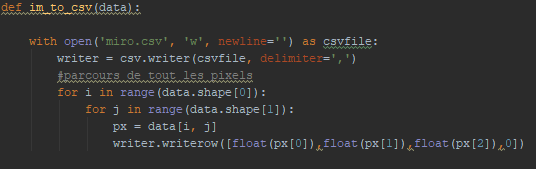
TP5 : RECOLORATION D’IMAGES ET EVALUATION

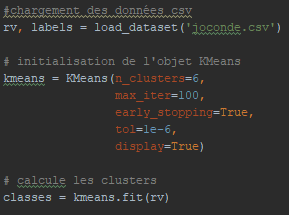
# 1 Recoloration des images avec les kmeans

Pour recolorer les images, nous allons utiliser l’algo des kmeans fait précédemment afin d’assigner à chaque pixel une classe.

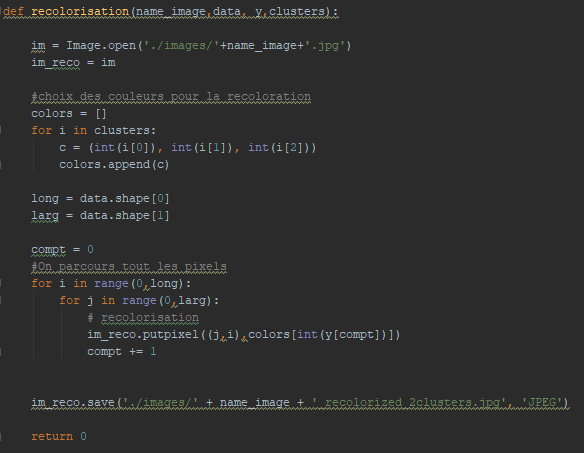
J’ai dans un premier temps créer une fonction pour transformer l’image à recolorer en fichier csv :



Une fois le csv obtenu on peut appliquer l’algo des kmeans et récupérer la liste des classes pour chaque pixels :



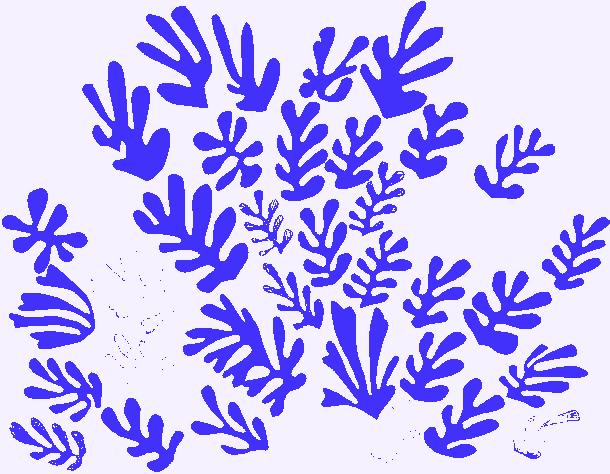
Ensuite une fois la liste des classes on peut lancer la fonction de recoloration :



Dans un premier temps on récupère les couleurs des clusters, qui seront par la suite les couleurs des classes.

On parcourt chaque pixel de l’image et en fonction de la classe on donne une couleur définie dans le tableau colors.

# 2 Résultats









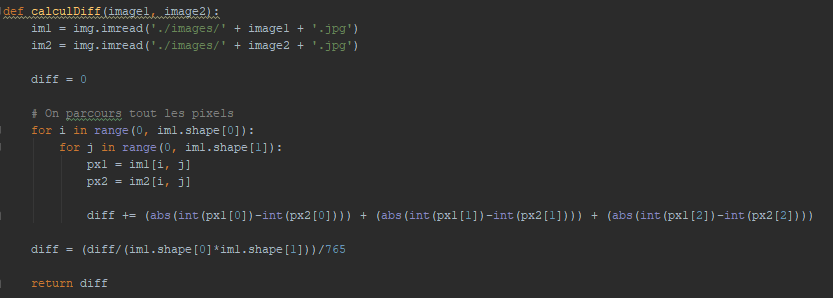




De gauche à droite : image originale, image recolorée 2 clusters, image recolorée 6 clusters

# 3 Evaluation de la qualité de la recoloration

Pour évaluer la qualité de la recoloration j’effectue une comparaison pixel par pixel entre l’image originale et l’image recolorée.



Je fais la moyenne des différences entre les 3 couleurs de chaque pixel. Je divise par 765 pour me ramener à un pourcentage, car 765 (255\*3) est la différence maximale que l’on peut obtenir entre 2 pixels (un blanc 255,255,255 et un noir 0,0,0).

Donc par exemple si on compare le tableau miro original et le recoloré à 2 cluster :

print("diff =", calculDiff("miro", "miro\_recolorized\_2clusters"))

diff = 0.18683218616846933

Avec 6 clusters la différence est logiquement plus faible :

print("diff =", calculDiff("miro", "miro\_recolorized\_6clusters"))

diff = 0.07411095044376322

On a le même schéma avec le tableau de picasso :

print("diff =", calculDiff("picasso", "picasso\_recolorized\_2clusters"))

diff = 0.16489172048158243

print("diff =", calculDiff("picasso", "picasso\_recolorized\_6clusters"))

diff = 0.07797362662055354

# 4 Bonus

On fait le test de recoloré un tableau dans un style bien différents des 3 premiers, la Joconde :



De gauche à droite : image originale, image recolorée 2 clusters, image recolorée 6 clusters

print("diff =", calculDiff("joconde", "joconde\_recolorized\_2clusters"))

diff = 0.08434813957385777

print("diff =", calculDiff("joconde", "joconde\_recolorized\_6clusters"))

diff = 0.04347334110216007

On peut constater que les résultats étonnamment tout aussi satisfaisant. La Joconde possède beaucoup plus de nuances de couleurs que les autres tableaux. Mais toutes ces couleurs ne sont pas si éloignées donc la recoloration est plutôt réussie.