

Wprowadzenie

W ramach tego laboratorium zapoznaliśmy się z podstawami stratnej kompresji obrazu w standardzie JPEG. Zaimplementowaliśmy uproszczony potok kompresji i dekompresji w Pythonie, używając bibliotek NumPy, SciPy (dla DCT) i OpenCV (dla konwersji kolorów). Skupiliśmy się na kluczowych krokach: transformacji do przestrzeni YCbCr, opcjonalnym podpróbkowaniu chrominancji (4:4:4 i 4:2:2), podziale na bloki 8x8, transformacji DCT, kwantyzacji (używając standardowych tabel JPEG lub tabeli jedynek) i skanowaniu zygzakowym. Celem było zrozumienie działania tych etapów i zaobserwowanie ich wpływu na jakość obrazu na wybranych fragmentach różnych zdjęć.

Zadania Zrealizowane

1. **Implementacja Algorytmu:** Napisaliśmy funkcje `compress_jpeg` i `decompress_jpeg` realizujące uproszczony potok JPEG. Implementacja obejmuje:

- Konwersję RGB \leftrightarrow YCbCr.
- Podprobkowanie chrominancji 4:4:4 (bez zmian) i 4:2:2 (poziome).
- Podział na bloki 8x8.
- Dwuwymiarowe DCT i IDCT na blokach (z centrowaniem danych +/- 128).
- Kwantyzację i dekwantyzację z użyciem tabel standardowych (QY_STD, QC_STD) lub tabeli jedynek (QN).
- Skanowanie Zigzag i odwrócone skanowanie Zigzag.
- Przechowywanie potrzebnych danych (skompresowane warstwy, kształt, ratio, tablice Q) w prostej klasie `JpegData`.

2. **Analiza Fragmentów Obrazów:**

- Wybraliśmy 4 różne obrazy testowe.
- Z każdego obrazu wycięliśmy po 4 fragmenty 128x128 pikseli, reprezentujące różne obszary (np. gładkie tło, krawędzie, tekstury, obszary o zmiennym kolorze/jasności).
- Dla każdego fragmentu przeprowadziliśmy kompresję i dekompresję, testując 4 kombinacje opcji:
- 4:4:4, kwantyzacja jedynekami (minimalna strata, referencja).
- 4:2:2, kwantyzacja jedynekami (wpływ samego subsamplingu).
- 4:4:4, kwantyzacja standardowa (wpływ kwantyzacji DCT).
- 4:2:2, kwantyzacja standardowa (pełny uproszczony JPEG).
- Wygenerowaliśmy obrazy porównawcze (oryginał vs. rekonstrukcja) dla każdej kombinacji i fragmentu, w tym oryginalne fragmenty dla łatwiejszego porównania.
- Opisałiśmy obserwacje i wyciągnęliśmy wnioski.

Implementacja – Kluczowe Kroki

Nasz uproszczony JPEG działa według następujących kroków:

1. Kompresja:

- Obraz wejściowy RGB jest konwertowany do przestrzeni YCbCr (jasność Y, różnice koloru Cb i Cr). Używamy do tego `cv2.cvtColor`.
- Opcjonalnie, warstwy chrominancji (Cb, Cr) są podpróbkowane w poziomie (tryb 4:2:2), redukując ich szerokość o połowę (`Cb = Cb[:, ::2]`). W trybie 4:4:4 ten krok jest pomijany.
- Każda warstwa (Y, Cb, Cr) jest dzielona na bloki 8x8 pikseli.
- Dla każdego bloku:
- Wartości pikseli są przesuwane o -128 (centrowanie wokół zera).
- Wyliczana jest transformata DCT (`scipy.fftpack.dct`).
- Współczynniki DCT są kwantyzowane przez dzielenie przez odpowiednią tablicę kwantyzacji (QY lub QC, standardową lub jedynkową) i zaokrąglanie do najbliższej liczby całkowitej.
- Skwantowane współczynniki są układane w wektor 1D za pomocą skanowania Zigzag.
- Wynikowe wektory dla każdej warstwy, wraz z metadanymi (oryginalny kształt, ratio, użyte tablice Q), są zapisywane w obiekcie `JpegData`.

2. Dekompresja:

- Z obiektu `JpegData` odczytywane są wektory Y, Cb, Cr i metadane.
- Dla każdego wektora (reprezentującego jeden blok):
- Wektor jest przekształcany z powrotem do macierzy 8x8 za pomocą odwrotnego skanowania Zigzag.
- Macierz jest dekwantyzowana przez mnożenie przez odpowiednią tablicę kwantyzacji (QY lub QC).
- Stosowana jest odwrotna transformata DCT (`scipy.fftpack.idct`).
- Do wyników dodawane jest 128 (powrót do zakresu jasności).
- Zrekonstruowane bloki 8x8 są składane w pełne warstwy Y, Cb, Cr.
- Jeśli użyto podpróbkowania 4:2:2, warstwy Cb i Cr są upsamplowane w poziomie (podwojenie kolumn przez `np.repeat`).
- Warstwy Y, Cb, Cr są łączone i konwertowane z powrotem do przestrzeni RGB (`cv2.cvtColor`), z przycięciem wartości do zakresu [0, 255].

Analiza Wyników i Obserwacje

Przeprowadziliśmy kompresję i dekompresję dla wybranych fragmentów obrazów, testując 4 warianty ustawień. Poniżej prezentujemy wybrane pary obrazów (oryginał fragmentu vs. rekonstrukcja), które dobrze ilustrują kluczowe efekty. Wszystkie wygenerowane obrazy

porównawcze oraz oryginalne fragmenty znajdują się w katalogu [jpeg_results/](#).

Przykład 1: Porównanie wpływu podpróbkowania chrominancji (Obraz 1, Fragment 1)

Fragment ten przedstawia [np. fragment nieba z delikatnymi chmurami]. Porównujemy warianty ze standardową kwantyzacją (**StdQ**), aby zobaczyć różnicę wprowadzaną przez samo podpróbkowanie 4:2:2.



Figure 1. Oryginalny fragment ([jpeg_results/1_Frag1_Original.png](#))

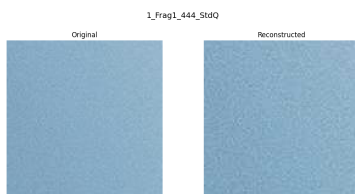


Figure 2. Rekonstrukcja: 4:4:4, StdQ ([jpeg_results/1_Frag1_444_StdQ.png](#))

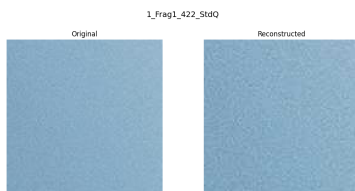


Figure 3. Rekonstrukcja: 4:2:2, StdQ ([jpeg_results/1_Frag1_422_StdQ.png](#))

- **Obserwacje (Obraz 1, Fragment 1):**
- Wariant **4:4:4, StdQ** (środkowy obraz) pokazuje artefakty typowe dla standardowej kwantyzacji, takie jak [np. lekkie rozmycie i delikatne artefakty blokowe]. Kolory są jednak dobrze odwzorowane.
- Wariant **4:2:2, StdQ** (prawy obraz), w porównaniu do **4:4:4, StdQ**, dodatkowo wprowadza [np. niewielkie "rozlanie" się kolorów na subtelnych przejściach tonalnych chmur, co jest efektem redukcji informacji o kolorze]. Artefakty kwantyzacji są porównywalne.

Przykład 2: Demonstracja najsilniejszych artefaktów (Obraz 1, Fragment 4)

Fragment ten charakteryzuje się [np. obszarem z drobnymi, kontrastowymi detalami i teksturowaną powierzchnią]. Porównujemy wersję referencyjną (minimalna strata) z wersją poddaną pełnej kompresji (podpróbkowanie i standardowa kwantyzacja).



Figure 4. Oryginalny fragment (jpeg_results/1_Frag4_Original.png)

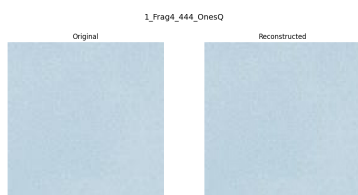


Figure 5. Rekonstrukcja: 4:4:4, OnesQ (jpeg_results/1_Frag4_444_OnesQ.png)

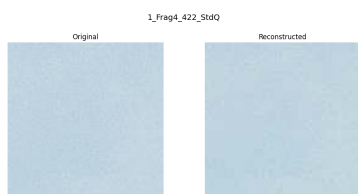


Figure 6. Rekonstrukcja: 4:2:2, StdQ (jpeg_results/1_Frag4_422_StdQ.png)

- **Obserwacje (Obraz 1, Fragment 4):**

- Wariant **4:4:4, OnesQ** (środkowy obraz) jest niemal identyczny z oryginałem, co potwierdza minimalne straty samych transformacji.
- Wariant **4:2:2, StdQ** (prawy obraz) prezentuje skumulowany efekt strat. Wyraźnie widać [np. znaczące rozmycie drobnych detali, pojawienie się artefaktów blokowych oraz "przeciekanie" kolorów na krawędziach kontrastowych elementów]. Jakość jest wyraźnie niższa od oryginału i wersji referencyjnej.

Przykład 3: Analiza wpływu kwantyzacji na obszar z teksturą (Obraz 1, Fragment 2)

Ten fragment zawiera [np. fragment ściany z cegieł lub innej powierzchni o wyraźnej, ale regularnej fakturze]. Sprawdzamy, jak standardowa kwantyzacja wpływa na odbiór takich obszarów, przy zachowaniu pełnej informacji o kolorze (4:4:4).



Figure 7. Oryginalny fragment (jpeg_results/1_Frag2_Original.png)

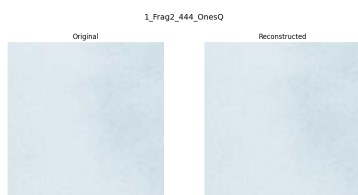


Figure 8. Rekonstrukcja: 4:4:4, OnesQ (jpeg_results/1_Frag2_444_OnesQ.png)

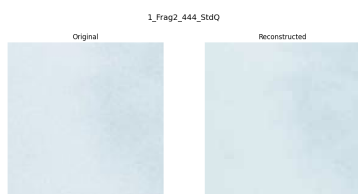


Figure 9. Rekonstrukcja: 4:4:4, StdQ (jpeg_results/1_Frag2_444_StdQ.png)

- **Obserwacje (Obraz 1, Fragment 2):**
- Wariant **4:4:4, OnesQ** (środkowy) dobrze oddaje oryginalną teksturę.
- Wariant **4:4:4, StdQ** (prawy) w porównaniu do **OnesQ** pokazuje, że [np. tekstura stała się bardziej "rozmyta", mniej ostra, a na krawędziach poszczególnych elementów tekstury mogą pojawić się delikatne artefakty blokowe]. Utrata detali wynikająca z kwantyzacji współczynników wysokich częstotliwości jest tu widoczna, mimo zachowania pełnej informacji o kolorze.

Podsumowanie / Wnioski

Implementacja i testy uproszczonego algorytmu JPEG pozwoliły zaobserwować kluczowe aspekty tej metody kompresji stratnej:

1. **Podstawy działania:** Potok JPEG wykorzystuje transformację do przestrzeni YCbCr, co pozwala oddzielić informację o jasności (na którą oko jest bardziej czułe) od informacji o kolorze. Kluczowe straty wprowadzane są przez podpróbkowanie chrominancji i kwantyzację współczynników DCT.
2. **Podpróbkowanie chrominancji (4:2:2 vs 4:4:4):** Redukcja informacji o kolorze (4:2:2) jest najbardziej zauważalna na ostrych krawędziach między różnymi kolorami, gdzie może powodować "przeciekanie" barw. Na obszarach o łagodnych przejściach tonalnych lub gładkich tłach jej wpływ jest znacznie mniejszy i często niezauważalny. Pozwala jednak na redukcję

danych dla warstw Cb i Cr o 50%.

3. **Kwantyzacja DCT:** Zastąpienie dokładnych współczynników DCT wartościami całkowitymi (po podzieleniu przez tablicę Q) jest głównym źródłem utraty informacji o drobnych detalach i teksturach (wysokie częstotliwości w bloku DCT). Prowadzi to do rozmycia i pojawienia się artefaktów blokowych (widoczna siatka 8x8), szczególnie przy użyciu standardowych tabel kwantyzacji. Użycie tabeli jedynek minimalizuje ten efekt, ale kosztem znacznie mniejszej kompresji (bo większość współczynników pozostaje niezerowa).
4. **Kombinacja metod:** Największe artefakty i utrata jakości występują przy jednoczesnym zastosowaniu podpróbkowania chrominancji (4:2:2) i silnej kwantyzacji (standardowe tablice). Jest to jednak typowy tryb pracy JPEG, oferujący najlepszy kompromis między rozmiarem pliku a jakością wizualną dla naturalnych obrazów.
5. **Ograniczenia:** Nasz algorytm jest uproszczony (pomija m.in. kodowanie Huffmana/arytmetyczne dla skwantyzowanych współczynników), więc realny stopień kompresji byłby inny. Zakładaliśmy również, że wymiary fragmentów są wielokrotnością 8, co uprościło obsługę bloków.