**Analiza i Przetwarzanie Obrazów Biometrycznych**

**Sprawozdanie I**

Michalik Piotr

Spis treści

[Wstęp 2](#_Toc497779561)

[Skala szarości 2](#_Toc497779562)

[Inwersja 3](#_Toc497779563)

[Edycja jasności 4](#_Toc497779564)

[Kontrast 5](#_Toc497779565)

# 

# Wstęp

W poniższym dokumencie przedstawię metody używane w procesie analizy obrazów. Opiszę ich działanie oraz przestawię własną implementacje. Do każdej metody dodatkowo załączę screenshoty pokazujące przykładowe działanie.

# Skala szarości

## Teoria

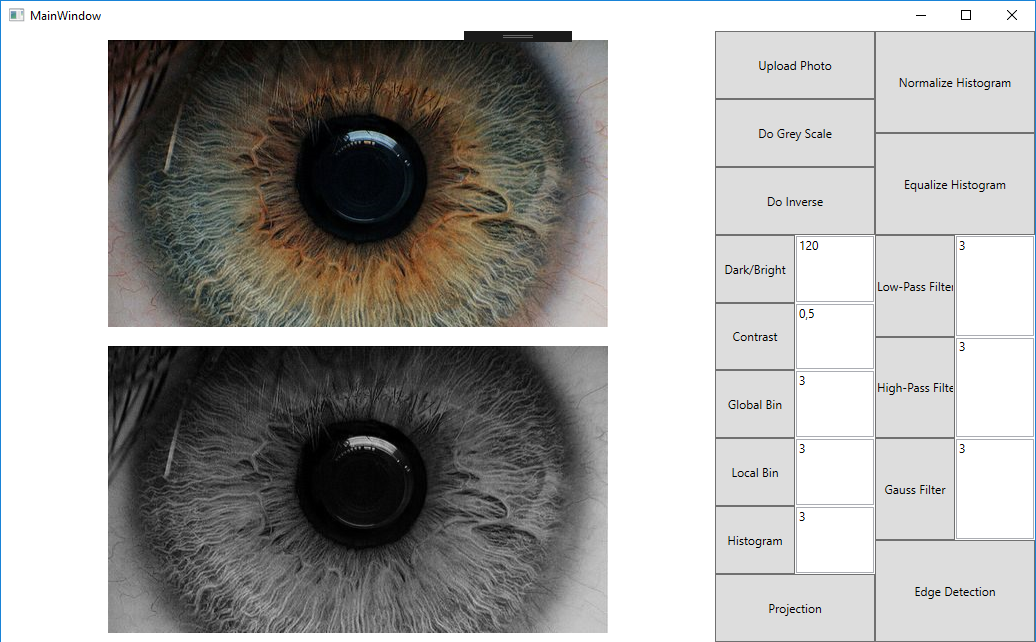
Każdy kolorowy obraz zawierający trzy kolory składowe (czerwony, zielony, niebieski) można przestawić w odcieniach szarości. To znaczy że każdy piksel będzie posiadał taką samą wartość koloru w miejscu pojedynczych składowych (czerwony = zielony = niebieski). Skala szarości jest jednym z najczęściej używanych algorytmów ponieważ jest często potrzebny do realizowania innych bardziej zaawansowanych metod i algorytmów.

Wartość nowego koloru (szarości) uzyskujemy według wzoru

## Implementacja

Program dla wczytanego zdjęcia iteruje po wszystkich jego pixelach. Następnie zgodnie z powyższym wzorem zamienia piksele kolorowe na szare. Następnie nowy obraz jest zwracany.

## ScreenShot



# Inwersja

## Teoria

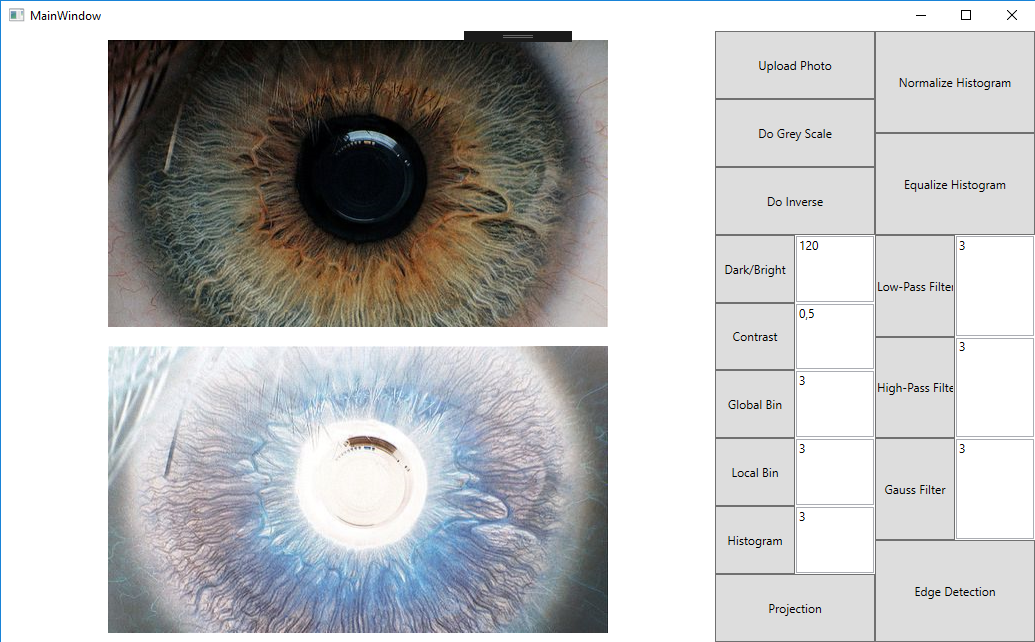
Potocznie nazywana negacją, jest jakoby odwróceniem kolorów w obrazie. Dla przykładu: biały kolor stanie się czarnym i na odwrót. Dla każdej składowej kolory obliczamy inwersję osobno według wzoru

Ponieważ składowe kolorów przyjmują wartości z przedziału [0;255] widzimy, że dokładnie odwracamy wartości kolorów.

## Implementacja

Program dla każdego pixela w obrazie oblicza negacje wszystkich jego składowych. Następnie tak obliczony nowy kolor zastępuje stary. Nowy obraz jest wyświetlany.

## ScreenShot



# Edycja jasności

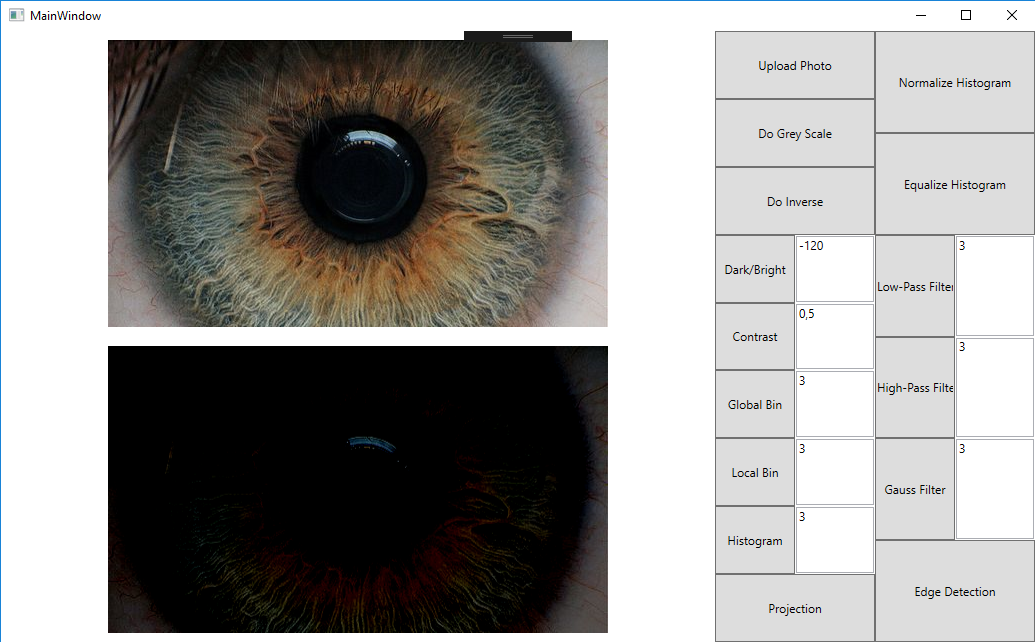
## Teoria

Dla każdego koloru możemy określić jego jaśniejszy lub ciemniejszy odcień. Wynika to z tego że każdy kolor może być na 255 poziomów jasny lub ciemny. Przyjrzyjmy się dwóm kolorom : czarnemu i białemu. W reprezentacji RGB biały jest opisany jako (255,255,255). To znaczy że niebieski, zielony i czerwony są najjaśniejsze. Z kolei dla koloru czarnego (0,0,0) każda ze składowych jest zerowa, czyli mówiąc potocznie – nie ma żadnego światła. Jeżeli weźmiemy dowolny kolor czerwony (150,0,0) to możemy zauważyć, że kolor (200,0,0) zawiera więcej czerwonego, zatem czerwony świeci bardziej, a z tego wynika że czerwony będzie jaśniejszy.

## Implementacja

W programie mamy do czynienia z kolorami o różnych wartościach. Zatem użytkownik może kazać programowi rozjaśnić obraz podając mu liczbę z przedziału [-255, 255]. Dzięki temu nawet kolor biały (255,255,255) może stać się czarny (0,0,0) lub odwrotnie. Dzieje się tak ponieważ dla każdego pixela program dodaje do każdej wartości składowej koloru wartość wybraną przez użytkownika.

## ScreenShot



# Kontrast

## Teoria

Kontrast jest określany potocznie jako intensywność różnic pomiędzy kolorami. To znaczy, dla obrazu z dużym kontrastem obraz jest jaskrawy, pełen odwrotnych kolorów (np. czerń i biel). Z kolei obraz o mały kontraście jest bardziej szary, ponieważ jego kolory są do siebie podobne.

## Implementacja

Program dla każdej składowej każdego pixela oblicza nową wartość według wzoru

Widzimy tutaj że dla współczynnika kontrastu równego 1, zachowamy starą wartość koloru. Natomiast im bardziej zmieniamy współczynnik w jedna lub drugą stronę, tym bardziej kolory będą się oddalać lub przybliżać do wartości środkowej (127).

## ScreenShot

