

Dernière mise à jour	Informatique	Denis DEFAUCHY
14/05/2022	7 - Matrices de pixels et images	TD 7-10 – Avant – Après

Informatique

7

Matrices de pixels et images

TD7-10

Avant – Après

Dernière mise à jour	Informatique	Denis DEFAUCHY
14/05/2022	7 - Matrices de pixels et images	TD 7-10 – Avant – Après

Contexte

On souhaite réaliser un outil qui permette de réaliser un effet « Avant – Après » à l'aide d'un curseur à l'aide de deux photos, comme le proposent des applications pour smartphone payantes.

Affichage des images

Afin d'assurer un fonctionnement rapide sur tous les ordinateurs, je vous mets à disposition un dossier à télécharger COMPLETEMENT, soit le dossier contenant tous les fichiers, et non les fichiers pris séparément.

Dossier_Partagé

Enregistrer dans Dropbox Télécharger

Nom

7-10 - TD - Avant Après

Sans ouvrir le dossier, faite juste « Télécharger – Téléchargement direct » puis mettez ce dossier dans votre répertoire personnel.

LIEN

Si le téléchargement est sous forme de Rar, Zip... Pensez à dézipper l'archive afin d'avoir le dossier voulu !

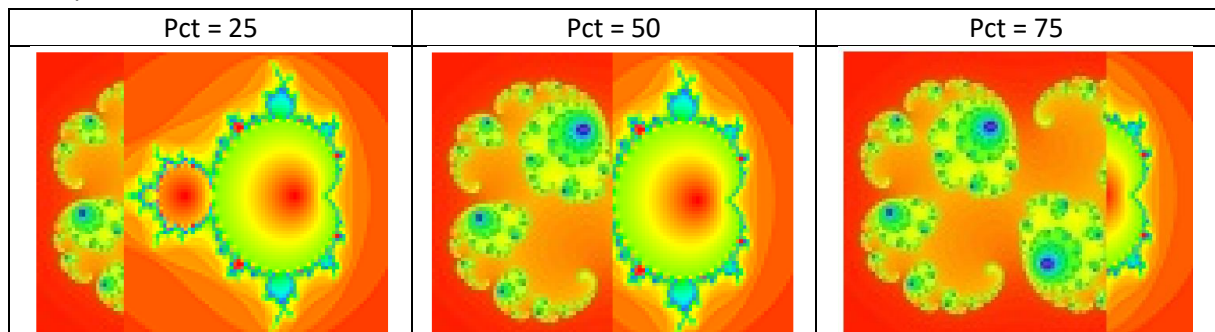
Question 1: Télécharger et exécuter le code fourni qui affichera les deux images fournies sous Python

Réalisation du montage

On souhaite créer une fonction `f_Traitement(Im_Av,Im_Ap,Pct)` prenant en argument les deux images (format array) `Im_Av` (Image avant) et `Im_Ap` (Image après) et le pourcentage `Pct` (entier dans $[0,100]$) permettant de générer une image contenant

- à gauche : `Pct %` de l'image `Im_Ap`
- à droite : $100 - \text{Pct} \%$ de l'image `Im_Av`

Exemples :



Question 2: Mettre en place la fonction `f_Traitement(Im_Av,Im_Ap,Pct)`

Question 3: Ecrire les lignes de code permettant de réaliser et d'afficher le montage

Dernière mise à jour	Informatique	Denis DEFAUCHY
14/05/2022	7 - Matrices de pixels et images	TD 7-10 – Avant – Après

Utilisation d'un curseur interactif

Vous avez à disposition dans la seconde partie du fichier élève partagé le code suivant :

```
from tkinter import *
master = Tk()
valeur = StringVar()
def miseAJourCurseur(event):
    resultat = curseur.get()
    valeur.set(resultat)
    print(resultat)

master.wm_title("Fenetre")
master.geometry("300x100")
curseur = Scale(master, from_=0, to=100, orient=HORIZONTAL,
command=miseAJourCurseur)
curseur.pack()
texte = Label(master, textvariable=valeur)
texte.pack()
mainloop()
```

Il a volontairement été placé dans une partie en chaîne de caractères (''' '''). Il faudra donc supprimer les guillemets.

Question 4: Exécuter ce code et visualiser ce qu'il se passe (vous noterez qu'une fenêtre avec un curseur s'ouvre)

Question 5: Modifiez ce code afin qu'il permette d'afficher en direct, le montage des photos avant/après selon la position du curseur

Remarque : pour que cela soit fluide, j'ai volontairement proposé dans ce TP des images de faible qualité

Pour aller plus loin

Vous pouvez maintenant utiliser des images personnelles en remplaçant dans le dossier les images Avant et Après. Pour cela, il faudra par exemple, avec Paint sur Windows :

- Les réenregistrer au format BMP
- Les redimensionner pour qu'elles aient exactement le même nombre de lignes et colonnes

Par ailleurs, dans le but de générer une animation de type gif animé ou vidéo, vous pourrez générer des images en utilisant la fonction f_Affiche_Save suivante :

```
def f_Affiche_Save(fig, image, Pct):
    plt.figure(fig)
    plt.clf()
    plt.imshow(image)
    plt.axis('off')
    plt.savefig("Pct"+str(Pct)+".png")
```

Remarques :

- Il ne faut pas de show dedans, sinon il y aura des problèmes de redimensionnement
- Pour afficher et sauvegarder, utiliser f_Affiche et f_Affiche_Save sur 2 figures différentes