

# TP1: Qualité et rehaussement d'image

## GBM8770 – Automne 2024

Professeure: Farida Cheriet

Chargés de laboratoire: Zacharie Legault et Emmanuelle Richer

---

**Objectifs :** L'objet de ce travail est de mettre en pratique certains des concepts élémentaires concernant la qualité des images médicales qui ont été présentés en cours, tout en permettant une familiarisation avec le langage de programmation Python. Le TP sera divisé en 2 parties : la première partie porte sur la qualité des images (résolution, contraste, bruit), et la seconde partie porte sur le rehaussement d'images.

**Remise du travail :** La date de remise est le 3 octobre à 23h30. Une pénalité de 3 points par jour sera appliquée lors d'un retard.

**Documents à remettre :** Le code et les réponses aux questions sont à compléter dans les Notebook IPython fournis avec cet énoncé : `tp1_seance1.ipynb` et `tp1_seance2.ipynb`.

Le dossier contenant le notebook et toutes les ressources nécessaires à son exécution sont à remettre sous la forme d'une archive ZIP nommée `GBM8770_TP1_<équipe>_<nom>_<matricule>_<nom>_<matricule>` (par exemple : `GBM8770_TP1_0_Legault_1234567_Richer_7654321` ).

Commentez votre code ! Lorsque votre code (et donc vos résultats) est incorrect, vos commentaires peuvent nous permettre de valoriser votre approche.

---

## Séance I.

# Qualité d'image : Résolution, contraste, et bruit

## Exercice I : Étude de la résolution

### Mise en place des fonctions

Les prochains exercices vous serviront dans la suite du TP. Vous pouvez évidemment vous référer au TP0 ainsi qu'à la **Cheatsheet** fournie pour vous aider. Nous utiliserons une image IRM du cerveau afin d'étudier la résolution d'une image.

**Q1.** Ouvrez et affichez l'image **brain\_tumor.jpg**

**Q2.** Déterminez graphiquement la largeur approximative de la tumeur en pixel (selon l'axe transversal). Affichez ce diamètre sous forme d'une ligne en pointillés délimitée par deux , avec la fonction **ax.plot(x, y)**.

**?** Vous pouvez utiliser **ax.imshow(img)** puis **ax.plot(x, y)** pour afficher une image **img** et y superposer les points **(x, y)**. Dans ce cas-ci, **ax** représente le système d'axes de votre figure, obtenu par exemple par l'appel **fig, ax = plt.subplots()**. Vous pouvez également afficher la grille sur l'image avec **ax.grid()** pour mieux visualiser les positions en pixels des points qui vous intéressent.

**Q3.** Définissez la fonction **taille\_pixel(x0, y0, x1, y1, ref)** qui calcule la taille physique d'un pixel en millimètre.

**Q4.** La largeur de la tumeur est de 3 cm. Déduisez la largeur d'un pixel en millimètre.

**Q5.** Définissez la méthode **downsample(img, f)** qui prend en paramètre une image et un facteur de sous-échantillonnage et renvoie l'image dont seulement 1 sur **f** pixel a été conservé.

**i** Il existe plusieurs façon de sous-échantillonner une image. Choisissez la méthode qui vous semble la plus appropriée. Ça n'a pas besoin d'être compliqué! Regardez la syntaxe d'indexage de matrices en Python, par exemple. Évitez d'utiliser des boucles `for`.

**Q6.** Affichez l'image de la tumeur sous-échantillonnée avec **f=2**, **f=3**, et **f=4**. Affichez le facteur de downsampling ainsi que la taille de l'image downsamplée dans le titre de la figure. Comment évolue la largeur d'un pixel lors du sous-échantillonnage? Donnez un avantage et un inconvénient de l'utilisation d'une image médicale downsamplée plutôt que l'image pleine résolution.

**?** Il existe encore une fois plusieurs façons d'afficher la valeur d'une variable dans un **string**. Vous pouvez utiliser la syntaxe suivante afin d'afficher la valeur de la variable **x** : `f"ma variable {x}"` ou encore `"ma variable {:.2f}".format(x)` pour forcer l'affichage à 2 décimales.

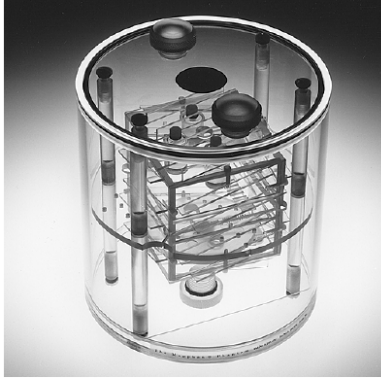
## Calibration d'un IRM

La mise en place de contrôles de qualité réguliers pour les dispositifs d'imagerie médicale permettent une détection rapide d'une éventuelle dégradation progressive de la qualité des images. En effet, en particulier dans le cas de dérives lentes, les opérateurs peuvent ne pas détecter immédiatement une perte de qualité. La dégradation de ces images peut entraîner des erreurs d'interprétation, ce qui pourrait avoir de lourdes conséquences cliniques. Une seconde justification pour la mise en oeuvre de ce contrôle est la vérification des performances du dispositif suite à une opération de maintenance, susceptible d'avoir affecté un paramètre en lien avec la qualité image par exemple.

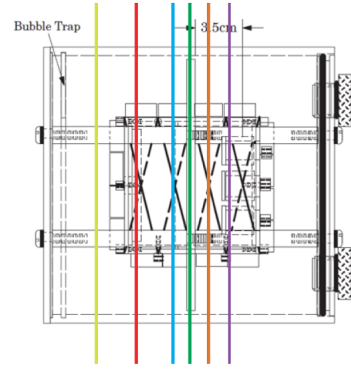
Pour vérifier le bon fonctionnement d'un dispositif d'imagerie, on image un *fantôme*, c'est-à-dire un objet dont les formes et les caractéristiques sont connues. Dans cet exercice, nous allons travailler sur un fantôme d'imagerie par résonance magnétique (Figure 1a), observable selon différentes coupes en fonction du type de contrôle qualités désirés (Figure 1b). L'objectif sera d'établir la qualité de résolution, de contraste et de bruit d'un IRM.

Pour plus d'information sur le fantôme, voir ce lien.

Le constructeur a inclus dans le fantôme un test de haute résolution. On retrouve dans le fantôme une série de mires en acrylique (horizontales et verticales) dont la taille varie de 1/2 à 1/8 cm (mires horizontales) et de 1 à 1/12 mm (mires verticales). Les mires sont disposées



(a) Schéma descriptif du fantôme d'imagerie étudié dans ce TP (un Magphan cylindrique de *The Phantom Laboratory*).



(b) Coupe du fantôme pour les différents tests de contrôle de qualité.

FIGURE 1 – Fantôme IRM

en blocs, dont les caractéristiques sont notées dans le tableau 1. La figure 2 vous montre la disposition de ces blocs, chacun ayant une couleur distincte. Lorsqu'on parlera des profils d'intensités, on se référera à l'intensité des pixels définis par les lignes blanches sur cette figure.

TABLEAU 1 – Caractéristiques des mires

Position	Taille des mires individuelles	# mires / bloc
Horizontales	1/8 cm	4
	1/6 cm	3
	1/4 cm	3
	1/2 cm	3
Verticales	1 mm	5
	1/2 mm	5
	1/4 mm	5
	1/6 mm	5
	1/8 mm	5
	1/10 mm	5
	1/12 mm	5

**Q7.** Ouvrez et affichez l'image **resolution.png**. Affichez l'image en niveau de gris, cachez les axes et ajoutez un titre. Par dessus l'image, dessinez des lignes pointillées le long des mires verticales et horizontales. Affichez une légende pour les profils.

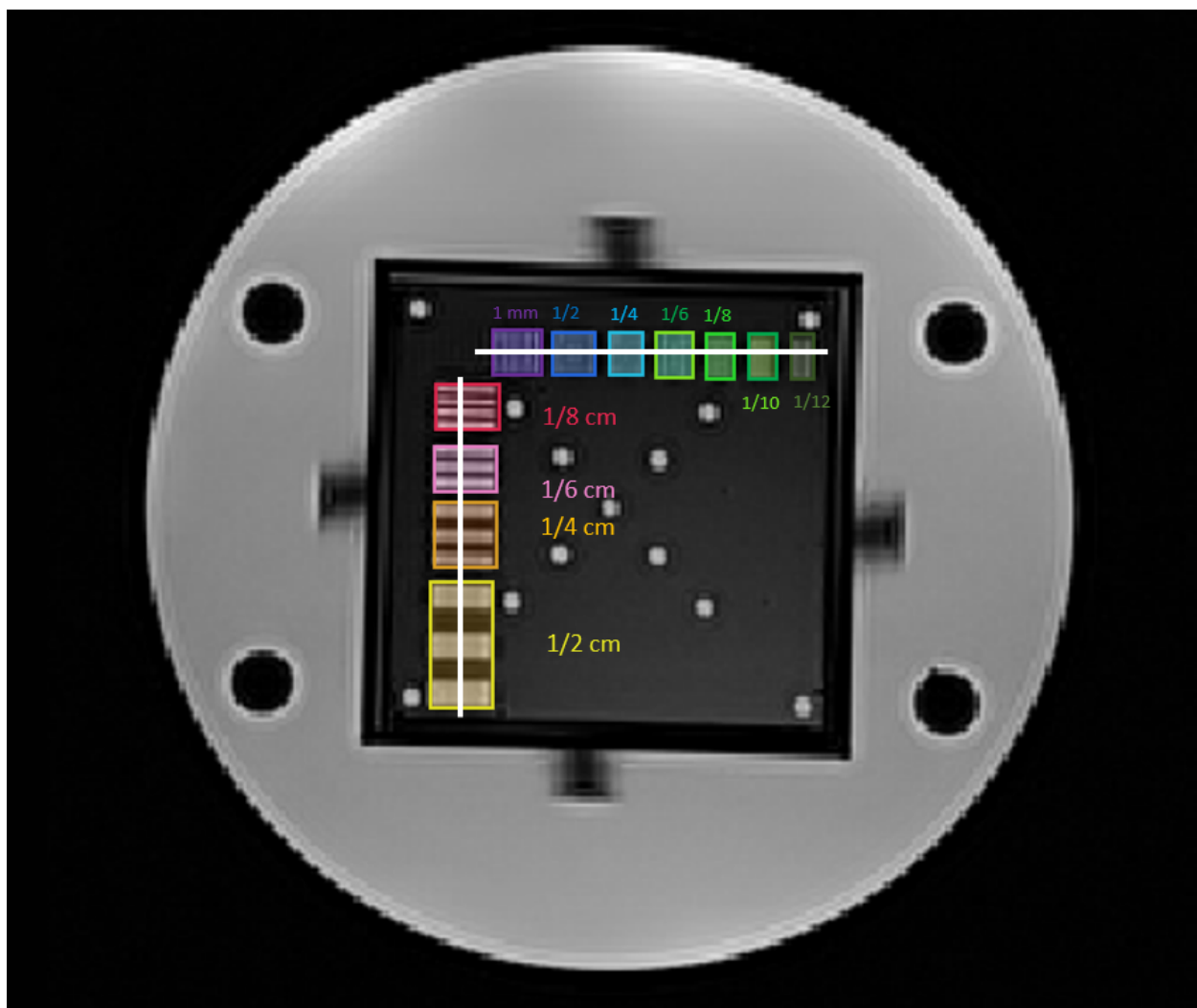


FIGURE 2 – Mires hautes résolutions.

? Vous pouvez prendre **y=201:341** et **x=200** pour le profil vertical et **y=190** et **x=206:346** pour le profil horizontal. Utilisez les fonctions de **numpy** et de **matplotlib** utilisées dans le TP0 et dans les exercices précédents. Par exemple **np.arange**, **np.linspace**, **plt.plot**, **plt.imshow**, **plt.legend**, ...

**Q8.** Le constructeur indique que la largeur du cube est de 10 cm. En déduire la largeur d'un pixel en millimètre. (Les coordonnées des coins supérieurs du carré sont **(160, 151)** et **(356, 157)**.)

**Q9.** Quelle est la relation entre la largeur d'un pixel et la taille du plus petit objet observable ? Selon cette relation, quelle serait la taille du plus petit objet observable dans cette image ?

**Q10.** Afin de vérifier cette relation, affichez l'évolution des intensités des pixels le long des 2 axes (affichés en blanc sur la figure 2) qui coupent les mires perpendiculairement. Utilisez la distance en millimètres comme abscisses. Quelle est l'épaisseur de la plus petite mire observable ? Est-ce cohérent avec votre réponse donnée à la question précédente ?

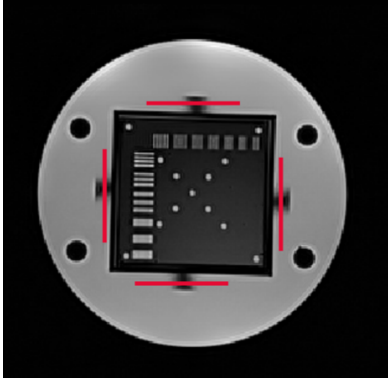
? Fiez-vous au tableau 1 et à la figure 2 de l'énoncé.

## Mesure de l'épaisseur de coupe

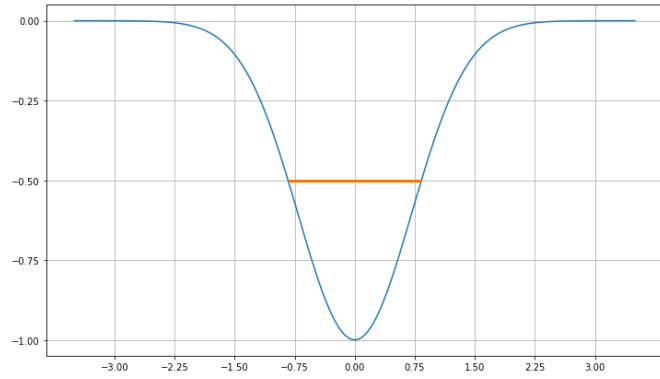
Outre la résolution latérale, le fantôme permet aussi de mesurer l'épaisseur de la coupe réalisée par l'IRM. Pour ce faire on mesure la largeur des rampes contenues dans le fantôme (voir figure 3a). Ces rampes sont légèrement inclinées sur le fantôme ; le profil d'intensité de ces rampes sera donc plus ou moins large selon l'épaisseur de coupe de l'IRM. Les paramètres de l'IRM indiquent que les images ont été acquises avec une épaisseur de coupe théorique de 2 mm. Nous allons maintenant vérifier cette valeur.

**Q11.** Affichez les profils d'intensités des 4 rampes et mesurez graphiquement la largeur des gaussiennes à mi-hauteur.

? Vous pouvez normaliser le signal sur une gamme d'intensité plus facile à manipuler. Vous pouvez ensuite tracer une ligne horizontale pour vous aider à mesurer la largeur à mi-hauteur. Afin d'adapter l'échelle des abscisses, référez-vous à la documentation de la fonction **ax.set\_ticks()**.



(a) Rampes de mesure d'épaisseur de la coupe. Les lignes de profils à extraire sont affichés en rouge.



(b) Largeur à mi-hauteur d'une gaussienne.

FIGURE 3 – Largeur à mi-hauteur des rampes

Sachant que l'inclinaison des rampes est de  $14^\circ$ , l'épaisseur  $e$  de la coupe se calcule par :

$$e = \tan(14^\circ) \times \text{FWHM}$$

où FWHM est la largeur à mi-hauteur **en millimètre** de la gaussienne formée par le profil d'intensité.

**Q12.** Sachant que la tolérance sur l'épaisseur de coupe est de  $\pm 0.3 \text{ mm}$ , l'IRM passerait-il le contrôle qualité ?

## Exercice II : Analyse du contraste et du bruit

La tomographie par cohérence optique (OCT) est une modalité d'imagerie qui permet d'imager des structures fines en haute résolution. On utilise l'OCT particulièrement pour imager la rétine (figure 4a) et ainsi diagnostiquer diverses pathologies l'affectant, comme la dégénérescence maculaire liée à l'âge (figure 4b, 4c) ou encore la rétinopathie diabétique (figure 4d). Les images OCT sont souvent bruitées. Cet exercice vise à évaluer le contraste et le bruit de ces images.

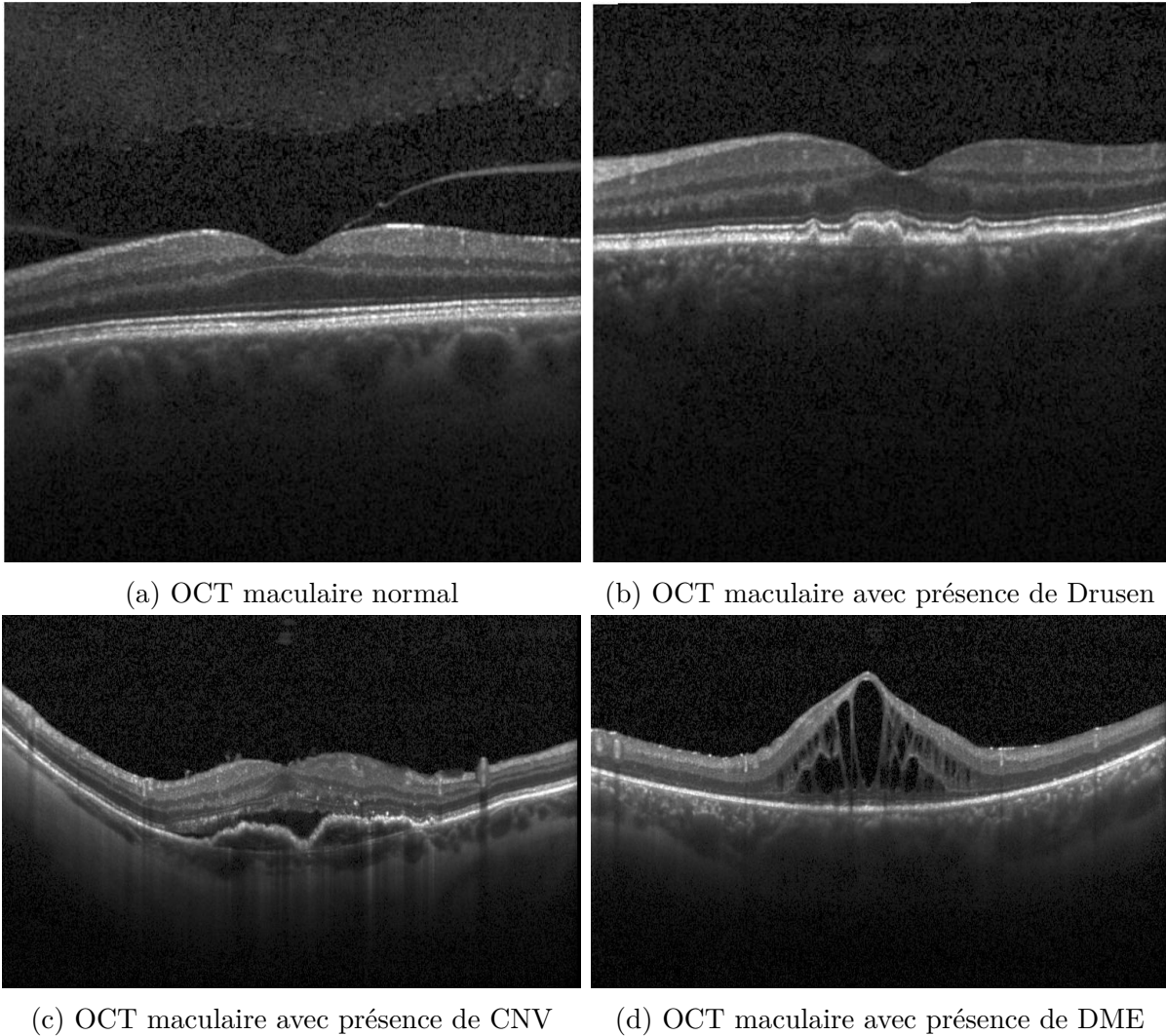


FIGURE 4 – Images OCT présentant certains biomarqueurs de diverses maladies oculaires. Images issues du jeu de données de l'Université de Californie à San Diego.

## Mesure du contraste

Des segmentations de différentes structures rétinienne observables à l'OCT vous sont fournies. Nous utiliserons l'équation du contraste local telle que donnée dans les notes de cours.

$$C = \frac{I_{\text{cible}} - I_{\text{fond}}}{I_{\text{fond}}}$$

Les prochaines questions vous guident dans le calcul du contraste sur ces différentes structures afin d'évaluer laquelle possède le meilleur contraste.

**Q1.** Chargez et affichez l'image `oct.png`.



**Q2.** Chargez et affichez aussi les 5 matrices numpy

- `mask_choroid_left.npy`
- `mask_choroid_right.npy`
- `mask_rpe_left.npy`
- `mask_rpe_right.npy`
- `mask_onh.npy`

Chacune contient le masque<sup>1</sup> d'une des structures de la rétine que nous étudierons. N'oubliez pas de donner un titre à vos figures.

**Q3.** Regrouper les segmentations de la choroïde et de la RPE en un seul masque par région (1 masque pour la choroïde, gauche et droite, etc.). Affichez les segmentations regroupées.

**?** Vous pouvez par exemple effectuer une opération **OR** sur les deux masques, avec la fonction `np.logical_or`.

**Q4.** Utilisez le masque `mask_retina.npy` pour définir un masque du fond de l'image (en soustrayant la rétine).

**?** Vous pourriez par exemple créer une matrice de 1 de la même taille que l'image et ensuite mettre à 0 tous les pixels à l'intérieur de la segmentation de la rétine.

**Q5.** Calculez l'intensité moyenne sur ces 3 régions (choroïde, RPE et tête du nerf optique) et sur le fond de l'image. En déduire le contraste entre chaque région et le fond de l'image OCT. Quelle est la zone de la rétine où le contraste est le meilleur ?

## Mesure du rapport contraste/bruit

Vous avez vu en cours le rapport signal sur bruit (SNR). Cette métrique est définie de plusieurs façons (SNR d'amplitude, de puissance, différentiel, etc.). Typiquement, le SNR définit le signal moyen dans une région divisé par le bruit autour de cette région. Une autre métrique intéressante dans la quantification de la qualité des images est le rapport contraste sur bruit (CNR). L'équation 1 définit cette métrique (ou  $\mu$  représente la moyenne des intensités dans une région et  $\sigma$  l'écart-type mesuré dans une région uniforme). On calculera uniquement le CNR dans la suite de cet exercice.

Un des problèmes typiquement relié à l'acquisition des images médicales est le bruit qu'on y trouve (exemple : radiographie en figure 5). Ce bruit peut être caractérisé par différentes

---

1. Un masque est une image binaire (dont les pixels valent soit 0 soit 1) qui identifie une région de l'image.

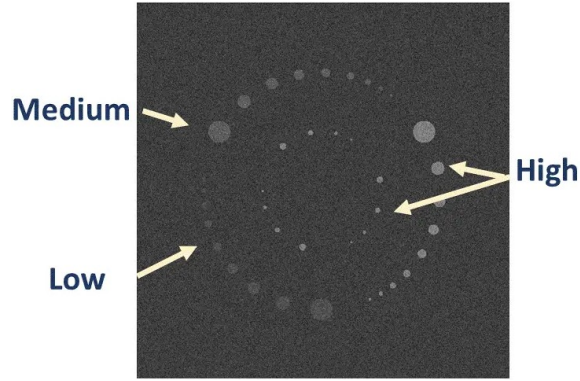


FIGURE 5 – Notre capacité à distinguer un objet dans une image bruitée dépend de la taille de l’objet et de son intensité par rapport au bruit.

distributions (sel et poivre, gaussien, speckle, etc.) en fonction du système d’imagerie utilisé. Une technique souvent utilisée afin de débruiter des images OCT est de faire l’acquisition rapide de plusieurs images et de les moyenner. Si le temps d’acquisition est suffisamment rapide, les structures ne bougent pas et la seule source de variation entre les images est le bruit aléatoire.

$$\text{CNR} = \frac{\mu_{\text{cible}} - \mu_{\text{fond}}}{\sigma_{\text{fond}}} \quad (1)$$

$$\text{CNR}_{\text{dB}} = 20 \cdot \log_{10}(\text{CNR}) \quad (2)$$

**Q6.** Ouvrez et affichez les images **oct.png**, **oct\_2.png**, **oct\_3.png**, **oct\_4.png** et **oct\_5.png**.

**Q7.** En utilisant la segmentation de la rétine et la segmentation de la région avec le plus grand contraste (question précédente), calculez le CNR en décibel sur ces 5 images.

**Q8.** Calculer une image moyenne de ces 5 images, qui sera une estimation de l’image OCT débruitée. Affichez côte à côte une des images originale et l’image moyennée. Quelles différences remarquez-vous ?

**Q9.** Calculer le CNR sur une des images originales, l’image moyennée calculée à la question précédente, et sur l’image **ave.png** fournie. Affichez les trois images côte-à-côte et comparez leurs CNRs. Quelle image obtient le CNR le plus élevé ?

**Q10.** Que pouvez-vous conclure sur la technique de moyennage comme technique de débruitage ? (Pourriez-vous utiliser la technique de moyennage en tout temps ? Quelles conditions doivent être respectées pour que la technique fonctionne bien ?)

**Q11.** Que représente le rapport contraste/bruit ? Est-il préférable d'avoir un CNR faible ou fort ?

**Q12.** Nommez un avantage de calculer le rapport contraste/bruit par rapport au rapport signal/bruit.

## Séance II.

# Rehaussement d'image

**Note :** Lors de la conception de vos algorithmes, portez attention aux types de vos données (**uint8**, **double**, etc.). Lors de la manipulation des images, vous devrez parfois passer en **double** ou en **uint8** en fonction des exercices afin d'effectuer vos calculs.

## Exercice III : Prétraitement d'images grayscale

Le but de cet exercice est d'implémenter un traitement de rehaussement des contrastes et des contours pour différentes images médicales.

**!** Dans toutes les questions de cet exercice, prenez soin de spécifier les bornes des intensités lors de l'affichage.

**i** Par défaut, la fonction **imshow** du package **matplotlib** change la colormap d'affichage de l'image pour couvrir l'intervalle des valeurs minimales et maximales de l'image. Par exemple, si une image contient des valeurs d'intensités n'allant que de 10 à 233, la couleur noire pure correspondra à la valeur d'intensité de 10 et la couleur blanche pure correspondra à la valeur d'intensité de 233. Pour afficher une image en forçant le noir pur à 0 et le blanc pur à 255, on spécifie les arguments **vmin** et **vmax**. Lorsqu'on compare le contraste d'images, il est mieux d'avoir les mêmes bornes d'affichage en tout temps.

## Égalisation d'histogramme

**Q1.** Ouvrez, convertissez en valeurs entières et affichez l'image **radio\_thoracique.png**.

**Q2.** À l'aide de **np.histogram**, calculez puis affichez son histogramme. Pour gagner du temps aux questions suivantes, vous pouvez définir une fonction qui effectue ces deux opérations.

**Q3.** Complétez la fonction **equalize\_histogram(img)** qui effectue l'égalisation d'histogramme

d'une image `img` et renvoie l'image égalisée. Comparez l'image et son histogramme avant et après l'égalisation. Prenez soin de spécifier les bornes des intensités lors de l'affichage. Que constatez vous ? Quel impact a l'égalisation sur le contraste de l'image ?

**?** La marche à suivre est décrite dans vos notes de cours (chapitre 4). Vous aurez besoin de la fonction `np.cumsum`.

**i** Astuces : pour transformer l'intensité de chaque pixel d'une image  $I$  selon un mapping  $T$  (c'est-à-dire pour calculer  $I_{eq}[i, j] = T[I[i, j]]$  pour tous les  $i$  et  $j$ ), pas besoin de boucle `for` : il suffit de faire :  $I_{eq} = T[I]$

**Q4.** Effectuez les mêmes opérations qu'à la question précédente (égalisation d'histogramme puis comparaison de l'image et histogramme avant et après égalisation) sur l'image `fundus_mauvais_contraste.png`. Que constatez-vous ? Que proposeriez-vous afin d'améliorer le contraste et la visibilité des structures de l'oeil dans cette image ?

## Filtrage high-boost

**Q5.** Ouvrez l'image `angiography.jpg`. Convertissez l'image en valeurs flottantes (normalisez par la valeur maximale) et conservez uniquement le premier canal de cette image. Affichez l'image résultante.

**!** Rappel : prenez soin de spécifier les bornes des intensités lors de l'affichage.

**Q6.** Calculez la convolution de l'image angiographique avec un filtre gaussien  $3 \times 3$  :

$$W_{\text{Gaussienne3}} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Affichez côte à côte l'image angiographique et le résultat de la convolution.

**Q7.** Effectuez la même opération (convolution et affichage) avec une gaussienne de taille

$7 \times 7$  :

$$W_{\text{Gaussienne7}} = \frac{1}{1115} \begin{bmatrix} 1 & 4 & 7 & 10 & 7 & 4 & 1 \\ 4 & 12 & 26 & 33 & 26 & 12 & 4 \\ 7 & 26 & 55 & 71 & 55 & 26 & 7 \\ 10 & 33 & 71 & 91 & 71 & 33 & 10 \\ 7 & 26 & 55 & 71 & 55 & 26 & 7 \\ 4 & 12 & 26 & 33 & 26 & 12 & 4 \\ 1 & 4 & 7 & 10 & 7 & 4 & 1 \end{bmatrix}.$$

Qu'observez-vous lorsque l'écart-type de la gaussienne augmente (sachant que l'écart-type est proportionnel à la taille du masque) ? Quel est le type (passe-haut, passe-bas, passe-bande ou coupe-bande) de ces deux filtres gaussiens ?

**Q8.** Calculez et affichez le laplacien de l'image angiographique. On rappelle que le calcul du laplacien d'une image est approximé par la convolution de cette image avec le filtre

$$W_{\text{Laplacien}} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

Quel est le type de ce filtre ?

**!** Attention : le laplacien d'une image contient des valeurs négatives. Pour les visualiser correctement, les bornes d'affichages doivent être de  $-1$  et  $1$  !

**Q9.** Implémentez une fonction `high_boost(img, k)` qui prend en entrée une image et un paramètre `k` et retourne l'image rehaussée selon le filtre vu en cours :

$$I_g = G * I$$
$$I_r = I + k \cdot \nabla^2(I_g)$$

où  $G$  est un masque gaussien (on prendra celui de taille 3) et  $\nabla^2$  est l'opérateur laplacien. Calculez et affichez le résultat pour  $k = 0$ ,  $k = 1$  et  $k = 2$ . Discutez de l'impact du facteur  $k$  sur la lisibilité de l'image (et donc la qualité du rehaussement). Dans ce filtrage, pourquoi calculer le laplacien sur  $I_g$  plutôt que sur  $I$  directement ?

**!** Limitez l'affichage entre 0 et 1.

# Exercice IV : Filtres médians et morphologiques

On souhaite mettre en évidence le réseau vasculaire et le disque optique d'une image de fond d'oeil.

## Élimination du bruit par filtrage médian

**Q1.** Chargez et affichez l'image `fundus_bruit.png`. Quel est le nom du bruit qui détériore cette image de la rétine ?

Pour la suite de l'exercice nous ne nous intéresserons qu'au canal vert de l'image.

**Q2.** À l'aide d'un filtrage médian (avec un disque pour élément structurant), filtrez le bruit identifié à la question précédente. Trouvez un rayon de disque approprié pour le filtrage. Affichez côte à côte le canal vert de l'image avant et après le filtrage.

**Q3.** Filtrez le bruit identifié à la question précédente avec les filtres gaussiens utilisés à l'exercice précédent (convolution de l'image avec un filtre gaussien). Le filtrage gaussien présenté dans l'exercice précédent aurait-il été un meilleur ou moins bon choix pour cette tâche ? Pourquoi ?

Les traitements présentés dans les deux sections suivantes seront appliqués sur la version filtrée du canal vert de l'image notée  $I_V$ .

## Rehaussement de vaisseaux

On se propose d'implémenter un algorithme de rehaussement du réseau vasculaire rétinien. En notant  $F_m$  un filtre médian avec un large élément structurant (un disque de rayon 12), l'image prétraitée  $I_P$  est calculée par l'expression

$$I_P = I_V - F_m(I_V).$$

**Q4.** Calculez et affichez le résultat du filtrage médian  $F_m(I_V)$  et du prétraitement  $I_P$ . Cette correction d'illumination est-elle efficace ? Les vaisseaux sont-ils un contenu hautes ou basses fréquences spatiales ? Même question pour le résultat du filtrage médian  $F_m(I_V)$ .

**i** On ne vous demande pas ici de forcer l’affichage entre 0 et 1. Vous pouvez laisser les comportements par défaut de la fonction `imshow` et uniquement fournir en arguments l’image à afficher et la colormap à utiliser.

## Segmentation du disque optique

On se propose de segmenter le disque optique (voir figure 6) avec un seuil et des opérations morphologiques.

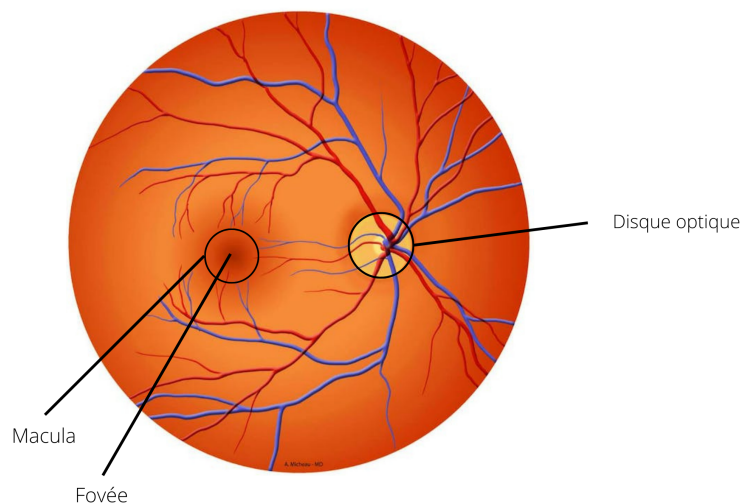


FIGURE 6 – Représentation d’un fond d’oeil et les structures anatomiques le composant.

**Q5.** Implémentez une fonction `binariser(img, seuil)` qui met à 0 tous les pixels se trouvant en dessous du seuil, et à 1 tous les pixels se trouvant sur le seuil ou au-dessus. Cette fonction prend en entrée une image et un seuil, puis retourne l’image binaire. Binarisez votre image avec un seuil de 0.6 et affichez-la.

**Q6.** Vous devriez constater deux erreurs que vous devrez corriger avec une fermeture et une ouverture :

1. Les vaisseaux émergeant du disque optique forment des trous dans sa segmentation.
2. Un artefact est détecté à tort dans le coin supérieur droit de l’image.

Refermez les trous causés par les vaisseaux à l’aide de l’opération morphologique adéquate. Vous devrez choisir un élément structurant adapté. Effacez l’artefact supérieur droit sans



altérer la segmentation du disque avec l'autre opération morphologique. Une fois encore, choisissez un élément structurant adapté.

**i** Les opérations d'ouverture et de fermeture sont définies dans le chapitre 4 de vos notes de cours. Vous pouvez également aller voir leur définition et implémentation [ici](#) et [ici](#).

**Q7.** On veut finalement extraire la position du centre du disque optique et son rayon. Pour cela, on procède à des amincissement de sa segmentation en utilisant une érosion avec un élément structurant en disque de rayon variable. Augmentez le rayon de l'élément structurant jusqu'à ce que la segmentation disparaisse totalement. Déduisez le rayon et la position du centre du disque optique.

**?** La fonction `np.any(I)` renvoie vrai si tous les pixels de `I` valent 0. La fonction `np.nonzero(I)` renvoie les coordonnées des pixels *différents* de 0 dans `I`.