

# PROJET TRAITEMENT D'IMAGES

## RAPPORT DE BERTRAND BLANQUE

---

### Introduction

Ce projet qui a été très long (plus d'une centaine d'heures) a demandé d'écrire de nombreuses structures de données que je vais aborder juste après. Je vais aborder ce thème classe par classe même si la classe `MyImage` possède la majorité des structures.

Ensuite je parlerai de mon projet innovation, j'ai décidé de travailler sur le projet RLE qui me paraissait plus intéressant et de plus fondamental et sur un système de cryptage discret de texte.

### Justifications structures de données

Dans la classe **`MyImage`** la structure donnée la plus importante est là où je stocke les données correspondants à mes valeurs de pixels. J'ai choisi de stocker mes données dans une matrice de pixel. Un pixel est composé d'une valeur rouge, verte et bleue. Toutes ces valeurs sont des entiers, j'aurai pu choisir des bytes car ils sont inférieur à 256 mais j'avais l'habitude de travailler avec des entiers. La matrice de pixel me semblait bien plus facile à utiliser et à comprendre qu'un tableau de bytes extrêmement long. Mon choix a été fait quand j'ai pensé au calculs de matrices que l'on devrait faire avec les matrices de convolutions.

Ensuite j'ai stocké le header en une suite de byte (`byte[]`) avec toutes les informations à l'intérieur. Quand je veux modifier la taille, la hauteur, la largeur je modifie mes trois propriétés `Largeur`, `Hauteur`, `Taille` qui sont des entiers. Ensuite ma fonction `From_Image_To_file()` modifie dans le header aux bonnes positions les attributs hauteur,

---

---

largeur, taille. Ainsi mon image est écrite sans problème et ne génère pas d'erreurs. A part le header et ma matrice de pixel, je n'ai pas d'autres structures de données.

Pour dire un mot sur les constructeurs j'en ai deux, un qui construit une image à partir d'un fichier bmp existant et un autre qui construit une image "from scratch" : juste à partir d'une hauteur et d'une largeur en pixels. Ca me permet de couvrir les deux situations que l'on rencontre dans le projet notamment pendant l'encodage photo et son décodage.

Pour le QR code j'ai utilisé dans celui une Image de la classe MyImage, des données peu importantes et les données à coder qui elles sont une suite très longue de '0' ou de '1' au format string. Je n'aurai pas pu avoir ses données dans d'autres formats.


Pour résumer et de manière général les formats de stockages et de manipulations de données ont toujours été string pour des suites binaires et int pour des entiers ou des bytes. Le format string est très pratique car beaucoup de fonctions de manipulations sont accessibles à partir de celui-ci.

### **Compression RLE (+innovation)**

Explication sur mon format de compression:

Le RLE convertit une suite de 3 pixel noir en 3N, 3 blancs en 3B mais ne permet pas de compresser des images couleurs. J'ai adapté le RLE pour les images couleurs.

Je prends pour chaque pixel d'une matrice le Rouge, le Vert, le Bleu, je les mets en 8 bits et j'additionne dans une même string successivement tous les RGB de la matrice. J'obtiens une string très très longue de de bits. Ensuite j'applique le RLE a cette string. Je compte le nombre de 0 et de 1 de suite dans cette string et je les notes 'B' pour 0 et 'N' pour 1 comme dans le vrai RLE même si ici ils ne désignent ni le noir ni le blanc. J'ai effectué un test sur une image noir et blanc pour commencer:

J'ai effectué la compression RLE sur TEST001 () car je voulais prendre en main l'algorithme avant d'innover dessus. De plus ça m'a donné des idées par la suite. J'ai pour une image noir et blanc du coup fait des conditions pour savoir si chaque pixel était noir ou blanc et ainsi calculer le nombre de pixels blanc qui se suivent et le nombre de noir qui se suivent. Pour TEST001 j'ai eu:

```
504B 216N 264B 216N 264B 216N 264B 216N 264B 216N 264B 216N 264B 216N 264B 216N 480B 216N 264B 216N 264B 216N
264B 216N 264B 216N 264B 216N 264B 216N 264B 216N 264B 216N 503B Attendre 10 sec... la suite arrive
```

201N 18B 2N 18B 2N 18B 2N 18B 2N 18B 2N 18B 2N 18B 2N 18B 2N 18B 20N

2B	3N	1B	2N	2B	3N	1B	2N	1B	4N	1B	2N	2B	2N	1B	3N	1B	3N	1B	3N	2B	2N	1B	3N	2B	2N	1B	3N	1B	3N	1B	3N	2B	2N	1B	3N	
2B	2N	1B	3N	1B	3N	1B	3N	2B	6N	2B	6N	1B	7N	1B	1N	4B	3N	5B	2B	6N	2B	6N	1B	7N	2B	6N	2B	6N	1B	7N	2B	3N	1B	2N		
2B	6N	1B	3N	1B	3N	2B	3N	1B	2N	2B	6N	1B	3N	1B	3N	2B	2N	1B	3N	1B	3N	1B	3N	2B	3N	1B	2N	2B	3N	1B	2N	1B	4N	1B	2N	
2B	6N	2B	3N	1B	2N	1B	4N	1B	1B	4B	2N	2B	6N	1B	7N	2B	6N	2B	3N	1B	2N	1B	4N	1B	2N	2B	2N	1B	3N	1B	2N	1B	3N	1B	2N	
2B	3N	1B	2N	2B	2N	1B	3N	1B	3N	2B	6N	2B	3N	1B	2N	1B	4N	1B	2N	2B	6N	2B	6N	1B	7N	2B	3N	1B	2N	2B	3N	1B	2N	1B	4N	
2B	3N	1B	2N	2B	3N	1B	2N	1B	4N	1B	2N	2B	6N	2B	6N	1B	7N	2B	2N	1B	3N	1B	1N	4B	2N	1B	4N	1B	1N	8N	2B	6N	1B	6N		
1B	3N	1B	4N	1B	1N	6B	3N	2B	1N	3B	2N	1B	3N	3B	2N	2B	1N	2B	6B	2N	1B	3N	1B	1N	1B	1N	1B	1N	1B	1N	1B	4N	2B	2N		
3B	1N	2B	1N	1B	1N	1B	1N	2B	1N	1B	4N	1B	3N	4B	2N	1B	2N	2B	1N	1B	1N	2B	1N	1B	3N	3B	1N	2B	1N	2B	2N	1B	1N	1B		
1B	1N	1B	2N	2B	1N	2B	1N	1B	1N	1B	4N	2B	2N	2B	4N	1B	2N	2B	4N	1B	2N	2B	1N	1B	2N	2B	1N	4B	2N	1B	2B	3N	1B	3N		
1B	7N	3B	6N	1B	1N	3B	3N	1B	6N	1B	2N	1B	6N	4B	1B	7N	2B	1N	1B	1N	1B	2N	2B	2N	1B	1N	1B	1N	1B	4N	2B	1N	2B	1N		
1B	1N	1B	3N	2B	2N	1B	2N	2B	2N	3B	4N	1B	3N	4B	2N	3B	1N	1B	1N	1B	3N	1B	3N	1B	1N	1B	1N	4B	2N	2B	2N	1B	1N	1B		
1B	2N	3B	4N	2B	3N	2B	1N	3B	2N	3B	1N	1B	2N	3B	1N	2B	3N	1B	1N	1B	1N	4N	4B	2N	2B	4N	1B	2N	1B	2N	1B	1N	1B	2N		
1B	2N	1B	2N	1B	1N	1B	1N	1B	4N	2B	2B	1B	2N	2B	4N	1B	2N	2B	6N	1B	3N	1B	3N	2B	2N	2B	1N	1B	2N	2B	6N	1B	4N	2B	3N	
2B	2N	1B	3N	1B	3N	1B	3N	2B	3N	1B	2N	2B	2N	2B	2N	1B	7N	2B	3N	1B	2N	2B	2B	2N	1B	7N	2B	3N	1B	2N	2B	2B	2N	1B	3N	
1B	2N	2B	2N	1B	3N	1B	3N	2B	3N	1B	2N	2B	6N	1B	3N	1B	3N	2B	2N	2B	2N	2B	6N	1B	3N	2B	2N	2B	2N	2B	6N	1B	4N	1B	2N	
2B	2N	2B	2N	2B	3N	1B	3N	5B	2N	2B	1N	4B	2B	3N	1B	2N	1B	3N	1B	3N	2B	1N	1B	4N	2B	3N	1B	2N	1B	3N	1B	2N	1B	4N	2B	3N
1B	2N	1B	3N	1B	3N	2																														

3

---

## Innovation 2 : code discret

Je voulais coder un message secret ou un mots de passe dans une photo à tel point que ce soit indétectable. Nous avons étudié le codage dans un photo avec poid fort mais même si ça ne se détecte pas a l'oeil nu je pense que ce n'était pas assez discret et ça concerne de grosse photo, beaucoup de données, beaucoup de texte à encoder...

Pour des messages très courts (mots de passe, ordre, ect..) j'ai imaginé un codage :

- 1) On prend le texte, on le transforme en bits de données pour chaque lettres
- 2) On prend une image (prenons coco)
- 3) On ajoute a la fin de chaque byte de R,G,B de coco un 0 ou un 1
- 4) On mets une condition d'arrêt
- 5) On récupère le code

Cette méthode est donc assez discrète, vous pouvez la tester dans mon main.

## Auto-critique et problèmes rencontrés

Dans ce projet ma plus grosse erreur a été d'aborder le plus de choses possibles sans avoir un format pivot efficace. Je me suis retrouvé à plusieurs moment à avoir une fonction bien écrite mais un format pivot qui ne fonctionnait pas complètement ou qui n'était pas assez complet. J'ai particulièrement sous estimé l'importance du header qui est essentiel pour que l'ordinateur lise l'image.

Un autre gros problème et je pense que le sujet n'insiste pas assez dessus était que si la taille de l'image n'était pas multiple de 4 l'image allait être distordue, grises... J'ai pris beaucoup de temps à comprendre que je devais essentiellement travailler avec des images multiples de 4 comme les images TEST et me méfier des images comme coco qui ne l'étaient pas.

Pour conclure ce projet était très enrichissant et m'a permit de toucher à tout. Ca demande beaucoup de travail et de temps mais ça en vaut la peine. Il m'a permit de comprendre les bases de la compression d'images et du transport de données d'un endroit à un autre. J'ai pu découvrir le traitement d'images et je vois mieux (m'intéressant à l'IA) comment ce type de choses s'appliquent au deep learning, facial recognition, convolutional neural network...