

Nom Prénom 1 :
Nom Prénom 2 :
Nom Prénom 3 :
Nom Prénom 4 :

IA pour l'image

TD_Conv-images_Python et OpenCV E5FI_5I-IN7

1. Introduction

1.1 Objectif

L'objectif de ce TD est de prendre en main la librairie générale (OpenCV) de traitement d'image :

- Lire et traiter une image, vidéo ou le flux d'une Webcam avec OpenCV
- Appliquer les compétences en langage Python
- Programmer certains opérateurs de traitement d'image avec OpenCV

1.2 OpenCV

OpenCV est une librairie de **traitement d'image** qui offre un bon éventail de fonctions intégrées et puissantes. Une description détaillée de ces fonctions est disponible en ligne à l'adresse suivante : <http://docs.opencv.org/master>

Dans ce TD nous allons récupérer le flux vidéo d'une webcam tout en analysant chaque commande pour bien comprendre leur utilité.

2. Opérations de base

1. Création et manipulation d'images avec Python et OpenCV.

Avec les versions récentes de la librairie OpenCV, les images sont représentées sous forme de matrice de valeurs numériques.

Une image en couleur est enregistrée en mémoire sous la forme d'une matrice 3D ➔ Image [B,G,R]

Tapez et exécutez le code suivant pour créer et afficher une image noire de taille 128*128 :

Rostom Kachouri _ E5FI_5I-IN7

```

import cv2
import numpy as np

img = np.zeros((128,128,3),np.uint8)
print(img) # pour afficher dans la console

cv2.imshow("Image",img)
cv2.waitKey(0)
cv2.destroyAllWindows("Image")

```

2. Modifiez ce code pour afficher trois images de taille 512*512 respectivement de couleur Rouge, Verte et Bleue.

3. Insérez des lignes, rectangles, cercles et du texte dans ces images ;

Pour cela, les fonctions « *cv2.line* », « *cv2.rectangle* », « *cv2.circle* », et « *cv2.putText* » seront utilisées. Vous devez identifier et fixer les paramètres de ces fonctions.

4. Commentez votre programme et expliquez les paramètres utilisés et illustrez des imprime-écrans de vos rendus

3. Traitement d'images

Mise en œuvre du TD

1. Dans votre répertoire de travail, télécharger l'archive compressée « RK_Python-OpenCV_CONV.rar ».
2. Décompressez l'archive, il contient trois programmes python pour lire une image, une vidéo et le flux d'une webcam ainsi qu'un dossier « RK_Data » contenant les ressources requises par ces programmes.
3. Tester ces trois programmes et apporter les modifications nécessaires pour afficher d'autres images et vidéos.
4. Commentez votre programme et expliquez les paramètres utilisés et illustrez des imprime-écrans de vos rendus

Conversion couleur

1. Pour convertir une image couleur en une image niveau de gris, on applique souvent la formule suivante : $\# \text{ Gray} = 0.2989 R + 0.5870 G + 0.1140 B$

Ecrire un programme permettant de convertir une entrée couleur (Image / Vidéo / Webcam) en niveau de gris et afficher deux fenêtres pour visualiser l'entrée et la sortie.

2. Commentez votre programme et expliquez les paramètres utilisés et illustrez des imprime-écrans de vos rendus.

3. La fonction OpenCV « *cv2.cvtColor* » permet d'assurer ce type de conversion vers d'autre espace couleur (HSV par exemple en employant le paramètre « *cv2.COLOR_BGR2HSV* ») ou bien vers un espace niveau de gris (en employant le paramètre « *cv2.COLOR_BGR2GRAY* »).

Modifiez le programme précédent pour afficher d'autres fenêtres pour illustrer les résultats de conversion des mêmes entrées.

Comparez les résultats de votre développement et ceux obtenus par la fonction « *cv2.cvtColor* ».

4. Commentez votre programme et expliquez les paramètres utilisés et illustrez des imprime-écrans de vos rendus.

Convolution d'images

5. Appliquez un filtre moyenneur (noyau 5*5) pour lisser les entrées en niveau de gris et un filtre Sobel pour la détection des contours dans ces entrées.

Ecrire un programme permettant de convertir une entrée couleur (Image / Vidéo / Webcam) en niveau de gris et afficher plusieurs fenêtres pour visualiser les traitements apportés.

6. Commentez votre programme et expliquez les paramètres utilisés et illustrez des imprime-écrans de vos rendus.

7. La fonction OpenCV « *cv2.blur* » permet d'appliquer un filtre moyenneur. Modifiez le programme précédent pour appliquer cette fonction sur les mêmes entrées.

Comparez les résultats de votre développement et ceux obtenus par la fonction « *cv2.blur* ». Explorer d'autres filtres et comparer les résultats.

8. Commentez votre programme et expliquez les paramètres utilisés et illustrez des imprime-écrans de vos rendus.