Univerza v Ljubljani, Fakulteta za elektrotehniko
Podiplomski magistrski študijski program 2. stopnje Elektrotehnika
Informacijsko komunikacijske tehnologije - Obdelava slik in videa (64238)

**ZAGOVOR**

# Zagovor laboratorijskih vaj - Naloga 2022-3

Pripravila: Gašper Podobnik & Tomaž Vrtovec

## Naloga

RGB barvni model sliko predstavi s tremi kanali; kanalom za rdečo, zeleno in modro barvno komponento. Kot alternative temu načinu zapisa so se uveljavili še drugi barvi modeli, npr. CMYK, HSV, YUV. Danes boste spoznali barvni model HSL, kjer prvi kanal predstavlja odtenek barve (ang. *hue*), drugi intenzivnost (ang. *saturation*), tretji pa svetlost (ang. *lightness*).
Vaša naloga bo preslikava intenzitet iz modela RGB v model HSL, kjer boste opravili filtriranje. Dobljeno sliko boste nato preslikali nazaj v model RGB.

Vhodna RGB slika

Izhodna RGB slika

Dana je dvodimenzionalna (2D) barvna (RGB) slika planina-uint8.jpeg velikosti $X \times Y \times RGB_{dim} = 612 \times 408 \times 3$ 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 standardize_image(iImage):
    # ...
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno RGB sliko. 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 določna s spodnjo enačbo:

$$I_{std} = \frac{I - \hat{v}}{\sigma},$$

kjer $I$ predstavlja vhodno sliko, $\hat{v}$ povprečno intenziteto v vhodni sliki, $\sigma$ standardno deviacijo intenzitet v vhodni sliki, $I_{std}$ pa standardizirano sliko (intenziteta 0 v standardizirani sliki torej ustreza povprečni vrednosti, intenziteta 1 pa standardni deviaciji v vhodni sliki).
Funkcija naj standardizira vsak (barvni) kanal slike posebej, torej ločeno za rdeč, zelen in moder barvni kanal. Izhodna slika je RGB slika, sestavljena iz vseh treh standardiziranih kanalov.
*Namig:* pred standardizacijo ustrezno nastavite podatkovni tip slike. Za izračun povprečna vrednosti in standardne deviacije lahko uporabite vgrajene funkcije.

Vhodno sliko standardizirajte in rezultat shranite v spremenljivko `img_std`. Dobljeno sliko prikažite.

2. Napišite funkcijo za pretvorbo slike iz barvnega modela RGB v HSL:

```
def rgb2hsl(iRGB):
    # ...
    return oHSL
```

kjer vhodni argument `iRGB` predstavlja vhodno RGB sliko. Izhodni argument `oHSL` pa je slika zapisana v HSL modelu. Pomagajte si z naslednjimi izračuni, pri čemer so $r_i$, $g_i$ in $b_i$ vrednosti rdeče, zelene in modre barvne komponente v poljubnem slikovnem elementu $(x_i,\ y_i)$:

$$
\begin{aligned}
v_i &= max(r_i,\ g_i,\ b_i) \\
c_i &= max(r_i,\ g_i,\ b_i) - min(r_i,\ g_i,\ b_i) \\
l_i &= v_i - \frac{c_i}{2} \\
h_i &= \begin{cases} 60° \cdot (\frac{g_i - b_i}{c_i}), & v_i = r_i \\ 60° \cdot (2 + \frac{b_i - r_i}{c_i}), & v_i = g_i \\ 60° \cdot (4 + \frac{r_i - g_i}{c_i}), & v_i = b_i \\ 0, & c_i = 0 \end{cases} \\
s_i &= \begin{cases} 0, & l_i = 0\ ali\ l_i = 1 \\ \frac{c_i}{1 - |2v_i - c_i - 1|}, & drugače \end{cases}
\end{aligned}
$$

Dobljene vrednosti $h_i$, $s_i$, $v_i$, ki sestavljajo HSL zapis (*hue*, *saturation* in *value*), zložite v matriko enake velikosti kot je vhodna RGB slika, tako da bo imel vsak slikovni element $(x_i,\ y_i)$ vrednosti $(h_i,\ s_i,\ v_i)$.

Preizkusite funkcijo na normalizirani vhodni sliki `img_std` in dobljeno sliko shranite v spremenljivko `img_hsl` ter jo prikažite.

3. Sledi korak, kjer filtrirate sliko. Iz HSL slike `img_hsl` izločite rezino, ki ustreza kanalu $H$, in jo shranite v spremenljivko `h_slice`. Vrednosti v `h_slice`, ki so večje ali enake od 100 **in** manjše od 250, delite s številom pet. Z dobljeno rezino sestavite novo HSL sliko (rezini S in V ostajata nespremenjeni), ki jo shranite v spremenljivko `img_hsl_transformed` in jo prikažite.

4. Napišite funkcijo za pretvorbo slike iz barvnega modela HSL v RGB:

```
def hsl2rgb(iHSL):
    # ...
    return oRGB
```

kjer vhodni argument `iHSL` predstavlja vhodno HSL sliko, `oRGB` pa izhodno RGB sliko. Pomagajte si z naslednjimi izračuni, pri čemer so $h_i$, $s_i$ in $l_i$ vrednosti treh HSL komponent v poljubnem slikovnem elementu $(x_i, y_i)$, *mod* pa pomeni modulo (ostanek pri celoštevilskem deljenju):

$$
\begin{aligned}
c_i &= (1 - |2l_i - 1|) \cdot s_i \\
\hat{h_i} &= \frac{h_i}{60^\circ} \\
x_i &= c_i \cdot (1 - |\hat{h_i} \; mod \; 2 - 1|) \\
(\hat{r_i}, \hat{g_i}, \hat{b_i}) &= \begin{cases} (c_i, x_i, 0), & 0 \leq \hat{h_i} < 1 \\ (x_i, c_i, 0), & 1 \leq \hat{h_i} < 2 \\ (0, c_i, x_i), & 2 \leq \hat{h_i} < 3 \\ (0, x_i, c_i), & 3 \leq \hat{h_i} < 4 \\ (x_i, 0, c_i), & 4 \leq \hat{h_i} < 5 \\ (c_i, 0, x_i), & 5 \leq \hat{h_i} < 6 \end{cases} \\
m_i &= l_i - \frac{c_i}{2} \\
(r_i, g_i, b_i) &= (\hat{r_i} + m_i, \hat{g_i} + m_i, \hat{b_i} + m_i)
\end{aligned}
$$

Dobljene vrednosti $r_i$, $g_i$, $b_i$ zložite v matriko enake velikosti kot je vhodna HSL slika, tako da bo imel vsak slikovni element $(x_i,\ y_i)$ vrednosti $(r_i,\ g_i,\ b_i)$.

Preizkusite funkcijo na sliki `img_hsl` in `img_hsl_transformed` ter prikažite dobljeni sliki.