**Минобрнауки РФ**

**ЮЗГУ**

**ФФиПИ**

Кафедра программной инженерии

**ЛАБОРАТОРНАЯ РАБОТА №3**

**ПОПИКСЕЛЬНЫЕ ПРЕОБРАЗОВАНИЯ**

Выполнил: ст. гр. ПО-22б

Алешин К.Р.

Проверил: Ефремов В. В.

Курск, 2024 г

**Задание:**  
****

**Листинг программы:**from PIL import Image

def invert\_pixel(pixel):

"""Инверсия цвета одного пикселя (R, G, B)."""

r, g, b = pixel

return 255 - r, 255 - g, 255 - b

def invert\_image(image):

"""Попиксельная обработка: инверсия цветов всего изображения."""

width, height = image.size

pixels = image.load()

for x in range(width):

for y in range(height):

pixels[x, y] = invert\_pixel(pixels[x, y])

return image

def blend\_images(image1, image2, alpha=0.5):

"""

Наложение двух изображений вручную.

- image1, image2: изображения одинакового размера.

- alpha: коэффициент смешивания (0 - только image1, 1 - только image2).

"""

if image1.size != image2.size:

raise ValueError("Размеры изображений должны совпадать!")

width, height = image1.size

pixels1 = image1.load()

pixels2 = image2.load()

# Результирующее изображение

blended\_image = Image.new("RGB", (width, height))

blended\_pixels = blended\_image.load()

for x in range(width):

for y in range(height):

r1, g1, b1 = pixels1[x, y]

r2, g2, b2 = pixels2[x, y]

# Смешивание цветов пикселей

blended\_pixels[x, y] = (

int(r1 \* (1 - alpha) + r2 \* alpha),

int(g1 \* (1 - alpha) + g2 \* alpha),

int(b1 \* (1 - alpha) + b2 \* alpha)

)

return blended\_image

def main():

# Пути к файлам

input\_image1\_path = "input1.jpg"

input\_image2\_path = "input2.jpg"

output\_inverted\_path = "inverted\_image.jpg"

output\_blended\_path = "blended\_image.jpg"

# Чтение изображений

image1 = Image.open(input\_image1\_path).convert("RGB")

image2 = Image.open(input\_image2\_path).convert("RGB")

# Попиксельная обработка: инверсия цветов

inverted\_image = invert\_image(image1.copy())

inverted\_image.save(output\_inverted\_path)

print(f"Обработанное изображение сохранено как {output\_inverted\_path}")

# Наложение изображений

blended\_image = blend\_images(image1, image2, alpha=0.5)

blended\_image.save(output\_blended\_path)

print(f"Наложенное изображение сохранено как {output\_blended\_path}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Пример работы программы:  
Изображение выглядит как снимок экрана, компьютер

Автоматически созданное описание**

**Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение

Автоматически созданное описание**