

TP1 - Premières manipulations des images

ayoub.karine@u-paris.fr

Préambule

OpenCV (Open Source Computer Vision) est une bibliothèque libre de traitement d'images. Plusieurs langages sont supportés par cette bibliothèque. Vous allez utiliser cette bibliothèque dans le TP1 et le TP2 en utilisant le langage Python.

Pour utiliser OpenCV, vous pouvez l'installer en local (<https://opencv.org/get-started/>) ou utiliser un notebook à l'instar de Google Colab (<https://colab.research.google.com/>) dans lequel il est déjà installé (le code va être interprété par un serveur Google). Vous allez utiliser Google Colab dans tous les TP. Les images à utiliser dans ce TP sont à télécharger depuis Moodle.

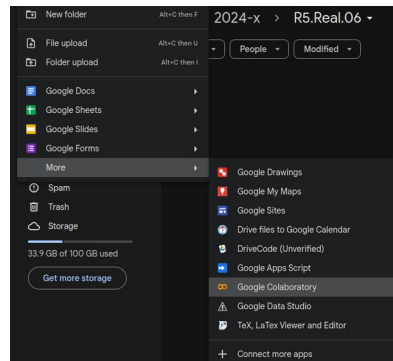
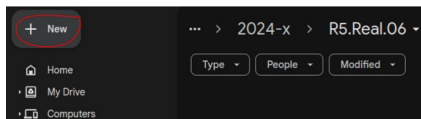
Pour les différentes questions de ce TP, il est demandé de se référer à la documentation officielle d'OpenCV 4.10 disponible via ce lien : <https://docs.opencv.org/4.10.0/>

Vous serez amené aussi à utiliser la documentation officielle de la bibliothèque NumPy :

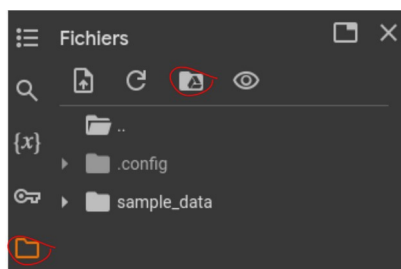
<https://numpy.org/>

Préparation de l'environnement de travail

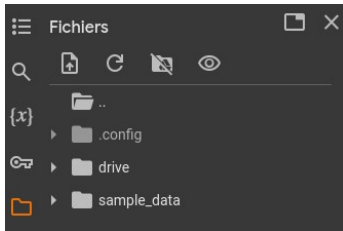
1. Connectez-vous à votre compte Gmail (créez un compte si vous n'en avez pas déjà un)
2. Depuis votre Google Drive,
 - a) dans le chemin de votre choix, créez un dossier «R5.Real.06» dans lequel vous allez stocker les solutions de tous les TP
 - b) à l'intérieur de ce dossier, créez un dossier «TP1» (il est conseillé de créer un dossier par TP)
 - c) dans le dossier «TP1», créez un notebook GoogleColab, nommé «tp1»



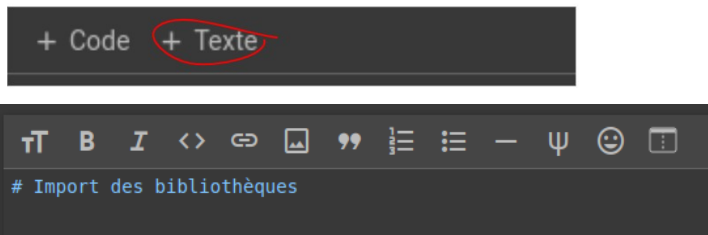
- d) dans le dossier «TP1», créez un dossier «images» dans lequel vous stockez les images disponibles dans la page moodle associées à «Sensibilisation à la programmation multimédia (R5.Real.06)»
- e) pour connecter le contenu de Google Drive avec Google Colaboratory, cliquez sur l'icône suivante (vous allez remarquer que mon notebook est en dark mode, si vous voulez l'activer cliquez sur Tools>Settings>Site>Theme>dark) :



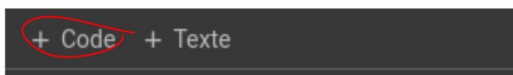
Comme résultat vous aurez l'affichage ci-dessous :



2. Depuis GoogleColab,
 - a) créez une section «Import des bibliothèques»



- b) créez une cellule dans laquelle vous collez le code suivant



```
import cv2 # import de la bibliothèque OpenCV
import numpy as np
import matplotlib.pyplot as plt
```

et exécutez la cellule en cliquant sur (ou ctrl+entrée)



Pour tous les TP, il est demandé de créer une cellule de code par exercice.

Premières manipulations d'une image en niveaux de gris

1. Créez une section «Premières manipulations d'une image en niveaux de gris»
2. Utilisez la fonction `imread` pour lire l'image «**checkerboard_18x18.png**» et stockez le résultat dans une variable. Comme deuxième argument de cette fonction, utilisez la valeur 0 pour lire une image en niveaux de gris (la valeur par défaut est 1 = image couleur). Pensez à mettre le bon chemin de l'image avec un clic droit sur l'image (depuis Google Colab) et puis cliquez sur copier le chemin de l'image
3. Affichez le type de cette variable avec la fonction `type`
4. Affichez la forme matricielle de l'image avec la fonction `print` pour visualiser les valeurs de chaque pixel
5. Affichez la taille (hauteur, largeur) de l'image avec la fonction `shape` de la bibliothèque NumPy
6. Affichez l'image en utilisant ce code :
`plt.imshow(cb_img, cmap="gray")`
7. Comparez le résultat visuel et la forme matricielle de l'image
8. Affichez les valeurs du premier pixel de l'image (l'origine de l'image est toujours en haut à gauche) ainsi que celui d'indice [0,6]
9. Affectez la valeur 200 aux pixels d'indices [2,2], [2,3], [3,2] et [3,3] et affichez l'image résultante

10. L'image «checkerboard_18x18.png» est une image binaire (les pixels sont soit blanc soit noir). Pour voir un exemple d'une image avec plus de valeurs de niveaux de gris, affichez la forme matricielle et visuelle de l'image «**checkerboard_fuzzy_18x18.jpg**»

Premières manipulations d'une image couleur

L'image à traiter dans cette première partie est «**New_Zealand_Lake.jpg**»

1. Lisez cette image et stockez le résultat dans une variable
2. Affichez ses dimensions (hauteur, largeur et le nombre de canaux)
3. Affichez l'image. Vous allez remarquer que les couleurs ne sont pas les mêmes si vous la comparez avec l'image stocké dans votre GoogleDrive. En effet, OpenCV stocke chaque image en format BGR (Blue Green Red) tandis que la bibliothèque Matplotlib affiche l'image en utilisant le format RGB
4. Pour la conversion BGR->RGB, utilisez la fonction `cvtColor` et affichez l'image résultante
5. Affichez la forme matricielle de chaque canal RGB. Utilisez la fonction `split`
6. Affichez l'image correspondante à chaque canal RGB
7. Appliquez une conversion RGB->HSV et affichez l'image résultante
8. Affichez la forme matricielle de chaque canal HSV
9. Affichez l'image correspondante à chacun des trois canaux (H, S, V) de cette image
10. Ajoutez 100 à toutes les valeurs du canal «H» et affichez l'image HSV résultante. Utilisez les fonctions `split` et `merge`
11. Enregistrez l'image HSV résultante dans votre Gdrive en utilisant la fonction `imwrite`

L'image à traiter dans cette deuxième partie est «**New_Zealand_Boat.jpg**»

12. Appliquez une conversion BGR->RGB et affichez l'image résultante
13. Sélectionnez la région d'indices [200:400, 300:600], stockez le résultat dans une nouvelle variable et affichez le résultat
14. Retournez l'image de manière verticale, horizontale et verticale/horizontale en utilisant la fonction `flip`
15. Redimensionnez l'image originale en taille 200x100 (HxW) en utilisant la fonction `resize` et affichez le résultat
16. Annotez l'image avec une ligne, un rectangle, un cercle et un texte en utilisant les fonctions `line`, `circle`, `rectangle` et `putText`