

Laboratorium 5: Segmentacja obrazów

Superpiksele

1. Wstęp

Elementem kluczowym każdej ścieżki przetwarzania i analizy obrazów w kierunku procesu rozpoznania jest segmentacja obrazów. Ogólnie, termin segmentacji obrazów odnosi się do wyodrębnienia obiektów z tła obrazu. Jednak interpretacja co jest obiektem, a co tłem obrazu jest wyjątkowo subiektywna i zależy od konkretnego problemu.

Formalnie, proces segmentacji obrazu można określić w sposób następujący:

Niech \hat{I} będzie skończonym zbiorem wszystkich pikseli danego obrazu oraz niech $A(\bullet)$ będzie jednoargumentowym predykatem homogeniczności określonym dla dowolnego podzbioru \hat{I} , wiążący piksele. Wtedy proces segmentacji obrazów jest to rozbitcie zbioru \hat{I} w skończoną rodzinę niepustych podzbiorów $I =_{df} \{I_1, I_2, \dots, I_k\}$, takich że:

- i. $I_r \cap I_s = \emptyset$, dla $r \neq s$ ($r, s \in \{1, 2, \dots, k\}$),
- ii. $\bigcup_{s=1}^k I_s = \hat{I}$,
- iii. $A(I_s) = \text{true}$ (dla każdego $I_s \in I$),
- iv. $A(I_r \cup I_s) = \text{false}$ (dla dowolnych par $I_r, I_s \in I$).

Jedną z najbardziej bezpośrednich implementacji powyższej definicji stanowią algorytmy klasteryzacji będące częścią ogólnego podejścia uczenia nienadzorowanego. Najprostszym i często stosowanym algorytmem z tej dziedziny jest algorytm *k-średnich* (ang. *k – means clustering*).

Klasyczny algorytm *k – średnich*, zakłada poniższe kroki:

Niech $X = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$, będzie pewnym niepustym zbiorem danych, określonym w pewnej p wymiarowej przestrzeni euklidesowej R^p ($\bar{x}_m = (x_{m1}, x_{m2}, \dots, x_{mp}) \in R^p$, dla każdego $m = 1, \dots, n$)

1. Podaj liczbę klastrow k , na które rozbijesz cały zbiór danych (w przypadku obrazów – pikseli obrazu),
2. Określ k centrów klastrow: $\bar{m}_1^{(1)}, \bar{m}_2^{(1)}, \dots, \bar{m}_k^{(1)}$
3. Krok przydziału wartości do klastrow, na podstawie założonej metryki: określ zbiór klastrow $\{C_1, C_2, \dots, C_k\}$, gdzie:

$$C_i^{(t)} =_{df} \{ \bar{x}_j \mid \| \bar{x}_j - \bar{m}_i^{(t)} \|^2 \leq \| \bar{x}_j - \bar{m}_q^{(t)} \|^2 \forall q, 1 \leq q \leq k \}$$

(dla każdego \bar{x}_j w każdym kroku, należy przydzielić dokładnie jeden klastrow – dotyczy przypadków, gdy najmniejsza odległość dla określonego elementu jest taka sama dla więcej niż jednego klastra),

4. Krok aktualizacji wartości: określ nowe wartości centrów klastrow, na podstawie przydzielonych do nich elementów:

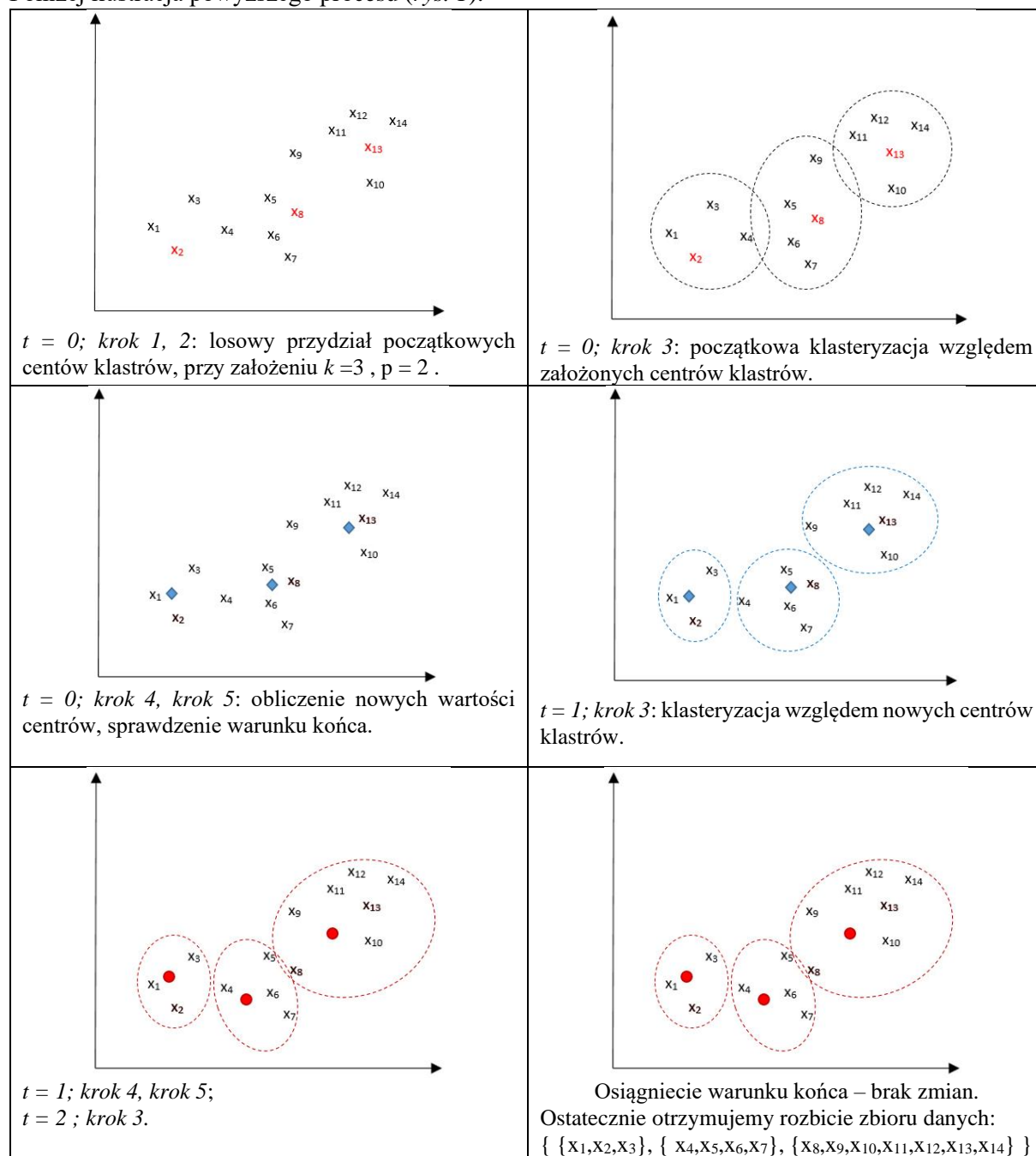
$$\bar{m}_i^{(t+1)} =_{df} \frac{1}{|C_i^{(t)}|} \sum_{\bar{x}_j \in C_i^{(t)}} \bar{x}_j$$

5. Jeśli dla każdego \bar{m}_i , $\bar{m}_i^{(t+1)} = \bar{m}_i^{(t)}$ (lub została osiągnięta założona a priori liczba iteracji) to zakończ algorytm, w przeciwnym przypadku wróć do kroku 3 przyjmując jako nowe wartości centów klastrow $\bar{m}_i^{(t+1)}$ dla każdego $i = 1, \dots, k$.

Problem optymalnej klasteryzacji, można zdefiniować jako:

$$\arg \min_C \sum_{i=1}^k \sum_{\bar{x} \in C_i} \|\bar{x} - \bar{m}_i\|^2$$

Poniżej ilustracja powyższego procesu (rys. 1).



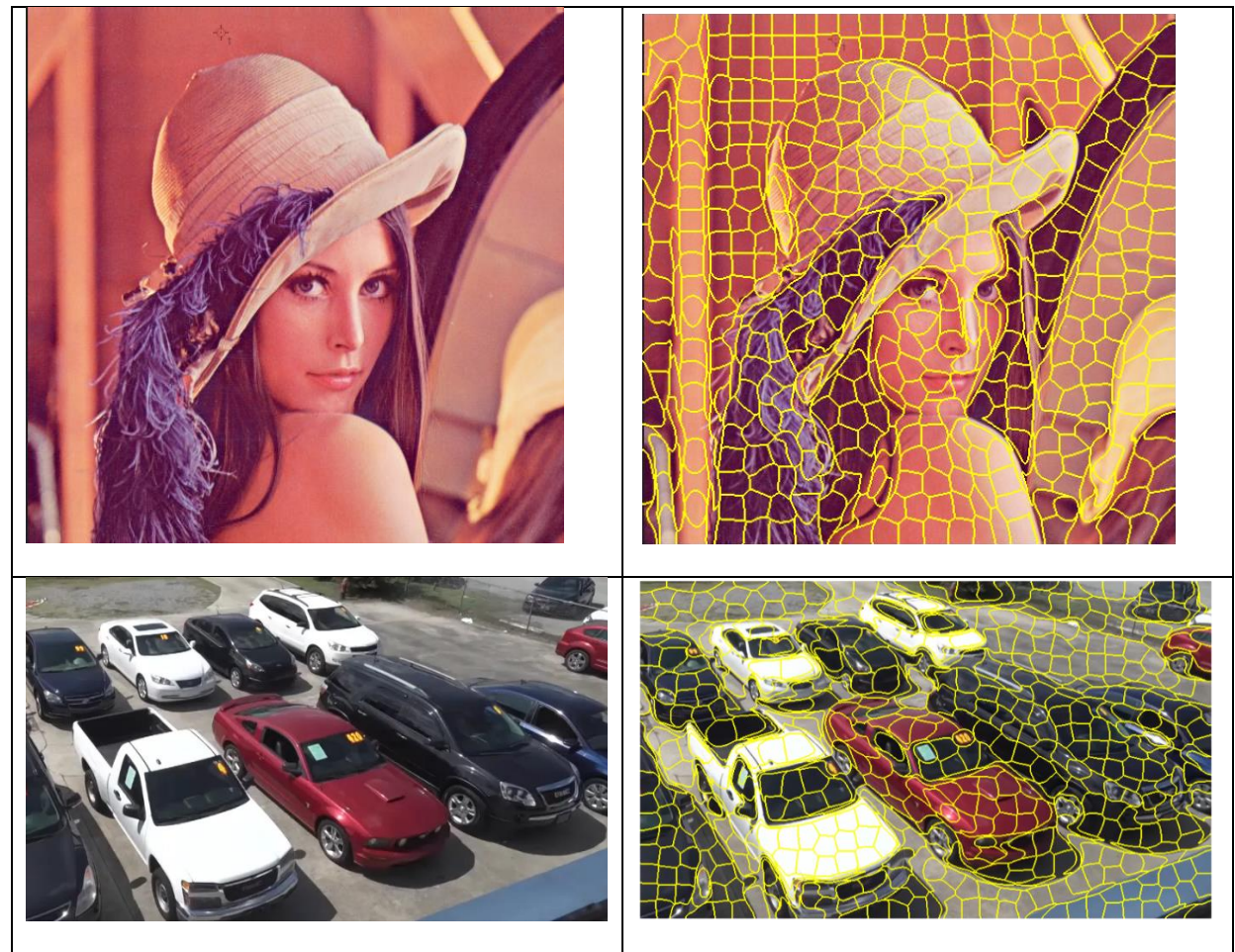
Rys. 1. Przykładowy proces klasteryzacji

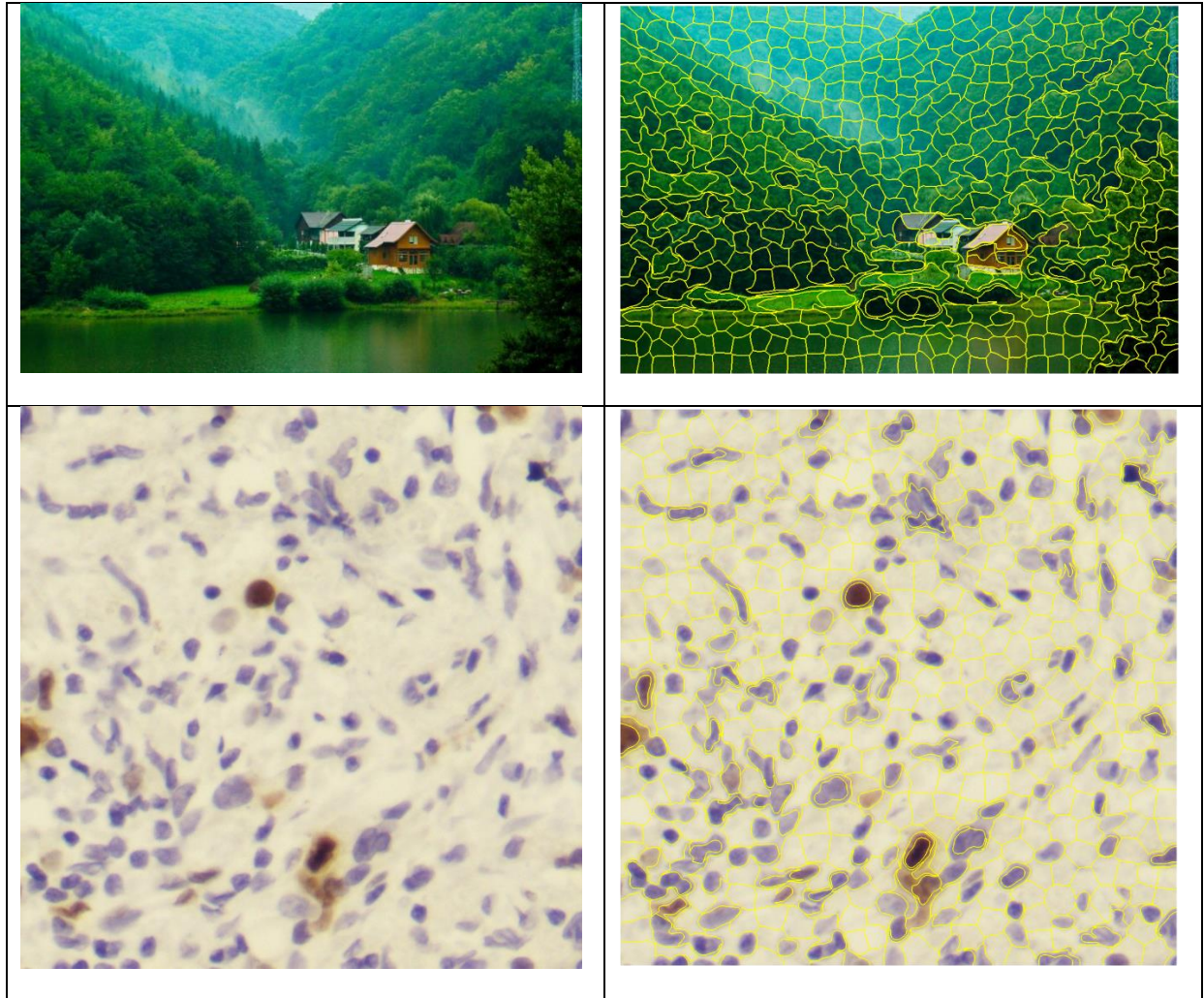
Przy założeniu segmentacji obrazu jako proces rozbicia zbioru jego pikseli, wszelkie algorytmy klasteryzacji bezpośrednio mogą być używane do tego celu.

2. Super piksele

Idea segmentacji obrazów za pomocą super pikseli (*ang. superpixels*) dotyczy procesu grupowania podzbiorów pikseli np. za pomocą procesów klasteryzacji. Pojęcie ‘*superpiksela*’ odnosi się do względnie małych podobszarów obrazów homogenicznych względem przyjętych założeń segmentacji. Super piksele często stanowią dobry preprocessing dla bardziej zaawansowanej segmentacji lub też mogą stanowić dobre wejście dla odpowiedniego modelu uczenia maszynowego w celu rozpoznania obiektów.

Poniżej, kilka przykładów podziału obrazu na super piksele (algorytm SLIC [1], liczba klastrów = 600):



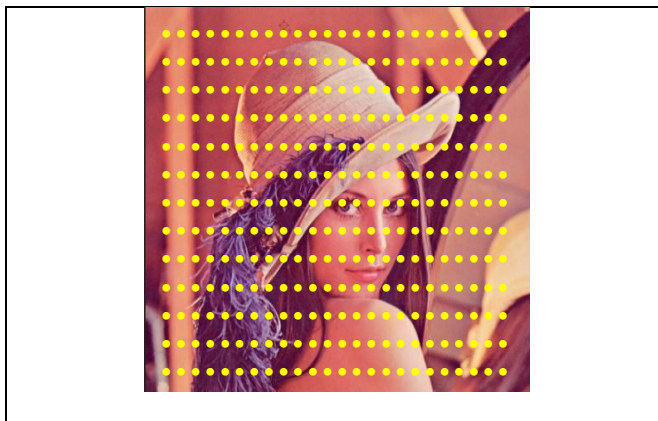


3. Super piksele – algorytm SLIC superpixels (SLIC: *simple linear iterative clustering*)


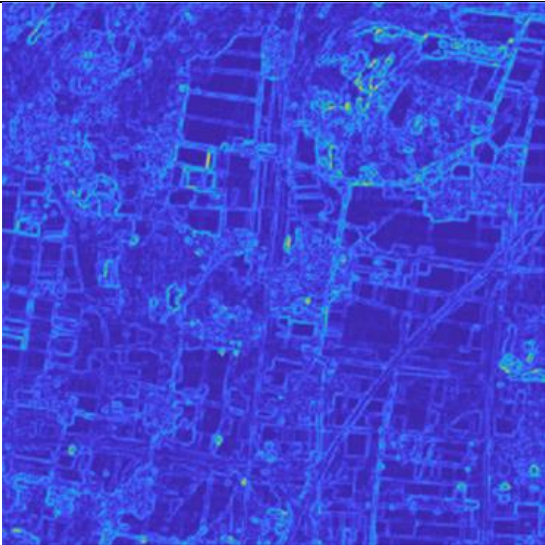
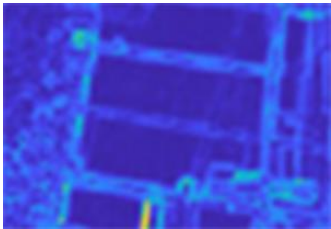
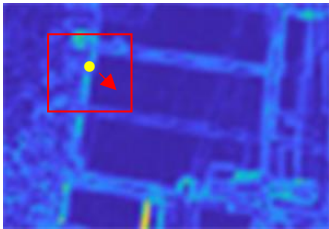
Dość bezpośrednim zastosowaniem podejścia klasteryzacji do tworzenia super pikseli, dotyczy bezpośredniego zastosowania założeń algorytmu *k-średnich* na obrazie, ale w sensie lokalnym. Twórcy algorytmu SLIC z powodzeniem zastosowali klasteryzację metodą *k-średnich* do celu generacji super pikseli, stosując poniższe ogólne założenia:

- Użytkownik wprowadza liczbę klastrów, względem których dokonuje rozbiecie zbioru pikseli obrazu,
- Centra klastrów układane są w postaci odpowiedniej siatki na obrazie – idea polega na działaniu lokalnym (zob. rys. 2) oraz konkretnym umiejscowieniu centrów klastrów na obrazie,
- Zakłada się odpowiednią wielkość klastra – tj. klaster nie jest wyłącznie zbiorem danych, jako że twórcy wykorzystują informacje związane z lokalizacją pikseli (współrzędne x i y), klaster staje się de facto odpowiednikiem super piksela,
- Informacje dotyczące pikseli – wektor cech, to: kolory + współrzędne, twórcy korzystają tu z modelu barw CIE $L^*a^*b^*$ ze względu na mniejszą wrażliwość na małe wartości ‘odległości pomiędzy kolorami’. A więc wektor cech dla dowolnego piksela, to: (L, a, b, x, y) ,

- Unika się lokalizacji centrów klastrów na krawędziach, poprzez odpowiednią analizę wartości gradientu w danym punkcie obrazu (zob. rys. 3).



Rys. 2. Przykładowa inicjalizacja algorytmu SLIC

	
Przykładowy obraz ¹	Ilustracja wartości gradientu za pomocą obrazu typu 'mapa ciepła'.
	
Powiększony podobszar obrazu	Jeśli punkt środkowy klastra wpada w dużą wartość gradientu – czyli potencjalnie reprezentuje krawędź, to należy wyszukać w jego otoczeniu wartość min jako nową wartość środka klastra.

¹ Źródło: D. Liang, J. Ding and Y. Zhang, "Efficient Multisource Remote Sensing Image Matching Using Dominant Orientation of Gradient," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 2194-2205, 2021, doi: 10.1109/JSTARS.2021.3052472.

Pozostałe założenia algorytmu:

- Względne pole powierzchni super piksela (przy inicjalizacji):
$$\frac{N}{K},$$

gdzie N – liczba pikseli obrazu, K – liczba klastrów,

- Odległość pomiędzy centrami klastrów:

$$S =_{\text{df}} \sqrt{\frac{N}{K}},$$

- Odległość pomiędzy dowolnym pikselem, a określonym centrum klastra:

$$D_S =_{\text{df}} d_{CIE\ L*a*b*} + \frac{m}{S} d_{xy},$$

gdzie m jest parametrem określającym stopień uwzględnienia odległości między pikselami (w sensie wartości współrzędnych).

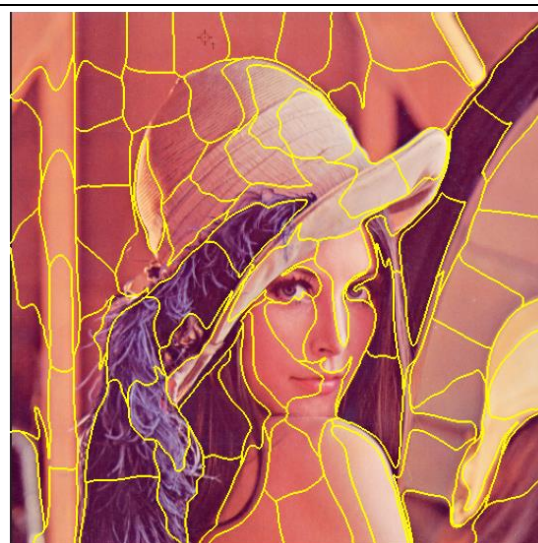
Przy uwzględnieniu powyższych założeń, poniżej przedstawiono oryginalny algorytm [1]:

Algorithm: Efficient superpixel segmentation

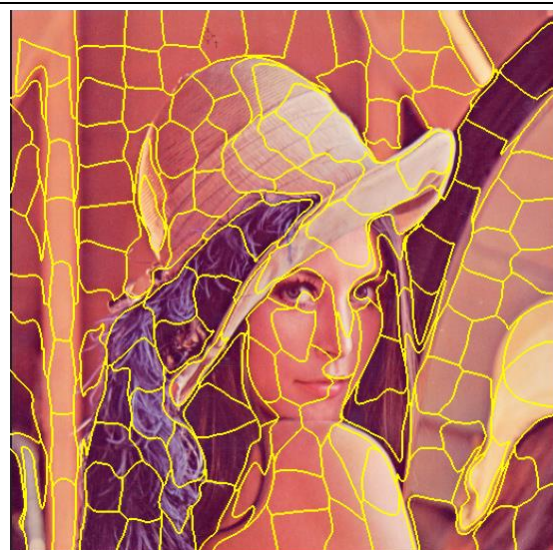
1. Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S .
2. Perturb cluster centers in an $n \times n$ neighborhood, to the lowest gradient position.
3. Repeat
4. for each cluster center C_k do
5. Assign the best matching pixels from a $2S \times 2S$ square neighborhood around the cluster center according to the distance measure (D_S).
6. end for
7. Compute new cluster centers and residual error E (L_1 distance² between previous centers and recomputed centers)
8. until $E \leq \text{threshold}$
9. Enforce connectivity.

² Metryka Manhattan

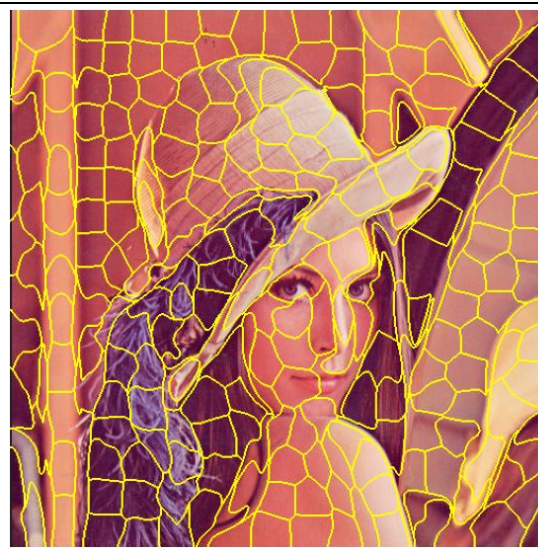
Oczywiście, w zależności od głównego parametru klasteryzacji, czyli wartości k , uzyskamy różne wyniki – poniżej przykłady:



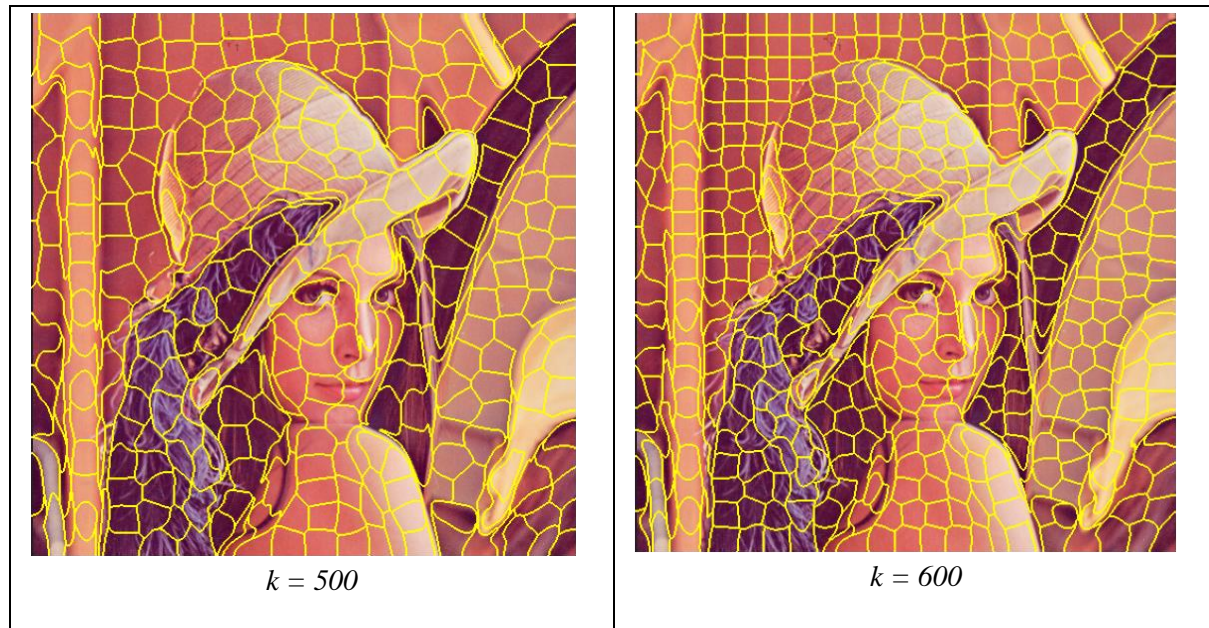
$k = 100$



$k = 200$



$k = 300$



Literatura:

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Ssstrunk. *Slic superpixels*. Technical report, 2010.