

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Звіт до лабораторної №1
з дисципліни
“Обробка зображень методами штучного інтелекту”

Виконав:
ст. гр. КН-410
Жигайло Ярослав

Викладач:
Пелешко Д. Д.

Лабораторна робота №1

Попередня обробка зображень

Мета роботи: вивчити просторову фільтрацію зображень, методи мінімізації шуму, морфології, виділення країв і границь та елементи бібліотеки OpenCV для розв'язання цих завдань.

Завдання: вибрати з інтернету два зображення з різною деталізацією об'єктів та два зображення з різним контрастом. Без використання жодних бібліотек для обробки зображень (наприклад OpenCV), виконати гістограмне збільшення контрастності. Провести порівняльний аналіз.

З лекції №1 отримаємо формулу гістограмного збільшення контрастності:

$$I_{(x,y)}(r, g, b) = a \cdot (I_{(x,y)}(r, g, b) - s) + t, \text{ where } 0 < a < 1, \\ s \in \{1, 2, 3, \dots, 255\} \\ t \in \{1, 2, 3\}$$

Рис. 1. Формула гістограмного збільшення контрастності

Створимо програму для обрахунку даної формули, а також для виведення оригінального та трансформованого зображення:

```
from matplotlib import image as img
import matplotlib.pyplot as plt
from copy import deepcopy

def increase_contrast(image, a, s, t):
    res = deepcopy(image)
    width, height, depth = res.shape
    for i in range(width):
        for j in range(height):
            red, green, blue = res[i][j]
            new_red = min(255, max(0, a * (red - s) + t))
            new_green = min(255, max(0, a * (green - s) + t))
            new_blue = min(255, max(0, a * (blue - s) + t))
            res[i][j] = [new_red, new_green, new_blue]
    return res

def plot_image(original, transformed, a, s, t):
    fig = plt.figure(figsize=(10, 7))
```

```

fig.add_subplot(1, 2, 1)
plt.imshow(original)
plt.title("Original Image")

fig.add_subplot(1, 2, 2)
plt.imshow(transformed)
plt.title("Transformed Image a = "+str(a)+' s = '+str(s)+' t = '+str(t))

plt.show()

if __name__ == '__main__':

    img_blurred = img.imread('img/img_blured.jpg')
    img_detailed = img.imread('img/img_detailed.jpg')
    img_high_contrast = img.imread('img/img_high_contrast.jpg')
    img_low_contrast = img.imread('img/img_low_contrast.jpg')

    images_n_params = [
        {'image': img_blurred, 'a': 0.8, 's': 20, 't': 3},
        {'image': img_detailed, 'a': 0.8, 's': 20, 't': 3},
        {'image': img_high_contrast, 'a': 0.8, 's': 20, 't': 3},
        {'image': img_low_contrast, 'a': 0.8, 's': 20, 't': 3}
    ]

    for entity in images_n_params:
        trans = increase_contrast(image=entity['image'], a=entity['a'], s=entity['s'], t=entity['t'])
        plot_image(original=entity['image'], transformed=trans, a=entity['a'], s=entity['s'], t=entity['t'])

```

Оберемо зображення в мережі інтернет, та виконаємо програму (рис. 2-5).

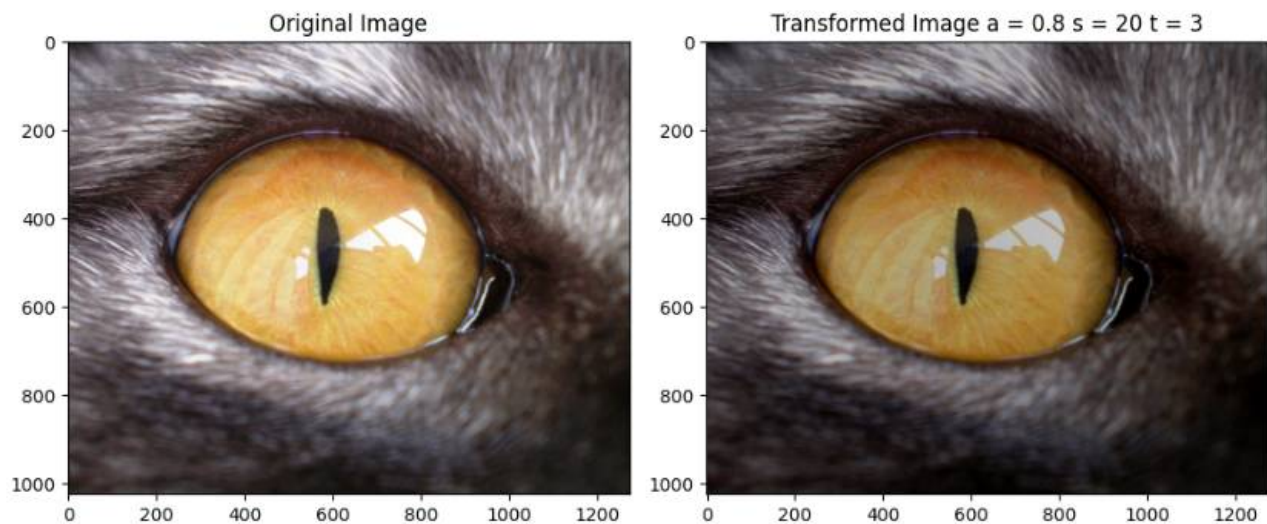


Рис. 2. Результат обробки зображення з низькою деталізацією

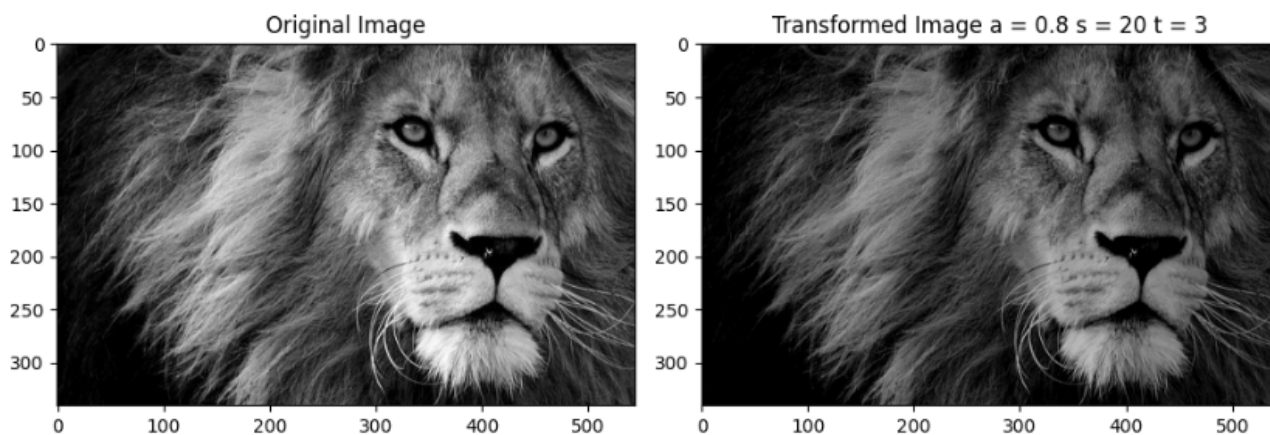


Рис. 3. Результат обробки зображення з високою деталізацією

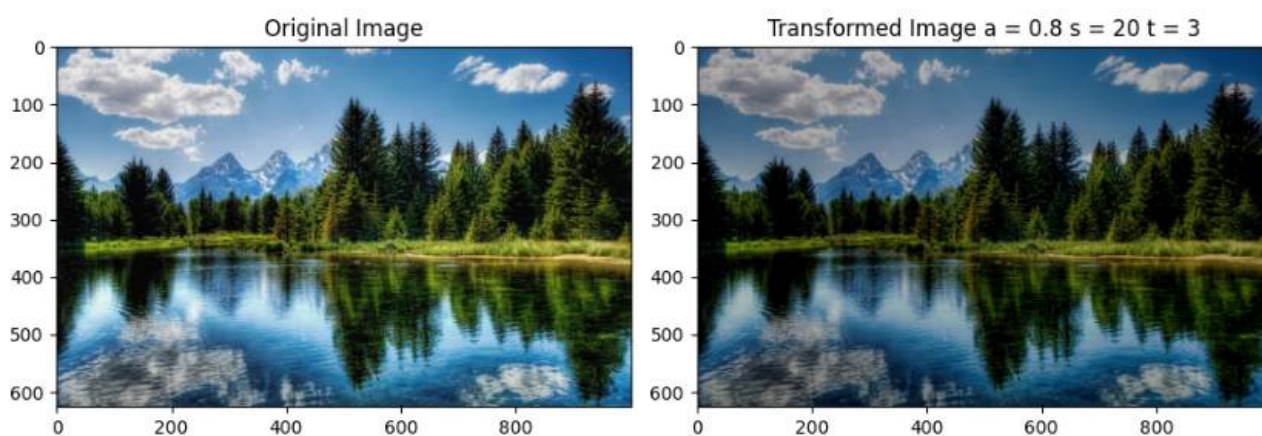


Рис. 4. Результат обробки зображення з високою контрастністю

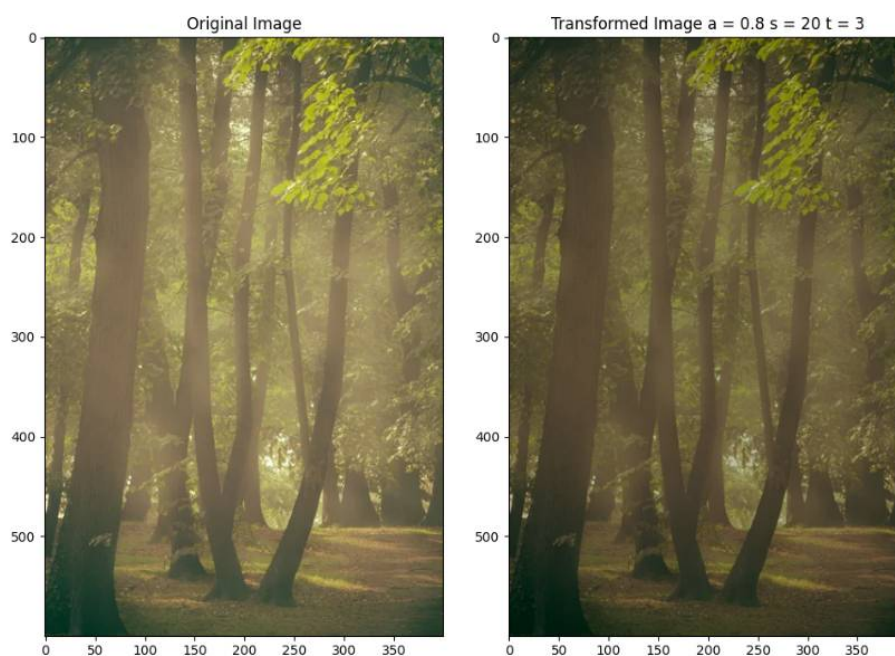


Рис. 5. Результат обробки зображення з низькою контрастністю

Як бачимо з результатів обробки, найбільш помітний ефект є зі зображенням, що мало низьку контрастність. Усі зображення отримали ефект затемнення, проте разом з тим зросла і контрастність.

Висновок: у ході виконання даної лабораторної роботи, я ознайомився з просторовою фільтрацією зображень, методами мінімізації шуму, морфології, виділення країв і границь, а також створив прикладну програму для гістограмного збільшення контрастності зображень.

.