Akademia Górniczo – Hutnicza im. S. Staszica w Krakowie

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Automatyczna segmentacja nerek i nowotworów nerek w obrazach tomografii komputerowej - analiza problemu i proponowane rozwiązanie

PROJEKT REALIZOWANY W RAMACH PRZEDMIOTU

TECHNIKI OBRAZOWANIA MEDYCZNEGO

Karolina Makuch, Jagoda Łebska, Klaudia Młynarczyk, Aleksandra Pieszka

Kraków, 22.06.2020 r.

Spis treści

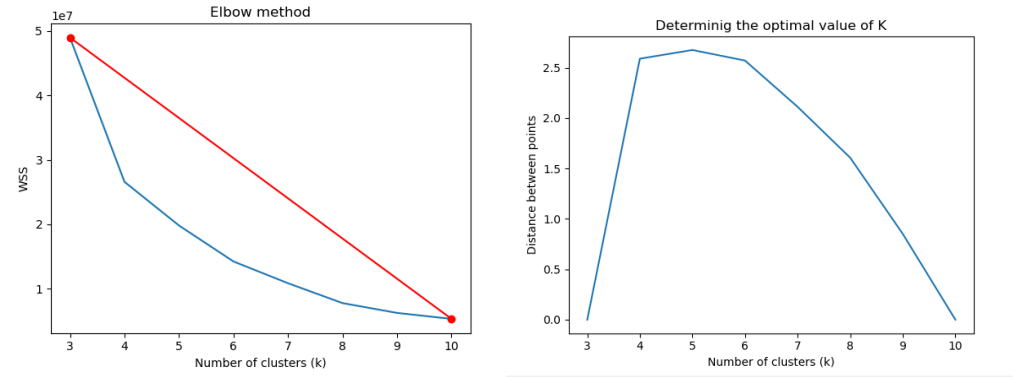
1. Cel projektu

Celem projektu było rozwiązanie problemu dotyczącego automatycznej segmentacji nerek i nowotworów nerek w obrazach tomografii komputerowej. Jego realizacja przebiegała w kolejnych krokach: zapoznanie z literaturą związaną z zagadnieniami automatycznej segmentacji w analizie obrazów medycznych, wybór algorytmu, implementacja algorytmu, dobór metody ewaluacji na podstawie literatury i informacji umieszczonych na stronie opisującej szczegóły zadania oraz ewaluacja wyników. Na podstawie uzyskanych wyników oceniono stopień poprawności dobranej metody, a także określono problemy stawiane przez algorytm oraz możliwości rozwoju projektu.

2. Wykorzystane metody

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

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

Ewaluacja:

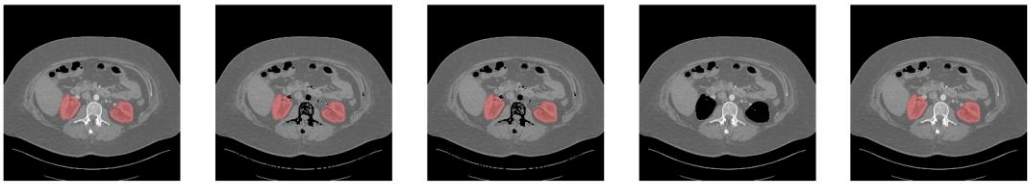
Ewaluacja została przeprowadzona według algorytmu zaczerpniętego z repozytorium challengu [3]. Algorytm ten sprawdza ile pikseli reprezentujących nerki w obrazie testowym udało się pozytywnie zidentyfikować. Otrzymany wynik jest podany jako procent poprawnie zidentyfikowanych pikseli do wszystkich reprezentujących nerki. Jako miarę ilościową ewaluacji dla przypadków testowych użyto średniej oraz odchylenia standardowego. Średnią wyliczono w pierwszej kolejności

dla wszystkich zdjęć testowych, a następnie tylko dla obrazów, na których widoczne są nerki, ponieważ algorytm w przypadku zdjęć bez nerek przyjmował wynik ewaluacji jako 0, co zaniżało średnią.

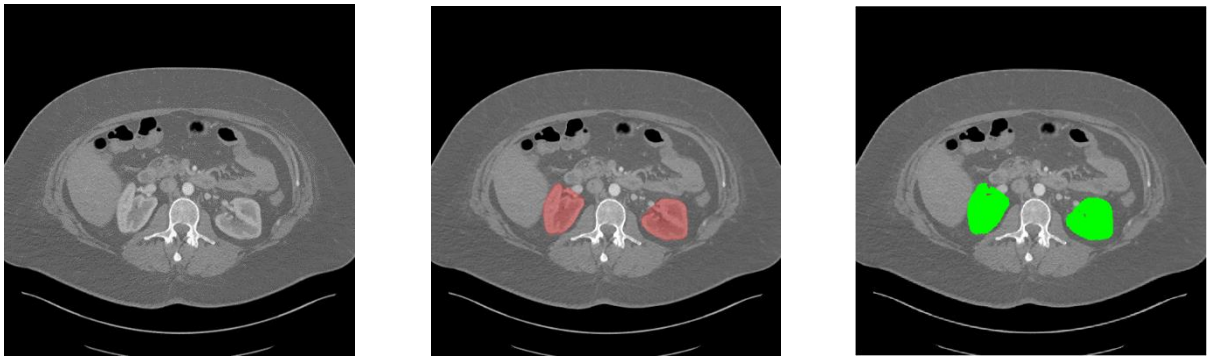
3. Prezentacja wyników

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

Wyniki:



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



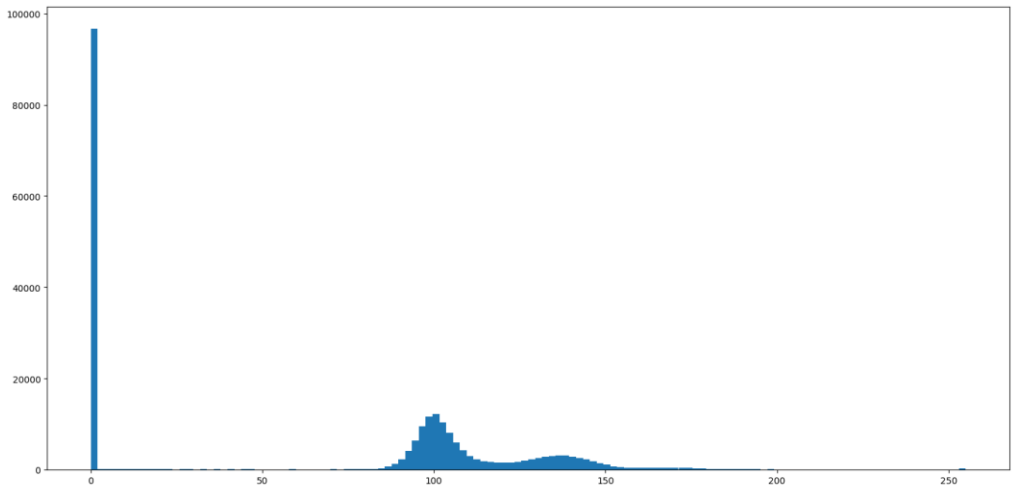
Rys. 6. Badany obraz, obraz z zaznaczonymi nerkami (ze zbioru) oraz obraz z zaznaczonymi nerkam poprzez algorytm (najlepszy przypadek)

Wyniki takiej odmiany algorytmu nie były powtarzalne ze względu na losowy charakter umieszczania centroidów. Tylko pojedyncze wyniki dawały satysfakcjonujące oznaczenie. Średnia i odchylenie standardowe wyników ewaluacji dla całego badanego zbioru zostały umieszczone w tabeli 1.

Tab. 1. Średnia i odchylenie standardowe dla aktualnego etapu implementacji

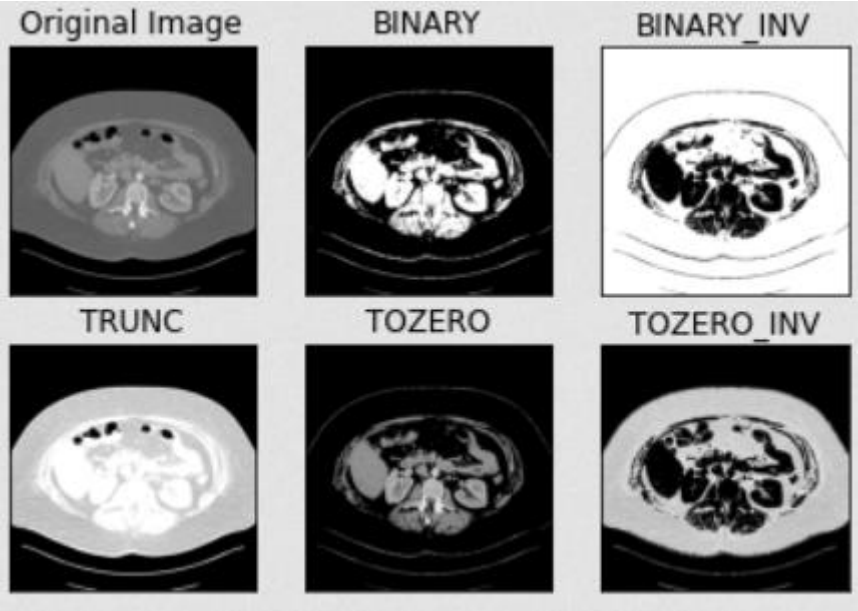
Parametr	**Wartość**
Średnia [%]	1,61
Odchylenie standardowe	0,01

Ze względu na brak powtarzalności zaznaczania odpowiednich klastrów kontynuowano implementację kodu. Stwierdzono, że należy ograniczyć widoczność struktur, różnych od nerek i ich zmian nowotworowych. Strategię realizowano poprzez wyświetlenie histogramu dla pewnego obrazu i analizie, które wartości będą odpowiadały za nerki.



Rys. 7. Histogram dla pewnego obrazu z bazy

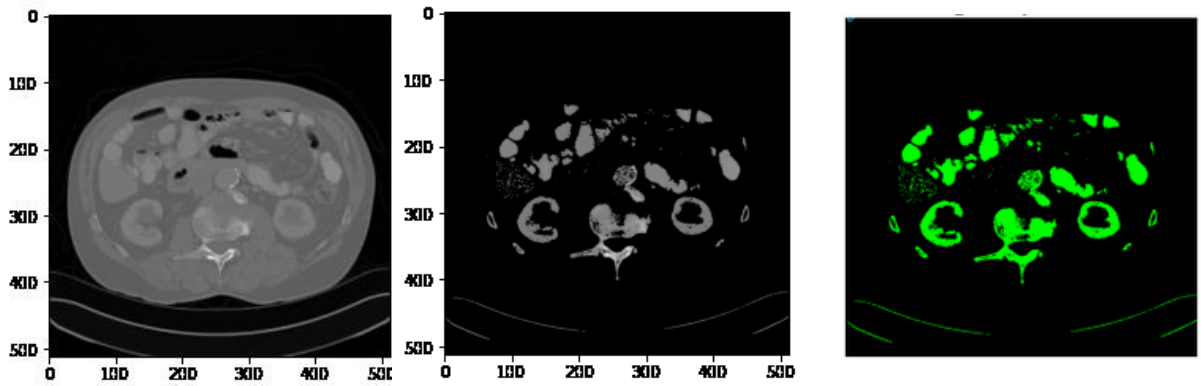
Stwierdzono, że wartość 0 to tło, a obraz nerek i ich zmian znajduje się w przedziale od 120 do 255. Do okrojenia obrazu wykorzystano wbudowaną funkcję do thresholding'u z biblioteki OpenCV. Empirycznie sprawdzono, który rodzaj progowania będzie najlepszy do preprocessing'u obrazów CT.



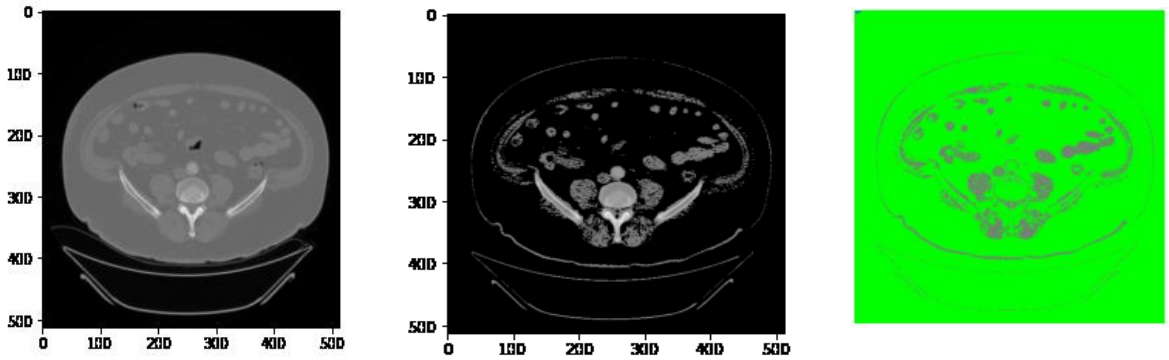
Rys. 8. Różne rodzaje threshold'u dostępne w bibliotece *OpenCV*

Stwierdzono, że na potrzeby kodu najlepszy będzie algorytm *TOZERO,* ponieważ nie zwraca obrazu w postaci binarnej, a także łączy niepotrzebne struktury z tłem.

Algorytm wykonano dla losowych zdjęć z bazy. Na rysunku 9 przedstawiono wyniki działania algorytmu dla najlepiej oznaczonego obrazu, a na rysunku 10 najgorzej oznaczoną strukturę.



Rys. 9. Najlepiej oznaczony obraz CT
(procent poprawnie zaznaczonych pikseli: 28,4%)



Rys. 10. Najgorzej oznaczony obraz CT
(procent poprawnie zaznaczonych pikseli: 0%)

W tabeli 2 umieszczono wyniki ewaluacji dla nowej postaci kodu.

Tab. 2. Średnia i odchylenie standardowe dla kodu z thresholding'iem

Parametr	Wartość
Średnia [%]	4,99
Odchylenie standardowe	0,03

Ze względu na losowość wybranych zdjęć, poddane ewaluacji zostały również obrazy niezawierające poszukiwanych struktur (nerek i ich zmian nowotworowych), dla których zgodność zaznaczenia tkanek wynosiła 0%. W związku z tym obliczono także średnią oraz odchylenie standardowe dla niezerowych wyników. Wartości umieszczono w tabeli 3.

Tab. 3. Średnia i odchylenie standardowe dla niezerowych wyników dla kodu z thresholding'iem

Parametr	Wartość
Średnia [%]	16,63
Odchylenie standardowe	0,06

4. Dyskusja i podsumowanie

Na osiągniętym etapie implementacji rozwiązania udało się uzyskać segmentację nerek. Poprzez ewaluację wyników 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. Dla populacji losowych zdjęć średnia wartość poprawnie zaznaczonych pikseli wyniosła jedynie 1,61 %.

Z racji na tak niskie wartości kontynuowano implementację kodu, który uzupełniono o algorytm thresholdingu. Dla takiej formy uzyskano już wyższe wartości - rzędu 16,63 %. Jest to znaczący krok w kierunku poprawnego oznaczania struktur nerkowych.

Jednakże kod wciąż posiada pewne mankamenty. Jednym z nich jest fakt, że algorytm zaznacza klastry na wszystkich obrazach, nawet na tych, na których nerki nie są widoczne. Problemem jest także losowość rozmieszczenia centroidów, co może powodować, że nerki w ogóle nie znajdą się w żadnym klastrze.

Nieznajomość określenia, do którego klastra zostaną przyporządkowane nerki także stanowi mankament zastosowanego rozwiązania. Dwa z powyższych problemów dałoby się rozwiązać poprzez wyznaczanie lokalnych histogramów dla poszczególnych elementów obrazu, gdzie wartości intensywności są mniej więcej zgodne z tymi oczekiwanymi dla nerki. Dzięki temu możliwe będzie określenie obszarów optymalnych dla centroidów. Podjęto próbę implementacji takiego rozwiązania, lecz nie zakończyła się ona sukcesem.

W ramach rozwoju kodu przemyślano dalsze ulepszenia możliwe do implementacji. Jednym z nich byłoby automatyczne rozpoznawanie obrazów, na których widnieje struktura nerek. Można byłoby to rozwiązać poprzez wnioskowanie z wyników ewaluacji (np. wyniki o zgodności większej niż 50% umieszczano by w osobnej grupie).

5. 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?fbclid=IwAR3t5Lxht5QDPSjGD0_rRAHYayJu40aD4UYPg5LO1OGRe9l8pDxrnvmOxqw
[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*