

	Section	2ème année Ingénierie des Données et Systèmes Décisionnels	Année Universitaire :
	Matière	TP Traitement Image & Vidéo	2022 - 2023

## ***TP N°1 : Traitement de base des images avec Jupyter Notebook***

### **Introduction:**

Notre monde d'aujourd'hui regorge de données et les images en constituent une partie importante. Cependant, pour pouvoir être utilisées, ces images doivent être traitées. Le traitement d'images consiste donc à analyser et à manipuler une image numérique principalement dans le but d'en améliorer la qualité ou d'en extraire des informations qui pourraient ensuite être utilisées.

Python devient un choix judicieux pour de telles tâches de traitement d'images. Cela est dû à sa popularité croissante en tant que langage de programmation scientifique et à la disponibilité gratuite de nombreux outils de traitement d'images de pointe dans son écosystème.

Ce TP comporte deux parties. La première partie est essentiellement dédiée à la description du logiciel Jupyter Notebook. La deuxième partie est consacrée à la manipulation des images.

### **I. Prise en main de Jupyter Notebook**

(cf. support de cours initiation Jupyter Notebook fourni avec l'énoncé du TP)

### **II. Manipulation**

#### ***Exercice 1: Convertir une image RVB en niveaux de gris***

La transformation d'une image en couleurs en niveaux de gris s'effectue avec la commande ```color.rgb2gray```.

1. Lire l'image "lena.jpg" et l'afficher en couleurs RVB.
2. Convertir l'image Lena en niveaux de gris
3. Sauvegarder sous le nom '*lena\_Gray*' sous format ".jpg" .

#### ***Exercice 2: Inverser les niveaux de gris***

Pour inverser les contrastes d'une image en niveaux de gris, il suffit d'appliquer à chaque pixel  $x$  la valeur  $(255 - x)$ . Le blanc devient noir et vice-versa.

1. Inverse les niveaux de gris de l'image *lena\_Gray*.
2. Sauvegarder sous le nom '*lena\_Gray\_Inv*' sous format ".jpg"..

### ***Exercice 3: Affichage multiple des canaux RGB d'une image en couleurs***

Etant donnée une image en couleurs RVB, on peut afficher chaque bande (canal) séparément en fonction de ses composantes rouge, verte et bleue. Par exemple, la bande rouge est le premier canal de l'image 3D, on peut l'afficher comme suit:

```
bande_rouge = image_couleur[:, :, 0] # ou image_couleur[..., 0]
```

Pour l'afficher, il suffit de taper:

```
plt.imshow(bande_rouge);
```

```
'''
```

Python offre la possibilité d'afficher plusieurs images simultanément. On peut utiliser ```matplotlib.subplots```:

```
fig, axes = plt.subplots(1, 3)
axes[0].imshow(image0)
axes[0].set_title('image0')
axes[1].imshow(image1)
axes[1].set_title('image1')
axes[2].imshow(image2)
axes[2].set_title('image2')
plt.show()
```

1. Lire l'image "lena.jpg" et l'afficher en couleurs RVB.
2. Séparer les trois canaux correspondants à la couleur rouge, verte et bleue et les afficher dans la même figure.
3. Sauvegarder chaque canal sous format ".jpg".

### ***Exercice 4: Convertir une image RVB en HSV***

La transformation Teinte saturation luminance (en Anglais, HSV) est l'une des transformations les plus connues dans les espaces des couleurs. La commande ```color.rgb2hsv``` permet de passer d'un espace RVB par exemple en espace HSV.

1. Lire l'image "lena.jpg" et l'afficher en couleurs RVB.
2. Convertir l'image Lena de l'espace RVB vers l'espace HSV.
3. Séparer les trois canaux correspondants à la Teinte (H), saturation (S) et luminance (V) et les afficher dans la même figure.
4. Sauvegarder chaque canal (H, S, V) sous format ".jpg".

### ***Exercice 5: Mise en place d'un cadre autour d'une image***

1. Lire l'image "lena.jpg" et l'afficher en couleurs RVB.
2. Écrire une fonction ***cadre\_noir(im,ep)*** prenant en entrée une image ***im***, ainsi qu'un (petit) entier ***ep***, renvoyant une nouvelle image, identique à la précédente mais dont les pixels situés sur les bords de l'image ont été remplacés par des pixels noirs, sur une épaisseur ***ep***.
3. Une fois votre fonction écrite, appliquez-la à Léna avec une épaisseur de 5.

### ***Exercice 6: Rajout d'un cadre autour de l'image***

1. Lire l'image "lena.jpg" et l'afficher en couleurs RVB.
2. Écrire une fonction ***rajout\_cadre(im,ep)*** fonctionnant comme précédemment, mais retournant une image de taille ***(h + 2ep, l + 2ep)*** (si l'image initiale a pour taille ***(h, l)***), le cadre ayant été ajouté à l'extérieur plutôt que par modification des pixels du bord.
3. Une fois votre fonction écrite, appliquez-la à Léna avec une épaisseur de 20.