

Laboratorium 11

Biblioteki: numpy, matplotlib, scikit-image, scikit-learn

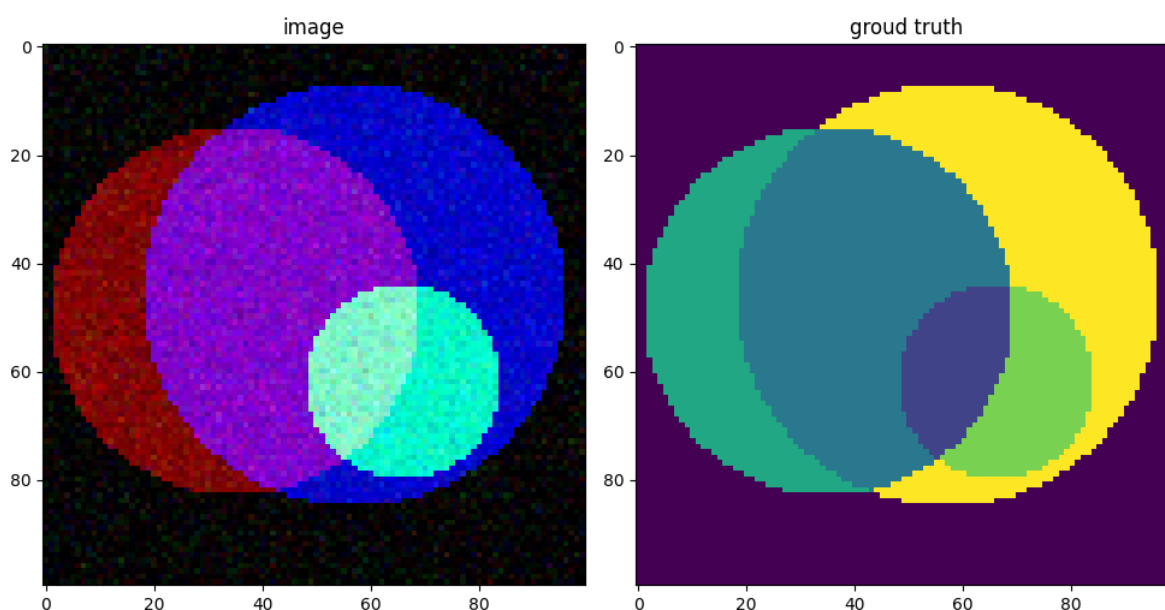
Celem laboratorium 11. jest zapoznanie się generowaniem danych syntetycznych oraz z wykorzystaniem metod klasteryzacji.

Zadanie 1:

Celem zadania pierwszego jest wygenerowanie syntetycznego obrazu, który poddany zostanie następnie klasteryzacji.

- Należy przygotować barwny obraz 100 x 100 x 3 pikseli, składający się z samych zer oraz macierz *ground-truth* 100 x 100, również składającą się z samych zer.
- Obraz zawierać będzie 3 dyski (`skimage.draw.disk`) o losowym rozmiarze z przedziału 10-40. Centrum każdego dysku będzie leżeć w losowym miejscu obrazu. Warto zadbać, aby dysk w całości mieścił się w obrębie obrazu (odpowiedni wybór współrzędnych).
- Wartość piksela w losowym kanale barwnym obrazu oraz w macierzy *ground-truth* w rejonie dysku należy zwiększyć o losową wartość z przedziału (100, 255).
- Warto w *ground-truth* zamienić kolejne wartości na indeksy etykiet (funkcja `skimage.measure.label`).
- Obraz zaszumić rozkładem normalnym o średniej w 0 i odchyleniu standardowym równym 16.
- Użyć funkcji `np.clip`, aby ograniczyć wartości pikseli od 0 do 255.
- Obraz i *ground-truth* przedstawić na wykresie.

Przykładowy efekt zadania 1:



Zadanie 2:

- Obraz i *ground-truth* należy przekształcić w celu przeprowadzenia klasteryzacji.
- Zmienić kształt obrazu (`np.reshape`) na 100*100 x 3. Wynik operacji nazwać `x`.
- Utworzyć współrzędne każdego piksela przy użyciu funkcji `np.meshgrid` :
`xx, yy = np.meshgrid(np.arange(100), np.arange(100))`
- Współrzędne `xx`, `yy` przekształcić w wektory przy użyciu funkcji `reshape` lub `flatten`.
- Każdy piksel stanie się wzorcem i będzie reprezentowany przez wartości jego kanałów barwnych oraz współrzędne. Należy wobec tego złączyć (`np.concatenate`) przekształcony obraz `x` oraz przekształcone współrzędne `xx`, `yy` w osi `axis=1`.
- Wynik poddać normalizacji standardowej.
- *Ground-truth* również przekształcić w wektor i nazwać `y`.
- Wyświetlić w wierszu poleceń:
 - kształt `x` i `y` (`.shape`),
 - 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:

- Zadeklarować metody do klasteryzacji: *KMeans*, *MiniBatchKMeans*, *Birch* oraz *DBSCAN* z biblioteki *scikit-learn* (`sklearn.cluster`).
- Wykonać klasteryzację posiadanych danych (funkcja `fit_predict(X)`).
- Wynik przekształcić w obraz dwuwymiarowy 100 x 100 pikseli i przedstawić na wykresie, obok danych oryginalnych i *ground-truth*.
- Obliczyć wartość metryki *adjusted_rand_score* (dostępna w `sklearn.metrics`) i umieścić w opisie komórki wykresu, wraz z nazwą metody.

Przykładowy efekt zadania 3:

