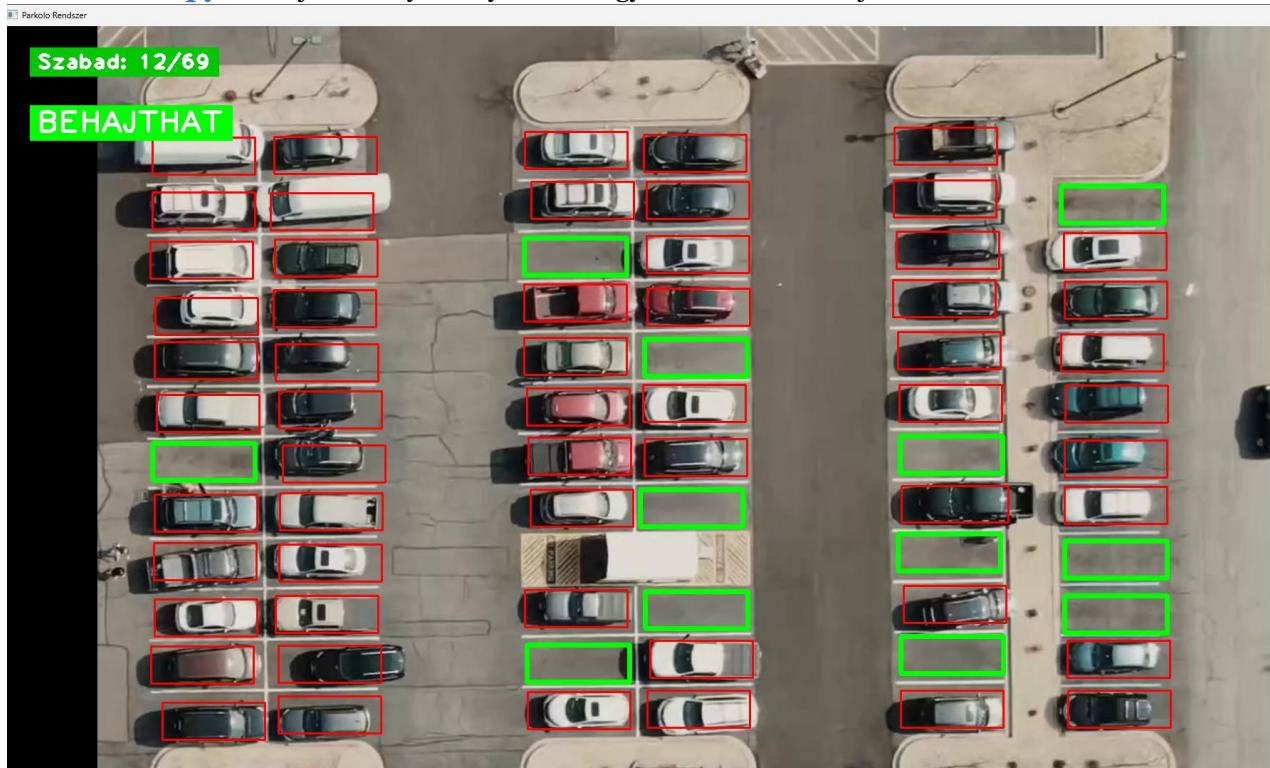


Parkolófoglaltság-figyelő Program

1. Rövid áttekintés

Ez a program egy parkoló biztonsági kamerajából megállapítja, mely parkolóhelyek szabadok vagy foglaltak. Két fájlból áll:

- **parkolo_kijelolo.py** – A parkolóhelyek manuális kijelölése a videó keretén
- **main.py** – A kijelölt helyek folyamatos figyelése és státusz kijelzése



2. A program használata

2.1. Telepítés

Szükséges fájlok a mappában:

- `carPark.mp4` – parkolóról készült videó
- `parkolo_kijelolo.py` és `main.py` – Python scriptek

Python csomagok telepítése:

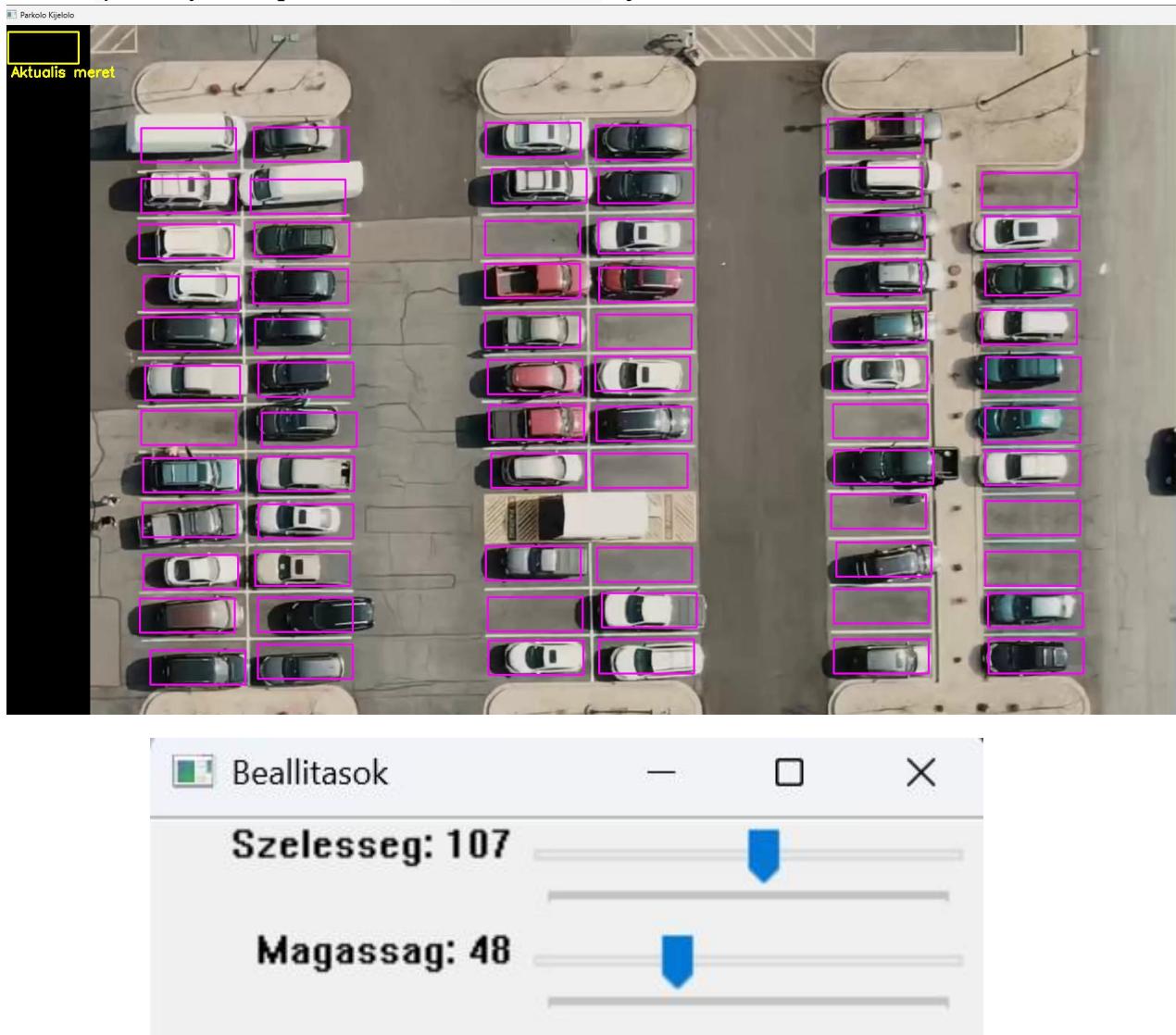
`pip install opencv-python cvzone numpy`

2.2. 1. lépés: Parkolóhelyek kijelölése

Indítsd el: python
parkolo_kijelolo.py

Megjelenik a kijelölő ablak:

- **Beallitasok ablak:** Két csúszka a parkolóhely téglalap méretéhez ([translate:Szelesseg] és [translate:Magassag])
- **Videó ablak:** Az eredeti parkolóról szóló videó Hogyan jelölök ki a helyeket:
 1. Állítsd be a csúszka segítségével a parkolóhely méretét (sárga téglalap mutatja az aktuális méretet)
 2. Bal egérkomb: Kattints a parkolóhelyek bal felső sarkára → lila keret jelenik meg
 3. Jobb egérkomb: Törléshez kattints egy meglévő keretre
 4. q billentyű: Kilépés és mentés (CarParkPos fájlba)



3. Kódmagyarázat – parkolo_kijelolo.py

Alapfeladat

Ez a program lehetővé teszi, hogy a felhasználó manuálisan kijelöljön parkolóhelyeket a videó keretén. Az adatokat a CarParkPos fájlban tárja el.

Fő komponensek

Adatok betöltése/mentése: try: with open('CarParkPos', 'rb') as f: posList = pickle.load(f) except: posList = [] # Üres lista, ha nincs mentett adat

- Betölti a korábban kijelölt helyeket vagy üres listával indul **Egérkezelő**:
- **Bal kíkk:** Az aktuális mérettel (w, h) hozzáadja az egérpozíciót (x, y) a listához
- **Jobb kíkk:** Ha az egér egy meglévő téglalap fölött van, azt törli
- **Kilepés (q):** Elmenti az összes kijelölést a CarParkPos fájlba **Videó megjelenítés:**

- Folyamatosan olvassa a videó képkockáit
- Kirajzolja az összes mentett helyet lila kerettel
- Mutatja az aktuális méret sárga téglalapjának segítségével

4. Kódmagyarázat – main.py

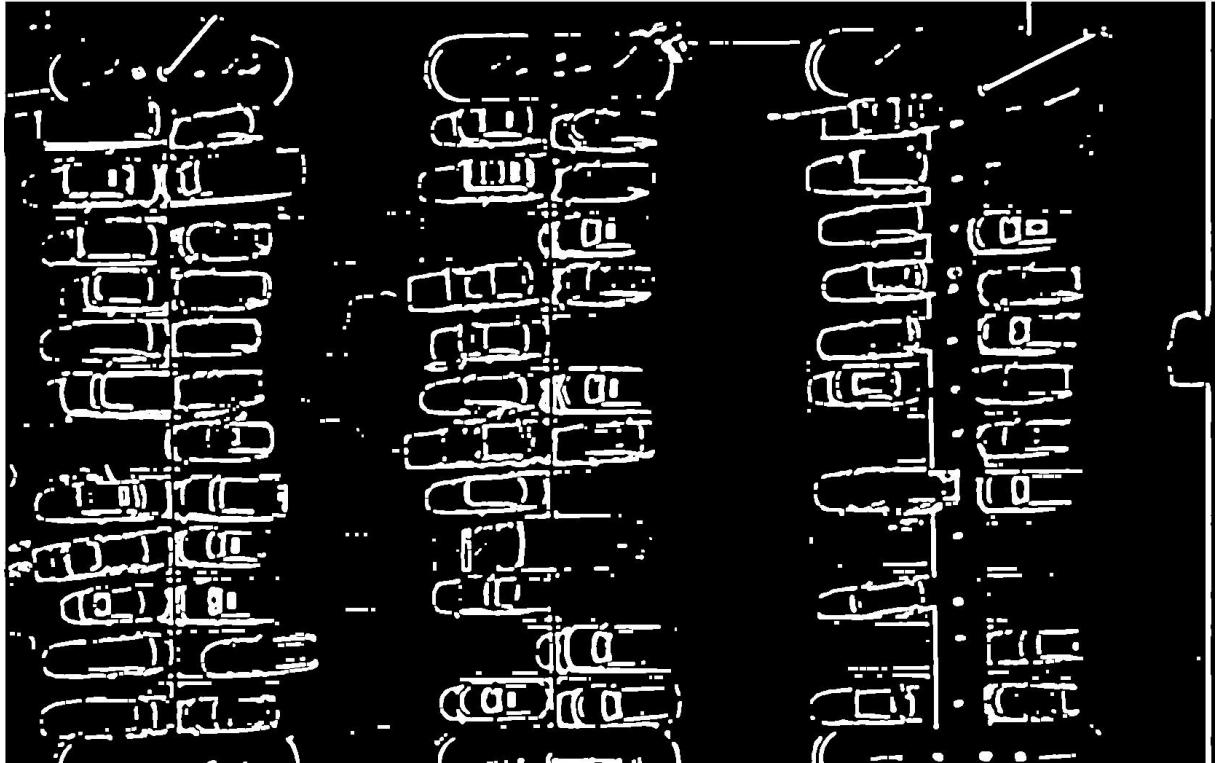
Alapfeladat

Ez a program elemzi a videót és megállapítja, mely kijelölt parkolóhelyek szabadok (kevés világos pixel) vagy foglaltak (sok világos pixel).

Képfeldolgozási lánc

1. **Szürkeárnyalatos konverzió** – RGB → szürke
2. **Gaussian elmosás** – Zajok eltávolítása
3. **Adaptív küszöbölés** – Fekete-fehér kép; sötét objektumok fehérek, háttér fekete
4. **Median szűrő** – Szórt zajpixelek eltávolítása
5. **Dilatáció** – Fehér vonalak vastagítása a stabil detektáláshoz

Amit a program valójában lát:



Analízis függvény

```
def checkParkingSpace(imgPro, imgOriginal):
    for pos in posList: x, y, w, h = pos
        imgCrop = imgPro[y:y+h, x:x+w]
        count = cv2.countNonZero(imgCrop) # Fehér pixelek száma
        if count < 800: # Küszöbérték
            color = (0, 255, 0) # Zöld - szabad
            spaceCounter += 1
        else:
            color = (0, 0, 255) # Piros - foglalt

        cv2.rectangle(imgOriginal, (x, y), (x+w, y+h), color, thickness)
```

- Lekivágja a feldolgozott képből minden parkolóhelyet
- Számolja a fehér pixeleket
- Ha kevés (< 800): szabad → zöld keret
- Ha sok (≥ 800): foglalt → piros keret
- Kiírja az eredményt a képre

Fő ciklus

```
while True:
    # Videó újraindítása ha véget ért
```

```
if cap.get(cv2.CAP_PROP_POS_FRAMES) == cap.get(cv2.CAP_PROP_FRAME_COUNT):
    cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
    success, img = cap.read() #
    Képfeldolgozás és analízis...
    checkParkingSpace(imgDilate, img)
    cv2.imshow("Parkolo Rendszer",
               img)
    if cv2.waitKey(40) & 0xFF ==
        ord('q'):
        break
```

- Folyamatos videó lejátszás (loopol a végén)
 - minden képkockán lefuttatja az analizist
 - q lenyomásával kilép
-

5. Finomhangolás

- **800-as küszöbérték:** Módosítható más videók vagy megvilágítás esetén
- **Csúszka értékek:** Nagyobb/kisebb parkolóhely detektálásához módosítható