Studenci: Michał Leszczyński, Marek Kiełtyka

Numery albumów: 297883, 297870

Rok studiów: trzeci

Kierunek: Informatyka Stosowana

**BINARYZACJA**

Dokumentacja projektu zrealizowanego

w ramach ćwiczeń projektowych

przedmiotu „Analiza Obrazów”

1. Opis działania i przeznaczenia projektu

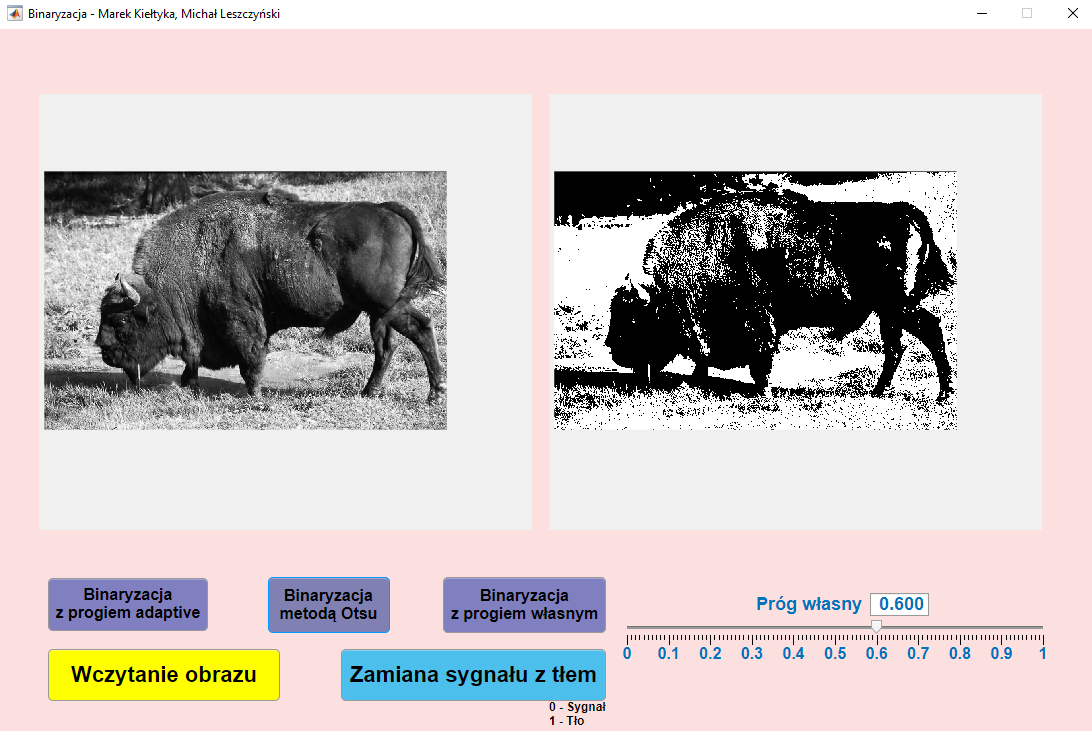
Projekt został zrealizowany w formie aplikacji programu Matlab z graficznym interfejsem użytkownika. Jej głównym zadaniem jest wczytywanie i progowanie (*thresholding*) obrazów. Procesowanie danych wejściowych odbywa się na odpowiednich, sąsiadujących ze sobą pulpitach. Po załadowaniu obrazu użytkownik może za pomocą przycisków dokonać binaryzacji z progiem:

* ustalonym przez niego (należy odpowiednio ustawić suwak lub bezpośrednio wprowadzić wartość w okienku nad suwakiem)
* wyznaczanym metodą Otsu
* wyznaczanym metodą Adaptive

Dodatkową funkcjonalnością jest możliwość zamiany kolorów tła i sygnału na obrazie, niezależnie od użytej metody. Obraz musi najpierw zostać wczytany oraz zbinaryzowany. Aplikacja została zabezpieczona przed niepoprawną kolejnością czynności wykonywanych przez użytkownika. W momencie zaistnienia takiego przypadku program wyświetla stosowne komunikaty.

Aby uruchomić program należy otworzyć plik z rozszerzeniem .*mlappinstall* w menu *Apps->Install App*, a następnie wybrać dodaną aplikację (działanie sprawdzone na wersji R2018b).

Program wczytuje obrazy z rozszerzeniami .*jpg, .png, .jpeg*.

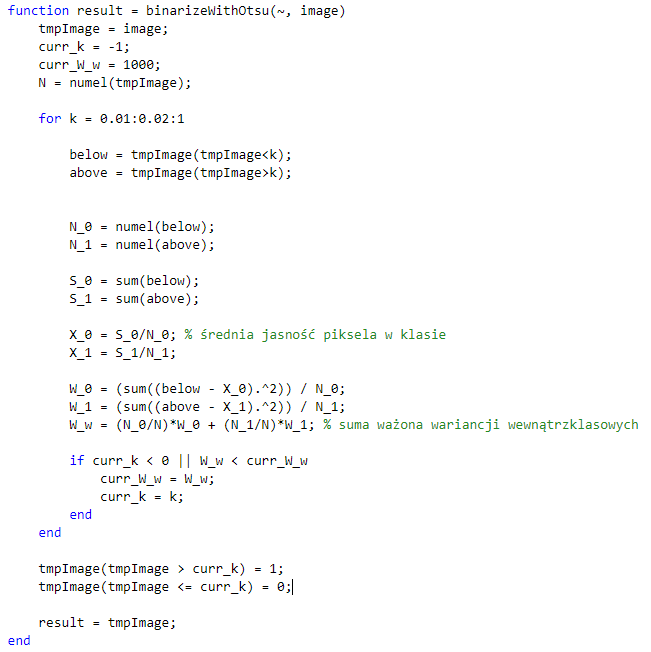
Rysunek 1. Wygląd aplikacji po starcie 

Rysunek 2. Przykładowe działanie programu

1. Zastosowane algorytmy

* **Metoda Otsu**

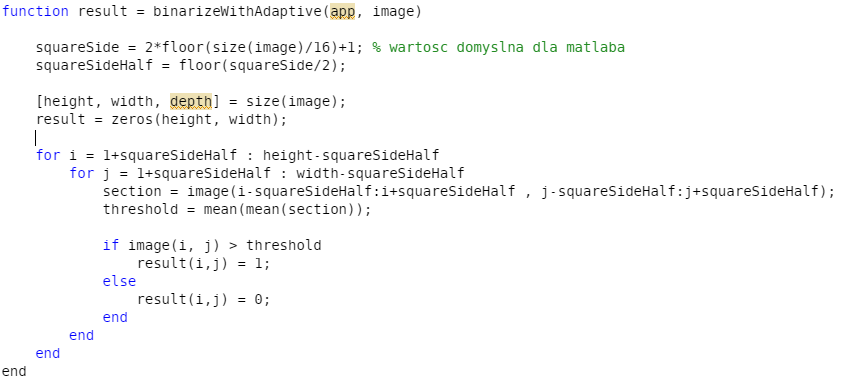
minimalizacja wariancji wewnątrzklasowych za mało Michał, dopisz tu coś więcej jak przy Adaptive



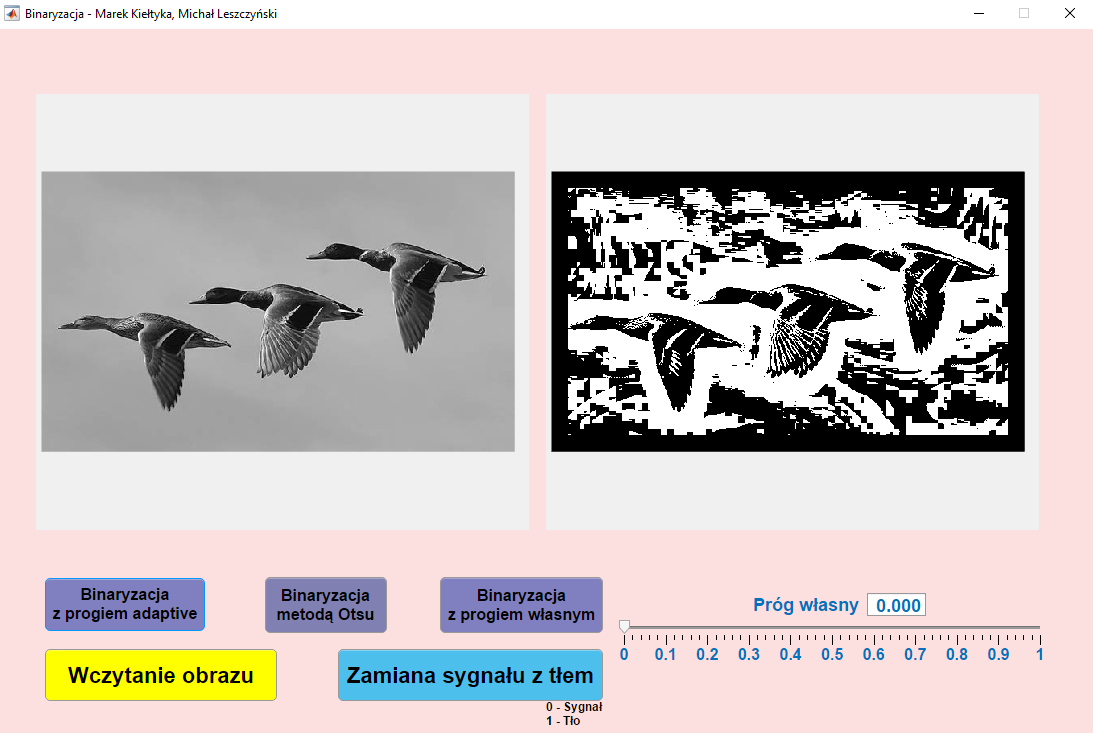
Rysunek 3. Kod metody Otsu

* **Adaptive**

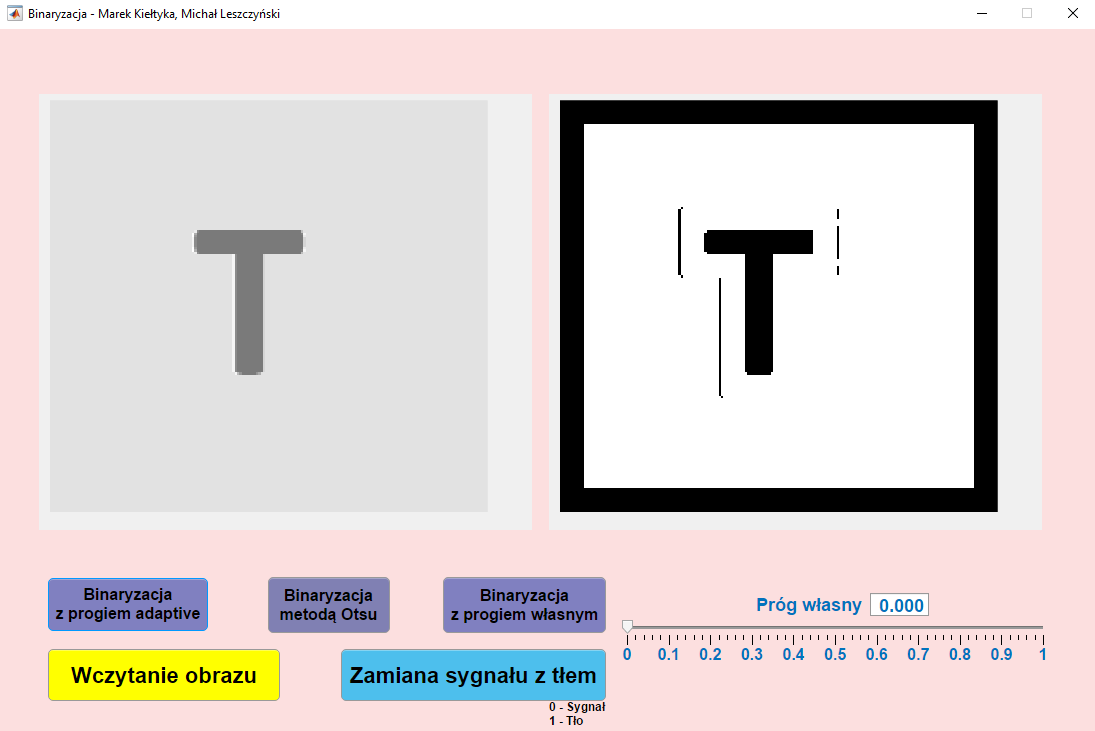
Idea algorytmu opiera się na lokalnej analizie obrazu (fragment w kształcie kwadratu, dla którego w każdej iteracji obliczana jest średnia progu sąsiadujących pikseli). Okno fragmentu jest przesuwane kolumnami, zaś w momencie dotarcia do prawej krawędzi rysunku, obniża się o jeden wiersz pikseli w dół i ponownie iteruje od lewej do prawej. Uzyskuje się zadowalające rezultaty zarówno dla nieregularnych kształtów obiektów, jak i zbliżonych do foremnych figur geometrycznych.



Rysunek 4. Kod metody ‘adaptive’



Rysunek 5. Metoda progowania ‘adaptive’ dla obiektów o nieregularnych kształtach

Rysunek 6. Metoda progowania ‘adaptive’ dla obiektów o regularnych kształtach

1. Wady projektu

Przy algorytmie ‘adaptive’ dla większych obrazów (krawędź o długości tysiąca pikseli i więcej) czas oczekiwania na wynik jest odczuwalnie dłuższy.

1. Podział obowiązków

* Projekt oraz wykonanie GUI, binaryzacja metodą adaptive i z progiem użytkownika, dokumentacja – Marek Kiełtyka
* Binaryzacja metodą Otsu oraz metodą adaptive, dokumentacja – Michał Leszczyński

1. Ewentualne usprawnienia

W przyszłości można by dokonać optymalizacji obliczania średniego progu dla sąsiadujących wartości pikseli, co usprawniłoby procesowanie obrazów o dużych rozmiarach. Ponadto ciekawym dla użytkownika usprawnieniem byłoby dynamiczne progowanie obrazu w zależności od chwilowego położenia suwaka (wyeliminowanie konieczności klikania przycisku).