

Képannotációs Rendszer - Teljes Dokumentáció

Tartalomjegyzék

1. Általános Információ
2. Rendszer Követelmények
3. Telepítés
4. Program Architektúra
5. Főbb Funkciók
6. Osztály Referencia
7. Adatbázis Séma
8. Felhasználói Útmutató
9. Tipikus Munkafolyamatok
10. Hibaelhárítás
11. Fejlesztési Tipp

Általános Információ

Mi ez a program?

A **Képannotációs Rendszer** egy grafikus felhasználói felülettel (GUI) rendelkező asztali alkalmazás, amelyet Python-ban implementáltak. A program lehetővé teszi a felhasználók számára, hogy képeket annotáljon (jelöljön meg) sokszögek és téglalapok segítségével, valamint hogy különféle címkéket rendelhessenek hozzájuk.

Fő Célkitűzések

- **Képfeldolgozás:** JPG, PNG, BMP, GIF formátumú képek betöltése
- **Rajzoló Eszközök:** Sokszög és téglalap alakú annotációk létrehozása
- **Címke Kezelés:** Egyéni címkék hozzáadása, módosítása és törlése
- **Adatbázis Tárolás:** SQLite adatbázis segítségével az annotációk persistent tárolása
- **Statisztika:** Annotációk és címkék nyomon követése

Programnyelv és Verzió

- **Python verzió:** 3.7+
- **GUI Framework:** Tkinter (beépített, externos telepítés nem szükséges)

Rendszer Követelmények

Szoftver Függőségek

Csomag	Verzió	Leírás
tkinter	beépített	GUI framework

Csomag	Verzió	Leírás
Pillow (PIL)	8.0+	Képfeldolgozás
sqlite3	beépített	Adatbázis kezelés

Támogatott Képfarmátumok

- JPEG (.jpg, .jpeg)
- PNG (.png)
- BMP (.bmp)
- GIF (.gif)

Telepítés

1. Szükséges Csomagok Telepítése

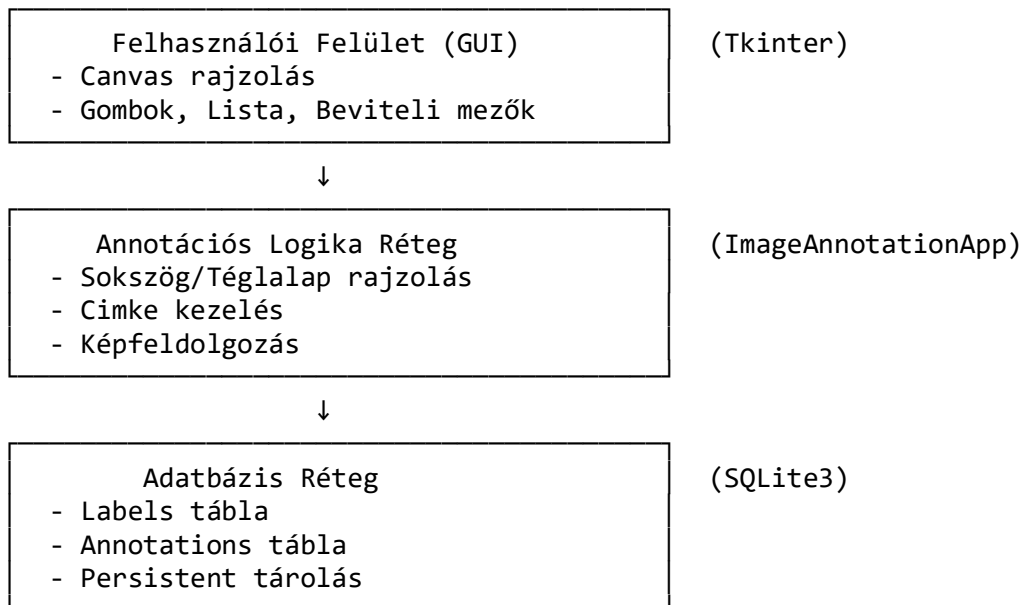
```
pip install Pillow
```

2. Program Futtatása

```
python main3.py
```

Program Architektúra

Szoftver Rétegek



Adatfolyam Diagram

Képbetöltés



Canvas megjelenítés



Rajzolási mód választása (sokszög/téglalap)



Felhasználó rajzol



Címke kiválasztása



Annotáció hozzáadása lista



Mentés adatbázisba (opcionális)

Főbb Funkciók

1. Képbetöltés

Funkció: Különbféle formátumú képek betöltése

```
def load_image(self):  
    """Képfájl dialóguson keresztüli betöltése"""
```

- Támogat: JPG, PNG, BMP, GIF
- Automatikus átméretezés 450×450 pixelre
- Előző annotációk törlése

2. Rajzolási Módo

A) Sokszög Mód

- Pontokra kattintás a sokszög csúcsainak meghatározásához
- Minimum 3 pont szükséges
- Enter billentyű a befejezéshez
- Valós idejű vizualizáció

B) Téglalap Mód

- Kattintás és húzás (drag-and-drop)
- Egy kattintás a kezdőpont, húzás a végpontig
- Automatikus befejezés az egérgomb felengedésekor

3. Címke Kezelés

Címke Hozzáadása

```
def add_label(self):  
    """Új címke létrehozása"""
```

- Egyedi név szükséges
- Automatikus szín hozzárendelés
- Adatbázisba mentés

Cimke Törlés

```
def delete_label(self):  
    """Cimke és kapcsolódó annotációk törlése"""
```

- Kiválasztott cimke törlése
- Kapcsolódó annotációk és adatbázis bejegyzések törlése
- Víz értékek módosítása

Szín Módosítás

- Színválasztó dialógus
- Adatbázis frissítés
- Azonnali megjelenítés

4. Annotációk Mentése

```
def save_annotations_to_db(self):  
    """Annotációk SQLite adatbázisba mentése"""
```

- Kép elérési útjának tárolása
- Cimke ID hivatkozása
- Koordináták (sokszög vagy téglalap)
- Szín információ
- Időbélyeg (CURRENT_TIMESTAMP)

5. Statisztika és Monitoring

Helyi Statisztika

- Jelenlegi annotációk száma
- Aktív cimke megjelenítése

Adatbázis Statisztika

```
def show_db_stats(self):  
    """Összesített statisztikák megjelenítése"""
```

- Cimkénkénti annotációk száma
- Összesen annotációk az adatbázisban

Osztály Referencia

ImageAnnotationApp Osztály

Ez az egyetlen osztály, amely az egész alkalmazás funkcionalitását tartalmazza.

Konstruktor

```
def __init__(self, root):  
    """  
    Alkalmazás inicializálása
```

```
Paraméterek:
    root (tk.Tk): Tkinter gyökér ablak
    """
```

Inicializált attribútumok:

Attribútum	Típus	Leírás
self.image	PIL.Image	Betöltött kép objektum
self.annotations	list	Annotációk listája (dict)
self.labels	list	Elérhető címkék
self.draw_mode	str	“polygon” vagy “box”
self.points	list	Jelenlegi rajzolás pontjai
self.current_label	str	Jelenleg kiválasztott címke
self.current_color	str	Jelenleg használt szín (hex)
self.conn	sqlite3.Connection	Adatbázis kapcsolat
self.cursor	sqlite3.Cursor	Adatbázis kurzor

Adatbázis Metódusok

```
init_database()
def init_database(self):
    """SQLite adatbázis inicializálása"""
```

Feladat: - Adatbázisfájl létrehozása (“annotations.db”) - Táblák felépítése (ha nem léteznek) - Alapértelmezett címkék hozzáadása

Létrehozott Táblák:

1. labels tábla

```
CREATE TABLE labels (
    id INTEGER PRIMARY KEY,
    name TEXT UNIQUE,
    color TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
```

2. annotations tábla

```
CREATE TABLE annotations (
    id INTEGER PRIMARY KEY,
    image_path TEXT,
    label_id INTEGER,
    annotation_type TEXT,
    coordinates TEXT,
    color TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
        FOREIGN KEY (label_id) REFERENCES labels(id)
    )
```

```
load_labels_from_db()
def load_labels_from_db(self):
    """Cimkék betöltése az adatbázisból"""
```

- Cimkék lekérése az adatbázisból
- Alapértelmezett cimkék hozzáadása, ha üres

Alapértelmezett Cimkék: - "kutya" (#FF6B6B - piros) - "macska" (#4ECDC4 - türkiz) - "madár" (#FFE66D - sárga) - "autó" (#95E1D3 - zöld)

GUI Setup Metódus

```
setup_ui()
def setup_ui(self):
    """Felhasználói felület elrendezésének felépítése"""
```

Bal Oldal (Canvas Area): - Kép betöltés gomb - Rajzolási mód gombok (Sokszög, Téglalap) - Undo gomb - Canvas widget (rajzoláshoz)

Jobb Oldal (Control Panel): - Cimke hozzáadása mező - Cimkék listája - Szín választó gomb - Cimke törlés gomb - Statisztika panel - Adatbázis információ panel

Rajzolási Metódusok

```
on_canvas_click(event)
def on_canvas_click(self, event):
    """Canvas kattintás esemény kezelése"""
```

Sokszög módban: - Pont hozzáadása a self.points listához - Canvas újrarajzolása

Téglalap módban: - Kezdőpont tárolása

Ellenőrzések: - Kép betöltve? - Cimke kiválasztva?

```
on_canvas_drag(event)
def on_canvas_drag(self, event):
    """Canvas húzás (drag) esemény kezelése"""
```

- Csak téglalap módban aktív
- Valós idejű téglalap megjelenítés
- Canvas újrarajzolása minden mozgás alkalmával

```
on_canvas_release(event)
def on_canvas_release(self, event):
    """Egér gomb felengedés esemény kezelése"""
```

- Téglalap hatásoknak befejezése
- Annotáció hozzáadása a listához
- Szín és címke információ tárolása

```
redraw_canvas()
def redraw_canvas(self):
    """Canvas teljes újrarajzolása"""
```

Lépések: 1. Kép megjelenítése 2. Meglévő annotációk rajzolása (sokszög/téglalap) 3. Címkek szövegének megjelenítése 4. Aktuális rajzoló pontok kijelölése

Vizualizáció: - Sokszög: hármass hex alfája (33) a kitöltéshez (pl. #FF6B6B33) - Téglalap: hármass hex alfája a kitöltéshez - Pontok: fekete szegéllyel körök pont

Címke Kezelési Metódusok

```
add_label()
def add_label(self):
    """Új címke hozzáadása"""
```

Validáció: - Üres mező ellenőrzése - Duplikáció ellenőrzése

Folyamat: 1. Név beolvasása input mezőből 2. Véletlenszerű szín kiválasztása 3. Adatbázisba mentés 4. Adatbázis frissítés 5. UI frissítés

```
delete_label()
def delete_label(self):
    """Kiválasztott címke törlése"""
```

Hatások: - labels táblából törlés - annotations táblából kapcsolódó bejegyzések törlése - UI lista frissítése - Statisztika frissítése

```
on_label_select(event)
def on_label_select(self, event):
    """Címke kiválasztása a listából"""
```

- Aktív címke beállítása
- Szín frissítése
- Szín gomb megjelenítésének frissítése

```
choose_color()
def choose_color(self):
    """Szín választása dialóguson keresztül"""
```

- Standard Tkinter szín választó dialógus
- Adatbázis frissítés
- UI frissítés

Megjelenítési Metódusok

```
load_image()
def load_image(self):
    """Képfájl betöltése"""
```

Fájl dialógus: - Képfájlok szűrése - Támogatott formátumok: JPG, PNG, BMP, GIF

Feldolgozás: 1. Kép megnyitása PIL-el 2. Átméretezés 450×450 pixelre (arányok megőrzésével) 3. Annotációk törlése 4. Canvas frissítés 5. Sikeres üzenet megjelenítése

```
clear_canvas()
def clear_canvas(self):
    """Canvas és adatok törlése"""
```

- Canvas tisztítása
 - Annotációk lista ürítése
 - Pont lista ürítése
 - Kép törlése
 - Statisztika frissítése
-

Annotációs Befejezésmetódusok

```
finish_polygon()
def finish_polygon(self):
    """Sokszög annotáció befejezése"""
```

Feltételek: - Minimum 3 pont szükséges - Cimke kiválasztva

Adat szerkezet:

```
{
    "type": "polygon",
    "label": str,
    "color": str,
    "points": [(x1, y1), (x2, y2), ...]
}
```

```
undo()
def undo(self):
    """Utolsó rajzolás vagy pont visszavonása"""
```

- Ha pontok vannak: legutolsó pont törlése
 - Ellenkező esetben: legutolsó annotáció törlése
 - Canvas frissítése
-

Statisztika Metódusok

```
update_stats()
def update_stats(self):
    """Helyi statisztika frissítése"""
```

Megjelenít: - Jelenlegi annotációk száma - Aktív címke neve

```
update_label_count()
def update_label_count(self):
    """Cimkéhez kapcsolódó annotációk száma"""
```

- Számolás az aktuális session-ben
- Lista frissítése címke mellett megjelenő számmal

```
update_labels_list()
def update_labels_list(self):
    """Cimkék listájának GUI frissítése"""
```

Format: • címkénév (szám)

```
update_db_stats()
def update_db_stats(self):
    """Adatbázis statisztika frissítése"""
```

- Összes annotáció lekérdezése az adatbázisból
- UI panel frissítése

Adatbázis Mentés és Lekérdezés

```
save_annotations_to_db()
def save_annotations_to_db(self):
    """Annotációk mentése az adatbázisba"""
```

Előfeltételek: - Kép betöltve - Annotációk léteznek

Mentett Információ: - Képfájl elérési útja - Címke ID (foreign key) - Annotáció típusa (polygon/box) - Koordináták (string formátum) - Szín (hex) - Timestamp (automatikus)

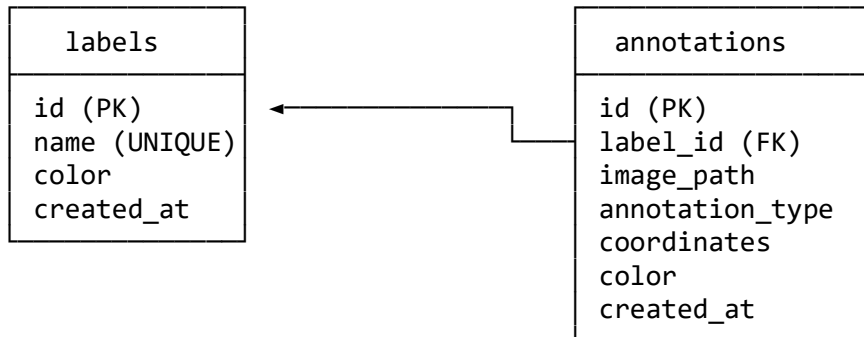
Koordináta Format: - **Sokszög:** "[(x1, y1), (x2, y2), ...]" (string) - **Téglalap:** "x0,y0,x1,y1" (string)

```
show_db_stats()
def show_db_stats(self):
    """Adatbázis statisztika megjelenítése"""
```

- Címkénkénti annotáció szám
 - Összes annotáció
 - MessageBox dialógusban megjelenítés
-

Adatbázis Séma

Teljes ER Diagram



Labels Tábla

Oszlop	Típus	Leírás
id	INTEGER PRIMARY KEY	Cimke egyedi azonosítója
name	TEXT UNIQUE	Cimke neve (egyedi)
color	TEXT	Szín HEX formátumban (#RRGGBB)
created_at	TIMESTAMP	Létrehozás időpontja

Példa adat:

```
INSERT INTO labels (name, color) VALUES ('kutya', '#FF6B6B');
```

Annotations Tábla

Oszlop	Típus	Leírás
id	INTEGER PRIMARY KEY	Annotáció egyedi ID
image_path	TEXT	Kép fájl elérési útja
label_id	INTEGER FK	Hivatkozás labels.id-re
annotation_type	TEXT	'polygon' vagy 'box'
coordinates	TEXT	Koordináták JSON/string formában
color	TEXT	Szín HEX formátumban
created_at	TIMESTAMP	Létrehozás időpontja

Példa adat:

```
INSERT INTO annotations  
(image_path, label_id, annotation_type, coordinates, color)  
VALUES ('/path/to/image.jpg', 1, 'polygon', '[(10,20), (50,60), (100,30)]',  
'#FF6B6B');
```

Relációk

- **1:N reláció:** labels → annotations
- **Foreign Key:** annotations.label_id → labels.id

- **Cascade Delete:** Nincs konfigurálva (manuális törlés szükséges)
-

Felhasználói Útmutató

Kezdeti Lépések

1. Program Indítása

python main3.py

2. Alapértelmezett Cimkék

A program automatikusan 4 alapértelmezett címkét hoz létre: - **kutya** (piros) - **macska** (türkiz) - **madár** (sárga) - **autó** (zöld)

Tipikus Felhasználói Feladatok

A. Kép Annotációja Sokszöggel

1. Kattintson a **“Kép”** gombra
2. Válasszon egy képfájlt a dialógusból
3. Kattintson a **“Sokszög”** gombra a rajzolási módhoz
4. Válasszon egy címkét a jobboldali listából
5. Kattintson a képre több ponton egy sokszög létrehozásához
6. Nyomja meg az **Enter** gombot a befejezéshez
7. Az annotáció megjelenik a képen a kiválasztott szín és címke betűvel

B. Kép Annotációja Téglalappal

1. Kattintson a **“Kép”** gombra
2. Válasszon egy képfájlt
3. Kattintson a **“Téglalap”** gombra
4. Válasszon egy címkét
5. **Kattintson és húzza** az egeret a téglalap rajzolásához
6. Engedje fel az egérgombot - az annotáció automatikusan hozzáadódik

C. Új Cimke Hozzáadása

1. Írja be a címke nevét a jobb oldalon a **“Cimkék Kezelése”** alatt
2. Kattintson a **“Hozzá”** gombra
3. A címke automatikus színt kap és hozzáadódik az adatbázishoz
4. Jelenik meg a listában

D. Cimke Szín Módosítása

1. Válasszon egy címkét a listából
2. Kattintson a nagy szín gombra a jobb oldalon
3. Válasszon egy új színt

4. A szín azonnal frissül az adatbázisban

E. Annotációk Mentése

1. Végezze el az annotációkat
2. Kattintson a **“DB Statisztika”** gomb mellett szereplő **“Mentés”** gombra
 - *Megjegyzés: A program bezárásnál is rákérdez a mentésre*

F. Statisztika Megtekintése

1. Kattintson a **“DB Statisztika”** gombra
2. Megjelenik egy dialog címkénkénti annotációs számmal

Billentyűzet Gyorsbillentyűk

Gomb	Funkció
Enter	Sokszög befejezése
Escape	Canvas törlése

Szín Paletta

Alapértelmezetten használt színek új címkékhez:

#FF6B6B - Piros
#4ECDC4 - Türkiz
#FFE66D - Sárga
#95E1D3 - Zöld
#F7B731 - Arany
#A29BFE - Lila
#FD79A8 - Rózsaszín
#6C5CE7 - Sötét lila
#00B894 - Moszat zöld

Tipikus Munkafolyamatok

Workflow 1: Képadatgyűjtés Projekthez

1. Projekt anyagait tartalmazó mappát megnyitni
2. Az első képet betölteni
3. Szükséges címkéket hozzáadni (ha nincsenek)
4. Képet annotálni sokszögekkel/téglalapokkal
5. Ciklus: következő kép → lépés 2-4
6. Minden kép után menteni az adatbázisba
7. Projekt végén adatbázis biztonsági másolat készítése

Workflow 2: Gépi Tanulás Dataset Előkészítése

1. Címkék definiálása (az eredeti címkék felülírása)
2. Képek tömeges betöltése és annotációja

3. Statisztika ellenőrzése (kiegyensúlyozottság?)
4. Adatbázis exportálása CSV-be (szükség esetén)
5. Dataset felhasználása a modellhez

Workflow 3: Hibakeresés és Korrekció

1. Annotáció megtekintése
2. Ha hiba: Undo gomb (vagy teljes törlés: Törlés gomb)
3. Újraannotáció
4. Mentés az adatbázisba

Hibaelhárítás

1. "ModuleNotFoundError: No module named 'PIL'"

Megoldás:

```
pip install Pillow
```

2. "ModuleNotFoundError: No module named 'tkinter'"

Megoldás (Linux):

```
sudo apt-get install python3-tk
```

Megoldás (macOS): - Telepítse újra a Pythont Homebrew-vel:

```
brew install python-tk
```

3. Adatbázis Fájl Nem Létezik

Oka: Hibás engedélyek vagy hely **Megoldás:** - Futtassa az alkalmazást egy olyan mappából, ahova van írási jogosultsága vagy manuálisan hozzon létre egy mappát és módosítsa a `self.db_path`-ot

4. Canvas Nem Rajzol

Oka: Kép nem betöltve vagy címke nem kiválasztva **Megoldás:** - Ellenőrizze, hogy van-e betöltött kép (jelenik meg?) - Jelöljön ki egy címkét a listából

5. Szín Nem Frissül

Oka: Adatbázis commitálás sikertelen **Megoldás:** - Ellenőrizze a disk szabad helyét - Próbálja meg újra indítani az alkalmazást

6. Ékezetes karakterek Problémái

Oka: Kódolási probléma **Megoldás:** Győződjön meg, hogy a Python fájl UTF-8 kódolásban van:

Fejlesztési Lehetőségek

A. Kódbővítések

- Export Lehetőség (CSV/JSON)
- Importálási Funkció
- Batch Processing
- Billentyűzet Shortcut-ok

B. UI Javítások

- Sötét Téma
- Nagyobb Canvas
- Zoomolás

C. Adatbázis Optimalizáció

- Index Létrehozása
- Adatbázis Biztonsági Mentés

D. Teljesítmény Optimalizálás

- Batch Insert
- Lusta Betöltés (Lazy Loading)

E. Biztonsági Fejlesztések

- Adat Validálás
-