Prooktatás projektfeladat Tari Enikő

Developer specifikáció

Cél:

A program képes képfájlok betöltésére internetről vagy számítógépről, szükség esetén RGB formátumba konvertálja őket, majd egyszerű képfeldolgozási műveleteket végez. Az eredményt PPM formátumban egy külön mappába menti, és a parancssorból könnyen futtatható.

Képfeldolgozás elméleti háttér:

A digitális képfeldolgozás célja, hogy matematikai műveletekkel alakítsuk át vagy javítsuk a képeket, illetve új információt nyerjünk ki belőlük. A képek pixelekből állnak, amelyek szín- vagy szürkeárnyalat-értéket hordoznak. Az egyszerű műveletek pixelenkénti aritmetikai átalakítások, míg az összetettebb szűrések (pl. elmosás, élkeresés) konvolúciós mátrixokkal történnek, a szomszédos pixelek súlyozott átlagát felhasználva.

Architektúra

1. **class PictureLoader**
   Ez az osztály tölti le URL alapján az internetről vagy elérési útvonal alapján a gépről a képet majd előkészíti őket a további munkára.

- *download_from_url(url: str) -> Image:*

Ha a megadott URL egy kép letölti, ha nem akkor kinyeri a megadott URL-en található képet. Végül a kinyert képet RGB-be konvertálja.

Az funkció a BeatifulSoup és requests modulokat használja az URL-ek feldolgozására.

- *load_from_computer(path: str) -> Image:*

A megadott elérési útvonalból a számítógépen a képet lekéri, betölti és RGB-be alakítja.

- *save_ppm(image: Image, filename: str) -> None:*

A program minden új kép fájlt a munkakönyvtárban létrehozott új PictureEditor_images mappába menti a megadott fájlnéven PPM formátumban.

Hibakezelés:

- Ellenőrzi, hogy létezik-e megadott elérési útvonal.
- Ellenőrzi, hogy a megadott elérési útvonal kép-e.

2. class PictureEdit

Ez az osztály képes a betöltött és előkészített képből képtranszformációk elvégzésére.

Aritmetikai műveletek: Futtatás során végigmegy a megadott kép összes pixelén és minden pixel 3 koordinátáján (RGB) elvégzi a műveletet, úgy, hogy a konstans helyére a paraméterben megadott értékeket helyettesíti.

- *brightness(image: Image, constant:int|float) -> Image:*

$$I' = I + c, \text{ ahol konstans, lehet pozitív/negatív}$$

- *contrast(image: Image, alpha: float) -> Image:*

$$I' = \alpha * (I - mid) + mid, \text{ ahol } mid = 128 \ (255/2).$$

- *invert(image: Image) -> Image:*

$$I' = max - I, \text{ ahol } max = 255$$

Konvolúciós mátrix műveletek: A program végigmegy a kép összes pixelén, az RGB csatornákat különválasztva. Minden pixelhez létrehoz egy 3×3-as ablakot a környező pixelekből, majd az ablak elemeit sorban összeszorozza a konvolúciós mátrix (kernel) megfelelő elemeivel, és ezek súlyozott átlagát veszi.

- *blurring(image: Image) -> Image:*

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

- *edge_detection(image: Image) -> Image:*

Vízszintes (X irány, vízszintes élek kiemelése):

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Függőleges (Y irány, függőleges élek kiemelése):

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- *sharpening(image: Image) -> Image:*

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Hibakezelés:

- A pixel értéke nem mehet 0 alá és 255 felé, egyik műveletben sem.
- A konvolúciós szűréseknél a kép szélén a hiányzó pixelek helyén a szélső pixelek értékét használjuk.

3. class CommandLineInterface

Ez az osztály biztosítja a program parancssori vezérlését. Az osztály az **argparse** modult használja az argumentumok feldolgozására.

- *build_parser()*

Létrehozza az argumentumokat kezelő parser objektumot. Ebben vannak definiálva a parancssori kapcsolók és azok leírásai.

- *main() -> None*

A program belépési pontja. Beolvassa a parancssori argumentumokat, meghívja a megfelelő PictureLoader és PictureEdit metódusokat, majd az eredményt elmenti.

Elérhető argumentumok:

--input PATH	A feldolgozandó kép elérési útvonala a számítógépen.
--url URL	A feldolgozandó kép URL-je, ha a felhasználó internetről szeretne képet letölteni.
--op OPERATION	A választott művelet (brightness, contrast, invert, blur, edge, sharpen).
--value N	Egész szám, a fényerő változtatásához (pozitív = világosítás, negatív = sötétítés).
--alpha N	Valós szám, a kontraszt növeléséhez vagy csökkentéséhez.
--output PATH	Az eredményül kapott kép mentési helye és fájlneve.
-h, --help	Súgó, amely megmutatja a program összes lehetséges paraméterét.

Használati példák:

python PictureEditor.py --input "kep.jpg" --op invert --output "negativ.ppm"

python PictureEditor.py --input "kep.png" --op brightness --value 30 --output "vilagos.ppm"

python PictureEditor.py --input "kep.png" --op edge --output "kontur.ppm"

Hibakezelés:

- Rossz típusú argumentum megadását jelzi a program.
- Ha nincs megadva output név, automatikusan {eredeti_nev}_{op}.ppm néven menti.

Megjegyzések:

- A program a SOLID, KISS, DRY és YAGNI elveket követve készült, így moduláris, karbantartható és bővíthető.
- Nem számítunk arra, hogy a memóriaigény túl nagy lesz a képek feldolgozása során.
- Az alkalmazás bővíthető a jövőben további képszerkeztési eljárásokkal és egyszerre több kép feldolgozásával vagy automatizálással.