

Znajdowanie naczyń krwionośnych w siatkówce

1. Skład grupy

Gustaw Kustoń 145215

2. Język programowania i biblioteki

Wybranym językiem programowania jest Python.

Do przetwarzania obrazów użyto biblioteki skimage, a do uczenia maszynowego sklearn. Do tego wykorzystano bibliotekę OpenCV do badania właściwości fragmentów obrazów oraz Matplotlib do wyświetlania obrazów.

3. Przetwarzanie obrazów

- a) Pierwszy krok to wyciągnięcie z obrazu kanału zielonego. Na nim lekko lepiej widać naczynia krwionośne.
- b) Drugi to zmiana koloru tła obrazu z czarnego na jasny szary. Czarne tło mocno zaburza działanie filtrów.
- c) Następnie nałożony zostaje filtr Sobel.
- d) Po filtrze Sobel stosuje się specyficzne progowanie, które rozjaśnia jasne miejsca obrazu, ale nie zmienia koloru tych ciemniejszych.
- e) Kolejny nakładany filtr to Frangi.
- f) Następny krok to wycięcie powstałej przez filtry okrągłej obwódki dookoła obrazu.
- g) Później stosuje się progowanie, które zamienia bardzo ciemne piksele na czarne, a szare na białe.
- h) Na koniec na obrazie wykonuje się operacje erozji i rozszerzania, co eliminuje małe białe kropki.

Taka sekwencja filtrów i progowania cieszy się najwyższą skutecznością ze wszystkich stosowanych.

4. Uczenie maszynowe

- a) Obraz podzielono na wycinki 5 x 5 pikseli, gdzie jeden wycinek odpowiada jednemu pikselowi na oryginalnym obrazie. Dla każdego wycinka obliczono momenty centralne, momenty Hu, oraz wariancję kolorów.
- b) Zastosowany algorytm to k-nearest neighbors, z parametrami n_neighbors = 1.
- c) Użyto klasyfikatory hold-out, oraz cross validation prediction (do zwizualizowania wyników).

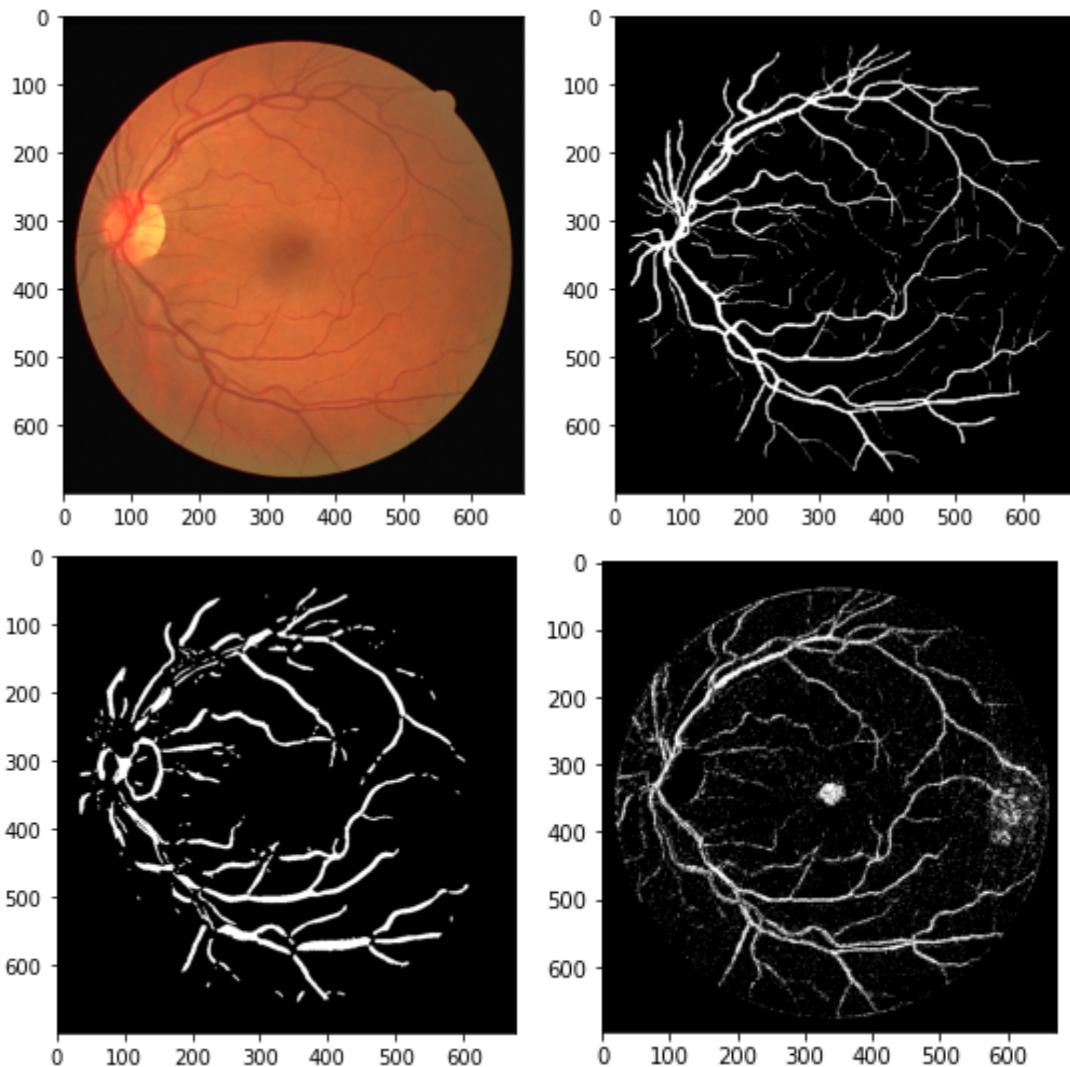
5. Wyniki

Oryginalny obraz (lewa góra),
Maska ekspercka (prawa góra),

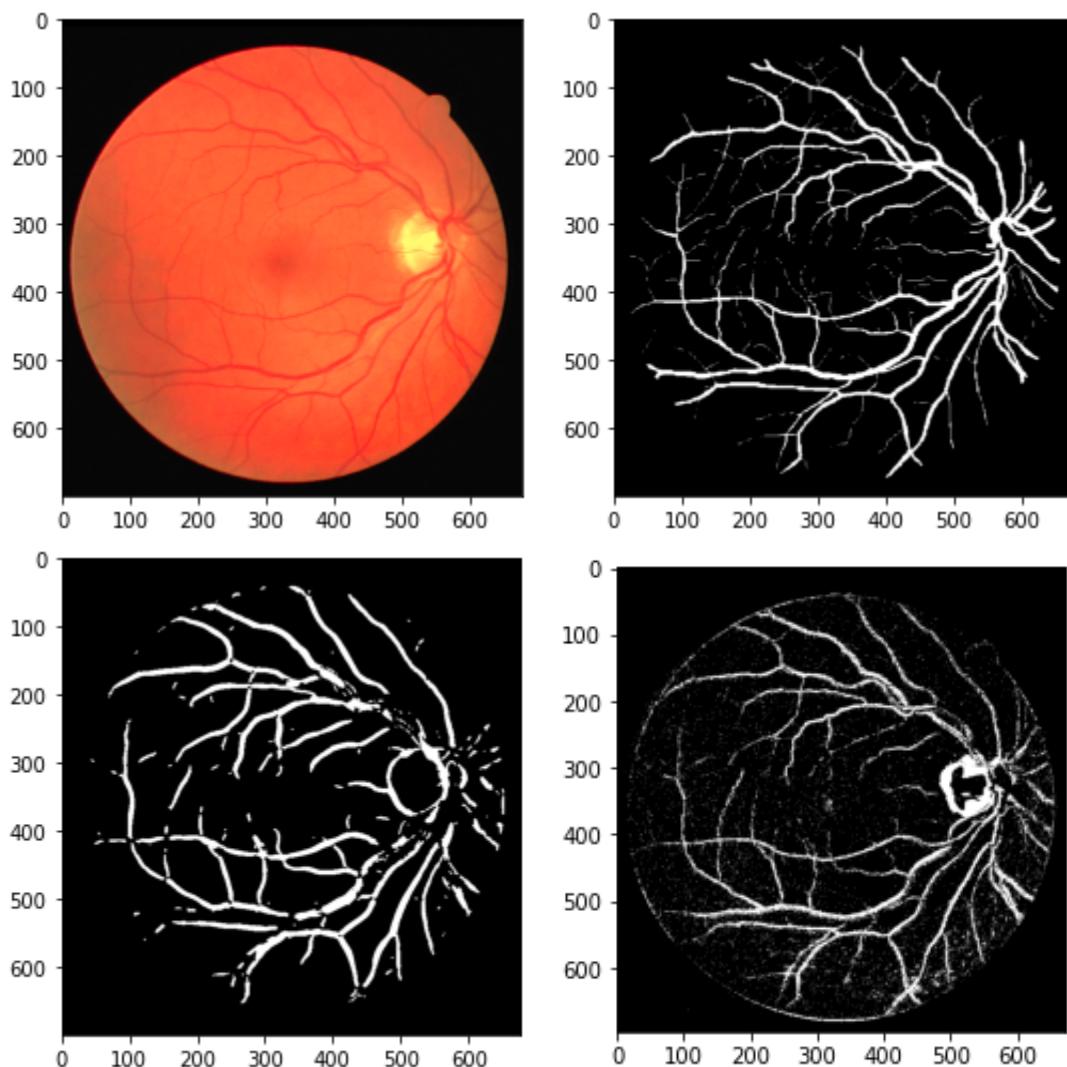
Wynik przetwarzania obrazu (lewy dół),

Wynik k-nearest neighbors wygenerowany przez Cross-validation (prawy dół)

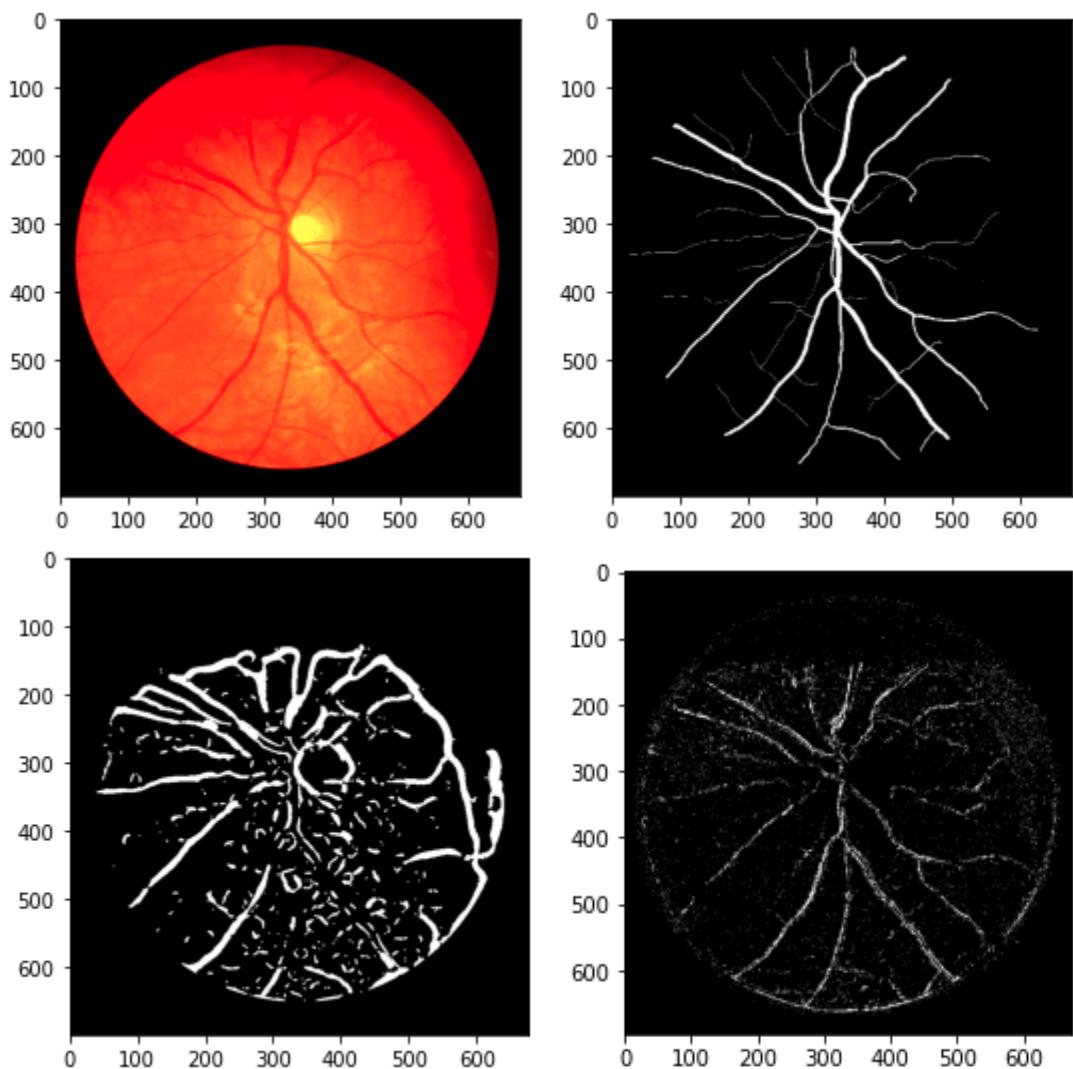
a)



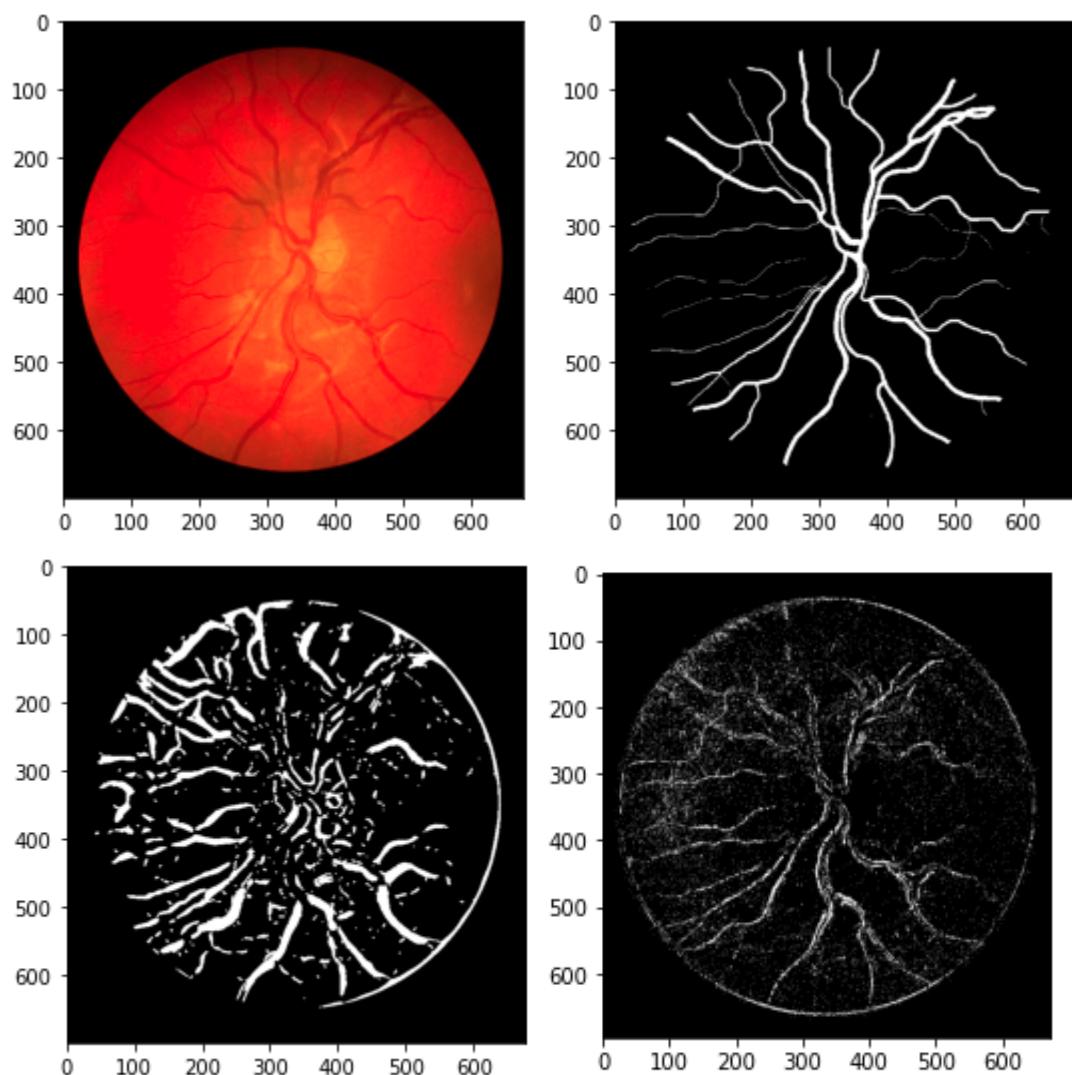
b)

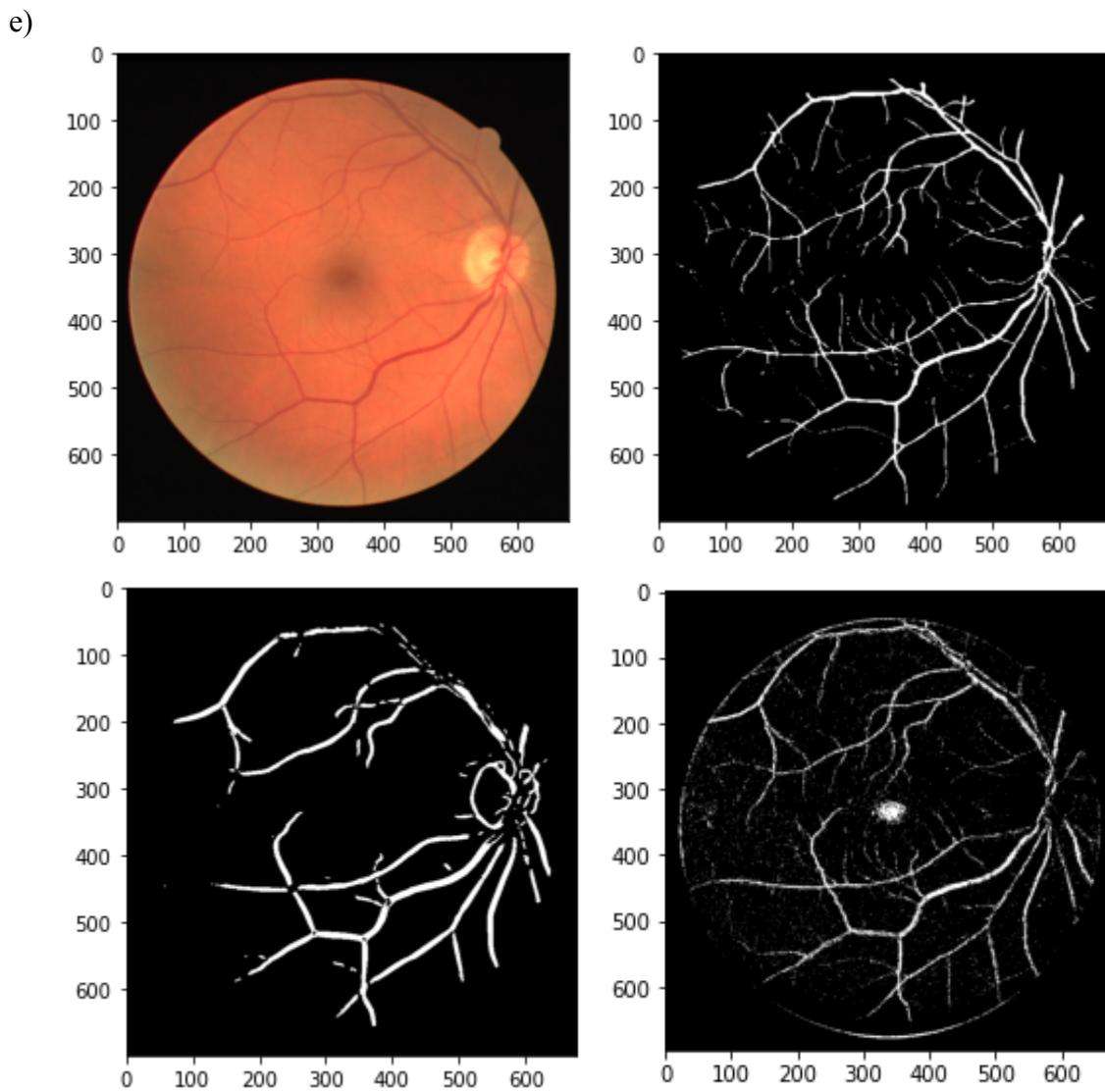


c)



d)





5. Analiza wyników

a) Przetwarzanie obrazów:

accuracy = 0.9327727368643174

sensitivity = 0.5290904040248156

specificity = 0.9623964611631683

k-nearest neighbors:

accuracy = 0.947215046647168

sensitivity = 0.5703703703703704

specificity = 0.975218458292241

b) Przetwarzanie obrazów:

accuracy = 0.9322388689596961

sensitivity = 0.6485117817279867

specificity = 0.9574720355885624

k-nearest neighbors:

accuracy = 0.9478661856066267
sensitivity = 0.6428110063299315
specificity = 0.9753435496445235

c) **Przetwarzanie obrazów:**

accuracy = 0.8978413167334881
sensitivity = 0.38102984070002244
specificity = 0.9180436422319282

k-nearest neighbors:

accuracy = 0.9602698490638543
sensitivity = 0.4035333707234997
specificity = 0.9823005903502153

d) **Przetwarzanie obrazów:**

accuracy = 0.8821544629668706
sensitivity = 0.4079985967374145
specificity = 0.9125001403248801

k-nearest neighbors:

accuracy = 0.9313955722550756
sensitivity = 0.35853359059814066
specificity = 0.9685159295757039

e) **Przetwarzanie obrazów:**

accuracy = 0.9452627136526693
sensitivity = 0.5244295634920635
specificity = 0.9679013937933059

k-nearest neighbors:

accuracy = 0.9652547981469226
sensitivity = 0.6308391897478297
specificity = 0.9834654000270137

Obie metody znajdowania naczyń krwionośnych cechują się wysoką trafnością (accuracy) i swoistością (specificity). Wynika to z faktu, że obrazy dominuje klasa negatywna - tło, a metody faktycznie oznaczają większość obrazu jako tło. W tych miarach we wszystkich przypadkach lekko lepsze wyniki daje algorytm uczenia maszynowego.

Gorsza czułość wynika z faktu, że naczyń krwionośnych jest na obrazku mało i mają specyficzne kształty. Filtry nie wyłapują mniejszych i ciemniejszych naczyń krwionośnych, a k-nearest neighbors znajduje czasami małe naczynia tam gdzie ich nie ma. Dla bardziej klasycznych/jaśniejszych przypadków (a, b i e) czułość jest bardzo zbliżona, a dla obrazów ciemniejszych lepiej sprawdza się k-nearest neighbor.

6. Wnioski

Na tym etapie obie metody dają podobne wyniki, ale uczenie maszynowe ma większy potencjał do ich poprawy. Można dokonać ekstrakcji innych cech fragmentu obrazu lub zmienić parametry algorytmu k-nearest neighbor, co ma szansę poprawić jego skuteczność. Aby uzyskać bardziej satysfakcyjne wyniki należy użyć bardziej skomplikowanego modelu uczenia maszynowego.

Obecne wyniki mogą stanowić pomoc lub punkt startu do ręcznego znajdowania naczyń krewionośnych

7. Źródła

<https://www.kaggle.com/code/amolbhivarkar/knn-for-classification-using-scikit-learn/notebook>

https://www.researchgate.net/publication/271483004_Optimized_Coronary_Artery_Segmentation_Using_Fangi_Filter_and_Anisotropic_Diffusion_Filtering

https://docs.opencv.org/3.4/d8/d23/classcv_1_1Moments.html

<https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a>

<https://scikit-image.org/docs/dev/api/skimage.morphology.html>