



Neumann János Egyetem  
Műszaki és Informatikai  
Kar

# Képszerkesztő alkalmazás

## Picture Editor Application

Rácz Tamás Csaba  
L2J7OU  
Mérnökinformatikus  
5.félév

2025

# Tartalomjegyzék

<b>BEVEZETÉS .....</b>	<b>7</b>
<b>PROGRAM FŐ FELÉPÍTÉSE .....</b>	<b>9</b>
1.1.    INIT.....	10
1.2.    SETUP_UI[2].....	10
1.3.    SETUP_BINDINGS .....	10
2.1.    ON_RIGHT_CLICK.....	11
2.2.    ON_RIGHT_DRAG.....	11
2.3.    ON_RIGHT_RELEASE.....	11
2.4.    ON_CLICK[1] .....	11
2.5.    ON_DRAG[1] .....	11
2.6.    ON_RELEASE[1].....	11
2.7.    ON_MOUSEWHEEL.....	12
3.1.    TOGGLE_DRAWING_MODE .....	13
3.2.    UPDATE_BRUSH_SIZE .....	13
3.3.    CHOOSE_DRAW_COLOR .....	13
3.4.    CHOOSE_TEXT_COLOR .....	13
3.5.    FINALIZE_DRAWING.....	13
3.6.    PLACE_TEXT_AT.....	13
4.1.    OPEN_IMAGE .....	14
4.2.    SAVE_STATE.....	14
4.3.    SYNC_DRAWING_LAYER_SIZE .....	14
4.4.    UNDO .....	14
4.5.    REDO .....	14
4.6.    RESET_IMAGE.....	14
4.7.    UPDATE_IMAGE_DISPLAY .....	15
4.8.    ZOOM .....	15
4.9.    SAVE_IMAGE.....	15
5.1.    REMOVE_BACKGROUND_GRABCUT[1] .....	16
5.2.    PROCESS_GRABCUT_REMOVAL[6].....	16
6.1.    EXTRACT_TEXT[5][4].....	17
7.1.    APPLY_FILTER.....	18
7.2.    APPLY_GRAYSCALE .....	18
7.3.    APPLY_BLUR .....	18
7.4.    APPLY_SHARPEN .....	18
7.5.    APPLY_SEPIA .....	18
7.6.    APPLY_VINTAGE .....	18
8.1.    ROTATE .....	19
8.2.    FLIP_HORIZONTAL .....	19
8.3.    RESIZE_IMAGE.....	19
9.1.    MAIN_BLOCK .....	20
<b>PROGRAM FUTTATÁSA ÉS TESZTELÉSE.....</b>	<b>21</b>
<b>FELHASZNÁLT ANYAGOK.....</b>	<b>32</b>

## Bevezetés

A projekt létrehozásának fő célja egy olyan átfogó képszerkesztő alkalmazás kifejlesztése volt, amely felhasználóbarát, modern felülettel rendelkezik, és széles körű szerkesztési lehetőségeket kínál. A projekt során célom volt, hogy bemutassam a Python programozási nyelv sokoldalúságát és a különböző könyvtárak együttes használatának lehetőségeit.

A program a Visual Studio Code integrált fejlesztői környezetben lett kidolgozva és főként a Python nyelvet használja gyors fejleszthetősége és sokoldalúsága miatt.

Python 3.13.9-ben készült az alábbi könyvtárakat felhasználva:

- CustomTkinter: A CustomTkinter egy modern, testreszabható Tkinter alapú GUI könyvtár, amely a projekt felhasználói felületének alapját képezi. Azért esett a választás erre a könyvtárra, mert beépített dark mode támogatást nyújt, vizuálisan vonzó, lekerekített sarkú komponenseket biztosít, és egyszerű színséma-kezelést tesz lehetővé. A hagyományos Tkinter-rel ellentétben modern kinézetű gombokat, csúszkákat és beviteli mezőket kínál, amelyek jobban illeszkednek a mai felhasználói elvárásokhoz. Az alkalmazásban az összes vezérlőelem (gombok, checkboxok, sliderek, entry mezők) CustomTkinter komponensként került implementálásra.
- Tkinter: A standard Python GUI könyvtár, amely a CustomTkinter alapját képezi. Bár az alkalmazás főként CustomTkinter komponenseket használ, bizonyos alap funkcionálisokhoz (Canvas widget, fájldialógusok, színválasztó, egyszerű input dialógusok) közvetlenül a Tkinter modulokat hívja meg. A Canvas widget különösen fontos, mert ez biztosítja a nagy felbontású képek megjelenítését görgetősávokkal, valamint az egéresemények pontos kezelését. A filedialog, messagebox, colorchooser és simpledialog modulok standard dialógusablakokat biztosítanak fájlműveletek, értesítések és felhasználói input bekérése céljából.

- Open-CV(cv2): egy átfogó számítógépes látás könyvtár, amely az alkalmazásban a fejlett képfeldolgozási műveletekért felel. A modul NumPy tömböket használ belső reprezentációként, ami hatékony képmanipulációt tesz lehetővé. Az alkalmazásban az OpenCV-t szűrők alkalmazására (Gauss-féle elmosás, élesítés, színtranszformációk), a GrabCut algoritmus futtatására (intelligens háttér-eltávolítás), valamint különböző színtér-konverziókra használjuk (RGB, BGR, szürkeárnyalatos, RGBA).
- Numpy: A NumPy egy alapvető Python könyvtár tudományos számításokhoz, amely hatékony többdimenziós tömb műveleteket biztosít. Az alkalmazásban a NumPy-t képadatok tömb formátumú tárolására használjuk, mivel a PIL Image objektumok könnyen konvertálhatók NumPy tömbökké és vissza. A tömbműveletek lehetővé teszik a pixelszintű manipulációkat, mint például a zajhuzzáadás a vintage szűrőben vagy a maszk alkalmazása a háttér-eltávolításnál.
- Pytesseract: A Pytesseract a Tesseract OCR (Optical Character Recognition) motor Python wrapper-je, amely szövegfelismerési funkcionálitást biztosít. Az alkalmazás ezt a modult használja a képeken található szövegek automatikus kinyerésére.
- Pillow: Képkezelés alapvető eszköze az alkalmazásban. Ez a modul felelős a képfájlok betöltéséért, mentéséért és alapvető manipulációjáért.
- Os: Az os modul operációs rendszer függő funkcionálitást biztosít. Az alkalmazásban elsősorban fájlútvonalak kezelésére használjuk

A felsorolt könyvtárak szorosan együttműködnek az alkalmazásban. A tipikus munkafolyamat a következő: a PIL betölti a képet fájlból, NumPy tömbé konvertáljuk az OpenCV feldolgozáshoz, az OpenCV alkalmazza a szűrőket vagy AI algoritmusokat, az eredményt visszaalakítjuk PIL Image-dzé, majd a CustomTkinter/Tkinter felületen jelenítjük meg.

## **Program fő felépítése**

A program egy PictureEditor osztályt tartalmaz amiben függvényenként vannak megvalósítva a különböző funkciók.

A következő alfejezetekben kerülnek kifejtésre az összes függvények által megvalósított funkciók.

## **1. Inicializálás és felhasználói felület**

### **1.1. init**

Az alkalmazás konstruktora, amely beállítja az ablak alapvető paramétereit (cím, méret, minimális méret), inicializálja a képkezelési változókat (original\_image, current\_image, display\_image), az előzmények kezelését (history, history\_index), a nagyítási és mozgatási változókat (zoom\_factor, pan\_x, pan\_y), valamint a rajzolási és szerkesztési módok paramétereit. A metódus végén meghívja a setup\_ui(), setup\_bindings() és update\_image\_display() függvényeket.

### **1.2. Setup\_ui[2]**

Ez a függvény felel a felhasználói felületért. Ide tartoznak a gombok, feliratok és eszközök, amik viselkedése és tulajdonsága a későbbi függvényekben kerül kifejtésre. Főként a CustomTkinter könyvtár használatával lett a felület megtervezve modern kinézete és könnyen használhatósága miatt. Létrehozza az oldalsávot a vezérlőelemekkel, a rajzolási eszközök paneljét, a szűrők és transzformációs gombok listáját, valamint a központi választatósávokkal.

### **1.3. Setup\_bindings**

Funkciók billentyűzet gombokhoz és egéreseményekhez való rendeléséért felel. A funkciók között szerepel a bal egérgomb események kezelése rajzoláshoz, jobb egérgomb a képmozgatáshoz, egérgörgő a nagyításhoz, valamint billentyűparancsok a visszavonáshoz (Ctrl+Z), újra végrehajtáshoz (Ctrl+Y), kép megnyitásához (Ctrl+O) és mentéséhez (Ctrl+S).

## **2. Egéresemények kezelése**

### **2.1. On\_right\_click**

Kezeli a jobb egérgomb lenyomását a képmozgatási funkció (panning) elindításához. Beállítja az is\_panning flag értékét True-ra, elmenti az egér kezdőpozíóját, és megváltoztatja a kurzor ikonját négyirányú nyílra.

### **2.2. On\_right\_drag**

Kezeli a jobb egérgomb lenyomva tartása közben történő mozgatást. Kiszámítja az egér elmozdulását az előző pozícióhoz képest, frissíti a pan\_x és pan\_y értékeket, majd újrarendezze a képet az új pozícióban az update\_image\_display() meghívásával.

### **2.3. On\_right\_release**

Kezeli a jobb egérgomb felengedését. Kikapcsolja a panning módot az is\_panning False értékre állításával, és visszaállítja az egérkonzort az alapértelmezett állapotba.

### **2.4. On\_click[1]**

A bal egérgomb lenyomásának kezelése. Ha háttér-eltávolítási mód aktív, elmenti a kijelölés kezdőpontját. Ha szöveg mód aktív, meghívja a place\_text\_at() metódust. Ha rajzolási mód aktív, inicializálja a rajzolást: radír esetén közvetlenül a permanent\_drawings rétegen dolgozik, más eszközöknel új draw\_overlay réteget hoz létre. Elmenti a kezdőpozíciót és frissíti a megjelenést.

### **2.5. On\_drag[1]**

A bal egérgomb lenyomva tartása közben történő mozgatás kezelése. Háttér-eltávolítási módban vizualizálja a kijelölési téglalapot. Rajzolási módban folyamatosan frissíti a rajzot: radír esetén átlátszó pixeleket állít be a permanent\_drawings rétegen, ecset esetén vonalakat és köröket rajzol a draw\_overlay rétegre.

### **2.6. On\_release[1]**

A bal egérgomb felengedésének kezelése. Háttér-eltávolítási módban feldolgozza a kijelölt területet a process\_grabcut\_removal() metódussal. Rajzolási módban végezzi az alakzatokat: vonal, téglalap vagy kör esetén az alakzatot a draw\_overlay rétegre rajzolja, majd a finalize\_drawing() metódussal egyesíti a permanent\_drawings réteggel.

## **2.7. On\_mousewheel**

Az egér görgetőjét használva tudunk nagyítani vagy kicsinyíteni a képen. A metódus detektálja a görgetés irányát: felfelé görgetés esetén 1.1-szeres nagyítást, lefelé görgetés esetén 0.9-szeres kicsinyítést hajt végre a zoom() függvény meghívásával.

### **3. Rajzolási funkciók**

#### **3.1. Toggle\_drawing\_mode**

A rajzolási funkció be- és kikapcsolását kezeli. Szinkronizálja a drawing változót a draw\_enabled checkbox állapotával. Aktiválás esetén információs ablakot jelenít meg, amely tájékoztatja a felhasználót az aktuálisan kiválasztott eszkösről és a használati útmutatóról.

#### **3.2. Update\_brush\_size**

A rajzolási eszköz méretének dinamikus módosítását valósítja meg. A csúszka értékét egész számmá konvertálja, frissíti a draw\_size változót, majd frissíti a méret címke szövegét, amely vizuálisan jeleníti meg az aktuális méretet pixelben.

#### **3.3. Choose\_draw\_color**

Színválasztó dialógust nyit meg a rajzolási eszközökhöz. A colorchooser.askcolor() standard tkinter színválasztó ablakot jelenít meg, és ha a felhasználó kiválasztott egy színt, a draw\_color változó frissítésre kerül a hexadecimális színkóddal.

#### **3.4. Choose\_text\_color**

Hasonló a choose\_draw\_color() metódushoz, de kifejezetten a szöveg elhelyezési funkcióhoz. Színválasztó ablakot nyit meg, és a kiválasztott színnel frissíti a text\_color változót, amely a képre helyezendő szöveg színét határozza meg.

#### **3.5. Finalize\_drawing**

Véglegesíti a rajzolt elemeket. Ha létezik draw\_overlay réteg, inicializálja a permanent\_drawings réteget (ha még nem létezik), majd egyesíti az új rajzot a permanens réteggel alpha compositing használatával. Törli az ideiglenes overlay-t, elmenti az állapotot, és frissíti a megjelenést.

#### **3.6. Place\_text\_at**

Szöveget helyez el a képre a megadott koordinátákon. Bekéri a felhasználótól a betűméretet egy dialógusablakban, betölti az Arial betűtípust, majd létrehoz egy RGBA overlay réteget, amelyre a megadott színnel és körvonalazással kirajzolja a szöveget. Az overlay-t egyesíti a current\_image-dzsel, elmenti az állapotot, és frissíti a megjelenést.

## **4. Képkezelési alapműveletek**

### **4.1. Open\_image**

Fájlválasztó dialógust nyit meg képfájlok betöltéséhez (PNG, JPG, JPEG, BMP, TIFF, WEBP formátumok). A betöltött képet RGB formátumra konvertálja, inicializálja az original\_image és current\_image változókat, törli a korábbi rajzolt elemeket, és alaphelyzetbe állítja az előzményeket, nagyítást és eltolást. Sikeres betöltés esetén információs üzenetet jelenít meg.

### **4.2. Save\_state**

Elmenti az aktuális képállapotot az előzmények listájába. A metódus dictionary formátumban tárolja a current\_image NumPy tömbjét, a permanent\_drawings réteg másolatát és a has\_transparency flag értékét. Az előzmények lista maximum 30 elemet tárolhat, régebbi elemek automatikusan törlődnek.

### **4.3. Sync\_drawing\_layer\_size**

Biztosítja, hogy a permanent\_drawings réteg mérete megegyezzen a current\_image méretével. Ha eltérés van, új RGBA réteget hoz létre a megfelelő méretben, és a régi rajzolt elemeket átmásolja rá. Ez különösen fontos átméretezés, forgatás vagy más transzformációk után.

### **4.4. Undo**

Visszalép egy lépést az előzmények listájában, ha lehetséges. Csökkenti a history\_index értékét, helyreállítja a tárolt képállapotot (image, permanent\_drawings, has\_transparency), és frissíti a megjelenést. Támogatja mind a dictionary, mind a régi formátumú NumPy tömb előzményeket.

### **4.5. Redo**

Előrelép egy lépést az előzmények listájában, ha lehetséges. Növeli a history\_index értékét, helyreállítja a következő tárolt állapotot, és frissíti a megjelenést. Támogatja mind a dictionary, mind a régi formátumú előzményeket.

### **4.6. Reset\_image**

Visszaállítja a képet az eredetileg betöltött állapotba. A metódus törli az összes szerkesztési műveletet, beleértve a rajzolt elemeket, szűrőket és transzformációkat. Az előzmények lista is újrainicializálódik, így a visszavonási funkciók is alaphelyzetbe

kerülnek. A nagyítási és eltolási paraméterek szintén visszaállnak az alapértelmezett értékekre.

#### **4.7. Update\_image\_display**

Ez a metódus felelős a munkaterületen megjelenített kép valós idejű frissítéséért minden szerkesztési művelet, nagyítás/kicsinyítés vagy eltolás után. A függvény kezeli az átlátszó képek megjelenítését sakktábla mintás háttérrel, egyesíti a permanent\_drawings réteget az alapképpel, alkalmazza a zoom faktort, majd megjeleníti a végeredményt a vászonon. Ha nincs betöltött kép, egy szöveges útmutatót jelenít meg a felhasználónak.

#### **4.8. Zoom**

Megvalósítja magát a nagyítási funkciót. A metódus egy faktort kap paraméterként, amellyel megszorozza az aktuális zoom\_factor értékét. A nagyítás mértéke 0.1 és 10 közé van korlátozva a használhatóság érdekében. Ha a zoom faktor értéke megváltozik, a kép megjelenítése frissítésre kerül az update\_image\_display() meghívásával.

#### **4.9. Save\_image**

Ha el szeretnénk menteni a képet, akkor azt ennek a függvénynek a segítségével tudjuk megtenni a kívánt helyre. Az alap formátum PNG, de bármilyen formában el tudjuk menteni. A metódus fájlválasztó ablakot nyit meg, ahol megadható a mentési hely és formátum. JPEG mentés esetén a kép automatikusan RGB formátumra konvertálódik, átlátszó háttérrel rendelkező képek esetén fehér háttér kerül alkalmazásra. PNG formátumnál megmarad az átlátszóság támogatása.

## **5. Háttér-eltávolítás**

### **5.1. Remove\_background\_grabcut[1]**

Elindítja a GrabCut algoritmussal történő háttér-eltávolítási folyamatot. Ha vannak rajzolt elemek a képen, megkérdezi a felhasználót, hogy szeretné-e egyesíteni őket a képpel a feldolgozás előtt. Információs üzenetet jelenít meg a használatról, majd aktiválja a bg\_removal\_mode-ot, beállítja a bg\_removal\_method értékét 'grabcut'-ra, és megváltoztatja a kurzort keresztszálra a kijelölés megkönnyítéséhez.

### **5.2. Process\_grabcut\_removal[6]**

A GrabCut algoritmus tényleges végrehajtását végzi el a felhasználó által kijelölt területen. Egyesíti a rajzolt elemeket a képpel, ha szükséges, majd OpenCV-vel alkalmazza a GrabCut algoritmust a megadott bounding box alapján. A felhasználó választhat átlátszó háttteret vagy egyedi színű hátteret. Az algoritmus 5 iterációval dolgozik, és maszkot hoz létre, amely elválasztja az előteret a háttértől. A végeredményt RGBA vagy RGB formátumban tárolja.

## 6. OCR funkció

### 6.1. Extract\_text[5][4]

Szövegfelismerést hajt végre a képen Tesseract OCR motor használatával. A pytesseract.image\_to\_string() függvényt magyar és angol nyelvű szövegfelismerésre konfigurálja. Az eredményt egy új CTkToplevel ablakban jeleníti meg CTkTextbox komponensben, ahol a felhasználó megtekintheti és másolhatja a felismert szöveget. Ha nem sikerül szöveget felismerni, erről tájékoztatja a felhasználót.

## **7. Szűrők és képmanipuláció**

### **7.1. Apply\_filter**

Általános szűrőalkalmazó metódus, amely paraméterként egy függvényt kap. Elmenti az aktuális állapotot, konvertálja a képet OpenCV BGR formátumra, alkalmazza a paraméterként kapott szűrőfüggvényt, majd visszakonvertálja RGB formátumra. Szürkeárnyalatos eredmény esetén BGR formátumra alakítja. Szinkronizálja a rajzolási réteg méretét, és frissíti a megjelenítést.

### **7.2. Apply\_grayscale**

Szürkeárnyalatos szűrőt alkalmaz a képre. Az apply\_filter() metódust hívja meg egy lambda függvénnyel, amely az OpenCV cvtColor() függvényét használja BGR-ről szürkeárnyalatosra történő konverzióhoz.

### **7.3. Apply\_blur**

Gauss-féle elmosási szűrőt alkalmaz a képre. Az apply\_filter() metódust használja egy lambda függvénnyel, amely 25x25-ös kernel mérettel hívja meg a cv2.GaussianBlur() függvényt, közepes erősségű elmosást eredményezve.

### **7.4. Apply\_sharpen**

Élesítő szűrőt alkalmaz a képre konvolúciós mátrix használatával. Az apply\_filter() metódust hívja meg egy lambda függvénnyel, amely a cv2.filter2D() függvényt használja egy 3x3-as kernel mátrixszal. A mátrix középső értéke 9, körülötte -1 értékek, ami élesítő hatást eredményez.

### **7.5. Apply\_sepia**

Szépia tónusú (meleg, barnás) szűrőt alkalmaz a képre. Az apply\_filter() metódust használja egy lambda függvénnyel, amely színtranszformációs mátrixot alkalmaz a cv2.transform() függvénnyel. Az eredményt 1.1-szeresére erősíti, majd 0-255 közé korlátozza az értékeket.

### **7.6. Apply\_vintage**

Vintage (retró) effektust alkalmaz véletlenszerű zaj hozzáadásával. Az apply\_filter() metódust hívja meg egy lambda függvénnyel, amely 0-40 közötti véletlenszerű értékeket ad hozzá minden pixelhez cv2.add() használatával, így régi fényképek hatását kelti.

## **8. Transzformációs műveletek[3]**

### **8.1. Rotate**

Elforgatja a képet a megadott szögben. Elmenti az aktuális állapotot, majd a PIL Image rotate() metódusával forgatja el a képet negatív szöggel (az óramutató járásával megegyező irányban) az expand=True paraméterrel, így a teljes kép látható marad. Ha létezik permanent\_drawings réteg, azt is ugyanúgy elforgatja. Szinkronizálja a réteg méreteket, elmenti az új állapotot, és frissíti a megjelenést.

### **8.2. Flip\_horizontal**

Horizontális tükrözést hajt végre a képen. Elmenti az aktuális állapotot, majd az ImageOps.mirror() függvényel tükrözi a current\_image-t. Ha létezik permanent\_drawings réteg, azt is tükrözi ugyanígy. Frissíti a megjelenést az új tükrözött képpel.

### **8.3. Resize\_image**

Átméretezi a képet a felhasználó által megadott méretre. Két simpledialog.askinteger() hívással bekéri az új szélességet és magasságot, majd elmenti az aktuális állapotot. A PIL Image resize() metódusával LANCZOS újramintavételezéssel átméretezi a current\_image-t és a permanent\_drawings réteget is, ha létezik. Frissíti a megjelenést.

## 9. Fő program indítása

### 9.1. Main block

Az if `__name__ == "__main__"`: blokkban létrehozza a PictureEditor osztály példányát, és elindítja a Tkinter főciklust a `mainloop()` metódus meghívásával. Ez biztosítja, hogy az alkalmazás csak akkor induljon el, ha a script közvetlenül fut, és ne importálás esetén.

- [1] A megjelölt funkciók a Claude Sonnet 4.5 nagy nyelvi model használatával valósultak meg részben. Pontosan a program 237-266,291-315,354-372 sorokban lévő kód lett általa írva külön módosítás nélkül.
- [2] Customtkinter dokumentáció felhasználásával
- [3] Pillow dokumentáció felhasználásával
- [4] Tessseract dokumentáció felhasználásával
- [5] Pytesseract dokumentáció felhasználásával
- [6] A megjelölt funkciók az Open-CV dokumentációban megadott kód módosításával és Claude Sonnet 4.5 model használatával készült(656-674 sor). Az Open-CV dokumentáció megtalálható a „Felhasznált anyagok” fejezetben.

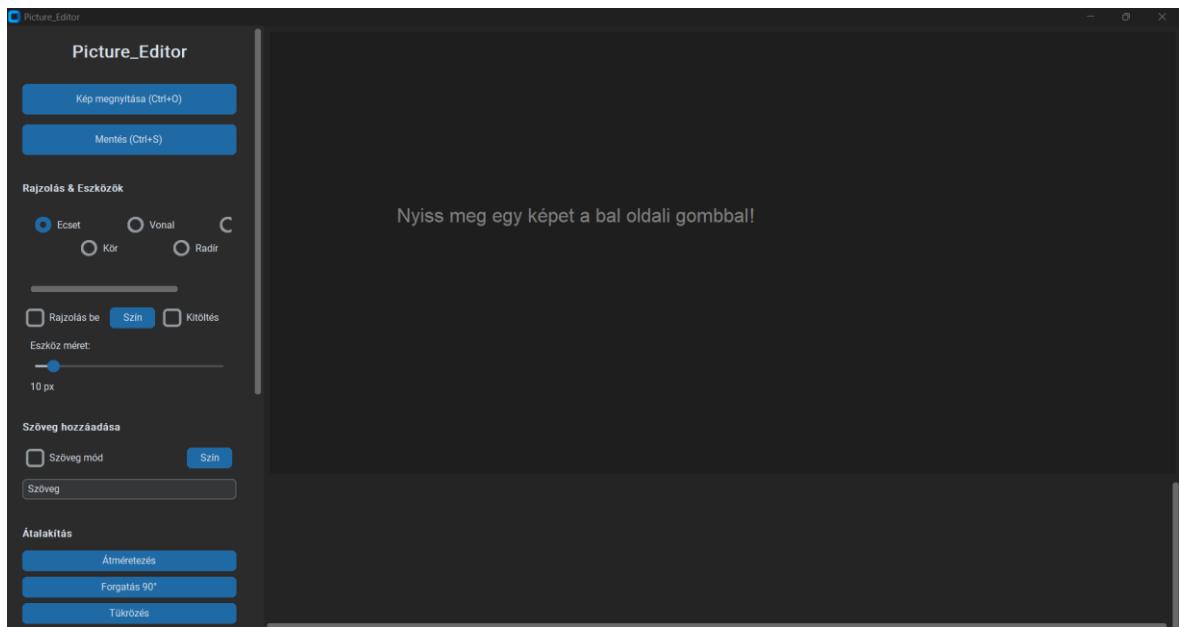
## Program futtatása és tesztelése

A projekt Githubról letölthető: [https://github.com/Tom-117/Picture\\_Editor](https://github.com/Tom-117/Picture_Editor)

Letöltés után nyissuk meg a projekt mappát a használni kívánt fejlesztői környezetben és töltsük le a requirements.txt fájlban lévő program függőségeket, például a *pip install -r requirements.txt* parancssal. Ahhoz hogy lehessen használni a szövegkinyerés funkciót le kell tölteni a Tesseract-OCR szövegfelismerő programot és bemásolni az elérési útját a „pytesseract.pytesseract.tesseract\_cmd =” sorhoz. A Tesseract letöltéséhez vezető link a “Felhasznált anyagok” fejezetben található.

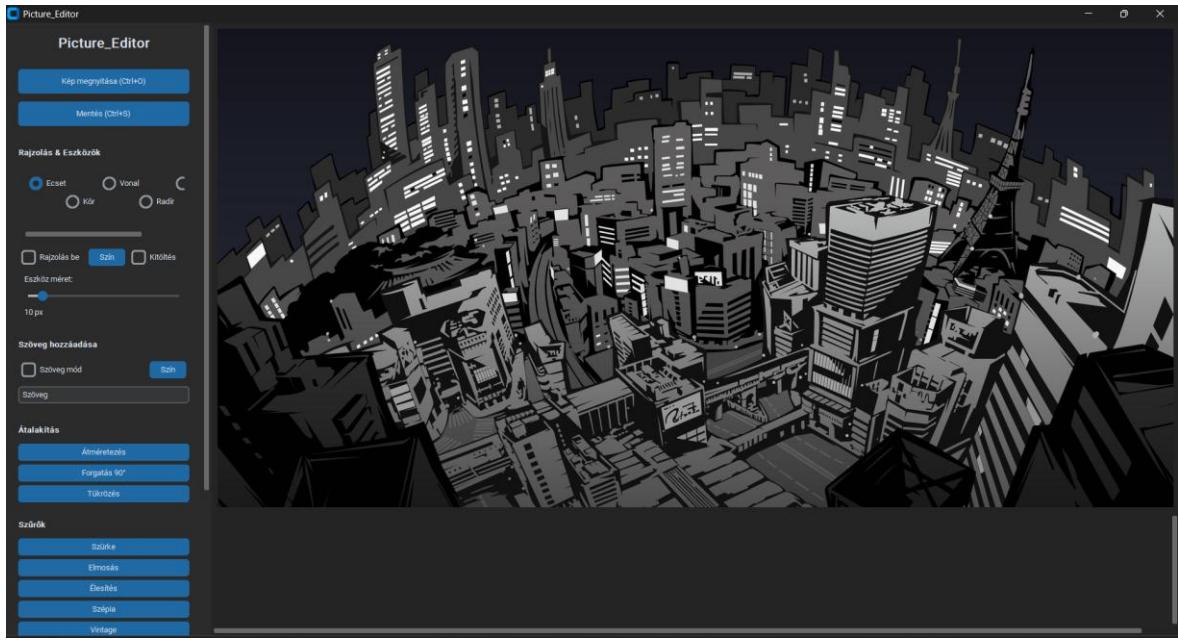
Ezután inditsuk el a programot parancssori parancssal: *python main.py*

Miután elindítottuk a programot és fut, akkor az alábbi felület lesz látható:

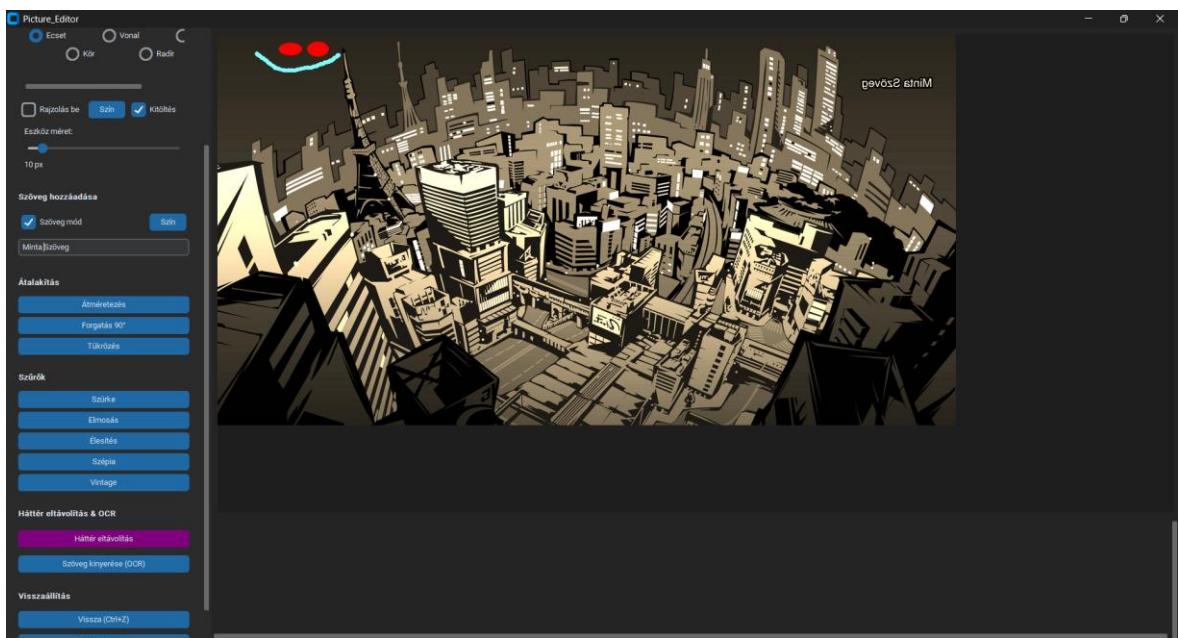


A bal oldali sávban láthatóak a fent leírt funkciók

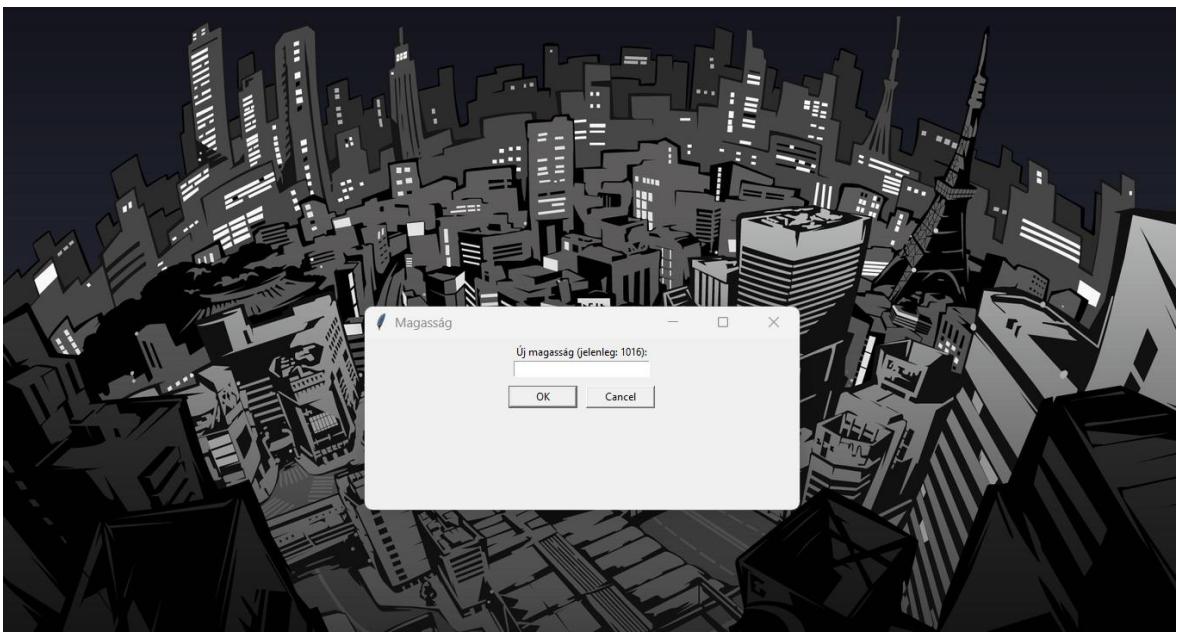
A „Kép megnyitása” gombbal töltsünk be egy képet az alkalmazásba:

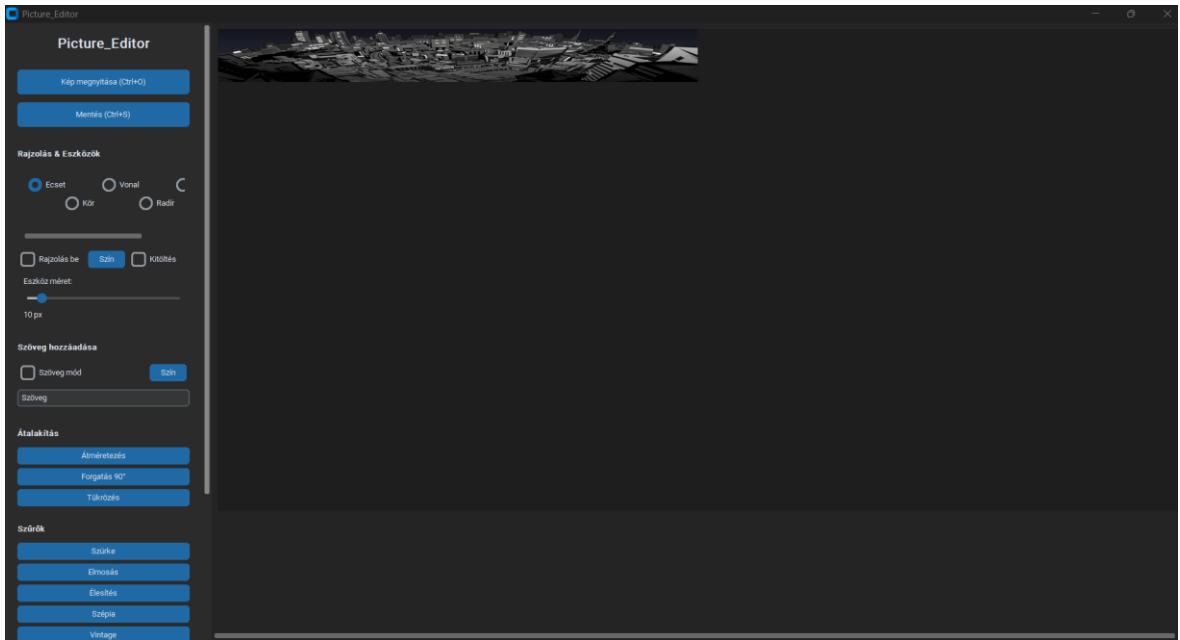


Ezután elkezdhetjük szerkeszteni, ahogy akarjuk a képet és a végén a módosított képet elmenthetjük a „Mentés” gombbal JPEG vagy PNG formátumban.



## Átméretezés

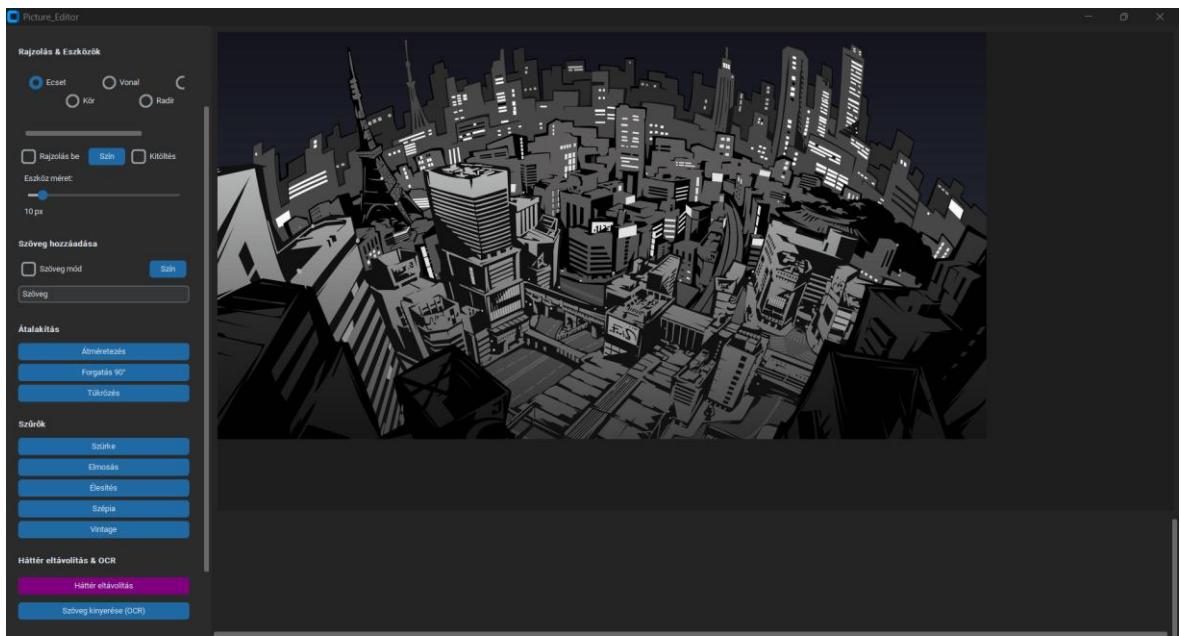




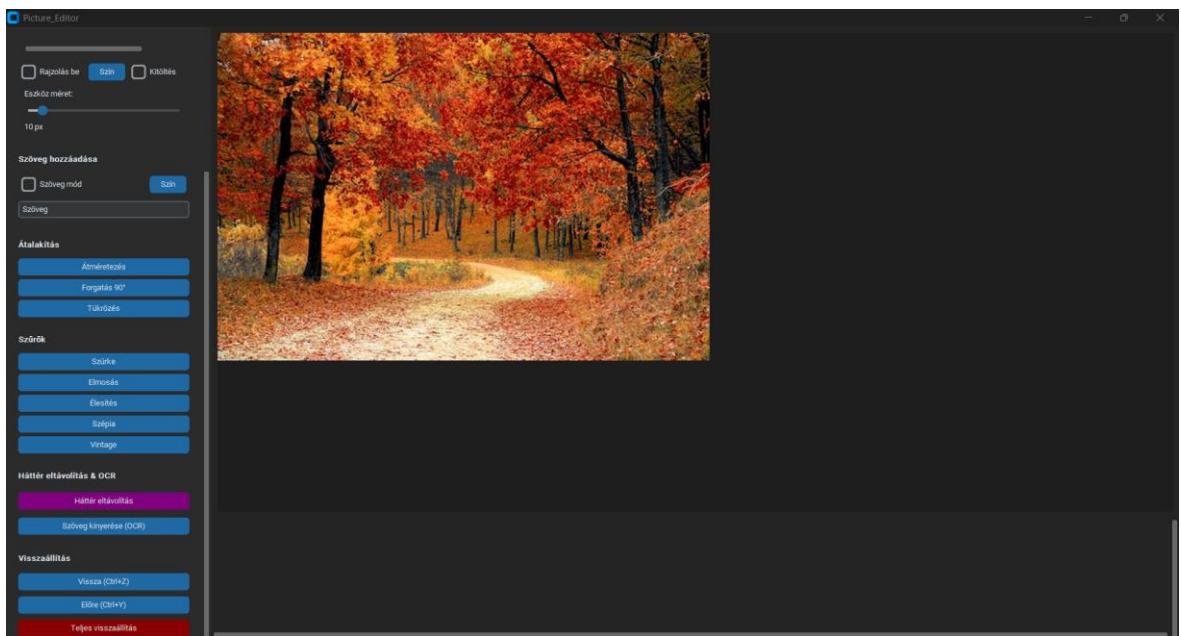
## Forgatás 90 fok

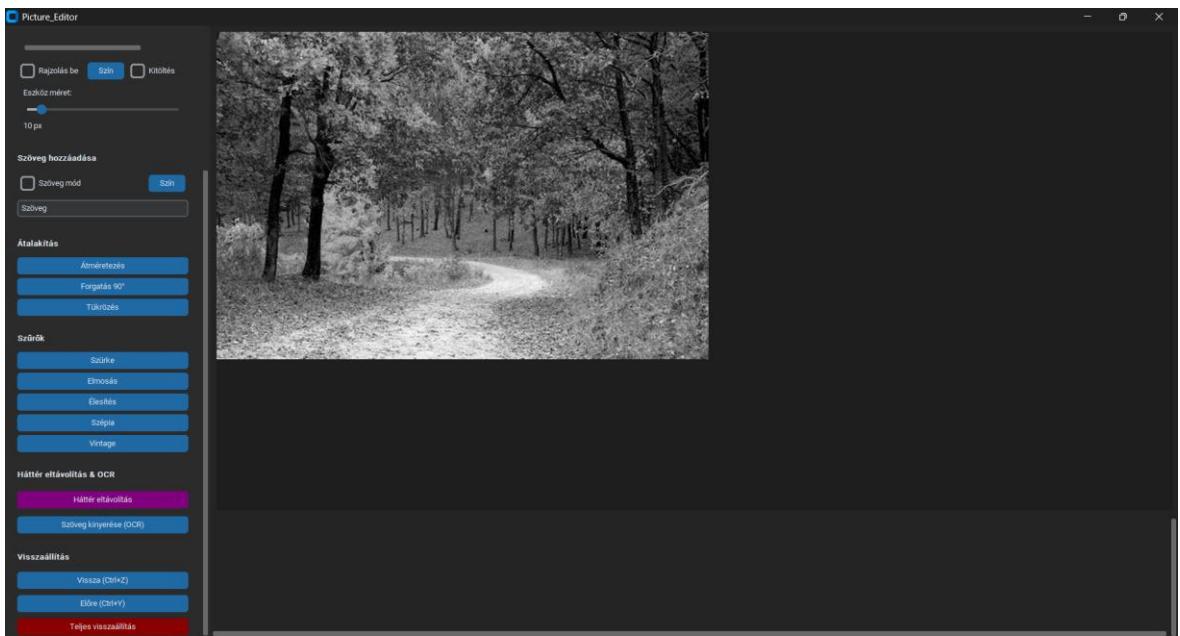


## Tükörzés

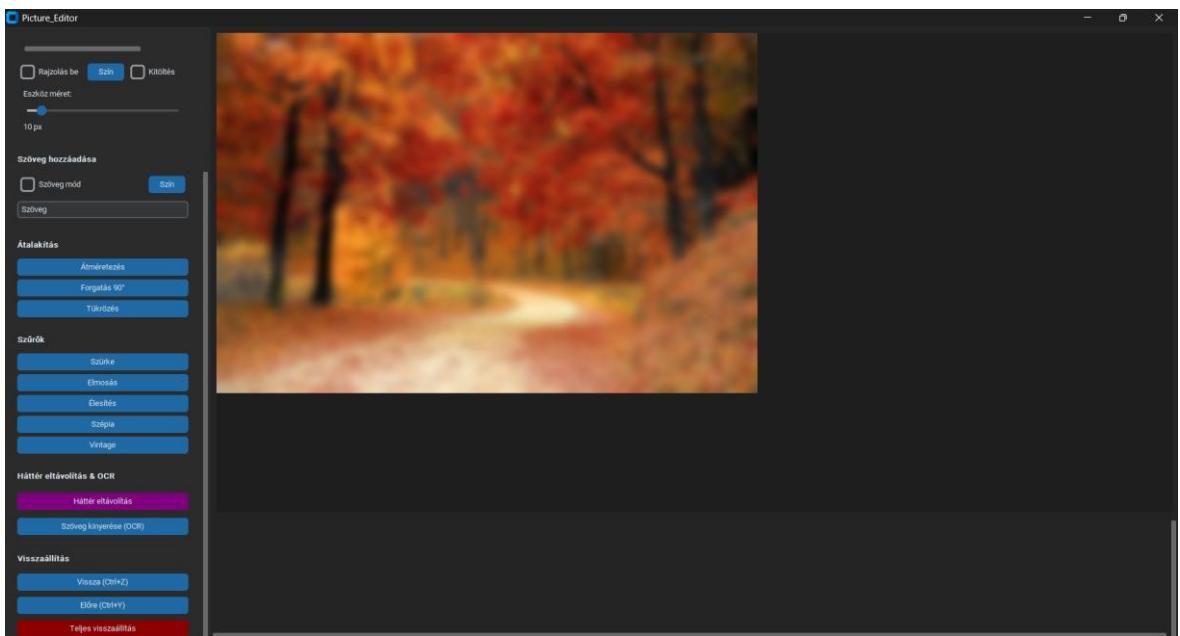


## Szürke filter

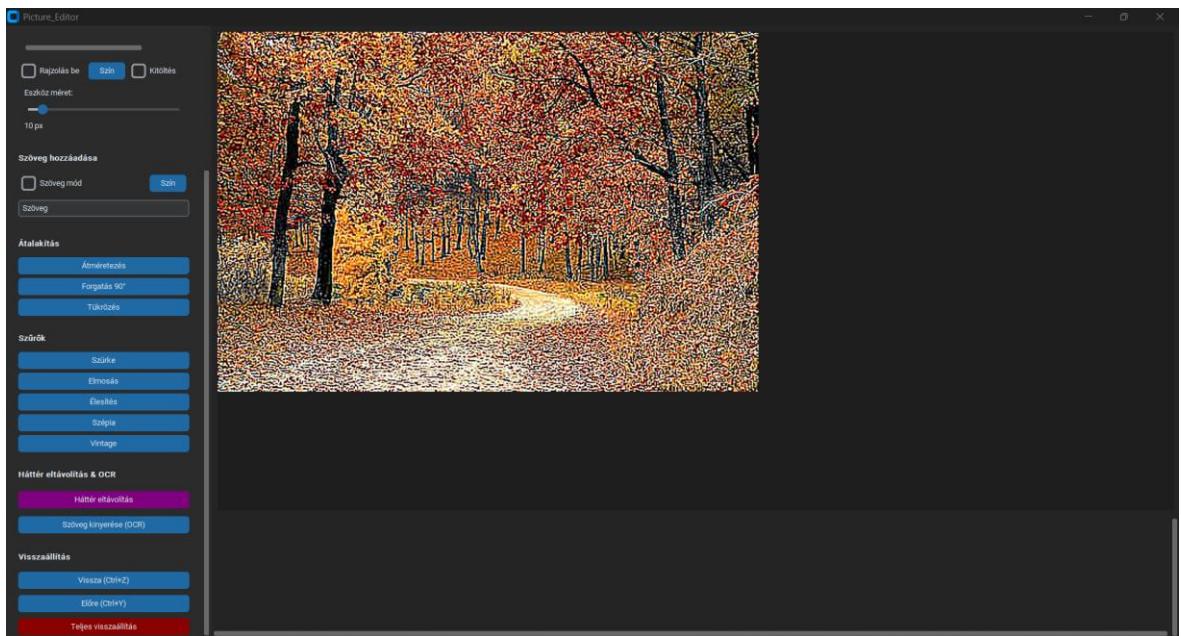




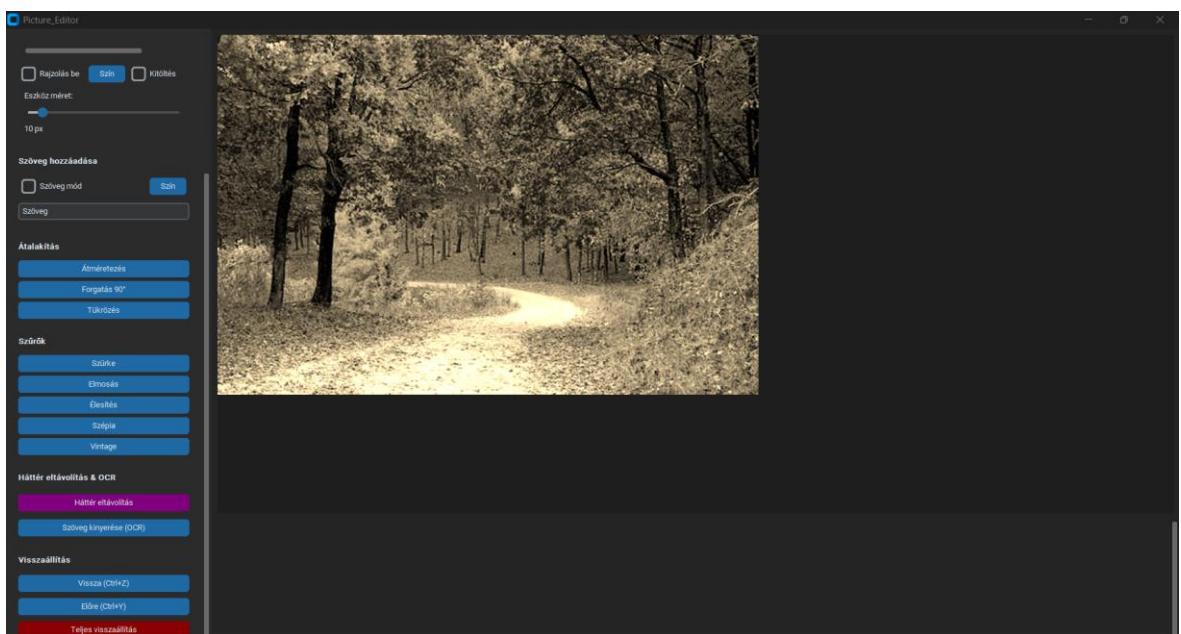
## Elmosás



## Élesítés



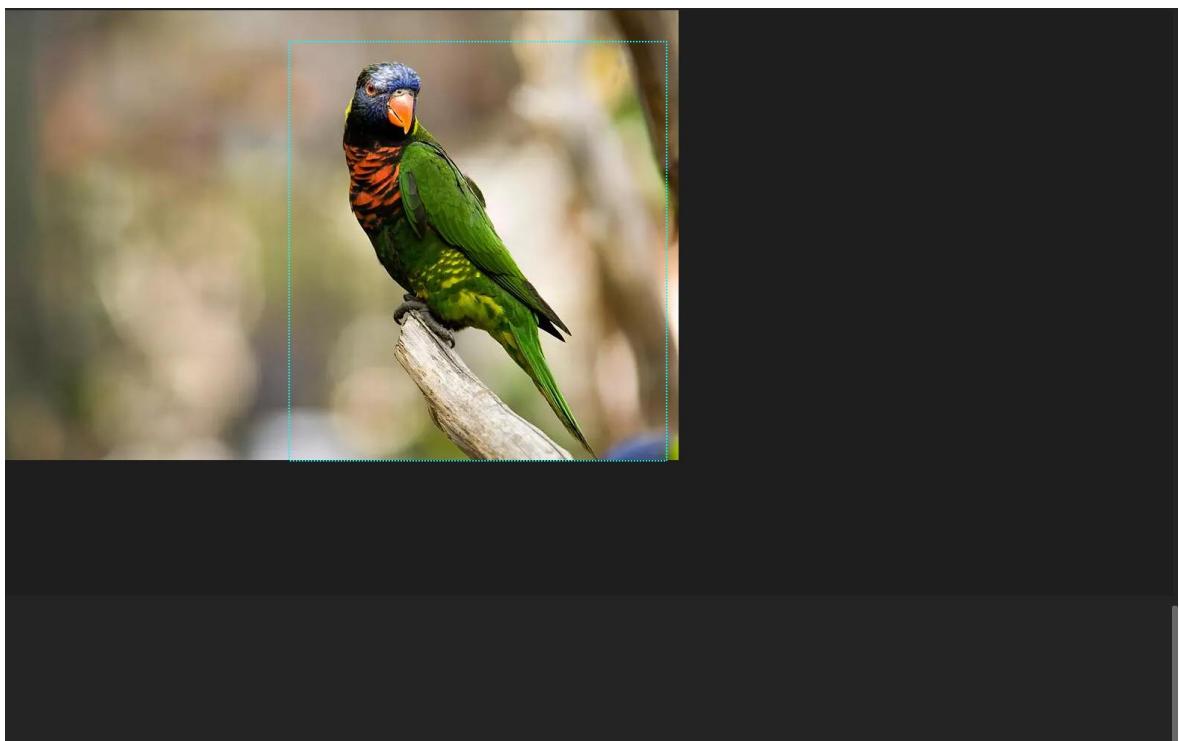
Szépítés

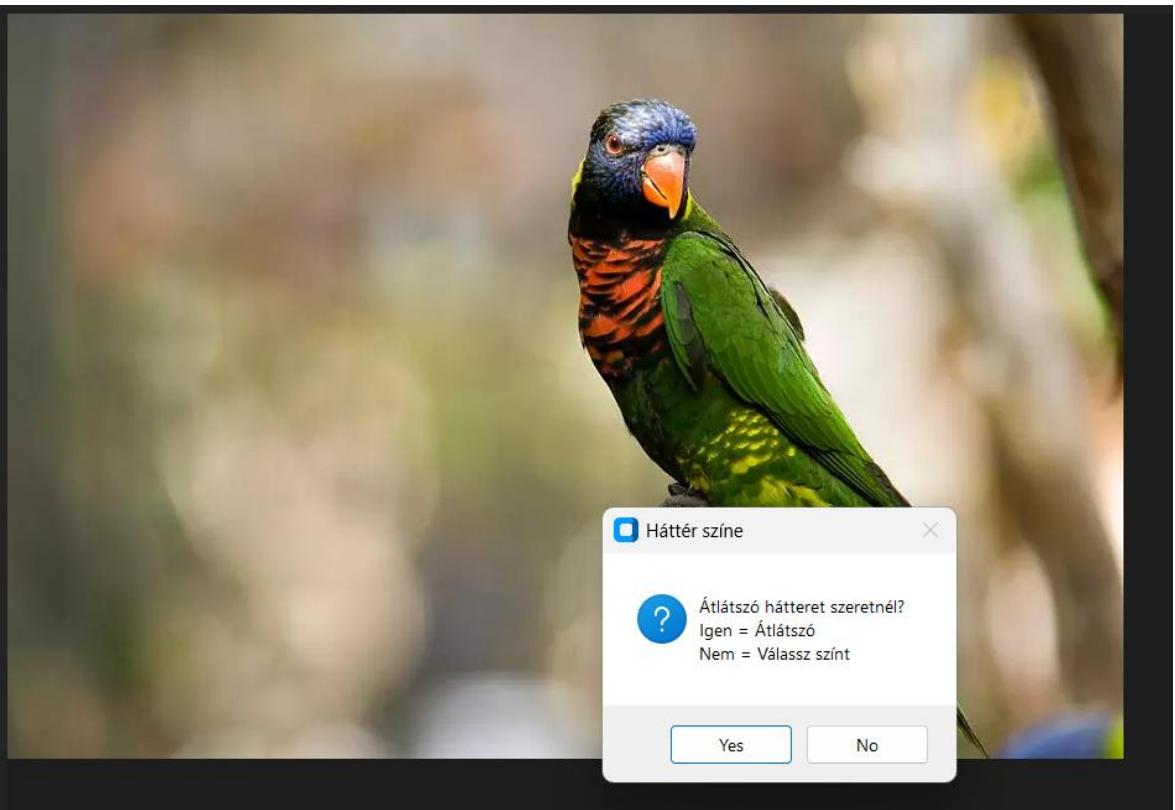


Vintage

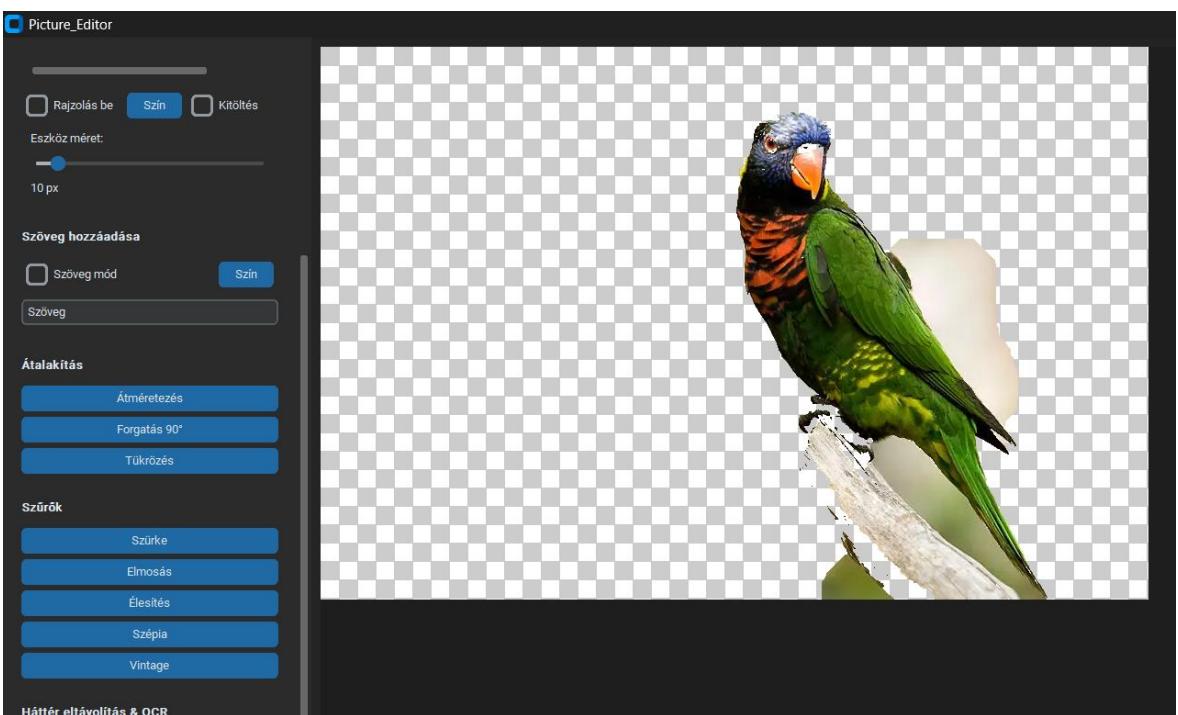


## Háttér eltávolítás

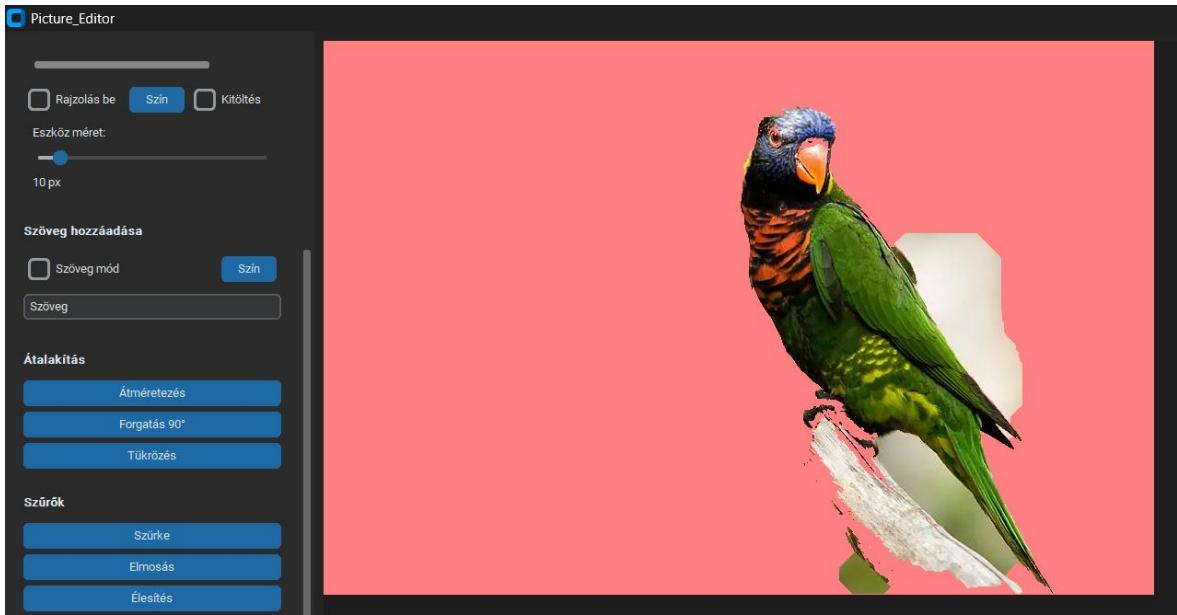




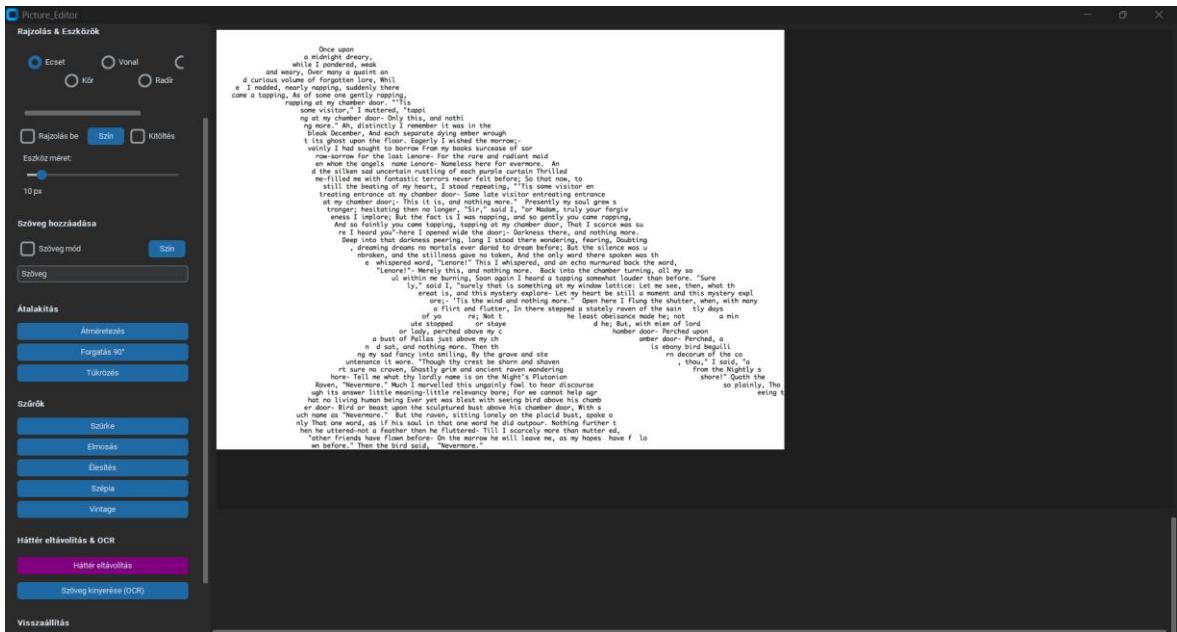
Átlátszó:



Színes:



## Szövegkinyerés



Once upon  
a midnight dreary,  
while I pondered, weak  
and weary, Over many a quaint an  
d curious volume of forgotten lore, Whil  
e I nodded, nearly napping, suddenly there  
came a tapping, As of some one gently rapp  
raping at my chamber door.  
"Tis some visitor," I muttered,  
ring at my chamber door, Only this, and nothi  
ng more. "Ah, distinctly I remember it was in the  
bleak December, And each separate dying ember wroug  
t its ghost upon the floor. Eagerly I wished the morrow;  
vainly I had sought to  
row-sorrow for the lost Lenore- For the rare and radiant maid  
en whom the angels name Lenore- Nameless here for evermore. An  
d the silken sad uncertain rustling of each purple curtain Thrilled  
me-filled me with fantastic terrors never felt before; So that now, to  
still the beating of my heart, I stood repeating, "Tis some visitor en  
treating entrance at my chamber door. Some late visitor entreating entrance  
at my chamber door: This it is, and nothing more. Presently my soul grew s  
tronger; hesitating then no longer, "Sir," said I, "or Madam, truly your forgiv  
eness I implore; But the fact is I was napping, and so gently you came rappin  
g, And so faintly you came tapping, tapping at my chamber door, That I scarce was su  
re I heard you"-here I opened wide the door: Darkness there, and nothing more.  
Deep into that darkness peering, long I stood there wondering, fearing, Doubting  
, dreaming dreams no mortals ever dared to dream before; But the silence was u  
nbroken, and the stillness gave no token, And the only word there spoken was th  
e whispered word, "Lenore!" This I whispered, and an echo murmured back the word,  
"Lenore!" Merely this, and nothing more. Back into the chamber turning, all my so  
ul within me burning, Soon again I heard a tapping somewhat louder than before. "Sure  
ly," said I, "surely that is something at my window lattice: Let me see, then, what th  
reat is, and this mystery explore. Let my heart be still a moment and this mystery expl  
ore;- 'Tis the wind and nothing more." Open here I flung the shutter, when, with many  
a flit and flutter, In there stepped a stately raven of the sain tly days

Kinyert szöveg (OCR)

re I heard you"-here I opened wide the door:- Darkness there, and nothing more.  
Deep into that darkness peering, long I stood there wondering, fearing, Doubting  
, dreaming dreams no mortals ever dared to dream before; But the silence was u  
nbroken, and the stillness gave no token, And the only word there spoken was th  
e whispered word, "Lenore!" This I whispered, and an echo murmured back the word,  
"Lenore!" Merely this, and nothing more. Back into the chamber turning, all my so  
ul within me burning, Soon again I heard a tapping somewhat louder than before. "Sure  
ly," said I, "surely that is something at my window lattice: Let me see, then, what th  
reat is, and this mystery explore. Let my heart be still a moment and this mystery expl  
ore;- 'Tis the wind and nothing more." Open here I flung the shutter, when, with many  
a flit and flutter, In there stepped a stately raven of the sain tly days

of yo re; Not t he least obeisance made he; not amin  
ute stopped or staye dhe; But, with mien of lord  
or lady, perched above my c hamber door- Perched upon  
a bust of Pallas just above my ch amber door- Perched, a  
nd sat, and nothing more. Then th is ebony bird beguili  
ng my sad fancy into smiling. By the grave and ste rn decorum of the co  
unternance it wore. "Though thy crest be shorn and shaven , thou," I said, "a

rt sure no craven, Ghastly grim and ancient raven wandering  
hore- Tell me what thy lordly name is on the Night's Plutonian  
Raven, "Nevermore." Much I marvelli this ungainly fowl to hear discourse  
ugh its answer little meaning-little relevancy bore; For we cannot help agr  
hat no living human being Ever yet was blest with seeing bird above his chamb  
er door- Bird or beast upon the sculptured bust above his chamber door, With s  
uch name as "Nevermore." But the raven, sitting lonely on the placid bust, spoke o  
nly That one word, as if his soul in that one word he did outpour. Nothing further t  
hen he uttered-not a feather then he fluttered- Till I scarcely more than mutter ed,  
"other friends have flown before- On the morrow he will leave me, as my hopes have fo  
wn before." Then the bird said, "Nevermore."

from the Nightly s  
shore! Quoth the  
so plainly, Tho  
eeing t

## **Felhasznált anyagok**

Tesseract-OCR letöltés: <https://github.com/UB-Mannheim/tesseract/wiki>

Tesseract-OCR github repo: <https://github.com/tesseract-ocr/tesseract>

CustomTkinter dokumentáció: <https://customtkinter.tomschimansky.com/documentation/>

Open-CV GrabCut dokumentáció:

[https://docs.opencv.org/3.4/d8/d83/tutorial\\_py\\_grabcut.html](https://docs.opencv.org/3.4/d8/d83/tutorial_py_grabcut.html)

Pytesseract-dokumentáció: <https://pypi.org/project/pytesseract/>

Pillow Dokumentáció: <https://pillow.readthedocs.io/en/stable/handbook/tutorial.html>

Open-CVdokumentáció:

<https://opencv-tutorial.readthedocs.io/en/latest/intro/intro.html#load-and-show-an-image>