DI5 - Polytech
Ramel JY

TP – Des images aux descripteurs

TP sous Python (Jupyter Notebook ou autres) avec OpenCV principalement.
Ma Python version = 3.6.10 |Anaconda, Inc.| OPENCV Version = 3.4.2

Tous les exercices des TP devront donner lieu à un descriptif/commentaire qui sera intégré dans le compte-rendu des TP à remettre après la fin des TP pour évaluation. Ces rapports seront inclus dans un fichier ZIP contenant les codes Python produits pour chaque exercice (un par binôme). Le fichier zip nommé **nom1_nom2.zip** sera déposé sur Celene dans la zone de dépôt 2020.

Partie 0 : Prise en main de Python + OpenCv
Récupérez les scripts Python imageProcessing.py.txt et videoProcessing.py.txt disponibles sur Celene qui serviront de base au exercices.
Analysez leur contenu et ajoutez les commentaires manquants.

```
# Image_Processing OpenCV
#
import cv2
import os
import sys
import numpy as np
from matplotlib import pyplot as plt

#
print("Python version")
print (sys.version)
print (sys.version_info)
print("OPENCV Version =", cv2.__version__)
```

```
    #
    rep_cour = os.getcwd()
    print(rep_cour)

    #
    img = cv2.imread('data/test_image.jpg',0)

    #
    laplacian = cv2.Laplacian(img,cv2.CV_64F)
    sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)
    sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)
```

```
    #
    plt.subplot(2,2,1),plt.imshow(img,cmap = 'gray')
    plt.title('Original'), plt.xticks([]), plt.yticks([])
    plt.subplot(2,2,2),plt.imshow(laplacian,cmap =
    'gray')
    plt.title('Laplacian'), plt.xticks([]), plt.yticks([])
    plt.subplot(2,2,3),plt.imshow(sobelx,cmap =
    'gray')
    plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
    plt.subplot(2,2,4),plt.imshow(sobely,cmap =
    'gray')
    plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])
    plt.show()
```

```
# Mon script OpenCV : Video_processing
#
import numpy as np
import cv2

#
def imgproc(imgc):
        return imgc

#
cap = cv2.VideoCapture('data/terasse.mp4')

#
while (True):
```

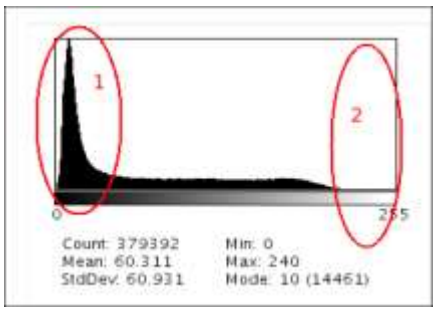
```
    #
    ret, frame = cap.read()

    #
    if ret == True:
        #
        img = frame.copy()
        gray = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)

        #
        gray = frame_processing(gray)
```

```
        #
        cv2.imshow('MavideoAvant', frame)
        cv2.imshow('MavideoApres', gray)

    else:
        print('video ended')
        break

    if cv2.waitKey(1000) & 0xFF == ord('q'):
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

Parcourez les documentations en ligne afin de pouvoir l'exploiter pour résoudre les exercices (vérifier la version dont vous disposez) : https://docs.opencv.org/3.4.2/index.html

Partie 1 : Types d'images et statistiques images

Q1 Ouvrez l'image lisa.png et convertissez-la en niveaux de gris. A l'aide d'openCV, générez et afficher l'histogramme de niveaux de gris, et expliquez la répartition des niveaux de gris dans l'histogramme aux points 1 et 2.
https://docs.opencv.org/3.4.2/d1/db7/tutorial_py_histogram_begins.html



Q2 Ouvrez et convertissez en niveaux de gris, les images paysage, mystere et soleil. Pour chacune d'entre-elles

- Générez l'histogramme de niveaux de gris
- Egalisez l'histogramme
- Générez l'histogramme de niveaux de gris après égalisation

Pour certaines images (au choix), insérez dans votre rapport l'histogramme et l'image avant et après égalisation. Expliquez

Partie 2 : Filtrage, convolution, détection de contours

Q1: Ouvrez lisa.png (joconde) et convertissez-les en niveaux de gris. Appliquez une seule fois des filtres moyenneur de taille/rayon variable:

1. 5X5
2. 9X9
3. 15X15

Quelle influence la taille du filtre a-t-elle cette étape de convolution ?
https://docs.opencv.org/3.4.2/d2/d96/tutorial_py_table_of_contents_imgproc.html

Q2 Que donnerait l'application d'un filtre moyenneur dont la dimension serait égale à celle de l'image ?

Q3: Complétez le code fourni en partie 0 afin d'appliquer différents traitements de filtrage (flou gaussien, masque median, …). Insérez certains résultats dans votre rapport et commentez-les afin de démontrer que vous avez compris le mécanisme de filtrage d'images

Q4: Effectuez les actions suivantes et reportez dans votre rapport les coefficients de filtres utilisés et les images filtrées obtenues :

- Ouvrez l'image zebre.jpg et convoluez la avec un filtre permettant de faire ressortir les traits horizontaux
- Ouvrez l'image suzan.jpg et convoluez la avec un filtre permettant de faire ressortir les traits verticaux

Expliquez les masques utilisés

Q5: Complétez le code fourni en partie 0 afin d'appliquer différents algorithmes de détection de contours (Sobel, Laplacien, Canny …) à la frame courante avant affichage. Insérez certains résultats dans votre rapport et commentez-les afin de démontrer que vous avez compris le mécanisme de filtrage d'images.

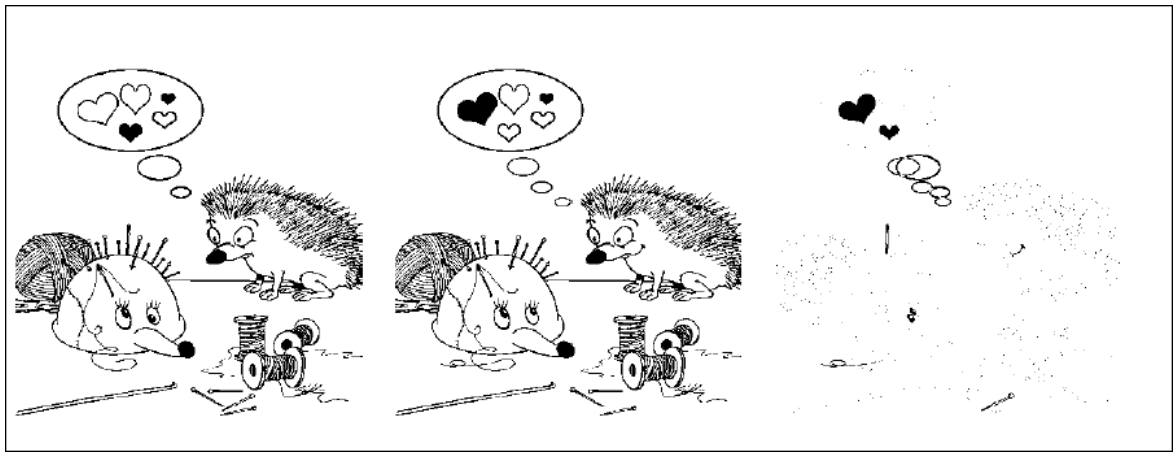
Q6: Selon vous, quels sont les points faibles de toutes ces méthodes de filtrage par convolution ?

Partie 3 : Images binaires et opérations entre images

Q1: Complétez le code fourni afin d'appliquer différents algorithmes de seuillage / binarisation (seuillage fixe ou adaptatif) à différentes images fournies. Insérez certains résultats dans votre rapport et commentez-les afin de démontrer que vous avez compris le mécanisme de binarisation
https://docs.opencv.org/3.4.2/d2/d96/tutorial_py_table_of_contents_imgproc.html

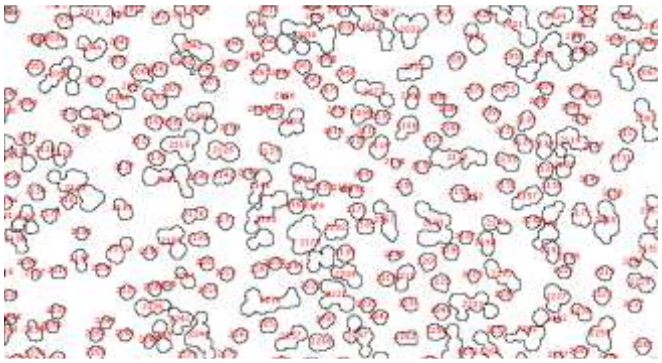
Q2: Ouvrez les images jeu1 et jeu2.
A l'aide d'opérations arithmétiques entre ces deux images, mettez (comme sous l'image ci-dessous) en évidence les 7 différences

existant entre ces deux images.
Proposez un algorithme capable de résoudre ce problème lorsque les images à comparer sont succeptibles d'etre décalées de quelques pixels l'une par rapport à l'autre (CF jeu3).



Q3: Complétez le code Video_processing fourni afin d'appliquer une séquence d'opérations (soustraction entre 2 frames successives, binarisation, érosion/dilatation, …) avant affichage du résultat. Expliquez l'intérêt de cette séquence (opérations entre frames successives) ?

Q4: Des biologistes aimeraient compter le nombre de cellules présentes dans des images. Proposez un algorithme permettant de compter les cellules automatiquement. Vérifiez la présence de 329 cellules dans l'image cellule.png.
https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#ga107a78bf7cd25dec05fb4dfc5c9e765f



Q5 : Malheureusement, une vérification faite manuellement par un biologiste indique que l'algorithme s'est trompé et que le nombre réel de cellules avoisine plutôt les 370.
Une rapide analyse de l'image vous permet de trouver l'origine de cette erreur. En effet, une mauvaise utilisation du système d'acquisition a généré deux types de bruits artificiels :
1. Des petites taches noires sont apparues un peu partout et sont interprétées comme étant des cellules (cf cas 1 de l'image ci-dessous)
2. Des cellules sont collées sur la photo alors qu'elles sont dissociées dans la réalité (cf cas 2 de l'image ci-dessous)

Trouvez la suite d'opérations morphologiques à réaliser afin que ces tâches disparaissent et que le moins de cellules possibles restent collées les unes aux autres. Un protocole "correct" devrait vous ramener vers une détection automatique de 370 cellules.

Partie 4 : Image Tagging

A vous de jouer, durant les séances de TP restantes, en créant un programme associant des mot-clés décrivant le contenu (couleur, contenu, qualité …) des images et montrant vos compétences en traitement d'images. Ce programme sera décrit de manière détaillé dans votre rapport.

Base d'apprentissage disponible sur Celene. Ce programme devra générer des fichiers « .txt » au format similaire à ceux de la base d'apprentissage. Une image = un fichier .txt portant le même nom que l'image ; seul l'extension change ; le fichier txt est créé dans le même dossier que le fichier image.

Bon courage

Partie 5 (optionnelle) : Vidéos, Détection de changement de plan et résumé automatique de vidéo

Q1 : Changement de plan

Exploitez certaines des méthodes vues précédemment pour mettre en place un mécanisme de détection automatique de changement de plan dans les vidéos. Testez et commentez ses performances. Proposez sans le coder un mécanisme que vous jugeriez plus performant.

Q2 : Création d'Images résumés de plan

Exploiter l'ensemble des méthodes vues précédemment pour mettre en place un mécanisme de génération automatique de résumés de plans dans les vidéos. Testez et commentez ses performances. Proposez sans le coder un mécanisme que vous jugeriez plus performant.