatiale des couleurs ou des intensités dans une image texturée ou une région sélectionnée d'une image.

Application de l'analyse de texture

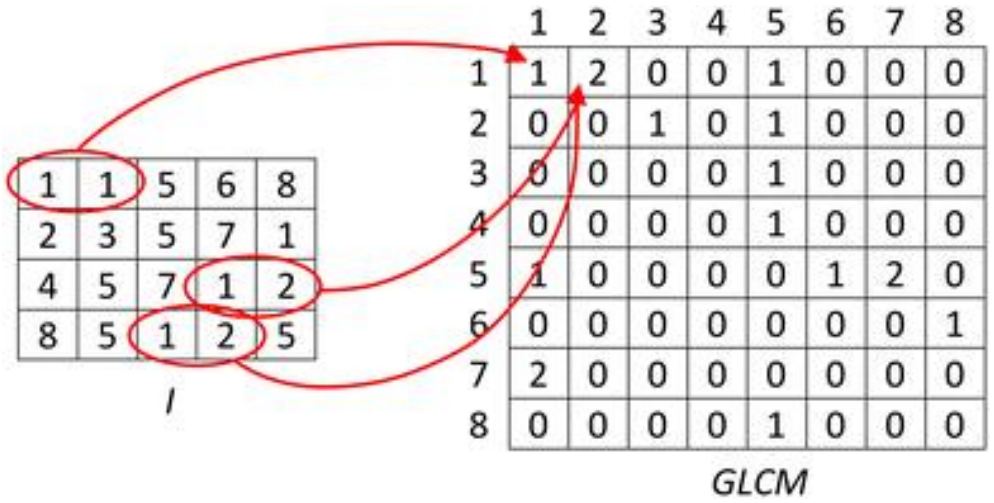
- ✓ Détection faciale
- ✓ Suivi d'objets dans les vidéos
- ✓ Diagnostic de la qualité des produits
- ✓ Analyse d'images médicales
- ✓ Télédétection
- ✓ Végétation



Descripteurs de texture

Matrice de cooccurrence de niveaux de gris (GLCM) : distribution conjointe

recherche des paires de valeurs de pixels distants de d suivant un angle ϕ qui apparaissent dans une image

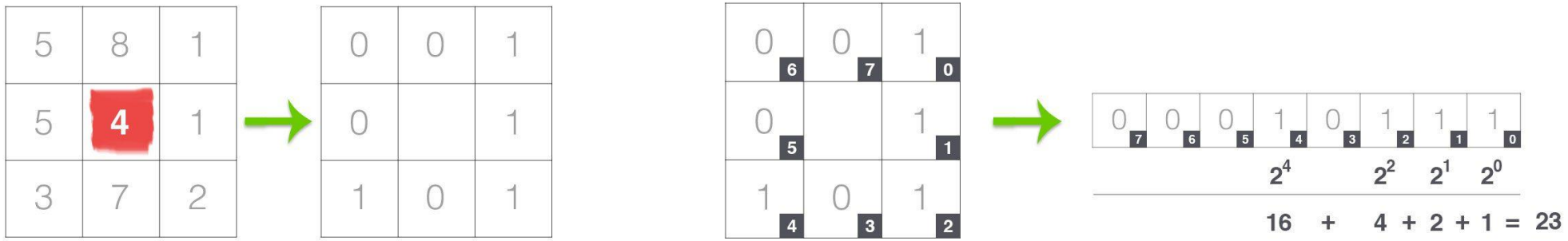


1	contrast	$f_1 = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \lvert i-j \rvert^2 P_{ij}$
2	entropy	$f_2 = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} P_{ij} \log_2 P_{ij}$
3	energy	$f_5 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} P_{ij}^2$
4	homogeneity	$f_3 = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \frac{P_{ij}}{1+\lvert i-j \rvert^2}$, k>=1

5	correlation	$f_4 = \frac{1}{\sigma_x \sigma_y}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}(i \cdot j)P_{ij} - \mu_x \mu_y$, $\mu_x = \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} P_{ij}$, $\mu_y = \sum_{j=0}^{L-1} j \sum_{i=0}^{L-1} P_{ij}$, $\sigma_x^2 = \sum_{i=0}^{L-1}(i-\mu_x)^2 \sum_{j=0}^{L-1} P_{ij}$, $\sigma_y^2 = \sum_{j=0}^{L-1}(j-\mu_y)^2 \sum_{i=0}^{L-1} P_{ij}$
6	cluster shade	$f_6 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}[(i-\mu_x)+(j-\mu_y)]^3 P_{ij}$
7	cluster prominence	$f_7 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}[(i-\mu_x)+(j-\mu_y)]^4 P_{ij}$

Descripteurs de texture

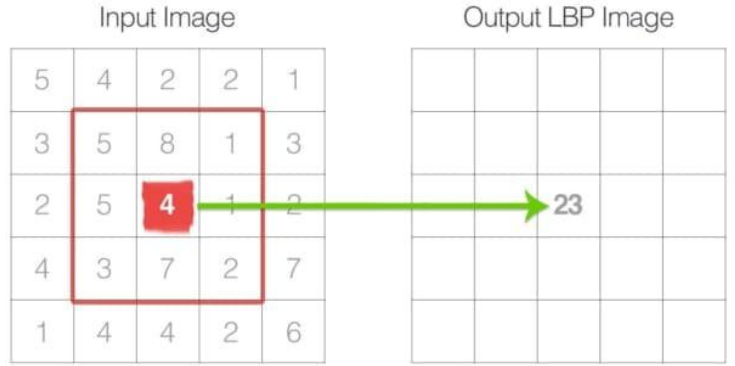
LBP est une mesure de descripteur de texture simple et invariante en niveaux de gris pour la classification. Dans LBP, un code binaire est généré pour chaque pixel en seuillant ses pixels de voisinage à 0 ou 1 en fonction de la valeur du pixel central .



https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/

Collectez les valeurs de seuil dans le sens des aiguilles d'une montre ou dans le sens inverse des aiguilles d'une montre.

Ensuite, convertissez le code binaire en décimal et placez-le au centre de la matrice.

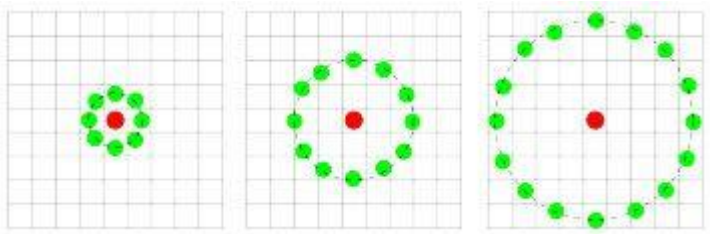


Faculté de Génie Electrique, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

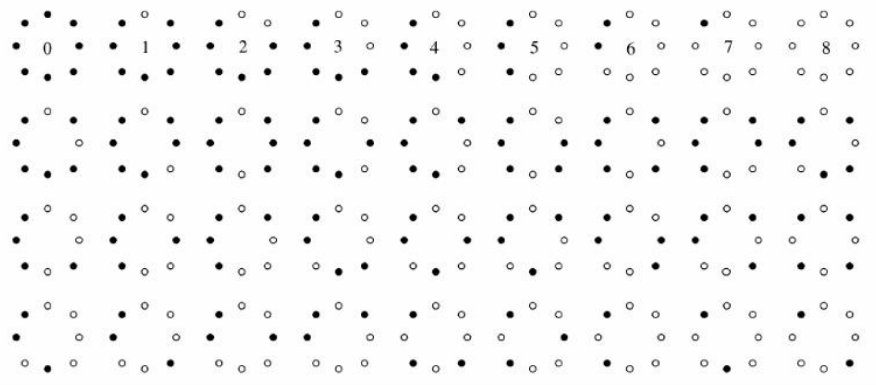
Descripteurs de texture

Extension LBP

- Le nombre de points p dans un voisinage
- Le rayon du cercle r, qui permet de prendre en compte différentes échelle



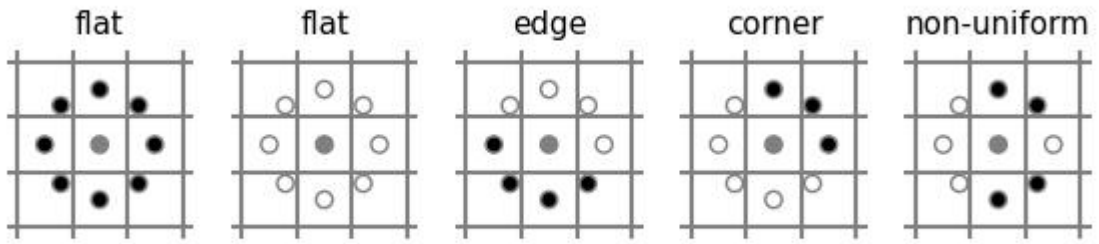
https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/



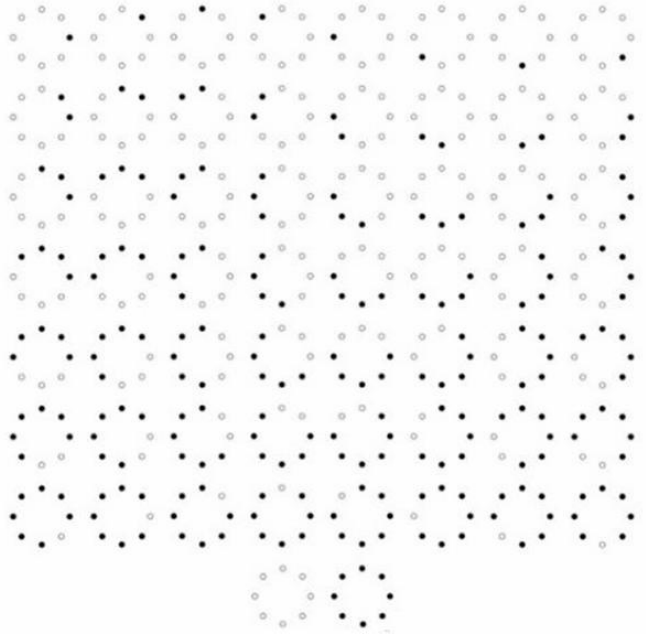
- Invariant de rotation R=8 → 36 codes
- uniforme : Une LBP est considérée comme uniforme

s'il elle comporte au plus deux transitions 0-1 ou 1-0 → 59 codes

- RI Uniformité LBP → 10 codes



https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_local_binary_pattern.html



Descripteurs

skimage.feature.blob_dog	Recherche des blobs dans l'image en niveaux de gris donnée.
skimage.feature.blob_doh	Recherche des blobs dans l'image en niveaux de gris donnée.
skimage.feature.blob_log	Recherche des blobs dans l'image en niveaux de gris donnée.
skimage.feature.canny	Edge filtre une image à l'aide de l'algorithme Canny.
skimage.feature.corner_fast	Extrayez les coins FAST pour une image donnée.
skimage.feature.corner_foerstner	Calculez l'image de réponse de la mesure du coin Foerstner .
skimage.feature.corner_harris	Calculez l'image de réponse de la mesure d'angle Harris.
skimage.feature.corner_kitchen_rosenfeld	Compute Kitchen et le coin Rosenfeld mesurent l'image de réponse.
skimage.feature.corner_moravec	Calculez l'image de réponse de la mesure d'angle de Moravec.
skimage.feature.corner_orientations	Calculez l'orientation des coins.

skimage.feature.corner_peaks	Recherchez les pics dans l'image de réponse de mesure de coin.
skimage.feature.corner_shi_tomasi	Calculez l'image de réponse de mesure de coin Shi-Tomasi (Kanade-Tomasi).
skimage.feature.corner_subpix	Déterminez la position des sous-pixels des coins.
skimage.feature.daisy	Extrayez de manière dense les descripteurs de fonctionnalités DAISY pour l'image donnée.
skimage.feature.draw_haar_like_feature	Visualisation des caractéristiques de type Haar.
skimage.feature.draw_multiblock_lbp	Visualisation de modèles binaires locaux multiblocs.
skimage.feature.fisher_vector	Calculez le vecteur de Fisher à partir de certains descripteurs/vecteurs et d'un GMM estimé associé.
skimage.feature.graycomatrix	Calculez la matrice de cooccurrence de niveau de gris.
skimage.feature.graycoprops	Calculez les propriétés de texture d'un GLCM.
skimage.feature.haar_like_feature	Calculez les caractéristiques de type Haar pour une région d'intérêt (ROI) d'une image intégrale.

https://scikit-image.org/docs/stable/api/skimage.feature.html#

https://scikit-image.org/docs/dev/auto_examples/

Descripteurs

skimage.feature.haar_like_feature_coord	Calculez les coordonnées des entités de type Haar.
skimage.feature.hessian_matrix	Calculez la matrice de Hesse.
skimage.feature.hessian_matrix_det	Calculez le déterminant hessien approximatif sur une image.
skimage.feature.hessian_matrix_eigvals	Calculez les valeurs propres de la matrice hessienne.
skimage.feature.hog	Extraire l'histogramme des dégradés orientés (HOG) pour une image donnée.
skimage.feature.learn_gmm	Estimez un modèle de mélange gaussien (GMM) à partir d'un ensemble de descripteurs et d'un nombre de modes (c'est-à-dire des Gaussiennes).
skimage.feature.local_binary_pattern	Calculez les modèles binaires locaux (LBP) d'une image.
skimage.feature.match_descriptors	Correspondance par force brute des descripteurs.
skimage.feature.match_template	Faites correspondre un modèle à une image 2D ou 3D à l'aide d'une corrélation normalisée.
skimage.feature.multiblock_lbp	Modèle binaire local multibloc (MB-LBP).
skimage.feature.multiscale_basic_features	Fonctionnalités locales pour une image nd mono ou multicanal.
skimage.feature.peak_local_max	Recherchez les pics dans une image sous forme de liste de coordonnées.
skimage.feature.plot_matches	Tracez les fonctionnalités correspondantes.
skimage.feature.shape_index	Calculez l'indice de forme.
skimage.feature.structure_tensor	Calculez le tenseur de structure en utilisant la somme des carrés des différences.
skimage.feature.structure_tensor_eigenvalues	Calculer les valeurs propres du tenseur de structure.
skimage.feature.BREF	BREF extracteur de descripteur binaire.
skimage.feature.CENSURE	Détecteur de points clés CENSURE.
skimage.feature.Cascade	Classe pour la cascade de classificateurs utilisée pour la détection d'objets.
skimage.feature.ORB	Détecteur de caractéristiques orienté RAPIDEMENT et tourné BRIEF et extracteur de descripteur binaire.
skimage.feature.SIFT	Détection de fonctionnalités SIFT et extraction de descripteurs.

https://scikit-image.org/docs/stable/api/skimage.feature.html#

Descripteurs de forme

Les caractéristiques de forme efficaces doivent présenter certaines propriétés essentielles telles que :

- ✓ Identifiabilité : les formes qui sont perçues comme similaires par l'homme ont la même caractéristique différente des autres.
- ✓ Invariance de translation, de rotation et d'échelle : l'emplacement, la rotation et le changement d'échelle de la forme ne doivent pas affecter les caractéristiques extraites
- ✓ Invariance affine : la transformation affine effectue un mappage linéaire des coordonnées 2D vers d'autres coordonnées 2D qui préserve la « rectitude » et le « parallélisme » des lignes.
- ✓ Résistance au bruit : les caractéristiques doivent être aussi robustes que possible au bruit, c'est-à-dire qu'elles doivent être les mêmes quelle que soit la force du bruit dans une plage donnée qui affecte le motif.

https://machinelearningmastery.com/k-nearest-neighbours-classification-using-opencv/
https://www.kaggle.com/code/olaniyan/image-classification-using-knn
https://github.com/poojasrini/Image-classification-using-KNN/blob/main/Image_Classification.ipynb
https://medium.com/swlh/image-classification-with-k-nearest-neighbours-51b3a289280

Descripteurs de forme

HOG : Analyse la distribution des orientations des contours au sein d'un objet pour décrire sa forme et son apparence.

- Tout d'abord, l'image du gradient dans les directions x et y est calculée.
- La fenêtre d'image est divisée en petites régions spatiales appelées « cellules ». Pour chaque cellule, un histogramme 1D local des orientations du gradient est accumulé sur tous les pixels de cette cellule.

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Changement dans la direction X (G_x) = 89 – 78 = 11

Changement dans la direction Y (G_y) = 68 – 56 = 8

Magnitude totale du gradient = $\sqrt{[(11)^2+(8)^2]}$ = 13.6

Φ = atan (Gy / Gx) = 36°

Magnitude		1							
Bin	0	20	40	60	80	100	120	140	160

		13.6							
Bin	0	20	40	60	80	100	120	140	160

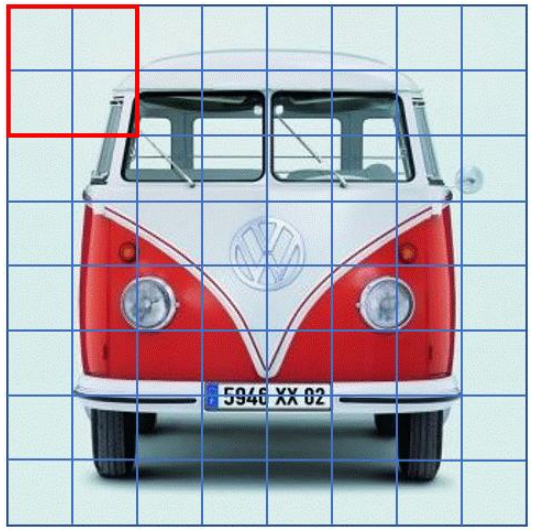
https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

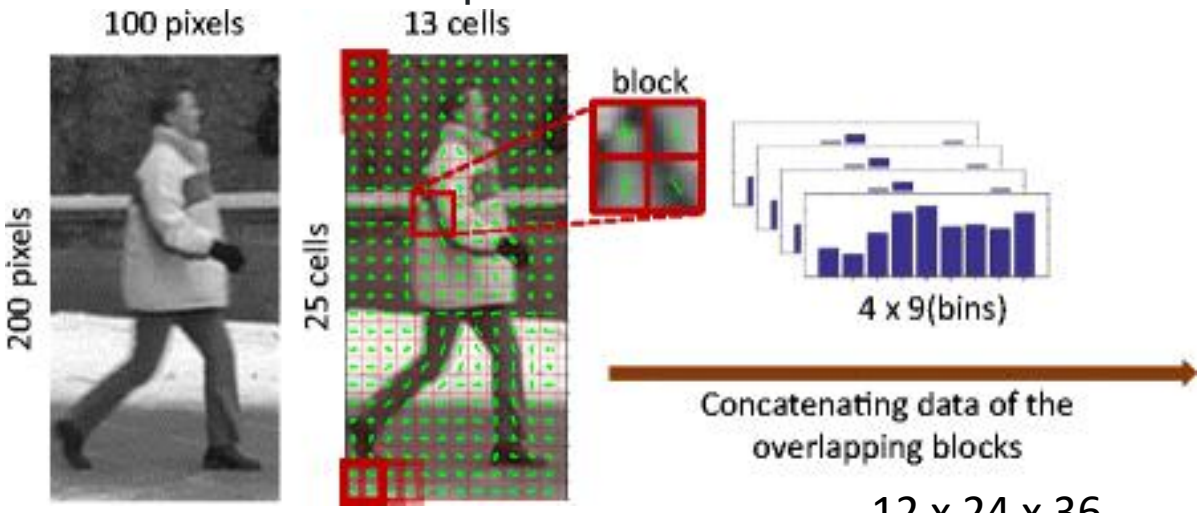
- La normalisation est effectuée sur des groupes locaux de cellules (appelés « blocs ») pour améliorer l'invariance à l'éclairage, aux ombres et au contraste des bords.

Shape descriptors

HOG : Analyse la distribution des orientations des contours au sein d'un objet pour décrire sa forme et son apparence.

- Tout d'abord, l'image du gradient dans les directions x et y est calculée.
- La fenêtre d'image est divisée en petites régions spatiales appelées « cellules ». Pour chaque cellule, un histogramme 1D local des orientations du gradient est accumulé sur tous les pixels de cette cellule.





https://images.app.goo.gl/LNBNcPi4eJ9LHLaF8

- La normalisation est effectuée sur des groupes locaux de cellules (appelés « blocs ») pour améliorer l'invariance à l'éclairage, aux ombres et au contraste des bords.

HOG Size = the total number of blocks x the number of cells per block x the number of orientation bins.

Shape descriptors

HOG : Analyse la distribution des orientations des contours au sein d'un objet pour décrire sa forme et son apparence

Image originale

Faculty of Electrical Engineering, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

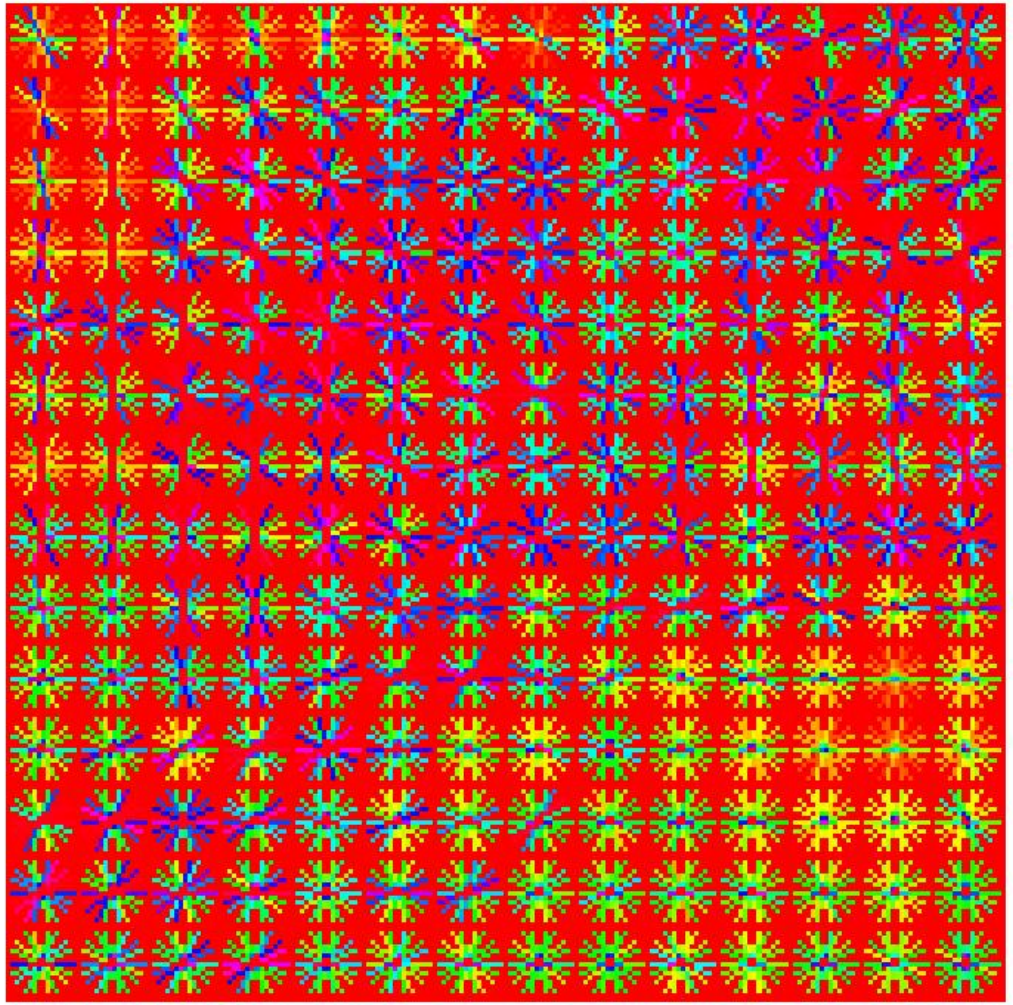
Shape descriptors

HOG : Analyse la distribution des orientations des contours au sein d'un objet pour décrire sa forme et son apparence

Shape descriptors

HOG : Analyse la distribution des orientations des contours au sein d'un objet pour décrire sa forme et son apparence

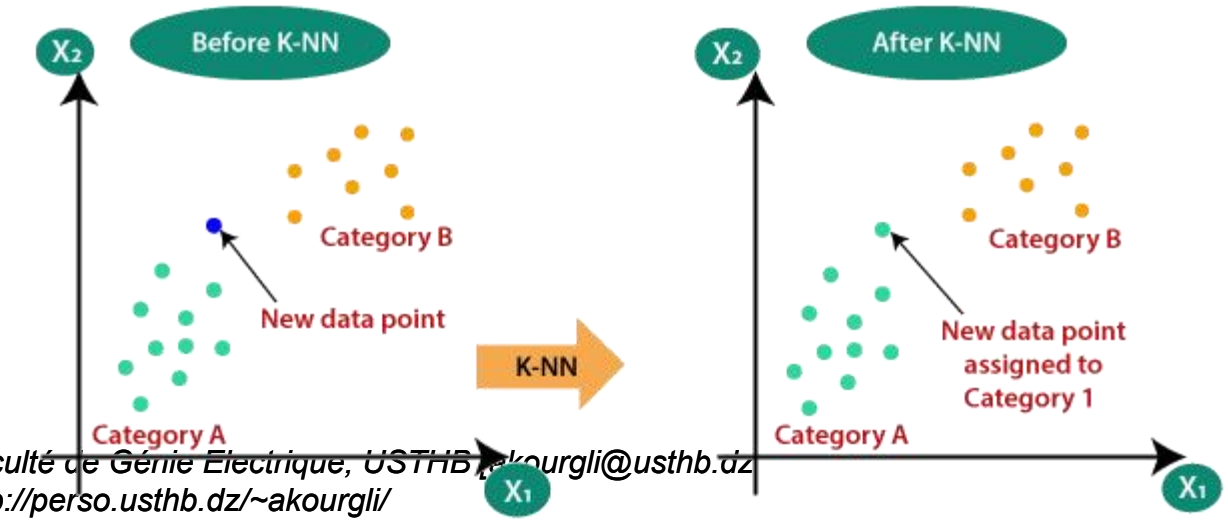
Image originale

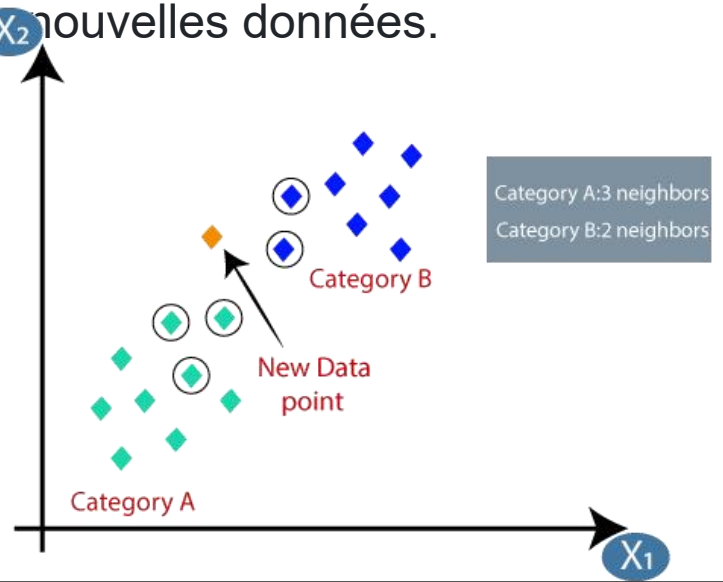


Descripteurs → Classification : exemple K-NN

K-Nearest Neighbour est l'un des algorithmes d'apprentissage automatique **non paramétriques les plus simples** basés sur la technique d'apprentissage supervisé.

- Il stocke toutes les données disponibles et classe un nouveau point de données en fonction de la similarité. Cela signifie que lorsque de nouvelles données apparaissent, elles peuvent être facilement classées dans une catégorie en utilisant l'algorithme K-NN.
- L'algorithme K-NN peut être utilisé pour la régression ainsi que pour la classification, mais il est principalement utilisé pour les problèmes de classification.
- On l'appelle également algorithme d'apprentissage paresseux car il **n'apprend pas** de l'ensemble de formation. Lors de la phase de formation, il stocke simplement l'ensemble de données et lorsqu'il obtient de nouvelles données, il classe ces données dans une catégorie très **similaire** au nouvelles données.

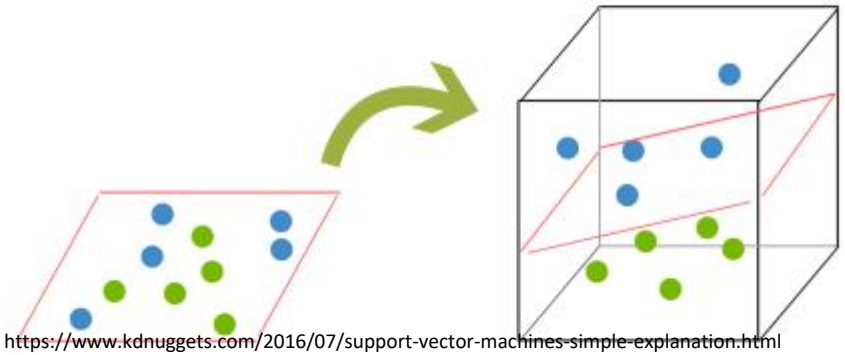




Descriptors → Classification : Other classifiers

SVM est utilisé pour résoudre des problèmes complexes de classification, de régression et de détection de valeurs aberrantes en effectuant des transformations de données optimales qui **déterminent les limites** entre les points de données en fonction de classes, d'étiquettes ou de sorties prédéfinies.

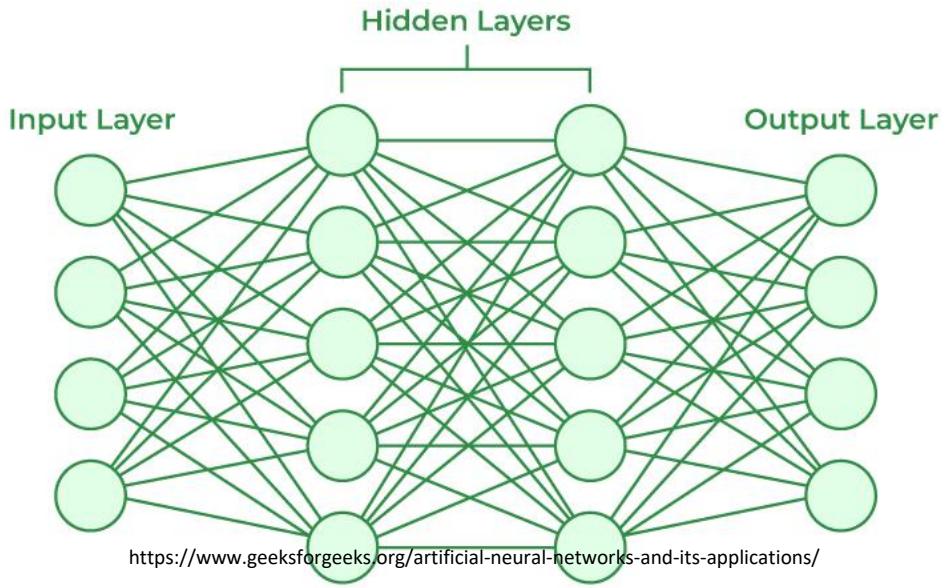
Objectif : Identifier un **hyperplan qui sépare** distinctement les points de données des différentes classes de telle manière que **la plus grande marge** sépare les classes considérées.



https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

NN contient des nœuds (**neurones artificiels**) disposés en une série de **couches interconnectées** d'une couche à l'autre. Chacune de ces connexions a des **poids** qui déterminent l'influence de chaque nœud.

Pendant la phase **d'entrainement**, les NN sont formés pour minimiser la différence entre le résultat prévu et les valeurs cibles réelles dans un ensemble de données donné.



https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/

Descriptors → Classification : Other classifiers

1.1. Linear Models
- 1.1.1. Ordinary Least Squares
- 1.1.2. Ridge regression and classification
- 1.1.3. Lasso
- 1.1.4. Multi-task Lasso
- 1.1.5. Elastic-Net
- 1.1.6. Multi-task Elastic-Net
- 1.1.7. Least Angle Regression
- 1.1.8. LARS Lasso
- 1.1.9. Orthogonal Matching Pursuit (OMP)
- 1.1.10. Bayesian Regression
- 1.1.11. Logistic regression
- 1.1.12. Generalized Linear Models
- 1.1.13. Stochastic Gradient Descent - SGD
- 1.1.14. Perceptron
- 1.1.15. Passive Aggressive Algorithms
- 1.1.16. Robustness regression: outliers and modeling errors
- 1.1.17. Quantile Regression
- 1.1.18. Polynomial regression: extending linear models with basis functions

1.2. Linear and Quadratic Discriminant Analysis
- 1.2.1. Dimensionality reduction using Linear Discriminant Analysis
- 1.2.2. Mathematical formulation of the LDA and QDA classifiers
- 1.2.3. Mathematical formulation of LDA dimensionality reduction
- 1.2.4. Shrinkage and Covariance Estimator
- 1.2.5. Estimation algorithms

1.3. Kernel ridge regressio

1.4. Support Vector Machines
- 1.4.1. Classification
- 1.4.2. Regression
- 1.4.3. Density estimation, novelty detection
- 1.4.4. Complexity
- 1.4.5. Tips on Practical Use
- 1.4.6. Kernel functions
- 1.4.7. Mathematical formulation
- 1.4.8. Implementation details

1.5. Stochastic Gradient Descent
- 1.5.1. Classification
- 1.5.2. Regression
- 1.5.3. Online One-Class SVM
- 1.5.4. Stochastic Gradient Descent for sparse dat
- 1.5.5. Complexity
- 1.5.6. Stopping criterion
- 1.5.7. Tips on Practical Use
- 1.5.8. Mathematical formulation
- 1.5.9. Implementation details

1.6. Nearest Neighbors
- 1.6.1. Unsupervised Nearest Neighbors
- 1.6.2. Nearest Neighbors Classification
- 1.6.3. Nearest Neighbors Regression
- 1.6.4. Nearest Neighbor Algorithms
- 1.6.5. Nearest Centroid Classifier
- 1.6.6. Nearest Neighbors Transformer
- 1.6.7. Neighborhood Components Analysi

1.7. Gaussian Processes
- 1.7.1. Gaussian Process Regression (GPR)
- 1.7.2. Gaussian Process Classification (GPC)
- 1.7.3. GPC examples

1.8. Cross decomposition
- 1.8.1. PLSCanonical
- 1.8.2. PLSSVD
- 1.8.3. PLSRegression
- 1.8.4. Canonical Correlation Analysis

1.9. Naive Bayes
- 1.9.1. Gaussian Naive Bayes
- 1.9.2. Multinomial Naive Bayes
- 1.9.3. Complement Naive Bayes
- 1.9.4. Bernoulli Naive Bayes
- 1.9.5. Categorical Naive Bayes
- 1.9.6. Out-of-core naive Bayes model fitting

1.10. Decision Trees
- 1.10.1. Classification
- 1.10.2. Regression
- 1.10.3. Multi-output problems
- 1.10.4. Complexity
- 1.10.5. Tips on practical use
- 1.10.6. Tree algorithms: ID3, C4.5, C5.0 a
- 1.10.7. Mathematical formulation
- 1.10.8. Missing Values Support
- 1.10.9. Minimal Cost-Complexity Pruning

1.11. Ensembles: Gradient boosting, random forests stacking
- 1.11.1. Gradient-boosted trees
- 1.11.2. Random forests and other randomized tree
- 1.11.3. Bagging meta-estimator
- 1.11.4. Voting Classifier
- 1.11.5. Voting Regressor
- 1.11.6. Stacked generalization
- 1.11.7. AdaBoost

1.12. Multiclass and multioutput algorithms
- 1.12.1. Multiclass classification
- 1.12.2. Multilabel classification
- 1.12.3. Multiclass-multioutput classification
- 1.12.4. Multioutput regression

1.13. Feature selection
- 1.13.1. Removing features with low variance
- 1.13.2. Univariate feature selection
- 1.13.3. Recursive feature elimination
- 1.13.4. Feature selection using SelectFromModel
- 1.13.5. Sequential Feature Selection
- 1.13.6. Feature selection as part of a pipeline

1.14. Semi-supervised learning
- 1.14.1. Self Training
- 1.14.2. Label Propagation

1.15. Isotonic regression

1.16. Probability calibration
- 1.16.1. Calibration curves
- 1.16.2. Calibrating a classifier
- 1.16.3. Usage

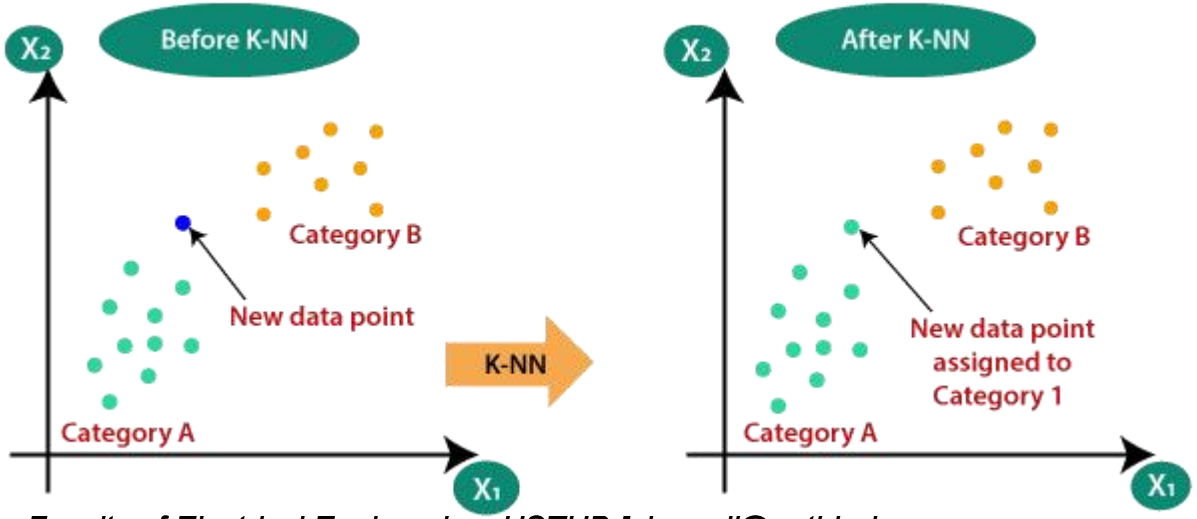
1.17. Neural network models (supervised)
- 1.17.1. Multi-layer Perceptron
- 1.17.2. Classification
- 1.17.3. Regression
- 1.17.4. Regularization
- 1.17.5. Algorithms
- 1.17.6. Complexity
- 1.17.7. Mathematical formulation
- 1.17.8. Tips on Practical Use

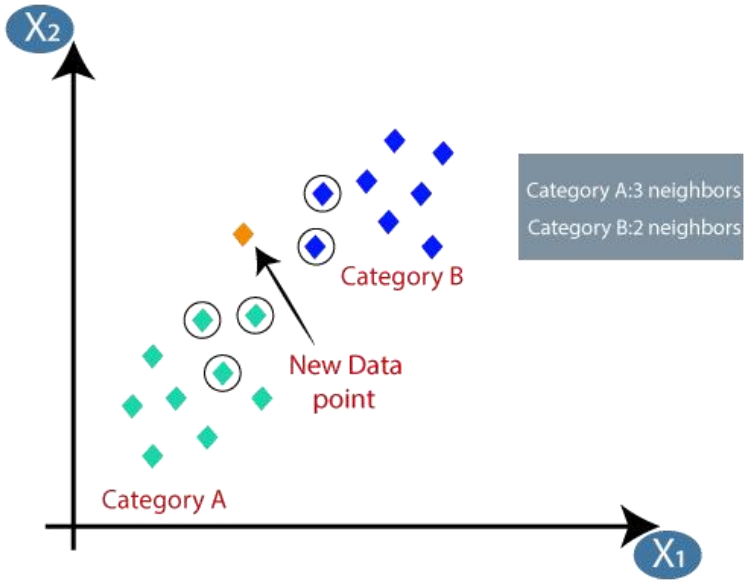
Engineering, USTHB [akourgli@usthb.dz
http://perso.usthb.dz/~akourgli/

Descriptors → Classification : example K-NN classifier

K-Nearest Neighbor is one of the **simplest non parametric** Machine Learning algorithms based on Supervised Learning technique.

- It stores all the available data and classifies a new data point based on the **similarity**. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification (mainly)
- It is a **lazy learner** algorithm because it **does not learn** from the training set. At the **training phase**, it just stores the dataset and when it gets new data, then it classifies that data into a category that is **much similar** to the new data.

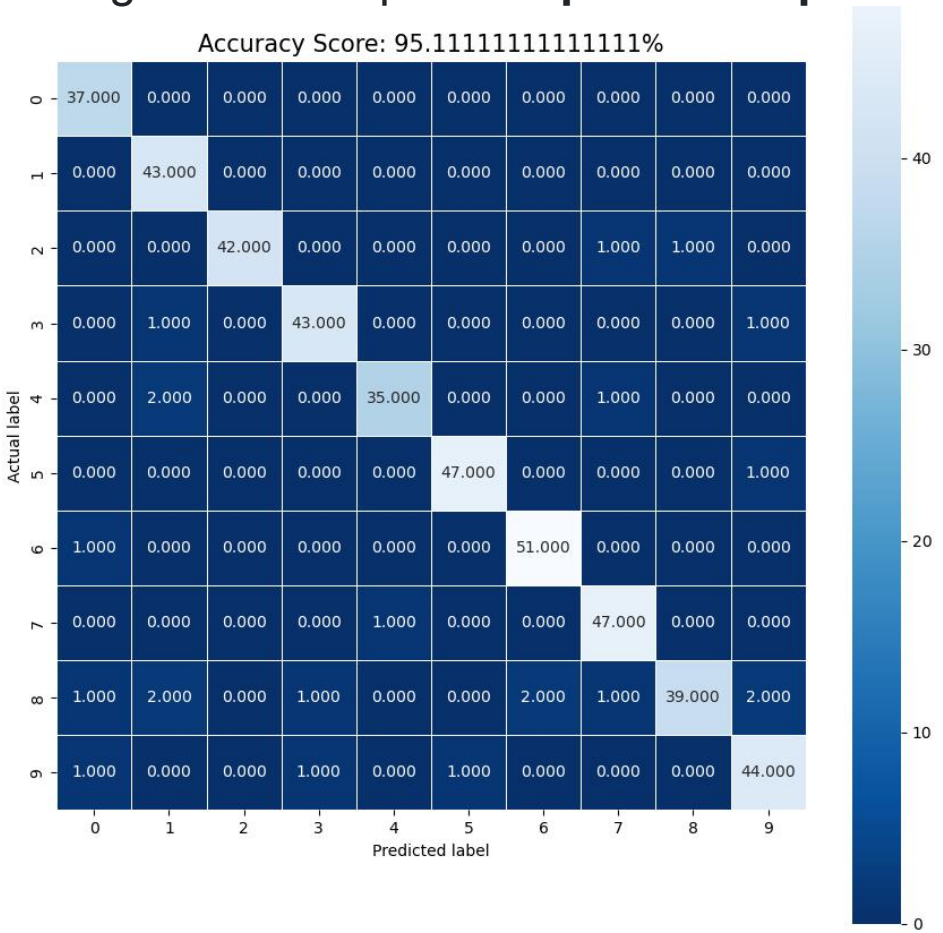




https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

Descripteurs → Classification : exemple K-NN

K-Nearest Neighbour est l'un des algorithmes d'apprentissage automatique **non paramétriques les plus simples** basés sur la technique d'apprentissage supervisé.

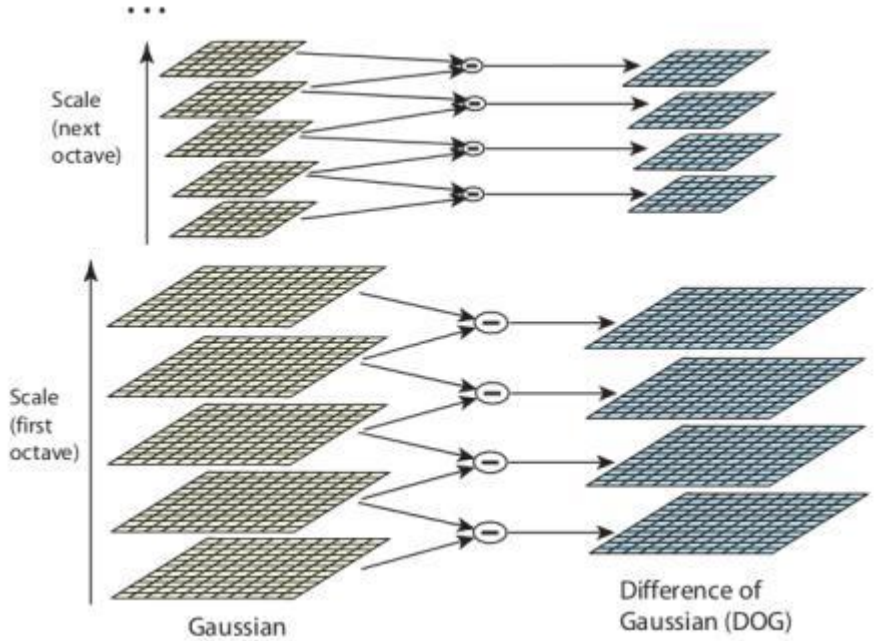


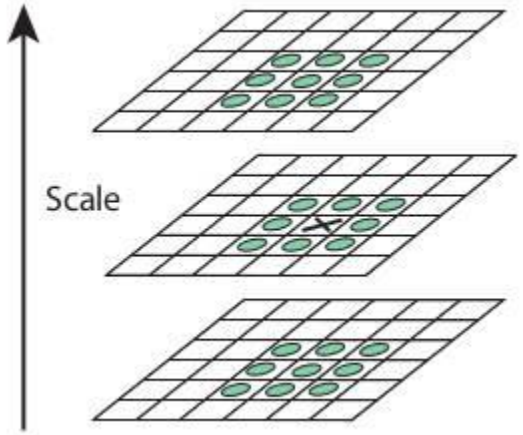
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier
https://medium.com/swlh/histogram-of-oriented-gradients-hog-for-multiclass-image-classification-and-image-recommendation-cf0ea2caaae8
https://www.kaggle.com/code/anirbanmalick/image-classification-using-hog-knn-98-acc
https://medium.com/swlh/histogram-of-oriented-gradients-hog-for-multiclass-image-classification-and-image-recommendation-cf0ea2caaae8

Descripteurs

SIFT (scale-invariant feature transform **)** est un algorithme qui permet de détecter, décrire et faire correspondre des caractéristiques locales dans les images **.**

1. Détection des extrema dans l'espace d'échelle : le laplacien du gaussien est calculé pour l'image avec différentes valeurs. LoG agit comme un détecteur de blobs qui détecte les blobs de différentes tailles en raison du changement de σ qui agit comme paramètre de mise à l'échelle. Une fois ce DoG trouvé, les images sont recherchées pour les extrema locaux sur l'échelle et l'espace.
2. des points clés : un concept similaire au détecteur de coin Harris est utilisé. Une matrice hessienne 2x2 (H) est utilisée pour calculer la courbure principale.

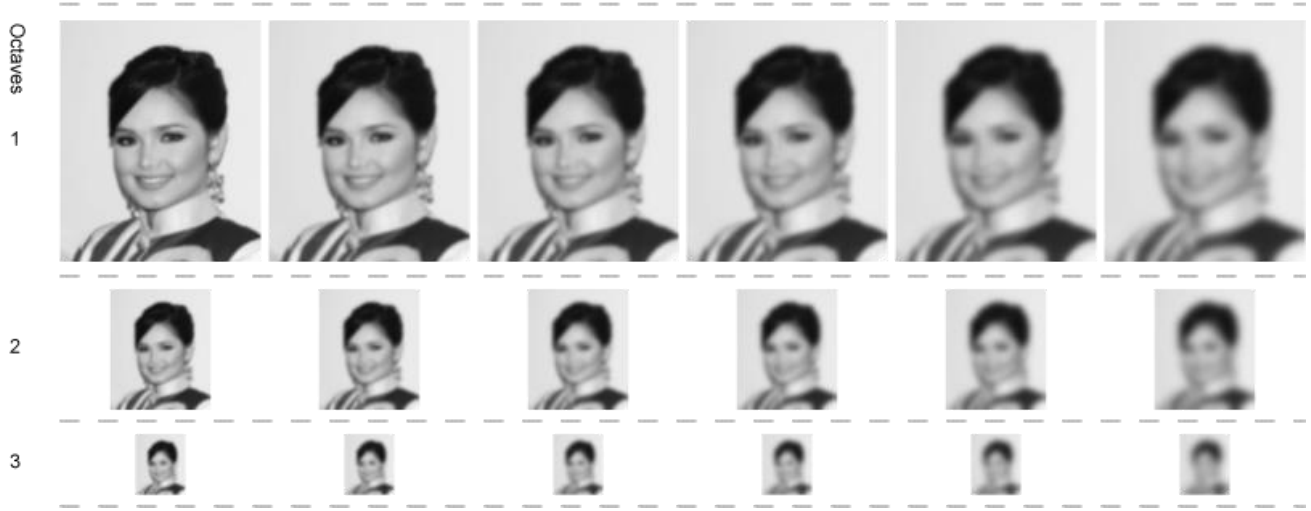




Descripteurs

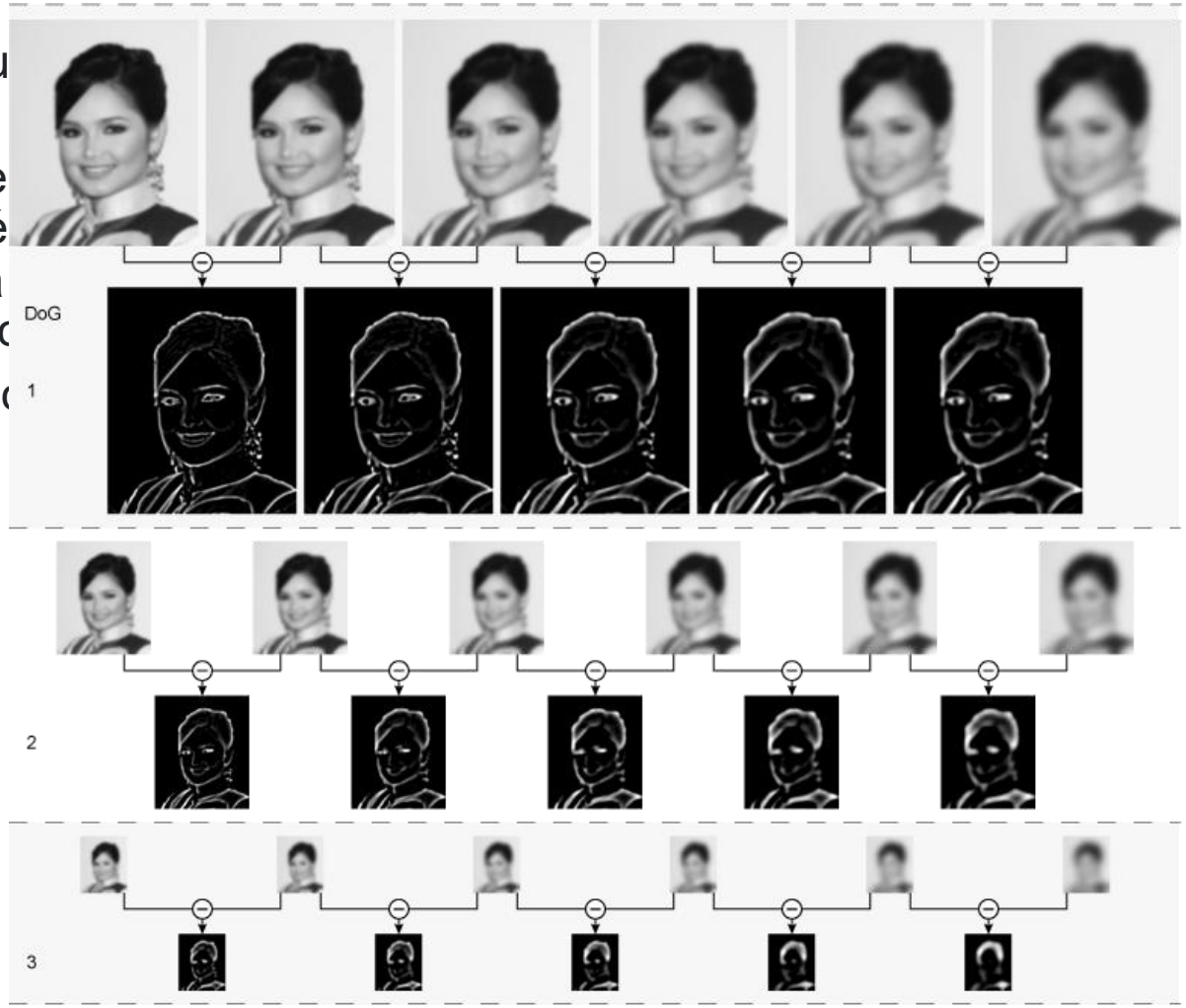
SIFT (scale-invariant feature transform **)** est un algorithme qu
caractéristiques locales dans les images **.**

1. Détection des extrema dans l'espace d'échelle : le laplacie
   valeurs. LoG agit comme un détecteur de blobs qui dé
   changement de σ qui agit comme paramètre de mise à
   recherchées pour les extrema locaux sur l'échelle et l'espac
2. Localisation des points clés : Un concept similaire au déte
   2x2 (H) est utilisée pour calculer la courbure principale.



https://www.researchgate.net/publication/331185020_A_novel_SIFT_architecture_and_ASIC_implementation_for_real_time_SOC_application/figures?lo=1

https://upload.wikimedia.org/wikipedia/commons/8/8e/Differences-of-Grandients.png

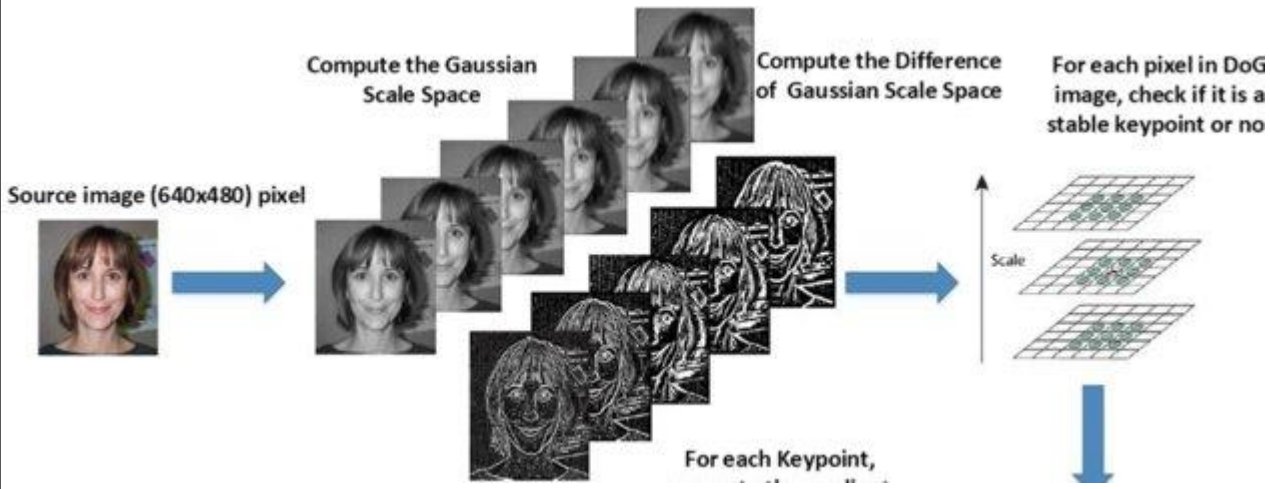


https://www.researchgate.net/publication/256546531_Workload_Analysis_and_Efficient_OpenCL-based_Implementation_of_SIFT_Algorithm_on_a_Smartphone/figures?lo=1

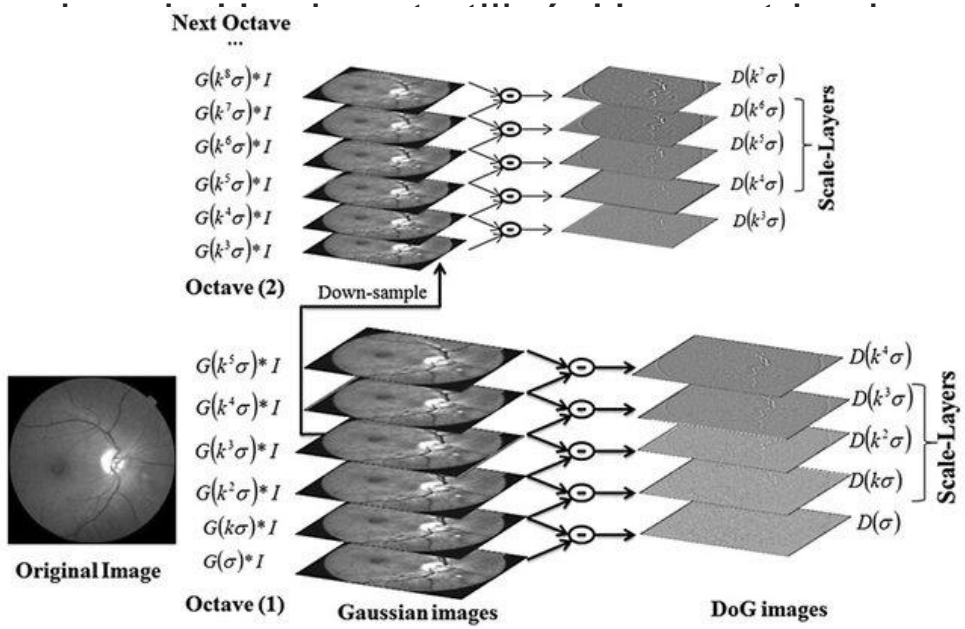
Descripteurs

SIFT (scale-invariant feature transform **)** est un algorithme qui permet de détecter, décrire et faire correspondre des caractéristiques locales dans les images **.**

1. Détection des extrema dans l'espace d'échelle : le laplacien du gaussien est trouvé pour l'image avec différentes valeurs. LoG agit comme un détecteur de blobs qui détecte les blobs de différentes tailles en raison du changement de σ qui agit comme paramètre de mise à l'échelle. Une fois ces DoG trouvés, les images sont recherchées pour les extrema locaux **sur l'échelle et l'espace** .
2. Localisation des points clés : Un concept similaire au détecteu ienne 2x2 (H) est utilisée pour calculer la courbure principale.



https://www.researchgate.net/publication/331185020_A_novel_SIFT_architecture_and_ASIC_implementation_for_real_time_SOC_application/figures?lo=1

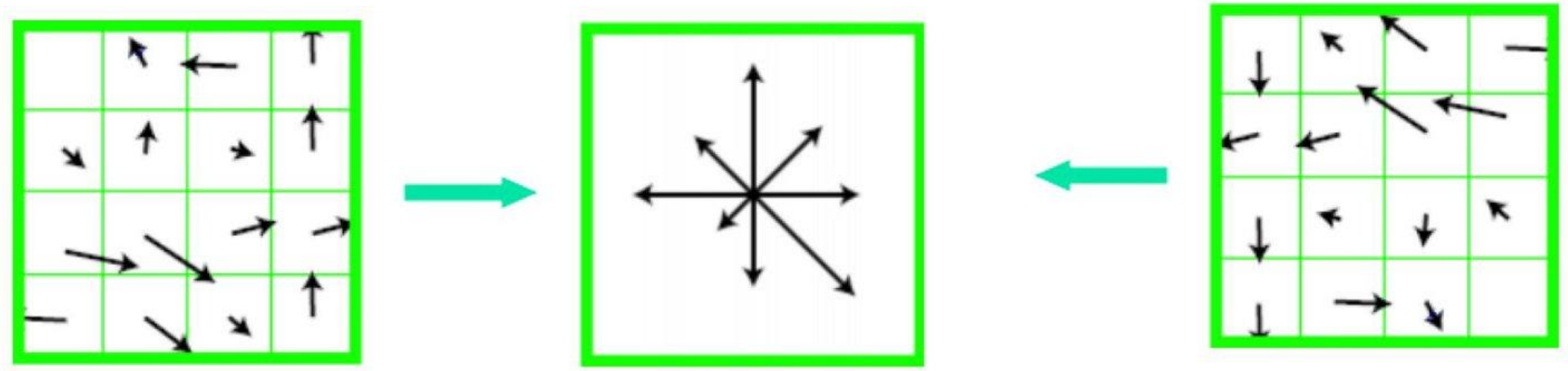


https://www.researchgate.net/publication/256546531_Workload_Analysis_and_Efficient_OpenCL-based_Implementation_of_SIFT_Algorithm_on_a_Smartphone/figures?lo=1

Descripteurs

SIFT (scale-invariant feature transform **)** est un algorithme qui permet de détecter, décrire et faire correspondre des caractéristiques locales dans les images **.**

3. Attribution d'orientation : Une orientation est attribuée à chaque point clé pour obtenir l'invariance à la rotation de l'image. Un voisinage est défini autour de l'emplacement du point clé en fonction de l'échelle, et l'amplitude et la direction du gradient sont calculées dans cette région. Un histogramme d'orientation avec 36 cases couvrant 360 degrés est créé (il est pondéré par l'amplitude du gradient et une fenêtre circulaire pondérée gaussienne avec σ égal à 1,5 fois l'échelle du point clé .

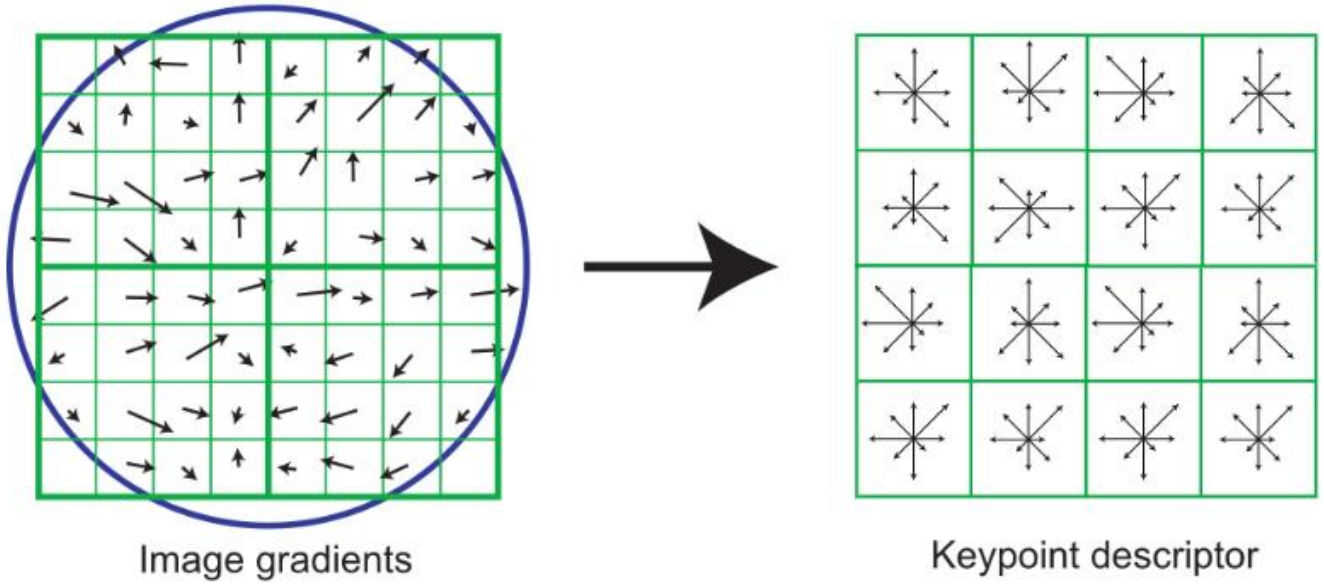


https://www.researchgate.net/figure/lassignement-de-lorientation-au-descripteur-SIFT64_fig10_344224269

Descripteurs

SIFT (scale-invariant feature transform **)** est un algorithme qui permet de détecter, décrire et faire correspondre des caractéristiques locales dans les images **.**

4. Descripteur de point clé : Uu voisinage 16x16 autour du point clé est pris. Il est divisé en 16 sous-blocs de taille 4x4. Pour chaque sous-bloc, 8 histogrammes d'orientation HOG sont créés → Un total de 128 valeurs de HOG sont disponibles. Il est représenté sous forme de vecteur pour former un descripteur de point clé . En plus de cela, plusieurs mesures sont prises pour obtenir une robustesse face aux changements d'éclairage, à la rotation, etc.



https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5053196-decrivez-efficacement-les-features-detectees-avec-sift

Descripteurs

SIFT (scale-invariant feature transform **)** est un algorithme qui permet de détecter, décrire et faire correspondre des caractéristiques locales dans les images **.**

Application : Localisation, tracking de robots ; modélisation 3D

Voir :

- L'amélioration **SURF** (Speeded-Up Robust Feature) de SIFT utilise des filtres de boîte plutôt que d'utiliser la différence de Gauss pour approximer LoG ,

- ORB (Oriented FAST and Rotated BRIEF) est une fusion du détecteur de points clés FAST et des descripteurs BRIEF avec de nombreux changements pour améliorer les performances. **FAST** (Features from Accelerated Segment Test) est utilisé comme détecteur de points clés tandis que **BRIEF** (Binary Robust Independent Elementary Features) est utilisé comme descripteur de point clé

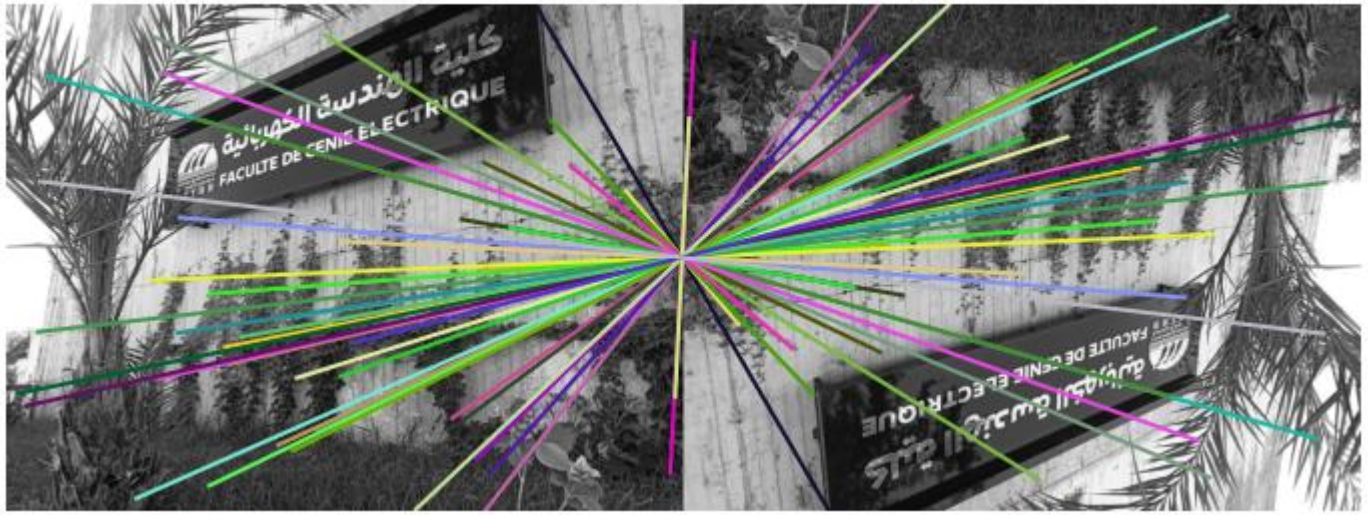
https://docs.opencv.org/3.4/db/d27/tutorial_py_table_of_contents_feature2d.html

Descripteurs

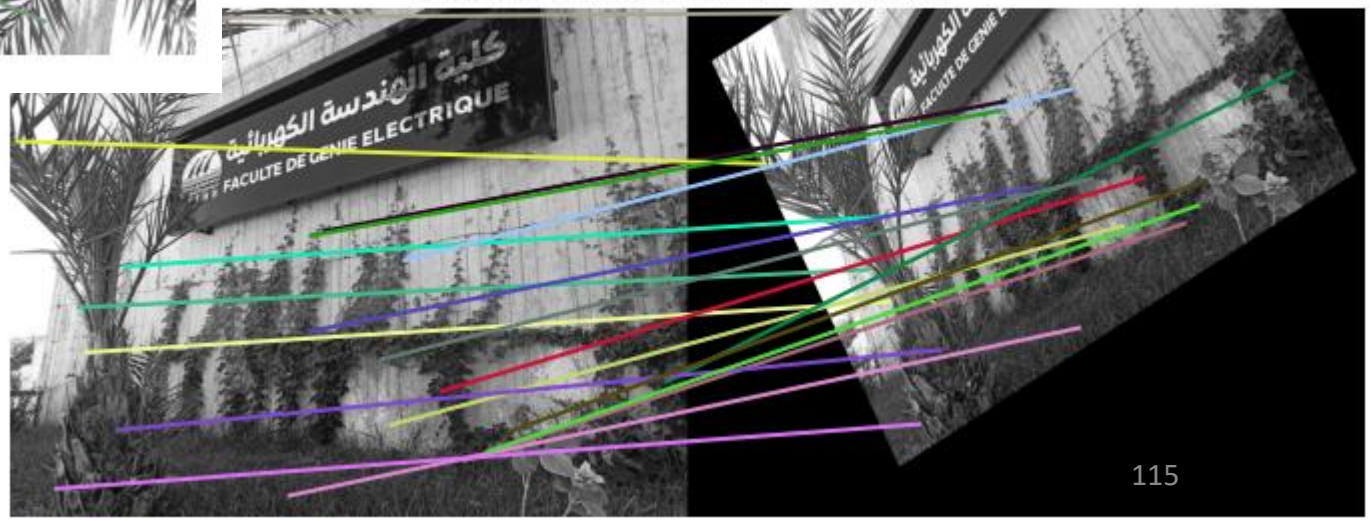
Correspondance à l'aide de SIFT (transformation de caractéristiques invariantes à l'échelle **)**

.

Original Image vs. Flipped Image
(subset of matches for visibility)

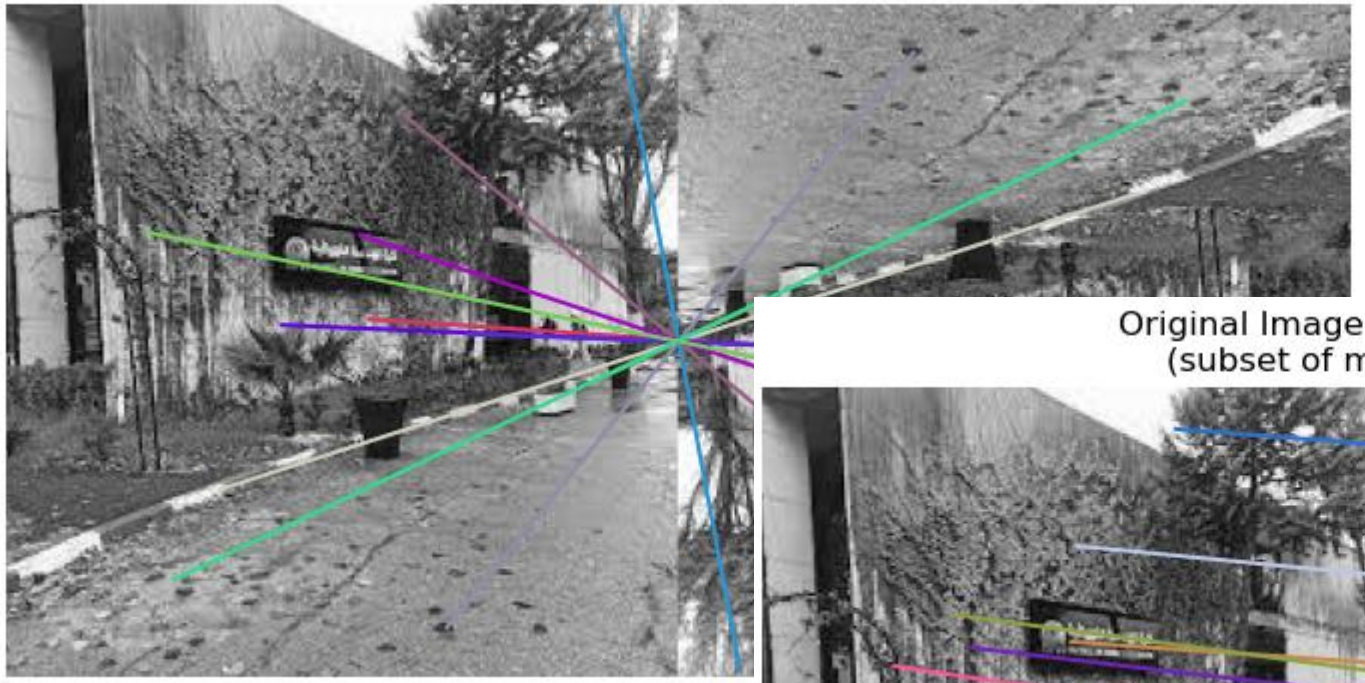


Original Image vs. Transformed Image
(subset of matches for visibility)



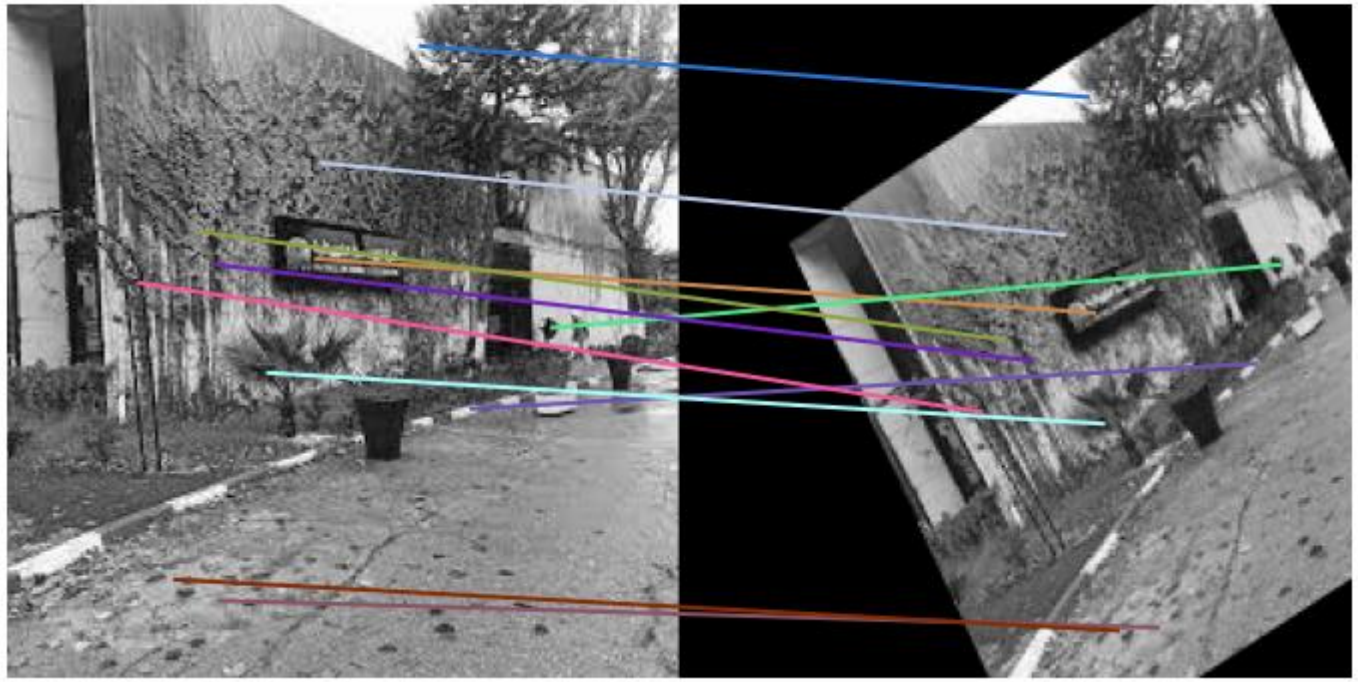
Descripteurs

Original Image vs. Flipped Image
(subset of matches for visibility)



permet de détecter, décrire et faire correspondre des

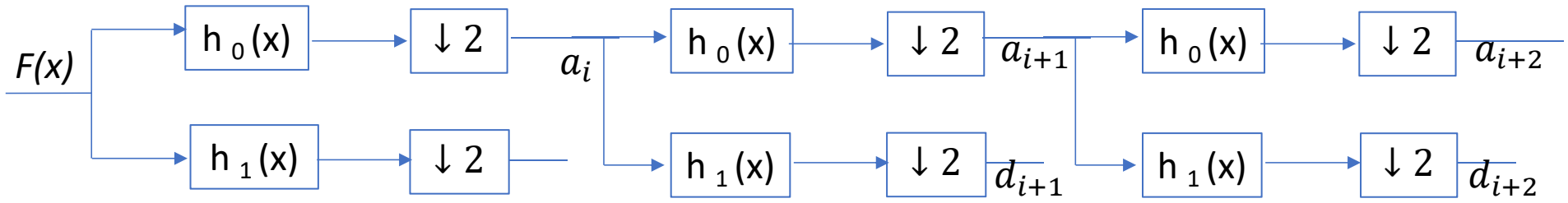
Original Image vs. Transformed Image
(subset of matches for visibility)

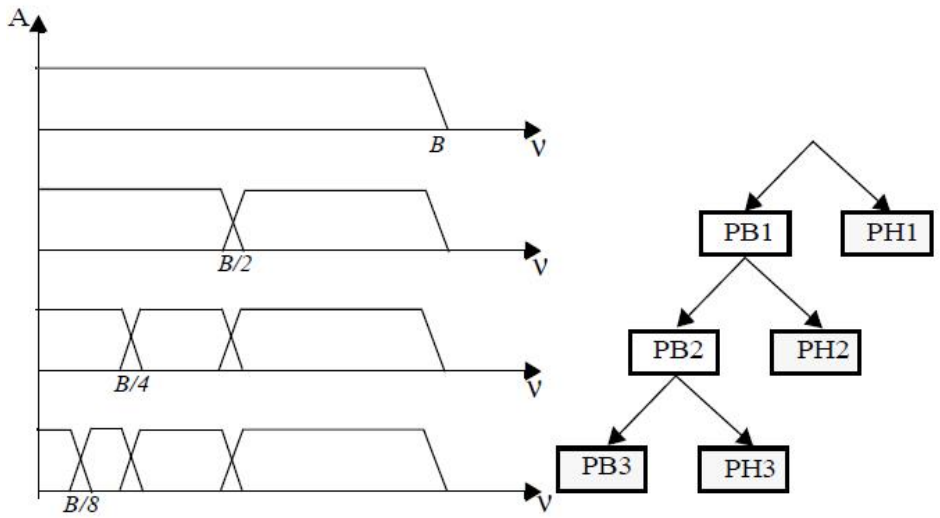


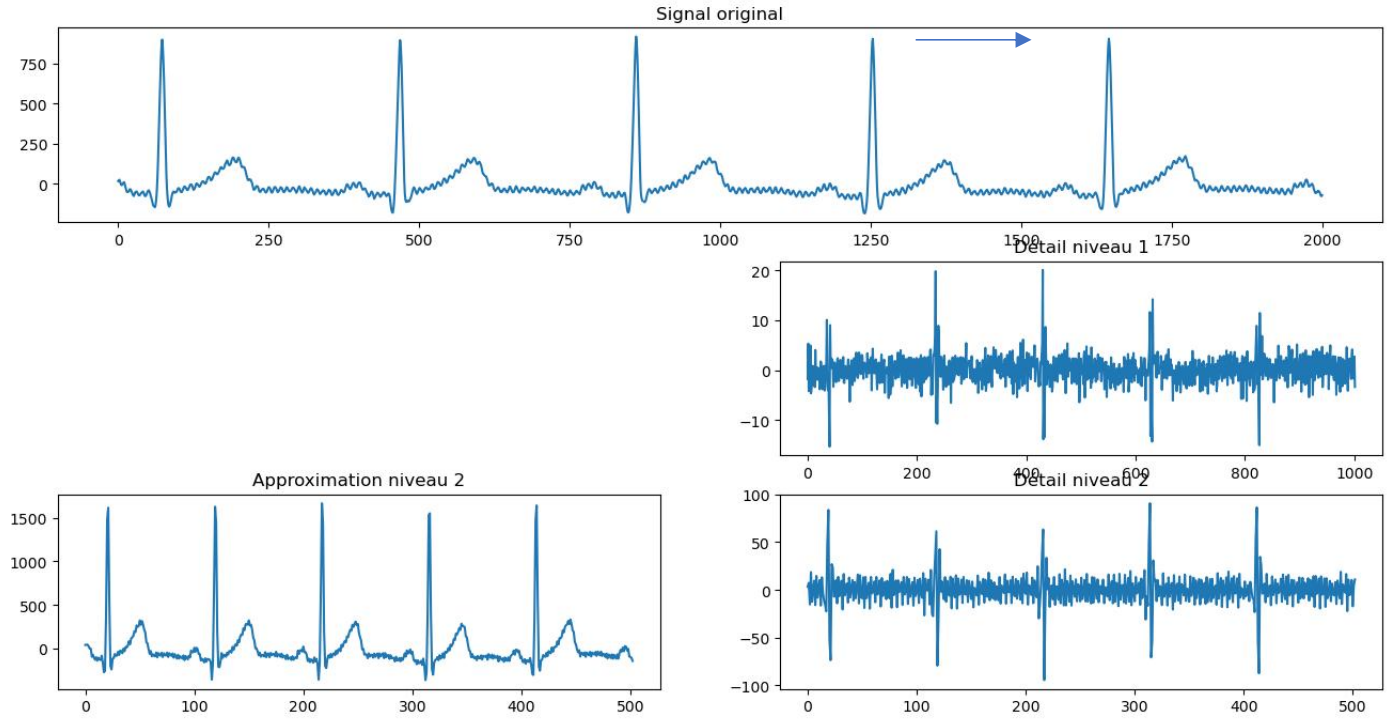
Descripteurs

DWT → 1 dimension

Un signal passe à travers deux filtres, des filtres passe-haut et passe-bas.



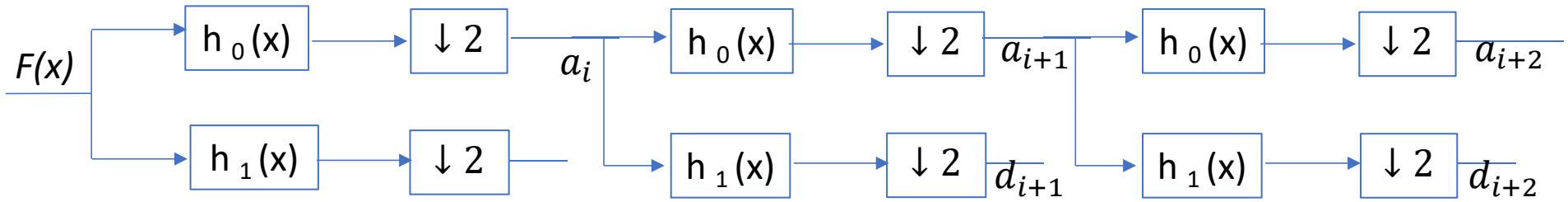




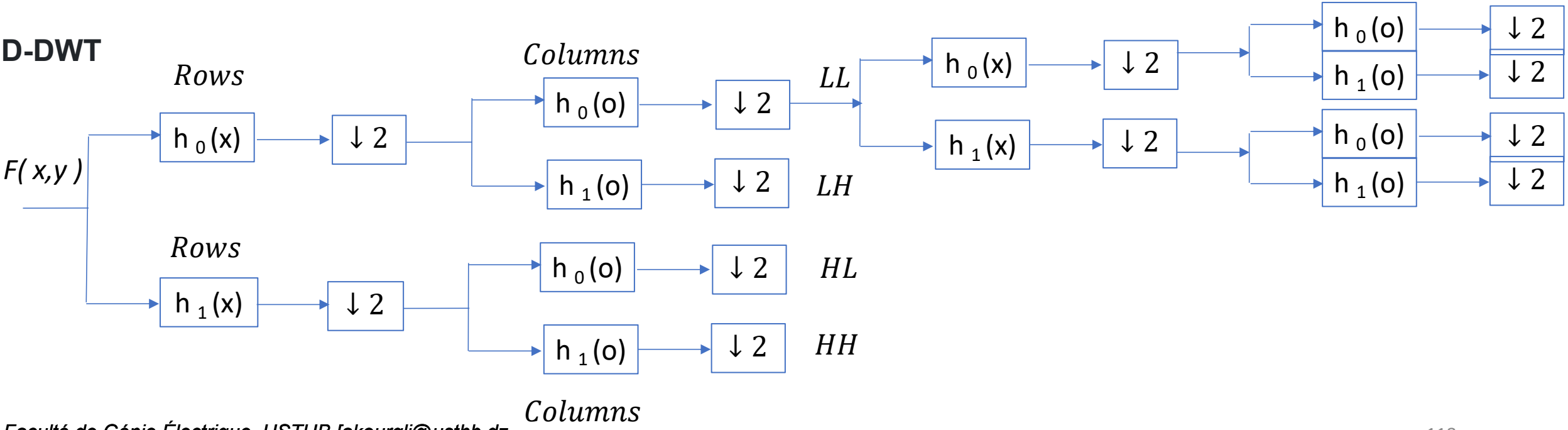
Descripteurs

DWT → 1 dimension

Un signal passe à travers deux filtres, des filtres passe-haut et passe-bas.



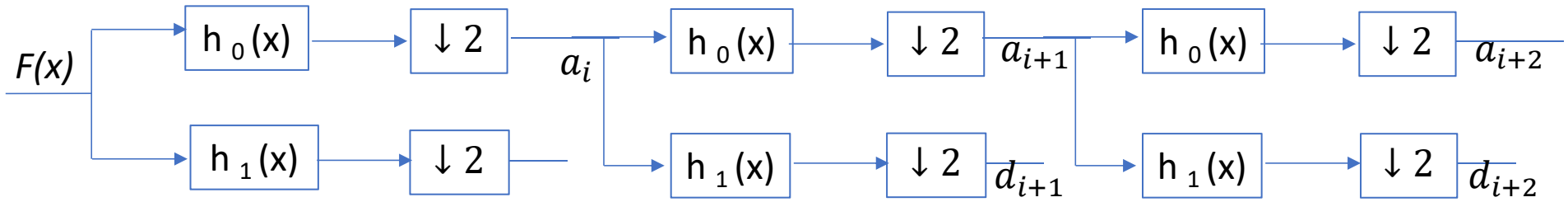
2D-DWT



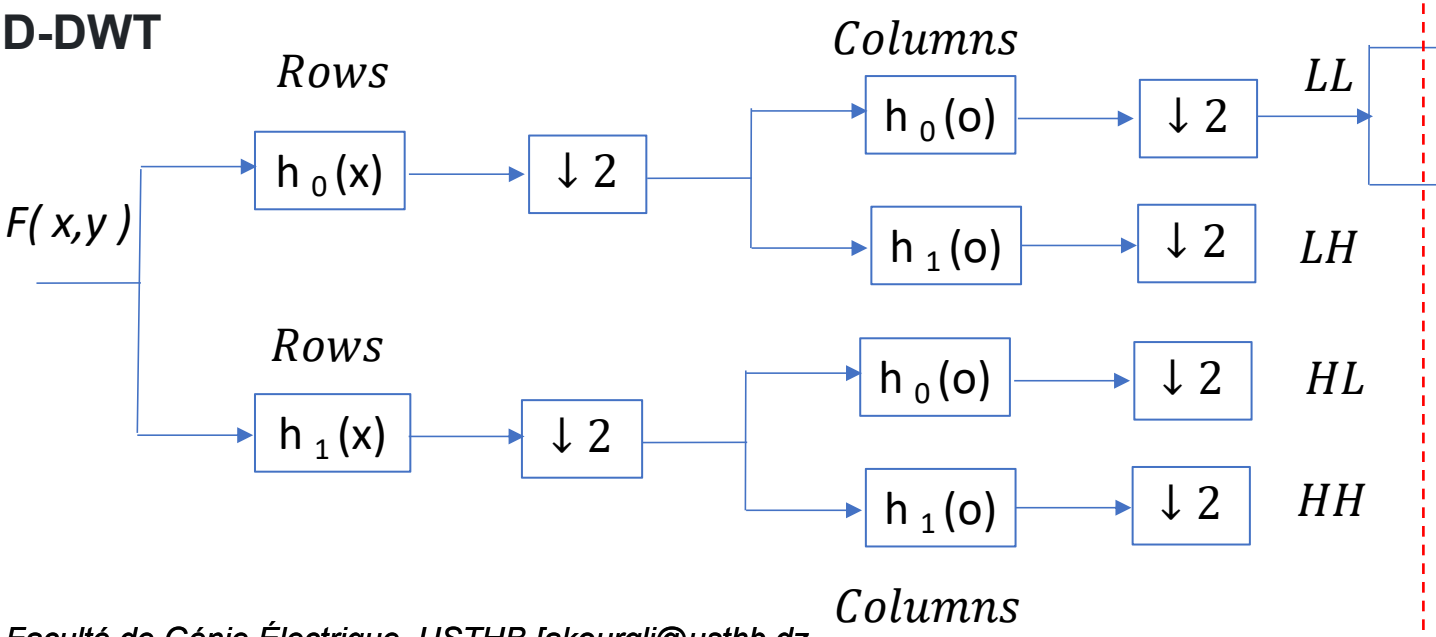
Descripteurs

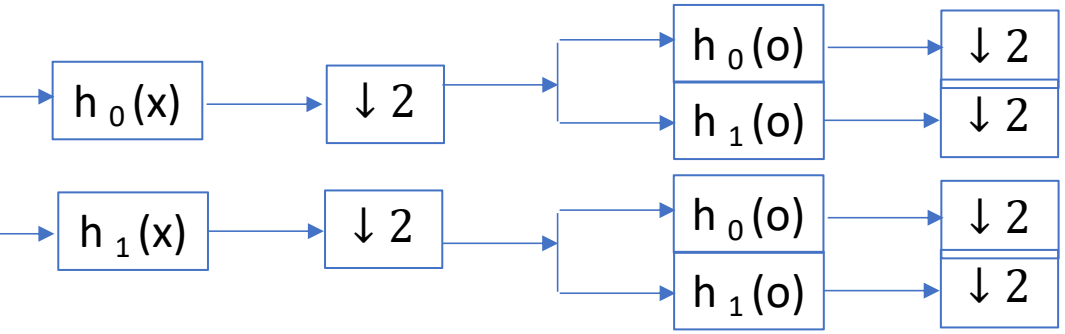
DWT → 1 dimension

Un signal passe à travers deux filtres, des filtres passe-haut et passe-bas.



2D-DWT

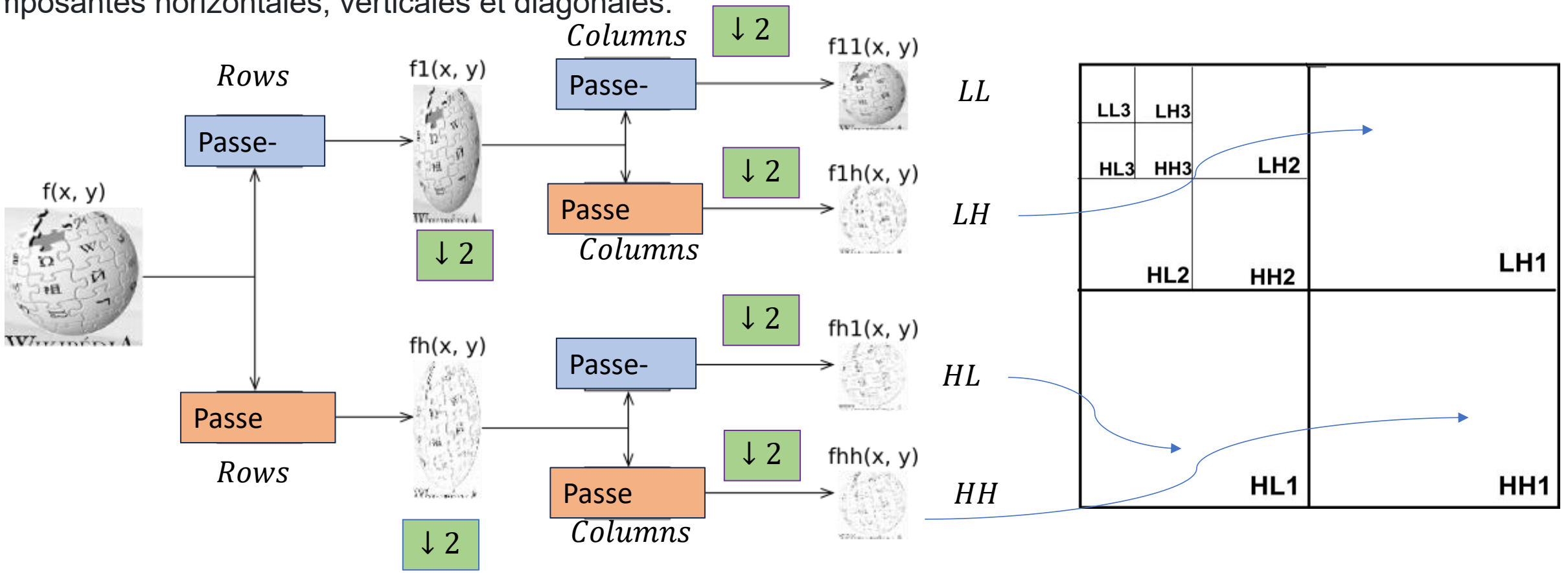




Descripteurs

2D-DWT

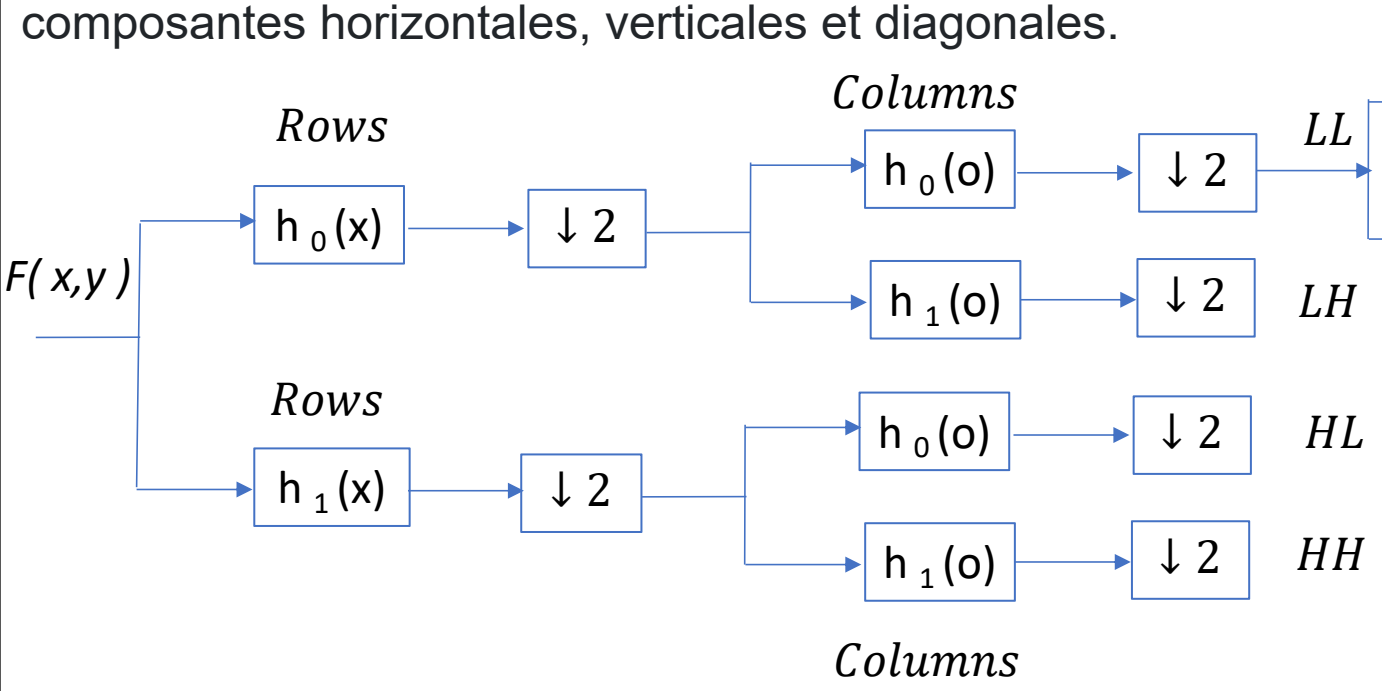
L'image passe à travers deux filtres, des filtres passe-haut et passe-bas. L'image est ensuite décomposée en composantes haute fréquence (détails) et basse fréquence (approximation). A chaque niveau, nous obtenons 4 sous-images. L'approximation montre une tendance globale des valeurs de pixels et des détails sous forme de composantes horizontales, verticales et diagonales.

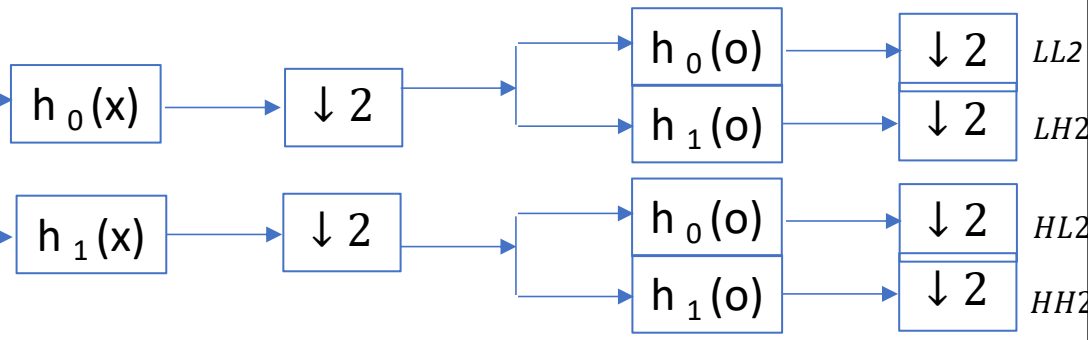


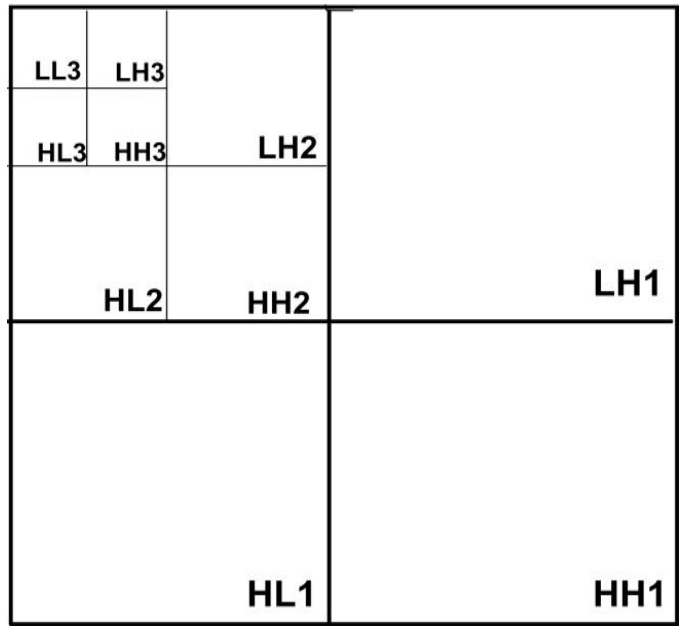
Transformer Descripteurs

2D-DWT

L'image passe à travers deux filtres, des filtres passe-haut et passe-bas. L'image est ensuite décomposée en composantes haute fréquence (détails) et basse fréquence (approximation). A chaque niveau, nous obtenons 4 sous-signaux. L'approximation montre une tendance globale des valeurs de pixels et des détails sous forme de composantes horizontales, verticales et diagonales.



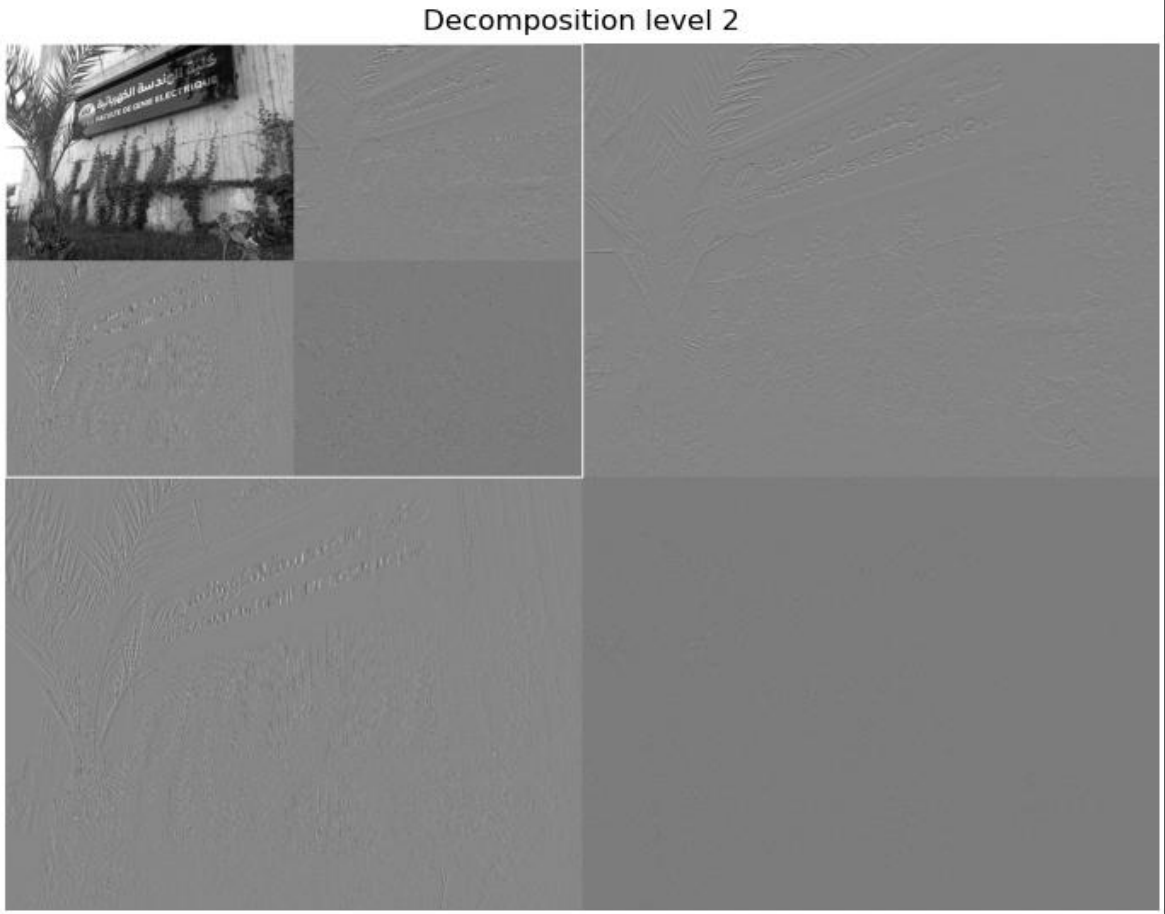




Transformer Descripteurs

2D-DWT

Si ces détails sont insignifiants, ils peuvent être mis à zéro sans impact significatif sur l'image, réalisant ainsi le filtrage et la compression.



Transformer Descripteurs

2D-DWT

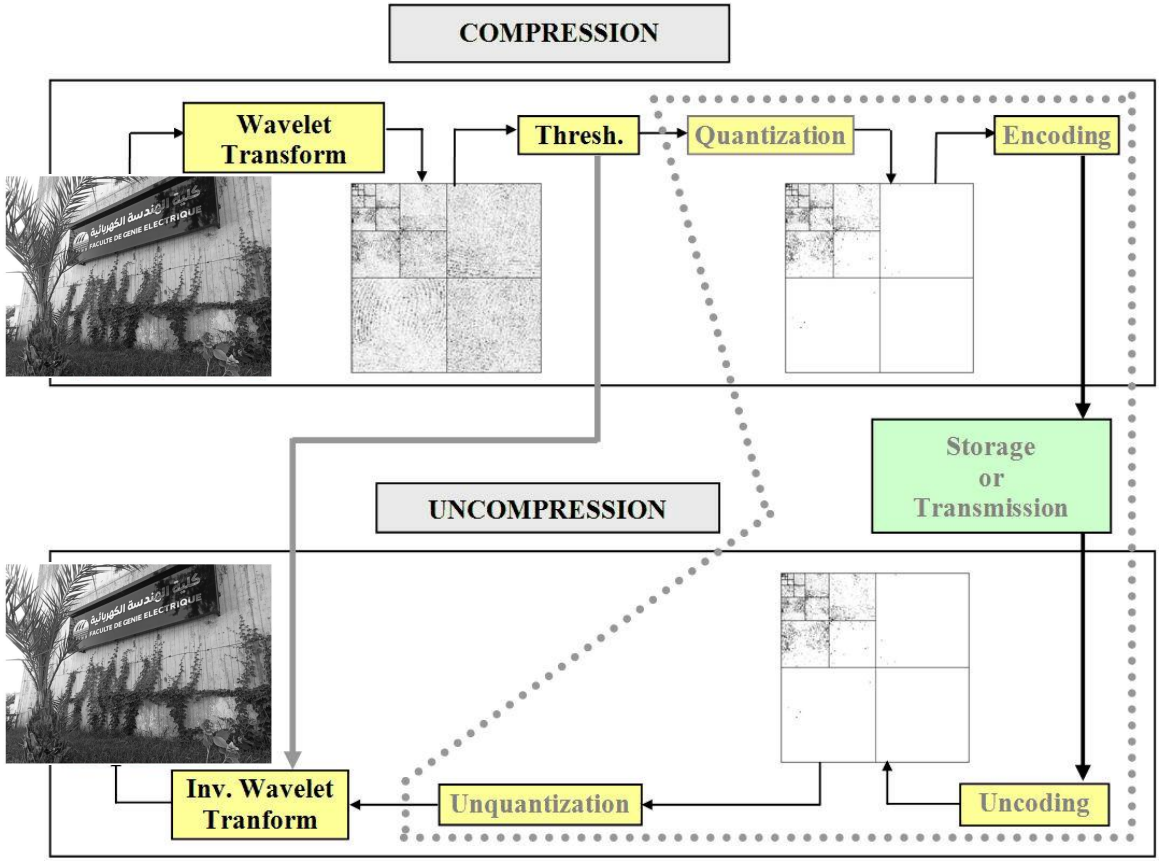
Si ces détails sont insignifiants, ils peuvent être mis à zéro sans impact significatif sur l'image, réalisant ainsi le filtrage et la compression.

Approximation level 2

Transformer Descripteurs

2D-DWT

Si ces détails sont insignifiants, ils peuvent être mis à zéro sans impact significatif sur l'image, réalisant ainsi le filtrage et la compression.



https://www.mathworks.com/help/wavelet/ug/wavelet-compression-for-images.html

Transformer Descripteurs

2D-DWT

Si ces détails sont insignifiants, ils peuvent être mis à zéro sans impact significatif sur l'image, réalisant ainsi le filtrage et la compression.

Transformer Descripteurs

2D-DWT

Si ces détails sont insignifiants, ils peuvent être mis à zéro sans impact significatif sur l'image, réalisant ainsi le filtrage et la compression.

Compressed Image

Transformer Descripteurs

2D-DWT

Si ces détails sont insignifiants, ils peuvent être mis à zéro sans impact significatif sur l'image, réalisant ainsi le filtrage et la compression.

Denoised Image

Descripteurs

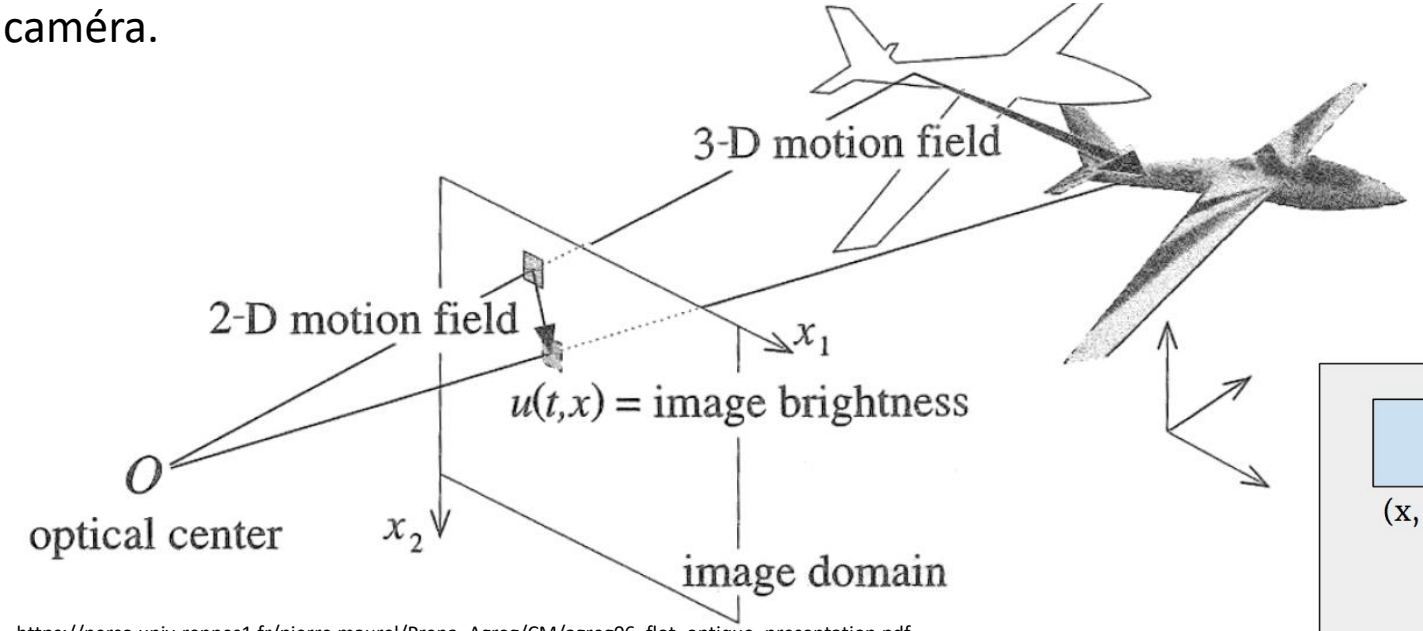
Objectif : Traiter l'entrée vidéo en temps réel. La plupart des techniques sont basées sur l'adressage des relations d'objets (x,y) dans le même cadre combinées à des informations temporelles *t*

Exemple :

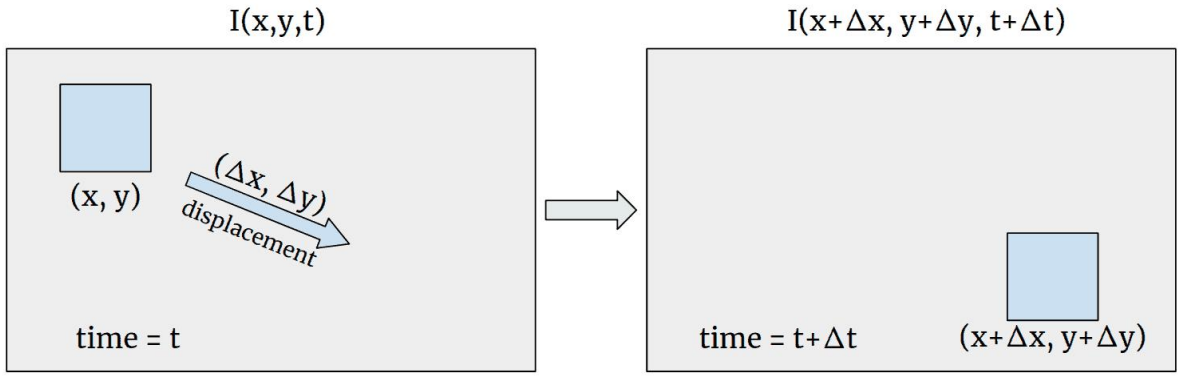
Suivez le mouvement des humains (piétons) ou des véhicules (voitures, camions, avions) sur plusieurs images pour **estimer** leur vitesse actuelle **et prédire** leur **position** dans l'image suivante ou **reconnaître les actions humaines.**

Flux optique :

Mouvement des objets entre **des images consécutives** de la séquence, provoqué par **le mouvement relatif** entre l'objet et la caméra.



https://perso.univ-rennes1.fr/pierre.maurel/Prepa_Agreg/CM/agreg06_flot_optique_presentation.pdf



https://learnopencv.com/optical-flow-in-opencv/

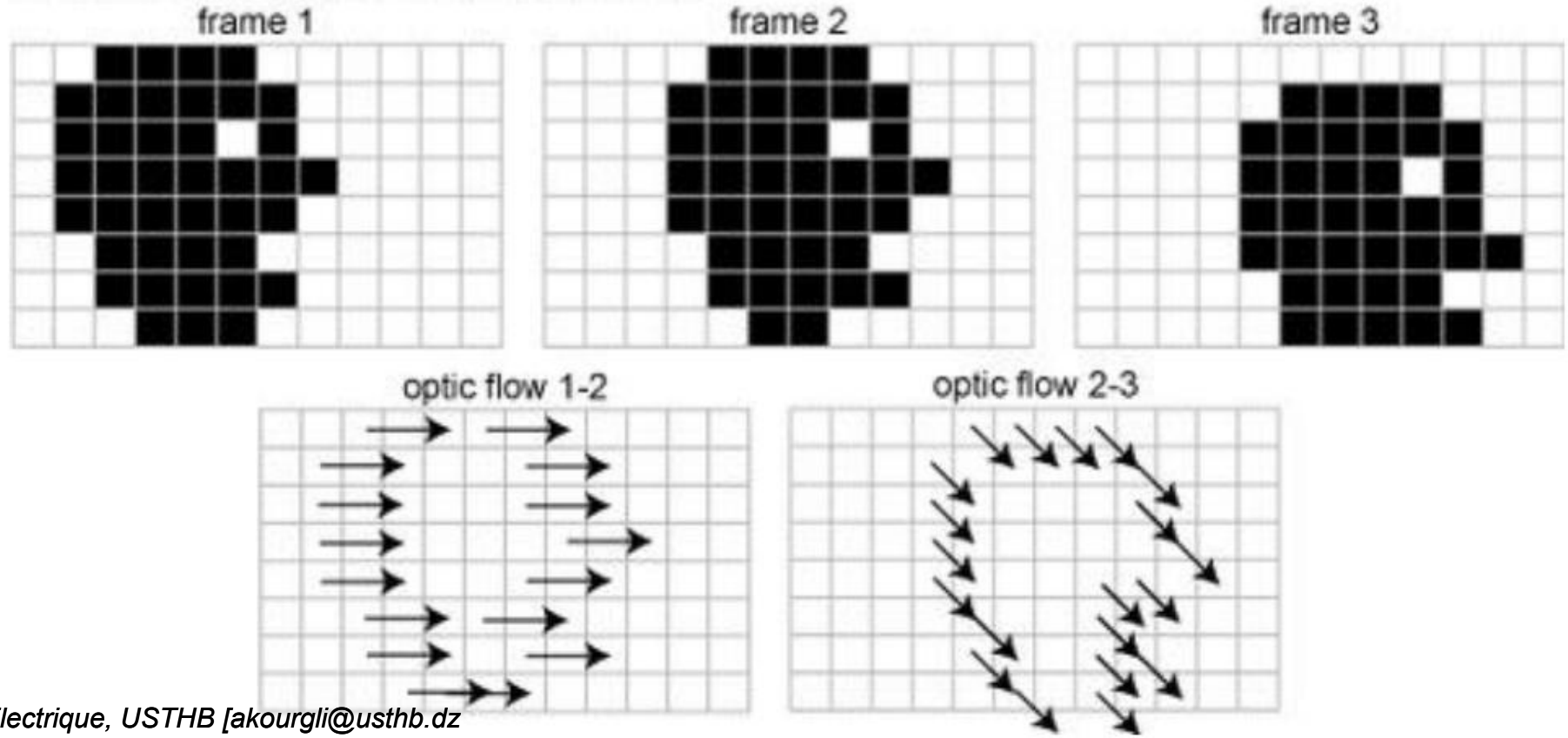
Descripteurs

Flux optique :

Mouvement des objets entre **des images consécutives** de la séquence, provoqué par **le mouvement relatif** entre l'objet et la caméra.

Nous recherchons un champ de **vecteurs déplacement** représentant le **mouvement apparent** de chaque point.

$$I(x, y, t) - I_(x + \Delta x, y + \Delta y, t + \Delta t) = 0$$

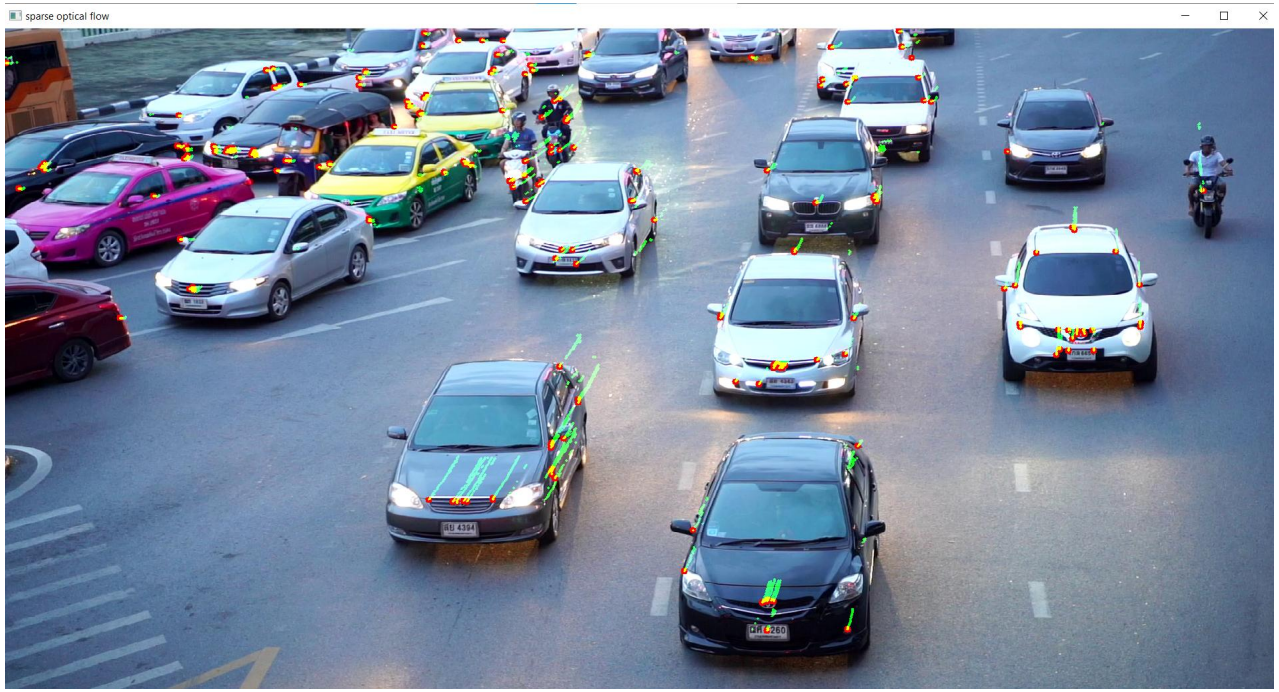


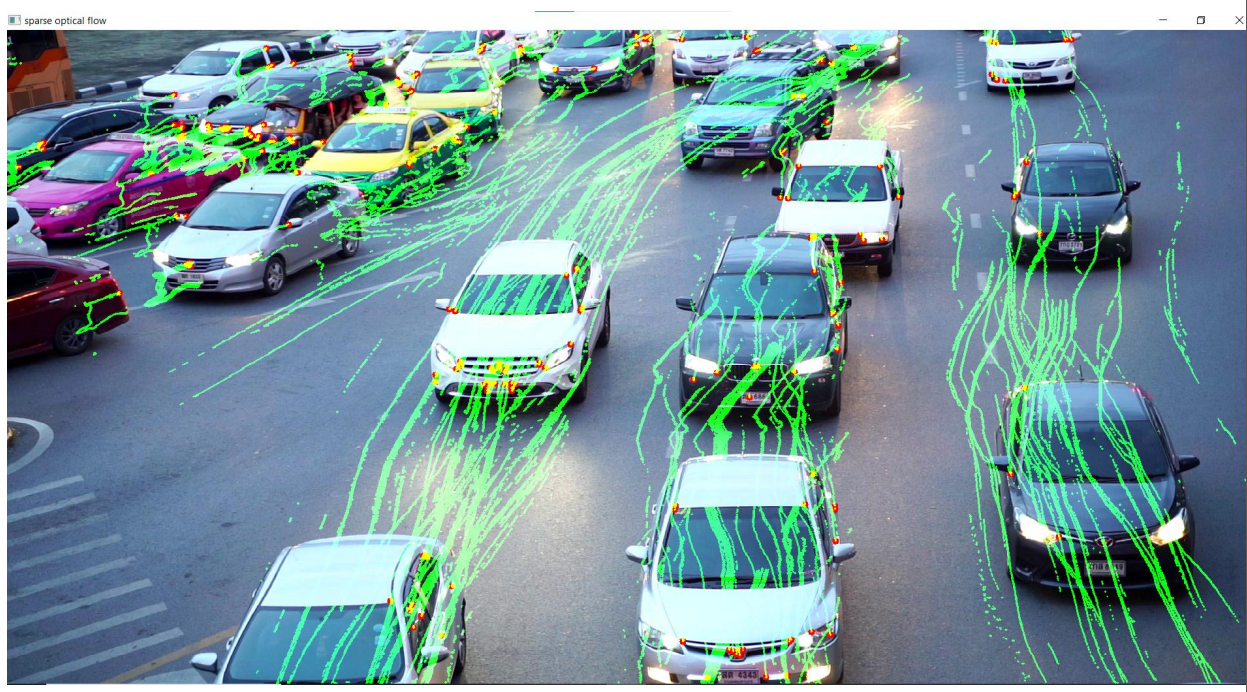
Descripteurs

Flux optique :

Mouvement des objets entre **des images consécutives** de la séquence, provoqué par **le mouvement relatif** entre l'objet et la caméra.

Nous recherchons un champ de **vecteurs déplacement** représentant le **mouvement apparent** de chaque point.





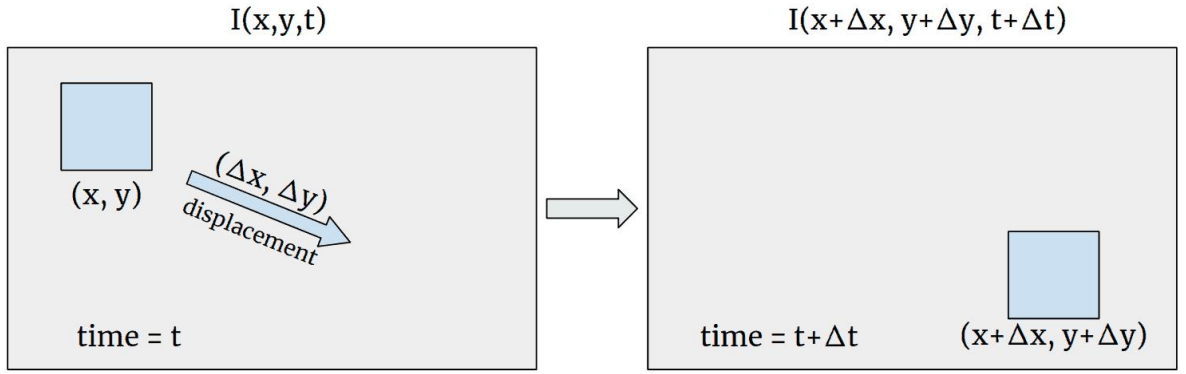
Descripteurs

Flux optique :

Mouvement des objets entre **des images consécutives** de la séquence, provoqué par **le mouvement relatif** entre l'objet et la caméra.

Le flux optique au temps t et au point (x,y) est défini comme la vite

du point image :$\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ → **Vecteur de déplacement**



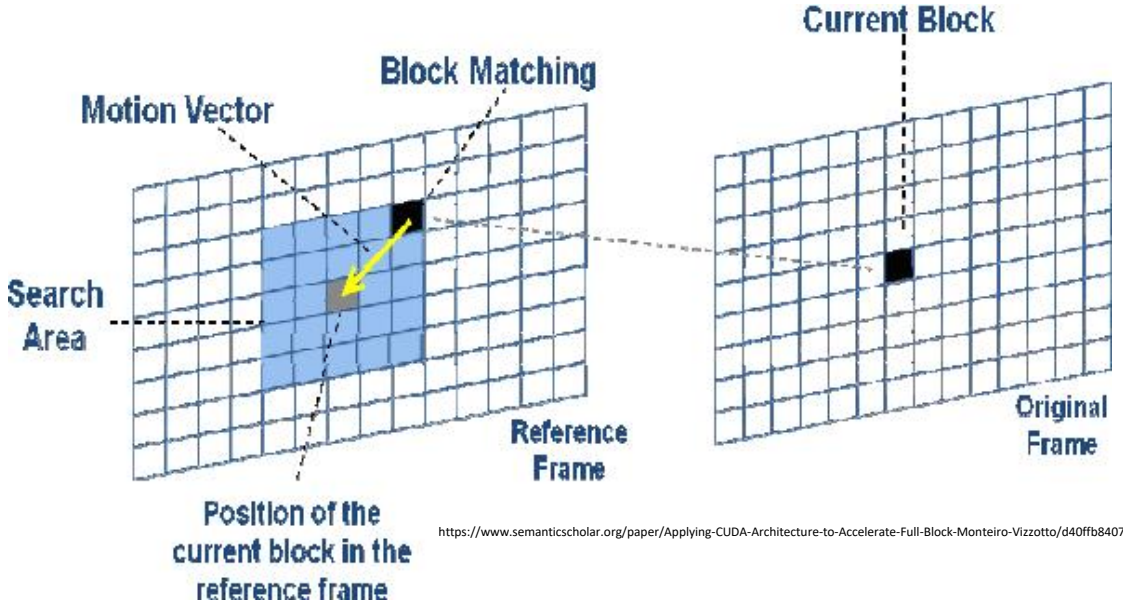
https://learnopencv.com/optical-flow-in-opencv/

Nous recherchons un champ de **vecteurs déplacement** représentant le **Mouvement apparent** de chaque point.

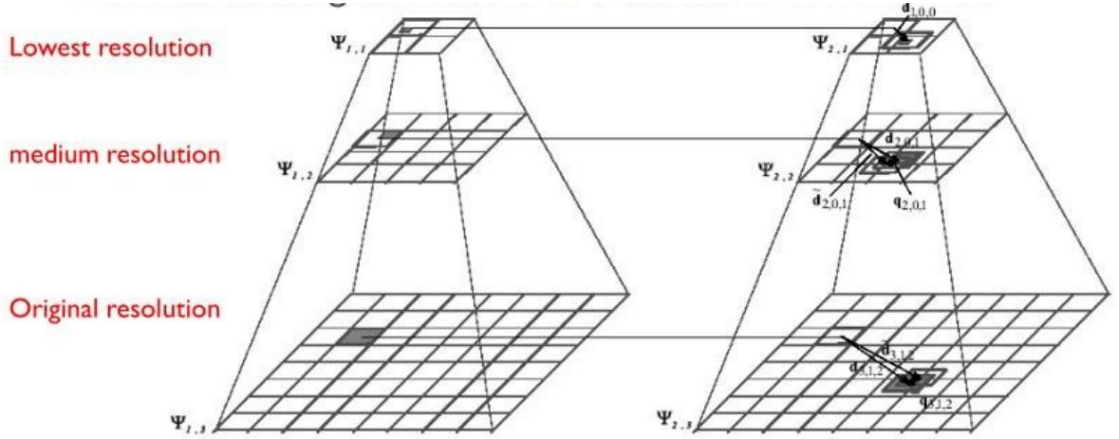
- *Correspondance de bloc* La vitesse est définie comme le décalage qui donne la meilleure correspondance entre les régions, l'image et les images.
- *Flux optique dense,* qui donne les vecteurs de flux de l'ensemble du cadre (tous les pixels) - jusqu'à un vecteur de flux par pixel.
- *Le flux optique sparse* donne les vecteurs de flux de certaines « caractéristiques intéressantes » (quelques pixels représentant les bords ou les coins d'un objet) dans le cadre

Descripteurs

Méthodes de correspondance de blocs : La vitesse est définie comme le décalage qui donne la meilleure correspondance (corrélation) entre les images des régions à différents moments. Trouver la meilleure correspondance revient à maximiser une mesure de similarité, comme minimiser une mesure de distance comme la somme des carrés des différences (SDC).



https://www.semanticscholar.org/paper/Applying-CUDA-Architecture-to-Accelerate-Full-Block-Monteiro-Vizzotto/d40ffb8407ed088f7042e64764f367932ff33a2f



https://www.slideserve.com/file-download/1987727

Utilisé pour la compression vidéo (Norme MPEG) : Découpe en blocs 16×16 ou 32×32, En supposant le même déplacement pour tous les pixels du bloc., Codage : déplacements + erreurs compressées/

Autres applications : Segmentation, détection de mouvement, prédiction

Descripteurs

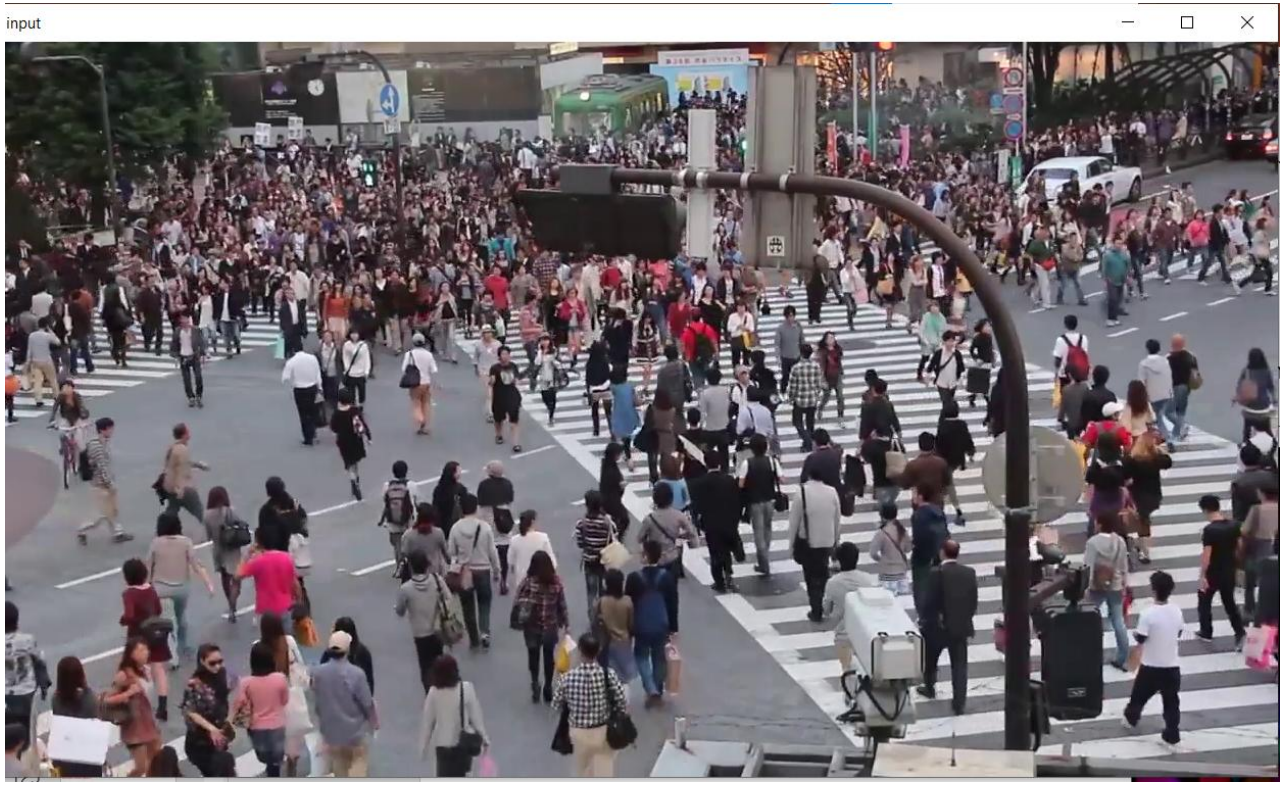
Méthodes différentielles comme différenciation de trames :

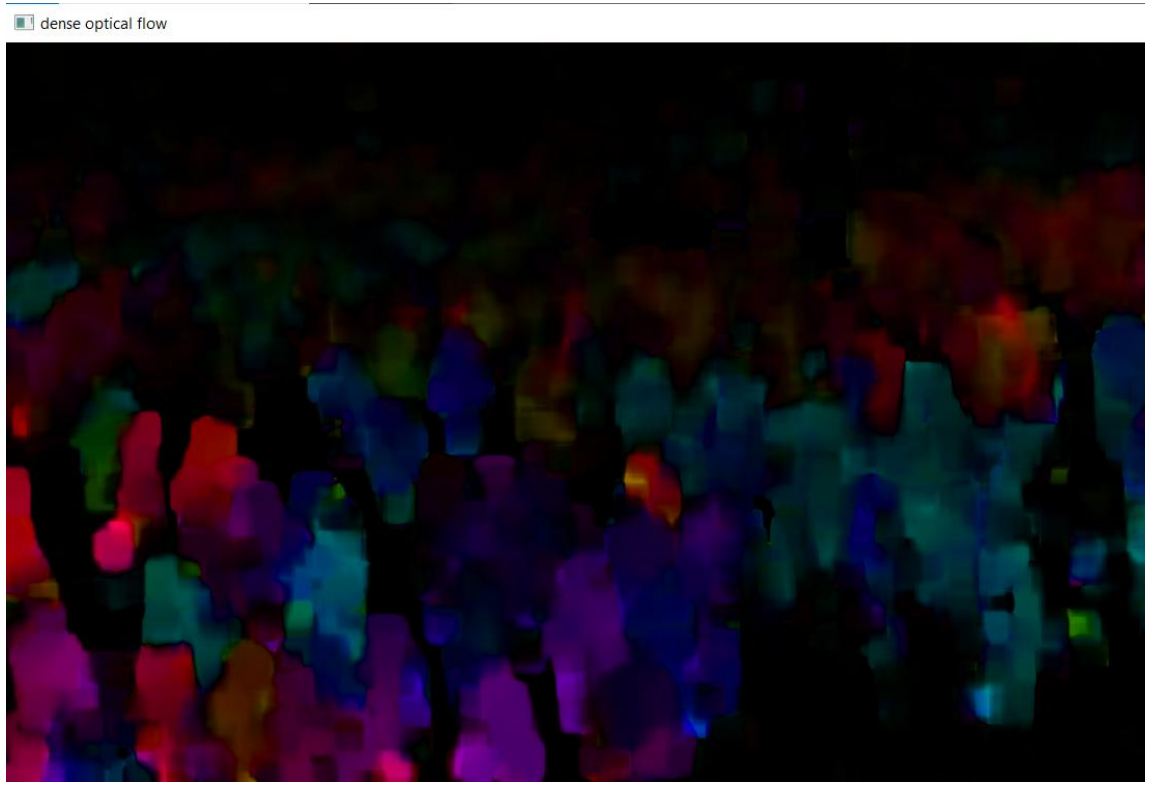
Trois étapes de traitement :

- ✓ Pré-filtrage ou lissage avec des filtres passe-bas pour obtenir la structure du signal d'intérêt et améliorer le SNR.
- ✓ Extraction de mesures de base, telles que les dérivées spatio-temporelles (pour mesurer les composantes de vitesse normale)
- ✓ Intégration de ces mesures pour produire un flux optique 2D en ajoutant des hypothèses sur la régularité .

Descripteurs

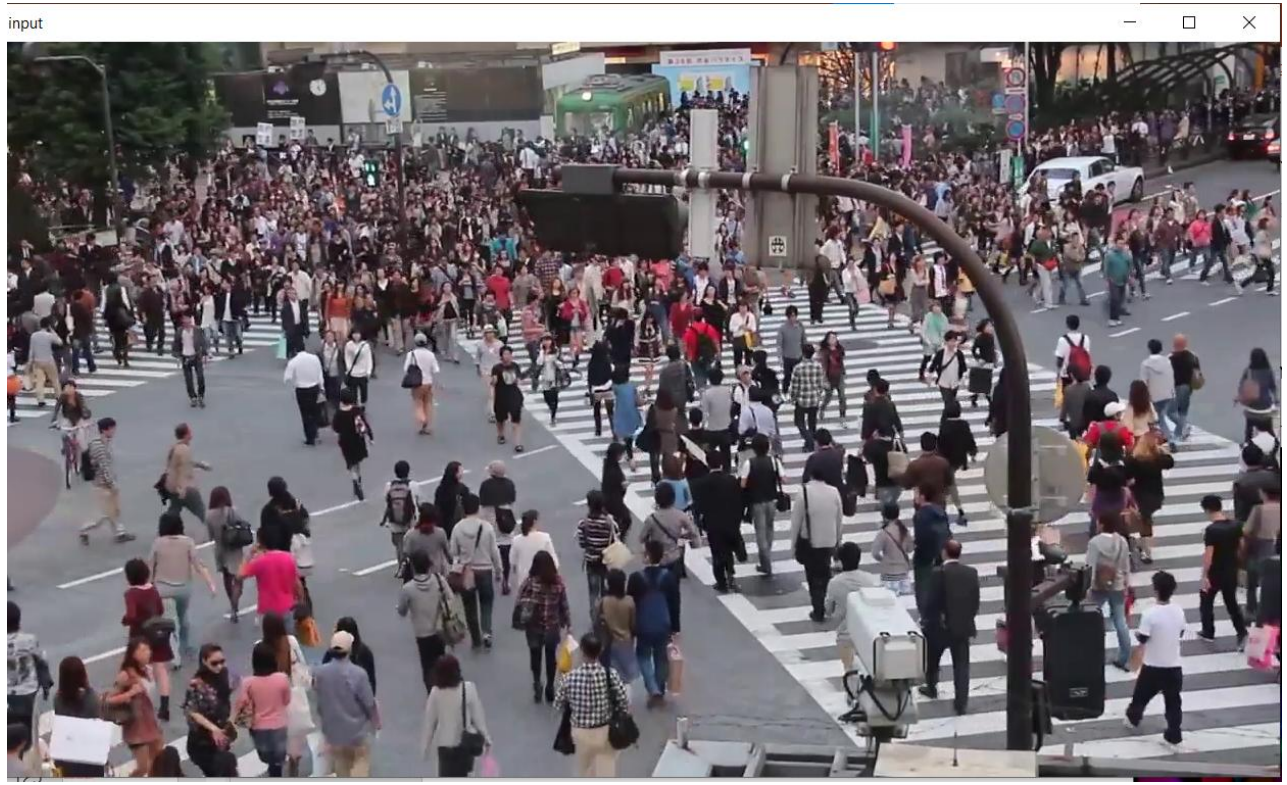
Méthode Farnback : une **méthode différentielle** utilisée pour l'estimation du flux optique **dense** . Cette méthode suppose que le flux est essentiellement constant dans un voisinage local du pixel considéré, et résout l'équation du flux optique pour tous les pixels de ce voisinage par la méthode des moindres carrés (méthode Lucas-Kanade).





Descripteurs

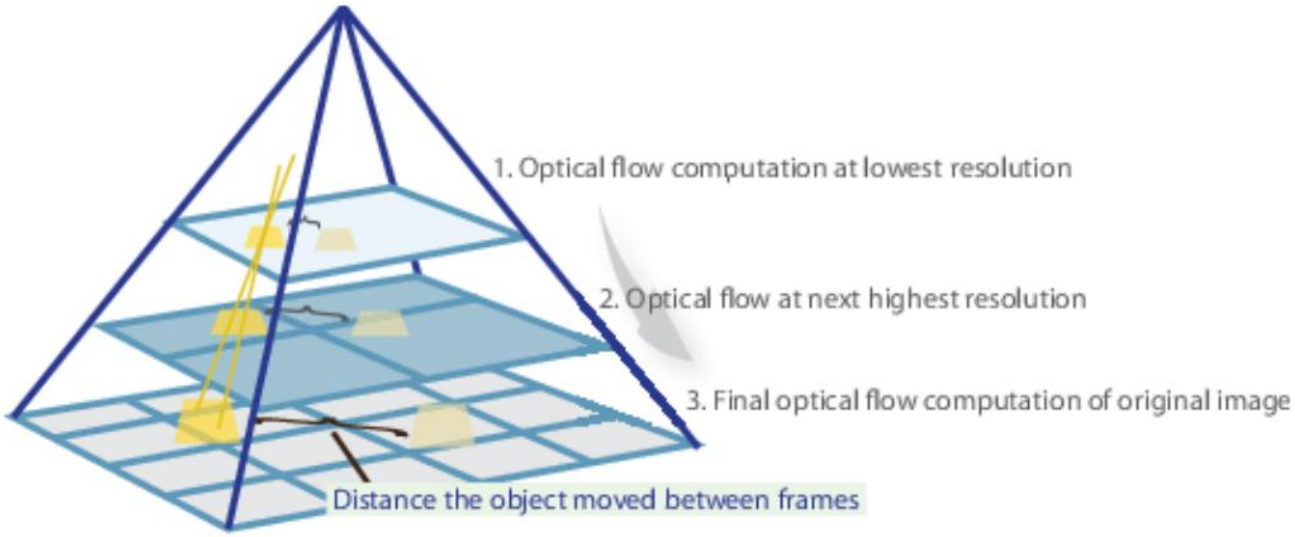
Pyramid Lucas Kanade : une **méthode différentielle** utilisée pour **les calculs clairsemés** estimation du flux optique. Il fonctionne uniquement sur **les points d'angle** détectés par l'algorithme Shi- Tomasi (une version modifiée de Harris Detector). Les caractéristiques extraites sont transmises dans la fonction de flux optique d'image en image pour garantir que les **mêmes points** sont **suivis** . Il existe différentes implémentations du flux optique sparse : méthode Horn- Schunck , méthode Buxton-Buxton, etc.



Descripteurs

Pyramid Lucas Kanade : une **méthode différentielle** utilisée pour **les calculs clairsemés** estimation du flux optique. Il fonctionne uniquement sur **les points d'angle** détectés par l'algorithme Shi- Tomasi (une version modifiée de Harris Detector). Les caractéristiques extraites sont transmises dans la fonction de flux optique d'image en image pour garantir que les **mêmes points** sont **suivis** . Il existe différentes implémentations du flux optique clairsemé : méthode Horn- Schunck , méthode Buxton Buxton, etc.

Une petite fenêtre 3x3 (quartier) autour des fonctionnalités détecté par Shi- Tomasi et supposer que les neuf points ont le même mouvement.



https://nanonets.com/blog/optical-flow/

- Pyramide Dense Lucas- Kanade
- Farneback
- PCAFlow
- Flux simple
- RLOF
- Flux profond
- DoubleTVL1

Descripteurs → Classement : exemple Classificateur K-NN

K-Nearest Neighbour est l'un des algorithmes d'apprentissage automatique **non paramétriques les plus simples** basés sur la technique d'apprentissage supervisé.

- Il stocke toutes les données disponibles et classe un nouveau point de données en fonction de la similarité. Cela signifie que lorsque de nouvelles données apparaissent, elles peuvent être facilement classées dans une catégorie en utilisant l'algorithme K-NN.
- L'algorithme K-NN peut être utilisé pour la régression ainsi que pour la classification, mais il est principalement utilisé pour les problèmes de classification.
- On l'appelle également algorithme d'apprentissage paresseux car il **n'apprend pas** de l'ensemble de formation. Lors de la phase de formation, il stocke simplement l'ensemble de données et lorsqu'il obtient de nouvelles données, il classe ces données dans une catégorie très **similaire** aux nouvelles données

Semestre : 2
Unité d'enseignement: UED 1.2
Matière 1 : Représentation des données Images et Vidéos
VHS : 22h30 (Cours : 1h30) Crédits : 1 Coefficient : 1

Chapitre 1. Introduction à la représentation d'images et de la vidéo (3 Semaines)

- ✓ Acquisition et formation d'une image, numérisation d'images.
- ✓ Espaces colorimétriques et transformations de couleur (RGB, HSV, YCrCb)
- ✓ Notions de résolution et quantification d'une image numérique (Taille, dpi, ppi , bpp , etc).
- ✓ Différents types d'images (Thermiques, Echos radar , satellites, images de capteurs sans fils, etc).
- ✓ Formats d'une image numérique (BMP, TIFF, JPG, GIF et PNG).
- ✓ Notions de la vidéo numérique, formats vidéo .
- ✓ Conversion des formats vidéo et extraction de trames.

Semestre : 2
Unité d'enseignement: UED 1.2
Matière 1 : Représentation des données Images et Vidéos
VHS : 22h30 (Cours : 1h30) Crédits : 1 Coefficient : 1

Chapitre 2. Traitement des images et des vidéos (4 semaines)

- Chaîne de traitement de l'image et de la vidéo

✓ *Méthodes de prétraitement : Amélioration de la qualité (manipulation d'histogramme, restauration et réduction du bruit, détection de contours)*

- Notions de post-traitements

Chapitre 3. Représentation de l'image et vidéo pour l'analyse (4 semaines)

✓ Représentation des images fixes : couleur, texture et forme.

✓ Descripteurs : couleurs, formes et textures .

- Descripteurs transformés (DCT et ondelettes)

- Descripteurs de mouvements

✓ Applications à l'analyse de l'image et de la vidéo

Semestre : 2
Unité d'enseignement: UED 1.2
Matière 1 : Représentation des données Images et Vidéos
VHS : 22h30 (Cours : 1h30) Crédits : 1 Coefficient : 1

Chapitre 4. Atelier de prise en main des outils de traitement (4 Semaines)

- ✓ Prise en main des outils de traitement d'images et de vidéo sous environnement C++ et OpenCV
- ✓ Lecture et écriture des fichiers images et vidéos
- ✓ Capture d'une séquence d'un fichier vidéo
- ✓ Représentation RVB , HSV, binaire, niveau de gris d'une image
- ✓ *Application des techniques de filtrage d'images*

Semestre : 2
Unité d'enseignement: UED 1.2
Matière 1 : Représentation des données Images et Vidéos
VHS : 22h30 (Cours : 1h30) Crédits : 1 Coefficient : 1

OpenCV est la bibliothèque OpenSource de référence en ce qui concerne la vision par ordinateur (ou Computer Vision en anglais). En 22 ans d'existence, elle a accumulé plus de 2500 algorithmes optimisés issus de la littérature scientifique tels que SIFT , les cascades de Haar , ou encore la transformée de Hough . Elle permet entre autre :

- L'ouverture, la modification, l'enregistrement et l'affichage des fichiers images et vidéos.
- L'extraction d'informations sur la répartition statistique des pixels dans une image (moyenne , variance , histogramme)
- La segmentation d'image (Seuillage d'Otsu , K-moyennes ,...).
- La localisation et l'extraction d'objets.
- L'extraction d'informations (Aire , périmètre , couleurs, ...) sur les objets contenus dans une image.
- La détection de contours (Filtre de Sobel , laplacien , Scharr , ...), de ligne ou de cercle (Transformée de Hough).
- Opérations d'image usuelles (redimensionnement, rotation, opérateurs morphologiques, filtrage, etc .....).
- Détection et suivi en temps réel d'objets (yeux, voiture, balle, ...) grâce à des algorithmes tels que les cascades de cheveux ou encore Mean -shift .
- La mise en œuvre d'algorithmes usuels du machine learning tels que par le perceptron multicouches ou encore les arbres de décisions.

Annexe

Module PIL

à partir de l'image d'importation PIL

A = Image.open (" chemin_fichier ") ouverture d'un fichier image

A .load () force le chargement de l'image en mémoire

A .mode mode de l'image : 'L' en niveaux de gris, 'RGB' en couleurs

A .format format de l'image

A .size taille de l'image sous la forme (Largeur,Hauteur)

A .save (" chemin_fichier ") sauvegarde d'une image dans un fichier

A = Image.new ('RGB',(L,H)) création d'une image 'RGB' de dimensions (Largeur,Hauteur)

r,g,b = A.split () récupère les composantes (r,g,b) de l'image

B = A .getpixel ((x,y)) lecture du pixel de coordonnées (x,y)

A .putpixel ((x,y),(r,g,b)) écriture de la valeur (r,g,b) dans le pixel (x,y

Annexe

Bibliothèques à installer

pépin installer opencv -python

Liens

https://docs.opencv.org/4.x/d7/da8/tutorial_table_of_content_imgproc.html

https://docs.opencv.org/4.x/modules.html

https://dontrepeatyourself.org/post/how-to-read-and-write-videos-with-opencv

https://www.analyticsvidhya.com/blog/2021/04/top-python-libraries-for-image-processing-in-2021/

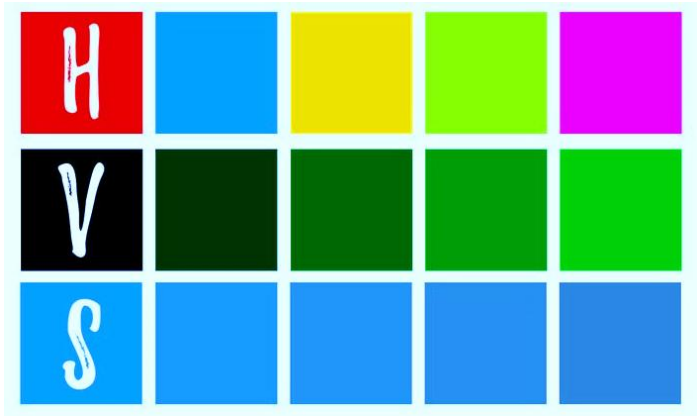
Annexe

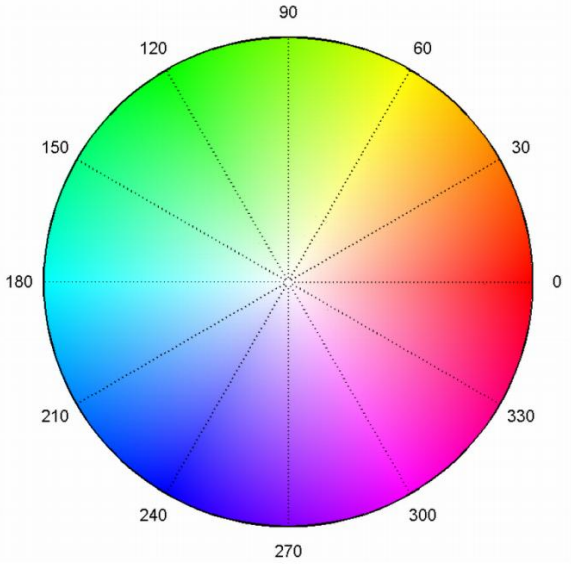
RVB vers HSV

1. R' = R/255

G' = G/255

B' = B/255*





2. Cmax = max(R', G', B')

Cmin = min(R', G', B')

Δ = Cmax - Cmin

3 . $V = CmaxS = \begin{cases} \frac{\Delta}{Cmax} & si \quad Cmax \neq 0 \\ 0 & si \quad Cmax = 0 \end{cases}$ $H = \begin{cases} 60^0 \frac{G'-B'}{\Delta} + 360^0 (mod 360^0) & si \text{Cmax} = R' \\ 60^0 \frac{B'-R'}{\Delta} + 120^0 & si \text{Cmax} = G' \\ 60^0 \frac{R'-G'}{\Delta} + 240^0 & si \text{Cmax} = B' \\ 0 & si \text{Cmax} = Cmin \end{cases}$

RVB vers YUV

Y = 0,299 R + 0,587 G + 0,114 B U'= (BY)*0,565 V'= (RY)*0,713

Annexe

Caractéristique de la texture	Description	Formule
1. **L'énergie**(ou moment angulaire second)	L'énergie mesure l'homogénéité de l'image. Plus cette valeur est faible, moins l'image est uniforme et l'image contient beaucoup de variation de couleur. (i , j) : coordonnées dans la matrice. P(i , j) : valeurs normalisées de la matrice. Valeur dans l'intervalle [0,1].	$f1 = \sum_i \sum_j P(i,j)^2$
2. **Le contraste** ou inertie)	L'inertie mesure les variations locales des couleurs. Si ces variations sont importantes, alors le contraste sera élevé. Ce paramètre est fortement non corrélé à l'énergie. Valeur dans l'intervalle $[0,(Ng-1)^2]$.	$f2 = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (i-j)^2 P(i,j)$
3. **Corrélation**	Où μ_x, μ_y, σ_x et σ_y sont respectivement les moyennes et les écarts type de p_x et p_y. Ce paramètre permet de déterminer si certaines colonnes de la matrice sont égales, c'est-à-dire s'il existe des dépendances linéaires dans l'image. En effet, plus les valeurs sont uniformément distribuées dans la matrice de cooccurrences et plus la corrélation est importante. Valeur dans l'intervalle [-1,1]. La corrélation n'est corrélée ni à l'énergie, ni à l'entropie.	$f3 = \frac{\sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (ij) P(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
4. **La variance**	La variance mesure la dispersion des valeurs autour de la moyenne. Plus ce paramètre est élevé et plus importants sont les écarts entre les valeurs et la moyenne. Elle représente l'hétérogénéité de la texture.	$f4 = \sum_i \sum_j (i-\mu)^2 P(i,j)$
5. **Moment différentiel inverse (Homogénéité)**	Ce paramètre mesure l'homogénéité de l'image, plus l'image contient de régions homogène, plus il est élevé. Valeur dans l'intervalle [0,1].	$f5 = \sum_i \sum_j \frac{P(i,j)}{1+(i-j)^2}$
6. **Moyenne des sommes**		$f6 = \sum_{i=1}^{2Ng} i.P_{x+y}(i)$
7. **Variance des sommes**		$f7 = \sum_{i=1}^{2Ng} (i-f8)^2.P_{x+y}(i)$
8. **Entropie des sommes**		$f8 = \sum_{i=1}^{2Ng} P_{x+y}(i).\log(P_{x+y}(i))$
9. **Entropie**	Ce paramètre mesure le désordre dans l'image.sa valeur sera élevé en cas de texture complètement aléatoire (sans structure apparente). Lorsque les valeurs de la matrice de cooccurrences sont presque toutes égales, l'entropie est élevée.	$f9 = -\sum_{i=1}^{Ng} \sum_{j=1}^{Ng} P(i,j).\log(P(i,j))$
10. **Variance des différences**		$f10 = \sum_{i=1}^{Ng} (\mu - \mu_{x-y})^2.P_{x-y}(i)$
11. **Entropie des différences**	Valeur dans l'intervalle [0,2.log(Ng-1)].	$f11 = \sum_{i=1}^{Ng} P_{x-y}(i) \log P_{x-y}(i)$
12. **Information sur la corrélation 1**	Soient : • Px, Py les probabilités selon x et y, et HX, HY leurs entropie respective. • $HXY = -\sum_{i=1}^{Ng} \sum_{j=1}^{Ng} P(i,j) \log(P(i,j))$ • $HXY1 = -\sum_{i=1}^{Ng} \sum_{j=1}^{Ng} P(i,j) \log(P_x(i)P_y(i))$ • $HXY2 = -\sum_{i=1}^{Ng} \sum_{j=1}^{Ng} P_x(i)P_y(j) \log(P_x(i)P_y(i))$	$f12 = \frac{(HXY - HXY1)}{Max(HX,HY)}$
13. **Information sur la corrélation 2**		$f13 = \sqrt{1 - e^{-2(HXY2-HXY)}}$
14. **Coefficient de corrélation maximal**	Soient la matrice : $Q(i;j) = \sum_k \frac{P(i,k)P(j,k)}{P_x(i)P_y(k)}$ $V = 2^{eme}$ Plus grande valeur propre de Q	$f14 = \sqrt{V}$