À rendre par mail avant le 10 janvier :
alexandrebenoit@yahoo.fr

Le but de ce DM est de programmer quelques fonctionnalités du logiciel **Mesurin**. On complétera le fichier python : **dm_1.py** disponible sur le site web. Le fichier devra juste contenir les fonctions demandées. Comme lors du TD, on ne travaillera pas sur l'image directement mais sur le tableau contenant la valeur des pixels de l'image.

On testera ces programmes à partir des deux images couleurs **retine.jpg** et **stomates.jpg** (ces images devront se trouver dans le même dossier que le fichier python).

Exercice 1 :

Dans cette première question, on écrira les fonctions permettant la décomposition d'une image en couleurs ou en niveau de gris.

La décomposition d'une image consiste à ne garder qu'une seule des trois couleurs.

(1) Écrire la fonction **image_rouge_gris** qui partant d'une image en couleur renvoie une image en niveau de gris dont chaque pixel contient l'intensité en rouge de l'image précédente. Par exemple si un pixel de l'image d'origine a comme intensité $(142, 12, 253)$, la valeur du même pixel de l'image de niveau de gris sera 142.

(2) Écrire la fonction **image_rouge** qui partant d'une image en couleur renvoie une image en couleur dont chaque pixel ne contient que l'intensité en rouge de l'image précédente. Par exemple si un pixel de l'image d'origine a comme intensité $(142, 12, 253)$, la valeur du même pixel de l'image en couleur sera $(142, 0, 0)$.

Exercice 2 :

On souhaite maintenant écrire des fonctions de transformation géométrique de l'image.

(1) Écrire la fonction **horizontale** qui effectue la symétrie de l'image par rapport à une droite horizontale (le haut se retrouve en bas)

(2) Écrire la fonction **rotation** qui effectue la rotation de 90 degré de l'image.

Attention pour cette fonction, les dimensions de l'image peuvent changer. En effet si l'image fait 200×50 *pixels, avec la rotation elle fera* 50×200 *pixels. On peut remarquer que la fonction* ***numpy.zeros([l,c,3], dtype=numpy.uint8)*** *crée un tableau de l lignes et c colonnes de* $(0, 0, 0)$ *(donc l'image noire).*

Exercice 3 :

Pour finir on va redimensionner l'image.

Encore une fois, les dimensions des images vont changer. On devra donc créer de nouveaux tableaux de tailles différentes. On utilisera encore la fonction **numpy.zeros([l,c,3], dtype=numpy.uint8)**.

(1) Écrire la fonction **agrandissement** qui double la taille d'une image (quadruple le nombre de pixels). Cette fonction créera une image contenant deux fois plus de ligne et deux fois plus de colonnes. Elle remplacera chaque pixel par un bloc de 2×2 pixels.

(2) Écrire la fonction **reduction** qui divise par deux la taille d'une image. Chaque bloc de 2×2 pixels sera remplacé par un unique pixel dont la valeur est la moyenne des 4 pixels originaux.