

## Projet Apprentissage non supervisé

### Info 215 (2019-2020)

L'objectif de ce projet était d'appliquer l'algorithme des mixtures gaussiennes à une image afin de la segmenter en ses différentes couleurs principales. L'image fournie pour le projet est une image de m&m's car les couleurs sont facilement distinguables. Nous avons aussi une autre image pour certaines questions du projet ; il s'agit d'une image tirée du film « Vaiana », produit par Disney. (voir les deux images ci-dessous).



*m&m's*



*Vaiana*

Après avoir utilisé l'algorithme des mixtures gaussiennes sur l'image des m&m's, nous avons choisi de traiter les points n° 1, 3 et 5 ainsi que la question de la compression. L'image de Vaiana est utilisée pour le point 3 et la compression.

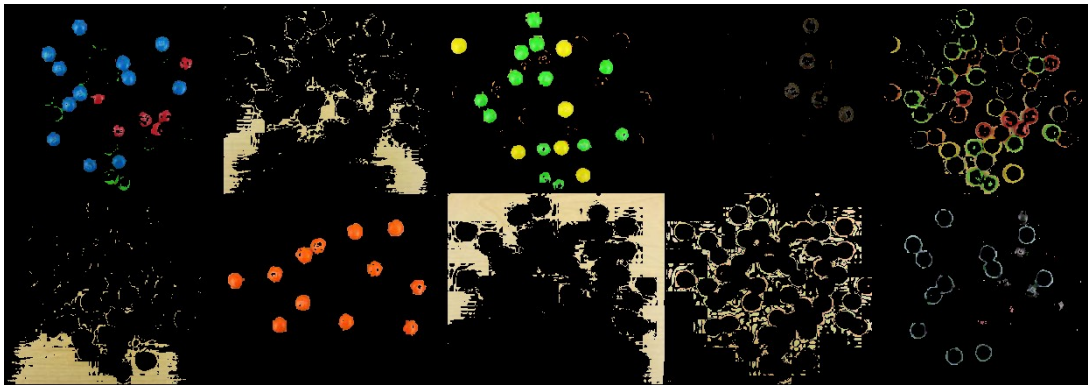
Chaque point abordé possède sa propre classe dans notre code. Cette classe permet de générer les figures qui sont présentes dans ce rapport.

Note : Pour la création du jeu de données, nous avons choisi de représenter les couleurs des images avec leurs composantes HSB (Hue Saturation Brightness). La représentation HSB comportait plusieurs avantages dans notre cas : les trois composantes sont comprises entre 0 et 1 et deux couleurs visuellement proches/éloignées auront une distance plus faible/grande entre leurs composantes HSB qu'entre leurs composantes RGB.

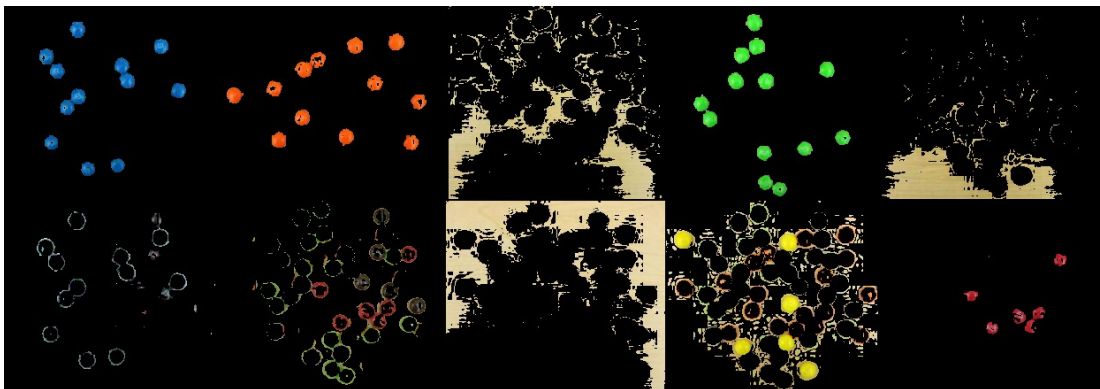
### 1. Point n° 1

Pour ce point nous devons faire tourner notre algorithme avec 10 gaussiennes 10 fois avec des conditions initiales différentes. Nous calculons ensuite leurs scores et nous comparons la meilleure et la pire.

Nous avons obtenu des scores compris entre 3,8 et 3.941.



*Pire score (3.800)*



*Meilleur score (3.941)*

On peut voir que l'algorithme a mieux détecté les couleurs sur l'image 2. Nous apercevons en effet le bleu, le orange, le vert, le jaune avec du bruit et un peu de rouge.

### 2. Point n° 3

Nous avons choisi de segmenter l'image de Vaiana pour son très grand nombre de nuances de couleurs (particulièrement bleu, vert et marron); nous voulions tester notre algorithme avec une image « compliquée » à segmenter. Étant donné la proximité des couleurs dans l'image, nous avons fait une segmentation avec 20 centres. Nous obtenons les résultats suivants :

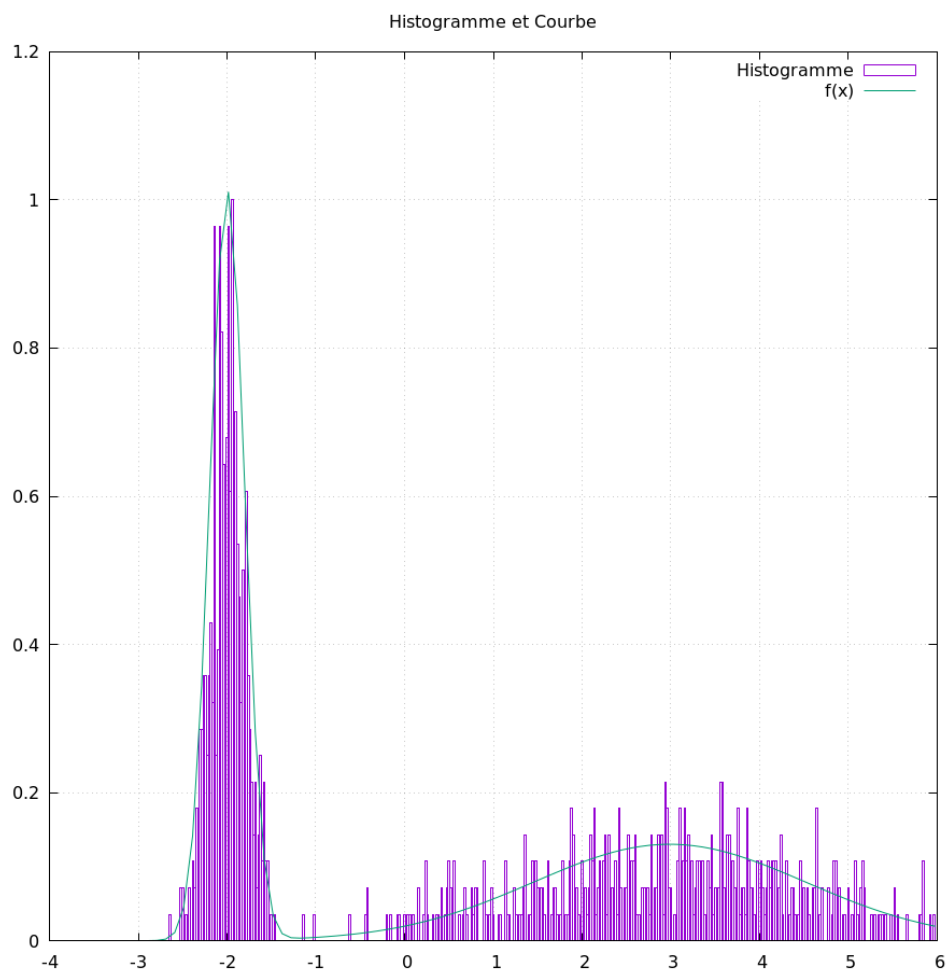
#### PHOTO

On peut voir que les nuances de bleu et de vert ont très bien été séparées. La mer a été séparé en 8 couleurs, la forêt en arrière-plan en 6 couleurs et le sable en 3. On peut objectivement dire que les nuances de couleurs ont bien été détectées.

### 3. Point n° 5

Pour cette partie nous devons générer un jeu de donnée à partir de deux distributions, l'une de 500 points centré en -2 avec une variance de 0.2 et l'autre de 500 points

centré en 3 avec une variance de 1.5. Nous devons ensuite afficher la gaussienne obtenue sur l'histogramme normalisé pour voir si l'algorithme était performant.



On voit clairement que la courbe correspond bien à l'histogramme.

#### 4. Compression

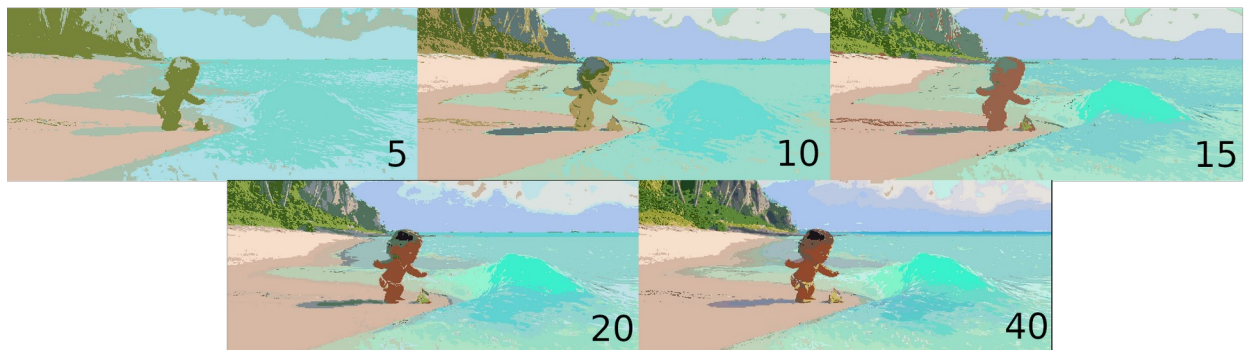
Pour la question de la compression, nous avons fait tourner l'algorithme des mixtures gaussiennes sur l'image de Vaiana avec 5, 10, 15, 20 et 40 centres. Nous obtenons un score moyen de 3.600. Pour créer l'image compressée, une fois que les centres sont calculés, nous parcourons la couleur de chaque pixel de l'image originale et nous écrivons la couleur du centre associé dans l'image compressée.

Pour rappel, voici l'image originale :



*Vaiana*

Et voici les résultats obtenus :



*Compression avec respectivement 5, 10, 15, 20 et 40 centres*

On remarque que même avec 5 centres, l'image est reconnaissable. Plus le nombre de centres augmente, plus les détails de l'image deviennent visibles (particulièrement pour la forêt en arrière-plan). Logiquement, la différence entre la compression à 20 centres et 40 centres est moins flagrante qu'entre 5 et 20 centres. Le gain d'espace de stockage est sans appel, 413.8 Ko pour l'image originale contre 29.5 Ko pour la compression avec 40 centres.