

# Laboratorium 11

słowa kluczowe: klasteryzacja, segmentacja, dane syntetyczne

## Zadanie 1:

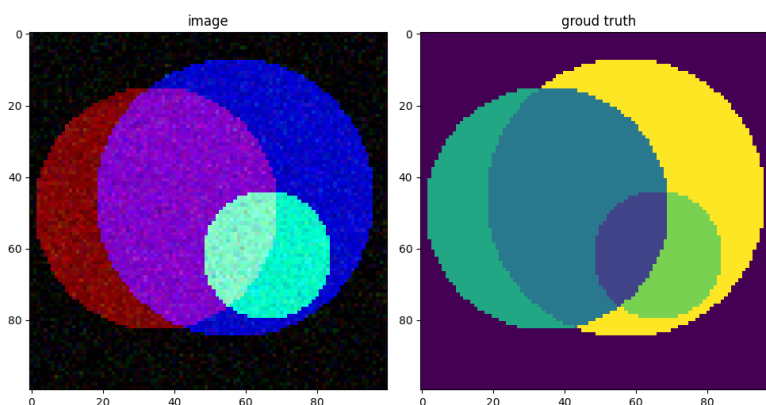
Celem zadania pierwszego jest wygenerowanie *syntetycznego* obrazu cyfrowego, który poddany zostanie klasteryzacji w kolejnych zadaniach.

- Przygotuj barwny obraz  $100 \times 100$  pikseli (3 kanały barwne), składający się z samych zer oraz macierz *ground-truth*  $100 \times 100$ , również składającą się z samych zer.
- Obraz zawierać będzie 3 dyski (`skimage.draw.disk`) o **wylosowanym rozmiarze** z przedziału od 10 do 40. Centrum każdego dysku będzie leżeć na **wylosowanych współrzędnych** obrazu. Warto zadbać, aby dysk w całości mieścił się w obrębie obrazu (odpowiedni wybór współrzędnych).
- Wartość pikseli w **wylosowanym kanale barwnym** obrazu **oraz** w macierzy *ground-truth* w rejonie dysku należy zwiększyć o **wylosowaną wartość** z przedziału (100, 255).
- W *ground-truth* zmień kolejne występujące wartości na indeks etykiet (funkcja `skimage.measure.label`).

Chodzi o to, żeby zamiast wartości unikalnych np. [0, 150, 212] dostać [0, 1, 2].

- Do obrazu dodaj szum cyfrowy z rozkładu normalnego o średniej w zerze i odchyleniu standardowym równym 16. Użyj funkcji `np.clip`, aby ograniczyć wartości pikseli od 0 do 255.
- Obraz i *ground-truth* przedstaw na wykresie.

Przykładowy efekt zadania 1:



## Zadanie 2:

- Obraz i *ground-truth* należy przekształcić w celu przeprowadzenia klasteryzacji.
- Zmień w tym celu kształt obrazu na (*liczba pikseli x liczba kanałów barwnych*).
- Określ współrzędne każdego piksela przy użyciu funkcji `np.meshgrid`. Współrzędne przekształć w wektory.
- Każdy piksel stanie się **wzorcem** i będzie reprezentowany przez **wartości jego kanałów barwnych** oraz **współrzędne**. Należy wobec tego złączyć przekształcony obraz oraz współrzędne tak, aby każdy piksel (teraz już wzorec) opisywany był przez 5 cech. Wynik przechowuj w zmiennej `x`.

- Nową reprezentację poddaj normalizacji **standardowej** w każdej z cech z osobna.
- *Ground-truth* również przekształć w wektor i nazwij go `y`.
- W wierszu poleceń wyświetl:
  - kształt `x` i `y`,
  - pierwszy obiekt z `x` i jego etykietę z `y`.

Przykładowy efekt zadania 2:

```
(10000, 5) (10000, )
[-0.35211148 -0.64595159 -0.6798283 -1.71481604 -1.71481604] 0
```

### Zadanie 3:

- Zadeklaruj metody klasteryzacji: *KMeans*, *MiniBatchKMeans*, *Birch* oraz *DBSCAN*. Wszystkie znajdziesz w bibliotece *scikit-learn* (`sklearn.cluster`). Wykorzystaj domyślne wartości hiperparametrów metod.
- Wykonaj klasteryzację posiadanych danych (`x`) za pomocą zadeklarowanych modeli (funkcja `fit_predict(x)`).
- Wynik klasteryzacji przekształć w obraz dwuwymiarowy  $100 \times 100$  pikseli i przedstaw na wykresie, obok danych oryginalnych i ground-truth.
- Oblicz wartość metryki *adjusted\_rand\_score* (dostępna w `sklearn.metrics`) na podstawie etykiet rzeczywistych (`y`) i tych uzyskanych za pomocą algorytmu klasteryzacji. Wyniki umieść w opisie komórki wykresu, wraz z nazwą metody.

Przykładowy efekt zadania 3:

