

# Traitement Numérique de l'Information

## Projet 1

Louise Robert - Alex Broussard - Damien Carrier

## I. Entropie d'un texte

### 0. Introduction

Pour cette partie nous avons décidé de parcourir le texte pour regarder chaque caractère. Nous initialisons un tableau de la manière suivant :

- Si le nouveau caractère n'existe pas dans le tableau on le rajoute et on initialise le nombre de fois ou il est présent à 1.
- Si il est déjà présent on incrémente de 1 son nombre d'occurrence.

De cette manière nous pouvons avoir un tableau dynamique pour lire n'importe quel texte qui nous génère un alphabet aléatoire et personnalisé en fonction du texte entrée.

Ensuite grace à la présence d'un caractère et le nombre total de caractère dans le texte on pourra calculer la probabilité, d'avoir ce caractère ainsi que l'entropie de l'alphabet générée.

### 1. Probabilité d'occurrence de chaque lettre de l'alphabet

Voici ce que retourne le tableau

```
[caractère, Nb occurrence, probabilité]
[' ', 3971, 0.16616453259687003]
['e', 2860, 0.11967528663486485]
['s', 1666, 0.06971294669009959]
['n', 1416, 0.05925182023600301]
['a', 1297, 0.05427232404385304]
['t', 1264, 0.052891455351912295]
['i', 1203, 0.05033894049711273]
['u', 1187, 0.04966942840405055]
['r', 1174, 0.04912544982843753]
['o', 1022, 0.042765084944346804]
```

Il y a 23398 caractères au total.

### 2. Entropie de l'alphabet

Grace à ce tableau, et à notre code nous pouvons calculer l'entropie grace à la formule suivante :

$$H = \sum_i p(x_i) * \log\left(\frac{1}{p(x_i)}\right)$$

L'entropie de notre alphabet est de : 4.3880255630216025

L'unité de notre entropie est : bits d'info/caractère.

### 3. Taille du fichier en octets

L'entropie de notre alphabet est d'environ 4.39, on a donc

$$4.39 * 23398 = 102717.22bits = 12839.625octets = 12.84ko$$

### 4. Algorithme de codage binaire

Grâce à un petit algorithme en python, nous avons fait passé la taille du texte de 25 467 octets à 13 904 octets.

$$\frac{25467}{13904} = 1.83$$

Soit une compression de facteur d'environ 1.83

## II. Entropie d'une image

### 0. Introduction

L'algorithme que nous avons utilisé dans la partie précédente s'appuie sur un parcours de tableau où chaque élément du tableau est un caractère de notre alphabet. Pour calculer l'entropie de l'image nous avons placé les valeurs du fichier Lena\_gray.raw dans un tableau afin de pouvoir lancer notre algorithme de la même manière que précédemment et ainsi pouvoir calculer l'entropie de l'image.

### 1. Estimation de l'entropie et comparaison avec celle du texte

L'entropie de notre image est de : 7,445552485185619

```
[amplitude pixel, Nb occurrence, probabilité]
['120', 2723, 0.010387381029582865]
['11', 2690, 0.01026149650002861]
['119', 2673, 0.010196646893894601]
['12', 2611, 0.009960136565641153]
['10', 2493, 0.00951000400541685]
['24', 2483, 0.009471857178279197]
['125', 2466, 0.009407007572145187]
...

```

Un pixel à une entropie moyenne de 7,4455... hors un pixel est sur 2 dimensions, pour pouvoir comparer avec l'alphabet il nous faut diviser par 2 ce resultat pour être homogène.

résultat : 3.72277624259281

On peut donc en conclure qu'il y a donc plus d'information dans notre alphabet que dans l'image en noir et blanc de Lena.