4. domača naloga: Poenostavljanje sivinskih slik z diskretno kosinusno transformacijo

Diskretna kosinusna transformacija (angl. discrete cosinus transform, DCT) se pogosto uporablja za pretvorbo slike iz prostorske domene v frekvenčno domeno in obratno, saj v frekvenčni domeni iz slike lažje izluščimo glavne značilnosti. Najpreprostejši način za izluščenje glavnih značilnosti je, da pred pretvorbo slike iz frekvenčne domene nazaj v prostorsko domeno odstranimo koeficiente visokofrekvenčnih komponent. V ta namen jo bomo uporabili tudi pri tej nalogi.

Diskretna kosinusna transformacija je reverzibilna in brez izgub. Gre za posebno obliko diskretne Fourierove transformacije (DFT). Pri DFT koeficienti vključujejo amplitudo in fazo, medtem ko pri DCT koeficienti predstavljajo le amplitudo. Ker pri odstranjevanju koeficientov DCT ne spreminjamo faze, funkcija počasneje izgublja obliko in zato lahko odstranimo več koeficientov ter nenazadnje dosežemo boljše izgubno stiskanje.

Od DFT do DCT

Spomnimo se, da z inverzno diskretno Fourierovo transformacijo (DFT) diskretno funkcijo x_k, $k \in [0, N-1]$, v frekvenčni domeni predstavimo s koeficienti

$$X_n = \sum_{k=0}^{N-1} x_k \mathrm{e}^{-i\frac{2\pi}{N}kn} \quad , \tag{1}$$

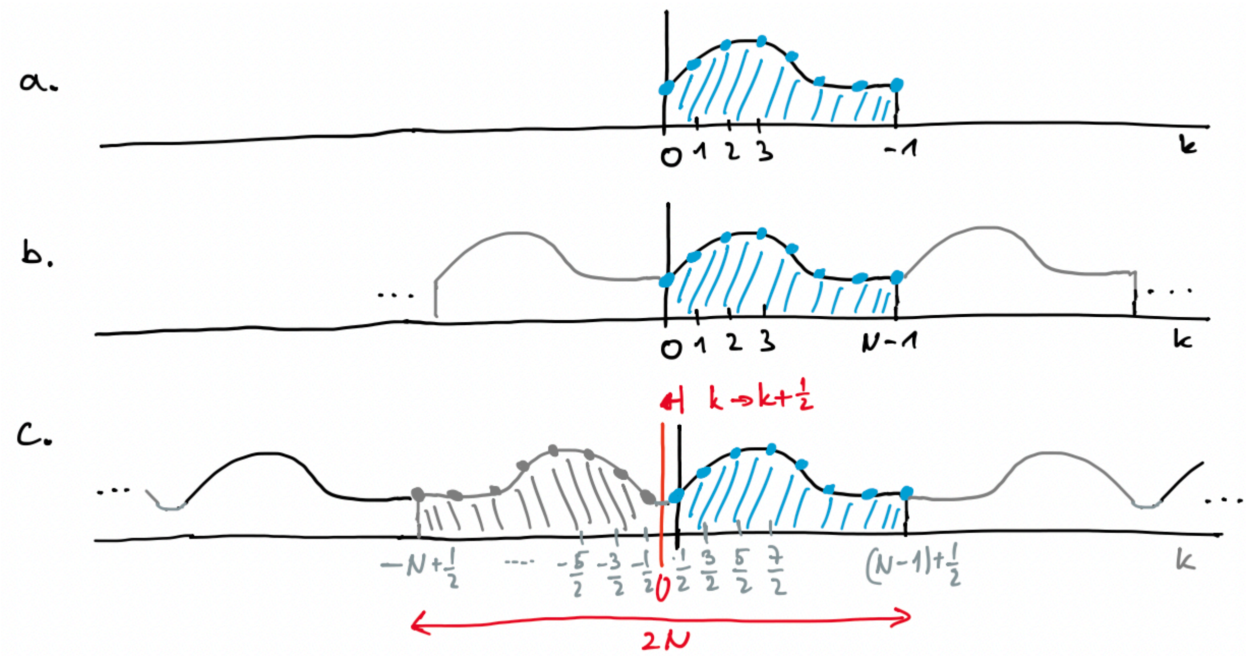
pri čemer kompleksni koeficient X_n z $n \in [-N/2, +N/2]$ bliže 0 ustreza nižji frekvenci, bolj oddaljen pa višji. Z inverzno diskretno Fourierovo transformacijo (IDFT) lahko iz koeficientov rekonstruiramo diskretno funkcijo,

$$x_k = \frac{1}{N} \sum_{n=-(N/2-1)}^{N/2} X_n \mathrm{e}^{i\frac{2\pi}{N}kn} \quad . \tag{2}$$

Diskretna Fourierova transformacija vključuje sode kosinusne in lihe sinusne funkcije različnih frekvenc, $\mathrm{e}^{i\frac{2\pi}{N}kn} = \cos(\frac{2\pi}{N}kn) + i\sin(\frac{2\pi}{N}kn)$, zato z njo lahko upodobimo poljubne funkcije. Na enačbo 2 lahko gledamo tudi kot na razvoj v Fourierovo vrsto, ki pa predpostavlja, da se diskretna funkcija periodično ponavlja (slika 1b).

Diskretna kosinusna transformacija vključuje samo kosinuse različnih frekvenc, s katerimi lahko upodabljamo samo sode funkcije. Zato naredimo dva preprosta trika (slika 1c): (i) abscisno os prestavimo za polovico koraka vzorčenja v negativno smer, $k \to k + \frac{1}{2}$, in (ii) diskretno funkcijo preslikamo čez novo abscisno os. Razširjena diskretna funkcija je še enkrat daljša, $N \to 2N$ in se razteza od $-N$ do $N-1$. Tudi za razširjeno diskretno funkcijo predpostavimo, da se periodično ponavlja. Perioda je daljša kot prej, ponavljajoča funkcija drugačna, na definicijskem območju osnovne diskretne funkcije pa ni sprememb. Na razširjeni diskretni funkciji izvedemo DFT:

$$X_n = \sum_{k=-N}^{N-1} x_k \mathrm{e}^{-i\frac{2\pi}{2N}(k+\frac{1}{2})n} \quad \text{in} \tag{3}$$



Slika 1: Diskretna funkcija (a) in periodičnost pri DFT (b) in DCT (c).

enak postopek ponovimo za IDFT:

$$x_k = \frac{1}{2N} \sum_{n=-(N-1)}^{N} X_n \mathrm{e}^{i\frac{2\pi}{2N}(k+\frac{1}{2})n} \quad . \tag{4}$$

V zgornjih enačbah smo zaradi boljše berljivosti ohranili stare oznake za x_k.

Ko v enačbi 3 upoštevamo sodost razširjene diskretne funkcije, $x_{-k} = x_{k-1}$, dobimo diskretno kosinusno transformacijo (DCT):

$$X_n = 2 \sum_{k=0}^{N-1} x_k \cos\left[\frac{\pi}{N}(k+\frac{1}{2})n\right] = \mathrm{DCT}(x_k) \quad , \quad n \in [0, N-1] \quad . \tag{5}$$

Vidimo, da koeficiente tvorimo iz kosinusov, zato velja sodost, $X_n = X_{-n}$. Ko sodost koeficientov upoštevamo v enačbi 4 in uporabimo zvezo $\cos(\pi k + \frac{\pi}{2}) = 0$, dobimo inverzno diskretno kosinusno transformacijo (IDCT):

$$x_k = \frac{1}{N}\left(\frac{1}{2}X_0 + \sum_{n=1}^{N-1} X_n \cos\left[\frac{\pi}{N}(k+\frac{1}{2})n\right]\right) = \mathrm{IDCT}(X_n) \quad , \quad k \in [0, N-1] \quad . \tag{6}$$

Standardni zapis diskretne kosinusne transformacije se nekoliko razlikuje od naše izpeljave, se pa pri njem podre neposredna povezava z diskretno Fourierovo transformacijo. Ostali bomo pri našem zapisu.

Pohitritev izračunov s FFT

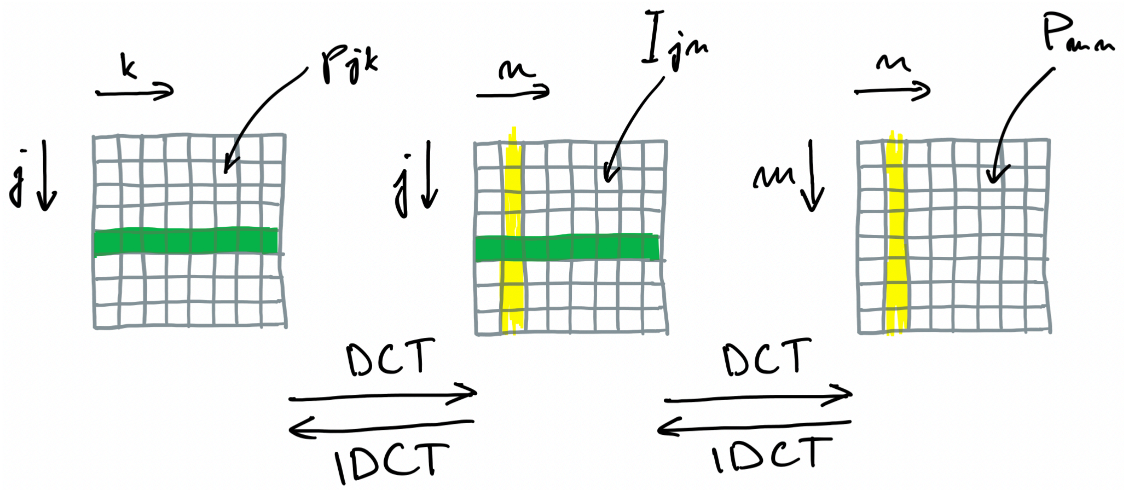
Časovna zahtevnost izračuna DCT po enačbi 3 je $O(N^2)$. Za hitrejši izračun lahko uporabimo hitro Fourierovo transformacijo (FFT) s časovno zahtevnostjo $O(N \log N)$.

FFT predpostavlja, da je razširjena funkcija definirana na intervalu $[0, 2N-1]$, zato jo moramo po transformaciji premakniti na interval $[-N, N-1]$. Izkoristimo lastnost premika v prostoru. Če diskretno funkcijo f_k z diskretno Fourierovo transformacijo F_n premaknemo za K, $g_k = f_{k-K}$, velja $G_n = \mathrm{e}^{-i2\pi K n} F_n$. Pri tem pazimo, da vrednost n sovpada s predznakom in vrednostjo indeksa koeficienta F_n. Kot rezultat vzamemo samo prvih N koeficientov, $X_0, \ldots, X_{N-1}$.

Na enak način lahko pohitrimo tudi izračun IDCT. Najprej pravilno dodamo koeficiente z negativnimi indeksi in vse koeficiente premaknemo v času v nasprotno smer kot prej. Nato izvedemo inverzno hitro Fourierovo transformacijo (IFFT) in izluščimo ustrezne vrednosti diskretne funkcije x_k.

DCT in IDCT v dveh dimenzijah

Diskretno kosinusno transformacijo lahko uporabljamo tudi na slikah, pri čemer jo razširimo na dve dimenziji. Transformacijo izvedemo v dveh korakih (slika 2):



Slika 2: Dvodimenzionalna DCT kot zaporedje dveh enodimenzionalnih DCT.

i. najprej izvedemo DCT na slikovnih točkah p_{jk} za vsako vrstico j posebej, da dobimo vmesne koeficiente, $I_{jn} = \mathrm{DCT}(p_{jk})$, $j \in [0, N-1]$,

ii. nato za vsak stolpec n posebej izvedemo DCT na vmesnih koeficientih, da dobimo koeficiente slike, $P_{mn} = \mathrm{DCT}(I_{jn})$, $n \in [0, N-1]$.

Na enak način v dveh dimenzijah izvedemo IDCT. Če sledimo sliki 2, postopamo v obratnem vrstnem redu:

i. najprej po stolpcih izvedemo IDCT na koeficientih slike P_{mn}, da dobimo vmesne koeficiente, $I_{jn} = \mathrm{IDCT}(P_{mn})$, $n \in [0, N-1]$,

ii. nato po vrsticah j izvedemo IDCT na vmesnih koeficientih, da dobimo slikovne točke, $p_{jk} = \mathrm{IDCT}(I_{jn})$, $j \in [0, N-1]$.

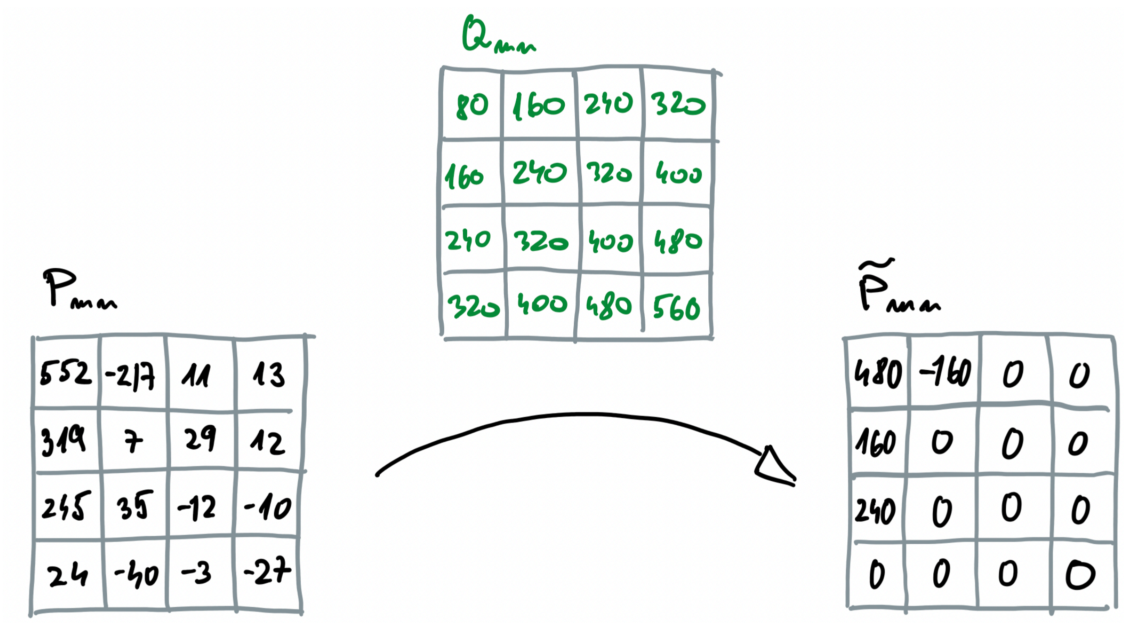
Kvantizacija koeficientov slike

Koeficienti v zgornjem levem delu slike predstavljajo nižje frekvence in imajo zato običajno večje absolutne vrednosti (slika 3). Ker človek zaradi psihovizualnih sposobnosti težko zazna vse spremembe na hitro se spreminjajočem delu slike, lahko koeficiente višjih frekvenc v precejšnji meri odstranimo. Da zagotovimo postopen prehod, koeficientov ne postavimo kar na nič, ampak jih raje kvantiziramo,

$$\tilde{P}_{mn} = \mathrm{sign}(P_{mn}) \left\lfloor \frac{|P_{mn}|}{Q_{mn}} \right\rfloor Q_{mn} \quad , \tag{7}$$

pri čemer vrednosti elementov kvantizacijske matrike nastavimo tako, da se povečujejo od levega zgornjega kota proti levemu spodnjemu kotu,

$$Q_{mn} = 20N(1 + m + n) \quad . \tag{8}$$



Slika 3: Kvantizacija koeficientov slike.

Kvantizirani koeficienti imajo omejen nabor vrednosti, zato so primernejši za stiskanje. Za učinkovito stiskanje slike bi na kvantiziranih koeficientih lahko izvedli verižno kodiranje v kombinaciji s Huffmanovim ali aritmetičnim kodiranjem, tako kot v formatu JPEG.

Poenostavljanje sivinske slike

Poenostavljanje slike z diskretno kosinusno transformacijo izvedemo v naslednjih korakih:

1. 8-bitno sivinsko sliko preberemo.
2. Vzemimo, da sta širina W in višina H slike večkratnika $N \in \{4, 8, 16\}$. Sliko razdelimo na naprekrivajoče kvadrate velikosti $N \times N$.
3. Na vsakem kvadratu posebej izvedemo naslednje operacije:
   a. Od vsake slikovne točke $o_{jk} \in [0, 255]$ odštejemo vrednost 128, $p_{jk} = p_{jk} - 128$.
   b. Izvedemo DCT v dveh dimenzijah – iz slikovnih točk p_{jk} dobimo slikovne koeficiente P_{mn}.
   c. Slikovne koeficiente P_{mn} kvantiziramo.
   d. Na kvantiziranih koeficientih $\tilde{P}_{mn}$ izvedemo IDCT, da dobimo slikovne točke poenostavljene slike $\tilde{p}_{jk}$. Vrednosti dobljenih slikovnih točk so še zamaknjene.
   e. Vrednosti zamaknjenih slikovnih točk zaokrožimo navzdol, prištejemo 128 in poskrbimo, da so vrednosti znotraj območja $[0, 255]$,

$$\tilde{o}_{jk} = \min\left(\max(\lfloor \tilde{p}_{jk} \rfloor + 128, 0), 255\right) \quad . \tag{9}$$

4. Sivinsko sliko po potrebi shranimo

Podobnost med originalno in poenostavljeno sliko

Mera PSNR (angl. peak signal-to-noise ratio) se pogosto uporablja za kvantitativno opredelitev kakovosti rekonstrukcije slik pri stiskanju z izgubami. V našem primeru bomo primerjali originalno in poenostavljeno 8-bitno sliko,

$$\mathrm{PSNR} = 20 \log_{10}(2^8 - 1) - 10 \log_{10}\left(\frac{1}{H\,W} \sum_{j=0}^{H-1} \sum_{k=0}^{W-1} (o_{jk} - \tilde{o}_{jk})^2\right) \quad . \tag{10}$$

Naloga

V datoteki `naloga4.py` v jeziku Python napišite funkcijo z imenom `naloga4`, ki izračuna:

- mero PSNR med originalno in poenostavljeno sliko.

Vhodna argumenta funkcije sta

- slikovne točke vhodne slike v obliki numpy matrike (`slika`) in
- velikost okna (`velikostOkna`).

Prototip funkcije:

```python
import numpy as np
def naloga4(slika: np.array, velikostOkna: int) -> float:
    """
    Poenostavi sliko

    Parameters
    ----------
    slika : numpy array
        vhodna slika
    velikostOkna : int
        velikost okna za DCT

    Returns
    -------
    PSNR : float
         Peak Signal to Noise Ratio
    """

    PSNR = float('nan')
    return PSNR
```

Testni primeri

Na učilnici se nahajajo tri testne slike, za katere imate podane tudi mere PSNR za velikosti kvadratov $N \in 4, 8, 16$. Podatki so podani v obliki datotek `.json`. Priloženo imate tudi funkcijo `test_naloga4`, ki jo lahko uporabite za preverjanje pravilnosti rezultatov. Pri testiranju vaših funkcij upoštevajte naslednje omejitve:

- rezultat je pravilen, če se od danega razlikuje za manj kot 10^{-4},
- izvajanje funkcije je časovno omejeno na 30 sekund.
- Vaš program lahko uporablja samo tiste pakete, ki so del standardne knjižnice Python 3.12 (https://docs.python.org/3.12/library/) in pakete `numpy`, `pillow` ter `scipy`. Na sistemu za preverjanje vaših rešitev drugi paketi niso nameščeni.

Točkovanje

- Pri vsaki nalogi dobite točko za pravilen izračun PSNR.
- Za rešitev, pri kateri račune pohitrite s FFT, dobite dodatne 3 točke.

Namigi

Delajte sami

Pri preverjanju nalog uporabljamo tudi sistem za zaznavanje plagiatorstva. Verjemite, da prepisovanje težko zakrijete.

Uporabne funkcije in razredi

- `numpy`: `cos`, `exp`, `fft.fft`, `fft.ifft`,

Vmesni rezultati za primer 1

- začetna sivinska slika

$$o = \begin{pmatrix} 0 & 255 & 0 & 255 \\ 255 & 0 & 255 & 0 \\ 0 & 255 & 0 & 255 \\ 255 & 0 & 255 & 0 \end{pmatrix}$$

- premik

$$p = \begin{pmatrix} -128 & 127 & -128 & 127 \\ 127 & -128 & 127 & -128 \\ -128 & 127 & -128 & 127 \\ 127 & -128 & 127 & -128 \end{pmatrix}$$

- DCT na p po vrsticah (enačba 5)

$$\begin{aligned} I &= \begin{pmatrix} -4{,}0 & -276{,}0 & 0{,}0 & -666{,}3 \\ -4{,}0 & 276{,}0 & 0{,}0 & 666{,}3 \\ -4{,}0 & \mathbf{-276{,}0} & 0{,}0 & -666{,}3 \\ -4{,}0 & 276{,}0 & 0{,}0 & 666{,}3 \end{pmatrix} \\ I_{21} &= 2 \cdot \left[-128 \cdot \cos\left(\frac{\pi}{8} \cdot 1\right) + 127 \cdot \cos\left(\frac{3\pi}{8} \cdot 1\right) - 128 \cdot \cos\left(\frac{5\pi}{8} \cdot 1\right) + 127 \cdot \cos\left(\frac{7\pi}{8} \cdot 1\right) \right] \\ &= -276{,}0 \end{aligned}$$

- DCT na I po stolpcih (enačba 5)

$$\begin{aligned} P &= \begin{pmatrix} -32{,}0 & 0{,}0 & 0{,}0 & 0{,}0 \\ 0{,}0 & -597{,}6 & 0{,}0 & -1442{,}5 \\ 0{,}0 & 0{,}0 & 0{,}0 & 0{,}0 \\ 0{,}0 & \mathbf{-1442{,}5} & 0{,}0 & -3482{,}5 \end{pmatrix} \\ P_{31} &= 2 \cdot \left[-276 \cdot \cos\left(\frac{\pi}{8} \cdot 3\right) + 276 \cdot \cos\left(\frac{3\pi}{8} \cdot 3\right) - 276 \cdot \cos\left(\frac{5\pi}{8} \cdot 3\right) + 276 \cdot \cos\left(\frac{7\pi}{8} \cdot 3\right) \right] \\ &= -1442{,}5 \end{aligned}$$

- rezultat kvantizacije (enačba 7):

$$\begin{aligned} \tilde{P} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -480 & 0 & \mathbf{-1200} \\ 0 & 0 & 0 & 0 \\ 0 & -1200 & 0 & -3360 \end{pmatrix} \\ \tilde{P}_{13} &= \lfloor -1442{,}5/400 \rceil \cdot 400 = -1200 \end{aligned}$$

- IDCT na P' po stolpcih (enačba 6)

$$
\begin{aligned}
\tilde{I} &= \begin{pmatrix} -0{,}0 & -225{,}7 & 0{,}0 & -598{,}6 \\ -0{,}0 & 231{,}2 & 0{,}0 & 661{,}2 \\ -0{,}0 & \mathbf{-231{,}2} & 0{,}0 & -661{,}2 \\ -0{,}0 & 225{,}7 & 0{,}0 & 598{,}6 \end{pmatrix} \\
\tilde{I}_{21} &= \frac{1}{4}\left[\frac{0}{2} - 480 \cdot \cos\left(\frac{5\pi}{8}\cdot 1\right) + 0 \cdot \cos\left(\frac{5\pi}{8}\cdot 2\right) - 1200 \cdot \cos\left(\frac{5\pi}{8}\cdot 3\right)\right] \\
&= -231{,}2
\end{aligned} \tag{11}
$$

- IDCT na I' po vrsticah (enačba 6)

$$
\begin{aligned}
\tilde{p} &= \begin{pmatrix} -109{,}4 & 116{,}7 & -116{,}7 & 109{,}4 \\ 116{,}7 & -130{,}6 & 130{,}6 & -116{,}7 \\ -116{,}7 & \mathbf{130{,}6} & -130{,}6 & 116{,}7 \\ 109{,}4 & -116{,}7 & 116{,}7 & -109{,}4 \end{pmatrix} \\
\tilde{p}_{21} &= \frac{1}{4}\left[\frac{0}{2} - 231{,}2 \cdot \cos\left(\frac{3\pi}{8}\cdot 1\right) + 0 \cdot \cos\left(\frac{3\pi}{8}\cdot 2\right) - 661{,}2 \cdot \cos\left(\frac{3\pi}{8}\cdot 3\right)\right] \\
&= 130{,}6
\end{aligned} \tag{12}
$$

- zaokrožitev, premik in omejitev na vrednosti $[0, 255]$ (enačba 9)

$$
\tilde{o} = \begin{pmatrix} 18 & 244 & 11 & 237 \\ 244 & 0 & 255 & 11 \\ 11 & 255 & 0 & 244 \\ 237 & 11 & 244 & 18 \end{pmatrix} \tag{13}
$$

- PSNR (enačba 9) med originalno sliko o in poenostavljeno sliko $\tilde{o}$ je 26,62.

Osnovna in končna slika ter razlika za primer 2

Čas izvajanja za primer 3

Neoptimizirana koda za primer 3, ki sledi enačbama 5 in 6, se na testnem računalniku izvaja 157 s.

Literatura

[1] Discrete Cosinus Transform, Wikipedia, 2024, `https://en.wikipedia.org/wiki/Discrete_cosine_transform`

[2] E. Roberts, Lossy Data Compresion, Stanford, 2024, `https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/index.htm`

[3] Peak Signal-to-Noise Ratio, Wikipedia, 2024, `https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio`