Лабораторная работа №1 «Частотный анализатор»

Задание 1.

Реализация программы частотного анализа файлов.

Код программы:

import numpy as np

import matplotlib.pyplot as plt

from collections import Counter

from PIL import Image

import math

# Вывод таблицы встречаемости символов

def print\_frequency\_table(freq):

    print("\nСимвол | Частота")

    print("----------------")

    for char, count in sorted(freq.items()):

        print(f"  {char:3}  | {count}")

# Анализ текста

def analyze\_text(file\_path):

    with open(file\_path, 'rb') as f:

        data = f.read()

    freq = Counter(data)

    return freq

# Анализ фото

def analyze\_bmp(file\_path):

    img = Image.open(file\_path)

    img = img.convert("RGB")

    pixels = np.array(img)

    channels = ['Red', 'Green', 'Blue']

    freq = {ch: Counter() for ch in channels}

    for i, ch in enumerate(channels):

        freq[ch] = Counter(pixels[:, :, i].flatten())

    return freq

# Подсчет энтропии

def calculate\_entropy(freq, total):

    entropy = -sum((count / total) \* math.log2(count / total) for count in freq.values())

    return entropy

# Построение гистограммы

def plot\_histogram(freq, title):

    plt.figure(figsize=(10, 5))

    plt.bar(freq.keys(), freq.values(), color='blue')

    plt.xlabel("Значения (0-255)")

    plt.ylabel("Частота")

    plt.title(title)

    plt.show()

# Главная функция

def main():

    file\_path = input("Введите путь к файлу: ")

    if file\_path.endswith(".txt"):

        freq = analyze\_text(file\_path)

        total\_chars = sum(freq.values())

        entropy = calculate\_entropy(freq, total\_chars)

        print(f"Энтропия текста: {entropy:.4f} бит/символ")

        print\_frequency\_table(freq)

        plot\_histogram(freq, "Частотный анализ текста")

    elif file\_path.endswith(".bmp"):

        freq = analyze\_bmp(file\_path)

        for ch, data in freq.items():

            total\_pixels = sum(data.values())

            entropy = calculate\_entropy(data, total\_pixels)

            print(f"Энтропия {ch} канала: {entropy:.4f} бит/символ")

            print\_frequency\_table(data)

            plot\_histogram(data, f"Частотный анализ {ch} канала")

    else:

        print("Неподдерживаемый формат файла")

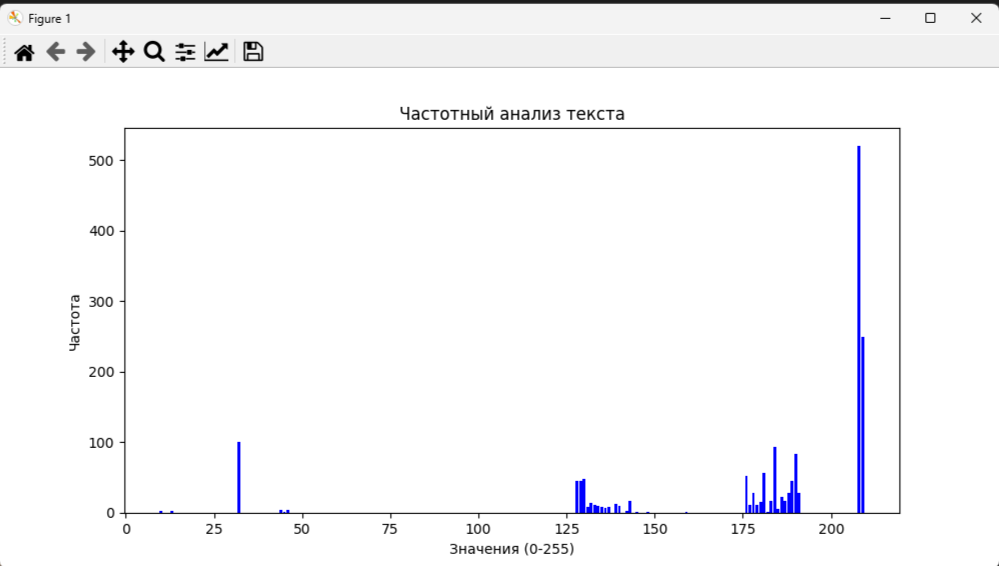
if \_\_name\_\_ == "\_\_main\_\_":

    main()

Задание 2.

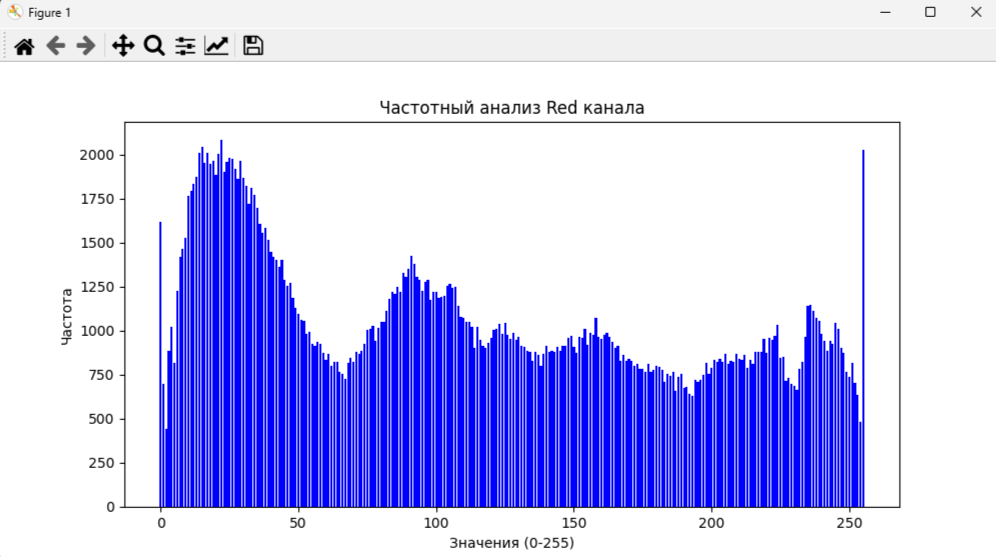
Тестирование данной программы.

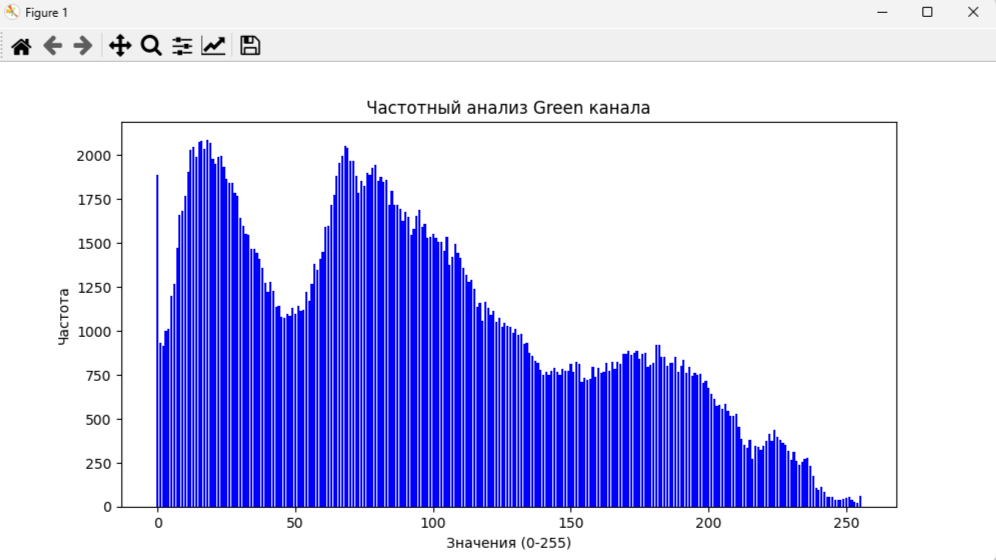
1. Результат тестирования текстового файла:

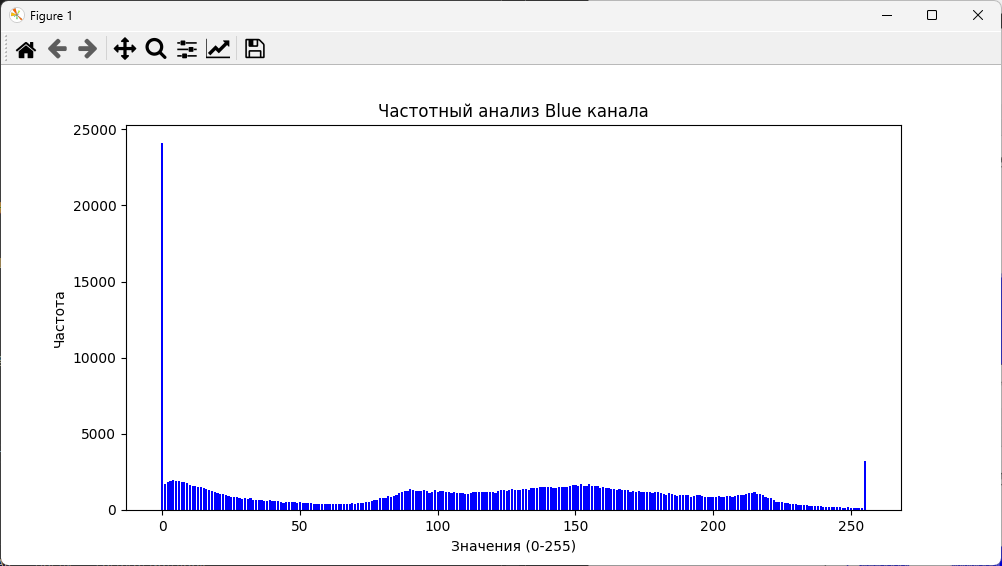


Энтропия текста: 3.8451 бит/символ

2. Результат тестирования изображения:







Энтропия Red канала: 7.9286 бит/символ

Энтропия Green канала: 7.7636 бит/символ

Энтропия Blue канала: 7.5548 бит/символ

Задание 3.

Информационная мера файла рассчитывается с использованием **энтропии Шеннона**. Она показывает, насколько случайно распределены символы в файле и можно ли его сжать без потерь. Чем ниже энтропия, тем больше повторяющихся символов, а значит, файл лучше сжимается.  
Чем выше энтропия, тем случайнее распределение символов, и его сложно сжать без потерь.

Возможности сжатия:

* Тексты на естественном языке (русский, английский) → часто содержат много повторяющихся букв. Значит, хорошо сжимаются алгоритмами как Huffman, LZ77, bzip2.
* Изображения с однородными цветами (например, однотонные .bmp). Значит, хорошо сжимаются.
* Изображения с шумами или сложными деталями. Значит, плохо сжимаются.