

# Travail pratique #1

## Traitements d'images élémentaires

INF600F - Traitement d'images

Hiver 2022

### Table des matières

Pondération . . . . .	1
Exercice 1 : Imagerie de la rétine & Message secret	1
Exercice 2 : Avatar UQAM	2
Exercice 3 : Code-barres mystère	4

L'objet de ce travail est de mettre en pratique certains des traitements élémentaires dans le domaine spatial qui ont été présentés en cours, tout en permettant une familiarisation avec Python et ses nombreuses bibliothèques. Ce travail comporte 3 exercices, et il vaut pour 12.5% de la note finale.

Les données et le notebook à utiliser pour effectuer ce travail pratique se trouvent dans l'archive ZIP de ce TP disponible sur le site web du cours. Consultez également le document **Instructions générales pour les travaux pratiques** sur Moodle pour connaître les exigences du rapport de laboratoire.

### Pondération

- Exercice 1 : Imagerie de la rétine & Message secret (5 pts)
- Exercice 2 : Avatar UQAM (8 pts)
- Exercice 3 : Code-barres mystère (12 pts)
- **Total** : 25 points

### Exercice 1 : Imagerie de la rétine & Message secret

*Les images à utiliser pour cet exercice sont `tp1_ex1_img1.tiff` et `tp1_ex1_img2.tiff`.*

Il s'agit d'une image des vaisseaux sanguins au fond de la rétine. Les deux images ont été créées à partir de la même image de référence, mais un message secret a été caché dans l'intensité de la seconde image.

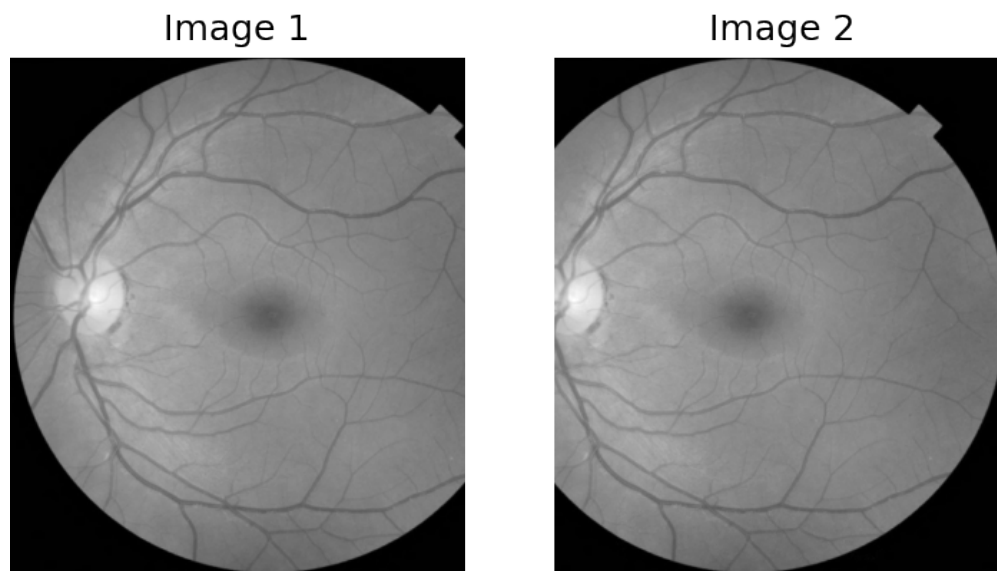


FIG. 1 : Images à traiter pour l'exercice 1

- Utilisez Python pour lire les images, les aligner et pour les comparer afin de trouver le message secret caché dans la seconde image. (*Indice* : Vous cherchez ici la différence entre les deux images.)
- Affichez tous les résultats que vous jugerez pertinents pour inspecter les images et pour identifier le message secret.
- **Question** : Pourquoi est-ce que le message secret est imperceptible à votre avis ?
- **Question** : Comment a été créée l'image contenant le message secret selon vous ?

## Exercice 2 : Avatar UQAM

*Vous devez utiliser l'image `tp1_ex2_logo.tiff` ainsi qu'une photo de vous-même pour cet exercice.*

Pour rendre les conférences Zoom plus conviviales, vous décidez de créer une photo de profil qui sera affichée à vos interlocuteurs lorsque votre caméra est fermée. Pour vous donner un petit défi et pour tester vos connaissances en traitement d'images, vous décidez d'écrire une fonction Python pour créer automatiquement cette image.

Votre fonction doit :

- Rogner l'image pour obtenir une photo carrée de taille `s` centrée sur la position `centre=(r,c)` correspondant à votre nez ou au centre de votre visage.
- Conserver le contraste et les couleurs originales
- Ajouter le logo de l'UQAM en vous servant du gabarit `tp1_ex2_logo.tiff`
- Afficher l'image initiale et l'image modifiée.
- Sauvegarder l'image sous format JPG.

Voici le squelette de fonction que vous devez compléter



FIG. 2 : Exemple de résultat attendu pour l'exercice 2.

```
def creation_profil(input_file, output_file, centre, s):
    """TP1/Ex3 : Création d'une photo de profil UQAM.
    Paramètres
    -----
    input_file : str
        Chemin vers l'image originale
    output_file : str
        Chemin vers l'image modifiée
    centre : tuple (2,)
        Tuple ou liste de longueur 2 contenant les coordonnées (r,c) du centre du visage
    s : int
        Taille de la photo de profil en pixel
    """
    pass # Remplacez pass par votre code
```

- Créez une fonction respectant les contraintes indiquées.
- Ensuite, en utilisant une caméra, prenez une photo de face dont le format ressemble à l'exemple fourni avec ce TP (si vous effectuez le TP en équipe, prenez 1 photo par membre de l'équipe). Nommez la photo "profil\_<nom\_etudiant>.jpg"
- Utilisez votre fonction pour créer une image avatar que vous nommerez "profil\_<nom\_etudiant>\_uqam.jpg"
- Fournissez les photos originales et modifiées avec votre notebook sur Moodle pour la correction.
- **Note** : Pour faciliter l'identification du centre de votre visage ( $cx, cy$ ), je vous conseille

d'utiliser la fonction `plt.scatter` pour dessiner un marqueur par-dessus une image affichée avec `plt.imshow`.

- **Note** : Vous pouvez utiliser la fonction `skimage.transform.resize` pour ajuster la taille de l'image `tp1_ex2_logo.tiff` à la taille de votre photo de profil rognée.
- **Note** : À la fin de ce TP, vous pourriez utiliser la photo de profil générée pour vos appels Zoom, vos professeurs vous en seront reconnaissants ! Cela nous évitera d'observer des grilles d'avatar vides lors des cours.

## Exercice 3 : Code-barres mystère

L'image à utiliser pour cet exercice est `tp1_ex3_livre_mystere.png`.

Vous avez trouvé un code-barres abandonné à l'UQAM. Ce code-barres s'est décollé d'un livre emprunté à la bibliothèque. Vous souhaitez identifier le livre auquel ce code appartient.

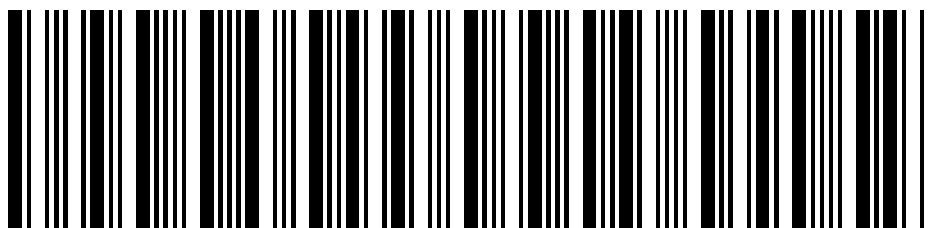


FIG. 3 : Code-barres à décoder

La norme suivie pour créer le code-barres est [Codabar](#), une vieille convention utilisée dans les bibliothèques. Pour cette convention, les symboles [0-9] et \$, - sont représentés par 4 lignes et 3 espaces. Les lignes/espaces peuvent être larges ou étroits, et chaque symbole d'un code-barres est séparé par un espace étroit. Voici la correspondance entre les patrons de barres-espaces et chaque symbole possible pour cet exercice.

Symbole	Barres	Espaces
0	0001	001
1	0010	001
2	0001	010
3	1000	100
4	0100	001
5	1000	001
6	0001	100
7	0010	100
8	0100	100
9	1000	010
-	0010	010

Symbole	Barres	Espaces
\$	0100	010

**Note :** “0” indique une barre ou un espace mince, et “1” indique une barre ou un espace large. Par exemple, pour le symbole "0" le code barre suivant sera utilisé :

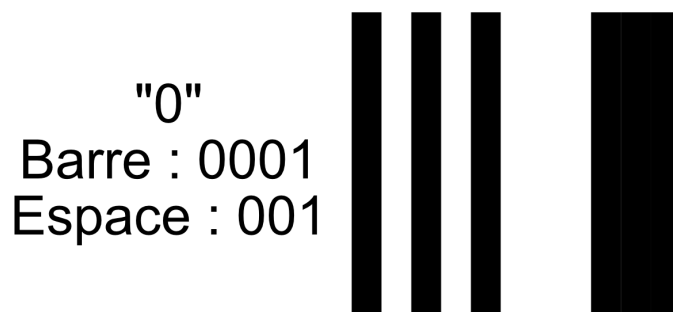


FIG. 4 : Exemple de codabar pour "0"

- Développez un décodeur de code-barres qui reçoit une image, extrait le niveau d'intensité le long d'une ligne et qui retrouve ensuite l'identifiant ISBN-13 de ce livre.
- Utilisez votre algorithme pour identifier l'ISBN associé au code-barres de l'image `tp1_ex3_livre_mystere.png`.
- Quels sont le titre et les auteurs de ce livre (*recherchez l'ISBN dans Google*) ?