

Uwierzytelnianie oraz rekonstrukcja obrazów cyfrowych JPEG (semestr letni 2013/2014)

Klaudia Popko 259027, Dawid Rymarczyk 259032

I. WPROWADZENIE

TEN projekt ma na celu zakodowanie informacji o obrazie w postaci znaku wodnego w tym samym obrazie wybranym przez użytkownika aplikacji, a następnie samozrekonstruowanie się tego obrazu. Projekt zakładał porównanie ze sobą efektywności dwóch algorytmów oraz porównanie i uodpornienie techniki kodowania znaku wodnego na zmiany jasności oraz rozszerzenie ich na obrazy nie tylko czarno-białe, ale i kolorowe.

II. INSTRUKCJA KONFIGURACJI I UŻYTKOWANIA

Aby uruchomić aplikację należy uruchomić program „gui” za pomocą komendy run w MatLabie. Po uruchomieniu aplikacji pojawia się okno, które podczas testowania algorytmu Chaddad’a, podzielone jest na dwie części. Pierwsza część pozwala programowi na pracę w trybie kodera a druga w trybie dekodera. Za pomocą przycisku “wybierz zdjęcie” należy wybrać plik graficzny. Następnie za pomocą suwaka określamy krok kwantyzacji, minimalny to 6, a maksymalny to 26. W kolejnym kroku dla funkcji kodera należy wybrać typ kompresji jakiej plik wynikowy ma zostać zapisany. W polu tekstowym i w pierwszej jak i w drugiej części okna „Podaj nazwę” należy wpisać nazwę pod jaką oczekuje się zapisania obrazu wynikowego, odpowiednio zakodowanego, czy zdekodowanego. Przycisk „Zapisz obraz” służy do wykonania funkcji kodującej, natomiast przycisk „Odzyskaj obraz” pozwala na odzyskanie zakodowanego payload’u oraz rekonstrukcję obrazu. W chwili gdy kodowanie/rekonstrukcja się zakończy, aktywnym stanie się przycisk „Pokaż obraz”, który pozwoli nam jak wygląda obraz z zakodowanym nadmiarem/zrekonstruowany. W przypadku, gdy nie zostanie wybrany plik, okno do wybrania pliku uruchomi się automatycznie. Natomiast w przypadku, gdy nie zostanie wpisana nazwa pliku, wynik działania programu zostanie zapisany pod nazwą „Obraz.jpg”. Aplikacja domyślnie ma ustawiony krok kwantyzacji na 6 i zapisuje obrazy w formacie JPG z jakością na poziomie 80.

III. WYNIKI TESTÓW

Testy zostały przeprowadzone na niżej zamieszczonych zdjęciach. Przedstawiają one kolejno: zdjęcie oryginalne, zdjęcie z zakodowanym znakiem wodnym, zdjęcie, które zostało uszkodzone, zdjęcie zrekonstruowane. Wybrany krok kwantyzacji to 14.



Obraz oryginalny.



Obraz z zakodowanym znakiem wodnym z kompresją JPG Jakość 100.



Uszkodzony obraz



Obraz zrekonstruowany.

Jak można zauważyć to co zostało odzyskane z obrazka pozwala jednoznacznie zobaczyć, czy jakieś elementy widoczne na zdjęciu zostały zmienione, czy też nie. Niestety jakość samorekonstrukcji nie jest aż tak dobra jak jakość oryginału. Im większy krok kwantyzacji, tym większa jest ingerencja w obraz oryginalny, w który kodujemy znak wodny, przez co zmiany w zakodowanym obrazie są bardziej widoczne dla ludzkiego oka.

Zostały również przeprowadzone testy tego algorytmu dla różnych stopni kompresji, przykładowe wyniki dla kompresji odpowiednio 80, 90 i 100 są przedstawione poniżej w postaci tylko obrazów zrekonstruowanych. Kompresja na poziomie niższym niż 80 jest nieefektywna i obraz zrekonstruowany ciężko oddzielić od szumu.



Obraz odzyskany z obrazu o kompresji z jakością na poziomie 80

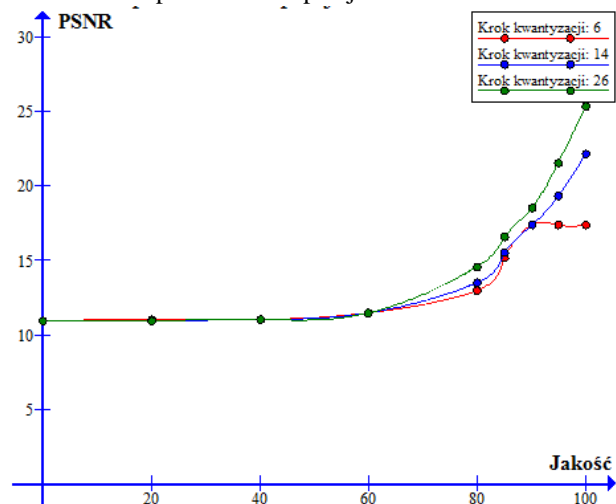


Obraz zrekonstruowany z obrazu skompresowanego z poziomem jakości 90.



Obraz zrekonstruowany z obrazu skompresowanego z poziomem jakości 100.

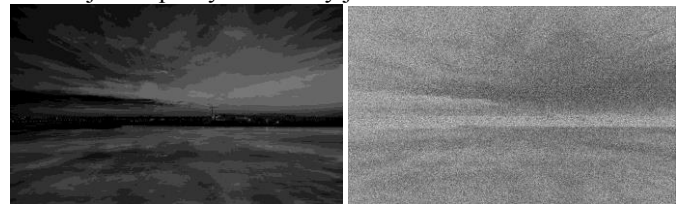
Poniżej został zaprezentowany wykres dla kilku przykładowych kroków kwantyzacji funkcji wartości PSNR w zależności od poziomu kompresji obrazu.



Wykres zależności PSNR do jakości kompresji obrazu.

Jak wynika z wykresu im wyższy krok kwantyzacji tym łatwiej odzyskać zakodowany znak wodny oraz im wyższa jakość kompresji tym lepsza jest też jakość odzyskiwanego obrazu.

Został przeprowadzony jeszcze jeden test, test odporności na zmianę jasności, jak widać na poniższym obrazie algorytm ten nie jest odporny na zmiany jasności obrazu.



Obraz ze zmienioną jasnością i jego rekonstrukcja

IV. PODSUMOWANIE

Udało się zaimplementować pierwszy z algorytmów i przetestować jego działanie na obraz czarno-białych. Co więcej zostały przeprowadzone testy funkcjonowania tego algorytmu w zależności od różnych stopni kompresji. Za tę funkcjonalność odpowiedzialni byli Dawid Rymarczyk i Klaudia Popko. Za przygotowanie funkcjonalnego interfejsu użytkownika odpowiedzialna była Klaudia Popko, a za oprogramowanie go był odpowiedzialny Dawid Rymarczyk. Nie udało się zaimplementować drugiego z algorytmów oraz wykonać odporności tych algorytmów na zmiany jasności.