WYKRYWANIE NACZYŃ DNA OKA

Mateusz Ksok, 136751
Marcin Jachymski, 136718

1) Zastosowany język programowania i biblioteki

Do wykonania projektu zastosowany został język Python wraz z następującymi bibliotekami:

- scikit-image
- sklearn
- matplotlib
- cv2
- numpy
- pandas

2) Opis zastosowanych metod

a. Metoda oparta o przetwarzanie obrazu:

Preprocessing:

1) Transfer zdjęcia do odcieni szarości
2) Zwiększenie kontrastu na zdjęciu – wybranie wartości percentyli z niezerowych elementów tablicy i dopasowanie wartości na ich podstawie

Wykrywanie krawędzi:

1) Zastosowanie filtru Sobela do wykrycia krawędzi
2) Poszerzenie kontrastu na wyjściowym zdjęciu

Postprocessing:

1) Transfer wyniku przetwarzania do maski binarnej
2) Wykorzystanie erozji i dylatacji w celu usunięcia szumów z maski

Uzasadnienie:

Wykorzystanie zwiększenia kontrastu w preprocessingu i tuż po wygenerowaniu maski ma na celu zwiększenie różnic pomiędzy konkretnymi punktami na zdjęciu – uwydatnienie kolorów sprawi, że samo zdjęcie będzie o wiele prostsze do przetwarzania na późniejszych etapach. Wybrany filtr Sobela jest wykorzystywany do wykrywania krawędzi na zdjęciach, więc do wyróżniających się na tle reszty obrazu naczyń krwionośnych można go z powodzeniem użyć. Ostatnim etapem jest wykorzystanie erozji i dylatacji (w tej kolejności) aby usunąć nadmiarowe szumy ze zdjęcia.

b. Metoda oparta o sztuczną inteligencję:

1) Zdjęcia wykorzystywane do uczenia modelu w trakcie przetwarzania wstępnego poddawane są:
   - konwersji na skalę szarości,
   - normalizacji (sprowadzeniu intensywności pikseli do przedziału (0;1),
   - zwiększeniu kontrastu,

- operacji wykrywania krawędzi przy pomocy filtru Sobela.

2) Następnie obrazy dzielone są na kwadraty o boku `slice_size`, pobierane z odstępem `slice_step` (fragmenty mogą się nakładać). Na podstawie wycinków tworzone są wektory próbek, które składają się z wartości składowych pikseli oraz momentów Hu obliczonych dla danego fragmentu.
Zdjęcia poddawane klasyfikacji jak i te tworzące zbiór treningowy poddawane są identycznemu przetwarzaniu.
W przypadku zbioru uczącego próbki następnie etykietowane są przez pobranie środkowych pikseli z analogiczne podzielonych masek eksperckich. Tak utworzona ramka danych poddawana jest downsamplingowi w celu wyrównania liczności klas.

3) Do klasyfikacji wykorzystaliśmy las losowy w implementacji z biblioteki scikit-learn. Zmiana parametrów klasyfikatora (n_estimators, min_samples_leaf, max_depth) nie prowadziła do istotnej poprawy, zatem pozostawiliśmy argumenty domyślne.

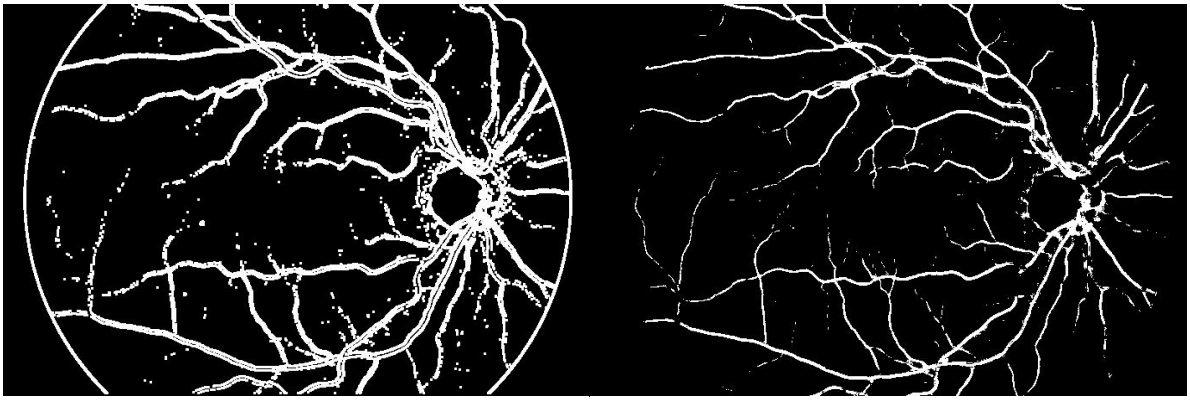
4) Klasyfikator oceniliśmy za pomocą 5-fold cross validation, otrzymując wyniki accuracy ok. 85% Inne metryki (sensitivity, specificity) obliczyliśmy przez analizę macierzy pomyłek dla klasyfikacji obrazów spoza zbioru uczącego.

5) Zdecydowaliśmy się na wykorzystanie algorytmu lasu losowego ze względu na jego wydajność (np. w porównaniu do perceptronu wielowarstwowego) oraz łatwość w zastosowaniu. Wydaje się jednak, że to nie algorytm klasyfikacji, lecz wstępne przygotowanie danych miało najistotniejszy wpływ na jakość przewidywania i stanowiłoby najlepsze pole do dalszej optymalizacji. Wyzwaniem w tym problemie okazało się zwłaszcza niezrównoważenie klas decyzyjnych, powodujące niską czułość klasyfikatora, którą jednak udało się poprawić dzięki resamplingowi zbioru uczącego.

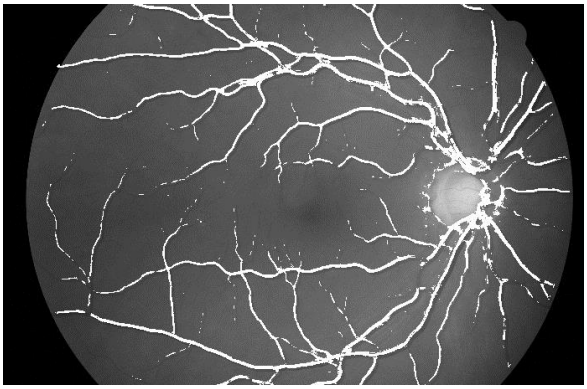
3) Wizualizacja

Zdjęcie 1 wraz z maską ekspercką – 07_h.jpg:

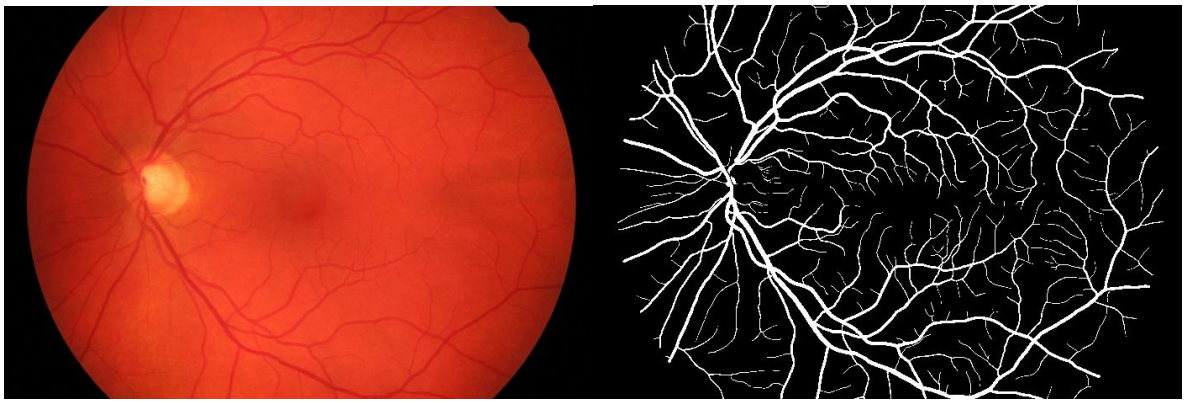
Wynikowe maski: przetwarzanie proste (Sobel) i za pomocą AI (las):



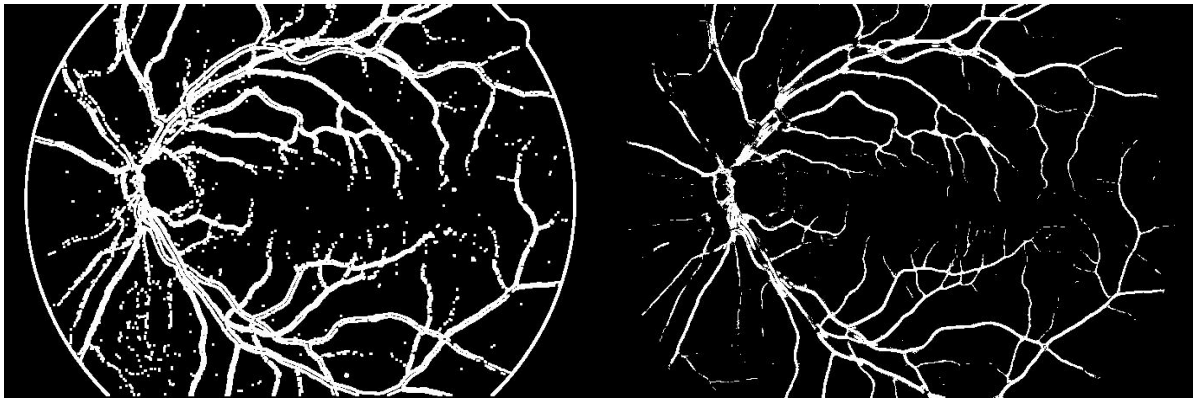
Wyjściowe zdjęcie (w skali szarości, z nałożoną maską AI)



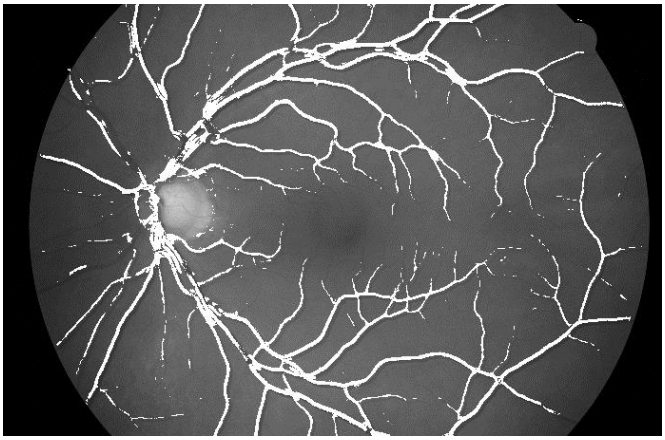
Zdjęcie nr. 2 – 08_h.jpg



Maski: prosta i zaawansowana



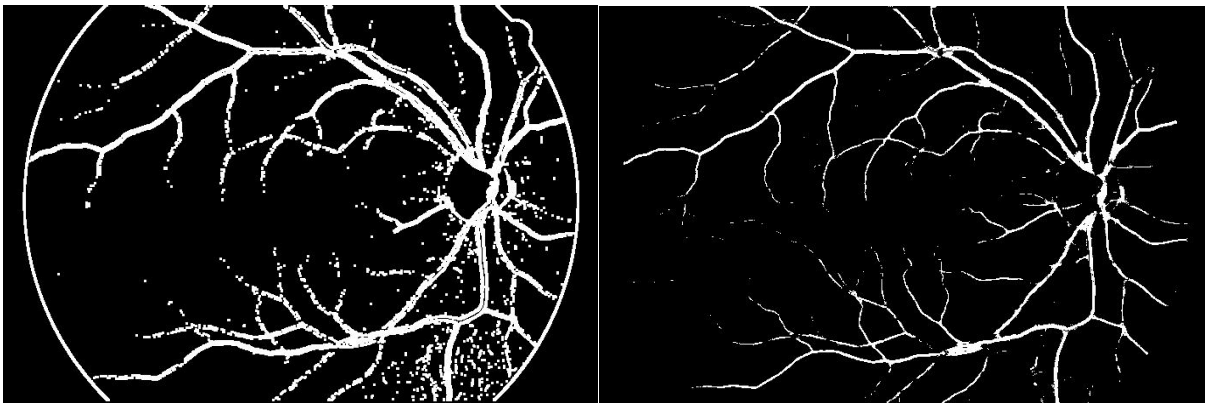
Wyjściowe zdjęcie z nałożoną maską:



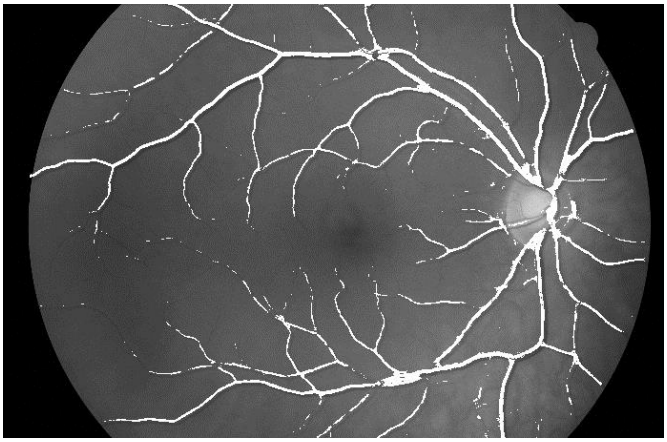
Zdjęcie 3: 09_h.jpg

Zdjęcie wejściowe wraz z maską ekspercką

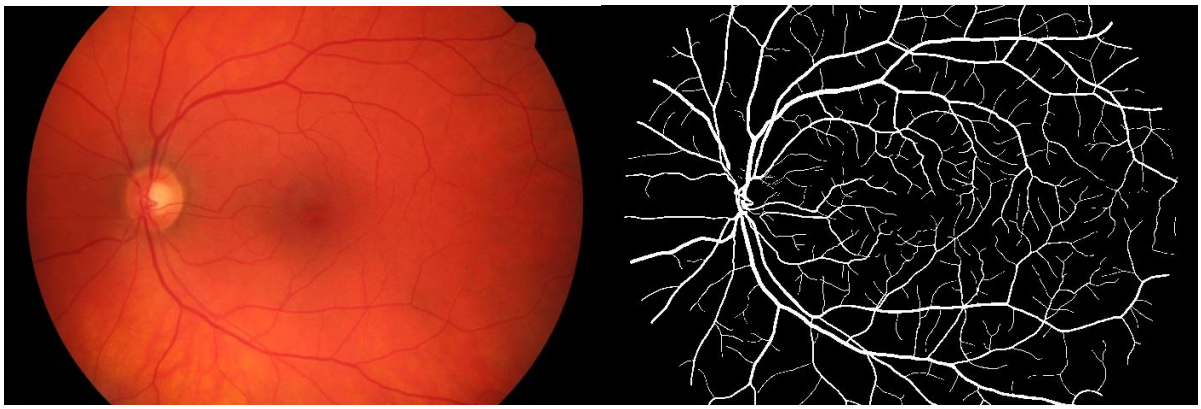
Maski:



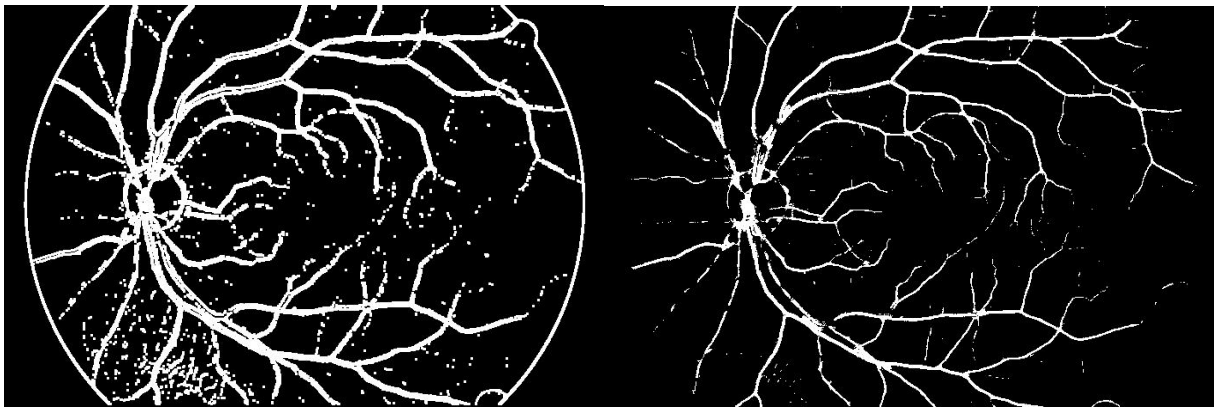
Zdjęcie wyjściowe:



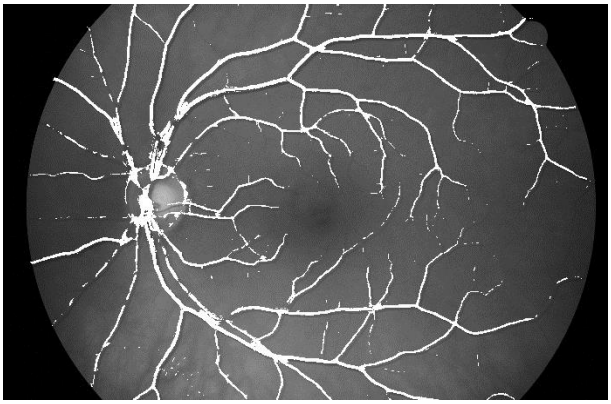
Zdjęcie 4: 10_h.jpg



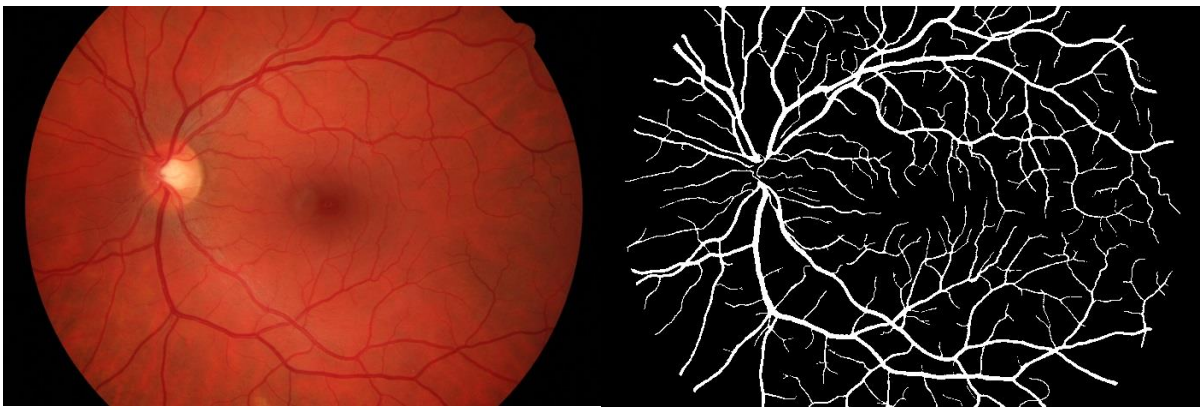
Maski:



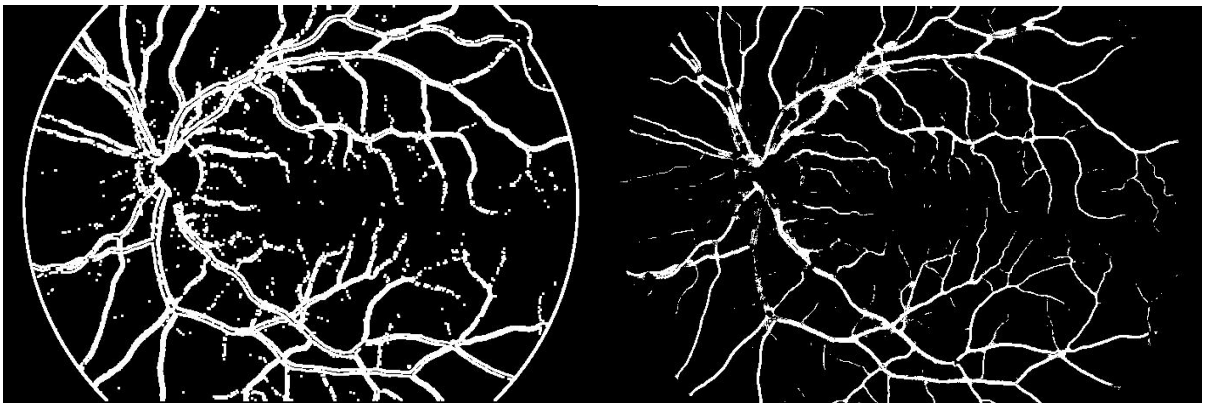
Wyjście:



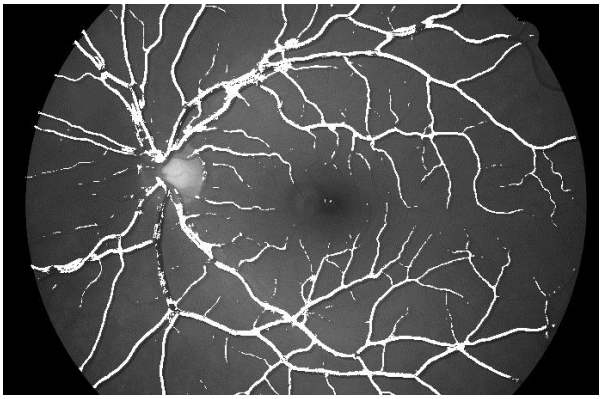
Zdjęcie 5: 12_h.png



Maski:



Wyjście:



4) Analiza statystyczna obrazów

Obraz 1:

Przetwarzanie obrazów:

Macierz pomyłek:

18421	30939
10033	255922

Specificity: 0.3731969205834684
Sensitivity: 0.962275572935271
Accuracy: 0.8700600986315272

Sztuczna inteligencja:

Macierz pomyłek:

20571	34521
3310	256913

Specificity: 0.3097990311285913
Sensitivity: 0.9253054682034598
Accuracy: 0.8846074560360274

Obraz 2:

Przetwarzanie obrazów:

Macierz pomyłek:

20571	34521
3310	256913

Specificity: 0.37339359616641254
Sensitivity: 0.9872801404948832
Accuracy: 0.8800215657358514

Sztuczna inteligencja:

Macierz pomyłek:

7774	24305
16107	267129

Specificity: 0.24233922503818697
Sensitivity: 0.9431322289539465

Accuracy: 0.871836100407529

Obraz 3:

Przetwarzanie obrazów:

Macierz pomyłek:

13192	27145
2584	272394

Specificity: 0.3270446488335771
Sensitivity: 0.9906028845944039
Accuracy: 0.9057165057165057

Sztuczna inteligencja:

Macierz pomyłek:

4918	20363
10858	279176

Specificity: 0.19453344408844586
Sensitivity: 0.9625630098540171
Accuracy: 0.9009847295561582

Obraz 4:

Przetwarzanie obrazów:

Macierz pomyłek:

17908	32899
2233	262275

Specificity: 0.35247111618477767
Sensitivity: 0.991557911291908
Accuracy: 0.8885812600098314

Sztuczna inteligencja:

Macierz pomyłek:

6045	21596
14096	273578

Specificity: 0.2186968633551608
Sensitivity: 0.951000090380083

Accuracy: 0.8868052582338296

Obraz 5:

Przetwarzanie obrazów:

Macierz pomyłek:

23003	33044
10746	248522

Specificity: 0.410423394650918
Sensitivity: 0.9585525402286437
Accuracy: 0.8611230039801469

Sztuczna inteligencja:

Macierz pomyłek:

8910	17033
24839	264533

Specificity: 0.34344524534556525
Sensitivity: 0.9141623930442475
Accuracy: 0.8672058100629529

Wnioski:

Pierwszą rzucającą się w oczy różnicą pomiędzy wynikami prezentowanych algorytmów jest podejście do klasyfikacji pikseli – przez zastosowanie erozji i dylatacji algorytm oparty o filtr Sobela nabrał skłonności do klasyfikacji do klasy pozytywnej częściej - stąd wyniki zwracane przez tą metodą charakteryzują się wyższym wskaźnikiem parametru *false positive* (120-200% wyższy niż w przypadku wyników zwróconych przez AI). Średnia trafność oscyluje w okolicach 88%, co daje całkiem wysoką poprawność algorytmu, mimo prostoty zastosowanej metody maska pokrywa zdecydowaną większość naczyń obecnych na zdjęciu. Algorytm oparty na lesie losowym zwrócił podobne wyniki co do trafności (również ok. 88%), jednakże zwracane wyniki wskazują na większą „ostrożność" tej metody w klasyfikacji pikseli. Niższy wskaźnik *false positive* i większy *false negative* oznaczają, że program woli być bardziej *pewny* w swoich decyzjach aniżeli klasyfikować nieco bardziej na wyrost.

Oba algorytmy zwróciły dla wszystkich testowanych instancji wyniki o podobnej jakości (trafność z przedziału [86;91]%), różniące się jedynie *czułością* i *swoistością*. Niskie wartości swoistości nie wynikają z jakości algorytmów, raczej z dysproporcji między ilością pikseli o klasie negatywnej a pozytywnej (niewielka część pikseli negatywnych zaklasyfikowanych jako pozytywne może przewyższyć ilość pikseli pozytywnych, co znacznie wpłynie na wartość).