

Wykrywanie naczyń dna siatkówki oka

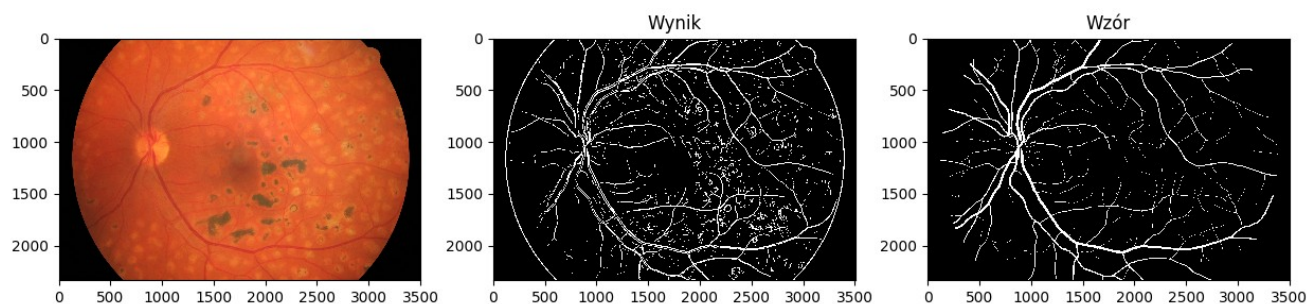
Karol Krakowski, Dominik Grzelak

1. Wersja na 3.0

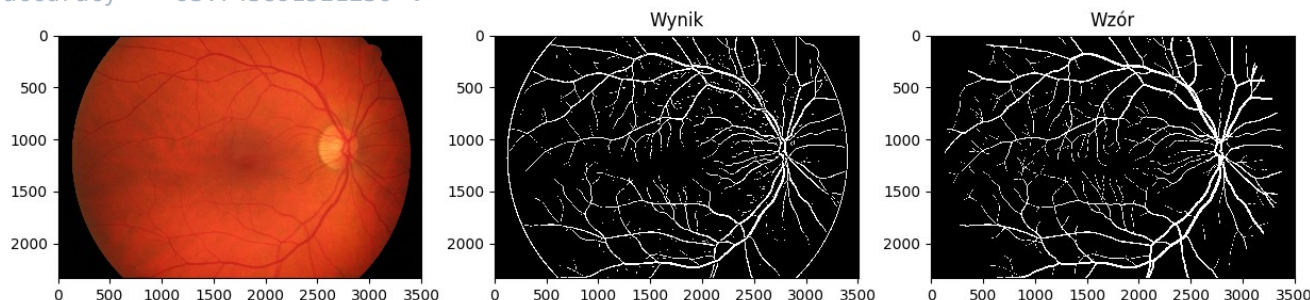
Opis programu

W najprostszej wersji programu korzystamy z technik przetwarzania obrazu w celu zlokalizowania naczyń krwionośnych. Na samym początku wczytujemy kanał zielony obrazu, ponieważ ze wszystkich trzech kanałów RGB to na nim naczynia są najbardziej kontrastowe w stosunku do tła. Tak wczytany obraz poddajemy działaniu funkcji *equalize_adapthist*, która oblicza histogramy dla obrazu i powoduje zwiększenie kontrastu, dzięki czemu szczegóły obrazu stają się bardziej widoczne. Następnie stosujemy filtr Gaussa, aby zmniejszyć zaszumienie, a po nim filtr Frangiego, który wykrywa ciągle krawędzie na obrazie. Dla tak obrobionego obrazu obliczamy wartość progową z użyciem funkcji *threshold_triangle*, której następnie używamy do wygenerowania maski binarnej na podstawie przygotowanego obrazu.

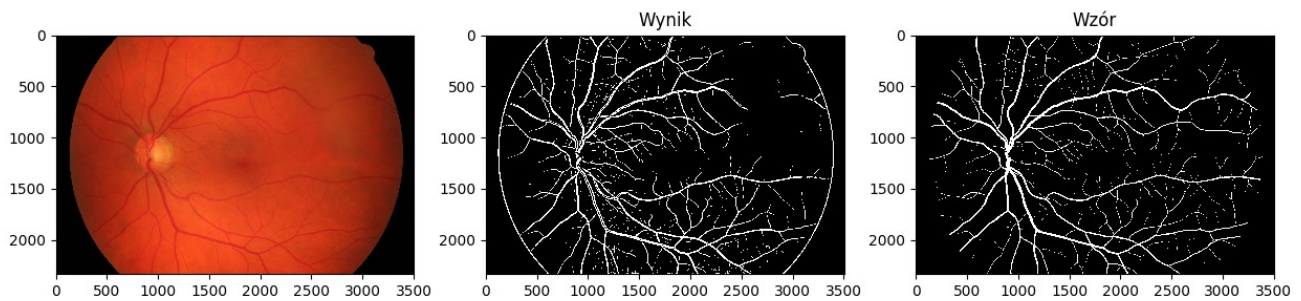
Analiza statystyczna jakości wyników



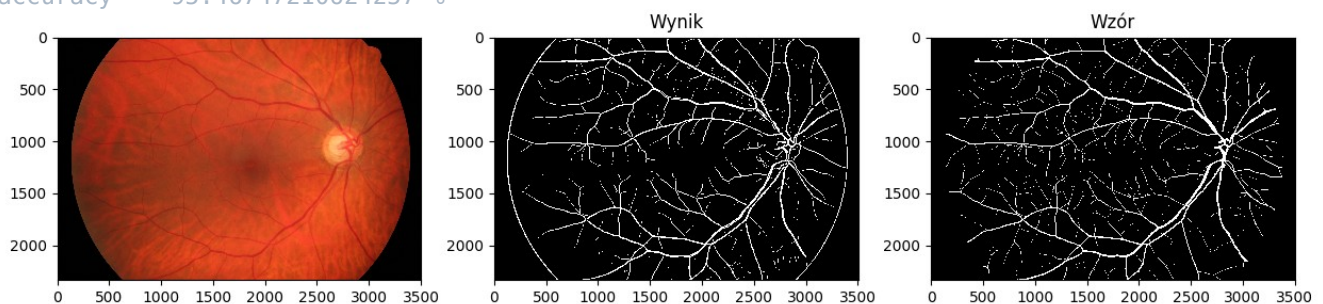
03_dr.JPG
sensitivity 48.69030690840147 %
specificity 97.79391400309765 %
accuracy 93.743891521236 %



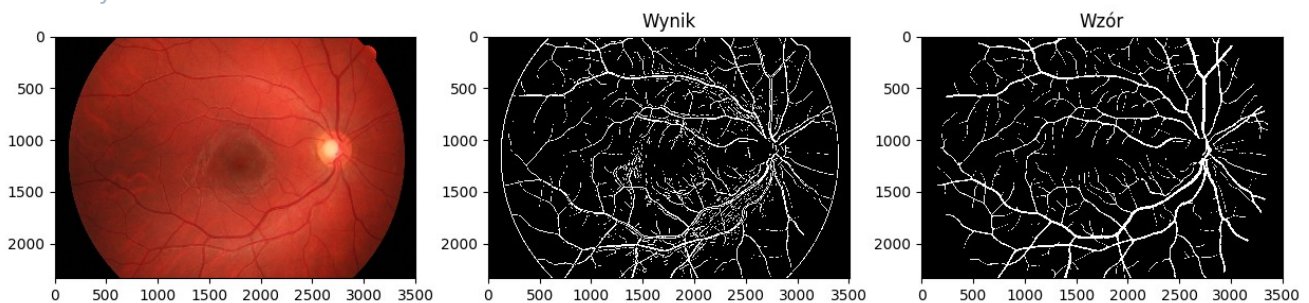
05_h.jpg
sensitivity 79.14243018746701 %
specificity 97.50363512627057 %
accuracy 95.93947426033652 %



06_g.jpg
sensitivity 64.21734624399708 %
specificity 98.22559329920894 %
accuracy 95.46747210624257 %



07_g.jpg
sensitivity 69.40401059951566 %
specificity 98.2743133240706 %
accuracy 96.23384185197348 %



13_h.jpg
sensitivity 66.8976276609655 %
specificity 97.43316584000313 %
accuracy 94.5230890723713 %

2. Wersja na 4.0

Opis programu

Program generuje losowo fragmenty obrazu rozmiaru 5x5 px, które następnie zostaną wykorzystane do wyuczenia klasyfikatora. Następnie dla każdego z fragmentów są obliczane miary statystyczne, które posłużą do obliczenia metryk euklidesowych. W dalszej kolejności tak przygotowany klasyfikator poddawany jest testom – dla każdego testowanego fragmentu metodą 3 najbliższych sąsiadów sprawdzamy, które 3 przypadki z klasyfikatora są najbliższe niego i na ich podstawie podejmowana jest decyzja o klasyfikacji.

Analiza statystyczna jakości wyników

```
03_dr.JPG
sensitivity 100.0 %
specificity 94.69469469469469 %
accuracy 94.69999999999999 %
```

```
05_h.jpg
sensitivity 28.57142857142857 %
specificity 90.5337361530715 %
accuracy 90.10000000000001 %
```

```
06_g.jpg
sensitivity 0.0 %
specificity 93.7937937937938 %
accuracy 93.7 %
```

```
07_g.jpg
sensitivity 0.0 %
specificity 94.69469469469469 %
accuracy 94.6 %
```

```
13_h.jpg
sensitivity 57.14285714285714 %
specificity 92.54783484390735 %
accuracy 92.30000000000001 %
```

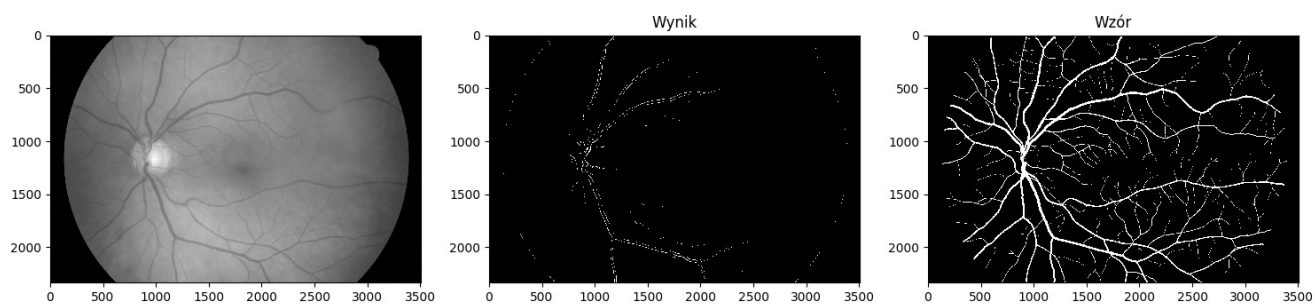
3. Wersja na 5.0

Opis programu

Algorytm oparty jest na sieci neuronowej o trzech warstwach ukrytych i funkcji aktywacji *relu* ($f(x) = \max(0, x)$) wytrenowanej w oparciu o algorytm LBFGS. Generowanie zbioru próbek odbywa się analogicznie, jak w przypadku wersji z klasyfikatorem kNN. Korzystamy także z k-krotnej walidacji skrośnej, aby podzielić zbiór próbek na 10 równych podzbiorów, z których jeden jest zbiorem testowym, a pozostałe służą do uczenia sieci.

Analiza statystyczna jakości wyników

Prezentujemy tylko 1 przypadek z uwagi na bardzo długotrwałe obliczenia (prawie 70 minut).



```
06_g.jpg
sensitivity 27.63434443031734 %
specificity 93.24558830281822 %
accuracy    92.93578779710855 %
```