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 2025-1

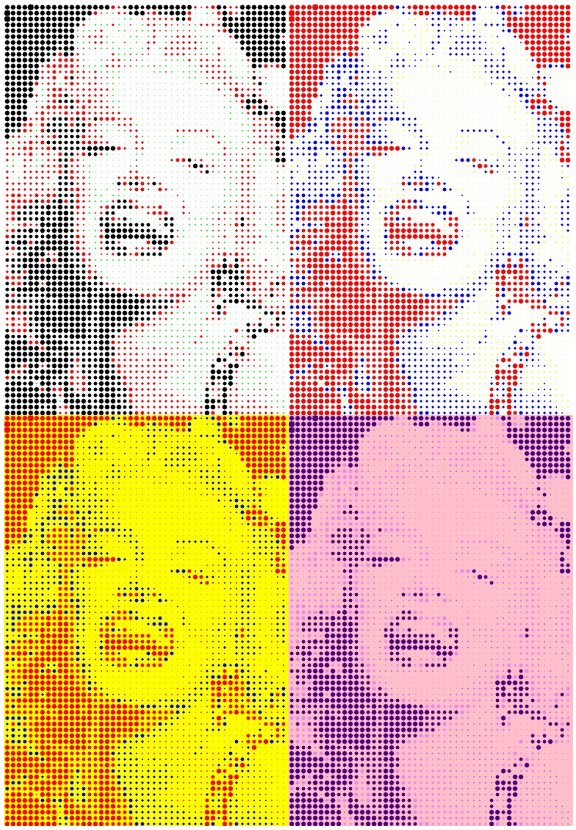
Pripravila: Luka Škrlj & Tomaž Vrtovec

Naloga

Pri obdelavi slik pogosto naletimo na nalogo ustvarjanja umetniških filtrov, ki simulirajo specifične umetniške stile. Eden takih stilov je pop-art, ki ga je priljubljen umetnik Andy Warhol s svojim prepoznavnim barvnim slogom povzdignil v ikonografijo moderne umetnosti. Pop-art stil vključuje močne barve, preproste oblike in večkratne ponovitve slikovnih elementov.
Vaša naloga je ustvariti pop art filter.

Vhodna RGB slika

Izhodna RGB slika



Dana je dvodimenzionalna (2D) barvna (RGB) slika marlyn-monroe-484x699-08bit.raw, velikosti $X \times Y = 484 \times 699$ slikovnih elementov z 8 biti na slikovni element (vrsta podatkov `'uint8'`).

1. (10) Naložite dano sliko in napišite funkcijo, ki iz vhodne slike ustvari sliko z enakomerno porazdeljenim histogramom svetlosti (ang. histogram equalization):

```
def equalize_image(iImage):
    # ...
    return oImage
```

Vhodni argument `iImage` predstavlja vhodno RGB sliko, izhodni argument `oImage` pa predstavlja sliko z enakomerno porazdeljenim histogramom svetlosti. (Namig: Za izravnavo histograma potrebujemo črno belo sliko)

Prikažite dano sliko pred in po filtriranju z izravnavo histograma v črno beli barvi.

2. (10) Implementirajte funkcijo, ki nariše pobarvan krog na kanvas:

```
def draw_circle(canvas, center, radius, color):
    # ...
    return canvas
```

Funkcija sprejme naslednje vhodne argumente:

- `canvas`: (2D array) kanvas, na katerega se bo risalo krog.
- `center`: (x, y) koordinati središča kroga.
- `radius`: (int) Polmer kroga kot število slikovnih elementov.
- `color`: (R, G, B) barva kroga.

Funkcija naj spremeni vhodni kanvas, tako da nastavi barve vseh slikovnih elementov, katerih evklidska razdalja do središča je manjša ali enaka polmeru kroga. Ostal del kanvasa ostane nespremenjen. Funkcija naj vrne spremenjen kanvas. (Namig: Na vhodnem kanvasu lahko preverimo le slikovne elemente znotraj kvadrata okoli centra)

Preizkusite funkcijo tako, da na prazno bel kanvas velikosti 100×100 **narišete črn krog s središčem** `(50, 50)` **in polmerom** `25`**.**

3. (10) Implementirajte funkcijo za ustvarjanje pop-art slike:

```
def create_pop_art(iImage, max_dot_radius, background_color,
   dot_colors):
    # ...
    return oImage
```

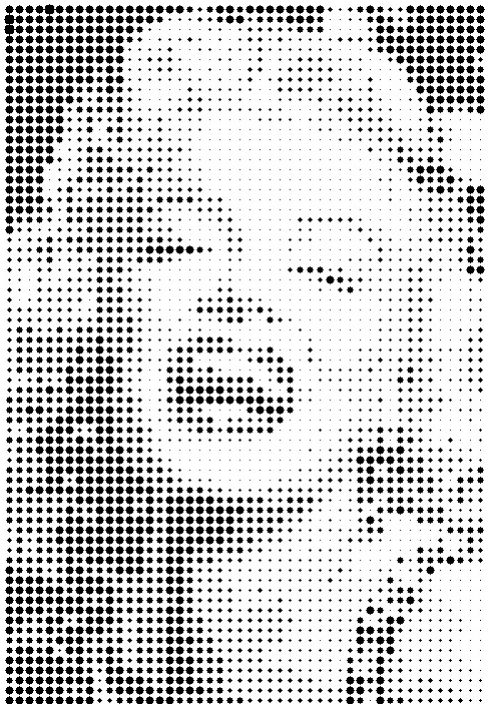
Funkcija sprejme naslednje vhodne argumente:

- `iImage`: vhodna sivinska slika,
- `max_dot_radius`: največja velikost pike,
- `background_color`: barva ozadja,
- `dot_colors`: seznam barv za pike.

Najprej funkcija ustvari prazen kanvas enake velikosti kot vhodna slika in ga zapolni z barvo ozadja. Nato prehaja čez sliko v enakomernih korakih, določenih z $2 \times$ `max_dot_radius`, tako da se pike ne prekrivajo. Na vsakem vzorčenem mestu določi velikost pike glede na svetlost slikovnega elementa – temnejši kot je slikovni element, večja naj bo pika, od 0 do največje velikosti `max_dot_radius`. Velikost pike naj se linearno spreminja z svetlostjo slikovnega elementa. Na izračunano lokacijo funkcija s pomočjo `draw_circle` nariše krog ustrezne barve in velikosti. Po končani obdelavi funkcija vrne generirano pop-art sliko.

Preverite delovanje funkcije z uporabo filtrirane slike iz prve naloge kot vhodne slike, bele barve ozadja, črne barve pik in maksimalnega polmera pik enako `5`**.**

Izhodna pop-art slika



4. (5) Spremenite funkcijo `create_pop_art()` tako, da omogoča vnos treh barv prek argumenta `dot_colors`. Funkcija naj glede na svetlost slikovnega elementa izbere eno od treh barv: če je svetlost manjša od 85, izbere prvo barvo, če je med 85 in 170, drugo, sicer pa tretjo.

   Sestavite 4 slike v 2x2 mrežo z uporabo različnih barvnih shem.

```
color_choices_rgb = [
    {"background": (255, 255, 255),
    "dots": [(0, 0, 0), (255, 0, 0), (0, 255, 0)]},
    {"background": (255, 255, 255),
    "dots": [(255, 0, 0), (0, 0, 255), (255, 255, 0)]},
    {"background": (255, 255, 0),
    "dots": [(255, 0, 0), (0, 0, 255), (255, 0, 255)]},
    {"background": (255, 192, 203),
    "dots": [(75, 0, 130), (238, 130, 238), (147, 112, 219)]}]
```