

Pour commencer : Créez un nouveau projet Pycharm et utilisez un répertoire source différent par exercice

Exercice 1 : Traitement d'images avec Numpy- Détection de contours avec des noyaux de convolution

Objectif : Manipuler des tableaux Numpy, et en profiter pour comprendre le principe des filtres en imagerie numérique et la détection de contours à partir de la convolution avec des noyaux

Commencer par lire [cet article](#) qui explique le principe de filtre de convolution

Partie 1 :

Dans le fichier `convolution.py` implémenter la fonction `convolution()`, selon le principe de la fenêtre glissante expliquée dans l'article.

(Observer la différence entre votre convolution et celle de Numpy, elle devrait être explicable)

Partie 2 :

Dans un fichier `exercice1.py` implémentez les étapes suivantes

1. Charger une image

- Utilisez une petite image en niveaux de gris (par exemple `data.camera()` de `skimage.data` pour commencer).
- Affichez-la avec `matplotlib`.

2. Appliquer un filtre de flou

On définit le noyau de moyenne (blur) 3×3 :

$$K_{flou} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Convoquez (en utilisant `signal.convolve2d`) l'image avec ce noyau.
- Affichez le résultat et observez l'effet.

3. Détecter les gradients

On utilise les noyaux de Sobel :

Horizontal (dérivée par rapport à X) :

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Vertical (dérivée par rapport à Y)

$$K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Convoluez l'image floutée avec K_x puis séparément avec K_y .
- Affichez les deux images obtenues G_x et G_y (gradient horizontal et vertical).

4. Calculer l'amplitude du gradient

Combinez les deux images précédentes en calculant le gradient :

$$G = \sqrt{G_x^2 + G_y^2}$$

- Affichez cette nouvelle image G, qui fait apparaître les contours.

5. Appliquer un seuillage

- Appliquez un seuil (par exemple 100). Autrement dit, transformez l'image en binaire : pixels blancs = contour détecté, pixels noirs = fond d'image, de sorte d'afficher les contours en noir et le fond en blanc

6. Préparer la remise

- Enlever les affichages précédents et afficher les 6 images simultanément sur 3 lignes et 2 colonnes avec des titres à chaque image pour présenter les résultats successifs

- Mettez de l'ordre dans votre code (utiliser des fonctions pour regrouper les opérations qui doivent l'être, etc.)
- Choisissez d'autres images de votre choix pour expérimenter les effets du seuil et sélectionnez en une pour la remise.
- Essayez d'améliorer le processus, en réfléchissant aux paramètres sur lesquels vous pourriez jouer et expliquez vos choix en commentaires!

Pour ceci, basez-vous sur les questions de réflexions suivantes (non évaluées en tant que telle mais utile pour optimiser votre détection de bord)

1. Pourquoi applique-t-on un flou avant la détection de contours, quel est son effet?
2. Que se passe-t-il si vous appliquez un seuil plus bas ou plus haut ?

Exercice 2 : Manipulation symbolique avec SymPy

Objectifs :

- *Déclarer des symboles et expressions symboliques*
- *Calculer des dérivées et intégrales*
- *Factoriser et simplifier des expressions*
- *Résoudre des équations et systèmes*
- *Tracer des fonctions*

1. On considère

$$f(x) = 3x^3 - 5x^2 + 2x - 7$$

- a) Afficher convenablement à la console la fonction f ,
- b) Calculez la dérivée de $f(x)$,
- c) Résolvez l'équation $f'(x) = 0$ pour trouver les points critiques de f ,
- d) Évaluez f aux points critiques pour obtenir les valeurs correspondantes,
- e) Calculer l'intégrale indéfinie de f ,
- f) Calculer l'intégrale définie de f entre $x = 0$ et $x = 2$,
- g) Tracer les graphes de $f(x)$ et $f'(x)$ sur l'intervalle $[-2,3]$ de deux couleurs différentes et bien identifiées,
- h) Ajoutez la mise en évidence de points critiques et de leurs coordonnées,
- i) Ajoutez le tracé de la tangente à f au point $x = -1$.

2. On considère le système :

$$\begin{cases} 2x + 3y + z = 2 \\ -x + 2y + 3z = -1 \\ -3x - 3y + z = 0 \end{cases}$$

- a) Résoudre le système avec SymPy .
 - b) Vérifier la solution.
3. Soit une nouvelle fonction

$$g(x) = 6x^5 - 9x^4 - 49x^3 + 87x^2 - 17x + 30$$

- a) Factorisez $g(x)$
- b) Développez de nouveau pour vérifier.