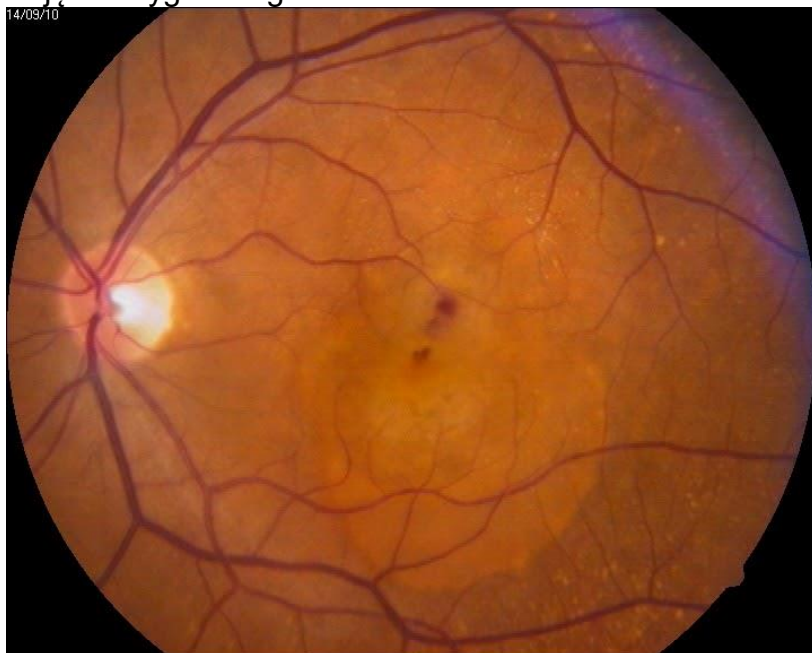
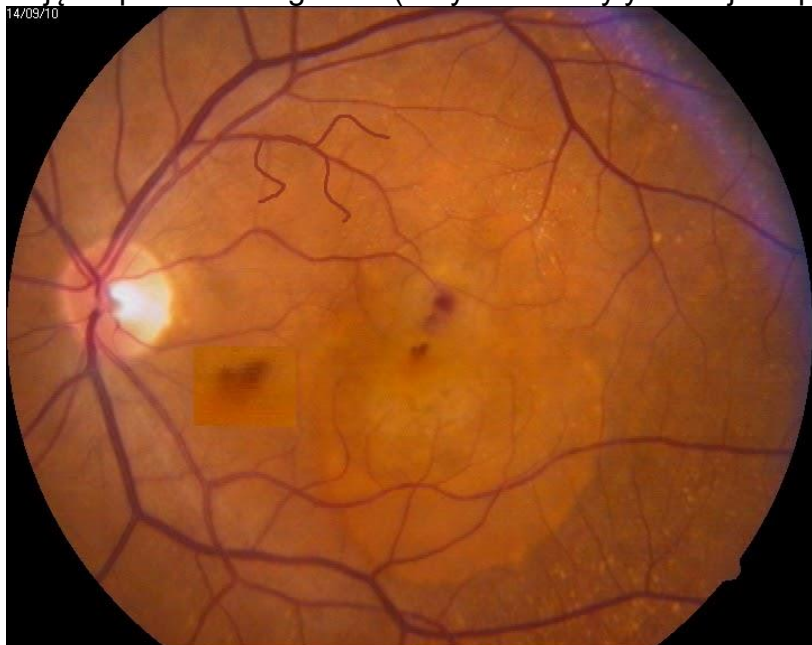


Program wczytuje dwa zdjęcia o rozszerzeniu .bmp po ich nazwach (np. "zdjęcie.bmp"). Zdjęcia te nie muszą mieć takich samych wymiarów ponieważ program automatycznie wybierze pokrywające się fragmenty. Zdjęcia te nie mogą być jednak większej rozdzielczości niż 1000x1500 jeżeli chce się skorzystać z wariantu generowania bitmapy różnicowej. Przy większych wymiarach nie mieści się ona w oknie konsoli, a jej przetwarzanie trwa prawie minutę. Program liczy jasność pixeli za pomocą sumowania wektorów kolorów rgb. Testowane zdjęcia to okoOriginal.bmp i okoZDefektem.bmp które są oryginalnym i przerobionym zdjęciem przekonwertowane na .bmp podanym w zadaniu okiem. Przerobione zdjęcie posiada wklejoną czarną plamkę po lewej stronie i dorysowane żyły pędzlem w paintcie. Obydwa zdjęcia dodane są do projektu. Warianty 1-2 wskazują na różnice za pomocą liczb. Łatwiejszym wynikiem do analizy dla człowieka jest ostatnia opcja, czyli bitmapa różnic.

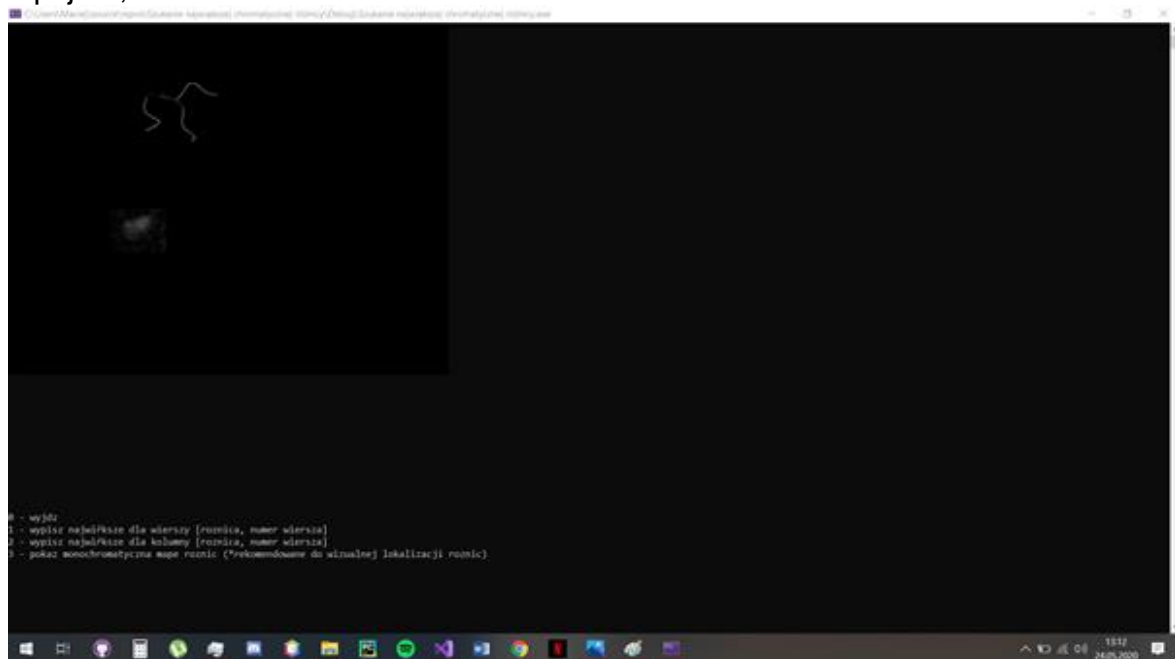
Zdjęcie oryginalnego oka:



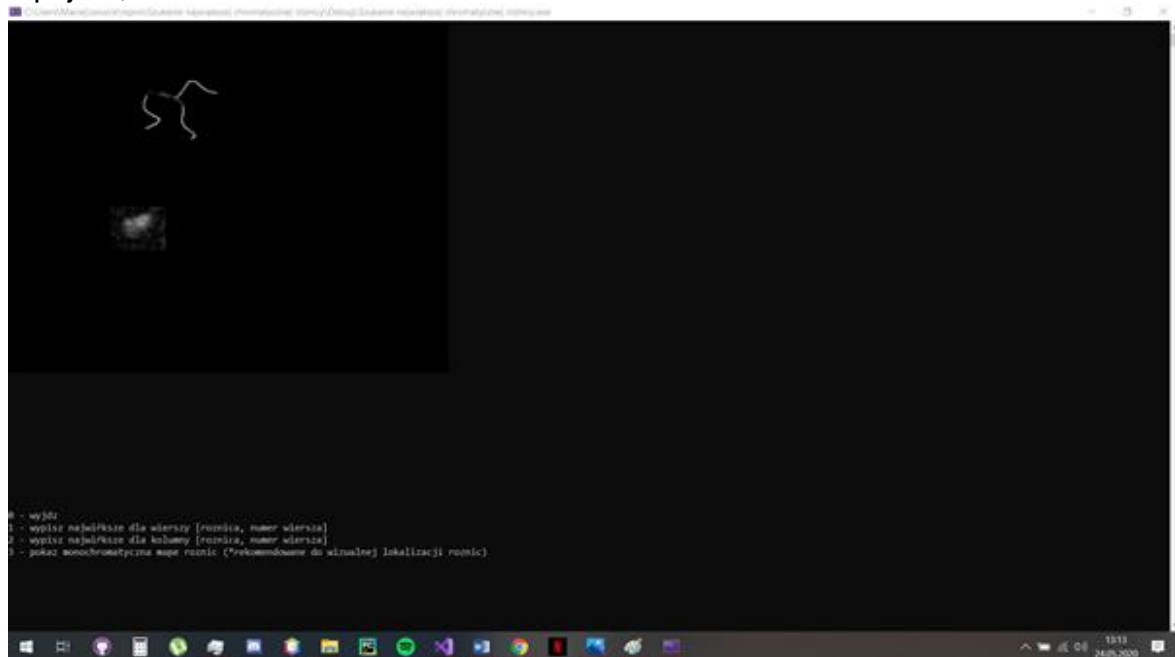
Zdjęcie przerobionego oka (dorysowane żyły i wklejona planka po lewej stronie):



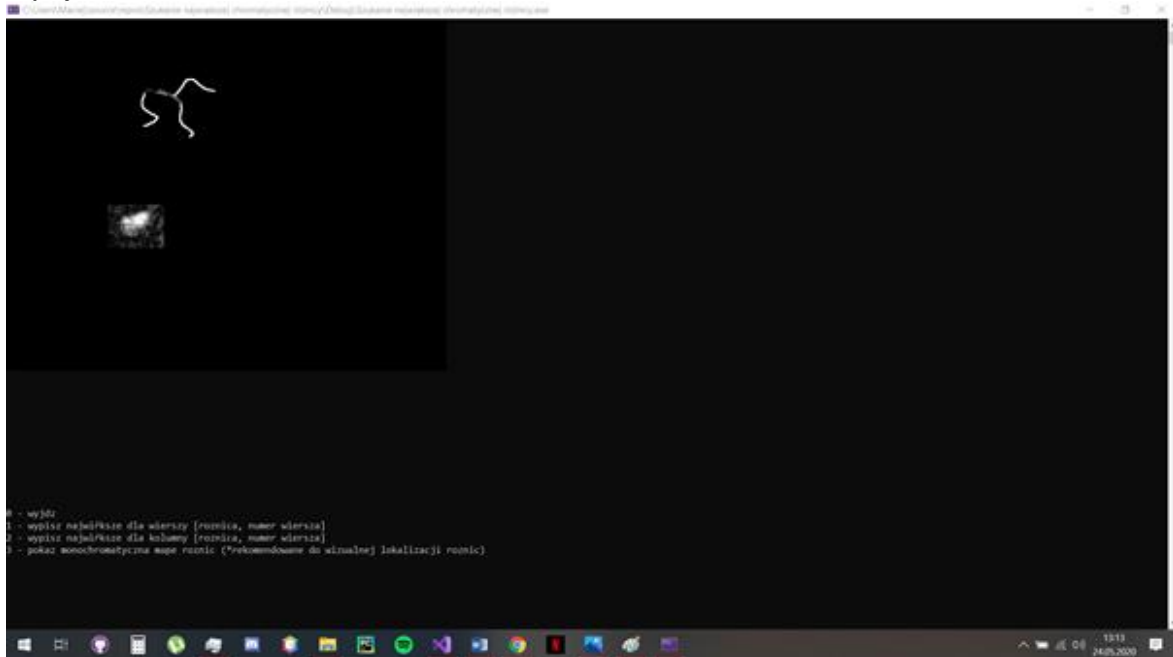
Wyniki wariantu dodatkowego bitmapy różnicowej dla różnych kontrastów:  
Opcja 3, kontrast 10



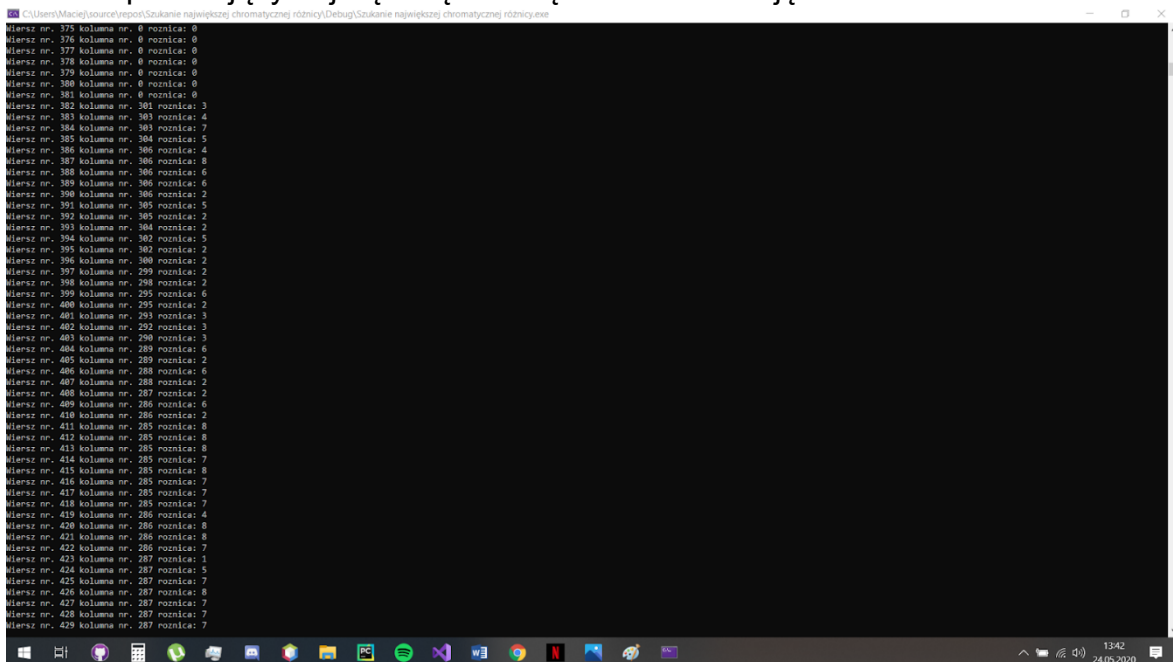
Opcja 3, kontrast 20



### Opcja 3, kontrast 30



Wariant pokazujący największą różnicę wiersza dla zdjęć oka:



Różnice zdjęć widać dobrze kiedy są one ze sobą spasowane. Jednak kiedy kamera ujęła by inny kadr tego samego oka algorytm wykrył by to jako różnice w zdjęciach. Problem ten można rozwiązać wykonując obliczenia brute force w dopasowywaniu zdjęć i nakładaniu ich elementów, jednak zużyłoby to ogromne ilości mocy obliczeniowej którymi nie dysponuje przeciętny komputer. Innym rozwiązaniem są sieci neuronowe i uczenie maszynowe które dopasowało by zdjęcia na podstawie punktów charakterystycznych.