

МІНІСТЕРСТВО ОСВІТИ І НАУКИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА  
ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



Лабораторна робота №1  
з курсу “Обробка зображень методами штучного інтелекту”

Виконала:  
студентка групи КН-410  
Досяк Ірина

Викладач:  
Пелешко Д. Д.

**Тема:** Попередня обробка зображень.

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

### Теоретичні відомості

Комп'ютерний зір - технологія створення машин, які можуть проводити виявлення, стеження та визначення об'єктів.

Попередня обробка: перед тим, як методи комп'ютерного зору можуть бути застосовані, необхідно обробити дані, щоб вони задовольняли деяким вимогам залежно від методу, що використовується.

Приклади:

- видалення шумів задля того, щоби видалити спотворення, які вносяться давачем;
- покращення контрастності для того, щоб потрібна інформація могла бути виявлена;
- масштабування для кращого розрізнення структур на зображенні.

Для того, щоб отримати загальне уявлення про правильність експозиції, контраст і колірне насичення зображення, оцінити необхідну корекцію (зміна експозиції, колірного балансу, освітлення, використовують гістограму.

Гістограма - це графік статистичного розподілу елементів цифрового зображення з різною яскравістю, в якому по горизонтальній осі представлена яскравість, а по вертикалі - відносна кількість пікселів з конкретним значенням яскравості.

Корекція зображень засобами гістограмних операцій:

1. Зсув ліворуч

$$I_{(x,y)}(r,g,b) = \begin{cases} I_{(x,y)}(r,g,b) + k, & \text{if } (I_{(x,y)}(r,g,b) + k) < 255, \\ 255, & \text{if } (I_{(x,y)}(r,g,b) + k) > 255 \end{cases}$$

2. Збільшення та Зменшення контрастності - у випадку зменшення контрастності кут нахилу основної частини зменшується, у випадку збільшення контрастності – збільшується. Оскільки, як правило,  $s > t$ , то в першому випадку динамічний діапазон зображення зменшується, а в другому – збільшується.

3. Зменшення гамми (гамма-корекція) - полягає у зміні кривизни центральної прямої, відносно якої розміщується CDF. Якщо викривлення вниз – зменшення гамми, якщо вгору – збільшення.

4. Збільшення гамми - своєрідна одночасна зміна і яскравості, і контрастності.

## Виконання

### Варіант 6

Вибрати з інтернету два зображення з різною деталізацією об'єктів та два зображення з різним контрастом. Без використання жодних бібліотек для обробки зображень (наприклад Open CV), виконати гістограмний зсув ліворуч. Провести порівняльний аналіз.

Гістограмний зсув ліворуч виконується за формулою:

$$I_{(x,y)}(r, g, b) = \begin{cases} I_{(x,y)}(r, g, b) + k, & \text{if } (I_{(x,y)}(r, g, b) + k) < 255, \\ 255, & \text{if } (I_{(x,y)}(r, g, b) + k) > 255 \end{cases}$$

## Результати

Original image



Illuminated image (k=30)



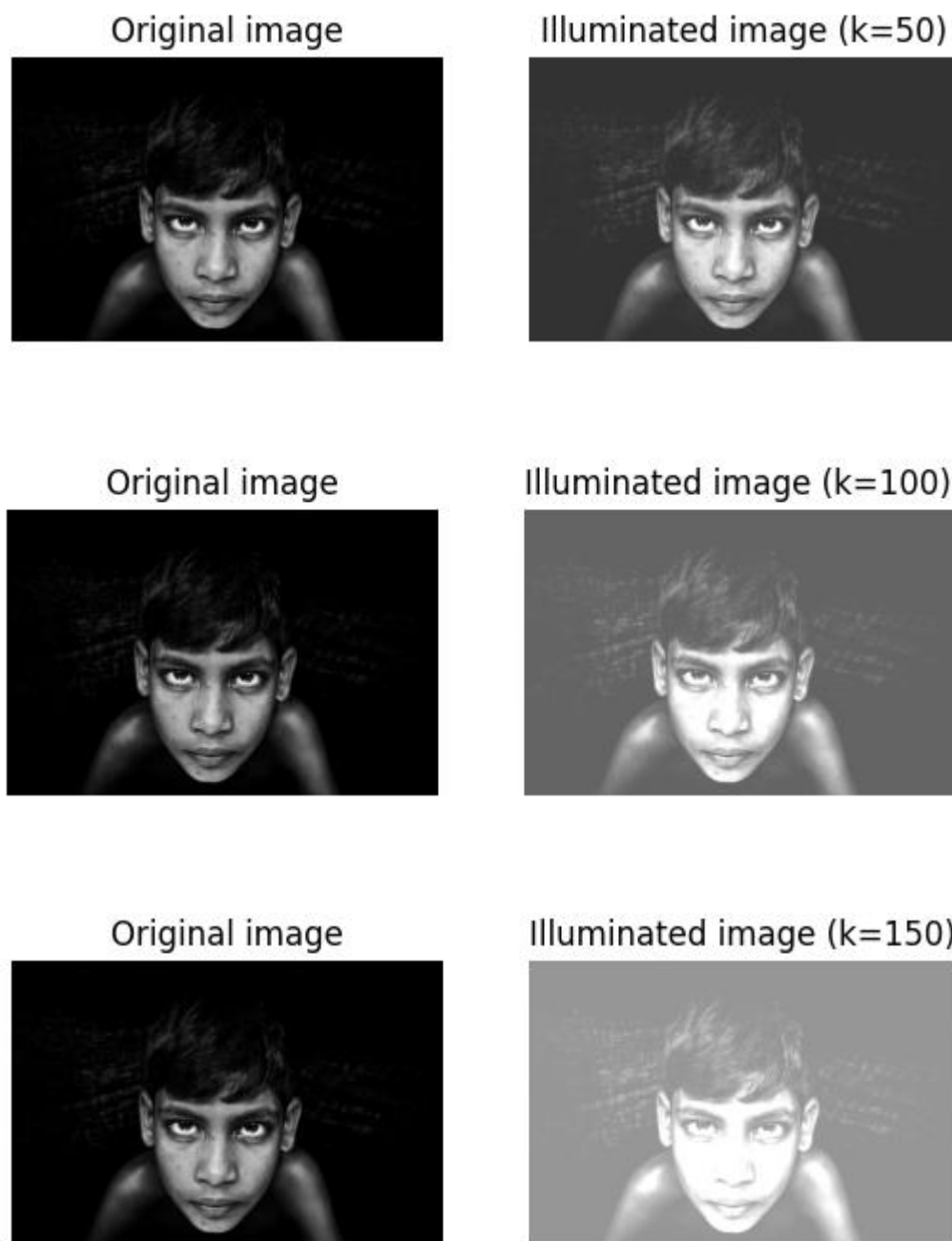
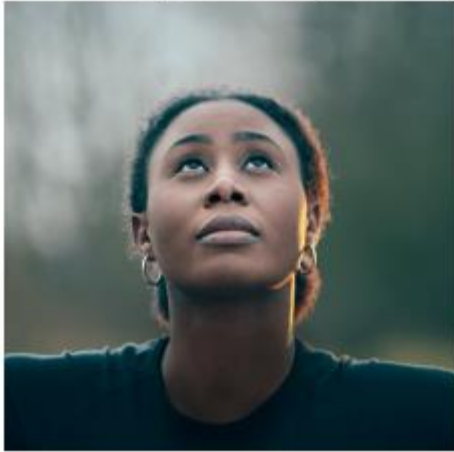
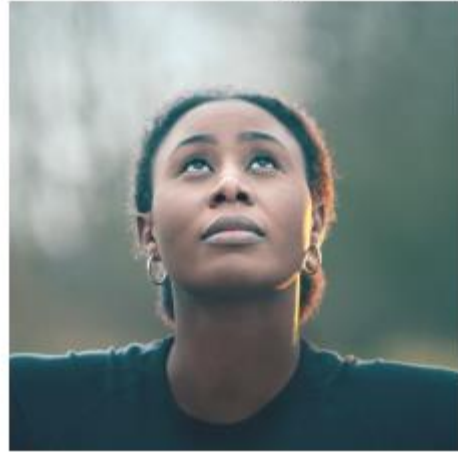


Рис 1. Результат зсуву гістограми при  $k = 30, 50, 100, 150$  пунктів інтенсивності для зображення з високим рівнем контрастності

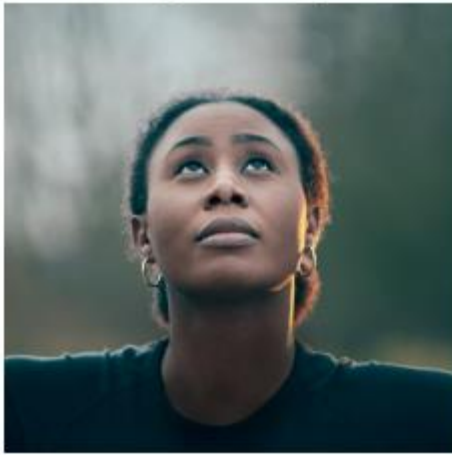
Original image



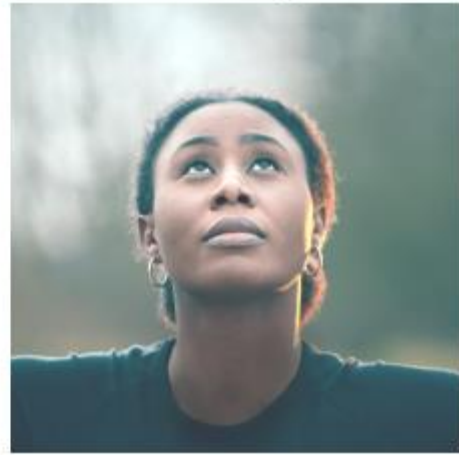
Illuminated image (k=30)



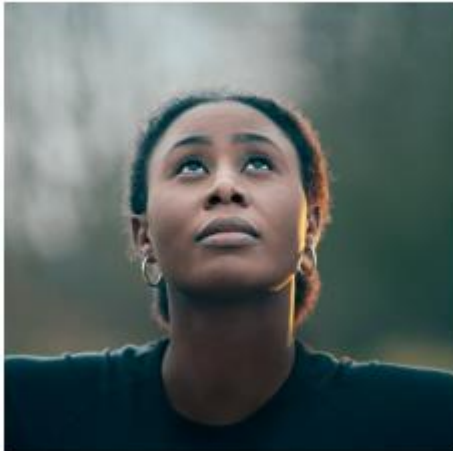
Original image



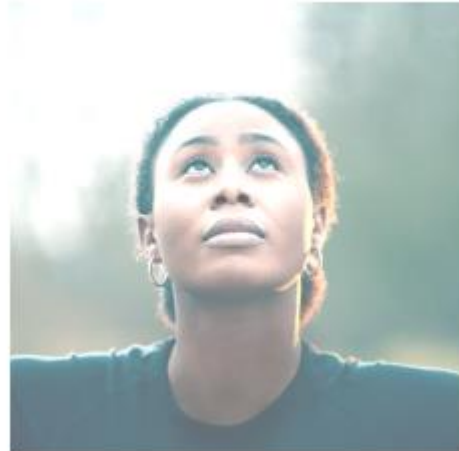
Illuminated image (k=50)



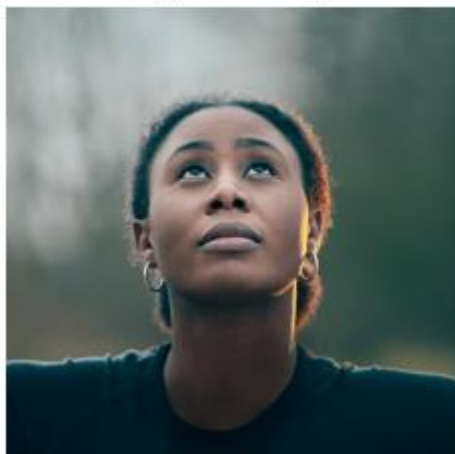
Original image



Illuminated image (k=100)



Original image



Illuminated image (k=150)

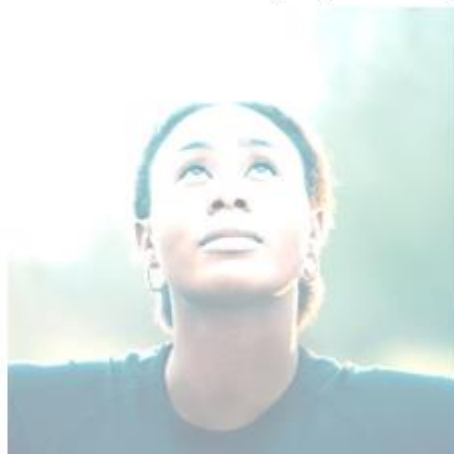


Рис 2. Результат зсуву гістограми при  $k = 30, 50, 100, 150$  пунктів інтенсивності для зображення з низьким рівнем контрастності

Original image



Illuminated image (k=30)





Original image



Illuminated image (k=50)



Original image



Illuminated image (k=100)



Original image



Illuminated image (k=150)



Рис 3. Результат зсуву гістограми при  $k = 30, 50, 100, 150$  пунктів інтенсивності для зображення з високим рівнем деталізації

Original image



Illuminated image (k=30)





Original image



Illuminated image (k=50)



Original image



Illuminated image (k=100)



Original image



Illuminated image (k=150)



Рис 4. Результат зсуву гістограми при  $k = 30, 50, 100, 150$  пунктів інтенсивності для зображення з низьким рівнем деталізації

## Висновок

RGB (червоний, зелений, синій) — адитивна колірна модель, що описує спосіб синтезу кольору, за якою червоне, зелене та синє світло накладаються разом, змішуючись у різноманітні кольори.

Зображення складається з трьох зображень (по одному для кожного каналу), де кожне зображення може зберігати дискретні пікселі зі звичайною інтенсивністю яскравості від 0 до 255.

Чим вища інтенсивність, тим світлішим є зображення. Саме тому, операція зсуву гістограми ліворуч - освітлення зображення. Оскільки згідно з формулою, ми збільшуємо інтенсивність пікселя на  $k$  одиниць.

З отриманих результатів, видно, що зображення з високим вмістом чорних кольорів освітлюються помітніше. Щодо зображень з високим та низьким рівнем деталізації, деталізація залишається такою, як і була. Проте, якщо застосувати зсув ліворуч до яскравого зображення, фільтр може відігравати роль розмиття.

## Додаток

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

from matplotlib import image as img

def validate_input_number(input_number, lower_condition, upper_condition,
message):
    while input_number > upper_condition or input_number < lower_condition:
        input_number = int(input(message))
    return input_number

class ImagePreprocessing:
    """ A class used to make image preprocessing """

    def __init__(self, image_name, k):
        self.image_name = image_name
        self.image = img.imread(f'images/{image_name}.jpg')
        self.k = k

    def perform_left_shift(self) -> np.array:
        image = img.imread(f'images/{self.image_name}.jpg')
        for r in range(0, image.shape[0]):
            for g in range(0, image.shape[1]):
                for b in range(0, image.shape[2]):
                    image[r][g][b] = min([image[r][g][b] + self.k, 255])
        return image
```

```

def _plot_result(self, preprocessed_image: np.array):
    original_image_label = 'Original image'
    illuminated_image_label = f"Illuminated image (k={self.k})"
    labels = [original_image_label, illuminated_image_label]
    images = [self.image, preprocessed_image]
    fig, axes = plt.subplots(ncols=len(images))
    for idx, ax in enumerate(axes):
        ax.imshow(images[idx])
        ax.set_title(labels[idx])
        ax.axis("off")
    plt.show()

def show_illuminated_image(self):
    """ Show origin and illuminated image in a figure"""
    preprocessed_image = self._perform_left_shift()
    self._plot_result(preprocessed_image)

if __name__ == '__main__':
    existing_images = [os.path.splitext(f)[0] for f in os.listdir("images") if
f.endswith('.jpg')]
    for number, image_name in enumerate(existing_images):
        print(f"{number + 1}. {image_name}")
    number = int(input("\nProvide a number of image name from the list above:
"))
    incorrect_image_number_message = (f"!!! The image does not exist\n"
f"Please provide a correct number of image
name from the list above: ")
    validate_input_number(number, 1, len(existing_images) + 1,
incorrect_image_number_message)
    image_name = existing_images[number - 1]
    k = int(input("Provide k: "))
    incorrect_k_message = ("!!! K must satisfy the condition 0 < k < 255\n"
        "Please provide a correct number: ")
    validate_input_number(number, 0, 255, incorrect_k_message)
    image_preprocessing_obj = ImagePreprocessing(image_name, k)
    image_preprocessing_obj.show_illuminated_image()

```