

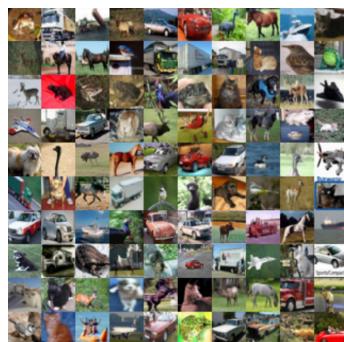
Compte Rendu 2 - Projet Image & Compression

Arthur Chateauneuf, Luna Bossu



Mise en place des classes utilitaires

Afin de charger et de sauvegarder des images, nous utilisons la librairie [STB](#). Après avoir initialisé les classes d'images et de dataset nous pouvons lire et enregistrer des images et des dataset ainsi que les transposer l'une dans l'autre. Voici, par exemple, une image composé des 100 premiers éléments du dataset [CIFAR-10](#) (60k images 32x32) :



Utilisons également les fonctionnalités de la librairie [STB](#) afin de réajuster la taille des images d'entrées. Cela nous permet d'utiliser comme entrée des images dont la taille n'est pas multiple de celle des sous images ainsi que de paramétriser la taille de la mosaïque obtenue.

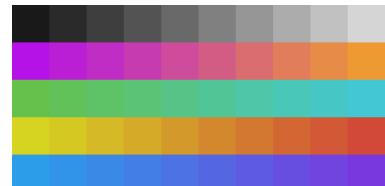
Afin de tester nos algorithmes, nous utiliserons ces images :



[source](#)



[source](#)



Degrade simple fait par
Arthur Chateauneuf

Première approximation de comparaison : Moyenne des Pixels

Nous proposons un premier algorithme permettant d'obtenir un score de ressemblance entre deux sous images grâce à la moyenne des pixels sur les 3 canaux. Nous avons testé cette approche sur les espaces RGB et HSL avec plusieurs résolutions, voici nos résultats :



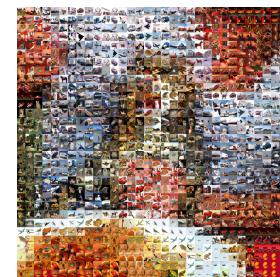
Gradient
CIFAR-10 10x5 RGB



Cat
CIFAR-10 8x8 RGB



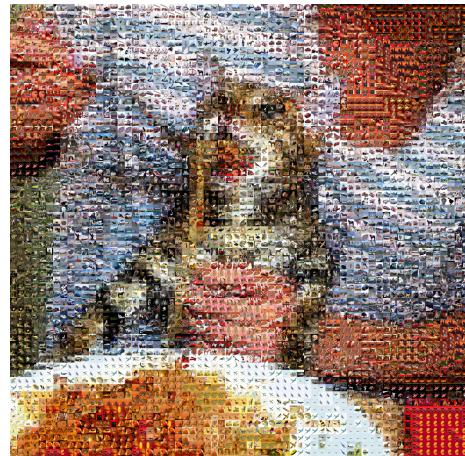
Cat
CIFAR-10 16x16 RGB



Cat
CIFAR-10 32x32 RGB



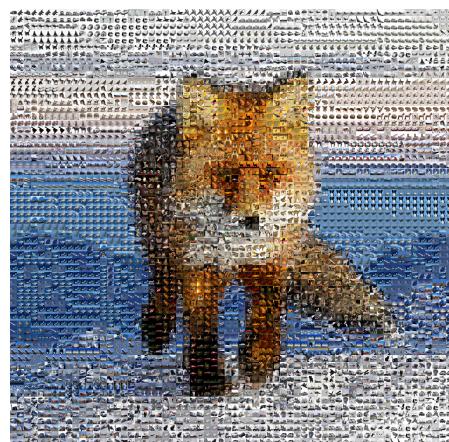
Cat
CIFAR-10 64x64 HSV



Cat
CIFAR-10 64x64 RGB



RedFox
CIFAR-10 64x64 HSV



RedFox
CIFAR-10 64x64 RGB

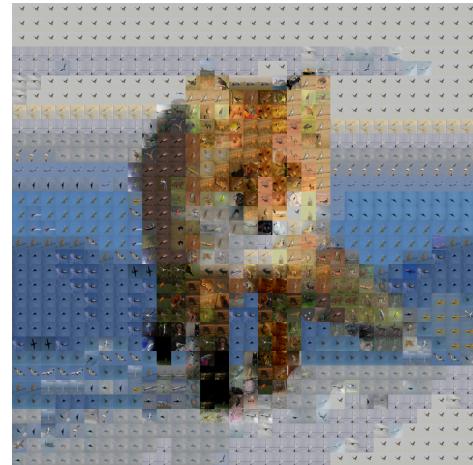
Nous remarquons que cette approche est rapide et permet d'obtenir des résultats convenables lorsque la résolution est élevée (32+). L'espace RGB obtient les meilleurs résultats, car la comparaison dans cet espace permet d'obtenir une moyenne par couleur, permettant de comparer plusieurs teintes. A l'inverse, l'espace HSV ne permet que d'extraire une seule teinte dans la moyenne. Cela entraîne des images de mauvaise teinte d'être choisi comme visible sur le CIFAR-10 64x64 HSV de l'image RedFox.

Deuxième approximation de comparaison : Différence des pixels

Nous proposons maintenant un algorithme utilisant comme score de comparaison la somme totale de la différence entre les pixels. Nous obtenons :



RedFox
CIFAR-10 32x32 Moyenne RGB
(pour comparaison)



RedFox
CIFAR-10 32x32 Différence des pixels



Cat
CIFAR-10 16x16 Différence des pixels



Cat
CIFAR-10 64x64 Différence des pixels

Nous remarquons que cette méthode obtient des résultats bien meilleure que la précédente. La différence met ainsi plus en évidence la ressemblance structurelle au détriment de la ressemblance de la teinte à l'inverse de la première.

Pistes d'améliorations

Les algorithmes que nous avons proposés restent simples dans leur concept. Nous pensons qu'une analyse portée sur la structure fréquentielle des sous image peut être une piste intéressante. De plus, nous pensons que plusieurs scores de comparaisons peuvent être utilisés et combinés avec différents poids afin d'obtenir de meilleurs compromis.

Nous pensons également qu'il sera nécessaire d'obtenir d'autres banques d'images, car celle que nous utilisons actuellement est utilisée dans la reconnaissance d'image. La majorité des éléments sont ainsi un sujet précis sous un fond relativement uniformes, entraînant une pauvre diversité structurelle entre les images.