

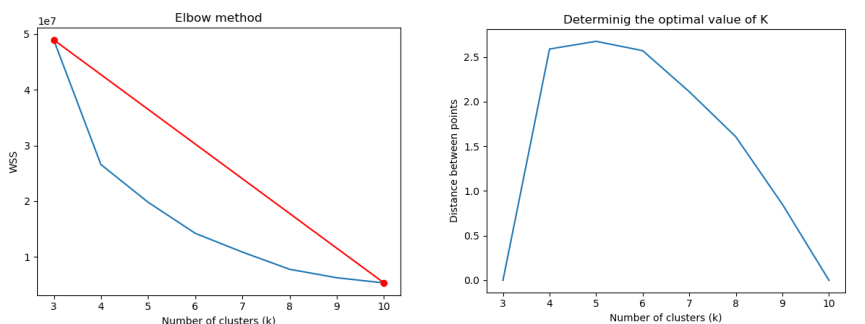
## Raport z projektu - zespół 8

### Problem:

W zadaniu należało automatycznie zlokalizować i zaznaczyć zmiany w obszarze nerek widocznych na zdjęciach z rezonansu magnetycznego.

### Wybrany algorytm:

Do segmentacji obrazu wykorzystano segmentację obrazu z klastrowaniem K-means. Do implementacji skorzystano z gotowej biblioteki OpenCV i dobrano parametry, aby otrzymać najlepsze wyniki. Początkowy dobór liczby K został ulepszony - wykorzystano zoptymalizowaną do zakresu (3,10) metodę Elbow. Metoda ta polega na obliczeniu sumy kwadratów odległości od punktów w klastrach do najbliższego im centroidu, dla wszystkich wartości K w zbiorze, a następnie wyznaczenie optymalnej wartości K jako tej dla której wykres staje się liniowy. Zautomatyzowano ten proces poprzez połączenie pierwszej i ostatniej wartości wykresu i obliczenie odległości między dwoma liniami. Wartość K, dla której odległość ta będzie największa jest optymalną liczbą klastrów.



Rys. 1. Wykresy konieczne w doborze optymalnego K

Segmentację przeprowadzono na konkretnym obrazie wybranym z bazy **2019 Kidney Tumor Segmentation Challenge** [1].

Poprawność opisanej wyżej metody sprawdzona została na obrazach udostępnianych z zewnętrznego źródła udostępnianym przez **Grand Challenges in biomedical images analysis** [2]. W tym celu pobrano dostępne na stronie repozytorium z danymi od anonimowych pacjentów zapisane w obrazach CT dla 300 przypadków. 210 przypadków było zbiorem treningowym z wyznaczonym już wskazaniem raka natomiast pozostałe 90 przypadków zbiorem testowym.

### Implementacja:

```
def Kmeans(image,k):
    pixel_values = image.reshape((-1, 3))
    pixel_values = np.float32(pixel_values)
    print(pixel_values.shape)
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.2)
    _, labels, (centers) = cv2.kmeans(pixel_values, k, None, criteria, 10, cv2.KMEANS_PP_CENTERS)

    centers = np.uint8(centers)

    labels = labels.flatten()
    segmented_image = centers[labels.flatten()]

    segmented_image = segmented_image.reshape(image.shape)
    return segmented_image, centers, labels
```

Rys. 2. Implementacja funkcji K-means

```
def elbow(image, kmax):
    sse = []
    points = image.reshape((-1, 3))
    points = np.float32(points)
    for k in range(3, kmax+1):
        curr_sse = 0
        [segmented_image, centroids, pred_clusters] = Kmeans(points, k)

        for i in range(len(points)):
            curr_center = centroids[pred_clusters[i]]
            curr_sse += (points[i, 0] - curr_center[0]) ** 2 + (points[i, 1] - curr_center[1]) ** 2

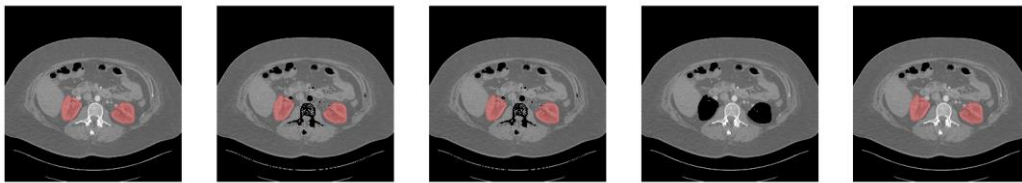
        sse.append(curr_sse)
    return sse
```

Rys. 3. Implementacja funkcji Elbow

```
def segmentation(image, OptimalK, cluster):
    [segmented_image, c, labels] = Kmeans(image, OptimalK)
    masked_image = np.copy(image)
    masked_image = masked_image.reshape((-1, 3))
    cluster = cluster
    masked_image[labels == cluster] = [0, 0, 0]
    masked_image = masked_image.reshape(image.shape)
    return segmented_image, masked_image
```

**Rys. 4.** Implementacja umożliwiająca wyświetlania poszczególnych klastrów

## Wyniki:



**Rys. 5.** Obrazy z zaznaczonymi różnymi klastrami (czarne obszary to uwidocznione klastry)



**Rys. 6.** Badany obraz, obraz z zaznaczonymi nerkami (ze zbioru) oraz obraz z zaznaczonymi nerkami przez nasz algorytm

## Ewaluacja:

Ewaluacja została przeprowadzona według algorytmu zaczerpniętego z repozytorium challeng'u [3]. Algorytm ten sprawdza ile pikseli reprezentujących nerki w obrazie testowym udało się pozytywnie zidentyfikować. Otrzymany wynik jest podany jako procent poprawnie zidentyfikowanych pikseli do wszystkich reprezentujących nerki.

## Podsumowanie i wnioski:

Na osiągniętym etapie implementacji rozwiązania udało się uzyskać segmentację nerek bez zaznaczonej zmiany nowotworowej. Jest to pierwszy krok w kierunku zaznaczania obszarów rakowych na obrazie MRI.

Poprzez wizualną weryfikację stwierdzono, że obszar nerek został zaznaczony poprawnie, jednakże ze względu na losowe umieszczanie centroidów nie zawsze są zaznaczane te same obszary, co jest dużą wadą ze względu na to, że wyniki dla tego samego obrazu mogą być różne.

W dalszych planach rozwoju algorytmu planowana jest próba poprawienia działania algorytmu oraz osiągnięcie założonego celu jakim było zaznaczanie guzów.

**Bibliografia:**

- [1] <https://kits19.grand-challenge.org/data/>
- [2] <https://grand-challenge.org/>
- [3] [https://github.com/neheller/kits19/blob/master/starter\\_code/evaluation.py/](https://github.com/neheller/kits19/blob/master/starter_code/evaluation.py/)
- [4] Cárdenes, R., de Luis-García, R., & Bach-Cuadra, M. *A multidimensional segmentation evaluation for medical image data. Computer Methods and Programs in Biomedicine*
- [5] Siddheswar Ray and Rose H. Turi, *Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation*
- [6] David Alvarez L. and Monica Iglesias M. *k-Means Clustering and Ensemble of Regressions: An Algorithm for the ISIC 2017 Skin Lesion Segmentation Challenge*
- [7] Nameirakpam Dhanachandra\*, Khumanthem Manglem and Yambem Jina Chanu, *Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm*