

VISION ET PERCEPTION DU NUMÉRIQUE

BOUKHOUH INSAF

DAOUDI WISSAL

TP1 :

Dans ce premier TP, nous avons pu créer les fonctions suivantes :

1- Charger une image en couleur et l'affiche à l'écran dans une fenêtre.

⇒ Ceci à travers l'appel des fonctions « imread » et « imshow ».



2- Afficher les propriétés de l'image en utilisant les fonctions de la classe Mat.

⇒ L'affichage essentiellement du nombre des colonnes et du nombre des lignes.

3- Calculer l'histogramme de couleur de l'image en RGB (de deux manières différentes) :

a. Un vecteur de deux dimensions

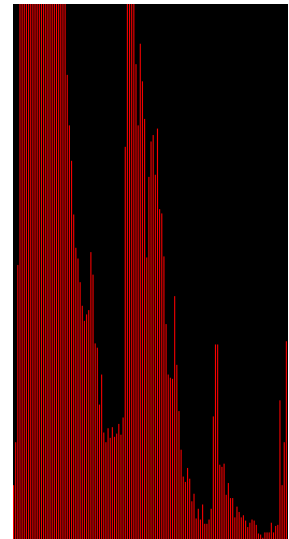
b. CalcHist

c. Calculer la distance (tester les différentes distances) de similarité entre les images (par histogrammes) en utilisant la fonction prédéfinie compareHist.

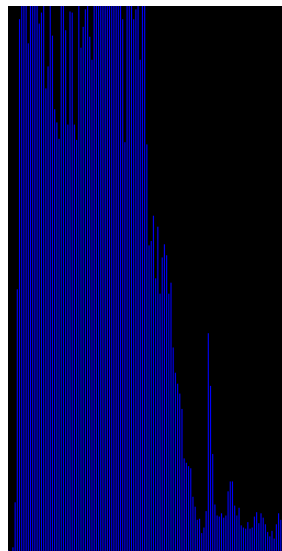
⇒ Nous avons chargé l'image. Ensuite, nous avons divisé l'image en ses R, G and B plans en utilisant la fonction « split ». Nous avons calculé l'histogramme de chaque plan grâce à la fonction calcHist.

⇒ Les résultats obtenus grâce à « CalcHist » sont :

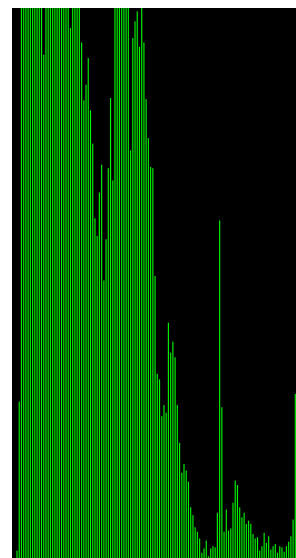
- L'histogramme relatif à la couleur rouge :



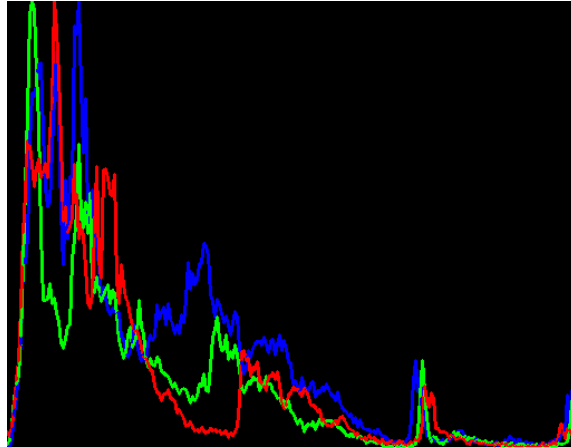
- L'histogramme relatif à la couleur bleue :



- L'histogramme relatif à la couleur verte :



- La superposition des histogrammes donne l'image ci-contre :



4- Convertir une image en image aux niveaux de gris :

- a. En utilisant la formule suivante : $\text{gray} = (R+G+B)/3$



b. En utilisant la fonction prédéfinie cvtColor



5- Sauvegarder l'image convertie dans un fichier.

⇒ A travers l'utilisation de la méthode « imwrite ».

TP2 :

A l'issue du programme qui rajoute un bruit blanc Gaussian dans une image, nous avons obtenu l'image suivante :



Filtre 1

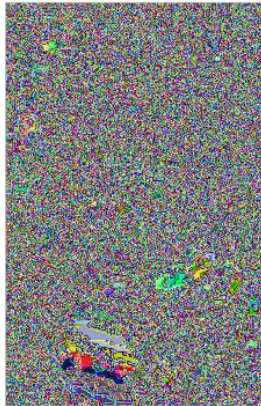
Suite à l'implémentation un filtre « moyen » en considérant un voisinage de connexités 4 et 8, l'image obtenue est :



Filtre 2

Le programme qui va remplacer, dans l'image de sortie, la valeur d'un pixel par la différence de ce pixel multipliée par 8 avec ses 8 voisins, donne l'image :

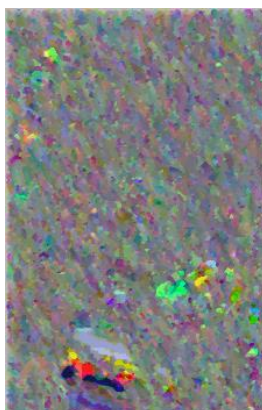
Something is wrong :(



Filtre 3

Le programme qui va remplacer, dans l'image de sortie, la valeur d'un pixel par la valeur médiane d'une série est par définition celle qui sépare l'échantillon en deux parties de population égale, produit l'image :

This one is right but the previous one is wrong so this one became wrong as well :(



TP3 :

Exercice 1

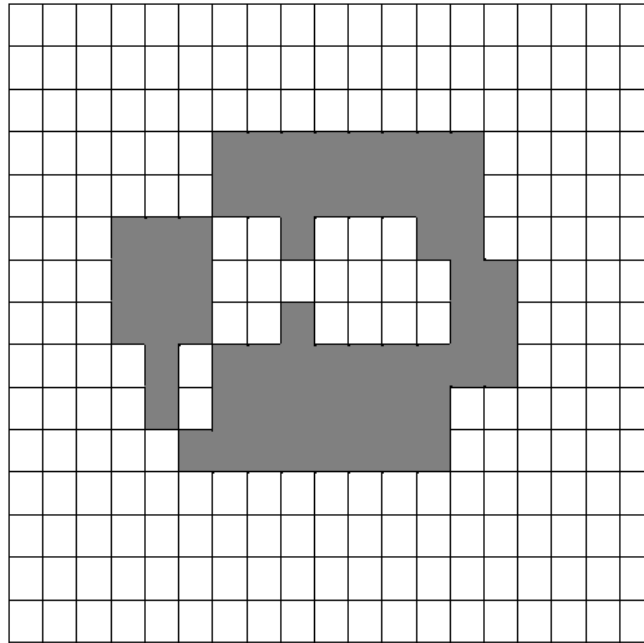


Figure 1

- 1) Le nombre de composantes connexes et de trous pour l'objet de la figure 1 en connexité 4 et en connexité 8 est :

En connexité 4 :

Le nombre des composantes connexes : $C = 2$

Le nombre de trous : $H = 0$

En connexité 8 :

Le nombre des composantes connexes : $C = 1$

Le nombre de trous : $H = 2$

- 2) Soit C le nombre de composantes connexes et H le nombre de trous, le nombre d'Euler est donné par $E = C - H$

Le nombre d'Euler (en connexité 4 et en connexité 8) pour l'objet de la figure1 est :

En connexité 4 :

Nombre d'Euler : $E = 2 - 0 = 2$

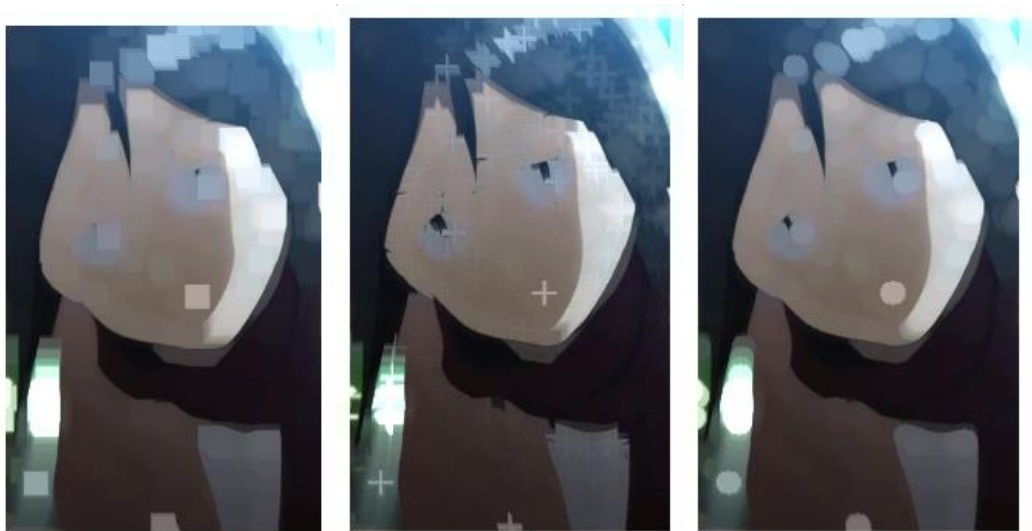
En connexité 8 :

Nombre d'Euler : $E = 1 - 2 = -1$

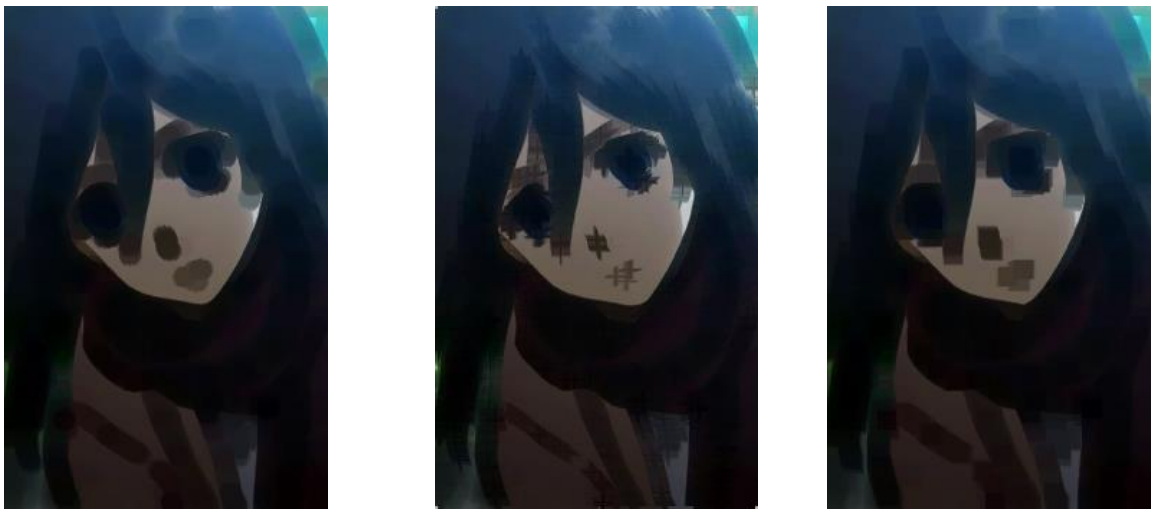
Exercice 2

Les effets de la dilation, de l'érosion, de la fermeture et de l'ouverture :

La dilatation :



L'érosion :



L'ouverture :



La fermeture :



Exercice 3

Les traitements effectués sur les images peuvent être effectués sur les vidéos.

L'objectif est d'écrire un programme qui récupère la vidéo envoyée par la caméra, la traite et l'affiche à l'écran dans une fenêtre.

- ⇒ Il suffit d'ouvrir la camera et grâce à une boucle d'assurer le traitement des images.
Les traitements que nous avons opérés sont : la transformation en gris et l'application du GaussianBlur et Canny.

Exercice 4 (Rajouter un curseur à une fenêtre)

L’objectif est d’écrire un programme permettant de modifier les valeurs du contraste et de la luminance d’une image donnée.

