

# Raport - Wykrywanie naczyń dna siatkówki oka

## 1. Skład grupy

Iwo Naglik 136774

Bartosz Przybył 136785

## 2. Cel Projektu

Celem projektu było napisanie aplikacji, która dla zadanego obrazu wejściowego przedstawiającego dno siatkówki oka automatycznie wykrywa naczynia krwionośne.

Wykorzystana baza obrazów STARE:

<http://cecas.clemson.edu/~ahoover/stare/probing/index.html>

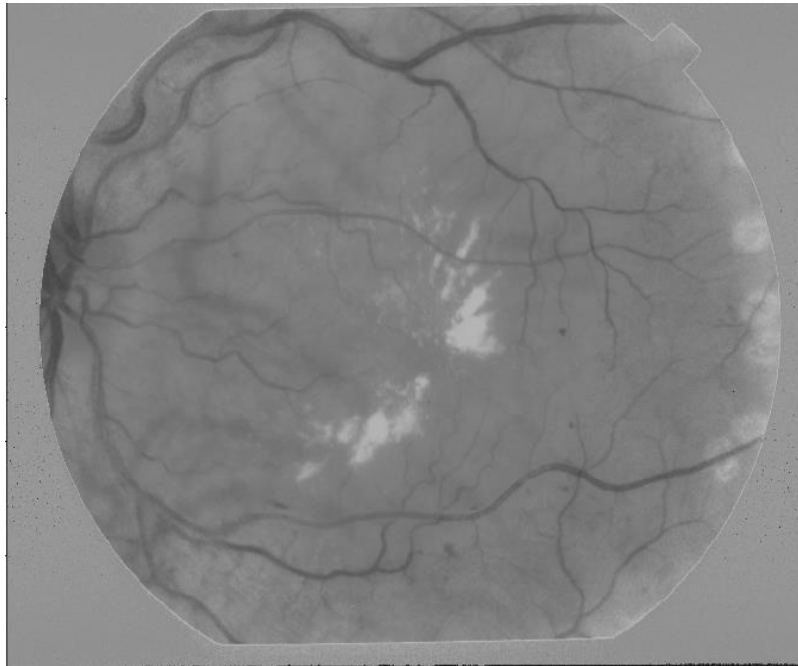
## 3. Wersja na 3.0

Cześć ta dotyczyła przetworzenia obrazu samymi technikami przetwarzania obrazu bez wykorzystania algorytmów uczenia maszynowego. Zadanie to zdaje się być ciężkie ze względu na różnorodność fotografii i problem znalezienia uniwersalnego przekształcenia dla każdej z nich. Wersję tę podzieliliśmy na etapy: wstępne przetworzenie obrazu, właściwe przetworzenie obrazu i końcowe przetworzenie obrazu. Operacje, które wykonaliśmy w tej części (obraz przykładowy, który wykorzystaliśmy do demonstracji poszczególnych operacji przetwarzania to 'im0001.ppm'):

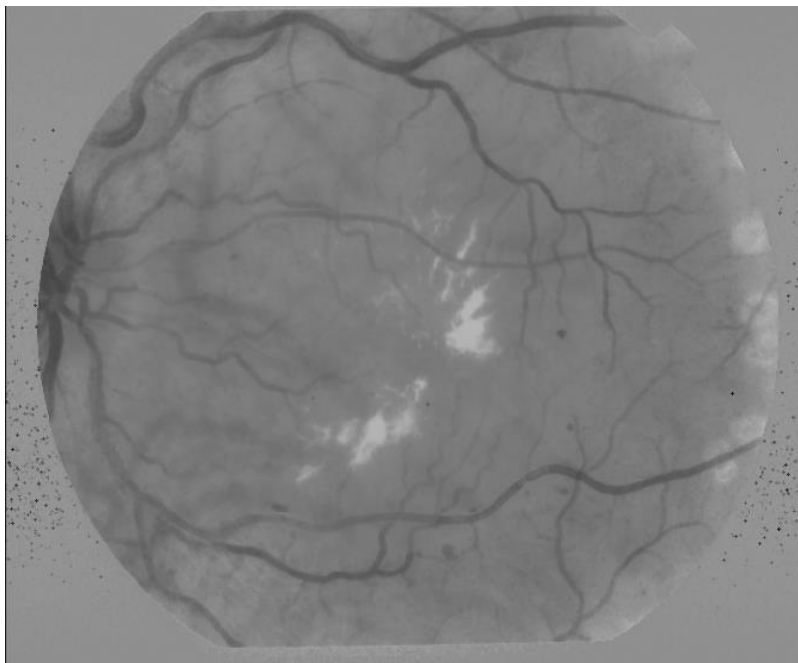
- wygląd oryginalnego zdjęcia:



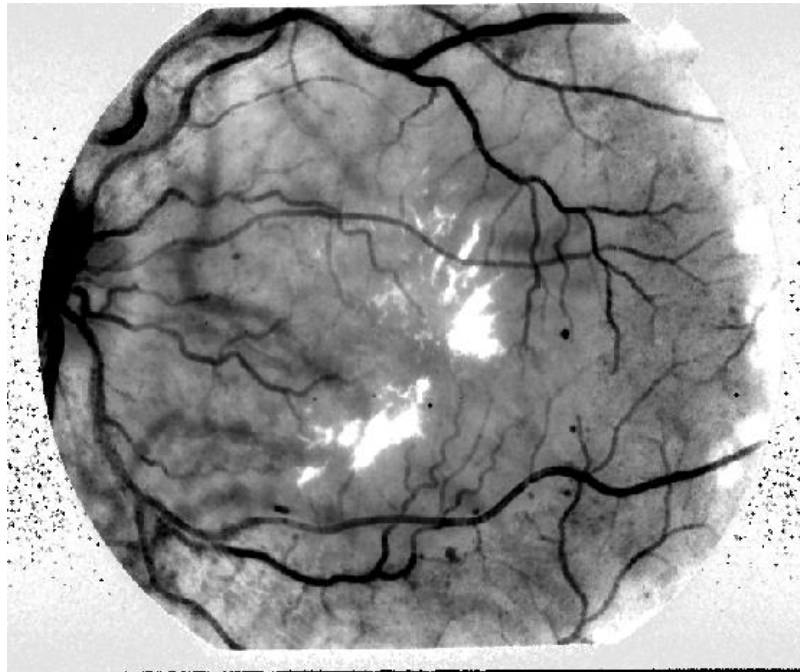
- konwersja zdjęcia do przestrzeni barw hsv, usunięcie składnika saturacji oraz poprawienie wartości value. Po tej transformacji zdjęcie przekonwertowaliśmy z powrotem do modelu barw RGB by ostatecznie rzutować je na skalę szarości i w niej do samego końca dokonywać kolejnych obliczeń



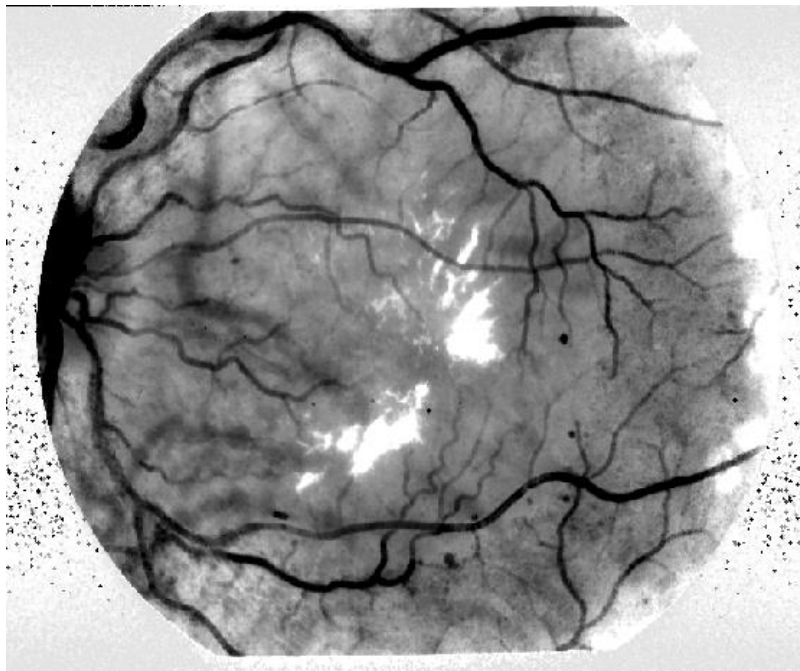
- kolejnym etapem było zastosowanie erozji w celu obniżenia wartości pikseli odstających od swojego najbliższego otoczenia. W ten sposób zmniejszyliśmy regiony jasne i powiększyliśmy te ciemne



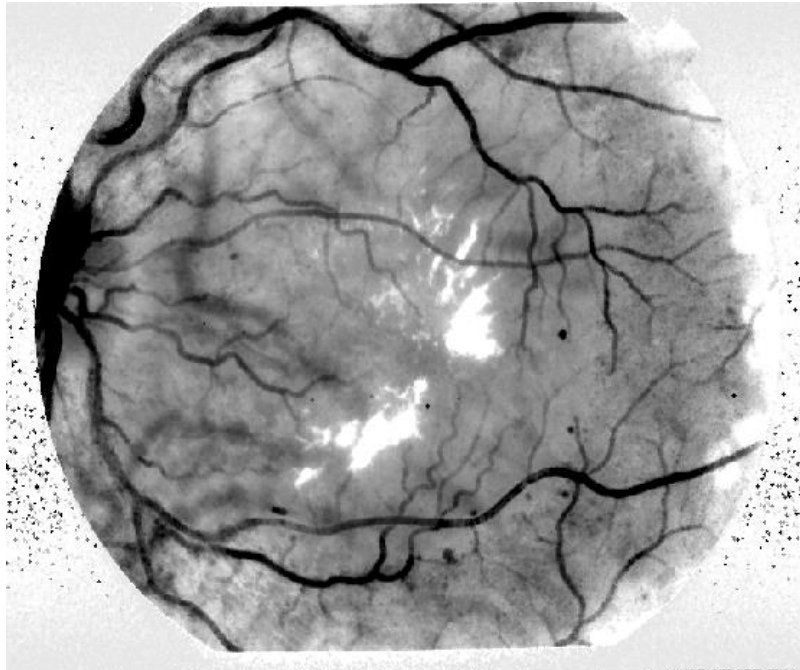
- następnie znormalizowaliśmy wartości pikseli, co uwypukliło znaczące cechy obrazu. Parametry użyte w normalizacji były oparte na wyliczeniu średniej i odchylenia standardowego wartości kolorów na obrazie. Operacja ta miała przyczynić się do uniwersalności programu względem różnych zdjęć



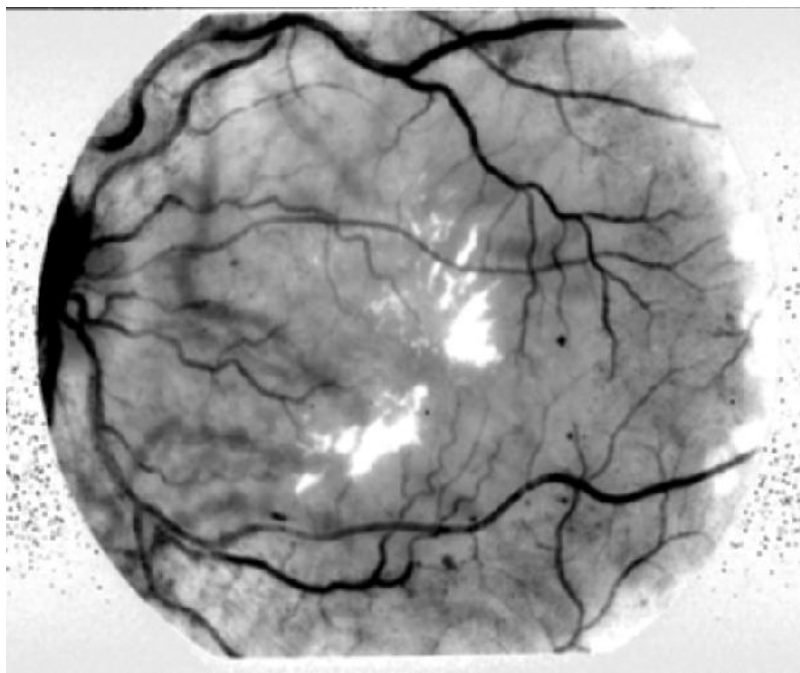
- następnie znowu dokonujemy normalizacji, lecz tym razem przy użyciu wartości percentyli obrazu



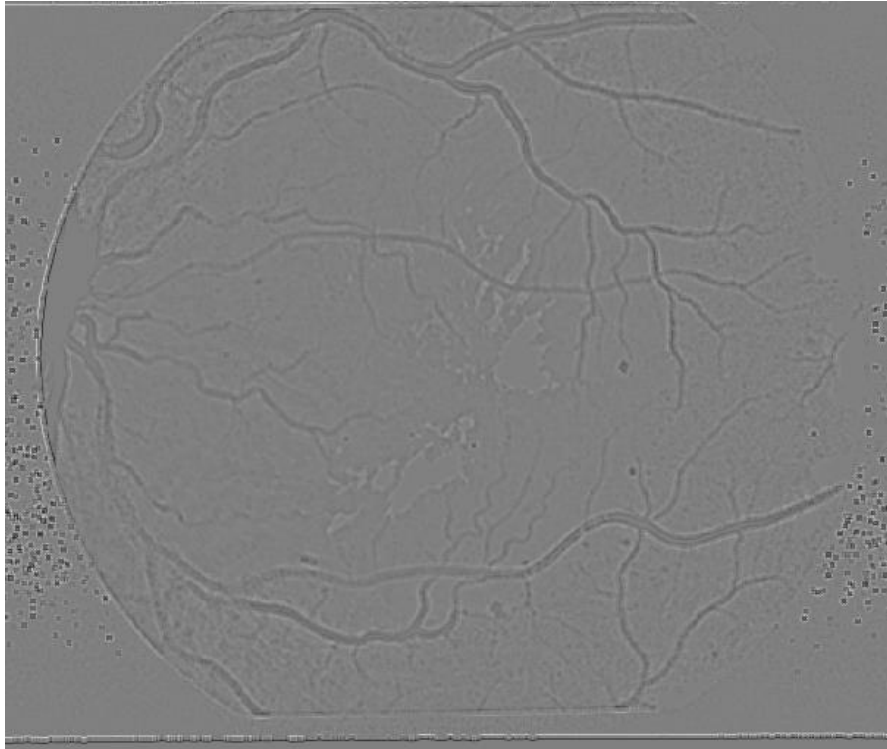
- kolejno dokonujemy transformacji gamma (podniesienie do potęgi, wyznaczonej na podstawie średniej i odchylenia standardowego kolorów obrazu) w celu zmniejszenia koloru nieznaczącego, a uwypukleniu cech, które nas najbardziej interesują



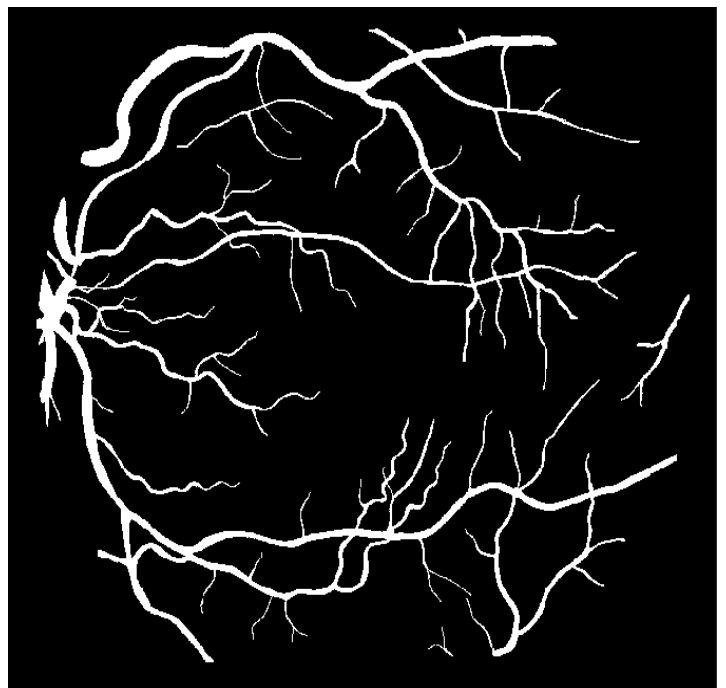
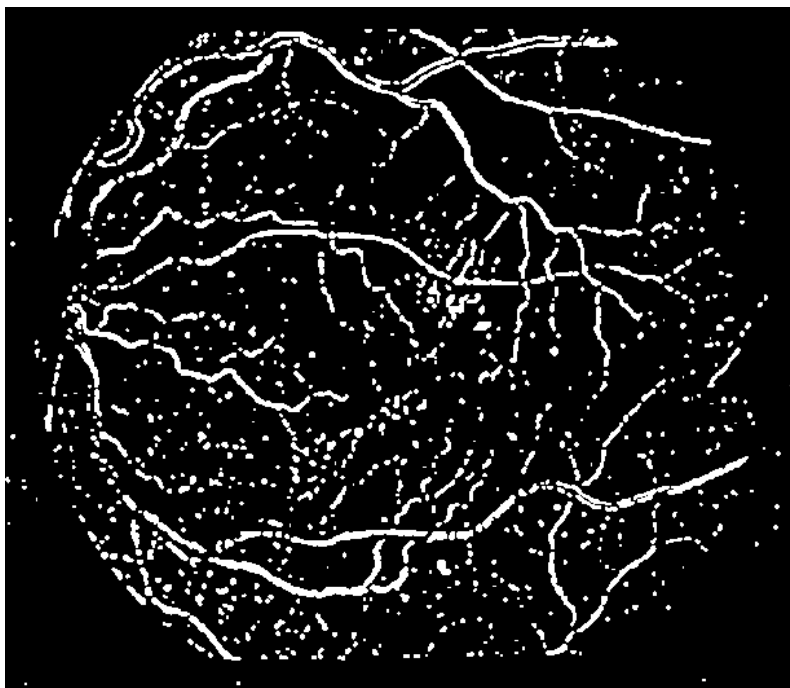
- ostatnią operacją jaką wykonaliśmy w wstępnym przetwarzaniu była operacja konwolucji



- właściwe przetworzenie obrazu polegało na zastosowaniu filtr Canny w celu pozostawieniu na obrazie potencjalnych elementów, które nas interesują.



- na zakończenie zastosowaliśmy przetwarzanie uzyskanego obrazu w celu poprawy skuteczności wykrywania naczyń poprzez „naprawę” błędów z poprzednich kroków. W tym celu użyliśmy: normalizacji na podstawie percentyli, erozji, dylatacji oraz filtru medianowego. Po tym wszystkim pozostało tylko zamienić obraz na binarny w celu porównania wyniku. Po lewej wynik otrzymany za pomocą przetwarzania obrazu, po prawej oryginalnie zaznaczone naczynia krwionośne.



Wyniki, które otrzymaliśmy były zróżnicowane. Dla niektórych obrazów algorytm spisywał się dobrze, dla niektórych nieco gorzej. Ostatecznie średnia dokładność obliczona na podstawie pięciu zdjęć wyniosła 93.8%. Czulość i swoistość wyniosła odpowiednio 63.66% i 96.31%.

Macierz pomyłek wyglądała następująco:

```
Confusion Matrix:  
[[377005  13744]  
 [   7216 25535]]
```

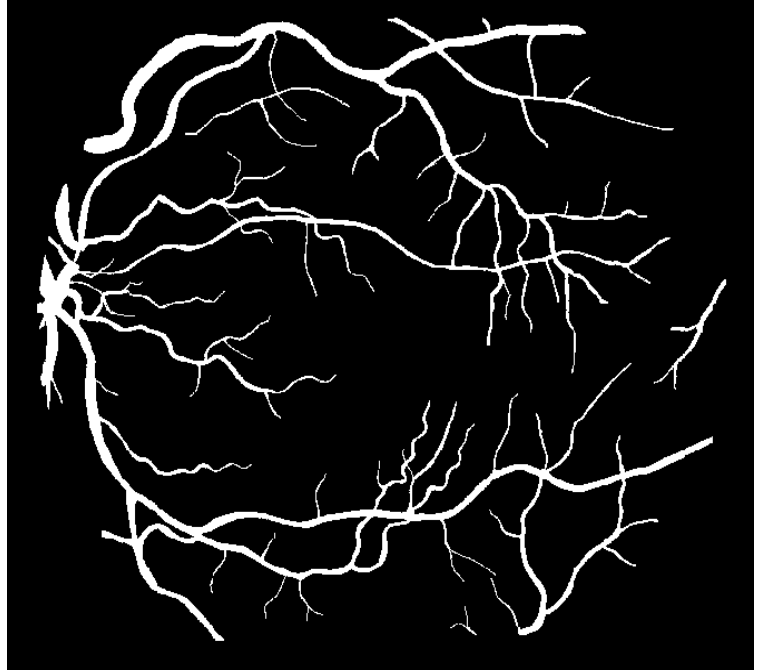
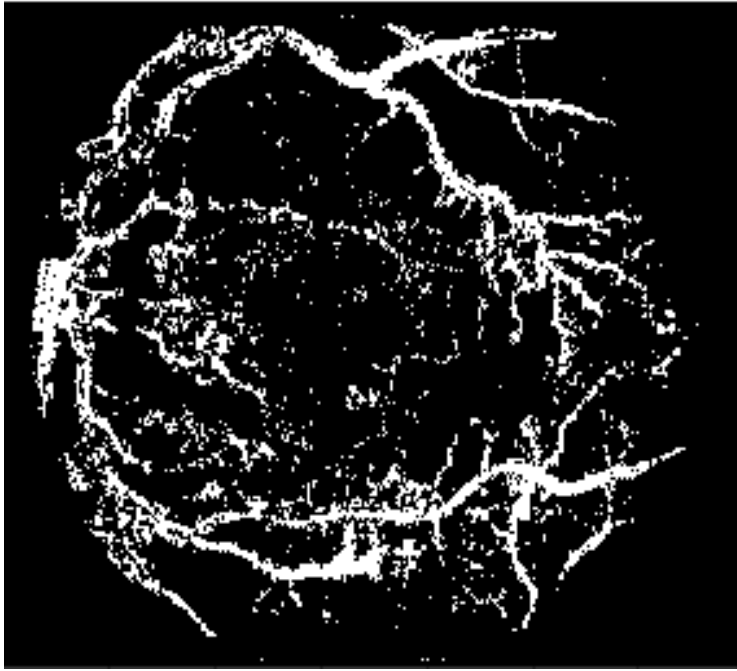
Efekt nie jest idealny, ale uważamy, że w oparciu o naszą wiedzę i doświadczenie, uzyskaliśmy w miarę dobry wynik, uwzględniając fakt, że są to operacje realizowane tylko na transformacjach obrazu.

#### 4. Wersja na 4.0

Druga część zadania opierała się na skonstruowaniu prostego klasyfikatora odległościowego. W naszym rozwiązaniu użyliśmy algorytmu kNN, czyli k-najbliższych sąsiadów. W celu przygotowania danych do analizy użyliśmy zdjęcia, które uzyskaliśmy po wstępnym przetworzeniu z poprzedniej części zadania, a następnie podzieliliśmy je na kwadraty o wymiarach 5x5 w celu ekstrakcji cech. Dla każdego kwadratu wybraliśmy cechy takie jak: momenty centralne, momenty Hu, wariancja i średnia wartość kolorów pikseli, odległość centralnego piksela od środka obrazu, oraz wartość środkowego piksela z maski. Podzieliliśmy również obraz na trzy regiony w celu rozdzielenia obrazu na ten bardziej znaczący i mniej znaczący region. Z uwagi na ograniczone zasoby komputerowe, postanowiliśmy zastosować losowy dobór punktów centralnych dla maski z uwzględnieniem proporcji ilości punktów zawierających naczynia do punktów ich niezawierających. W tym celu oszacowaliśmy ilość pikseli które zawierają naczynia krwionośne. Kolejno wylosowaliśmy 70% punktów z tych właśnie pikseli oraz taką samą ilość punktów z pikseli niezawierających tych naczyń. Zanim użyliśmy klasyfikatora, nasze dane przetasowaliśmy w celu uniknięcia skupiska podobnych wzorców. Do celów uczących użyliśmy ośmiu różnych zdjęć. Kolejno, wprowadziliśmy dane do naszego klasyfikatora, by ostatecznie poddać je operacji postprocessingu, na który składały się operacje: otwarcia, erozji, zamknięcia i użycia filtra medianowego. Następnie nasz wygenerowany model sprawdziliśmy na zbiorze testowym, który zawierał 4 nieznane wcześniej obrazy, z którego wzięliśmy wszystkie piksele. Średnia dokładność naszego modelu wyniosła 83.71%. Czulość i swoistość wyniosły odpowiednio 62.10% i 85.48%. Uważamy, że bylibyśmy w stanie uzyskać nieco lepszy wynik, lecz ze względu na ograniczenia, musieliśmy zrezygnować z większego zbioru uczącego oraz użycia metody przeszukiwania siatki w celu znalezienia najlepszych parametrów modelu. Macierz pomyłek prezentowała się w następujący sposób:

```
Confusion Matrix:  
[[1321983 224536]  
 [  48000 78665]]
```

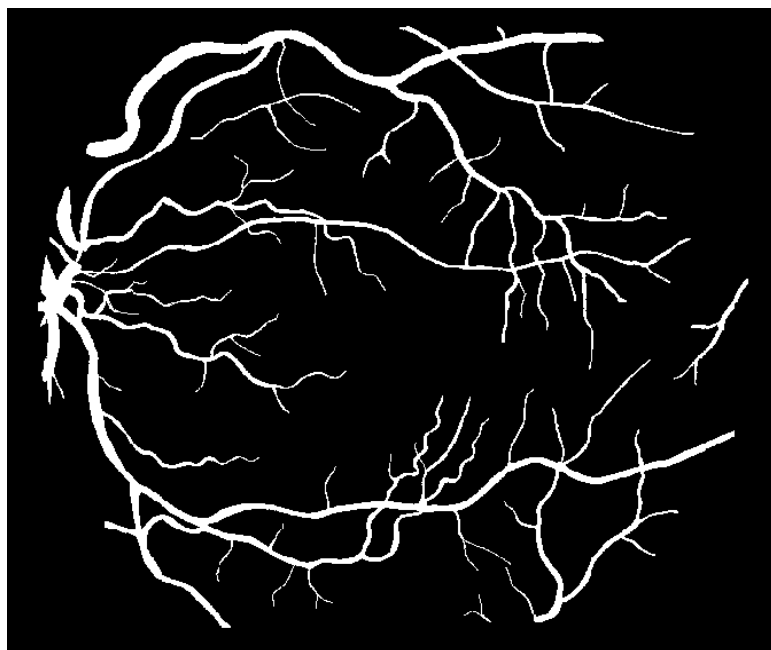
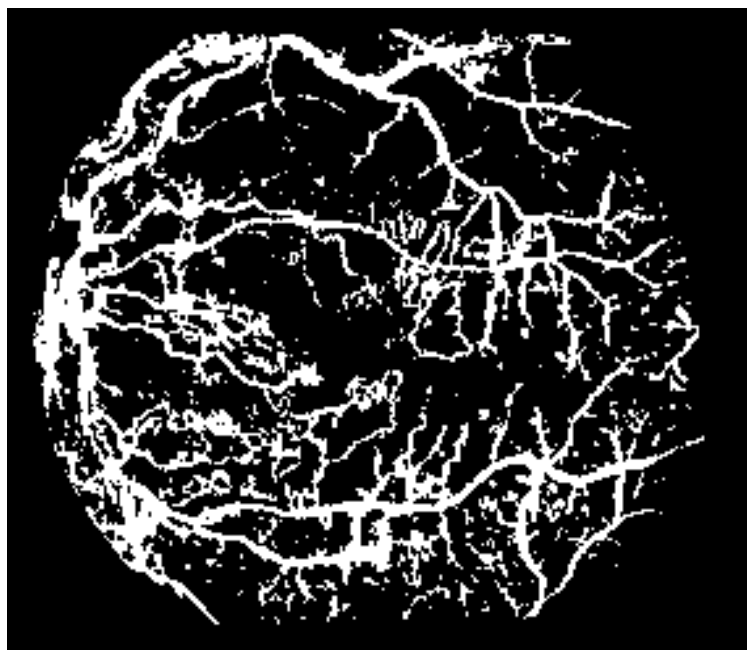
Po lewej przykładowe wynikowe zdjęcie ze zbioru testowego, po prawej oryginalnie zaznaczone naczynia krwionośne.



#### 5. Wersja na 5.0

Ostatnia część była zbliżona do drugiej, lecz w tym wypadku użyliśmy bardziej zaawansowanego modelu klasyfikacji, jakim jest losowy las. Również tutaj musieliśmy ograniczyć zbiór uczący oraz wykorzystać klasyfikator z domyślnymi parametrami. Jednak dużo lepiej sprawował się model losowego lasu. Dzięki operacji postprocessingu uzyskaliśmy wartość sensitivity (która dla niezbalansowanych danych jest bardzo istotna) 81.124%, a wartość specificity: 84.34%. Accuracy wyniosło: 84.11%. Wyniki uzyskane podczas procesu krosvalidacji wyniosły odpowiednio: [0.83733413 0.83239664 0.86116852].

Po lewej przykładowe wynikowe zdjęcie ze zbioru testowego, po prawej oryginalnie zaznaczone naczynia krwionośne.



6. Podsumowanie - główną rolę i bazę do celów uczenia algorytmów, stanowi dobre przetworzenie obrazu i dobre przygotowanie danych i od tego w dużej mierze, zależy dokładność naszego algorytmu klasyfikującego.