**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**Национальный исследовательский университет ИТМО**

 МЕГАФАКУЛЬТЕТ ТРАНСЛЯЦИОННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

 ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ

**ЛАБОРАТОРНАЯ РАБОТА №2**

**По дисциплине «Введение в цифровую культуру и программирование»   
Эффективное кодирование**

Выполнил Фадеев Артём Владимирович

(Фамилия Имя Отчество)

Проверила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество)

Санкт-Петербург, 2020 г.

Изображение до/после заданного формата

Изображение выглядит как мужчина

Автоматически созданное описание

🡪

Цифровая последовательность

Цифровая последовательность 64 строки изображения до/после измененияИзображение выглядит как стол

Автоматически созданное описание

Процесс получения

Для получения цифровой последовательность пикселей 64 строки моего изображения я написал программу на языке Python. Для начала необходимо подключить библиотеки:

1. numpy – позволяет работать с массивами
2. matplotlib – визуализация изображений
3. scikit-image – работа с изображениями

Изображение выглядит как текст

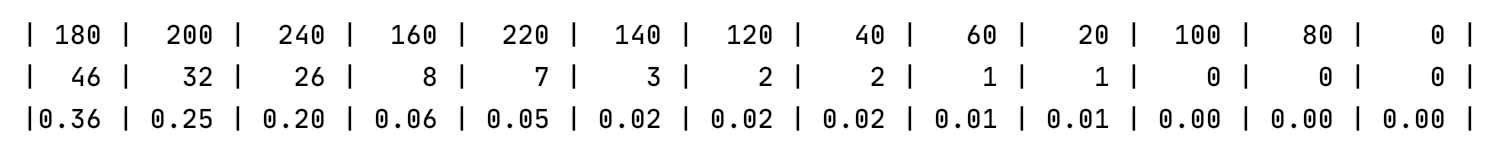
Автоматически созданное описаниеДалее приведу код программы:

Расчет частоты встречаемости и энтропии

Изображение выглядит как текст

Автоматически созданное описаниеАналогично воспользуемся программой:

Получим результат:

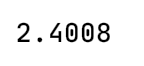


В первой строке находятся наши округлённые пиксели, во второй количество встречи пикселя в строке и его вероятность соответственно. Пиксели со значениями 100, 80, 0 не являются уникальными, поэтому **количество символов алфавита** = **10**. Соответственно **минимальная длина двоичного кода** = log2(10) = **4**.

Изображение выглядит как текст

Автоматически созданное описаниеРассчитаем энтропию:

**Итоговое значение энтропии:**

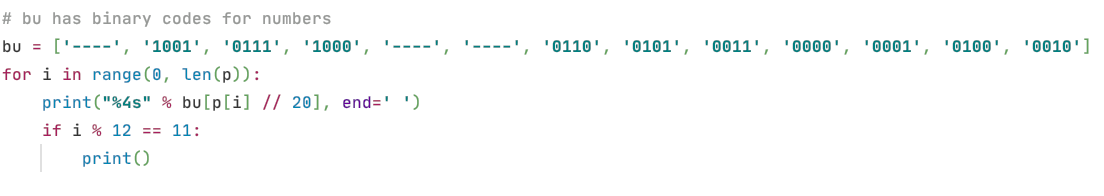


Двоичный равномерный код

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 180 | 200 | 240 | 160 | 220 | 140 | 120 | 40 | 60 | 20 |
| 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 |

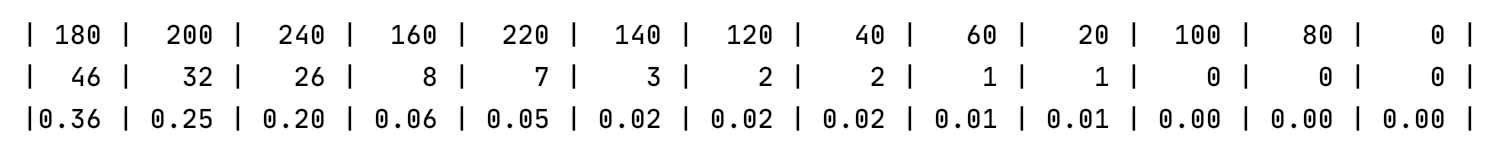
Изображение выглядит как текст

Автоматически созданное описание  
Полученная в результате двоичного равномерного кодирования последовательность:

Код:

Длина двоичного кода = 4 бит

Количество переданной информации = 4 \* 128 = 512 бит

Коды шеннона-фано

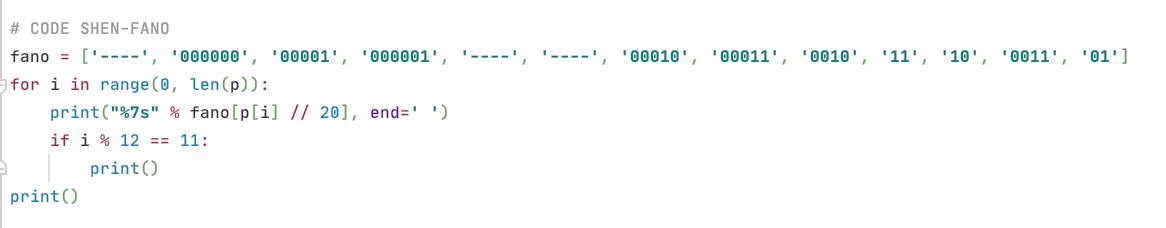
Присвоим вероятностям буквы от A до J по убыванию вероятности встречи элементов в множестве. И будем делить множество на две части, суммарные вероятности которых максимально близки.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | 1 | 11 | | | | |
| B | 10 | | | | |
| C | 0 | 01 | | | | |
| D | 00 | 001 | 0010 | | |
| E | 0011 | | |
| F | 000 | 0001 | 00011 | |
| G | 00010 | |
| H | 0000 | 00001 | |
| I | 00000 | 000001 |
| J | 000000 |

Изображение выглядит как стол

Автоматически созданное описаниеЗакодированная кодом Шеннона-Фано последовательность:

Код:



Средняя длина кодовой комбинации:

Относительная избыточность кода:

Количество переданной информации = 315 бит

Код хаффмана

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J |
| 11 | 10 | 01 | 0011 | 0010 | 00011 | 00010 | 00001 | 000001 | 000000 |

Изображение выглядит как стол

Автоматически созданное описаниеЗакодированная кодом Хаффмана последовательность:

Код:

Средняя длина кодовой комбинации:

Относительная избыточность кода:

Количество переданной информации = 315 бит

Оценка степени сжатия и вывод

При равномерном двоичном кодировании количество переданной информации - 512 бит.

При использовании метода кодирования Шеннона-Фано – 315 бит

При использовании метода кодирования Хаффмана – 315 бит

Степени сжатия соответственно:

Оба метода оказались наиболее эффективными, чем метод равномерного кодирования информации. Из-за того, что алгоритмы используют переменную длину при кодировании: часто встречающиеся символы получают кодовое слово меньшей длины, а редко встречающийся – наибольшей.