МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

09.04.01 «Информатика и вычислительная техника»

Отчет по практической работе №1
«Основы работы с изображениями»

МАШИННОЕ ОБУЧЕНИЕ

Исполнитель:

студент группы 8ВМ22 Ямкин Н.Н. 08.09.2023

Руководитель:

Доцент (ОИТ, ИШИТР) Друки А.А.

Томск – 2023

Содержание

1. Цель работы

Получить навыки работы с изображениями в платформе Google Colab на примере выделения границ на изображении.

2. Задачи

1. Получение навыков работы с Google Colab и веб-оболочкой iPython Notebooks;
2. Получение навыков обработки изображений;
3. Применение оператора Собеля для решения задачи выделения границ на изображении.

3. Ход работы

Создадим новый блокнот в Google Colab и импортируем исходное изображение.

Подключим необходимые для работы с изображениями библиотеки:

```
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
```

Загрузим исходное изображение и отобразим его.

```
img = Image.open(r"/content/sample_data/flower.jpg")
plt.imshow(img)
```

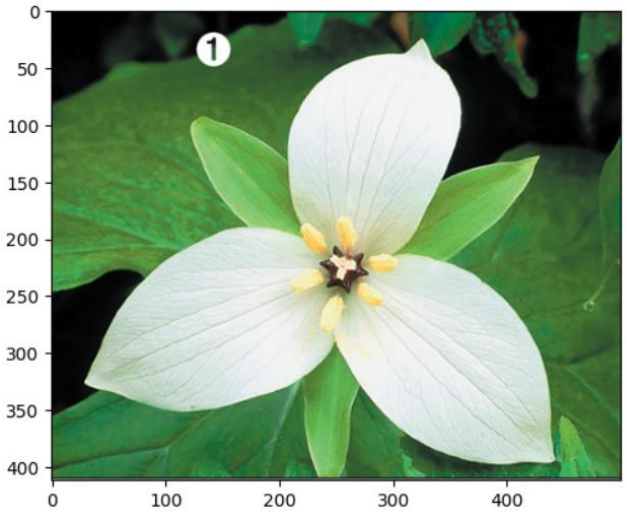

Рисунок 1 – Исходное изображение

Преобразуем исходное изображение в numpy-массив:

```
img = np.array(img)
```

Преобразуем изображение в градации серого через среднее значение каналов.

Этот метод вычисляет среднее значение интенсивности пикселя по всем каналам цветного изображения и устанавливает его в качестве значения оттенка серого. Формула для каждого пикселя: gray = (R + G + B) / 3. Выполним преобразование и отобразим полученное изображение:

```
img_gray = (img[:,:, 0] + img[:,:, 1] + img[:,:, 2]) / 3
plt.imshow(img_gray, cmap = 'gray')
```

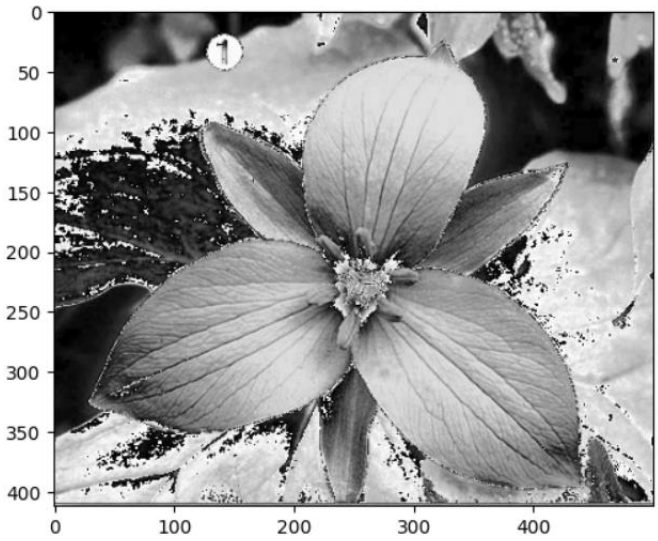

Рисунок 2 – Изображение в градациях серого

Определим функцию, выполняющую операцию свёртки:

```
def convolutinal(input_image, kernel):
  processed_img = np.zeros(input_image.shape)

  # перебираем строки начиная со второй по счету (индекс 1)
  for row in range(1, len(processed_img) - 1):
    # перебираем столбцы начиная со второго по счету (индекс 1)
    for column in range(1, len(processed_img[0]) - 1):

      # выделяем область исходного изображения для вычислений
      procesed_area = input_image[row-1:row+2, column - 1: column +2]

      # свертка
      processed_img[row,column] = np.abs(np.sum(procesed_area * kernel))
```

```
  return processed_img
```

Определим оператор Собеля для нахождения горизонтальных градиентов:

```
sobel_matrix_horizontal = np.array([
                            [1,0,1],
                            [-2,0,2],
                            [-1,0,1]
])
```

Применим операцию свёртки и выведем полученное изображение:

```
sobel_horiz_image = convolutinal(img_gray, sobel_matrix_horizontal)
plt.imshow(sobel_horiz_image, cmap = 'gray')
```

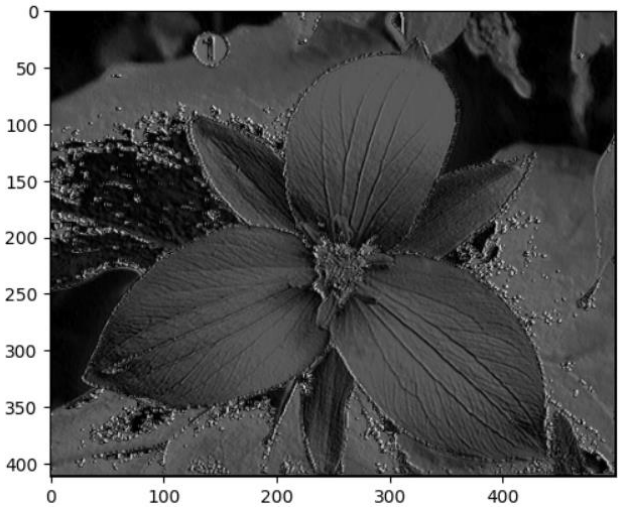

Рисунок 3 – Результат применения свёртки

Определим оператор Собеля для нахождения вертикальных градиентов:

```
sobel_matrix_vertical = np.array([
                            [-1,-2,-1],
                            [0,0,0],
                            [1,2,1]
])
```

Применим операцию свёртки и выведем полученное изображение:

```
sobel_vertical_image = convolutinal(img_gray, sobel_matrix_vertical)
plt.imshow(sobel_vertical_image, cmap = 'gray')
```

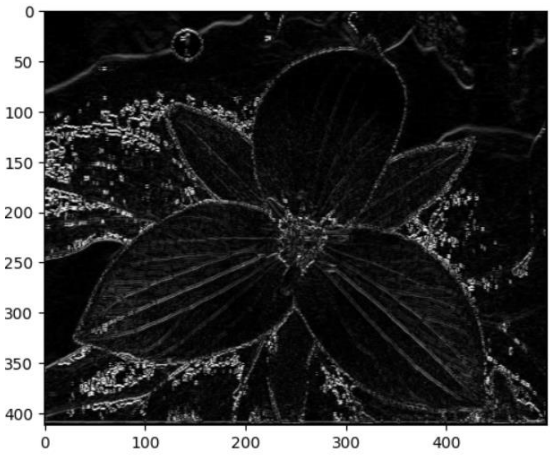

Рисунок 4 – Результат применения свёртки

Сформируем и выведем окончательное изображение:

```
sobel_result_image = np.sqrt(sobel_horiz_image**2 +
sobel_vertical_image**2)
plt.imshow(sobel_result_image, cmap = 'gray')
```

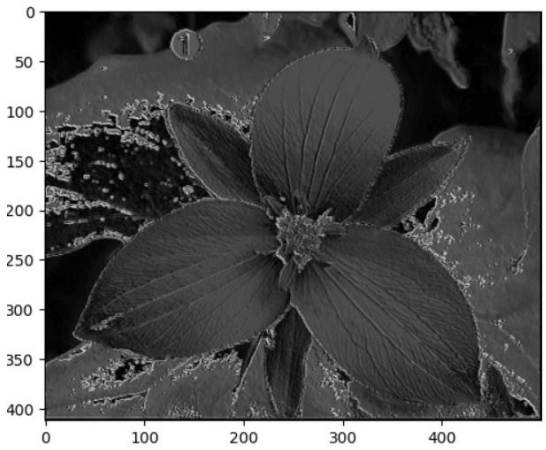

Рисунок 5 – Изображение с выделенными границами

4. Заключение

В результате выполнения лабораторной работы были получены навыки работы с Google Colab и iPython Notebooks, а также основы обработки изображений с помощью данных сервисов.