ЛАБОРАТОРНАЯ РАБОТА №13
АДАПТИВНАЯ БИНАРИЗАЦИЯ

Цель работы: Изучение и применение методов адаптивной бинаризации для обработки изображений с использованием языка программирования Python.

Необходимое программное обеспечение:

1. Python
2. Jupyter Notebook
3. Библиотеки: OpenCV, NumPy, Matplotlib

Шаги выполнения:

1. Введение:

- Ознакомьтесь с принципами адаптивной бинаризации и ее роли в обработке изображений.

2. Загрузка и визуализация изображения:

- Загрузите изображение, на котором будут проводиться эксперименты.
- Визуализируйте изображение в Jupyter Notebook.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Загрузка изображения
image_path = 'path/to/your/image.jpg'
image = cv2.imread(image_path)

# Преобразование BGR в RGB
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Визуализация изображения
plt.imshow(image_rgb)
plt.title('Original Image')
plt.axis('off')
plt.show()
```

3. Подготовка данных для адаптивной бинаризации:

- Преобразуйте изображение в оттенки серого, если оно в цвете.
- Примените гауссово размытие для сглаживания изображения.

```
# Преобразование изображения в оттенки серого
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Применение гауссового размытия
blurred_image = cv2.GaussianBlur(gray_image, (5, 5), 0)
```

4. Адаптивная бинаризация:

- Примените адаптивную бинаризацию с использованием метода **cv2.adaptiveThreshold**.
- Визуализируйте результаты.

 # Адаптивная бинаризация
adaptive_threshold = cv2.adaptiveThreshold(
blurred_image, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)

 # Визуализация результатов
plt.imshow(adaptive_threshold, cmap='gray')
plt.title('Adaptive Thresholding')
plt.axis('off')
plt.show()

5. Эксперименты с параметрами:

- Измените параметры адаптивной бинаризации, такие как размер блока, константа и т.д.
- Визуализируйте, как изменение параметров влияет на результаты.

```
# Эксперименты с параметрами
adaptive_threshold_exp = cv2.adaptiveThreshold(
   blurred_image, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 21, 3)

# Визуализация экспериментов с параметрами
plt.imshow(adaptive_threshold_exp, cmap='gray')
plt.title('Adaptive Thresholding (Experiment)')
plt.axis('off')
plt.show()
```

6. Сравнение с обычной бинаризацией:

- Примените обычную бинаризацию (например, **cv2.threshold**) к изображению.
- Сравните результаты адаптивной и обычной бинаризации.

```
# Обычная бинаризация
_, binary_threshold = cv2.threshold(blurred_image, 128, 255, cv2.THRESH_BINARY)

# Визуализация результатов сравнения
plt.figure(figsize=(12, 4))
```

```
plt.subplot(131)
plt.imshow(adaptive_threshold, cmap='gray')
plt.title('Adaptive Thresholding')
plt.axis('off')

plt.subplot(132)
plt.imshow(adaptive_threshold_exp, cmap='gray')
plt.title('Adaptive Thresholding (Experiment)')
plt.axis('off')

plt.subplot(133)
plt.imshow(binary_threshold, cmap='gray')
plt.title('Binary Thresholding')
plt.axis('off')

plt.show()
```

7. Эксперименты с другими методами бинаризации:

- Попробуйте другие методы бинаризации, такие как **cv2.threshold** с фиксированным порогом, или методы из библиотеки scikit-image.
- Визуализируйте результаты.

```
from skimage import filters

# Бинаризация с использованием cv2.threshold
_, binary_threshold_fixed = cv2.threshold(blurred_image, 128, 255, cv2.THRESH_BINARY)

# Бинаризация с использованием фильтра Otsu
otsu_threshold = filters.threshold_otsu(blurred_image)
binary_threshold_otsu = blurred_image > otsu_threshold

# Визуализация результатов
plt.figure(figsize=(12, 4))

plt.subplot(131)
plt.imshow(binary_threshold_fixed, cmap='gray')
plt.title('Binary Thresholding (Fixed)')
plt.axis('off')

plt.subplot(132)
plt.imshow(binary_threshold_otsu, cmap='gray')
plt.title('Binary Thresholding (Otsu)')
plt.axis('off')

plt.show()
```

8. Анализ результатов:

- Сравните эффективность адаптивной бинаризации с другими методами.

- Проанализируйте, в каких случаях адаптивная бинаризация может быть более предпочтительной.

9. Отчет:

- Составьте отчет, описывающий каждый этап работы, включая примеры кода, визуализацию результатов и выводы.