ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Звіт до індивідуального завдання №1

з предмету

«Обробка зображень та мультимедіа»

Виконав:

студент групи ПМі-43

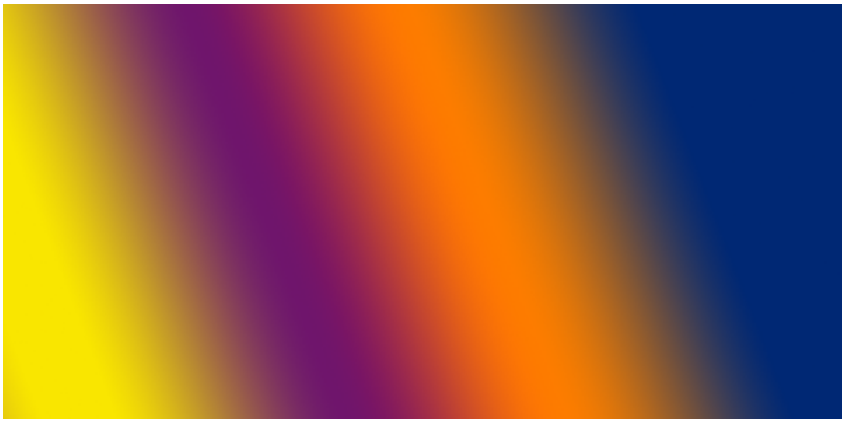
Ваврикович Михайло

Викладач:

доц. Гутік Олег

2023

1. Для виконання цієї роботи, я завантажив 24-бітний файл у форматі bmp



Для виконання наступних кроків, використовував IDE VS Code та бібліотеку Python Imaging Library, для роботи з зображеннями.

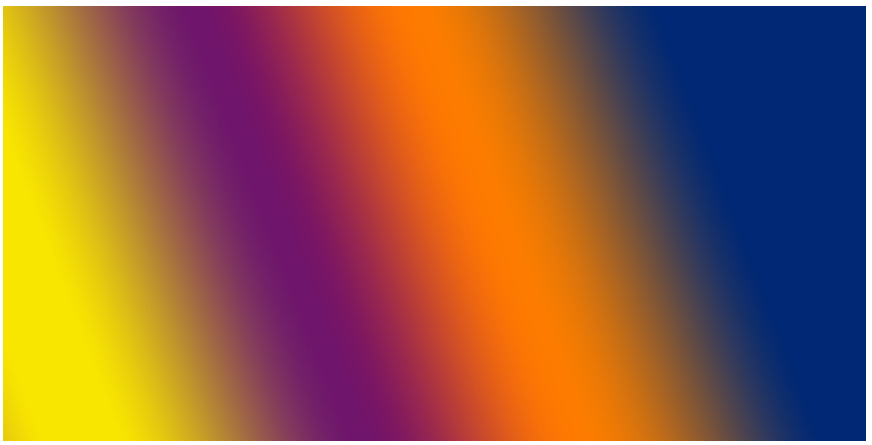
Відкриваю зображення за допомогою команди:

```
original_image = Image.open(bmp_path)
```

Зберігаю зображення в форматі BMP використовуючи стиснення RLE:

```
# 1. Збереження у форматі BMP (RLE)
bmp_rle_path = os.path.join(save_folder, "Image_rle.bmp")
start_time = time.time()
original_image.save(bmp_rle_path, format="BMP", compression="RLE")
end_time = time.time()
bmp_rle_size = os.path.getsize(bmp_rle_path)
```

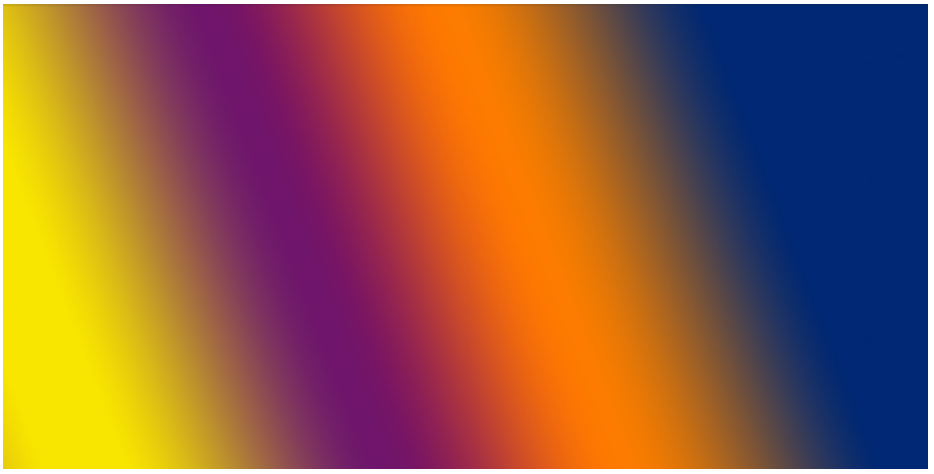
Отриманий результат:



Зберігаю зображення в форматі TIFF використовуючи стиснення LZW:

```
# 2. Збереження у форматі TIFF (LZW)
tiff_path = os.path.join(save_folder, "Image_lzw.tiff")
start_time = time.time()
original_image.save(tiff_path, format="TIFF", compression="LZW")
end_time = time.time()
tiff_size = os.path.getsize(tiff_path)
```

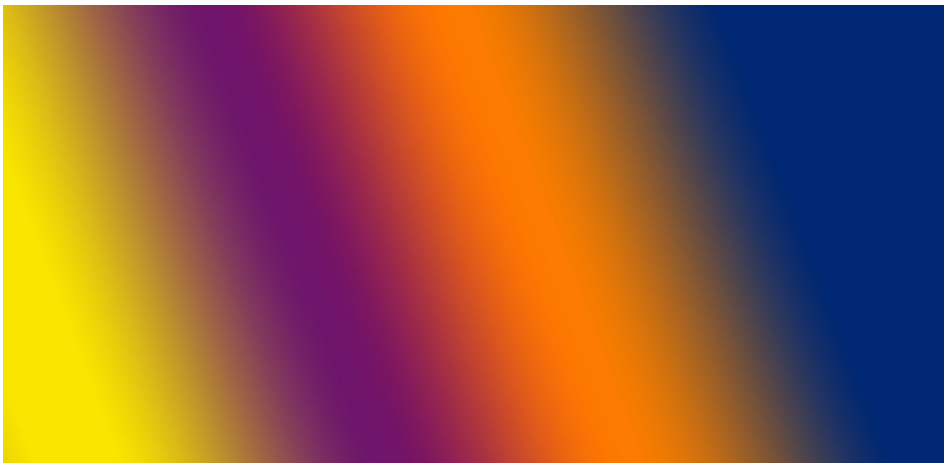
Отриманий результат:



Зберігаю зображення в форматі JPEG використовуючи Standart Encoding:

```
3. Збереження у форматі JPEG
eg_path = os.path.join(save_folder, "Image.jpg")
art_time = time.time()
iginal_image.save(jpeg_path, format="JPEG", quality=100)
d_time = time.time()
eg_size = os.path.getsize(jpeg_path)
```

Отриманий результат:



Порівняємо час збереження та розміри утворених файлів:

```
1. Збережено зображення у форматі BMP (RLE) в папці ToSave:
   Час збереження: 0.0078 сек.
   Розмір файлу: 1464.90 KB

2. Збережено зображення у форматі TIFF (LZW) в папці ToSave:
   Час збереження: 0.0030 сек.
   Розмір файлу: 1465.12 KB

3. Збережено зображення у форматі JPEG в папці ToSave:
   Час збереження: 0.0038 сек.
   Розмір файлу: 135.32 KB
```

Зображення у форматі BMP зайняло найбільше часу для зберігання. А зображення у форматі JPEG займає найменше місця.

2. Для віднімання втрат, я переводжу зображення в масиви з пікселями

```python
# Перетворюємо зображення у масиви NumPy
original_array = np.array(original_image)
jpeg_array = np.array(jpeg_image)
```

Далі віднімаю значення пікселів зображення у форматі JPEG від оригінального зображення. Та конвертую масиви назад в зображення

```python
# 1. Віднімання для всіх кольорів разом
difference_all_colors = (
    original_array - jpeg_array
)  # Віднімаємо значень пікселів оригінального зображення від значень пікселів зображення у форматі JPEG
difference_all_colors_image = Image.fromarray(
    np.uint8(difference_all_colors)
)  # конвертуємо назад в зображення
difference_all_colors_path = save_folder + "/all_colors_difference_jpeg.bmp"
difference_all_colors_image.save(difference_all_colors_path)
```

Такі ж дії проводжу для масивів де ми обнулили значення R, G, та B відповідно:

```python
# 2. Віднімання для кожного кольору окремо (R, G, B)
difference_R = original_array.copy()
difference_R[:, :, 1:3] = 0  # Обнулити G та B
difference_R = difference_R - jpeg_array

difference_G = original_array.copy()
difference_G[:, :, 0] = 0  # Обнулити R та B
difference_G[:, :, 2] = 0
difference_G = difference_G - jpeg_array

difference_B = original_array.copy()
difference_B[:, :, 0:2] = 0  # Обнулити R та G
difference_B = difference_B - jpeg_array
```

Всі утворені зображення конвертую, та зберігаю:

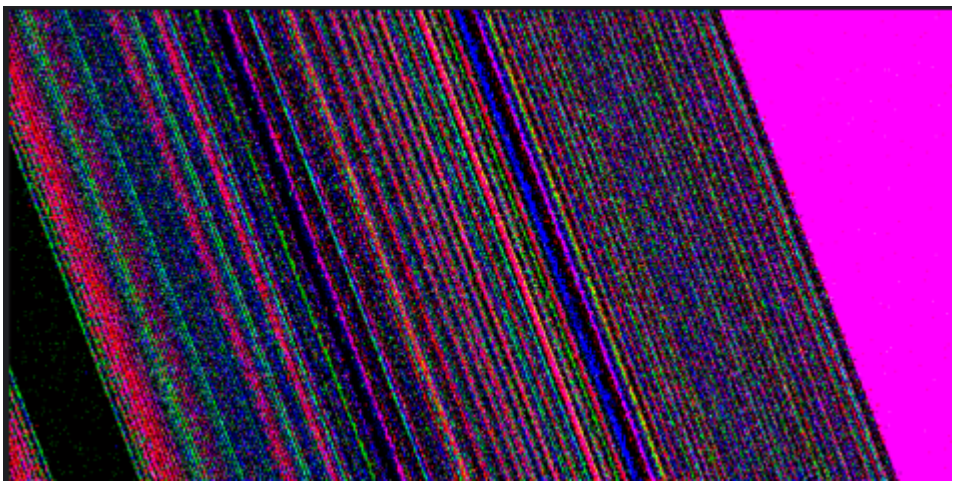
```
# Збереження зображень
difference_R_image = Image.fromarray(np.uint8(difference_R))  # конвертуємо в зображення
difference_R_path = save_folder + "/difference_R_for_jpeg.bmp"
difference_R_image.save(difference_R_path)

difference_G_image = Image.fromarray(np.uint8(difference_G))
difference_G_path = save_folder + "/difference_G_for_jpeg.bmp"
difference_G_image.save(difference_G_path)

difference_B_image = Image.fromarray(np.uint8(difference_B))
difference_B_path = save_folder + "/difference_B_for_jpeg.bmp"
difference_B_image.save(difference_B_path)
```

Аналогічні дії проводжу для зображень інших форматів.

Далі наведені результати таких дій:

Віднімання всіх кольорів

JPEG:

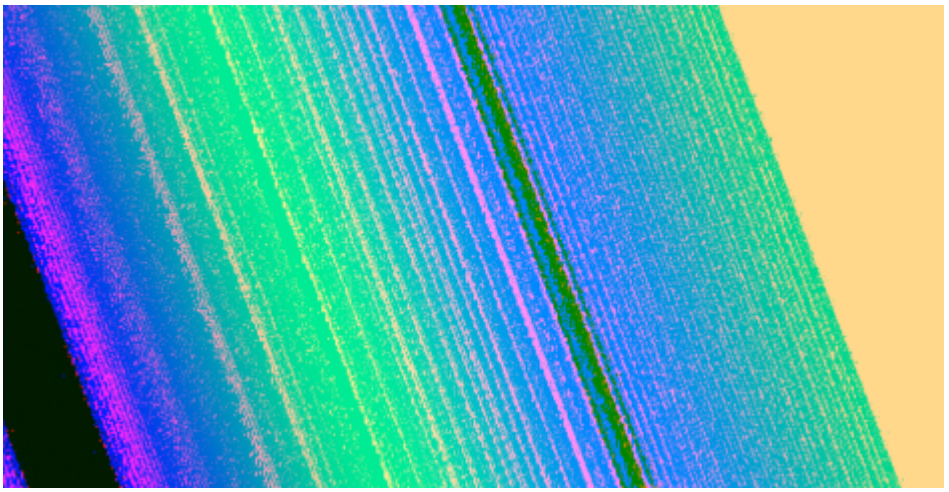


BMP:

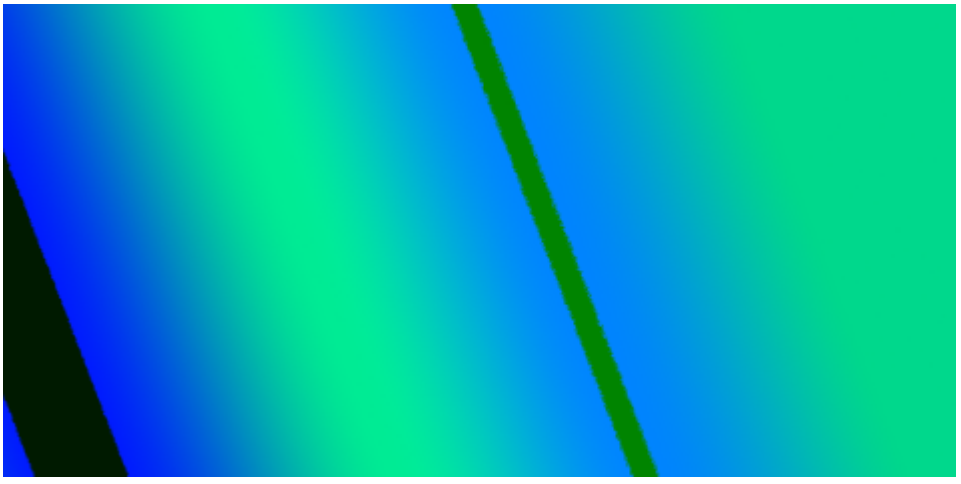
TIFF:

Віднімання R

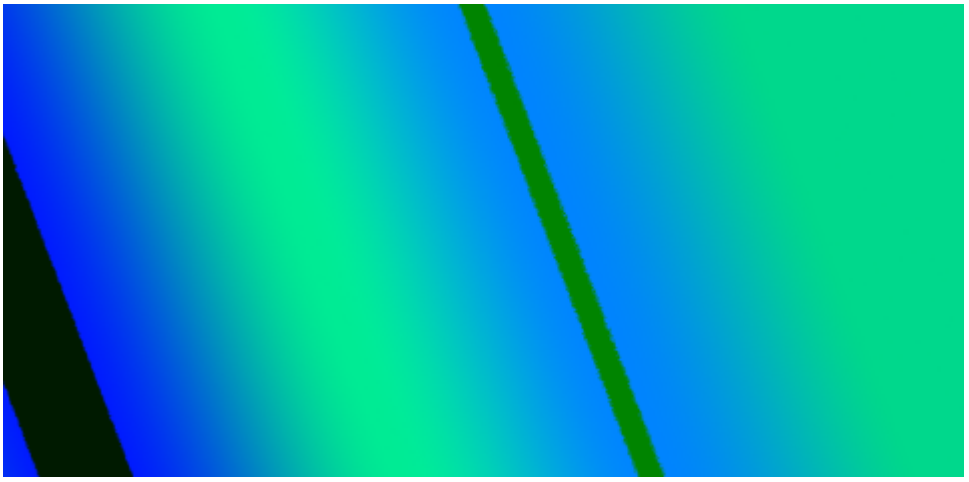
JPEG:



BMP:

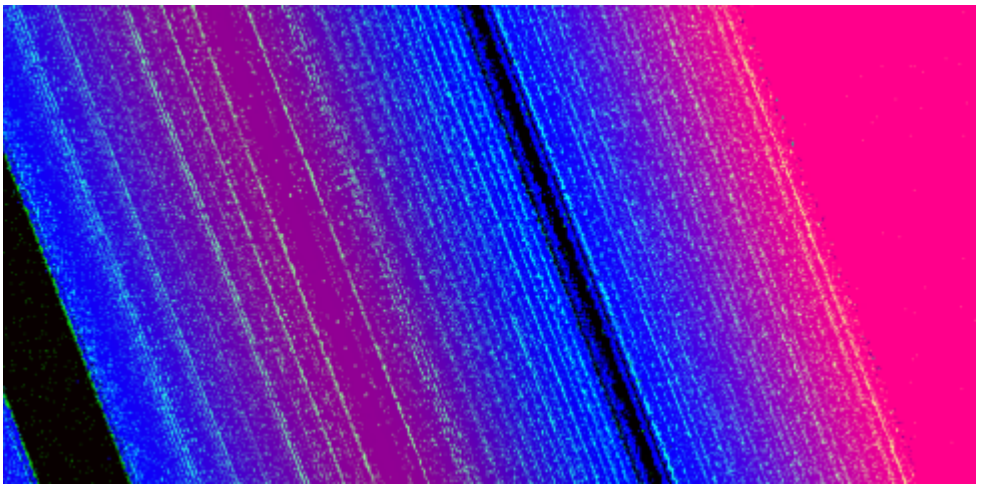


TIFF:

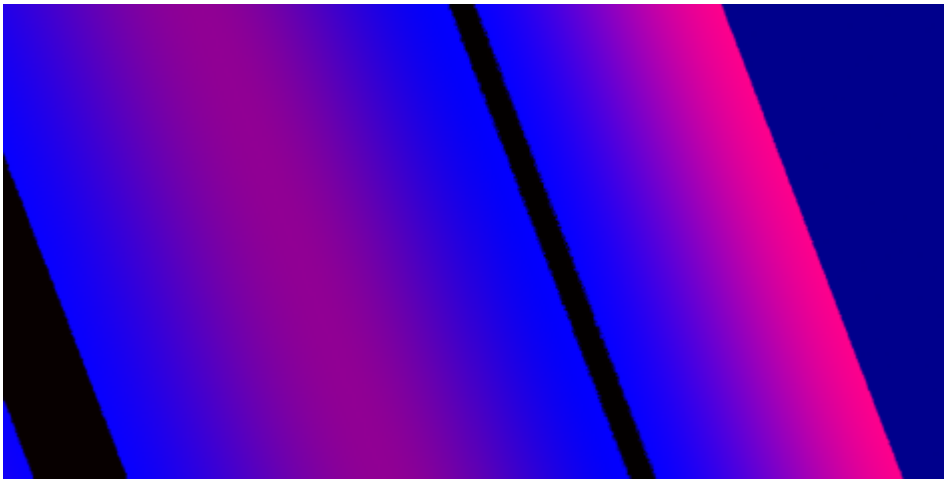


Віднімання G

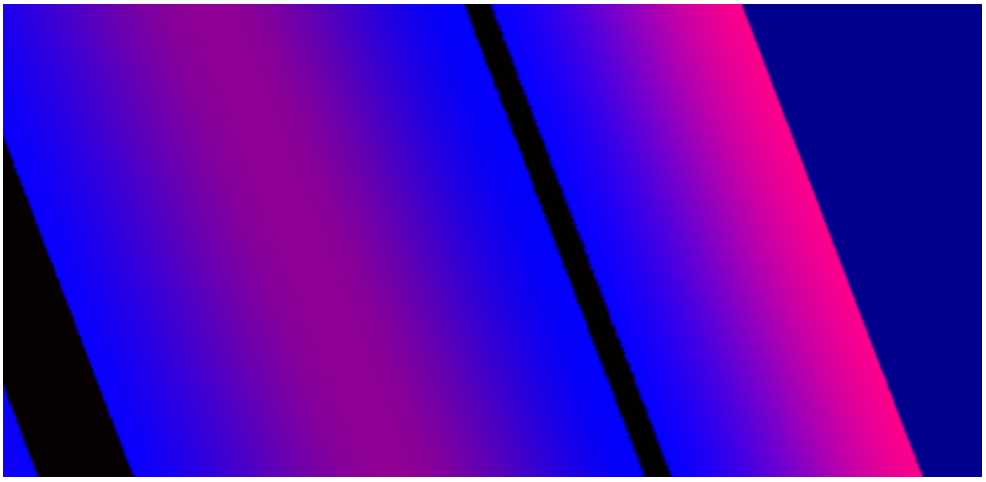
JPEG:



BMP:

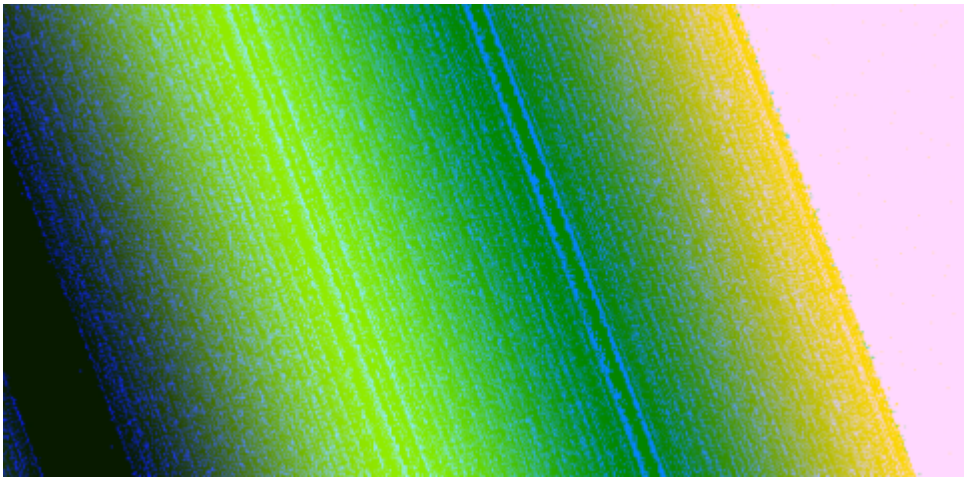


TIFF:



Віднімання B

JPEG:



BMP:

TIFF:

Також я порахував втрати під час стиснення зображення.

Рахував різниці між пікселями оригінального та стиснутого зображення. Якщо значення пікселя дорівнює 0, це означає, що для цього пікселя втрат немає.

Отримані результати:

```
Загальні втрати для jpeg: 72.08914066666667
Втрати для R для jpeg: 88.254422
Втрати втрати для G для jpeg: 31.099766
Втрати втрати для B для jpeg: 96.913234


Загальні втрати для bmp: 0.0
Втрати для R для bmp: 0.0
Втрати для G для bmp: 0.0
Втрати для B для bmp: 0.0


Загальні втрати для tiff: 0.0
Втрати для R для tiff: 0.0
Втрати для G для tiff: 0.0
Втрати для B для tiff: 0.0
```

Тут ми бачимо, що втрати для jpeg достатньо великі, а інші формати втрат взагалі не зазнали.

Висновки:

Отже, Формат BMP із стисненням RLE надійно зберігає зображення із збереженням всієї інформації. Проте, розмір файлу може бути великим у порівнянні з іншими форматами.

Формат JPEG використовує стиснення з втратами, одразу помітна втрата якості зображення. Але порівняно з іншими форматами, розмір файлу тут дуже малий.

Тому BMP (RLE) та TIFF(LZW) відмінно підходять для зберігання зображень без втрат, а JPEG підходить для випадків, коли розмір файлу є ключовим фактором.