

# Forgalomfigyelő és számláló rendszer

Készítette: Falusi Ferenc, Halász Ákos, Ollé Zoltán Csaba

Ez a Python alkalmazás mesterséges intelligencia segítségével elemzi a megadott videofelvételt, hogy automatikusan felismerje és nyomon kövesse a rajta áthaladó járműveket, például autókat, buszokat és teherautókat. A YOLOv8 modell folyamatosan figyeli az objektumok pozícióját és egyedi azonosítóval látja el őket, amiből a rendszer a mozgásuk alapján kiszámítja, hogy éppen felfelé vagy lefelé haladnak-e az úton. A képernyő közepén egy virtuális vonalat húz, amelyen áthaladva a program irány szerint szétválogatva megszámolja a forgalmat, így pontos statisztikát készít az áthaladó járművekről. A videó lejátszása közben valós idejű vizuális visszajelzést is kapunk: színes keretekkel és feliratokkal jelöli a járműveket, miközben a bal felső sarokban egy összesítő táblázaton folyamatosan frissíti a számlálót. A kód emellett automatikusan a monitor felbontásához igazítja a lejátszó ablak méretét, biztosítva, hogy a videó és az adatok mindig kényelmesen olvashatóak legyenek.

A programot Visual Studio Code-ban írtuk.

Felhasznált modellek és installjaik:

- Python 3.10
- OpenCV – videófeldolgozás és kirajzolás
- Ultralytics YOLOv8 – detektálás és tracking
- Tkinter – képernyőfelbontás lekérése
- NumPy – matematikai számítások

```
py -m pip install -U lapx
```

```
py -m pip install --upgrade pip
```

```
python -m ensurepip --upgrade
```

```
py -m pip install opencv-python ultralytics
```

```
pip install ultralytics opencv-python numpy
```

Program működése:

A program a detektáláshoz az **Ultralytics** könyvtárat és a **YOLOv8s** előre betanított mesterséges intelligencia modellt használja. A felismerés a `model.track()` függvény meghívásával történik minden egyes videóképkockán. A rendszer a `classes` paraméter segítségével szűri az eredményeket, így kizárólag a meghatározott járműtípusokat (autó, motor, busz, teherautó) figyeli. A `persist=True` beállítás biztosítja, hogy a modell emlékezzen az objektumokra, így folyamatosan követni tudja ugyanazt a járművet a képkockák között. A detektálás eredményeként a program megkapja a járművek pontos helyzetét (koordinátáit), típusát és egyedi azonosítóját.

Kód:

```
# Detektálás
results = model.track(frame, persist=True, verbose=False, classes=VEHICLE_CLASSES)[0]

if results.bboxes.id is not None:
    boxes = results.bboxes.xyxy.cpu().numpy()
    track_ids = results.bboxes.id.int().cpu().numpy()
    cls = results.bboxes.cls.int().cpu().numpy()

    for box, track_id, cls in zip(boxes, track_ids, cls):
        x1, y1, x2, y2 = map(int, box)
        cx, cy = int((x1 + x2) / 2), int((y1 + y2) / 2)

        # --- MOZGÁS LOGIKA ---
        track = track_history[track_id]
        track.append((cx, cy))
        if len(track) > 30: track.pop(0)

        # Irány meghatározása
        direction_text = ""
        color_dir = (0, 255, 0) # Alap zöld

        if len(track) >= 2:
            diff = track[-1][1] - track[-2][1]
            if diff > 2: # Y nő -> Lefele
                direction_text = "LE"
                color_dir = (0, 0, 255) # Piros
            elif diff < -2: # Y csökken -> Felfele
                direction_text = "FEL"
                color_dir = (255, 0, 0) # Kék
```

Futás közben:



A számlálás alapja egy virtuális vonal, amelyet a program a képernyő magasságának 60%-ánál húz meg. A rendszer folyamatosan figyeli az egyes járművek középpontjának (centroid) függőleges pozícióját. Amikor egy jármű közepe belép a vonal körüli szűk érzékelési sávba ( $\pm 15$  pixel), a kód vizsgálja az áthaladást. A program egy `counted_ids` nevű halmazban tárolja a már megszámolt járműveket, hogy elkerülje a többszöri számlálást. Az előzetesen meghatározott mozgásirány (FEL vagy LE) alapján a szoftver a megfelelő számláló értékét növeli eggyel. Sikeres áthaladáskor a vonal színe pillanatnyilag megváltozik, vizuális visszajelzést adva az eseményről.

Kód:

```
# --- SZÁMLÁLÁS ---
if line_y - 15 < cy < line_y + 15:
    if track_id not in counted_ids:
        if direction_text == "LE":
            count_down += 1
            counted_ids.add(track_id)
            cv2.line(frame, (0, line_y), (w, line_y), (0, 0, 255), 4)
        elif direction_text == "FEL":
            count_up += 1
            counted_ids.add(track_id)
            cv2.line(frame, (0, line_y), (w, line_y), (255, 0, 0), 4)

# --- KIRAJZOLÁS ---
# Keret
cv2.rectangle(frame, (x1, y1), (x2, y2), color_dir, 2)

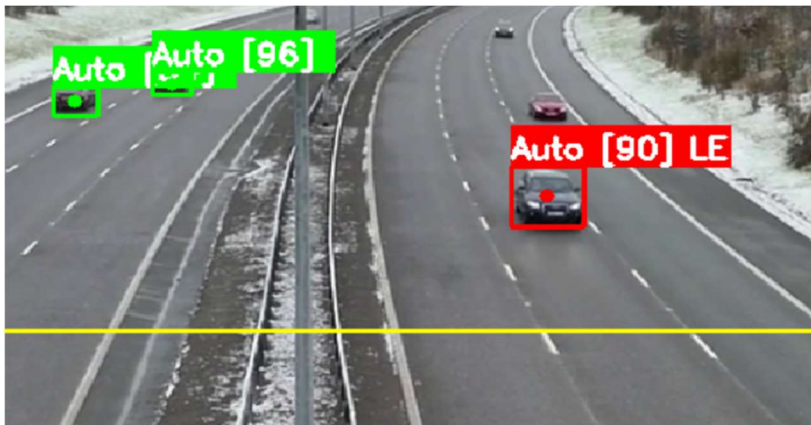
# Címke összeállítása: Típus + ID + Irány
# Pl: "Auto ID:42 LE"
veh_type = HU_NAMES.get(cls, "Jarmu")
label = f"{veh_type} [{track_id}] {direction_text}"

# Szöveg háttére (hogyan olvasható legyen)
(w_text, h_text), _ = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.6, 2)
cv2.rectangle(frame, (x1, y1 - 25), (x1 + w_text, y1), color_dir, -1)

# Szöveg kiírása
cv2.putText(frame, label, (x1, y1 - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 2)

# Pötty a közepére
cv2.circle(frame, (cx, cy), 4, color_dir, -1)
```

Futás közben:



A statisztikai táblázat a videó bal felső sarkában kapott helyet, ahol egy fekete téglalap szolgál háttérként a jobb olvashatóság érdekében. A program itt jeleníti meg külön sorokban a lefelé és a felfelé haladó járművek aktuális darabszámát. Egy harmadik sorban a kód automatikusan összeadja a két irány értékeit, így mutatva a teljes áthaladó forgalmat. A kiírt szövegek színe (piros és kék) szándékosan megegyezik a járművek körüli keretek színével a gyors vizuális tájékozódás érdekében. A felületet az OpenCV rajzoló függvényei (rectangle, putText) hozzák létre és frissítik valós időben minden egyes képkockán.

Kód:

```
# --- ÖSSZESÍTŐ TÁBLÁZAT (Dashboard) ---
# Háttér (kicsit nagyobb, hogy elférjen a 3 sor)
cv2.rectangle(frame, (10, 10), (280, 115), (0, 0, 0), -1)

# 1. sor: Lefele (Piros)
cv2.putText(frame, f"Lefele: {count_down}", (20, 40), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2)

# 2. sor: Felfele (Kék)
cv2.putText(frame, f"Felfele: {count_up}", (20, 75), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 0, 0), 2)

# 3. sor: Összesen (Fehér)
total_count = count_up + count_down
cv2.putText(frame, f"Összesen: {total_count}", (20, 110), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 255, 255), 2)

cv2.imshow("Forgalom Statisztika", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):\n    break
```

Futás közben:



A programból a q betű lenyomásával lehet kilépni.

Magához a programhoz szükséges természetesen egy videó anyag is amit a mi esetünkben erről a linkről töltöttünk le: <https://pixabay.com/videos/roads-motorways-highway-1952/>

Ezt a program mappájába helyezése után felismeri a kód.

Voltak kódrészletek, amikhez segítségül kellett vennünk a mesterséges intelligenciát.

Ilyen volt a sárga vonal, ami segített a járműszámlálásban, illetve az y koordináták felismerésében.

```
if len(track) >= 2:
    diff = track[-1][1] - track[-2][1]
    if diff > 2: # Y nő -> Lefele
        direction_text = "LE"
        color_dir = (0, 0, 255) # Piros
    elif diff < -2: # Y csökken -> Felfele
        direction_text = "FEL"
        color_dir = (255, 0, 0) # Kék
```

Felhasznált források:

- ChatGPT
- Google Gemini
- Teams kiadott segédanyagok
- PixaBay (videó)
- Huggingface