

---

# PPPD - Lab. 08

Copyright ©2021 M. Śleszyńska-Nowak i in.

---

Zadanie dotyczy przetwarzania obrazów (bitmap) w skali szarości.

Wklej kod poniższych funkcji:

```
import numpy
from PIL import Image

def png_read(filepath):
    img = Image.open(filepath)
    assert len(img.size)==2 # skala szarosci, nie RGB
    return (numpy.array(img)/255).reshape(img.size[1], img.size[0]).tolist()

def png_write(img, filepath):
    img = Image.fromarray((numpy.array(img)*255).astype(numpy.int8), 'L')
    img.save(filepath)
```

Wywołując następujący kod, otrzymasz w wyniku macierz  $A$  taką, że  $A[i][j]$  podaje informację o stopniu szarości (0-czarny, 1-biały) piksela w  $i$ -tym wierszu,  $j$ -tej kolumnie.

```
A = png_read("skimage_astronaut.png")
```

Pokażemy, że dokonywanie różnych prostych operacji arytmetycznych na macierzy  $A$  prowadzić może do ciekawych efektów wizualnych. Aby podejrzeć zmiany, zapisz macierz do pliku (kod poniżej) i otwórz go w jakiejś przeglądarce obrazków.

```
png_write(A, "output.png")
```

1. Znajdź i wypisz na konsolę minimalną i maksymalną wartość intensywności wszystkich pikseli, tj. minimum i maksimum z macierzy reprezentującej obrazek. Tym samym upewnij się, że intensywności pikseli są rzeczywiście w przedziale  $[0, 1]$ .
2. Dokonaj następujących przekształceń:
  - rozjaśnianie (lub przyciemnianie): dodaj do każdego piksela jakąś wartość  $b \in [-1, 1]$ . Piksele o jasności  $> 1$  powinny być „ucięte” do 1, a  $< 0$  do 0;
  - negatyw: odwróć wartość intensywności wszystkich pikseli, tj. zamień  $A[i][j]$  na  $1-A[i][j]$ ;
  - konwersja na obraz czarno-biały: piksele o intensywności  $> p$  dla pewnego  $p \in [0, 1]$  zmień na 1, a pozostałe – na 0;
  - rozciąganie kontrastu: zamień  $A[i][j]$  na  $f(A[i][j])$ , gdzie  $f(c) = \frac{1}{1+e^{-\theta(c-0.5)}}$  dla pewnego  $\theta \geq 1$ .
3. Zaimplementuj funkcję `convolution(A, B)`, która wyznacza tzw. spłot macierzy  $A$  typu  $n \times m$  z macierzą  $B$  typu  $(2k+1) \times (2k+1)$  (dla pewnego małego  $k$ ). W wyniku jej działania otrzymujemy macierz  $C$  typu  $n \times m$  taką, że

$$c_{i,j} = \sum_{u=-k}^k \sum_{v=-k}^k a_{i+u,j+v} \cdot b_{u+k+1,v+k+1},$$

---

przy założeniu  $a_{i,j} = 0$  dla  $i \notin \{1, \dots, n\}$  lub  $j \notin \{1, \dots, m\}$ .

4. Rozważ obrazy powstające przez splot oryginalnej bitmapy  $A$  z następującymi macierzami  $B$  („jądrami”, ang. *kernels*):
- rozmycie: macierz  $B$  zawiera identyczne wartości dodatnie, wszystkie elementy sumują się do 1 (rozważ różne  $k = 3, 5, \dots, 15$ );
  - inne przykładowe ze strony [https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)).