

# Laboratorium 2

Biblioteki: numpy, matplotlib, scikit-image, scipy

Celem laboratorium 2. jest poznanie podstawowych przekształceń afinicznych obrazów oraz metod interpolacji.

## Zadanie 1:

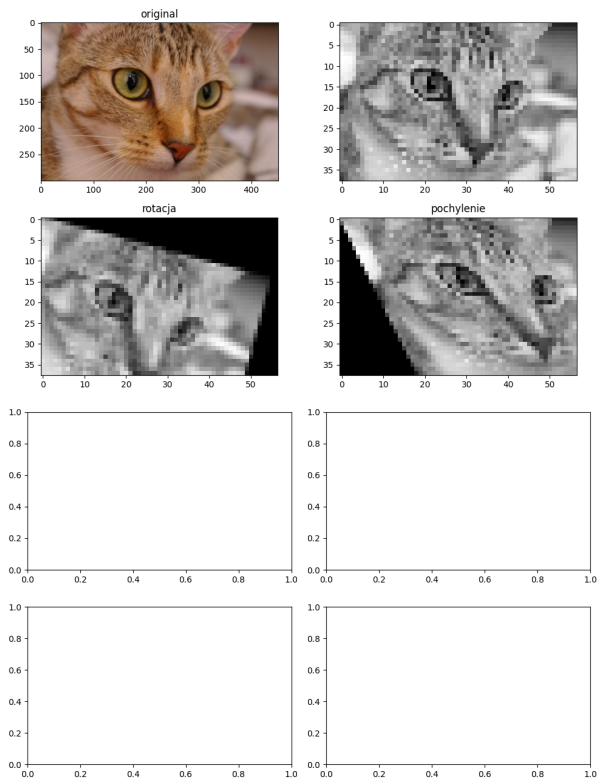
- Zadeklarować wykres składający się z 2 kolumn i 4 wierszy, o wymiarach 10 na 13 cali.
- Wczytać obraz `chelsea` z modułu `data` biblioteki `scikit-image`. Narysować obraz w pierwszej komórce wykresu.
- Obraz należy przekształcić w obraz monochromatyczny uśredniając wszystkie kanały barwne przy życiu metody `np.mean` oraz zredukować ośmiokrotnie jego rozmiar używając adresacji `::8, ::8`. Efekt przedstawić w kolejnej komórce wykresu używając colormapy `binary_r`.
- Użyć funkcji `AffineTransform` oraz `warp` z biblioteki `scikit-image` do przekształcenia obrazu przy pomocy macierzy. Wykonać dwie **oddzielne** transformacje obrazu bazowego monochromatycznego:
  - rotacja o kąt  $\pi/12$
  - pochylenie w osi X o 0.5.

Przykład użycia funkcji `AffineTransform` i `warp`:

```
transform = AffineTransform(matrix = matrix)
img = warp(original_image, transform.inverse)
```

- Wyświetlić efekty transformacji w drugim wierszu wykresu.

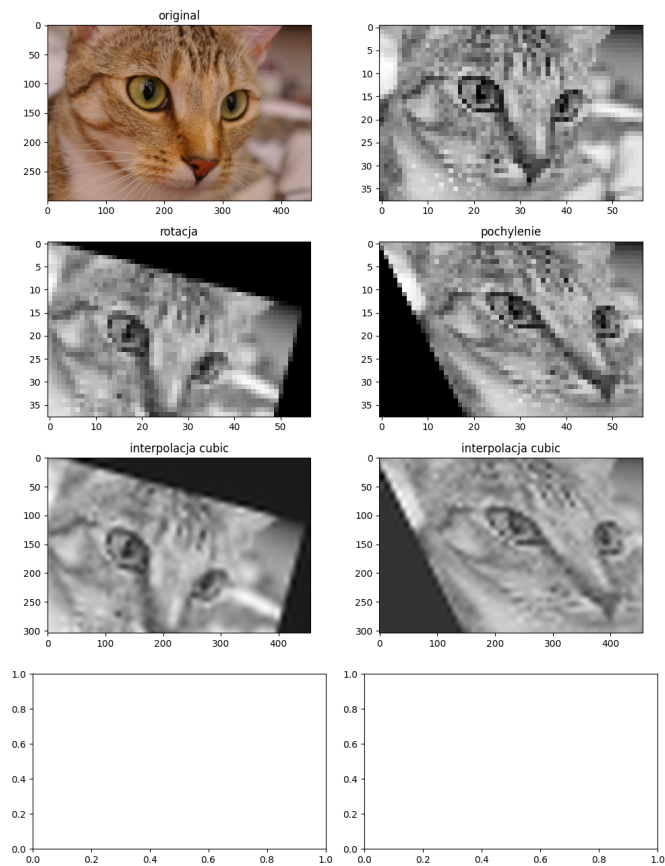
Efekt zadania 1:



## Zadanie 2:

- Wykonać interpolację bikubiczną (`kind='cubic'`) obrazu po transformacji. Należy wykorzystać funkcję `interp2d` z biblioteki `scipy` w celu zwiększenia rozdzielczości obrazu ośmiokrotnie.
- Wynik interpolacji wyświetlić w trzecim wierszu wykresu.
- Wyświetlić wartości pikseli dowolnego obrazu po interpolacji z pierwszych 15 kolumn i 15 wierszy w konsoli, z dokładnością do 1 miejsca po przecinku.

Efekt zadania 2:

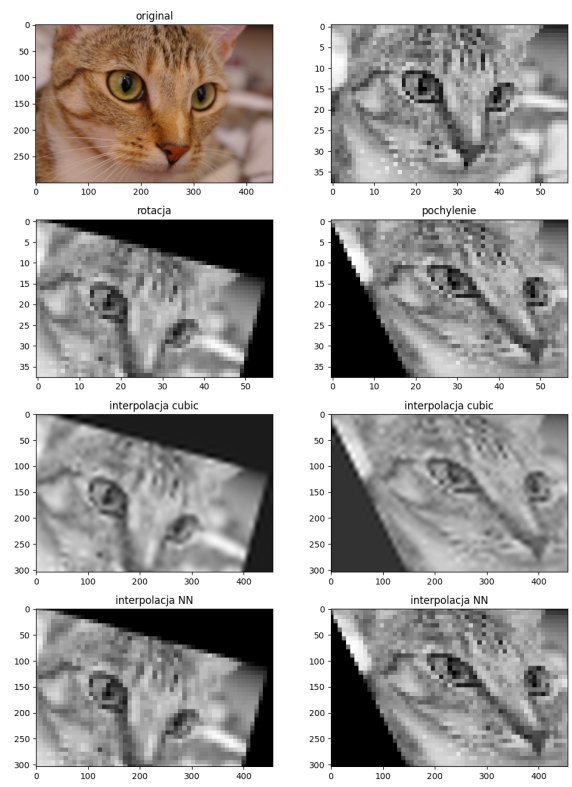


```
[[0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.7 0.7]
 [0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6]
 [0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6]
 [0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6]
 [0.5 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6]
 [0.5 0.5 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
 [0.4 0.5 0.5 0.6 0.6 0.6 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
 [0.4 0.5 0.5 0.6 0.6 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
 [0.3 0.4 0.5 0.6 0.6 0.7 0.7 0.7 0.7 0.8 0.7 0.7 0.7 0.7 0.7 0.7]
 [0.3 0.4 0.5 0.6 0.6 0.7 0.7 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.7 0.7]
 [0.2 0.4 0.5 0.6 0.7 0.7 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8]
 [0.2 0.3 0.5 0.6 0.7 0.7 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8]
 [0.1 0.3 0.4 0.6 0.7 0.7 0.8 0.8 0.9 0.9 0.9 0.9 0.9 0.8 0.8 0.8]
 [0.1 0.3 0.4 0.6 0.7 0.7 0.8 0.8 0.9 0.9 0.9 0.9 0.9 0.8 0.8]]
```

### Zadanie 3:

- Wykonać interpolację najbliższego sąsiada wyłącznie z użyciem biblioteki `numpy`. Rozdzielczość obrazu należy zwiększyć ośmiokrotnie.
- Wynik interpolacji wyświetlić w czwartym wierszu wykresu.
- Wyświetlić wartości pikseli dowolnego obrazu po interpolacji z pierwszych 15 kolumn i 15 wierszy w konsoli, z dokładnością do 1 miejsca po przecinku.

Efekt zadania 3:



```
[0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.7 0.7]
[0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.7 0.7]
[0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.7 0.7]
[0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.7 0.7]
[0.4 0.4 0.4 0.4 0.4 0.4 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
[0.4 0.4 0.4 0.4 0.4 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
[0.4 0.4 0.4 0.4 0.4 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
[0.4 0.4 0.4 0.4 0.4 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
[0.4 0.4 0.4 0.4 0.4 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
[0.4 0.4 0.4 0.4 0.4 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
[0.  0.  0.  0.  0.  0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.8 0.8]
[0.  0.  0.  0.  0.  0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.8 0.8]]
```