

Travail pratique #1

Traitements d'images élémentaires

INF600F - Traitement d'images

Automne 2024

Table des matières

Pondération	1
Exercice 1 : Convolution	1
Exercice 2 : Code-barres mystère	2
Instructions détaillées	3
Références	3

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 2 exercices, et il vaut pour 10% 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.

Pondération

- **Exercice 1** : Convolution (5 pts)
- **Exercice 2** : Code-barres mystère (5 pts)
- **Total** : 10 points

Exercice 1 : Convolution

Développez une fonction Python qui effectue une convolution entre une image en niveaux de gris et un noyau de convolution. Les deux matrices (image et noyau) sont données en entrée de votre fonction. Testez votre convolution avec l'image `tp1_ex1.tif` et avec 3 noyaux de convolution différents: (1) un filtre moyenneur de taille 5x5, (2) un filtre gaussien d'écart-type $\sigma = 2$, et (3) un filtre de Sobel horizontal.

La fonction doit recevoir un argument optionnel pour contrôler le type de remplissage pour les conditions de frontières. Les deux méthodes à implémenter sont un remplissage de zéro (*zero padding*, option par défaut) et un remplissage de type miroir (*mirror padding*).

Affichez les images originales et les images filtrées pour chaque noyau. Assurez-vous que les valeurs des pixels des images filtrées soient de même type que l'image originale (par exemple, si l'image originale est en `uint8`, l'image filtrée doit aussi être en `uint8`). Enregistrez les images filtrées dans un fichier `.png` pour les inclure dans le fichier compressé `.zip` de votre remise et dans votre rapport Jupyter.

Note : Pour cet exercice, vous devez écrire votre propre fonction de convolution. Vous ne pouvez pas utiliser de fonction de convolution prédéfinie de bibliothèques comme OpenCV, scikit-image ou scipy

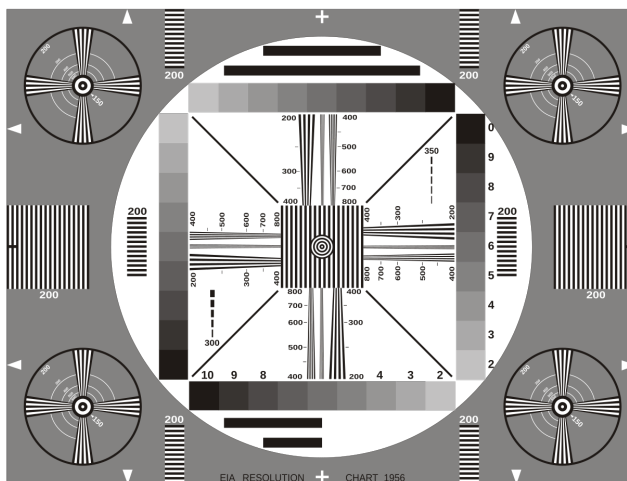


Figure 1: Image à traiter pour l'exercice 1

Voici un squelette de fonction que vous devez compléter :

```
def convolution(image:np.ndarray, noyau:np.ndarray, padding:str='zero') -> np.ndarray:
    """Convolution d'une image en niveaux de gris avec un noyau de convolution.
    Paramètres
    -----
    image : ndarray
        Image en niveaux de gris de taille (H, W)
    noyau : ndarray
        Noyau de convolution de taille (h, w)
    padding : str, optionnel
        Type de remplissage pour les conditions de frontière.
        'zero' pour un remplissage de zéro, 'mirror' pour un remplissage miroir.
    Retour
    -----
    ndarray
        Image convoluée de taille (H, W)
    """
    # Votre code ici

    return None # Remplacez `None` par le résultat de votre fonction
```

Exercice 2 : Code-barres mystère

L'image à utiliser pour cet exercice est *tp1_ex2.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.

La norme suivie pour créer le code-barres est *Codabar* (<https://en.wikipedia.org/wiki/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 étroites, 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

Symbole	Barres	Espaces
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
\$	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 :

- 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 fournie.
- Quels sont le titre et les auteurs de ce livre (*recherchez l'ISBN dans Google*) ?

Instructions détaillées

- Les travaux pratiques doivent être effectués en équipe de deux maximum. Vous pouvez utiliser l'activité Moodle « Choix d'une équipe pour les travaux pratiques » pour indiquer les membres de votre équipe. (vous devez choisir un groupe même si vous travaillez seul.e)
- Le compte rendu doit comporter une réponse concise, mais complète à chacune des questions, accompagnée au besoin des courbes, figures et images appropriées. Vous devez également expliquer vos algorithmes avec des commentaires et/ou des cellules de texte Markdown dans le notebook.
- Portez une attention particulière aux illustrations (visibilité, annotations, unités et pertinence).
- Vous devez soumettre le Jupyter Notebook (.ipynb), et une version PDF du notebook, les deux fichiers combinés dans un fichier .zip.
- Les fichiers additionnels à inclure dans le zip sont indiqués dans l'énoncé de chaque TP.
- Le compte rendu doit être remis au plus tard à la date indiquée dans l'activité Moodle.
- Le travail doit être remis par un seul des membres du groupe. Si tel n'est pas le cas, la version la plus récente du travail remis est prise en compte.
- **Retards :** 10% de la note sera retranché par jour de retard. Après 5 jours de retard, aucune remise ne sera acceptée.
- **Qualité du français :** La note des travaux pratiques sera réduite de 1% par faute, pour un maximum de 10% par travail.
- **IA :** L'utilisation d'outils de génération de contenu (ex. ChatGPT) est interdite pour ce travail et sera considérée comme une infraction académique.

Références

- L'image de l'exercice 1 est une cible de résolution EIA 1956 provenant de [Wikimedia](#).

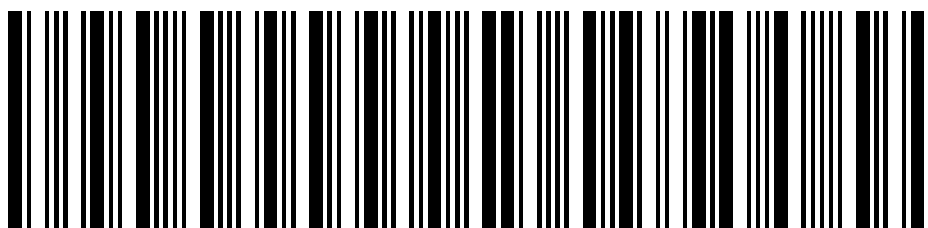


Figure 2: Code-barres à décoder

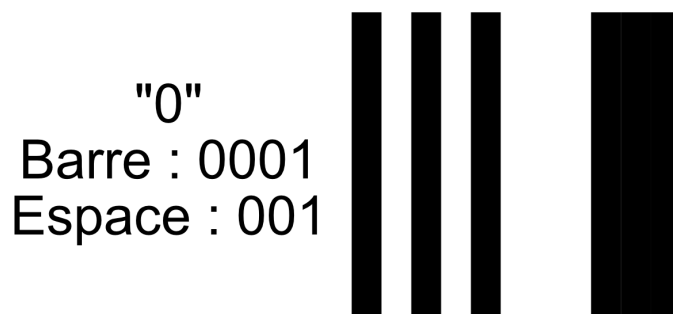


Figure 3: Exemple de codabar pour "0"