

Bases de l'IA

Traitement d'image

Elena CABRIO

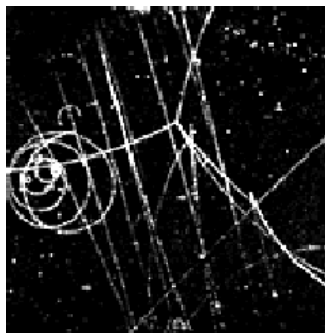
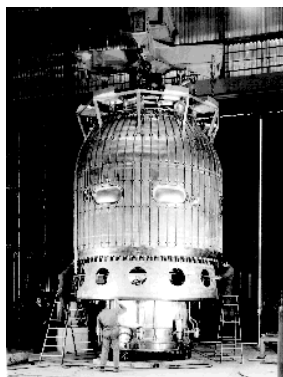
elena.cabrio@univ-cotedazur.fr

Plan pour cette séance

- Traitement d'image: intro
- Vision par ordinateur: Classification des images

Traitement d'image: un peu d'histoire

- Domaine de recherche relativement récent : les premiers travaux datent des années 60. Ils concernent quelques applications spécifiques :
 - en physique des particules : analyse automatique de milliers d'images de trajectoires de particules
 - Reconnaissance automatique de caractères imprimés



Bases de l'IA

NAME		RAMBARDE PASS. GCH.
N° 91- 203 217		SH NO - 1
10183		1018286 C
10183		NAME CAPACITOR HOLDER
10183		20 790 337

Traitement d'image: un peu d'histoire

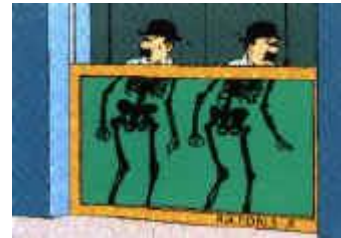
- Premier traitements :
 - analyse automatique des images satellitaires et aériennes
 - traitement d'images médicales : pour faire une première analyse ou sélection d'images pour les médecins mais peu efficace et enfin développement de la robotique médicale dans un premier temps (micro-caméra).



Images aériennes



Images satellitaires



Images médicales

MRI

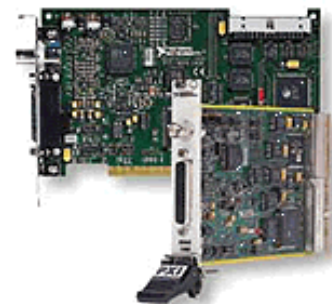


Traitement d'image: un peu d'histoire

- Premiers types de traitement de l'image :
 - restauration d'images afin d'éliminer les défauts causés par les dispositifs optiques (aberrations géométriques ou chromatiques)
 - stockage des images et traitement en temps réel => algorithmes de compression d'images dans les années 80 et développement de cartes de traitement d'images pour un traitement et un affichage rapides des images.



bases de l'IA



L'acquisition de l'image est très facile



KINECT
for XBOX 360

Traitement d'image: définition

- Ensemble d'algorithmes pour :
 - l'amélioration de la qualité des images (débruitage, débrouillage...)

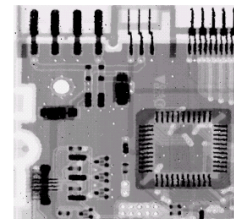
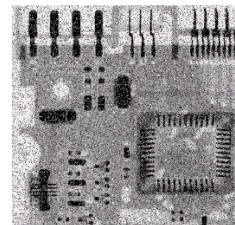


Réglage du contraste



Credit: M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester: Image inpainting. SIGGRAPH 2000

Restauration d'images



Suppression du bruit

Eye is the final judge !

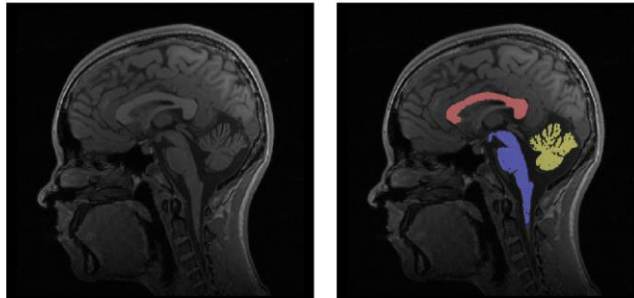
Traitement d'images: definition

- Ensemble d'algorithmes pour :
 - soit l'extraction automatique et l'interprétation d'informations (extraction de contours, segmentation d'objets, reconnaissance de caractères d'imprimerie, identification de personnes...)

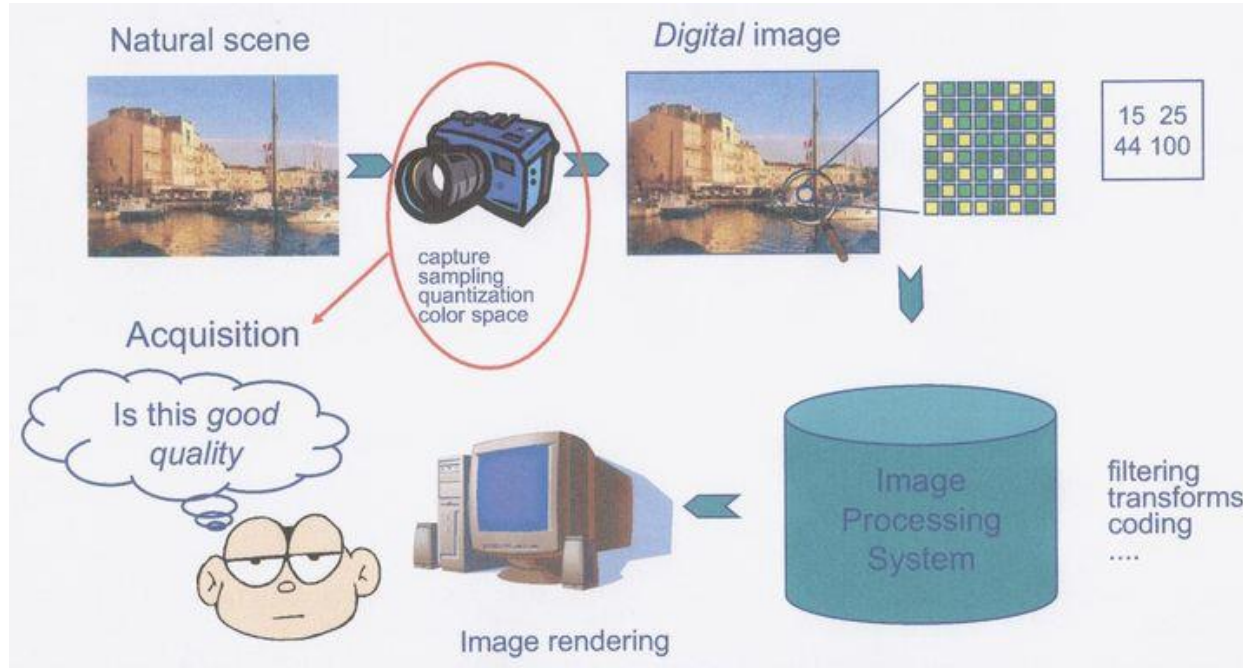


Extraction des bords

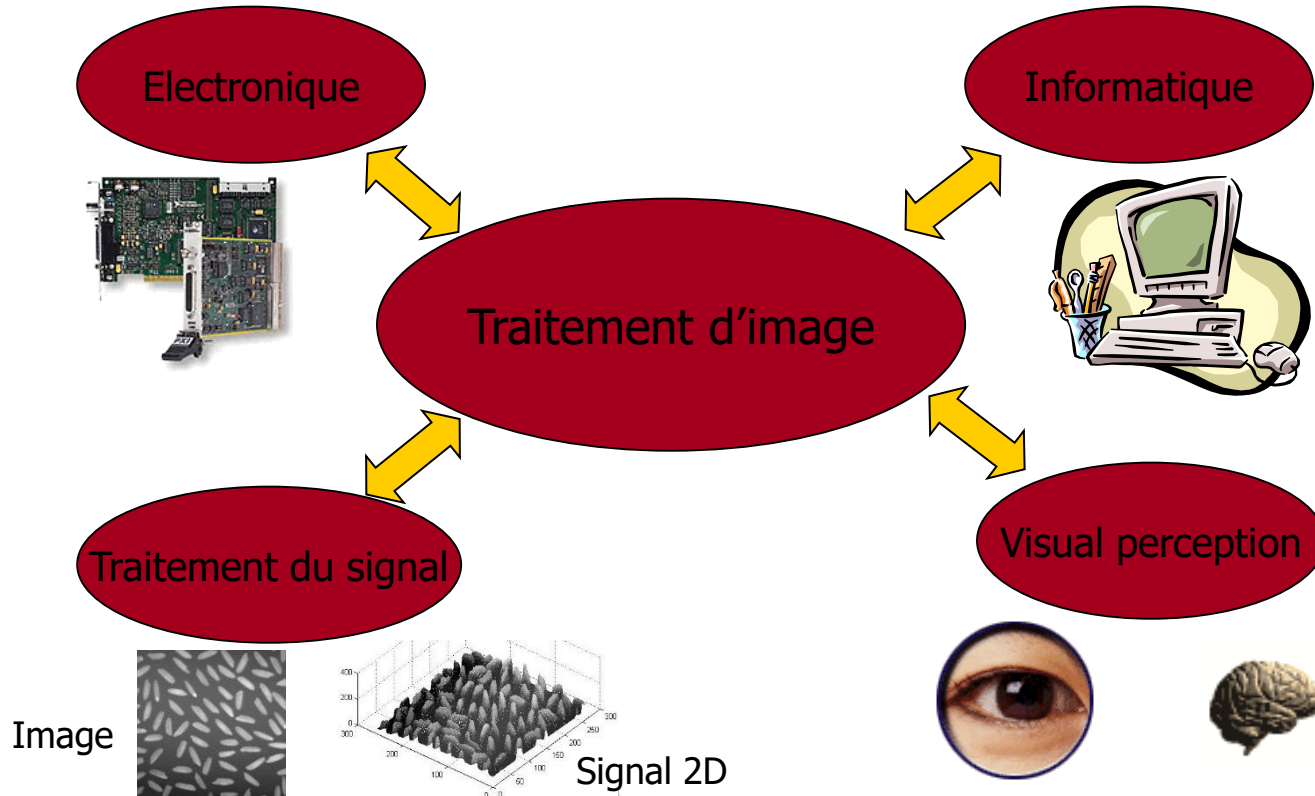
Segmentation des régions



Systeme pour le traitement d'images: définition



Traitement d'image: domaine multi-disciplinaire



Applications



Robotique

Interaction Homme Machine

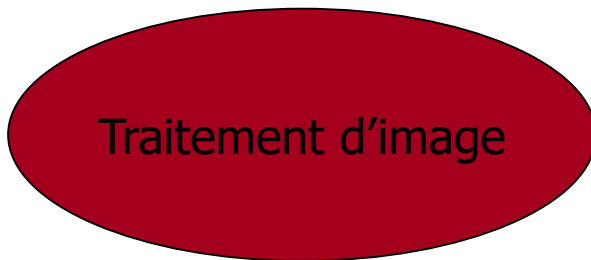
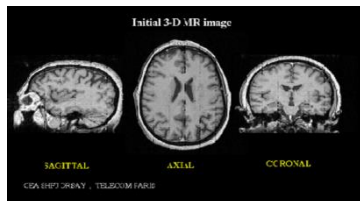
Compression

Smart rooms

Realite augmentée



Imagerie medicale



Navigation autonome

Bases de l'IA



Video surveillance

Assistance

Biometrie

Meteorologie

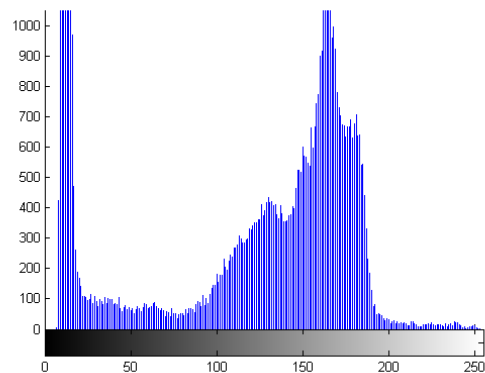


Astronomie

Contrôle de la qualité

27/03/2023

Images numériques



Acquisition d' images numériques

Image analogique



Image numérique

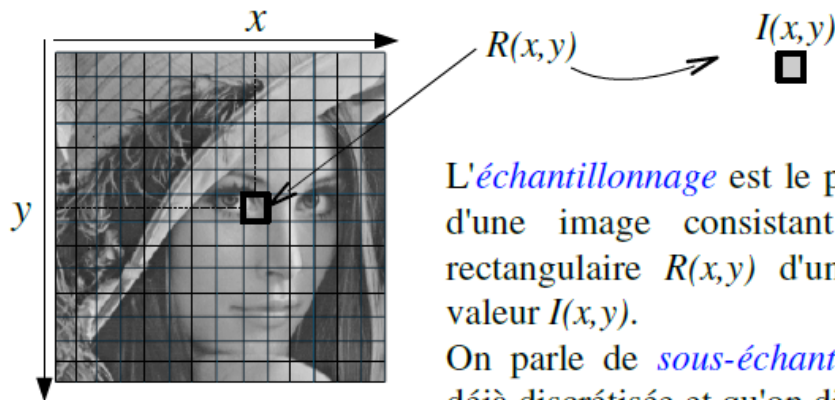


Conversion analogique-numérique

Stockage

Capteurs CCD 1152 * 864 photodiodes
Chaque capteur reçoit de la lumière
et effectue une conversion de la lumière en tension

ADC: Echantillonnage et quantification



L'*échantillonnage* est le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire $R(x,y)$ d'une image continue une unique valeur $I(x,y)$.

On parle de *sous-échantillonnage* lorsque l'image est déjà discrétisée et qu'on diminue le nombre d'échantillons.

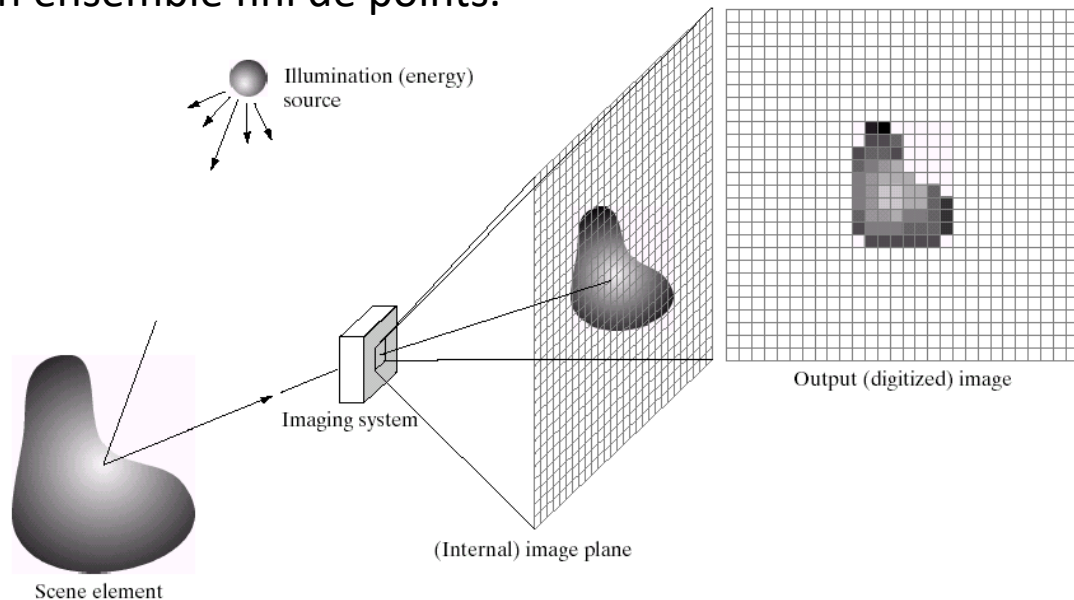
La *quantification* désigne la limitation du nombre de valeurs différentes que peut prendre $I(x,y)$.



Une *image numérique* est une image *échantillonnée et quantifiée*.

Echantillonnage

Une image numérique est une représentation d'une image bidimensionnelle sous la forme d'un ensemble fini de points.

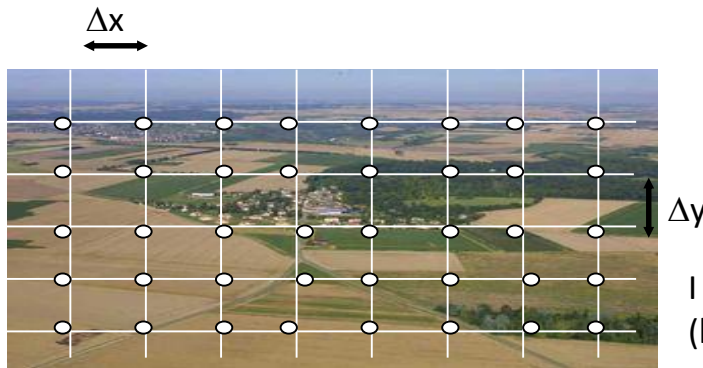


Il s'agit d'une approximation du monde réel

Bases de l'IA

27/03/2023

Un bon échantillonnage?



$$I : N_1 \times N_2 \rightarrow \mathbb{N}$$
$$(i_1, i_2) \mapsto I(i_1, i_2)$$

I fonction de luminance qui est une fonction numérique
(luminance est une grandeur correspondant à la
sensation visuelle de luminosité d'une surface)

Échantillonnage spatial, discrétisation de l'espace => Image numérique = tableau de pixels

Un bon échantillonnage ?

Intuitivement, c'est un échantillonnage qui fournit une bonne qualité visuelle, c'est-à-dire une bonne résolution et une image sans artefacts. Théoriquement, théorème de Shannon

$$\frac{1}{\Delta x} = f_{e_x} \geq 2F_{x_{\max}} \quad \text{et} \quad \frac{1}{\Delta y} = f_{e_y} \geq 2F_{y_{\max}}$$

Bases de l'IA

Echantillonnage

Exemple de distorsion d'image



Proper sampling of image



Under sampling of image – note the Moire patterns due to image aliasing

Quantization

Image : signal analogique spatial 2D

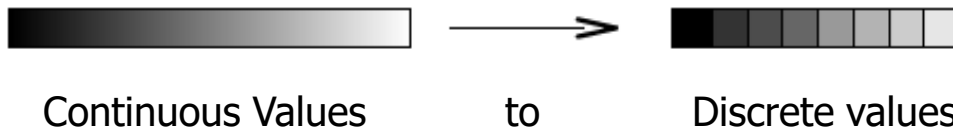
$(x,y) \longrightarrow f(x,y)$ avec f la fonction de luminance, f est continue
 $f(x,y)$ représente la quantité de lumière reçue au point (x,y)

Quantification : processus qui transforme la fonction de luminance à valeurs continues en une image à valeurs discrètes.

1ère étape : choix des valeurs possibles

2ème étape : correspondance entre les valeurs continues et discrètes au moyen de transformations non linéaires et non réversibles (arrondis...)

Perte d'information



Quantization

Quel est le meilleur ensemble de valeurs discrètes ?

- 8 bits est la valeur la plus courante \Rightarrow 256 valeurs possibles soit 256 niveaux de gris : 0 pour le noir pur et 255 pour le blanc pur \Rightarrow image en niveaux de gris
- 1 bit \Rightarrow 2 valeurs possibles seulement \Rightarrow image en noir et blanc



8 bits
256 levels



4 bits
16 levels



2 bits
4 levels



1 bit
2 levels

Échantillonnage et quantification

Résolution...

...spatiale :

Échantillonnage



256x256



128x128



64x64



32x32

Dépend du nombre de pixels différents
acquis dans l'image

...tonale :

Quantification

Dépend du nombre de bits alloués
à chaque valeur de pixel



6 bits



4 bits



3 bits

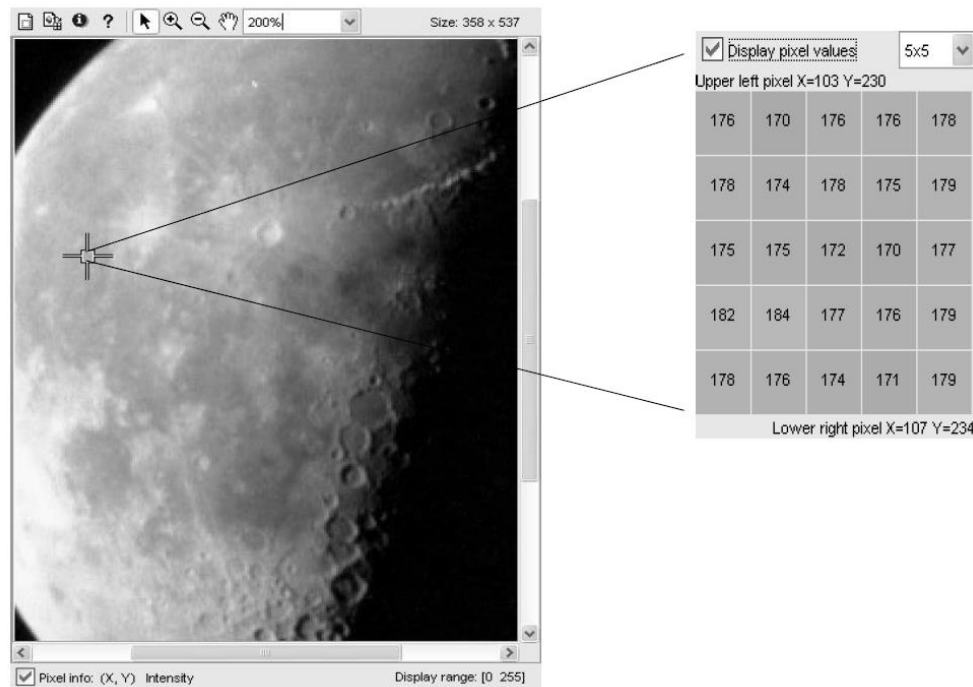


2 bits



1 bit

Une image : une matrice de valeurs de niveaux de gris

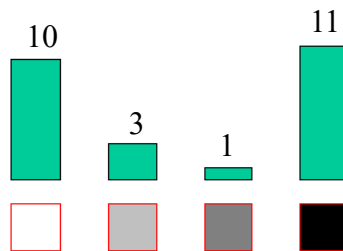
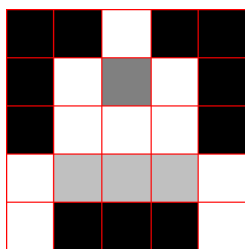


Représentation spatiale

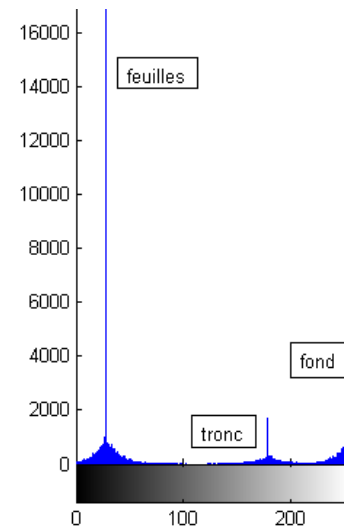
Histogramme H : donne pour chaque niveau de gris i (256 niveaux) le nombre de pixels de l'image ayant ce niveau de gris i . En divisant ce nombre par le nombre total de pixels dans l'image, on obtient la probabilité de chaque niveau de gris i dans l'image.

$$H(i) = \text{card} \{ p \in I : I(p) = i \}$$

$$\text{proba}(i) = \frac{H(i)}{\text{card}(I)}$$

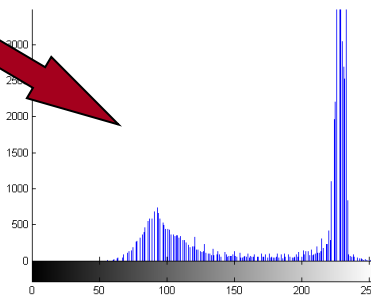
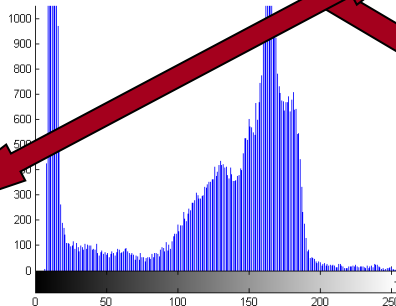
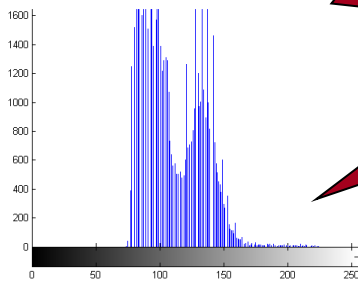


Bases de l'IA

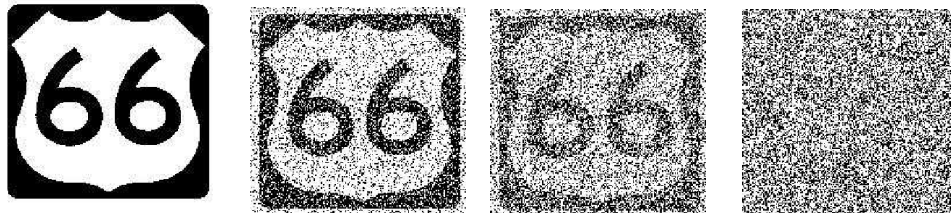


Représentation spatiale

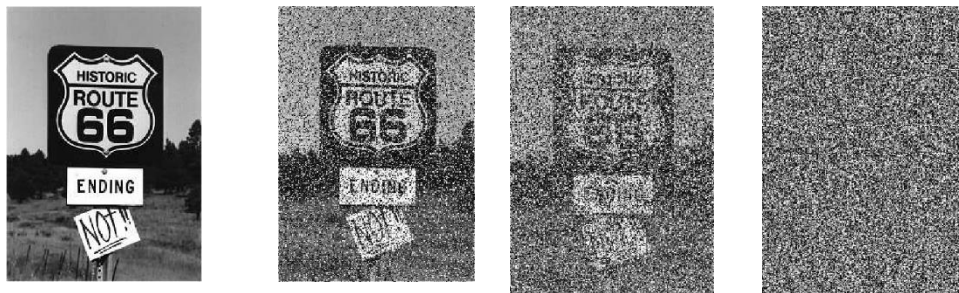
Images et histogrammes associés ?



Représentation spatiale



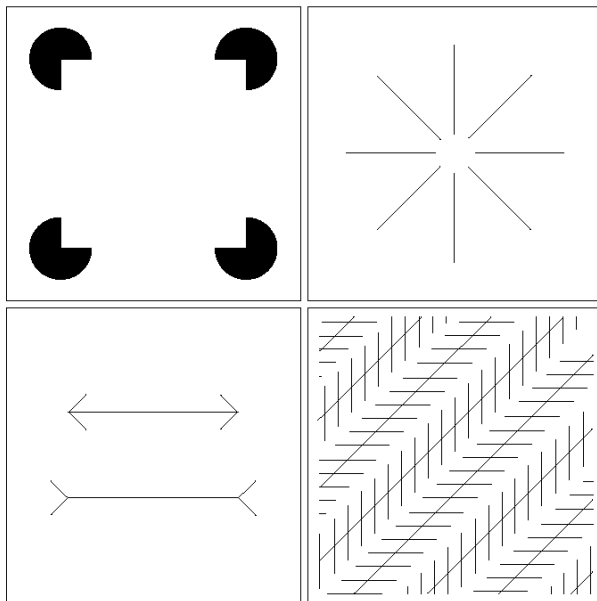
Dans toutes les images, le même nombre de pixels noirs et blancs.



Chaque niveau de gris est présent dans la même quantité

Le voisinage : une notion importante

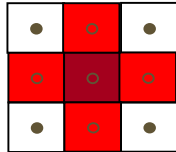
Chaque pixel est entouré d'autres pixels, ce qui influe sur la visualisation de l'image..



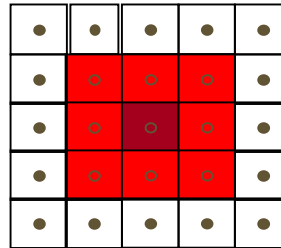
Illusions d'optique

Exemples de voisinage

Les quartiers carrés sont plus faciles à définir en raison de la grille d'échantillonnage carrée.



Voisins de premier ordre :
l'ensemble des pixels les
plus proches



Voisins de second ordre



pixel central



pixel voisin

Le voisinage : une notion importante



Tous les carrés intérieurs ont la même intensité mais apparaissent plus foncés lorsque le carré extérieur (zone environnante) s'éclaircit.

=> Plusieurs méthodes de traitement d'images agissent sur le pixel courant ET son voisinage.

Fréquence d'images



(a) Image



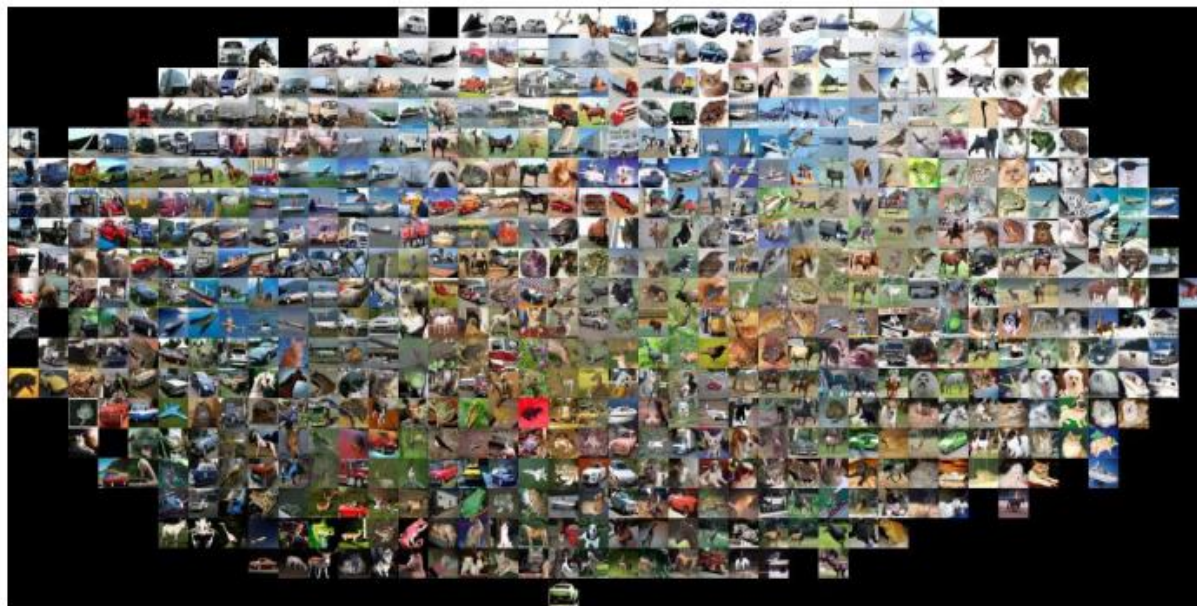
(b) Basses fréquences



(c) Hautes fréquences

Vision par ordinateur

Classification des images



Stanford CS231N –

<http://cs231n.stanford.edu/>

Slides from Fei-Fei Li, Justin Johnson, Serena Yeung
<http://vision.stanford.edu/teaching/cs231n/>

Bases de l'IA

27/03/2023

Qu'est-ce qui compte dans la reconnaissance ?

- **Méthodes d'apprentissage automatique** (par ex. classification linéaire, apprentissage profond)
- **Représentation** (par exemple, SIFT, HoG, apprentissage profond) caractéristiques apprises en profondeur)
- **Données** (par exemple, PASCAL, ImageNet, COCO)

Classification des images :

Une tâche essentielle dans le domaine de la vision par ordinateur

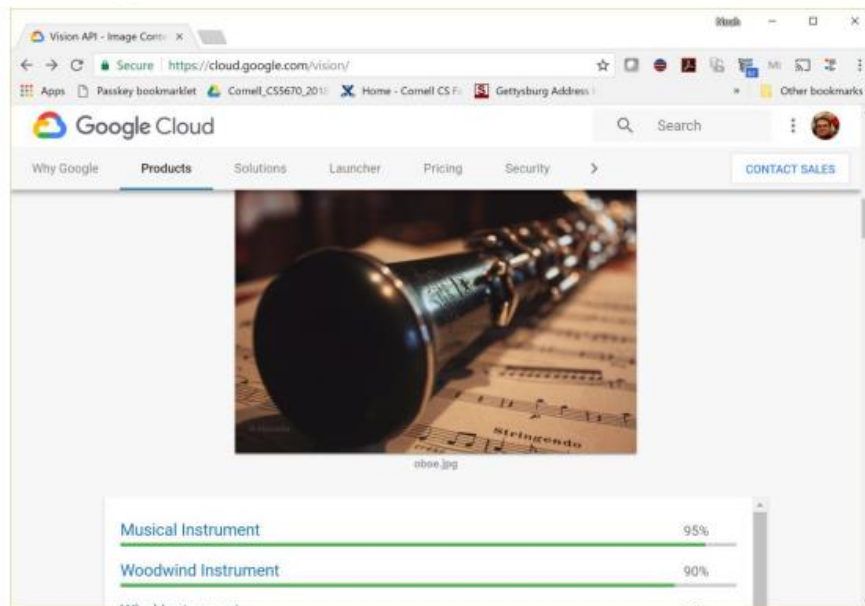
- Supposons un ensemble donné d'étiquettes discrètes, par exemple {chat, chien, vache, pomme, tomate, camion, ...}

$f(\text{apple}) = \text{"apple"}$

$f(\text{tomato}) = \text{"tomato"}$

$f(\text{cow}) = \text{"cow"}$

Démo de classification d'images



<https://cloud.google.com/vision/>

See also:

<https://aws.amazon.com/rekognition/>

<https://www.clarifai.com/>

<https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>

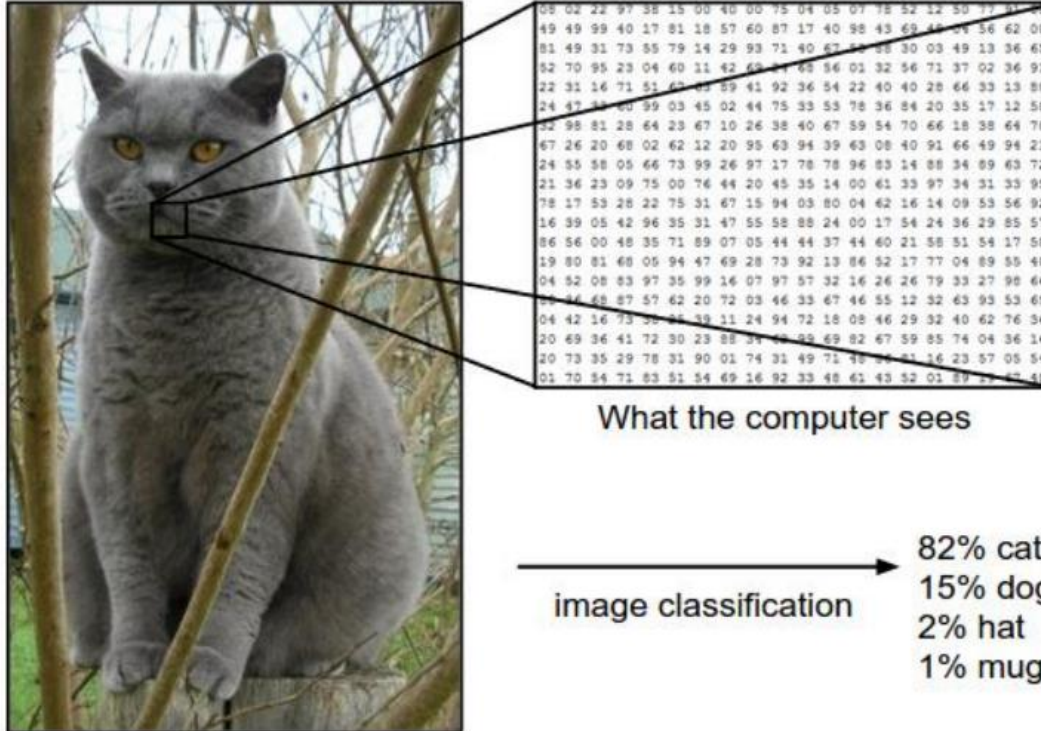
Classification des images



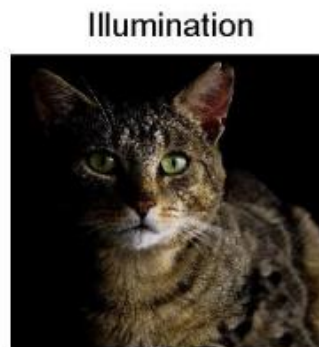
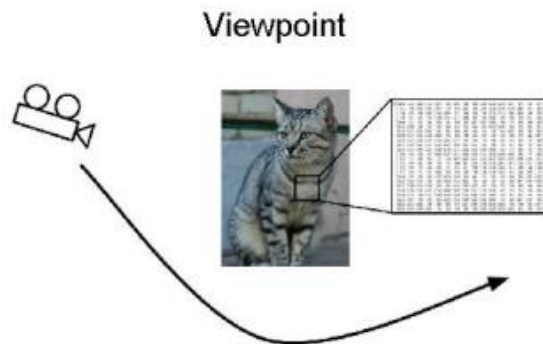
→ cat

- Supposons un ensemble donné d'étiquettes discrètes, par exemple {chat, chien, vache, pomme, tomate, camion, ...}

Classification des images: Problème



Les défis de la reconnaissance



This image is [CC0 1.0](#) public domain



This image by [Unkerto Saksoni](#) is licensed under [CC-BY 2.0](#)



This image by [jansson](#) is licensed under [CC-BY 2.0](#)



This image is [CC0 1.0](#) public domain



This image is [CC0 1.0](#) public domain

Un classificateur d'images

```
def classify_image(image):  
    # Some magic here?  
    return class_label
```

Contrairement par exemple au tri d'une liste de nombres, il n'y a pas de moyen évident de coder en dur l'algorithme de reconnaissance d'un chat ou d'autres classes.

Approche basée sur les données

- Collecte d'une base de données d'images avec des étiquettes
- Utiliser l'apprentissage automatique pour entraîner un classificateur d'images
- Évaluer le classificateur sur des images de test

Example training set



Slides from Andrej Karpathy and Fei-Fei Li
<http://vision.stanford.edu/teaching/cs231n/>

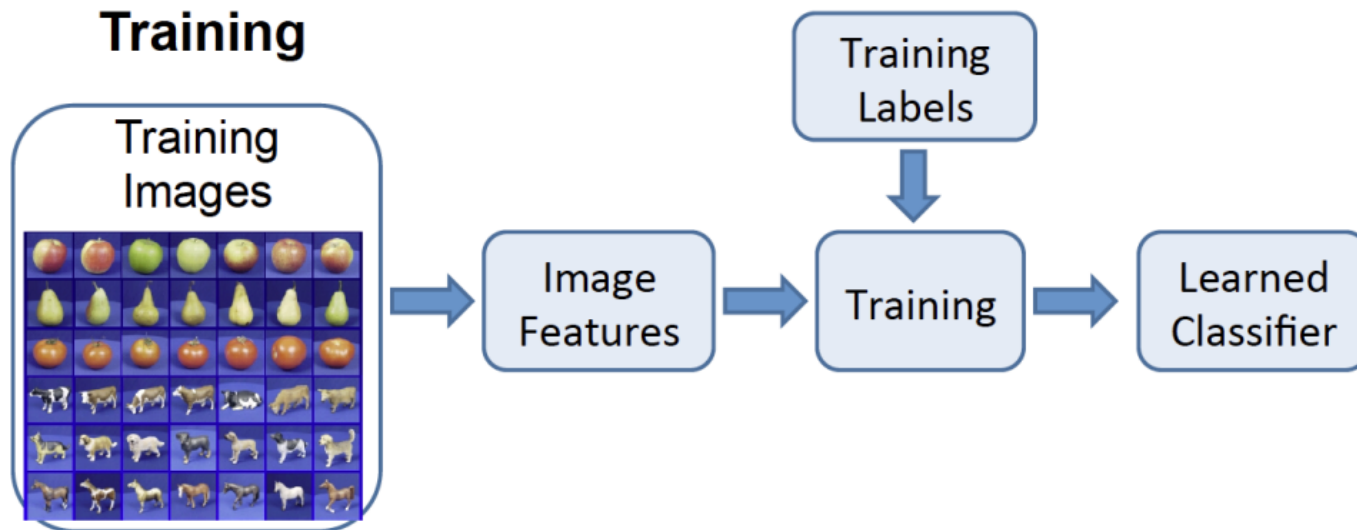
27/03/2023

Approche basée sur les données

- Collecte d'une base de données d'images avec des étiquettes
- Utiliser l'apprentissage automatique pour entraîner un classificateur d'images
- Évaluer le classificateur sur des images de test

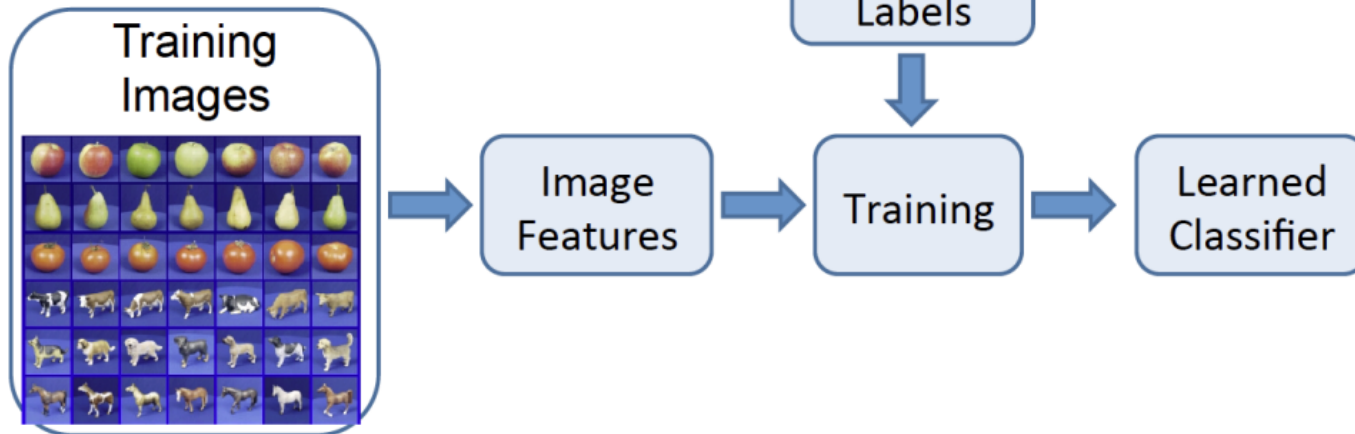
```
def train(train_images, train_labels):  
    # build a model of images -> labels  
  
def predict(image):  
    # evaluate the model on the image  
    return class_label
```

Entraîner un classificateur d'images



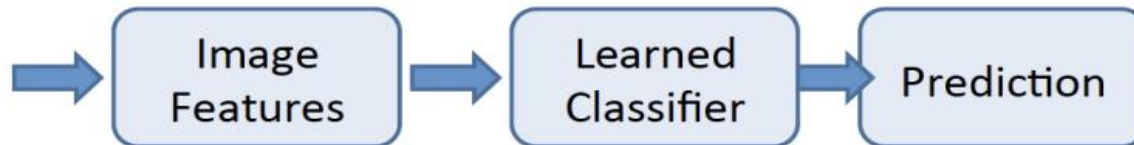
Tester un classificateur d'images

Training



Testing

Test Image



Classificateurs

- Voisin le plus proche
- kNN ("k-Nearest Neighbors")
- Classificateur linéaire
- Réseau de neurones
- Réseau neuronal profond
- - ...

Classificateur du plus proche voisin (NN)

- Entraînement
 - Se souvenir de toutes les images d'entraînement et de leurs étiquettes
- Prédire
 - Trouver l'image d'apprentissage la plus proche (la plus similaire)
 - Prédire que son étiquette est la vraie étiquette

10 labels

10,000 test images.



Résultats du CIFAR-10 et du NN



Example dataset: **CIFAR-10**

10 labels

50,000 training images, each image is tiny: 32x32

10,000 test images.

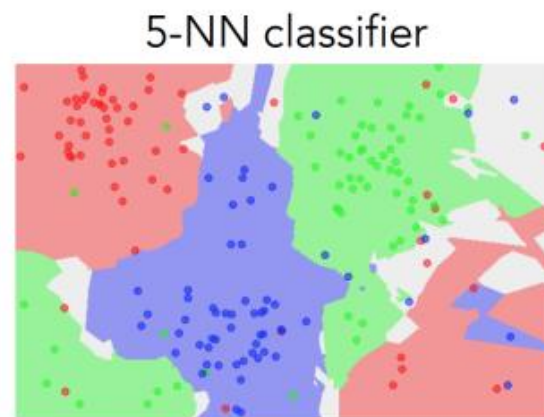
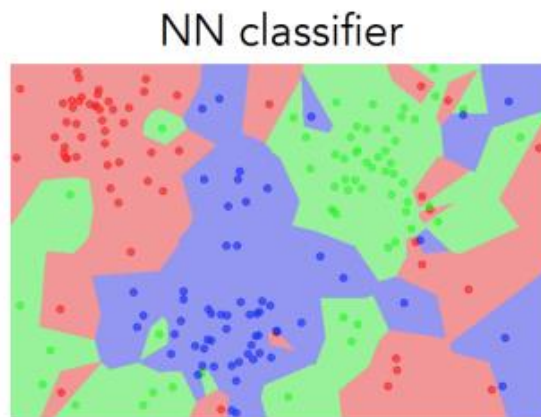
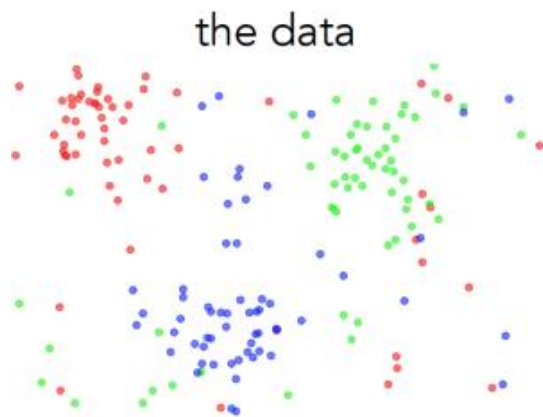


For every test image (first column),
examples of nearest neighbors in rows



k-voisin le plus proche

- Trouver les k points les plus proches à partir des données d'apprentissage
- Prendre le vote majoritaire des K points les plus proches



À quoi cela ressemble-t-il ?



À quoi cela ressemble-t-il ?



Comment trouver l'image de training la plus similaire ? Quelle métrique de distance ?

L1 distance:

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

Where I_1 denotes image 1,
and p denotes each pixel

test image					training image					pixel-wise absolute value differences				
56	32	10	18		10	20	24	17		46	12	14	1	
90	23	128	133		8	10	89	100		82	13	39	33	
24	26	178	200	-	12	16	178	170	=	12	10	0	30	→ 456
2	0	255	220		4	32	233	112		2	32	22	108	

Comment trouver l'image de training la plus similaire ? Quelle métrique de distance ?

L1 distance:

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

Where I_1 denotes image 1,
and p denotes each pixel

test image					training image					pixel-wise absolute value differences				
56	32	10	18		10	20	24	17		46	12	14	1	
90	23	128	133		8	10	89	100		82	13	39	33	
24	26	178	200	-	12	16	178	170	=	12	10	0	30	→ 456
2	0	255	220		4	32	233	112		2	32	22	108	

Choix de la métrique de distance

- Hyperparamètres

L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

K-voisins les plus proches : métriques de distance

- Hyperparamètres

L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



K = 1

L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



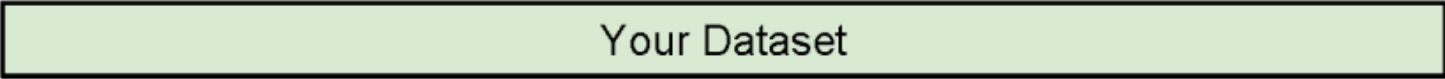
K = 1

Hyperparamètres

- Quelle est la meilleure distance à utiliser ?
- Quelle est la meilleure valeur de k à utiliser ?
- Il s'agit d'hyperparamètres : des choix concernant l'algorithme que nous fixons
- Comment les fixer ?
 - Une option : les essayer tous et voir ce qui fonctionne le mieux.

Définition des hyperparamètres

Idea #1: Choose hyperparameters
that work best on the data



Your Dataset

Définition des hyperparamètres

Idea #1: Choose hyperparameters that work best on the data

BAD: $K = 1$ always works perfectly on training data



Your Dataset

Définition des hyperparamètres

Idea #1: Choose hyperparameters that work best on the data

BAD: $K = 1$ always works perfectly on training data



Your Dataset

Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data



train

test

Définition des hyperparamètres

Idea #1: Choose hyperparameters that work best on the data

BAD: $K = 1$ always works perfectly on training data



Your Dataset

Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

BAD: No idea how algorithm will perform on new data



train

test

Définition des hyperparamètres

Idea #1: Choose hyperparameters that work best on the data

BAD: $K = 1$ always works perfectly on training data

Your Dataset

Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

BAD: No idea how algorithm will perform on new data

train

test

Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

Better!

train

validation

test

Définition des hyperparamètres

Your Dataset

Idea #4: Cross-Validation: Split data into **folds**,
try each fold as validation and average the results

fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

Utile pour les petits ensembles de données, mais peu utilisé dans l'apprentissage profond.