

## Vježba 1

---

**CILJ:** Upoznati osnovne mogućnosti za digitalnu obradu i manipulaciju slikama u programskom jeziku Python koristeći osnovne i neke dodatne programske knjižnice poput *os*, *NumPy*, *Pillow*, *OpenCV*, *Matplotlib* i sl.

- 1) Slike dostupne u mapi *images* (u sustavu Merlin) postavite u radnu mapu gdje ćete smjestiti i radnu datoteku *zadatak1.py*. Koristeći odgovarajuće naredbe u pythonu sliku *auto.jpg* učitajte i prikažite na zaslonu:

- a) Koristeći biblioteku [OpenCV](#).
- b) Koristeći biblioteku [Pillow](#).

**NAPOMENA:** Prema potrebi (ovisno o okruženju u kojem radite: Python interpreter instaliran na osobno računalo, udaljeni interpreter kao npr. *Jupyter Notebook* ili *Google Colab Notebook*) za prikaz slike na zaslonu dodatno koristiti *OpenCV* zakrpe ili *Matplotlib* biblioteku.

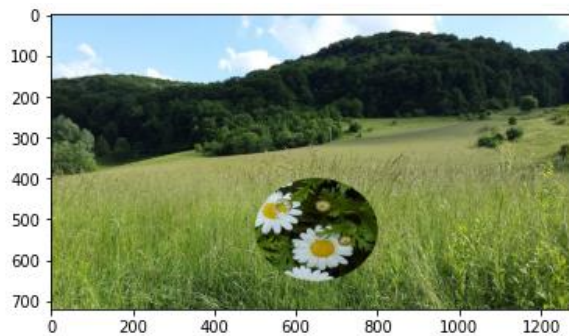
- 2) Učitajte sliku *pillow\_logo.png* koristeći biblioteku PIL. Nakon učitavanja, prikažite je na zaslonu.
  - a) Na zaslon ispišite njezin format.
  - b) Na zaslon ispišite njene dimenzije u pikselima.
  - c) Na zaslon ispišite visinu slike u pikselima.
  - d) Na zaslon ispišite *color mode* (naziv paleta boja).
- 3) Učitajte sliku *auto.jpg* koristeći biblioteku PIL, a potom je pretvorite u *grayscale* mod. Na zaslonu prikažite:
  - a) *Numpy* polje učitano objekta *Image*.
  - b) Ispišite visinu, širinu i broj kanala na temelju kreiranog polja.
  - c) Stvarnu sliku konvertiranu u *grayscale*.
  - d) Stvarnu sliku, ali tako da dubina boje bude 1-bitna.
  - e) Usporedno dvije slike (jednu pored druge) tako da je prvo prikazan izvornik slikovne datoteke, a uz nju i slika u *grayscale* modu.
- 4) Učitajte sliku *pillow\_logo.png* te je pretvorite u *grayscale* mode. Pohranite je na čvrsti disk kako novu slikovnu datoteku *pillow\_logo\_grayscale.png* te još jednu slikovnu datoteku u tonovima sive tako da joj *alpha* kanal ostane sačuvan (*pillow\_logo\_grayscale\_alpha.png*). Postoji li razlika u memoriji koju kreirane datoteke zauzimaju na čvrstom disku prije kompresije u **png** format? Zašto?
- 5) Napišite skriptu koja će sve slike iz mape *animals* koje su u formatu JPG, automatizirano konvertirati u PNG format i sve ih pohraniti u zasebnu mapu naziva PNGS.

- 6) Učitajte sliku *lav.jpg*. Prikažite je na ekranu, a potom je prikažite:
  - a) rotiranu za 45 stupnjeva,
  - b) transponirano (vertikalno),
  - c) transponiranu verziju pohranite kao datoteku u *pdf* formatu.
- 7) Učitajte sliku *paun.jpg*. Iz biblioteke PIL uključite *ImageFilter* te na sliku primijenite sljedeće filtere:
  - a) *Gaussian Blur* s radiusom 35,
  - b) *Contour* (iscrtavanje kontura),
  - c) *Emboss* (izgled kao da je uklesano u kamen),
  - d) *UnsharpMask* (s radijusom 85, 350% i threshold 55) za izoštravanje slike
  - e) *Median filter* za čišćenje šuma s radiusom 16.

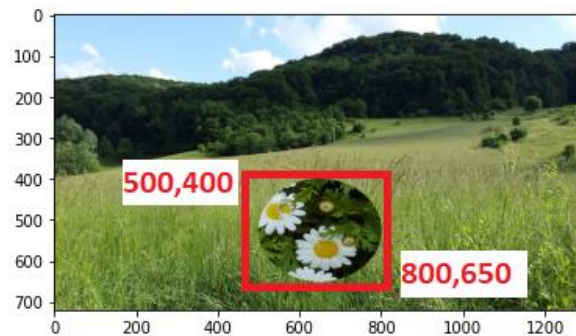
Slike dobivene filtriranjem pokažite na jednom plotu sa 6 slika koje su razvrstane u dva retka i tri stupca. Prva slika neka bude original, a sve ostale neka budu njene obrade s odgovarajućim filterima.

- 8) Koristeći metodu *point()* slici *orca.jpg* promijenite kontrast tako da ga pojačate. To učinite na način da svakom pikselu uvećate vrijednost kontrasta za 1,35. Rezultat prikažite na zaslonu.
- 9) Kreirajte novu bitmapu (sliku) veličine 200 x 200 piksela u RGB formatu tako da:
  - a) sadrži samo piksele crne boje,
  - b) sadrži samo piksele bijele boje,
  - c) sadrži samo piksele zelene boje (za boju koristiti zapis *rgb()*)
  - d) sadrži samo piksele žute boje (za boju koristiti zapis *hsl()*)
- 10) Kreirajte malu bitovnu mapu širene 5 piksela i visine 5 piksela koristeći CMYK paletu. Svi pikseli neka budu obojani u boju s vrijednostima (0, 255, 0, 0). Na zaslonu prikažite:
  - a) stvarni izgled slike,
  - b) vrijednosti svih piksela iz rasterske mreže (ispisano kao 2D matrica),
  - c) promijenite vrijednosti svih piksela u prvome retku na (255, 255, 0, 0),
  - d) ponovo nacrtajte sliku (listu s vrijednostima piksela pretvoriti u *nparray* a potom generirati sliku iz polja i prikazati je na zaslonu).
- 11) Postupkom miješanja (*image blend*) spojite dvije png datoteke (*py\_logo.png* i *changed\_py\_logo.png*):
  - a) Isprobajte različite vrijednosti transparentnosti, npr. 0,2 i 0,5.
  - b) Nakon toga, načinite kompozit (*image composite*) od istih dviju png datoteka.
  - c) Načinite kompozitnu sliku tako da masku čini datoteka *pillow\_logo.png* (u ovom postupku dimenzije svih slika u kompozitu je najprije potrebno uskladiti).
- 12) Tehnikom maskiranja načinite kompoziciju slika *ivancica.jpg* i *livada.jpg*. Kompozicija mora izgledati kao što je prikazano na slici ispod. Masku nacrtajte koristeći crnu pozadinu na kojoj će biti nacrtan bijeli krug. Za crtanje koristite metodu *ellipse()* iz *ImageDraw*.

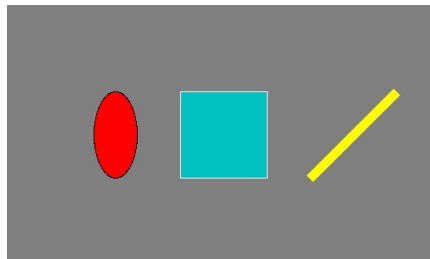
PREDLOŽAK:



KORDINATE MASKE:



- 13) Odredite boje svih piksela od kojih se sastoji slika *GIF-16boja.gif* i *znakovi.png* koristeći metodu *getcolors()*. Usporedite i objasnite dobivene rezultate. Odredite boju piksela koji se nalazi na koordinati (199, 199).
- 14) Koristeći modul PIL nacrtajte sliku u RGB sustavu boja i rezoluciji 500 x 300 tako da ispuna bitmape bude u boji R: 128, G: 128 i B: 128. Za crtanje koristite metode *Image.new* i *ImageDraw*. Na tako kreirano platno dodajte i geometrijske likove, a za crtanje koristite *draw.ellipse*, *draw.rectangle* i *draw.line*. Dimenzije likova odredite proizvoljno, a boje neka odgovaraju onima koje su prikazane na predlošku ispod. Kao rezultat, nacrtanu sliku pohranite kao JPEG slikovnu datoteku s faktorom kvalitete od 95%.



- 15) Napišite program koji će izraziti u postotku koliko piksela je na slici *a-slovo.jpg* crnih, a koliko bijelih.

