ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3

за курсом «Алгоритми аналізу та методи обробки зображень»

студента групи ПК-24М-1

Афанасьєва Дениса

кафедра комп’ютерних технологій, ДНУ

2024/2025 н.р.

Постановка задачі

Завантажте зображення. 

Зменшити кількість рівнів інтенсивності на зображенні з 256 до 2 в цілих числах, що мають ступінь 2. Бажана кількість рівнів інтенсивності повинна бути змінною на вході вашої програми.

Виконайте просте просторове усереднення пікселів зображення 3x3. Іншими словами, замініть значення кожного пікселя середнім значенням значень в його околиці 3x3. Якщо піксель знаходиться в позиції (0,0), це означає усереднення значень пікселів в позиціях (-1,1), (0,1), (1,1), (-1,0), (0,0), (1,0), (-1,-1), (0,-1) і (1,-1). Будьте обережні з пікселями на кордонах зображення.   
Повторіть процес для області 10x10 і знову для області 20x20.

Поверніть зображення на 45 і 90 градусів.

Для кожного 3×3 блоку зображення (без перекриття) замініть всі відповідні 9 пікселів на їхнє середнє значення. Ця операція імітує зменшення просторової роздільної здатності зображення. Повторіть цю операцію для блоків 5×5 та 7×7.

Опис розв’язку

Використовував мову програмування Python, з використанням бібліотек OpenCV, NumPy, Matplotli для написання програми. Програмам працює з зображеннями у відтінка сірого, та виконує усі завдання, які були подані в постановці задачі.

При застосуванні програми, просить користувача увести шлях до зображення та бажану кількість рівнів інтенсивності

У результаті застосування програми до фото, отримав такі результати:



Зображення зі зменшеною кількістю рівнів інтенсивності



Усереднення пікселів 3х3



Усереднення пікселів 10х10



Усереднення пікселів 20х20



Поворот зображення на 45 градусів



Поворот зображення на 90 градусів



Зменшення роздільності 3х3



Зменшення роздільності 5х5



Зменшення роздільності 7х7

Код програми

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
def reduce\_intensity\_levels(img, levels):  
 step = 256 // levels  
 return (img // step) \* step  
  
def average\_filter(img, ksize):  
 return cv2.blur(img, (ksize, ksize))  
  
def rotate\_image(img, angle):  
 h, w = img.shape[:2]  
 center = (w // 2, h // 2)  
 matrix = cv2.getRotationMatrix2D(center, angle, 1.0)  
 rotated = cv2.warpAffine(img, matrix, (w, h), flags=cv2.INTER\_LINEAR, borderMode=cv2.BORDER\_REFLECT)  
 return rotated  
  
def block\_average(img, block\_size):  
 h, w = img.shape[:2]  
 img\_out = img.copy()  
 for y in range(0, h - block\_size + 1, block\_size):  
 for x in range(0, w - block\_size + 1, block\_size):  
 block = img[y:y+block\_size, x:x+block\_size]  
 mean = np.mean(block, axis=(0,1), dtype=int)  
 img\_out[y:y+block\_size, x:x+block\_size] = mean  
 return img\_out  
  
def show(title, img):  
 cv2.imshow(title, img)  
 cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
def main():  
 path = input("Введіть шлях до зображення: ")  
 levels = int(input("Введіть бажану кількість рівнів інтенсивності (2, 4, 8, 16...): "))  
  
 img = cv2.imread(path)  
 if img is None:  
 print("Помилка: не вдалося завантажити зображення.")  
 return  
  
 gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
  
 # 1. Зменшення кількості рівнів інтенсивності  
 reduced = reduce\_intensity\_levels(gray, levels)  
 show("Зменшена кількість рівнів інтенсивності", reduced)  
  
 # 2. Просте згладжування 3x3, 10x10, 20x20  
 show("Усереднення 3x3", average\_filter(gray, 3))  
 show("Усереднення 10x10", average\_filter(gray, 10))  
 show("Усереднення 20x20", average\_filter(gray, 20))  
  
 # 3. Повороти  
 show("Поворот на 45 градусів", rotate\_image(gray, 45))  
 show("Поворот на 90 градусів", rotate\_image(gray, 90))  
  
 # 4. Просторове зменшення роздільності  
 show("Зменшення роздільності 3x3", block\_average(gray, 3))  
 show("Зменшення роздільності 5x5", block\_average(gray, 5))  
 show("Зменшення роздільності 7x7", block\_average(gray, 7))  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Висновок

У ході виконання лабораторної роботи навчився працювати з бібліотеками opencv, numpy, matplotlib для мови програмування Python. Зміг реалізувати алгоритм для зменшення кількості рівнів інтенсивності зображення, алгоритм для повороту зображення, та алгоритм для зменшення роздільності зображення.