

Precyzyjna korekcja gammy, jasności i kontrastu

Autorzy : Krzysztof Hrycalik, Aleksandra Rojek, Karol Śmiałek

Opis projektu

Celem naszego projektu było napisanie programu, który w sposób bardzo dokładny (unikający błędów zaokrągleń) pozwala zmieniać korekcję gamma, jasność oraz kontrast obrazu. Program miał pozwolić na ciągłą obserwację wyników swojej pracy.

Wszystkie funkcje programu zostały zrealizowane na zasadzie wzajemnej współpracy. Ponadto jednym z wymagań projektu było zaimplementowanie histogramu, czym zajął się Krzysztof, podczas gdy Karol zajmował się testami, a Aleksandra dokumentacją.

Założenia wstępne przyjęte w realizacji projektu

Program posiada dwa tryby pracy, które można zmienić w dowolnym momencie. W pierwszym trybie program prezentuje miniaturkę całego zdjęcia, w drugim – wycinek zdjęcia, który można wybrać za pomocą suwaków. Niezależnie od wybranego trybu jakakolwiek zmiana parametru natychmiast uwidacznia się na zdjęciu. Program pozwala również na ustawianie korekcji gamma, jasności oraz kontrastu obrazu dla każdego z kanałów RGB osobno. Każdy suwak przyjmuje wartości 0-100%, co 0.01%. Odpowiada to dla: kontrastu $[0, \infty]$, gamma $[e^{-5}, e^5]$, jasności, R, G oraz B $[-255, 255]$. Składowe kolorów są przechowywane w pamięci komputera jako liczby zmiennoprzecinkowe z przedziału 0.0 do 255.0. Na ekranie widoczny jest histogram obrazujący jasność pixeli, który uaktualnia się tylko wtedy, gdy użytkownik sobie tego zażyczy, klikając na przycisk „Rysuj histogram”.

Analiza projektu

Specyfikacja danych wejściowych

Na początku programu wybieramy zdjęcie w formacie *jpg*, na którym będziemy wykonywać operacje.

Opis oczekiwanych danych wyjściowych

W dowolnym momencie możemy zapisać efekt wykonanych operacji jako zdjęcie w formacie *jpg*, *bmp* lub *png*.

Zdefiniowanie struktur danych

Oryginał zdjęcia, kopia, miniaturka oraz tło histogramu są przechowywane w *wxImage*. Dane potrzebne do narysowania histogramu są przechowywane w tablicy dynamicznej typu *float*.

Specyfikacja interfejsu użytkownika

Użytkownik ma możliwość wyboru zdjęcia na jakim będą wykonywane operacje. Cały panel narzędzi znajduje się po lewej stronie okna. Za pomocą suwaków użytkownik może ustalić zakres gammy, jasności oraz kontrastu dla każdego z kanałów RGB osobno. Poniżej znajduje się przycisk, który pozwala na pracę na wycinku obrazu, który z kolei możemy wybrać za pomocą suwaków (ustawiając składową X oraz Y dla wycinka). Kolejno widzimy przycisk, który udostępnia możliwość narysowania histogramu. Na samym dole znajduje się przycisk, udostępniający opcję zapisu zdjęcia.

Wyodrębnienie i zdefiniowanie zadań

Projekt można podzielić zasadniczo na 6 głównych modułów : wczytanie obrazka, korekcja obrazka, wyrysowanie zmienionej miniaturki lub wycinka, aktualizacja danych do histogramu, zapis zmodyfikowanego obrazka do pliku. Każdy z modułów ma swoje zadania. Wczytanie obrazka odpowiada za pobranie od użytkownika zdjęcia, a następnie zapisanie go do zmiennej, która przechowuje go w oryginalnej wersji. Korekcja obrazka zajmuje się pobraniem wartości z suwaków, a następnie zapisaniem ich do danych obrazka ze zmienionymi już wartościami kontrastu, gamma, jasności oraz kanałów R, G, B. Kolejny moduł czyli wyrysowanie miniaturki lub wycinka powoduje zamieszczenie po korekcjach obrazka na odpowiednim panelu. Moduł zajmujący się aktualizacją danych do histogramu uaktualniany jest, kiedy użytkownik sobie tego zażyczy klikając w odpowiedni przycisk. Ostatnim modułem jest zapisywanie obrazka z naniesionymi już korekcjami w takim samym rozmiarze jak obrazek wejściowy, dzieje się to również za pomocą kliknięcia w odpowiedni przycisk.

Decyzja o wyborze narzędzi programistycznych

Projekt został napisany w języku C++ w programie *Visual Studio 2019*. Do kompilacji został użyty kompilator *Local Windows Debugger*. Postanowiliśmy użyć biblioteki *wxWidgets*, gdyż wspiera wygodny w budowaniu aplikacji *wxFormBuilder* oraz posiada szereg przydatnych struktur danych oraz funkcji.

Podział pracy i analiza czasowa

Wspólnie wykonaliśmy większą część projektu. Pierwszym krokiem było stworzenie GUI za pomocą *wxFormBuilder*. Była to ważna część projektu, ponieważ odpowiednie zbudowanie interfejsu wpływa na przyjemność pracy użytkownika w danym programie. Następnie rozpoczęliśmy kodowanie. Po zaimplementowaniu podstawowej funkcji wczytania obrazka zajęliśmy się kolejno jasnością, kontrastem oraz korekcją gamma. Kolejnym krokiem było zapewnienie możliwości wyboru wycinka oraz pracy na nim. Stworzeniem funkcjonalności histogramu zajął się Krzysztof, testami programu - Karol, natomiast dokumentacją - Aleksandra.

Opracowanie i opis niezbędnych algorytmów

Algorytm zmiany jasności:

```
For(i=0;i<x.size()*y.size()*3;i++)  
{  
    Pixel[i]=Pixel[i]+jasność;  
}
```

Algorytm zmiany gamma:

```
For(i=0;i<x.size()*y.size()*3;i++)  
{  
    Pixel[i]= pow((Pixel[i]/255),gamma)*255;  
}
```

Algorytm zmiany kontrastu:

```
For(i=0;i<x.size()*y.size()*3;i++)  
{  
    Pixel[i]= (Pixel[i] -128)*kontrast +128  
}
```

Algorytm zmiany kanału R,G,B:

```
For(i=0;i<x.size()*y.size();i++)  
{  
    Pixel[i*3 + 0]= Pixel[i]+R;  
    Pixel[i*3 + 1]= Pixel[i]+G;  
    Pixel[i*3 + 2]= Pixel[i]+B;  
}
```

Algorytm tworzenia wycinka:

```
Wykorzystaliśmy gotową funkcję:  
Rectangle rectangle(UpperLeft_Corner,x_size,y_size);  
Image.GetSubImage(rectangle);
```

Algorytm tworzenia histogramu:

```
Pobranie danych z obrazka  
Wyrysowanie osi układu  
Naniesienie danych przy użyciu pętli for oraz odpowiednich funkcji
```

Kodowanie

Program składa się z 5 plików :

main.cpp – pozwala uruchomić program.

GUI.h ; *GUI.cpp* – odpowiada za interfejs programu.

GUIMyFrame1.h ; *GUIMyFrame1.cpp* – w tych plikach znajduje się cała implementacja funkcji programu.

Opis Klas :

MyApp – klasa niezbędna do uruchomienia programu.

GUI – klasa wygenerowana przez program *WxFormBuilder*, pozwalająca na utworzenie interfejsu.

GUIMyFrame – klasa odpowiedzialna za wszystkie operacje wykonywane w programie.

Opis Funkcji :

refresh (*wxClientDC* (*wxPanel*, który ma zostać zaktualizowany), *int* - mówi, który *wxImage* ma zostać wyrysowany) – pozwala na aktualizację danych na zdjęciu.

drawAxis (*wxClientDC* (*wxPanel*, na którym zostaną wyrysowane osie)) – rysuje osie wykresu.

drawLine2d (*wxClientDC* (*wxPanel*, na którym zostaje wyrysowana linia), *double* -współrzędne początku i końca rysowanej linii) – rysuje linię pomiędzy dwoma punktami.

drawValue (*wxClientDC* (*wxPanel*, na którym zostaje wyrysowana wartość na wykresie), *int* - współrzędna x odpowiadająca jasności pixela, *double* - współrzędna y odpowiadająca ilości pixeli o danej jasności) – rysuje słupek odpowiadający wartości na histogramie.

onUpdateUI (*wxUpdateUIEvent* - jakakolwiek zmiana w GUI powoduje wywołanie funkcji) – funkcja zajmująca się inicjalizacją zmiennych, ustawianiem parametrów jasności, kontrastu i gammy. Pozwala na pracę na wycinku oraz miniaturce. Zajmuje się również wyrysowaniem histogramu.

LoadImg (*wxCommandEvent* - powiązany jest z przyciskiem „Wczytaj obrazek”, który powoduje wywołanie funkcji) – pozwala na wczytanie zdjęcia do programu w formacie jpg.

Scrolls_Updated (*wxScrollEvent* – powiązane z ruchem suwaków) – aktualizuje dane wypisywane podczas przesuwania suwaków.

HisUpdate (*wxCommandEvent* - powiązany jest z przyciskiem „Rysuj histogram”, który powoduje wywołanie funkcji) – aktualizuje dane histogramu.

Save (*wxCommandEvent* - powiązany jest z przyciskiem „zapisz”, który powoduje wywołanie funkcji) – funkcja pozwalająca zapisać obrazek do pliku.

Opis Zmiennych:

*wxImage * ImgOrg* – wskaźnik przechowujący oryginał wczytanego zdjęcia.

*wxImage * ImgCpy* – wskaźnik przechowujący kopię wczytanego zdjęcia. Służy do zapisu zmienionego obrazka o takim samym rozmiarze jak obrazek wejściowy.

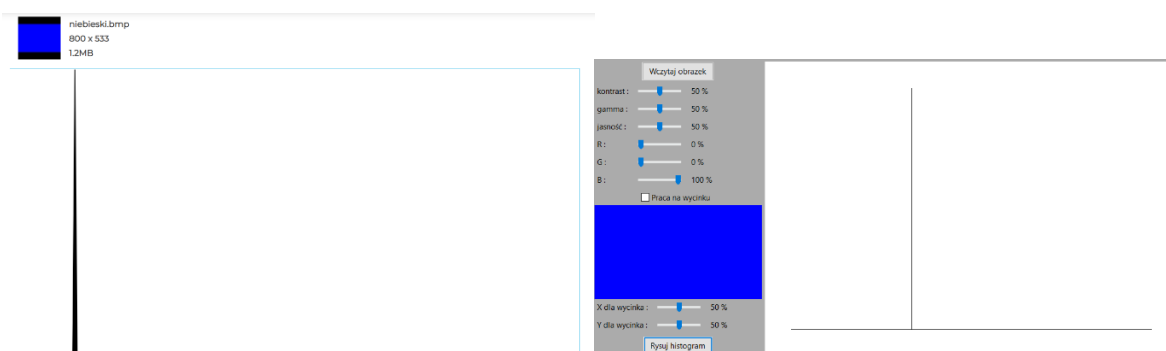
*wxImage * ImgCpym* – wskaźnik przechowujący kopię wczytanego zdjęcia. Służy do wyświetlania wycinka lub miniaturki obrazka po wykonanych korekcjach.

*wxImage * ImgHis* – wskaźnik przechowujący tło histogramu.

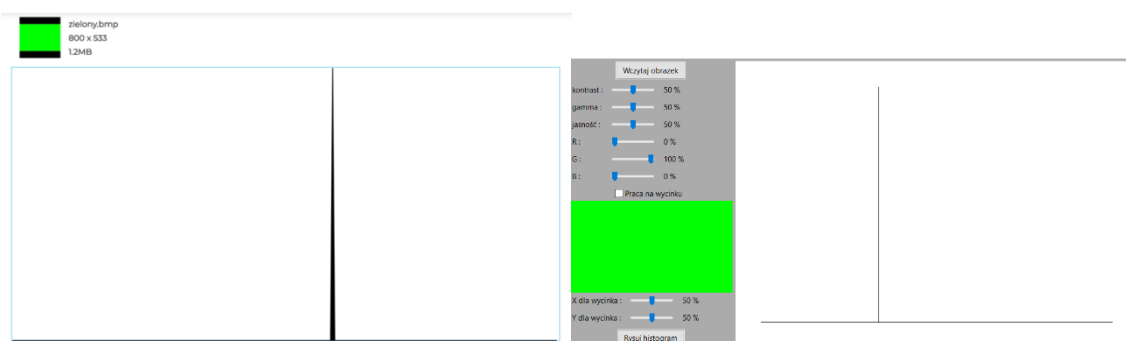
float _histab* – wskaźnik typu float, przechowujący dane na temat histogramu.

Testowanie

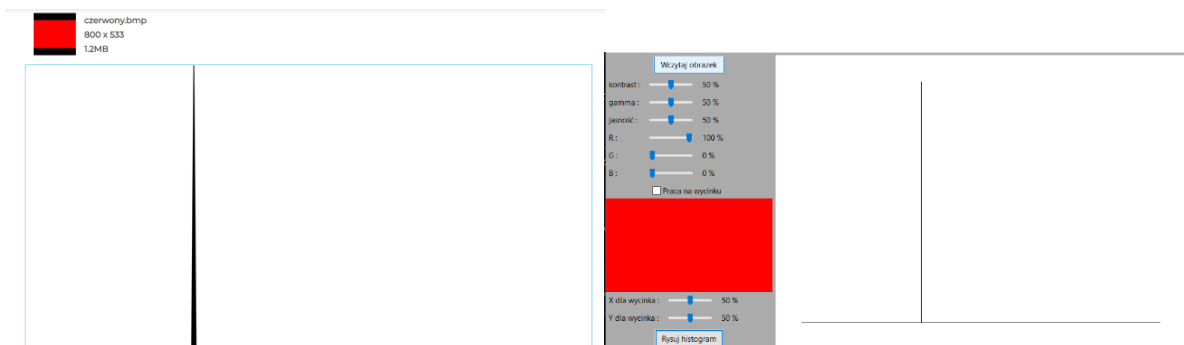
Działanie programu zostało przetestowane poprzez ręczne porównanie wyników operacji na zdjęciu z efektami uzyskanymi za pomocą programu *GIMP* oraz strony internetowej <https://www.sisik.eu/histo>.



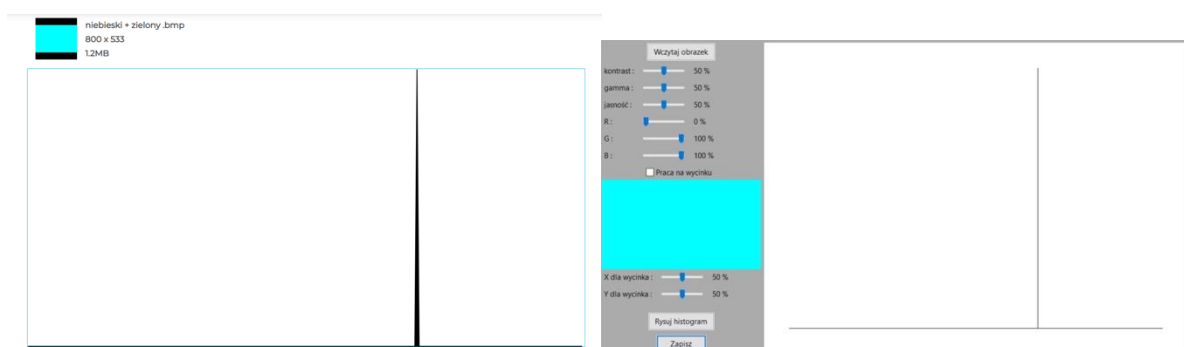
Rysunek 1. Korekcja na składowej B.



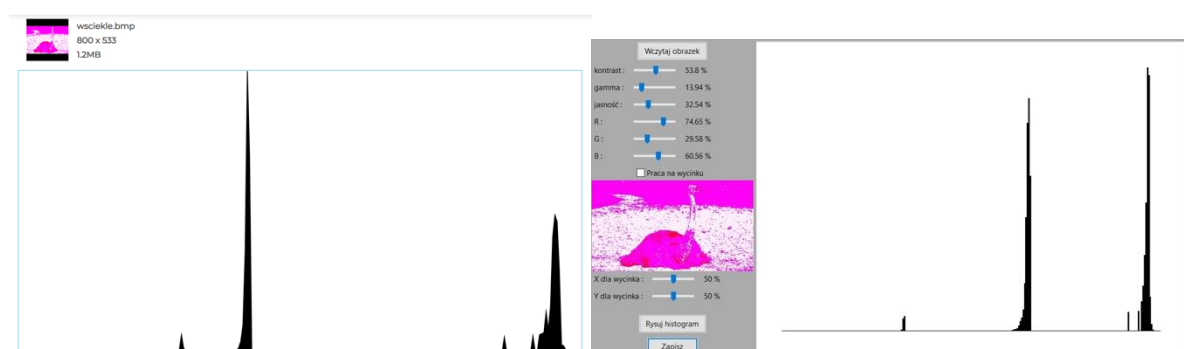
Rysunek 2. Korekcja na składowej G.



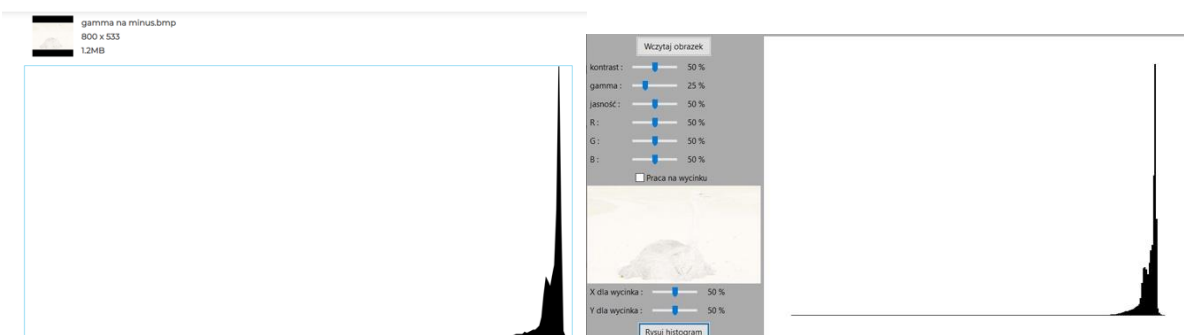
Rysunek 3. Korekcja na składowej R.



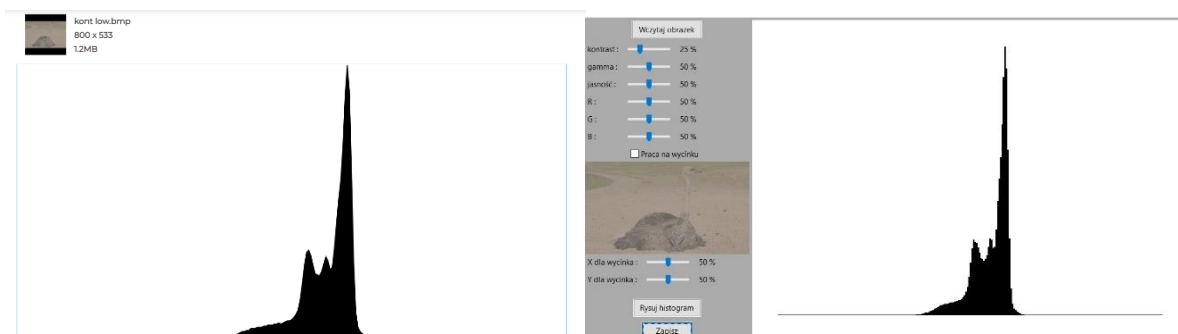
Rysunek 4. Korekcja na składowej G oraz B.



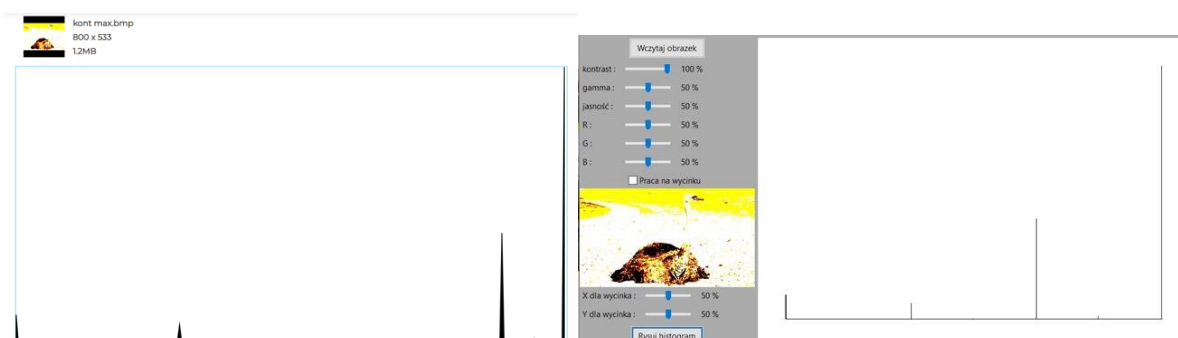
Rysunek 5. Korekcja na kanałach : kontrast, jasność, gamma, R, G, B.



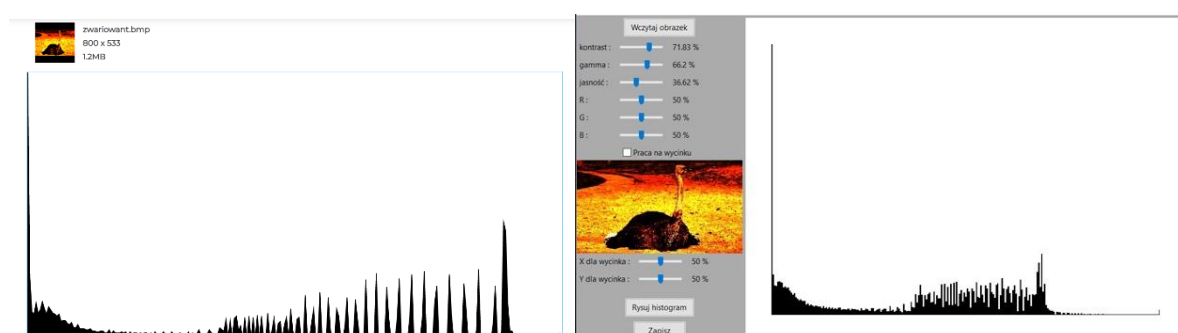
Rysunek 6. Korekcja na kanale gamma.



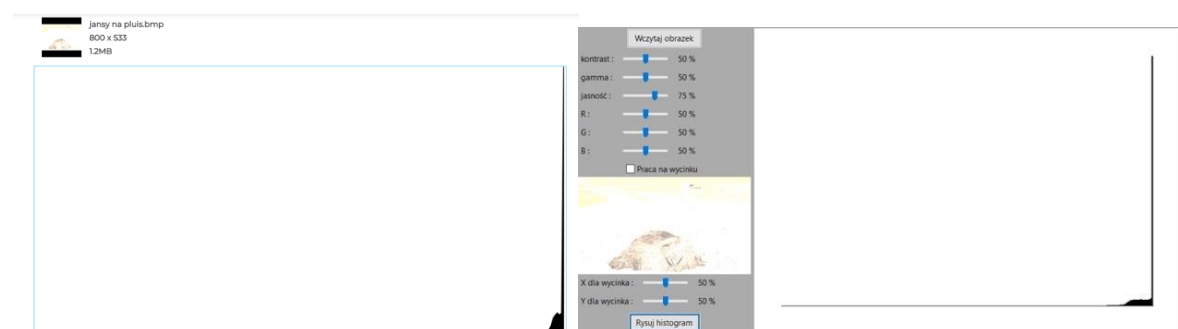
Rysunek 7. Korekcja na kanale kontrast.



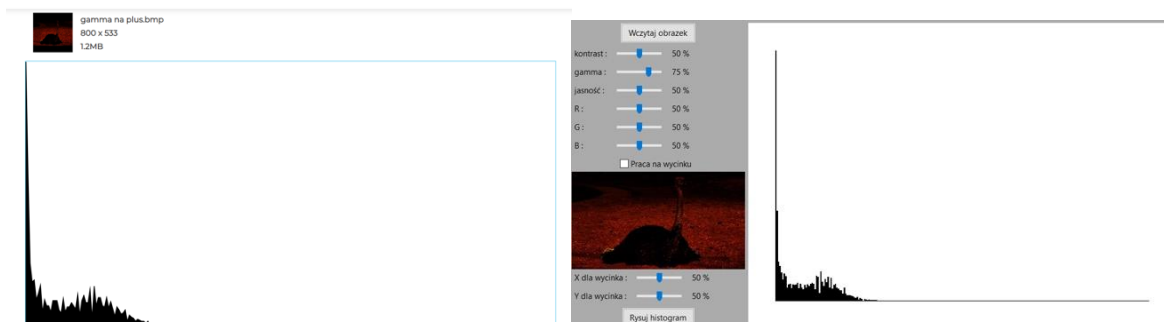
Rysunek 8. Korekcja na kanale kontrast.



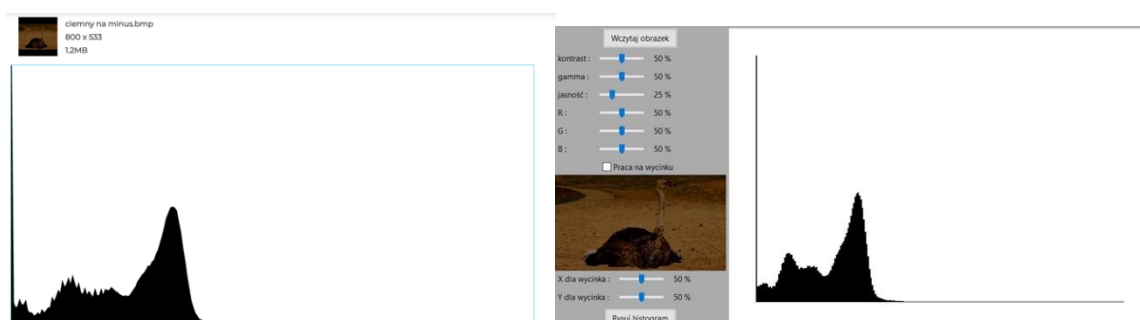
Rysunek 9. Korekcja na kanałach : kontrast, jasność, gamma.



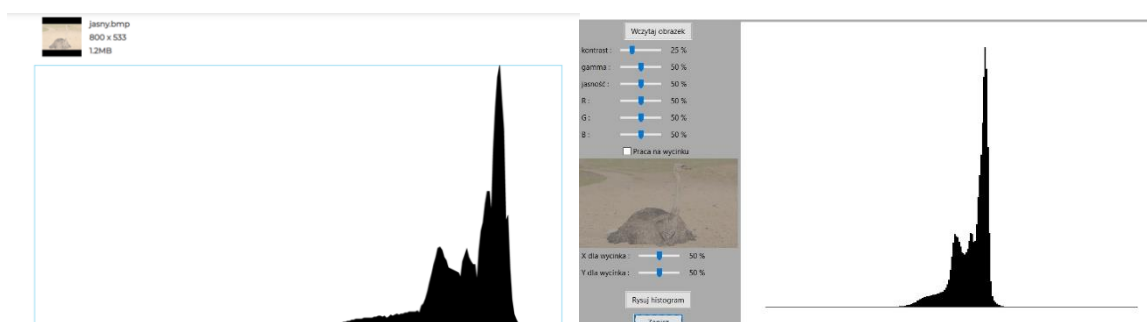
Rysunek 10. Korekcja na kanałach : kontrast, jasność, gamma.



Rysunek 11. Korekcja na kanale gamma.



Rysunek 12. Korekcja na kanale jasność.



Rysunek 13. Korekcja na kanałach : kontrast, jasność, gamma.

Wdrożenie, raport i wnioski

Program został uruchomiony z niezależnym zdjęciem, nieużyтым w trakcie projektowania i dał oczekiwane rezultaty.

Wszystkie wymagania podstawowe projektu zostały spełnione – posiada dwa tryby pracy (prezentacja miniaturki i wycinka), pozwala na ustawienie jasności, kontrastu i korekcji gamma dla każdego z kanałów RGB osobno, na ekranie widoczny jest histogram. Z wymagań rozszerzonych – udało się przejść na 32 bitową reprezentację zmiennoprzecinkową i na takich liczbach są dokonywane wszystkie obliczenia.

W przyszłości można udoskonalić program o niewykonane zadania z części rozszerzonej - wykonanie histogramu, który jest liczony dla miniaturki oraz możliwość edycji jasności i

kontrastu poprzez wybór przedziału na histogramie. Są to elementy, które mogą poszerzyć funkcjonalność programu i zapewnić większe możliwości potencjalnemu użytkownikowi. Do poprawy pozostaje również kwestia optymalizacji, aby w przyszłości program mógł działać szybciej i płynniej.