

## Zagovor laboratorijskih vaj - Naloga 2023-1

Priprava: Gašper Podobnik & Tomaž Vrtovec

### Naloga

Kolaž (beseda izhaja iz francoske besede *coller*, ki pomeni *(z)lepiti skupaj*) je tehnika likovnega ustvarjanja, ki se uporablja predvsem v vizualnih umetnostih, najdemo pa jo tudi v glasbi, s katero nastane umetnost iz skupka različnih oblik in tako ustvari novo celoto.

Vaša naloga bo preslikava intenzitet vhodne RGB slike, iz katere boste nato ustvarili enostaven kolaž.

Vhodna RGB slika



Dana je dvodimenzionalna (2D) barvna (RGB) slika `planina-509x339-08bit.jpeg` velikosti  $X \times Y \times RGB_{dim} = 509 \times 339 \times 3$  slikovnih elementov, ki je zapisana v formatu jpeg z 8 biti na slikovni element. Slikovni elementi so izotropni.

1. Naložite dano sliko in napišite funkcijo za standardizacijo intenzitet slike:

```
def std_channels(iImage, x_R, y_R, x_G, y_G, x_B, y_B):  
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno RGB sliko, argumenti  $\{x_R, x_G, x_B\}$  in  $\{y_R, y_G, y_B\}$  pa predstavljajo parametre  $x$  in  $y$  za posamezen barvni kanal RGB slike. Izhodni argument `oImage` je standardizirana slika velikosti  $X \times Y \times 3$  ( $X$  in  $Y$  sta dimenziji vhodne slike).

Standardizacija je sivinska preslikava, ki je določena s spodnjo enačbo:

$$I_{std} = \frac{I - x}{y},$$

kjer sta  $x$  in  $y$  parametra te preslikave,  $I$  predstavlja vhodno sliko,  $I_{std}$  pa standardizirano sliko.

Funkcija naj standardizira vsak (barvni) kanal slike posebej, torej ločeno za rdeč, zelen in moder barvni kanal, pri čemer se za vsak kanal uporabi pripadajočo vrednost iz vhodnih parametrov. Izhodna slika je RGB slika, sestavljena iz vseh treh standardiziranih kanalov. *Namig:* pred standardizacijo ustrezno nastavite podatkovni tip slike.

Vhodno sliko standardizirajte, pri čemer za manjkajoče vhodne parametre uporabite sledeče vrednosti:  $x_R=100$ ,  $y_R=50$ ,  $x_G=60$ ,  $y_G=30$ ,  $x_B=40$ ,  $y_B=100$ . Rezultat shranite v spremenljivko `img_std`. Dobljeno sliko tudi prikažite.

2. Napišite funkcijo, ki pripravi predlogo območij (koščkov), ki sestavljajo kolaž, za sliko velikosti  $X \times Y$ :

```
def image2pieces(X, Y, N):  
    return oLabelImage, oPts
```

kjer vhodna argumenta  $X$  in  $Y$  predstavljata velikost slike, ki jo želimo razdeliti na območja, argument  $N$  pa število območij, ki jih želimo imeti v novi sliki. Izhodni argument `oLabelImage` je matrika velikosti  $X \times Y$ , v kateri elementi z isto vrednostjo označujejo posamezno območje slike, `oPts` pa matrika velikosti  $N \times 2$ , ki nosi koordinate vseh središč območij. Število različnih vrednosti v matriki `oLabelImage` je torej enako številu območij  $N$ .

Funkcija naj naključno generira  $N$  parov koordinat  $oPts = \{(x_j, y_j), j = 1, 2, \dots, N\}$ , ki ležijo znotraj meja slike in predstavljajo središča območij. Pri tem si lahko pomagata s funkcijo `np.random.choice` (koordinate točk naj bodo cela števila). Pazite na to, da bodo vse točke različne (da ne pride do podvajanja). Vrednost posameznega elementa v izhodni matriki `oLabelImage` naj bo enaka indeksu najbližjega središča iz matrike `oPts`, torej  $j$  oz.  $j - 1$  (zaradi Pythonovega načina indeksiranja).

Preizkusite funkcijo, pri čemer za vrednosti  $X$  in  $Y$  uporabite velikost vhodne slike,  $N$  pa nastavite na 255. Dobljeno matriko `oLabelImage` prikažite kot sivinsko sliko, pri čemer nastavite `cmap='jet'`.

3. Napišite funkcijo, ki iz vhodne slike ustvari kolaž:

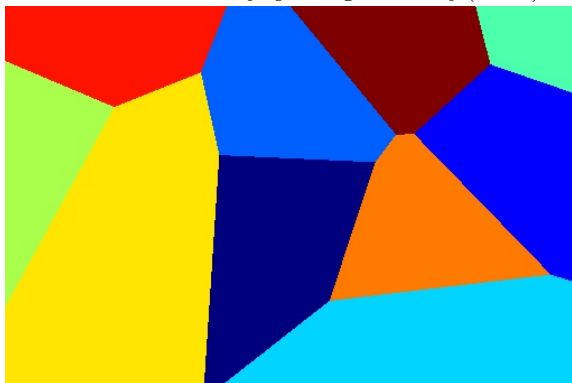
```
def img2collage(iImage, N):  
    return oImage, oLabelImage
```

kjer vhodni argument `iImage` predstavlja vhodno sliko,  $N$  pa število območij, ki jih želimo imeti v izhodni sliki.

S funkcijo `image2pieces` najprej pridobite predlogo območij (koščkov) za vhodno sliko. Na podlagi pridobljene predloge območij nato izračunajte intenzitete izhodne slike tako, da bo intenziteta na posameznem območju izhodne slike enaka povprečni intenziteti istoležnega območja na vhodni sliki (povprečno intenziteto ločeno izračunajte za vsak barvni kanal).

Preizkusite funkcijo na sliki `img_std` za dva različna parametra, in sicer za  $N=255$  in  $N=1000$ , ter prikažite dobljene slike. Prikažite tudi obe predlogi območij (`oLabelImage`).

Primer vizualizacije predloge območij ( $N=10$ )



Primer izhodne RGB slike kolaža ( $N=255$ )

