

# Haladó Programozás beadandó

## Képek módosítására script készítése

**Képmanipuláció: színcsere – szürkeárnyalatosítás – invertálás**

Készítette: Juhász Dávid, Versics Ádám

A program ellenőrzéséhez ChatGPT v

## **1. A program célja**

A program célja egy egyszerű, parancssori képmanipulációs eszköz megvalósítása, amely a felhasználó által megadott képfájlon az alábbi műveletek egyikét hajtja végre:

- Egy megadott szín kicserélése egy másik színre, tetszőleges tolerancia mellett.
- A kép szürkeárnyalatossá alakítása.
- A kép színeinek invertálása (negatív kép készítése), az esetleges átlátszóság (alpha csatorna) megőrzése mellett.

A program interaktív, szöveges menüt használ: a felhasználó megadja a bemeneti és kimeneti képfájl nevét, kiválasztja a műveletet, színcsere esetén pedig a forrás- és cél-színt, illetve a toleranciát.

## **2. Használt technológiák és könyvtárak**

### **Python**

A programozási nyelv, amelyben az alkalmazás készült. A standard könyvtárak közül a sys modult használja a program futásának megszakítására (hibák esetén).

### **Pillow (PIL)**

A Pillow könyvtár felel a képfájlok kezeléséért:

- képfájlok megnyitása (Image.open),
- képmódok kezelése (convert, split, merge),
- színek invertálása (ImageOps.invert),
- kép szürkeárnyalatossá alakítása (ImageOps.grayscale),
- a feldolgozott kép mentése (Image.save).

### **sys**

A sys modul használatával a program hibás bemenet vagy hiányzó kötelező adat esetén kulturáltan leállítható (sys.exit), egyértelmű hibaüzenettel.

### **3. A program felépítése és működése**

A program fő logikája a main() függvényben található. A szkript a szokásos Python belépési ponttal indul:

```
if __name__ == "__main__":
    main()
```

A futás során a program először betölti a felhasználó által megadott képfájlt, majd menüszerűen felkínálja a választható képmanipulációs műveleteket. A kiválasztott opción alapján a megfelelő átalakítást végzi el.

#### 3.1. Színbeviteli segédfüggvények

color\_input(text)

Feladata egy nyers, szöveges színmegadás (például "255 0 0" vagy "255, 0, 0") értelmezése, és abból érvényes RGB szín létrehozása. A függvény:

- eltávolítja a felesleges szóközöket;
- szóközre cseréli a vesszőt, majd felosztja a bemenetet;
- ellenőrzi, hogy pontosan három komponens (R, G, B) érkezett;
- megpróbálja az értékeket egésszé alakítani;
- ellenőrzi, hogy minden komponens 0 és 255 közötti tartományba esik.

Hibás formátum vagy tartomány esetén a függvény ValueError kivételt dob, amelyből jól látható, mi volt a probléma (pl. hiányzó komponens vagy érvénytelen szám).

color\_prompt(prompt)

Ez a függvény a felhasználótól kér be színkódot, és addig engedi újra próbálkozni, amíg érvényes RGB színt nem ad meg, vagy amíg a felhasználó úgy nem dönt, hogy kilép. A függvény:

- bekéri a színt input() segítségével;
- meghívja a color\_input() függvényt az értelmezéshez;
- hiba esetén tájékoztatja a felhasználót, majd rákérdez, hogy szeretné-e újra megadni a színt;
- ha nem, akkor sys.exit() segítségével megszakítja a programot.

### 3.2. Színcsere: replace\_color(img, src\_color, dst\_color, tolerance)

A replace\_color függvény feladata, hogy az img képen az src\_color-hoz a megadott tolerancián belül eső pixelek színét lecserélje dst\_color-ra.

#### A működés fő lépései:

1. A tolerance paraméter normalizálása (legalább 0, egész szám), majd négyzetének kiszámítása (tol2).
2. A bemenetként kapott forrás- és cél-szín RGB komponenseinek szétbontása.
3. A kép RGBA módra konvertálása, hogy minden pixel rendelkezzen alpha csatornával.
4. A kép összes pixelének bejárása, és a színkülönbség kiszámítása az aktuális pixel és az src\_color között Euklideszi távolság négyzeteként ( $(dr^2 + dg^2 + db^2)$ ).
5. Ha a kapott érték kisebb vagy egyenlő a tol2 küszöbnél, a pixel színe dst\_color-ra cserélődik, az alpha érték változatlanul marad; ellenkező esetben a pixel eredeti színe marad.
6. Az így előállított pixeladatok visszaírása a képre, majd a módosított képobjektum visszaadása.

### 3.3. Főprogram – main()

#### Képfájl beolvasása

A program a futás elején bekéri a bemeneti képfájl elérési útját és nevét. Üres bemenet esetén a program hibával leáll. A megadott fájlt Image.open() segítségével próbálja megnyitni, hiba esetén szintén egyértelmű hibaüzenettel fejeződik be.

#### Művelet kiválasztása

A program három műveletet kínál fel numerikus menüvel:

- 1) Szín csere
- 2) Szürke árnyalatosítás
- 3) Színek invertálása

A felhasználó addig kap újabb kérdést, amíg érvényes (1, 2 vagy 3) választ nem ad.

### Színcsere ág (1)

Színcsere választása esetén a program a color\_prompt() segítségével először a lecserélendő színt (src\_color), majd a cél-színt (dst\_color) kéri be. Ezután a tolerancia értékének megadására kérdez rá. Ha a felhasználó üresen hagyja a mezőt, egy alapértelmezett (40) értékkel dolgozik tovább. Hibás bemenet esetén figyelmezteti a felhasználót, és szintén az alapértelmezett értéket használja. Ezt követően meghívja a replace\_color() függvényt, amely elvégzi a színcserét.

### Szürkeárnyalatosítás ág (2)

Ebben az esetben a program a Pillow ImageOps.grayscale() függvényével alakítja a képet szürkeárnyalatos módra. Az eredmény egy egycsatornás kép, ahol minden pixel egy intenzitásértéket tartalmaz.

### Invertálás ág (3)

Színek invertálása esetén a program külön kezeli az átlátszóságot tartalmazó (RGBA) és az egyéb képmódokat:

- RGBA kép esetén az RGB csatornákat elválasztja az alpha-tól, csak az RGB-t invertálja, majd az eredményt az eredeti alpha csatornával együtt fűzi vissza.
- Egyéb módok esetén a képet RGB-vé konvertálja, invertálja a színeket, és az így kapott eredményt használja.

### Kimeneti fájl mentése

A manipuláció befejezése után a program bekéri a kimeneti fájl nevét és elérési útját. Üres bemenet esetén hibával leáll. A result\_img.save(output\_path) hívás menti el a képet. Sikeres mentésről visszajelzést ad, hiba esetén részletes hibaüzenetet jelenít meg.

## **4. Futtatás és követelmények**

Szükséges csomagok

A program futtatásához az alábbiakra van szükség:

- Python 3.x
- Pillow (pip install pillow)

### **Futtatás lépései :**

- A forráskódot tartalmazó .py fájlt (például kepmodosito.py) mentsük el egy tetszőleges mappába.
- Telepítsük a Pillow könyvtárat, ha még nincs fent (pip install pillow).
- Nyissunk egy parancssort, és lépjünk a fájlt tartalmazó mappába.
- Indítsuk el a programot: python kepmodosito.py
- Kövessük a program utasításait: adjuk meg a bemeneti és kimeneti képfájl nevét, válasszuk ki a műveletet, színcsere esetén pedig a szükséges színeket és toleranciát.

## **5. Példafuttatás**

Egy lehetséges próba futtatás röviden:

- Bemeneti kép: input.png
- Művelet: 1) Szín csere
- Forrás-szín: 255,0,0
- Cél-szín: 0,255,0
- Tolerancia: alapértelmezett (Enter)
- Kimeneti kép: output.png

A futás végén a program jelzi, hogy a kép sikeresen elmentésre került.