Alix Lanotte-Moro, Kentin Guillemot, Alexandre LAroudie

Problème scientifique Informatique

Notre projet est composé de 4 classes :

Tout d’abord la classe MyImage() qui est la plus importante du projet, elle permet de manipuler des images sans utiliser la classe Bitmap. Elle nous permettra de faire de nombreuse manipulations et transformation sur des images comme des rotations, de changements de couleurs, etc.…

Cette classe est la plus longue de notre projet car chaque partie y fait appel, par exemple les filtres utilisent une méthode pour mettre en place des convolutions nécessitant une Instance MyImage

On a délibérément fait le choix de créer des méthodes d’instance pour notre projet car on estimait que c’était plus pratique à la manipulation dans le main.

Ensuite la classe Pixel(), celle-ci est toute simple et permet de définir ce qu’est un pixel à partir de la valeur de 3 octets pour RGB. Elle est utilisée pour créer une instance de MyImage avec la matrice de pixel qui compose l’image.

Ensuite on a créé 3 autres classes qui sont interconnectées pour mener à bien la compression de Huffmann. Malheureusement on n’a pas réussi à aboutir dans cette compression. Les 3 classes concernés sont la classe Arbre(), Nœud() et Dictionnaire(). Le dictionnaire est défini par une clé qui est un Pixel et un nombre d’occurrence qui est un Int. Dans la classe arbre on va remplir une liste de dictionnaire. On parcourt la matrice pixel par pixel puis on parcourt la liste de dictionnaire si la clé d’un des éléments de la liste est égale au pixel de la matrice alors on augmente le nombre d’occurrence de ce pixel et dans le cas contraire on l’ajoute à la liste en mettant son nombre d’occurrence à 1. Une fois la matrice parcourue on est censé avoir une liste avec tous les pixels de l’image et le nombre de fois où ils apparaissent.

Notre innovation est une manipulation sur les images basées sur des miroirs. C’est-à-dire l’utilisateur sélectionne un des 4 quarts de l’image et celle-ci se retrouve réfléchie dans les 4 autres coins par symétrie axiale et centrale. Pour se faire nous avons donc bien évidement crée 4 méthodes pour chacun des 4 quarts en fonction du choix utilisateur.