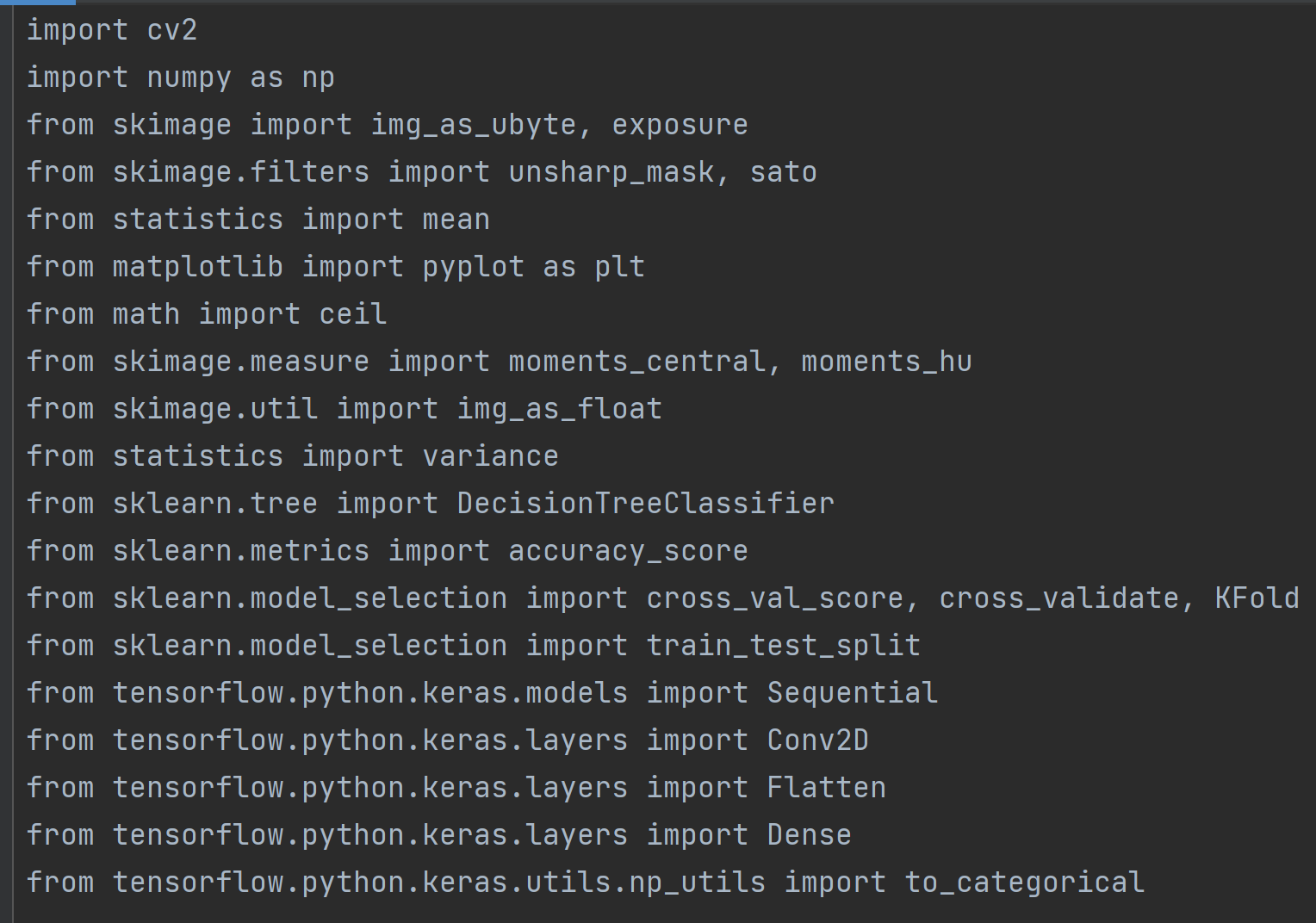
|  |  |  |  |
| --- | --- | --- | --- |
| **Laboratorium z przedmiotu Informatyka w Medycynie (IWM)** | | | |
|  | | | |
| Projekt nr 2 | | | |
|  | | | |
| ***Temat: Dno Oka*** | | | |
|  | | | |
| *Prowadzący:*  *mgr inż. Iwo Błądek* | *Autor*  Tobiasz Gruszczyński 145333  Sebastian Grabowski 145248 | *Grupa dziekańska:* | *L15* |
| ***Ocena:*** |  |
|  | | | |

**Zastosowany język programowania**

Zaimplementowaliśmy wykrywanie naczyń dna siatkówki oka w języku Python oraz wykorzystaliśmy Jupyther Notebook.

***Wykorzystane biblioteki:***

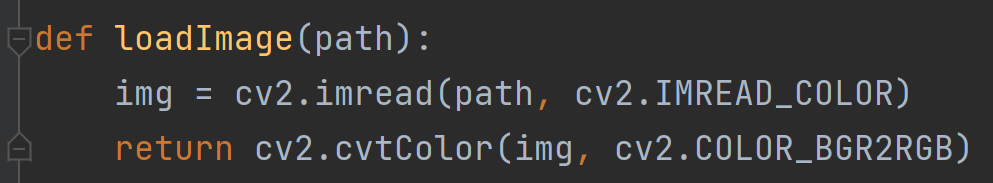


**Rys.1 Biblioteki**

**Opis zastosowanych metod**

***Przetwarzanie obrazu (poszczególne kroki):***

Krok pierwszy to wczytanie oryginalnego obrazu poniższą funkcją. Aby prawidłowo wyświetlać barwy ustawiamy tutaj własność ***cvtColor*** na ***COLOR\_BGR2RGB***.

** Rys.2 Wczytywanie zdjęcia**

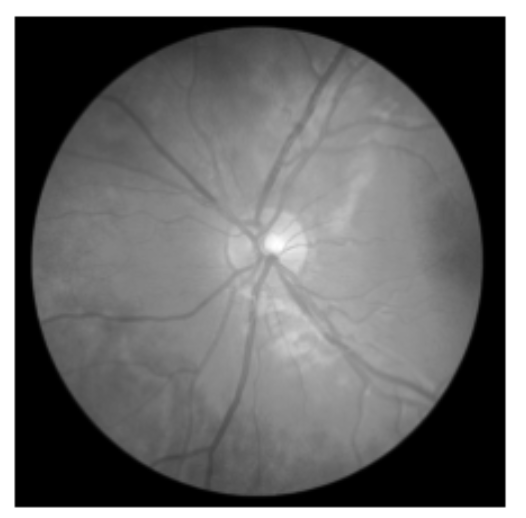
****

**Rys.3 Obraz po wykonaniu loadImage**

Krok drugi to preprocessing wykonywany na oryginalnym obrazie, który składa się z kilku etapów:

1. Zmienienie obrazu na skalę szarości

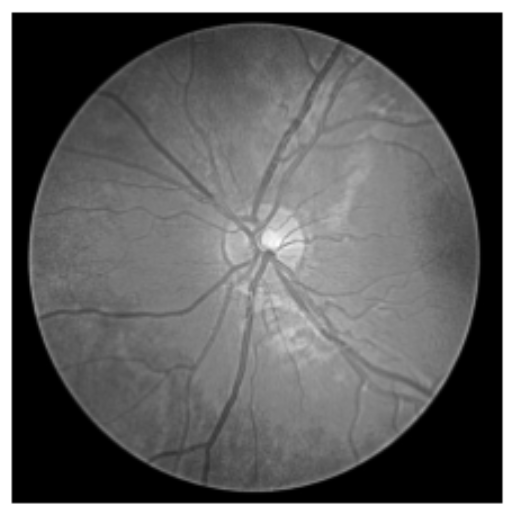




**Rys.4 Obraz po 1 kroku preprocessingu**

1. Poprawienie kontrastu zdjęcia przy wykorzystaniu filtru ***unsharp\_mask*** z biblioteki ***skimage***.

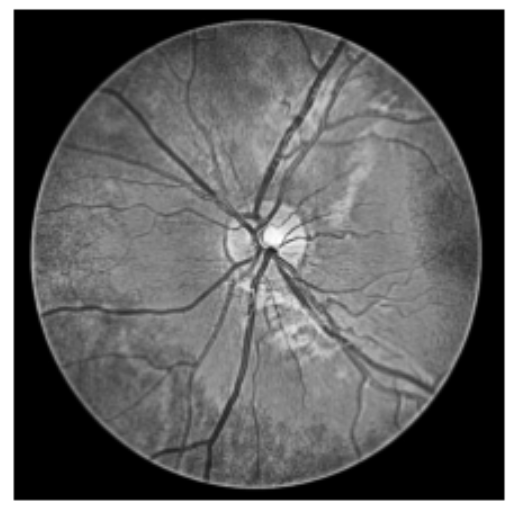




**Rys.5 Obraz po 2 kroku preprocessingu**

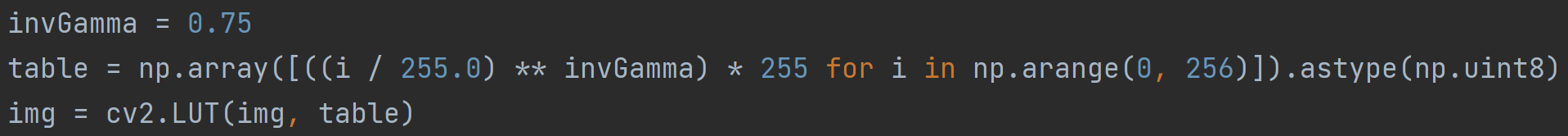
1. Znormalizowanie histogramu kolorów przy wykorzystaniu funkcji ***exposure.equalize\_adapthist()*** z biblioteki ***skimage***.

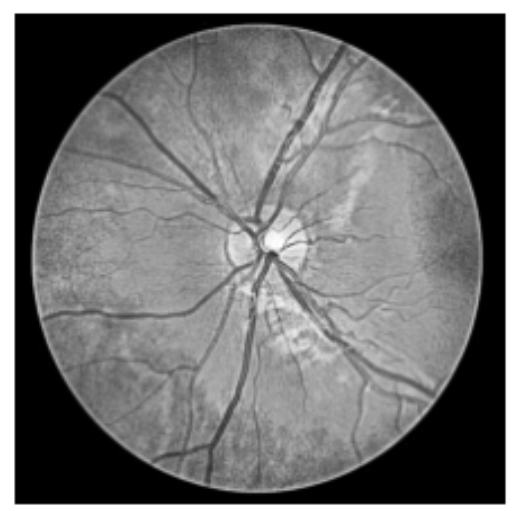




**Rys.6 Obraz po 3 kroku preprocessingu**

1. Zmienienie wartości gamma w celu rozjaśnienia ciemnych miejsc obrazu, pozostawiając przy tym skrajnie ciemne miejsca niezmienonymi.

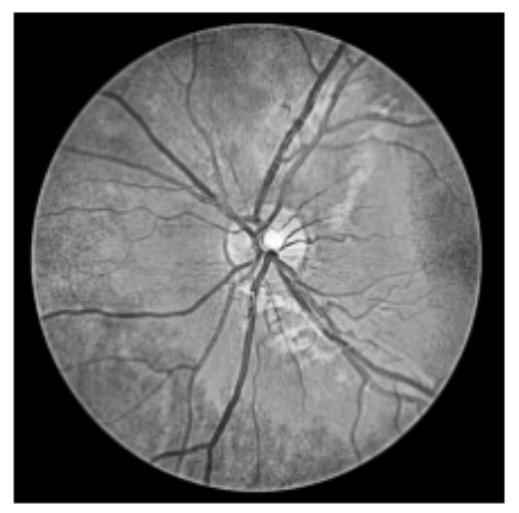




**Rys.7 Obraz po 4 kroku preprocessingu**

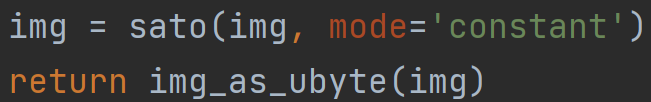
1. Zastosowanie erozji, aby uzyskać obraz łatwiejszy w dalszym przetwarzaniu.

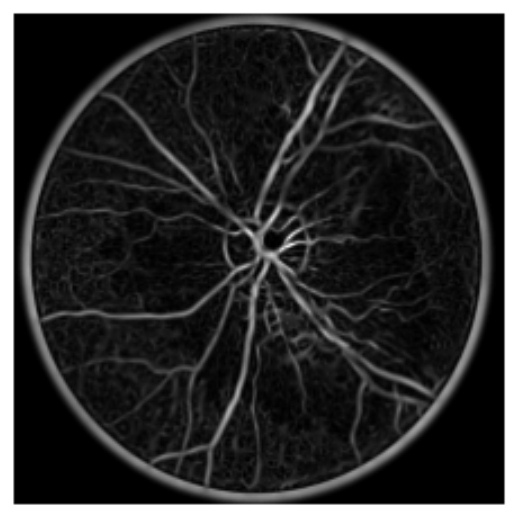




**Rys.8 Obraz po 5 kroku preprocessingu**

Krok trzeci to właściwe przetwarzanie obrazu. Zdecydowaliśmy się podejść do problemu podobnie jak do wykrywania rzek na mapie. Zastosowaliśmy w tym celu filtr ***sato***.





**Rys.9 Obraz po zastosowaniu filtru sato**

Krok czwarty polegał na poprawieniu kontrastu zdjęcia, zastosowaniu dylatacji, aby odwrócić proce erozji. Na końcu znormalizowano obraz.



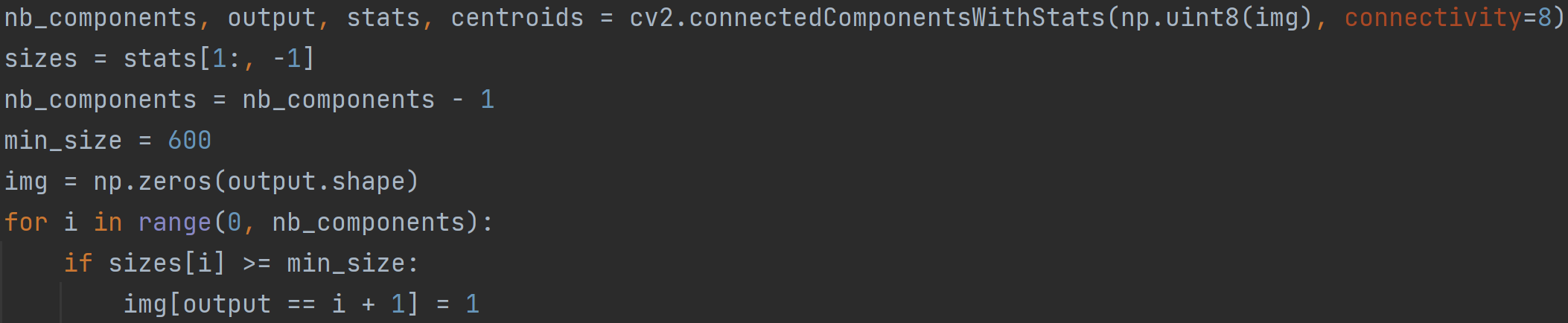


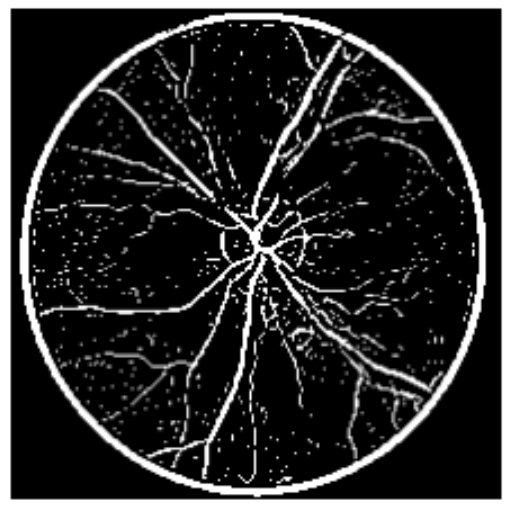




**Rys.10 Obraz po zastosowaniu powyższych funkcji**

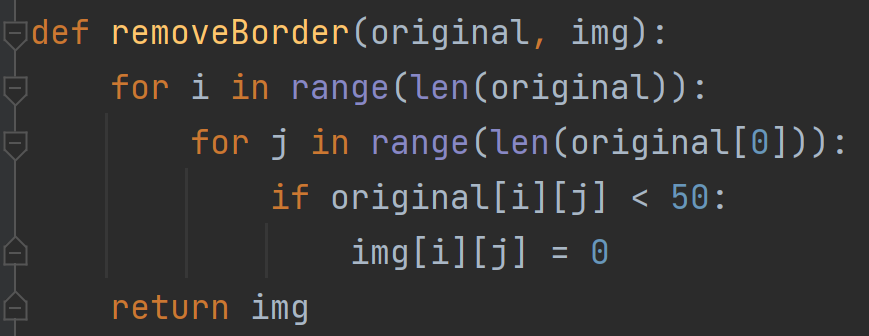
Krok piąty to utworzenie czarno-białej tablicy bez odcieni szarości na podstawie otrzymanego w poprzednim kroku obrazu.





**Rys.11 Obraz binarny**

Ostatni krok szósty polegał na usunięciu krawędzi wykrytych przez filtr ***sato***.



**Rys.12 Usuwanie krawędzi**



**Rys.13 Obraz końcowy**

***Uzasadnienie wybranego rozwiązania:***

Wykorzystując doświadczenie zdobyte na przedmiocie KCK, gdzie przyszło nam się zmierzyć z podobnym problem (wykrywanie samolotów na obrazie), udało nam się znaleźć opcje przetwarzania z filtrem ***sato***, którą uzyskaliśmy zadowalającą jakość obrazu.

Wybraliśmy filtr ***sato***, ponieważ powstał on właśnie w celu rozwiązywania takich problemów jak wykrywanie naczyń. Jedyny problem to obramowanie oka, które jest wykrywane jako naczynie. Ten problem rozwiązuje końcowe przetwarzanie, gdzie po usunięciu obramowania oraz stworzeniu maski binarnej, mamy możliwość porównania obrazu z maską ekspercką.