

PROJEKT 2 – Analiza danych multimedialnych – Filip Hałys

Do wykonania projekt wykorzystano 2 zdjęcia kolorowe, oba w formacie 1000 na 1000 pikseli.
Pierwsze z nich (kontener) wygląda następująco:



Rys. 1 – Kontener

Drugie z nich (obraz ukryty) wygląda następująco:

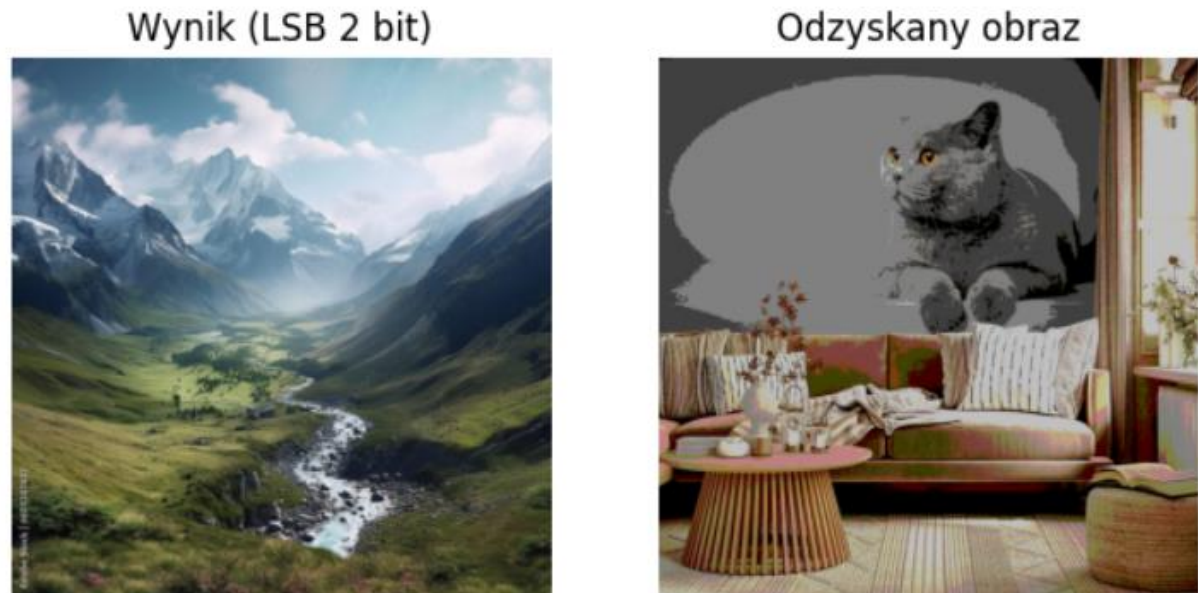


Rys. 2 – Obraz ukryty

Następnie trzykrotnie przy użyciu Pythona i metody LSB wykonano ukrycie zdjęcia z Rys. 2 wewnątrz zdjęcia z Rys. 1:

- na 4-ch najniższych bitach (LSB 4 bity),
- na 3-ch najniższych bitach (LSB 3 bity),
- na 2-ch najniższych bitach (LSB 2 bity).

Poniżej zaprezentowano wyniki operacji odzyskiwania zdjęcia ukrytego ze zdjęcia kontenera:



Rys. 3 – Wynik operacji odzyskiwania obrazu dla LSB 2 bity

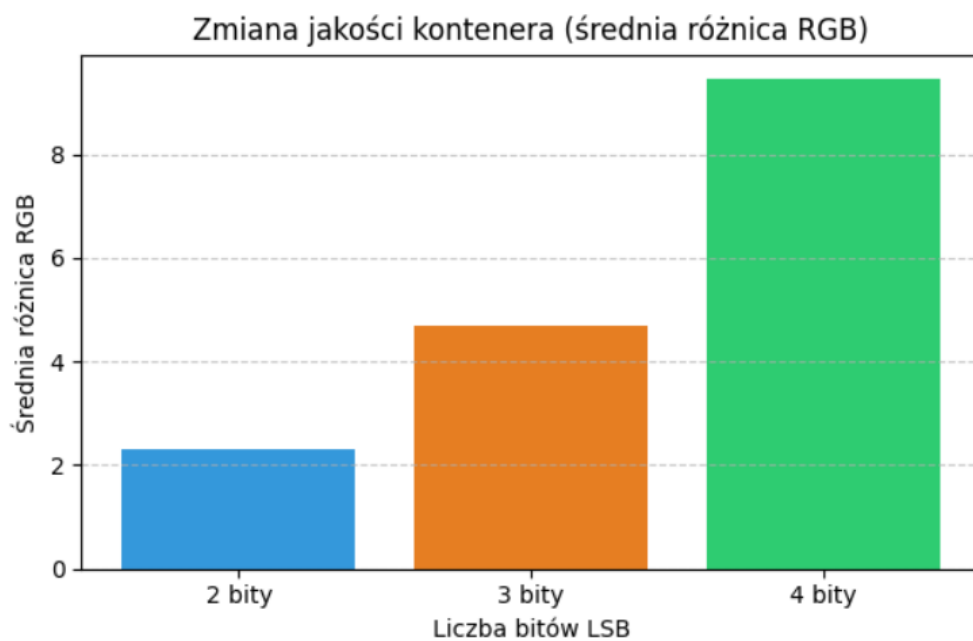


Rys. 4 – Wynik operacji odzyskiwania obrazu dla LSB 3 bity



Rys. 5 – Wynik operacji odzyskiwania obrazu dla LSB 4 bity

Finalnie stworzono wykres zależności między liczbą bitów służących do ukrywania zdjęcia, a średnią różnicą RGB (podobnie jak w projekcie 1). Poniżej zaprezentowano wykres:



Rys. 6 – Wykres zależności między liczbą bitów służących do ukrywania zdjęcia, a średnią różnicą RGB

Obserwacje, wnioski:

- Im więcej bitów użyto do ukrycia obrazu, tym bardziej kontener traci oryginalną jakość (o tym mówi wykres z Rys. 6). Dobrze to widać na kontenerze z Rys. 3 (LSB 2 bity), który jest dobrej jakości w porównaniu z Rys. 5 (LSB 4 bity), na którym to widać, że w prawym górnym rogu następuje „pikselizacja” nieba,

- Im więcej bitów użyto do ukrycia obrazu, tym lepsza jakość zdjęcia odzyskanego – dobrze to widać na tle za kotem i głową kota. Na Rys. 3 (LSB 2 bity) tło zlewa się z głową, natomiast na Rys. 5 (LSB 4) głowa kota wygląda zdecydowanie bardziej jakościowo,
- Wydaje mi się, iż są to dość oczekiwane i spodziewane wyniki.