

La recherche bibliographique (documentation) est indispensable.

Couleur noire (difficulté moyenne), **Bleu** (difficulté moyenne+), **Rouge** (nouveauté).

A) Projets personnels : s'il y en a

B) Apport de l'intelligence artificielle et Deep learning :

- 1) **Débruitage par Deep learning** (cf. Description du projet + cf article 2021_Denoising1_DL&CT_Wu_TRPMS)
- 2) **Segmentation des images médicales et suivi de l'évolution (labellisation)** (cf. Description du projet + cf. article 2020_Covid&AI)
- 3) **Classification et Segmentation jointe des images médicales par Deep learning** (cf. article 2022_JointClassification&Segmentation)
- 4) **Reconnaissance des empreintes digitales et Deep learning** (segmentation, squelettisation, détection de minuties, authentification, identification). (cf. Description du projet + cf article 2018_DeepMasterPrint)
- 5) **Analyse de rétines pour la Biométrie et Deep learning** (segmentation, squelettisation, détection de minuties, puis authentification et identification). (cf. Description du projet)
- 6) **Analyse de rétines pour l'imagerie médicale et Deep learning** (segmentation, squelettisation, détection de minuties, puis classification des pathologies : DMLA, Diabète). (cf. Description du projet)
- 7) **Segmentation de rétines par CNN 2016** (cf. article 2016)
- 8) **Segmentation de rétines par Deep Neural Ensemble 2016** (cf. article 2016)
- 9) **Segmentation de rétines par supervising Deep Neural Networks... 2017** (cf. article 2017)
- 10) **Segmentation de rétines par Demixing Network... 2020** (cf. article 2020)

- 11) Diabetic Retinopathy Diagnosis based on CNN...2020 (cf. article 2020)
- 12) The Segment Anything Model (SAM) 2023 (cf. article 2023_Segment Any Thing)
- 13) Current and Emerging Trends in Medical Image Segmentation with Deep Learning 2023 (cf. article 2023_Current_and_Emerging)
- 14) Predicting bounding boxes by neural network (cf. Description du projet)

C) Histogramme :

- 15)Histogramme des images couleurs

D) Classification :

- 16)Classification des images couleurs

E) Filtrage optimal :

- 17) Filtre de Canny (effets du filtre et applications)
- 18) Filtre de Gabor (formule mathématique, effets du filtre et applications)
- 19) Filtre Gaussien, déivateur Gaussien, DOG (difference of Gaussian), LOG (Laplacien of Gaussian). (formule mathématique, effets du filtre et applications)
- 20) Filtre de Deriche et filtre de Shen-Castan (formule mathématique, effets du filtre et applications).

Annexe

1) Guide de préparation des diaporamas (slides) pour la soutenance de projet :

S1 : Titre/Sujet

Auteurs

Formation

Année

S2 : Plan de présentation (Table des matières/Planning/Sommaire)

S3 : Début du contenu

.....

.....

S(fin-1) : Conclusion et Perspectives

S(fin) : Références

Numéroter tous les slides

2) Dépôt de travaux :

Projet : Rapport + Programmes sources + Slides de soutenance

TP : Rapport + Programmes sources

3) Appréciation

Soutenance : Compréhension du sujet,
Exposé clair,
Qualité de slides,
Question/Réponse.

TP avec résultats

Description des projets

Projet : Classification des images couleurs

Classification par Kmeans des images couleur : modifier l'algorithme de K-means pour classer les nuages de points.



Fig. 1

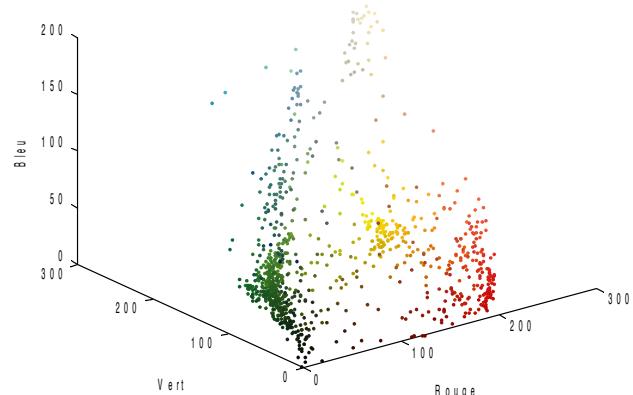


Fig. 2

Figure 1 : Image originale en 3 dimensions ((700 lignes, 527 colonnes donc 368 900 pixels à l'origine) qui est tronquée (en 350 lignes, 350 colonnes, RGB, 122 500 pixels)).

Figure 2 : Nuage de points de l'image originale (on y retrouve les couleurs les plus fréquentes de l'image 'perroquet.png').

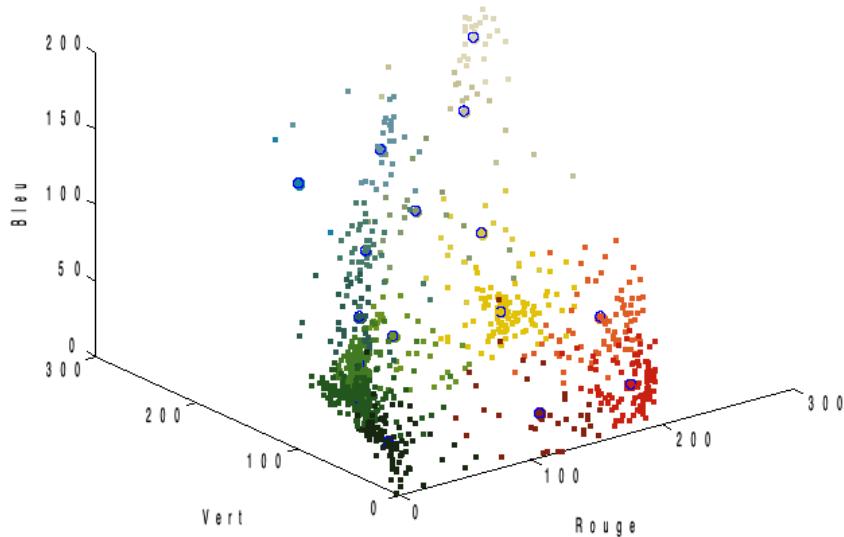
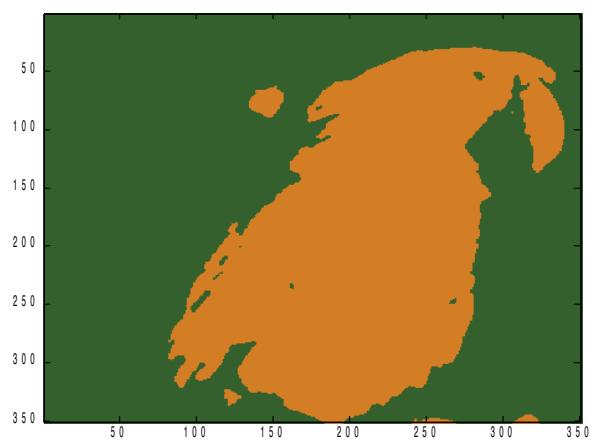
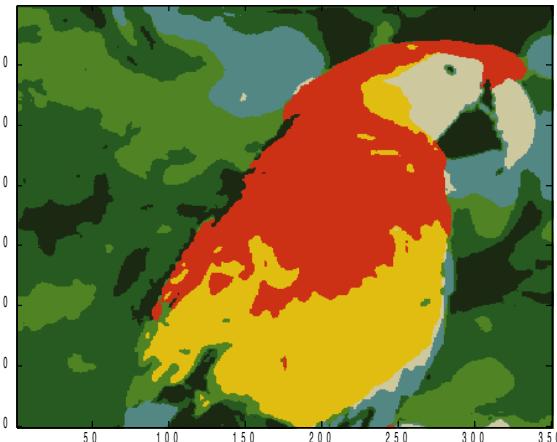


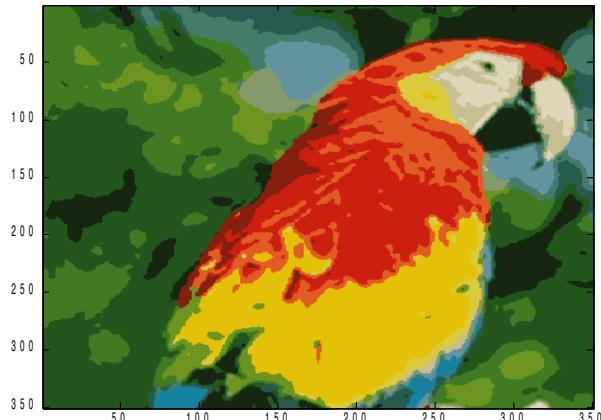
Figure 3 : On y ajoute les K classes (K couleurs)



K=2



K=7



K=16



K=32

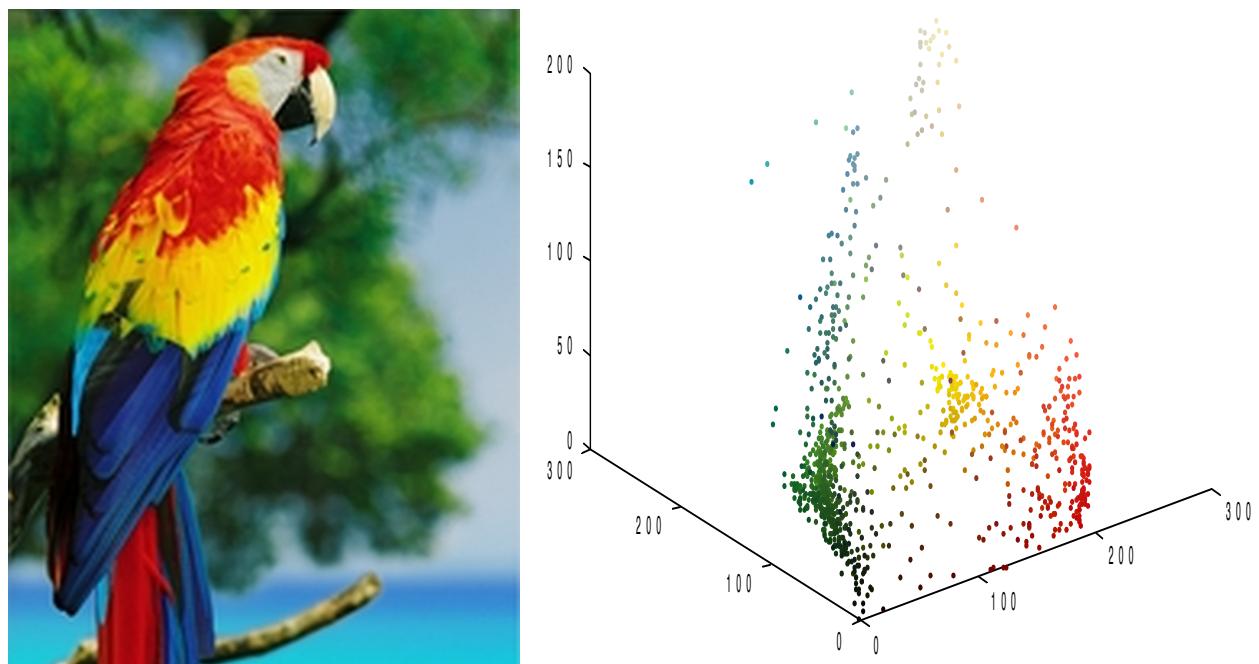


Fig. 1. Image originale et son nuage de points



Fig. 2. Classifications en 4, 8 et 32 couleurs

Projet : Segmentation

1) Segmentation par seuillage de l'histogramme et par Kmeans des images de poumons affecté et non affecté, puis Labellisation des images.

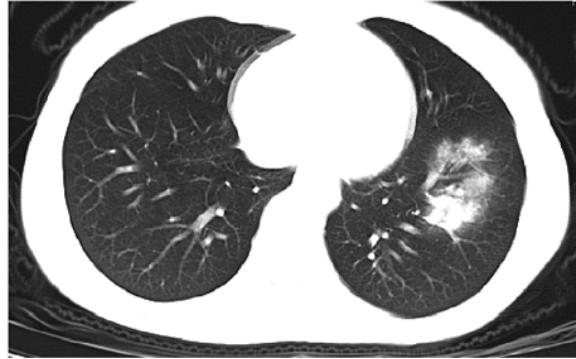


Fig 1. Poumon affecté



Fig 2. Poumon non-affecté

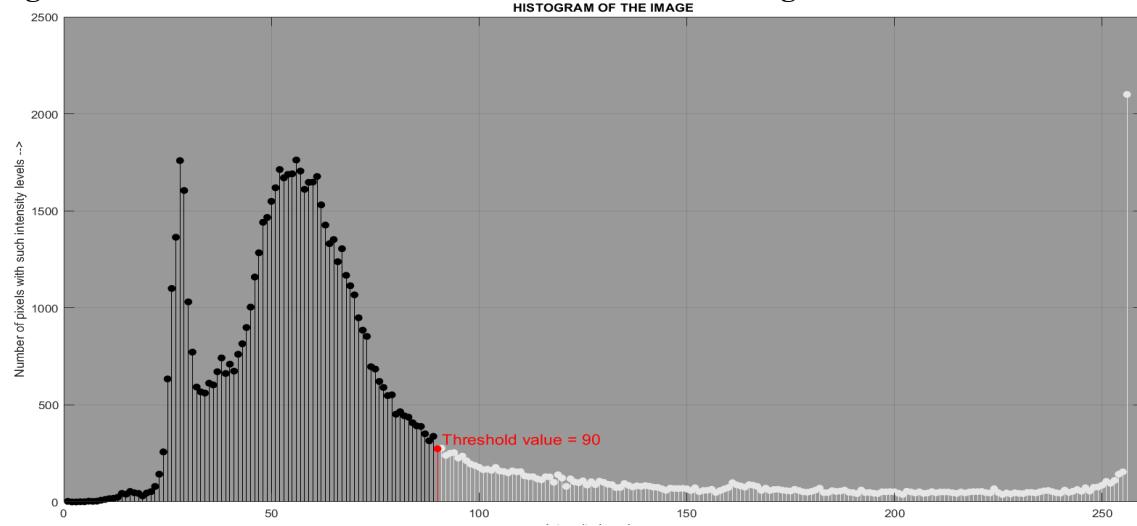


Fig.3 Histogramme du poumon affecté

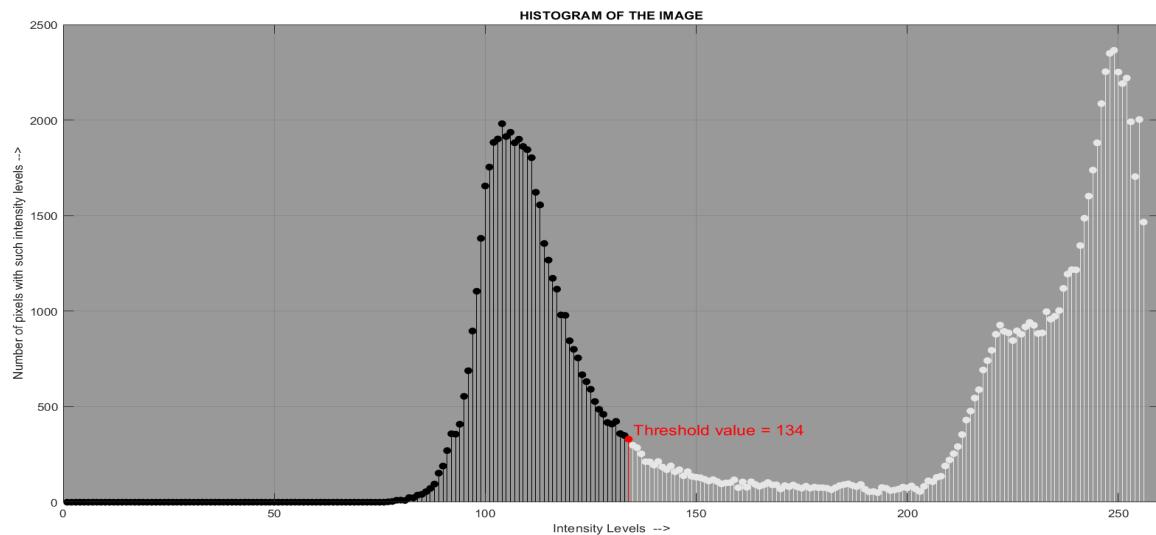


Fig.4 Histogramme du poumon non-affecté

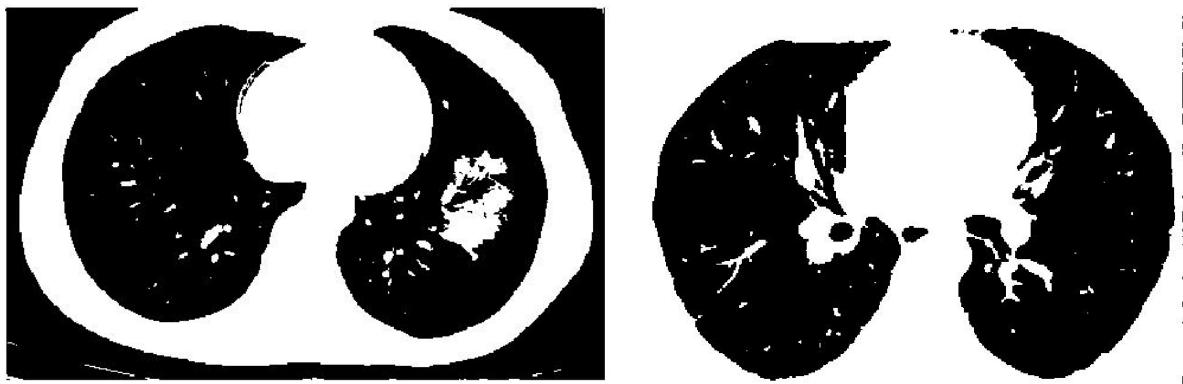


Fig 5. Segmentation par seuillage de l'histogramme en 2 classes

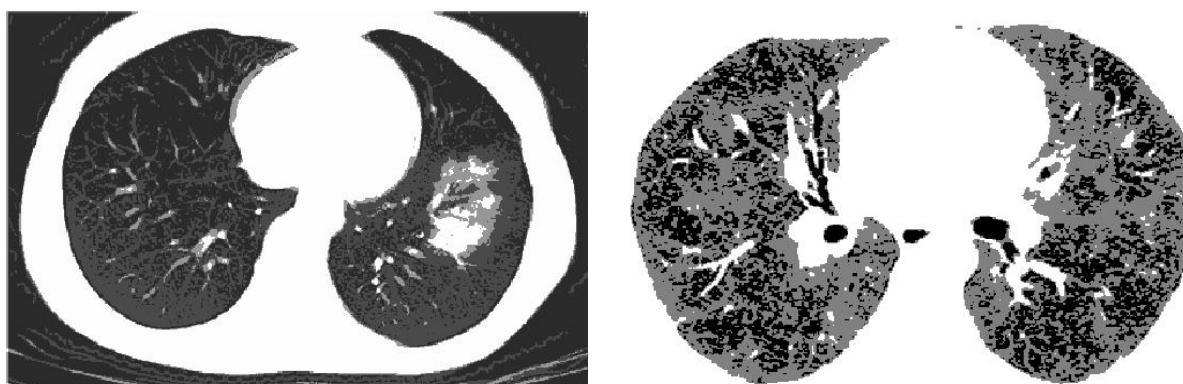


Fig 6. Segmentation par Kmeans en 4 classes

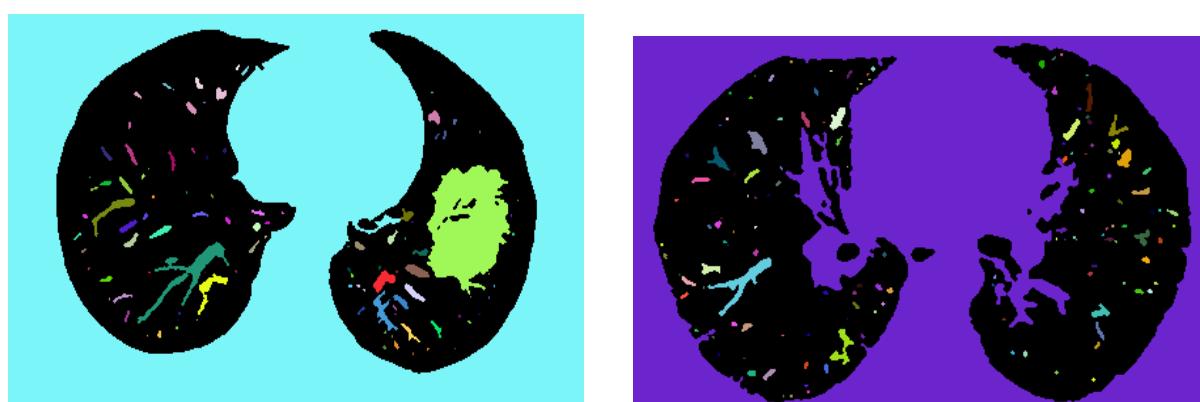


Fig. 7 Labellisation

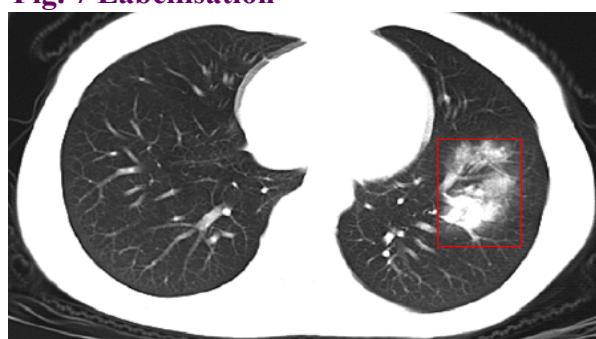


Fig. 8. Suivi de l'évolution de zone particulière

L'utilité de labellisation (Fig. 9) :

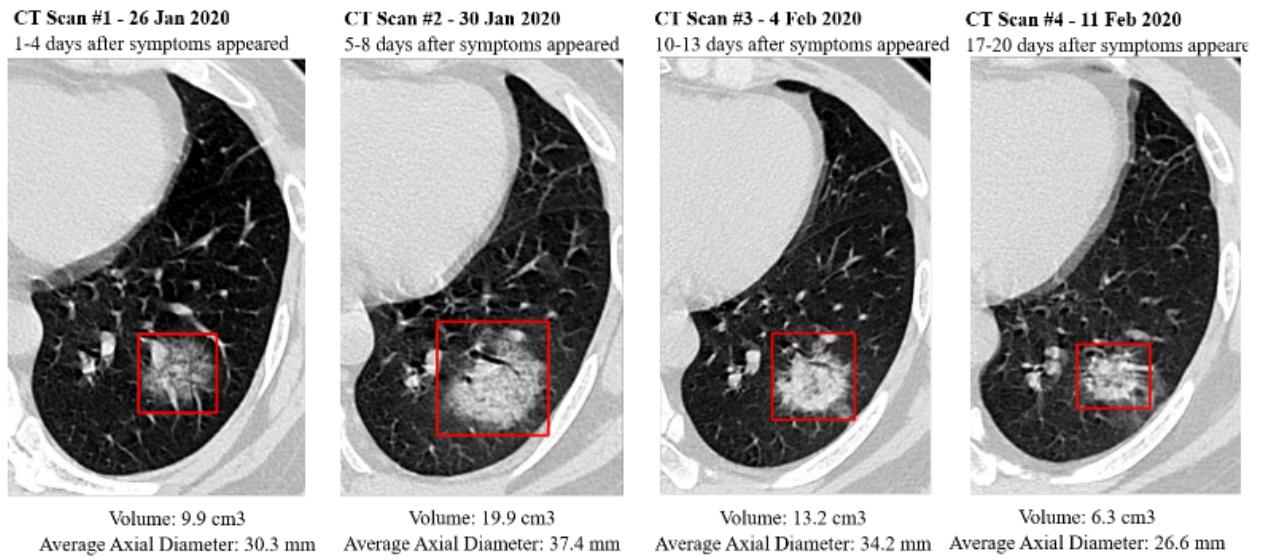


Fig. 9. (Multi time point tracking of patient disease progression)

2) Apport de l'intelligence artificielle et Deep learning.

(the standard clinical dataset called Phantom FDA downloaded from TCIA. The size of the images is 256×256 pixels).

Projet : Biométrie par des empreintes digitales

A. Introduction à la biométrie

La biométrie : effectuer des mesures sur du vivant. Elle remonte au 19^e siècle. Elle s'impose aujourd'hui comme une technique permettant de vérifier l'identité d'un individu à partir d'une ou plusieurs de ses caractéristiques personnelles (l'œil (iris, rétine), l'empreinte digitale, la voix, le visage, etc.). Contrairement aux mots de passe et codes pin ces caractéristiques ne peuvent être perdues, volées ou falsifiées puisqu'elles sont liées directement au corps de l'individu. Elle a un regain d'intérêt aussi bien dans le domaine médical, l'écologie, l'agronomie que dans le domaine de sécurité et la défense nationale.

La main est une région particulièrement riche d'informations pouvant être utilisée pour l'authentification ou pour l'identification des individus. Citons par exemple la morphologie de la main, les *réseaux veineux des doigts*, de la paume ou même de l'*avant bras*, l'empreinte et les lignes palmaires.

Parmi les diverses techniques identificatrices la reconnaissance des empreintes digitales est certainement une des modalités les plus utilisées. Elle est très fiable, peu coûteuse et facile à emploi. Cette technique est parfaitement adaptée à la gestion et au contrôle d'accès physique.

Une empreinte digitale est le dessin formé par les lignes de la peau (appelées les lignes de crête ou stries) et les espaces entre les stries appelées vallées.

L'approche la plus populaire pour identifier les ED est basée sur l'extraction des minuties qui désignent les différentes discontinuités des lignes d'une empreinte :

- une strie peut bifurquer et cette minutie s'appelle alors « bifurcation »,
- une strie peut s'arrêter soudainement, on parle de la « terminaison », etc.

B. Etat de l'art : avant de réaliser le projet effectuer une étude bibliographique sur l'état de l'art de la Biométrie en général et de la « Biométrie par empreintes digitales » en particulier.

C. Les étapes de « détection des minuties » et d' « Identification » :

- 1) Charger l'image initiale 'Empreinte1.bmp' en 3d, puis l'extraire en 2d et l'afficher (Fig. 1).
- 2) Calculer l'histogramme de l'image initiale et l'afficher (Fig. 2).
- 3) Binarisation par seuillage avec le seuil choisi à partir de l'histogramme dans la Fig. 2.
- 4) Afficher l'image binarisée (Fig. 3).
- 5) **Amincissement** des lignes dans l'image binarisée dans Fig. 3 (squelettisation-Hilditch).
- 6) Afficher le squelette.
- 7) Afficher l'image inverse du squelette (Fig. 4).
- 8) Détection des minuties par la technique de « **Crossing number** ».
- 9) Afficher les minuties sur la même image de squelette (Fig. 5).
- 10) Afficher les minuties dans une image séparée (carte des minuties, Fig.6).
- 11) **Identification** : recherche des méthodes d'identification, les comparer et tester.
- 12) **Authentification** : recherche des méthodes d'authentification, les comparer et tester.
- 13) **Appports de l'Intelligence artificielle et Deep Learning !**

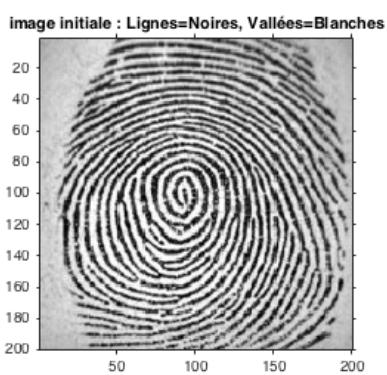


Fig.1 Image initiale en 2d

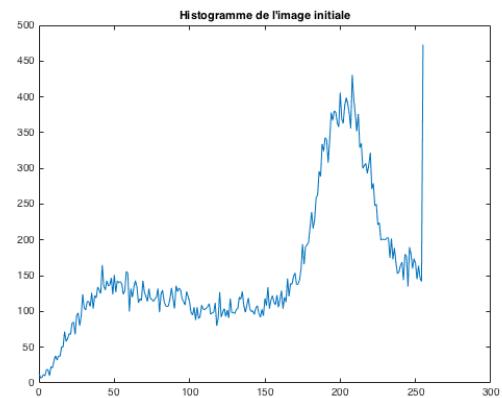


Fig. 2 Histogramme de l'image initiale

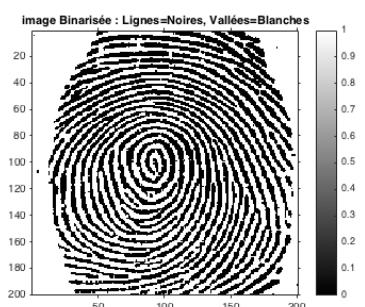


Fig. 3 Image binarisée

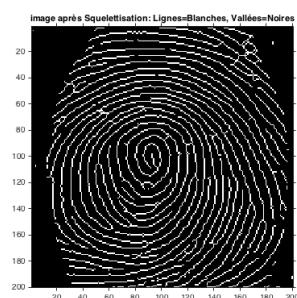


Fig. 4 Image squelettisée

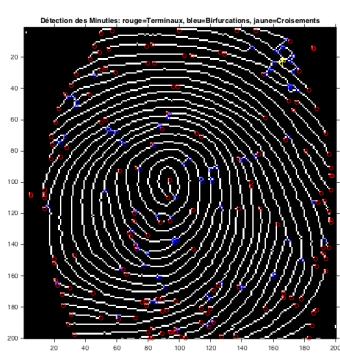


Fig. 5. Détection des minuties

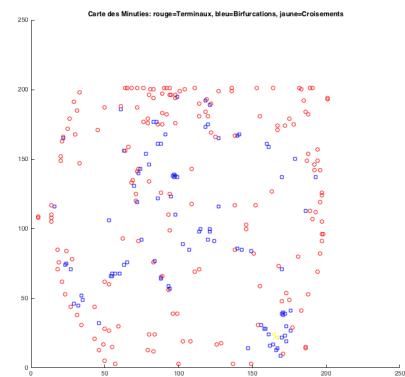


Fig. 6 Carte des minuties

Projets : Analyse de Rétines et applications en Imagerie médicale et Biométrie

A) Introduction à la biométrie

La biométrie : effectuer des mesures sur du vivant. Elle remonte au 19^e siècle.

Depuis plusieurs années, des efforts importants sont fournis dans le domaine de la recherche en biométrie. Ce constat s'explique par la présence d'un contexte mondial dans lequel les besoins en sécurité deviennent de plus en plus importants et où les enjeux économiques sont colossaux.

Les applications biométriques sont nombreuses et permettent d'apporter un niveau de sécurité supérieur en ce qui concerne des accès logiques (ordinateurs, comptes bancaires, données sensibles, etc.) ou des accès physiques (bâtiments sécurisés, aéroports, etc.).

L'œil qui est une merveille de la nature, offre les deux caractéristiques biométriques les plus fiables qui sont l'iris et la rétine. La biométrie par la rétine procure un haut niveau en matière de reconnaissance. Elle est bien adaptée pour des applications de haute sécurité. Elle est réputée d'être très fiable et difficile à falsifier.

Dans ce projet nous étudions les méthodes d'extraction de réseau vasculaire rétinien et de détection des points caractéristiques qui servent comme une signature biométrique afin d'authentification et d'identification des individus. Les méthodes de traitement d'images classiques seront confrontées aux traitements de type de réseau de neurones et de Deep learning.

Les différentes méthodes seront testées sur la base d'images ARIA (Analysis Retinal Image Archive), (Facultatives).

B) Etat de l'art : avant de réaliser le projet effectuer une étude bibliographique sur l'état de l'art de l'Analyse de rétine.

C) Les étapes à réaliser :

1. Charger l'image initiale en 3d, puis extraire l'image en canal vert en 2d, la convertir en image de niveaux de gris et l'afficher (étapes a, b, c).
2. Extraire les vaisseaux par filtrage (étape d).
3. Binarisation (étape f).
4. Suppression des régions mineures (étape g).
5. **Amincissement** (squelettisation-Hilditch) des vaisseaux dans l'image binarisée (étape h).
6. Détection des minuties par la technique de « **Crossing number** ».
7. Afficher les minuties sur la même image de squelette (étape i).
8. Afficher les minuties dans une image séparée (carte des minuties), (étape j).
9. **Identification** : recherche des méthodes d'identification, les comparer et tester.
10. **Authentification** : recherche des méthodes d'authentification, les comparer et tester.

11. Apports de l'Intelligence artificielle et Deep Learning !

Illustration : Les résultats des différentes étapes d'analyse. (a): image originale ; (b): image en niveau de gris du canal vert ; (c) image de niveaux de gris ; (d): résultats de la convolution entre l'image et les filtres gaussiens ; (e): génération du masque ; (f): image binarisée ; (g): image segmentée, (h): image squelettisée ; (i) : Extraction des points caractéristiques (minuties) ; (j) : Carte de minuties.

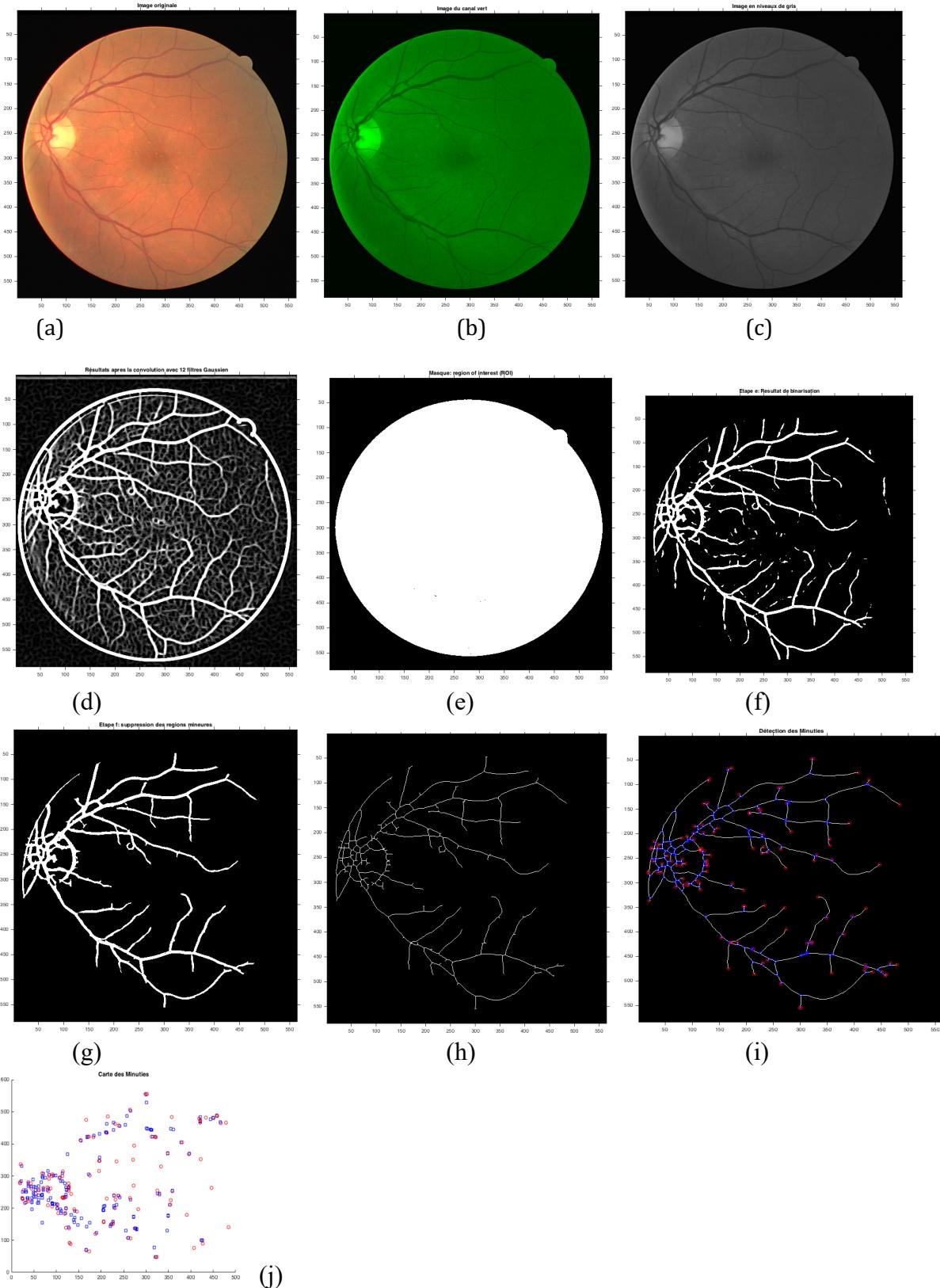


Image Denoising Project

1 Image denoising

Image denoising is the process of removing noise from an image, while preserving its original content and quality. Noise is any unwanted or random variation in the pixel values of an image, which can degrade its appearance and readability. Noise can be caused by various factors, such as low lighting, high ISO settings, sensor defects, transmission errors, or compression artifacts.

1.1 Auto-encoder

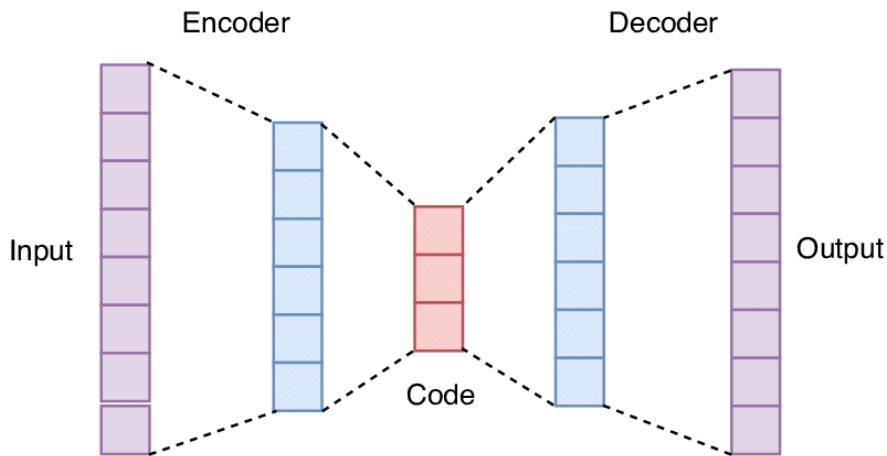
One way to perform image denoising is to use an autoencoder, which is a type of neural network that learns to reconstruct its input from a compressed representation. An autoencoder consists of two parts: an encoder and a decoder. The encoder maps the input to a lower-dimensional latent space, and the decoder maps the latent space back to the original input space.

An autoencoder is good for denoising because it can learn to filter out the noise and recover the clean images. By adding some noise to the original images and using them as the input of the encoder, the autoencoder is forced to learn the essential features and patterns of the images, while ignoring the irrelevant variations. The output of the decoder should be as close as possible to the original images, without noise. This way, the autoencoder can act as a denoising function that maps noisy images to clean images.

For the loss function we need to use something compares the output of the decoder with the original images, without noise. This way, we can quantify how well the autoencoder can reconstruct the clean images from the noisy images. One possible loss function for this task is the mean squared error (MSE),

$$\text{which is defined as: } \text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

1



1.2 Dataset

Figure 1: Architecture of an Autoencoder

We will be using the MNIST dataset, which is a collection of 60,000 images of handwritten digits from 0 to 9, with their corresponding labels. However, before we can use the MNIST dataset to train and test our neural network model, we need to do some preprocessing to it. For example, we may need to resize, crop, normalize, augment, or transform our images and labels to fit the input and output dimensions of our model.

More importantly we will add some artificial noise to the original images from the MNIST dataset. This will make our images more similar to the ones that we may encounter in real life, where noise is often unavoidable and undesirable. There are different types of noise that we can add to the images

- Gaussian noise: This is a type of noise that follows a normal distribution with a mean of zero and a standard deviation of sigma. Gaussian noise is also called white noise or additive noise, as it adds random values to each pixel independently.
- Salt-and-pepper noise: This is a type of noise that randomly replaces some pixels with either black or white values. Salt-and-pepper noise is also called impulse noise or binary noise, as it creates sharp spikes or impulses in the image.

2

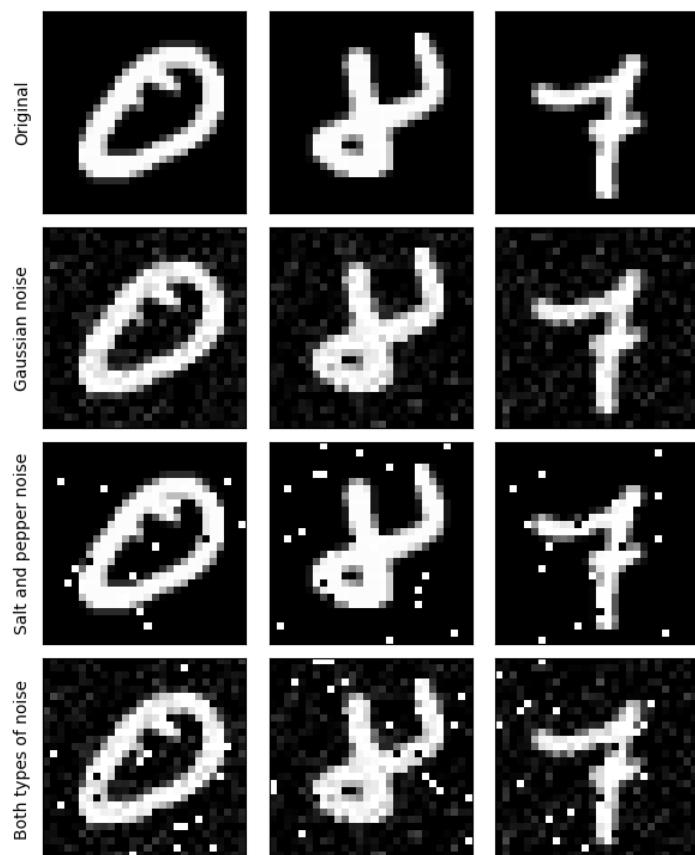


Figure 2: Gaussian and S&P noise applied to a sample of MNIST

2 Exercises

1. Download the MNIST dataset. Preprocess the data as necessary and add gaussian noise to the images.
2. Program the Autoencoder architecture in PyTorch or Pytorch lightning.
3. Calculate the loss function and set up the training loop with the optimizer.
4. Train the model and add metric plots to track the loss of the model as it trains, you could also optionally use Tensorboard for this.
5. Bonus: Try different types/combinations of noise like the salt and pepper noise
6. Bonus: Use a Variation Autoencoder (VAE) to denoise the images.

Projet : Filtre de Gabor

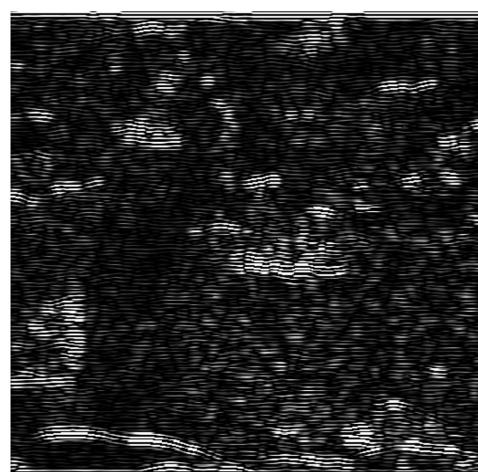
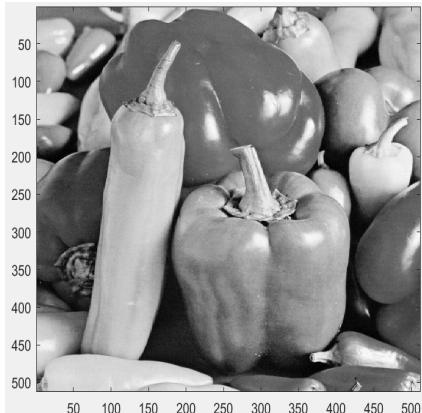
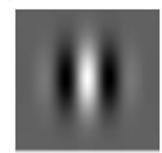
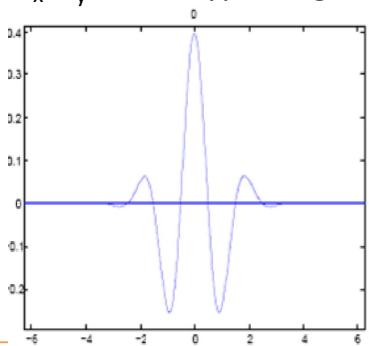
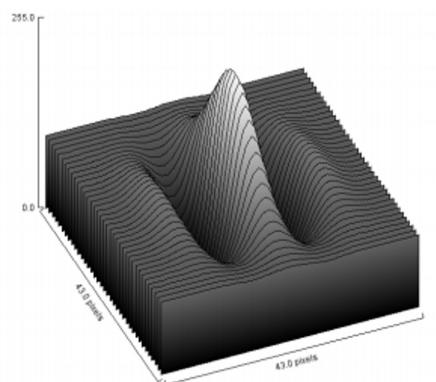
$$G(x, y, \theta, f) = e^{-\frac{1}{2} \left(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2} \right)} \cos(2\pi f x_\theta)$$

(x, y) : coordonnées de pixel dans l'image

θ : orientation de la sinusoïde

f : fréquence de l'onde

σ_x, σ_y : écart-type du gaussien selon l'axe (x,y)



Variation de fréquence



Variation de l'orientation



