

Vaja 1: Uvod v Python

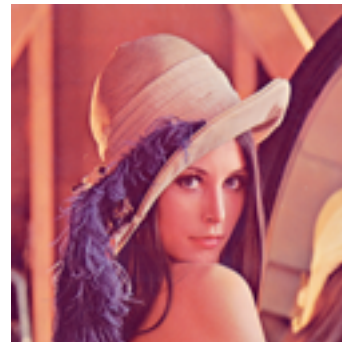
Pripravili: Luka Škrlić, Gašper Podobnik & Tomaž Vrtovec

Navodila

Naslednje naloge se opravi med laboratorijsko vajo.

Uvodna vaja služi spoznavanju osnovnih postopkov za nalaganje in prikazovanje slik v programskem okolju Python. Na spletni učilnici predmeta “Obdelava slik in videa” (<https://ucilnica-fe.uni-lj.si/>) se pod poglavjem “Uvod v obdelavo slik” nahaja mapa “Gradivo za laboratorijske vaje (Uvod v Python)”, ki vsebuje tri datoteke, in sicer:

- lena-color.png,
- lena-gray-410x512-08bit.raw in
- lena-color-512x410-08bit.raw.



Datoteke prenesite na izbrano lokacijo na računalniku, ki je običajno mapa z imenom **data** na enaki lokaciji, v kateri se bodo nahajale Pythonove skripte (programi in funkcije).

1. Datoteka lena-color.png vsebuje barvno sliko, zapisano v PNG formatu. Naložite sliko kot matriko v okolju Python. Pomagajte si s funkcijo `imread()`, ki jo najdete v knjižnici `matplotlib` v modulu `pyplot`. Sliko nato prikažite na zaslon s pomočjo funkcij `figure()` in `imshow()` ter jo shranite v JPEG formatu s pomočjo funkcije `imsave()` – tudi ti dve funkciji najdete v modulu `matplotlib.pyplot`.
2. Datoteka lena-gray-410x512-08bit.raw vsebuje sivinsko sliko velikosti $X \times Y = 410 \times 512$ slikovnih elementov, medtem ko datoteka lena-color-512x410-08bit.raw vsebuje barvno (RGB) sliko velikosti $X \times Y = 512 \times 410$ slikovnih elementov. Obe sliki sta zapisani v obliki surovih podatkov (RAW) z 8 biti na slikovni element (vrsta podatkov `'uint8'`). Napišite funkcijo za nalaganje poljubne sivinske oz. barvne slike v RAW formatu:

```
def loadImage(iPath, iSize, iType):  
    # ...  
    return oImage
```

kjer je `iPath` pot do slike (mapa in ime datoteke), `iSize` seznam velikosti slike (v slikovnih elementih), `iType` pa oblika zapisa (vrsta podatkov). Funkcija vrne sliko `oImage` v obliki matrike. Uporabite funkcijo `ndarray()` iz knjižnice `numpy` v povezavi s funkcijo `open()` in metodo `close()`.

3. Napišite funkcijo za prikazovanje poljubne sivinske oz. barvne slike na zaslon:

```
def displayImage(iImage, iTitle):  
    # ...  
    return fig
```

kjer je `iImage` slika, ki jo želite prikazati, `iTitle` pa naslov prikaznega okna, v katerem je koordinatni sistem prikazane slike. Uporabite funkcijo `imshow()` iz modula `matplotlib.pyplot`, prikaz pa prilagodite zapisu s pomočjo parametra `cmap` ter s parametrom `aspect` poskrbite, da bo velikost slikovnega elementa enaka v obeh dimenzijah. Izhodni argument `fig` naj predstavlja objekt prikaznega okna (`fig = plt.figure()`).

Gradivo

Naslednje naloge so neobvezne in namenjene boljšemu razumevanju vsebine.

Napišite funkcijo za shranjevanje poljubne sivinske slike v RAW format:

```
def saveImage(iImage, iPath, iType):  
    # ...
```

kjer je `iImage` slika, ki jo želite shraniti, `iPath` pot do slike (mapa in ime datoteke), `iType` pa oblika zapisa (vrsta podatkov). Uporabite funkcijo `open()` v povezavi z metodama `write()` in `close()`.