

SVD, czyli
rozkład według wartości osobliwych
na podstawie kompresji
obrazów

Autorzy:
Krzysztof Jędrejasz,
Kacper Kupiszewski,
Kamil Maczuga,

1. Cel projektu

Projekt miał na celu zaimplementowanie algorytmu SVD w oparciu o metody istniejące w bibliotece scikitlearn. Wizualizacja jak i przykład zastosowania działania algorytmu został zrealizowany na podstawie kompresji oraz dekompresji obrazów cyfrowych czarno-białych jak również i kolorowych w formacie ".jpg". Do przedstawienia wyników analizy oraz wizualizacji uzyskanych parametrów dołączona została biblioteka matplotlib. Ostatecznym rezultatem miał być program który kompresuje wyznaczony obraz z bazy danych scikitlearn'a oraz analizuje właściwości kompresji algorytmu SVD za pomocą szeregu kryteriów ocen przy stosowaniu różnych stopni kompresji.

2. Aspekty techniczne.

Projekt wykonany został w środowisku PyCharm w języku python w wersji 3.8.

Wykorzystane biblioteki:

- **PIL**, w celu prostego odtwarzania zdjęć z macierzy,
- **Sklearn**, jako baza zdjęć oraz właściwe narzędzie do kompresji/dekompresji SVD,
- **Skimage**, dla szeregu funkcji analizujących jakość kompresji/dekompresji,
- **numpy**, narzędzie do operacji na macierzach,
- **matplotlib**, dla czytelnej wizualizacji wyników,
- **os**, wykorzystana do badania zajmowanej pamięci plików/zdjęć,

Jako funkcję kompensacyjną SVD z biblioteki **Sklearn** użyto Truncated SVD jako transformator przeprowadzający redukcję wymiarowości liniowej macierzy za pomocą dekompozycji na wartości osobliwe.

3.Omówienie prgramów.

Całość projektu składa się na cztery oddzielne programy; dwa dla analizy zdjęć czarno-białych oraz dwa do RGB. Poniżej przedstawiony jest krótki opis tych skryptów.

a)Simple_Mono.

Pierwsza część programu zobowiązuje do wyboru ustawień konwersji. Do tego używane są zmienne: `Max_conversion_rate` oraz `Picture_choice`. Pierwsza zmienna służy jako ustawienie docelowego rozmiaru macierzy kompresji, natomiast druga pozwala na dość skromny dobór zdjęcia do analizy.

W kolejnej części z pobranej macierzy odtwarzany jest oryginalny obraz w wersji czarno-białej, a następnie jest analizowany pod względem zajmowanej pamięci oraz jego wymiarów.

Zmienna "Range" wyznacza wymiar macierzy kompresji. Jest ona zabezpieczona przed różnymi typami wymiarów zdjęcia, gdyż macierz kompresji nie może być równa wymiarowi oryginalnego zdjęcia (wymagania funkcji `decomposition.TruncatedSVD`), oraz macierz po kompresji jest zawsze kwadratowa o maksymalnych wymiarach najkrótszego boku zdjęcia oryginalnego; większa zmienna "Range" spowoduje błąd albo wynik nie będzie się różnił niczym od optymalnych wartości przy ustawieniu `Max_conversion_rate = 100`.

Po ustawieniu docelowego rozmiaru macierzy kompresji używane zostają funkcje: `svd.fit_transform` oraz `svd.inverse_transform`.

Powodują one skompresowanie oraz skompresowanie wybranego zdjęcia.

Gdy uzyskamy odpowiednie macierze obliczamy wartość błędów oraz wskaźniki podobieństwa między zdekompresowanym a oryginalnym zdjęciem. Używamy do tego kryteriów: **PSNR** - Peak signal-to-noise ratio, **SSIM** - structural similarity index measure, **MSE** - Medium Square Error, **VR** - Variance Ratio.

Odtwarzamy z macierzy kompresji i dekompresji zdjęcia. Mierzymy wartość zajmowanej pamięci i porównujemy z oryginałem otrzymując stosunek procentowy.

Finalnie wyświetlamy wszystkie wyniki oraz parametry w okienku.

b)Mono.

W kolejnym programie zasada działania jest identyczna jak w poprzednim programie z wyjątkiem dodatku pętli 'for' i zmiany zapisu danych do list. Program oblicza w pętli kolejne stopnie kompresji poczynając od największej, a kończąc na ustawionej przez użytkownika. Pozwala to zobrazować zmiany jakości kompresji i zajmowanej pamięci w przedziale zmieniającego się wymiaru macierzy kompresji.

c)Simple_RGB.

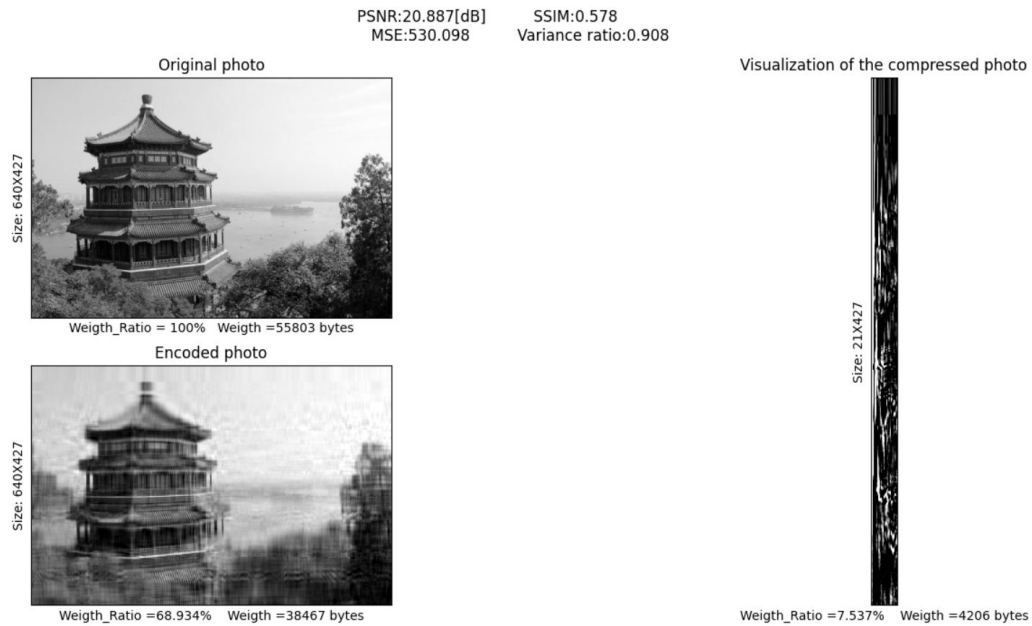
W skrypcie w dostosowano program pod analizę zdjęć kolorowych. W celu ich właściwej kompresji dzielono macierz obrazu na trzy składowe R, G, B, w których są zapisane odcienie poszczególnych kolorów, a następnie skompresowano każdą macierz oddzielnie. Po dekompresji scalono trzy macierze w jedną uzyskując właściwy wynik. Program daje dodatkowo możliwość prostej normalizacji wyników albowiem przy słabszej kompresji pojawiają się anomalie obliczeniowe powodujące wyraźne zniekształcenie obrazu wyjściowego (wartości macierzy wychodzą poza zakres 0-255 uint8 powodując problem z odczytaniem obrazu).

d)RGB.

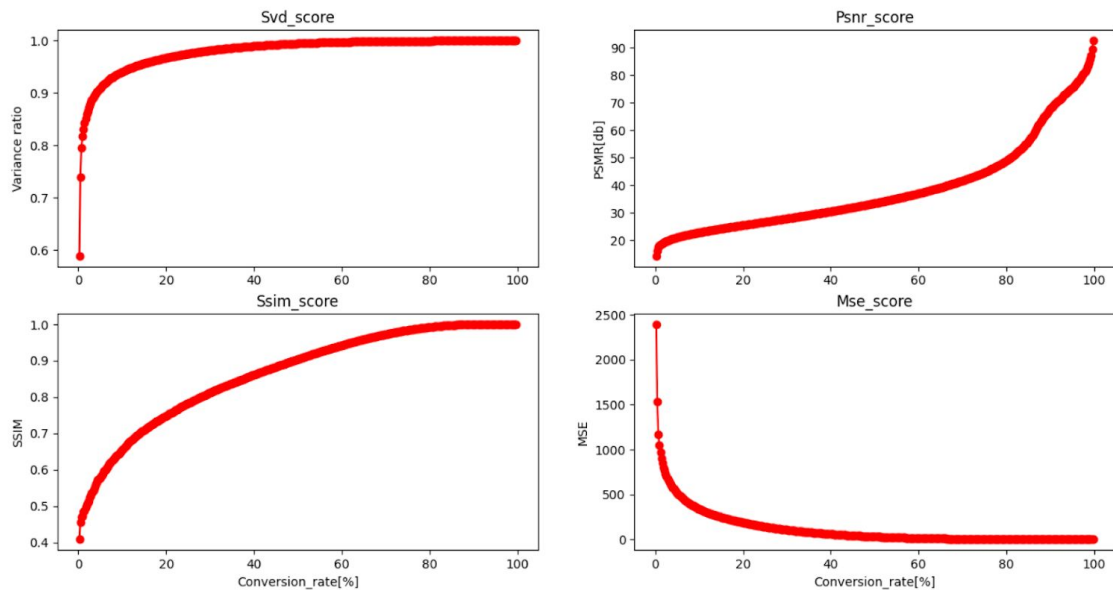
Jak w przypadku skryptu "Mono" program pozwala zaobserwować zmiany współczynników kompresji przy zmianie stopnia kompresji z dodatkiem funkcji normalizacji.

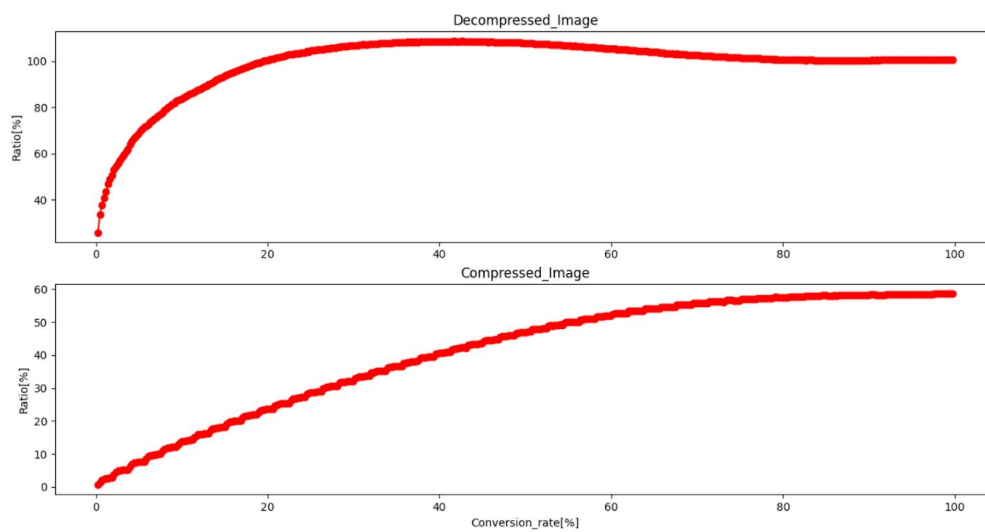
4. Testy oraz wnioski.

Poniżej przedstawiony efekt działania kolejnych skryptów:



Simple_mono z kompresją do pięciu % wymiarów oryginału.





Mono z ukazaniem pełnego spektrum kompresji.

PSNR:18.184[dB]
MSE:987.867

SSIM:0.565
Variance ratio:0.937

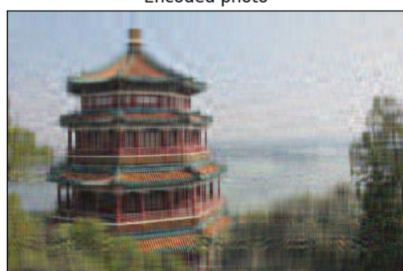
Original photo



Size: 640X427

Weigth_Ratio = 100% Weigth =61147 bytes

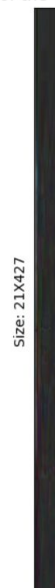
Encoded photo



Size: 640X427

Weigth_Ratio =59.861% Weigth =36603 bytes

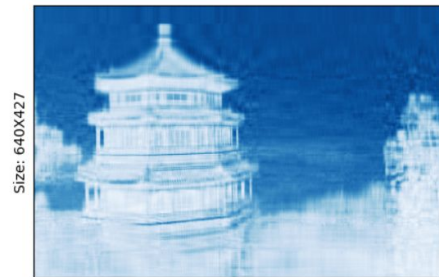
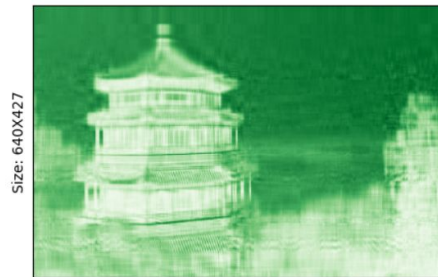
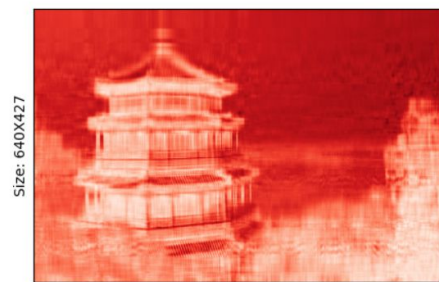
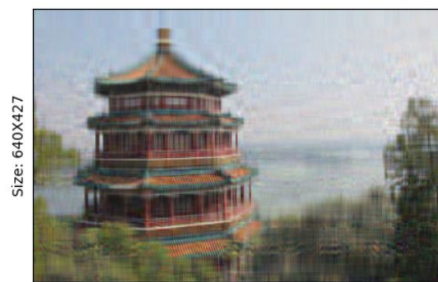
Visualization of the compressed photo



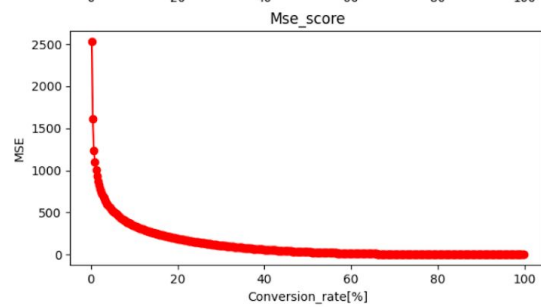
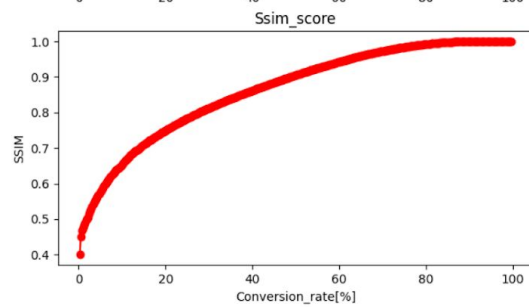
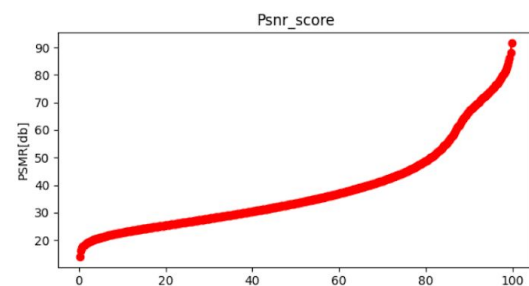
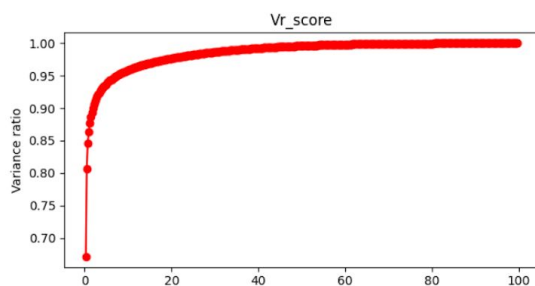
Size: 21X427

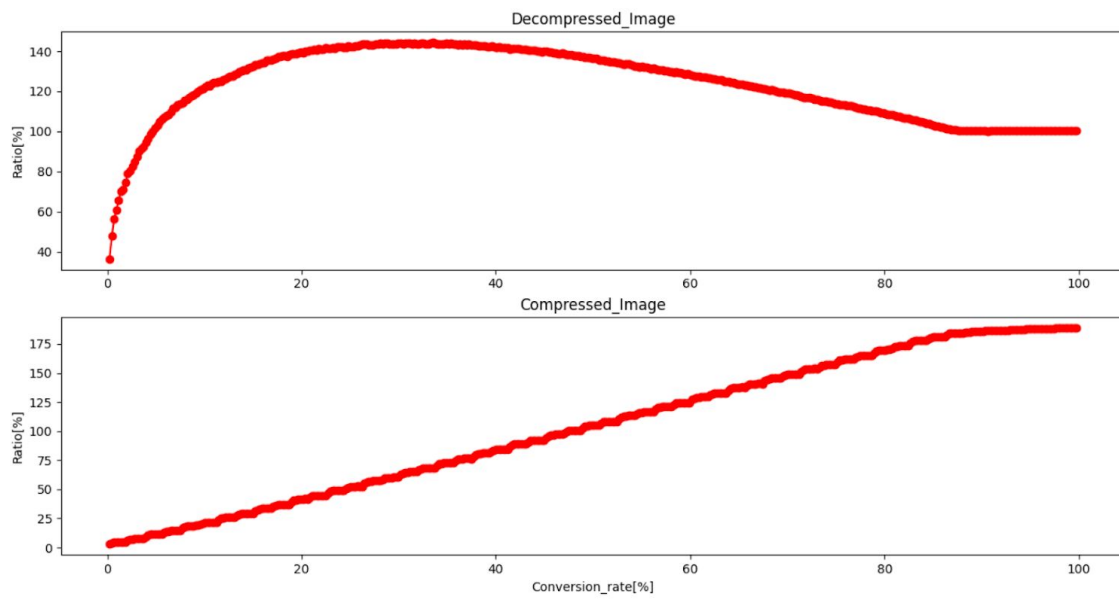
Weigth_Ratio =3.07% Weigth =1877 bytes

RGB distribution

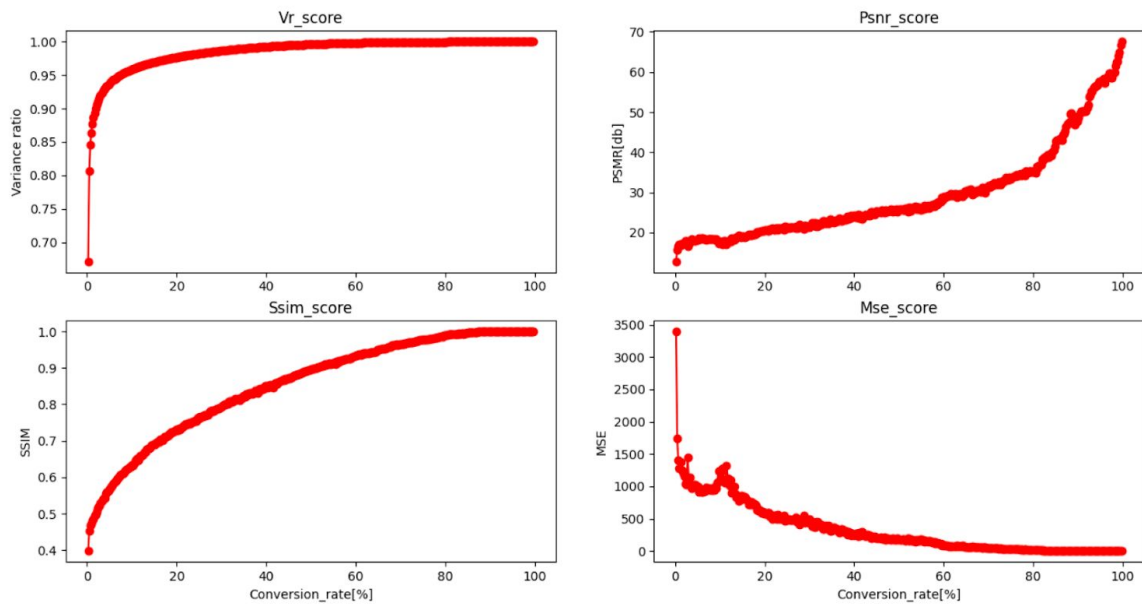


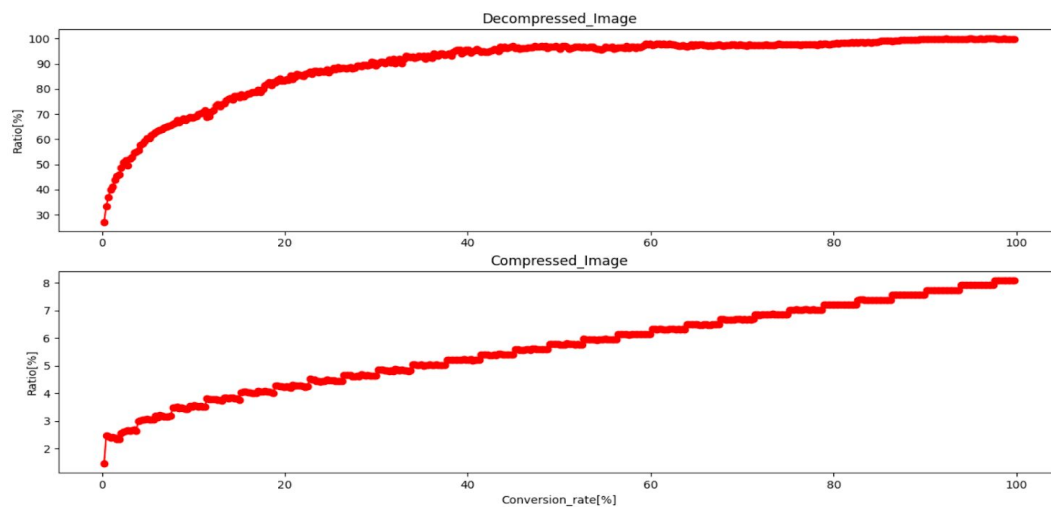
RGB_Simple z dodatkowym ukazaniem macierzy kolorów,



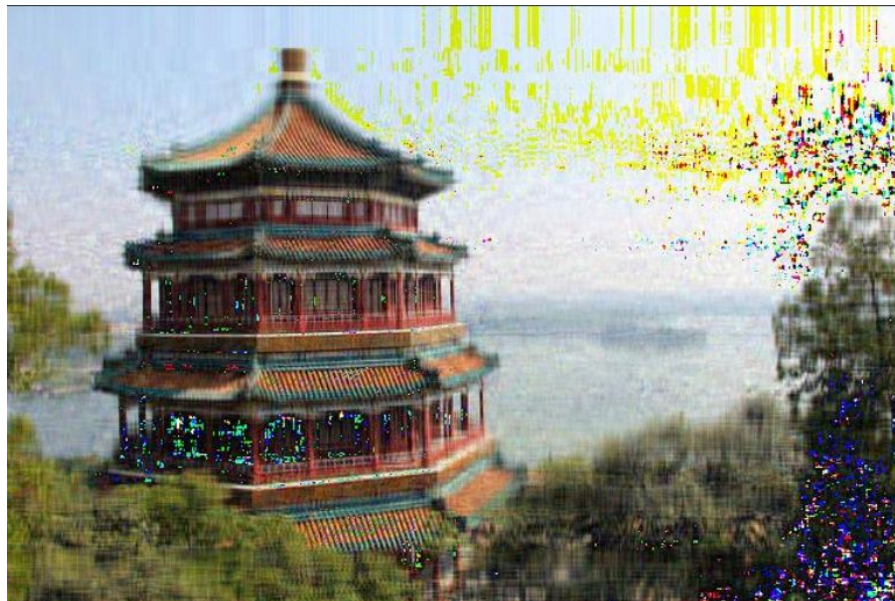


RGB spektrum bez uwzględnienia normalizacji.





RGB spektrum z uwzględnienia normalizacji.



Przykład braku użycia normalizacji przy odtwarzaniu zdjęć RGB.

Podsumowując:

Obrazy są w zasadzie macierzową reprezentacją pikseli. Zdjęcia czarno-białe składają się z jednej macierzy za to kolorowe zdjęcia składają się z trzech odcieni/macierzy. Zmniejszając takowe macierze uzyskujemy właściwą kompresję. Zauważyliśmy że, stopień kompresji n (który decyduje o wielkości macierzy kompresji) jest związany bezpośrednio z jakością obrazu oraz rozmiarem skompresowanego pliku. Oznacza to, że im mniejsza wartość n , tym większe zmniejszenie rozmiaru pliku, jednocześnie powodując pogorszenie jakości.

Optymalny stopień kompresji uzyskamy, gdy macierz kompresji jest wybrana tak, aby uzyskać optymalną redukcję rozmiaru pliku przy zachowaniu jakości obrazu. Do doboru najlepszych wyników, posłużyć się można odpowiednimi parametrami takimi jak: ssim, psnr, mse itd. przy jednoczesnej obserwacji zajmowanej pamięci skompresowanego zdjęcia.

Dodatkowo potrzeba dla tego algorytmu dostosować ewentualną normalizację, która utrzyma wyniki w odpowiedniej odczytywanej skali, a tym samym zoptymalizuje zajmowaną pamięć plików dla specyficznych matryc zdjęć.