

Lab5 - Zmiana wartości pikseli – Metody na obrazie

1. **im.getpixel** i **im.putpixel**
2. **im.load**
3. **im.point**

1. Metoda **im.getpixel((i , j))** pobiera wartość piksela z miejsca o współrzędnych (i , j) obrazu im. Metoda **im.putpixel((i , j), (a, b, c))** wstawia w obraz im w miejscu (i , j) piksel o wartości (a, b, c). Oczywiście format wartości piksela zależy od trybu obrazu. Zaleca się stosować te metody, gdy chcemy dokonać podmiany niewielu pikseli, bo metody działają wolno.
2. Metoda **im.load()** ładuje wszystkie piksele do tablicy tymczasowej, na przykład **pix = im.load()** skutkuje tym, że powstaje tymczasowa tablica **pix** zawierająca wartości pikseli z obrazu im. Adresy pikseli w tej tablicy są identyczne jak adresy w obrazie, **pix [i, j]** jest równoważne **im.getpixel((i , j))**, natomiast **pix [i, j] = (a, b, c)** jest równoważne **im.putpixel((i , j), (a, b, c))**. Zaleca się stosować tę metodę, gdy zmieniamy wiele pikseli różnymi nawet skomplikowanymi funkcjami. Metoda znacznie szybsza niż getpixel i putpixel.
3. Metoda **im.point** jest szybką metodą wykorzystywaną do zastosowania funkcji na wszystkich pikselach (i każdym z osobna). W połączeniu z funkcją lambda redukuje kod „do jednej linijki”.

Na przykład

```
im.point(lambda i: i * a + b)
```

działa tak, że wartość każdego piksela **i** obrazu im zmienia się zgodnie z funkcją $i * a + b$, gdzie **a** i **b** są parametrami. Jest to przykład filtru liniowego.

Uwaga: metody 1 i 2 działają bezpośrednio na obrazie, dlatego zalecane jest działanie na kopii obrazu oryginalnego.

Zadania

1. Wczytaj pod nazwą **obraz** obrazek wybrany na poprzednich zajęciach. Przekształcenia wykonuj na kopii tego obrazu. Wczytaj obraz z inicjałami pod nazwą **inicjaly**.
2. Korzystając z metod getpixel i putpixel:

2.1 Napisz funkcję `wstaw_inicjaly(obraz, inicjaly, m, n, kolor)`, gdzie `m, n` są współrzędnymi punktu, w którym wstawimy w `obraz, inicjaly` w kolorze równym wartości `kolor`. Utwórz i zapisz `obraz1`, w którym inicjały w kolorze czerwonym są wstawione tak, że prawy dolny róg obrazu inicjały pokrywa się z prawym dolnym rogiem obrazu.

2.2 Analogicznie do funkcji `rozjasnij_obraz_z_maska`, napisz funkcję `wstaw_inicjaly_maska(obraz, inicjaly, m, n, x, y, z)`, gdzie `m, n` są współrzędnymi punktu, w którym traktując `inicjaly` jako maskę zmienimy piksele `(a, b, c)` odpowiadające czarnym pikselom z maski na `(a+x, b+y, c+z)`. Utwórz i zapisz `obraz2`, w którym maska jest wstawiona mniej więcej na środku obrazu.

3. Stosując metodę `load` napisz funkcje

`wstaw_inicjaly_load(obraz, inicjaly, m, n, kolor)` oraz `wstaw_inicjaly_maska(obraz, inicjaly, m, n, x, y, z)` działające identycznie jak funkcje z pkt. 2.1, 2.2. Przetestuj te funkcje, ale nie zapisuj obrazów.

4. Stosując metodę `point` i funkcję `lambda`:

4.1 Napisz funkcję `kontrast(obraz, wsp_kontrastu)`, która działa tak, że każdy piksel `i` zmienia wartość zgodnie z funkcją kontrastu

$f(i) = 128 + (i - 128) * mn$, gdzie

$mn = ((255 + \text{wsp_kontrastu}) / 255) ** 2$, a `wsp_kontrastu` przyjmuje wartości o 0 do 100

Przetestuj różne wartości `wsp_kontrastu` i napisz w raporcie jak wartości `wsp_kontrastu` wpływają na uzyskany efekt (słownie i wklejając obrazek)

4.3 Napisz funkcję `transformacja_logarytmiczna(obraz)`, która działa tak, że każdy piksel `i` zmienia wartość zgodnie z funkcją

$f(i) = 255 * \text{np.log}(1 + i / 255)$

Przetestuj i napisz w raporcie jaki jest efekt (słownie i wklejając obrazek)

4.4 Napisz funkcję `transformacja_gamma (obraz, gamma)`, która działa tak, że każdy piksel `i` zmienia wartość zgodnie z funkcją

$f(i) = (i / 255) ** (1 / \text{gamma}) * 255$, gdzie `gamma` jest współczynnikiem większym od zera.

Przetestuj różne wartości gamma i napisz w raporcie jak wartości gamma wpływają na uzyskany efekt (słownie i wklejając obrazek)

4. Napisz, dlaczego obraz powstały z zastosowania poleceń

```
T = np.array(obraz, dtype='uint8')
```

```
T += 100
```

```
obraz_wynik = Image.fromarray(T, "RGB")
```

jest inny niż obraz powstały z zastosowania funkcji

```
obraz.point(lambda i: i + 100)
```

Wskazowka: co się dzieje, gdy wartość elementu tablicy przekracza 255 w typie uint8, oraz co się dzieje w przypadku obrazu.

5. Napisz funkcję, która działa na tablicy obrazu i daje taki sam efekt, co

```
obraz.point(lambda i: i + 100)
```

Raport, plik z kodem oraz wszystkie obrazy zaznaczone na czerwono wstaw na Moodle.