



Neumann János Egyetem
Műszaki és Informatikai
Kar

Haladó programozás

1. Járműszámlálás videó alapján

Székely Gábor Dániel
E5LG6T

2025

Bevezetés

Ez a dokumentáció a **Valós idejű Járműforgalom Számláló** demonstrációs alkalmazást mutatja be, amely **mélytanulás** és **objektumkövetés** módszerek integrációjával valósít meg automatizált forgalomfigyelést videófelvételeken.

1.1. Cél és Alkalmazási Terület

A projekt célja az **objektumdetektálás (YOLOv8)** és az **objektumkövetés (SORT)** egyszerű, de hatékony integrációjának bemutatása, egy stabil számlálási logikával kiegészítve. Az alkalmazás **forgalomfigyelésre** alkalmas (pl. új autópálya-szakaszok tehergépjármű-használatának százalékos felmérése).

1.2. Elérhetőség

https://github.com/AJake20/HaladoProg_SZGD

2. A Rendszer Architektúrája és Fő Komponensei

Az alkalmazás egy tipikus Computer Vision pipeline-t követ, ahol a feldolgozás képkockáról képkockára történik.

Komponens	Szerep	Implementáció
<i>Videó bement</i>	<i>A feldolgozandó forrás (videófájl vagy kamera).</i>	<i>OpenCV (cv2)</i>
<i>Objektum Detektor</i>	<i>Az objektumok (pl. teherautók) azonosítása és határ-dobozok (Bounding Box - BBox) megadása minden képkockán.</i>	<i>YOLOv8s (YOLOv8 Small) – ultralítics</i>
<i>Objektum Követés</i>	<i>A detektált BBox-ok összerendelése a korábbi képkockákon lévő objektumokkal, és egyedi azonosító (Track ID) hozzárendelése.</i>	<i>SORT (Simple Online and Realtime Tracking)</i>
<i>Számláló Logika</i>	<i>A járművek áthaladásának regisztrálása két virtuális vonal között.</i>	<i>Egyedi Python logika</i>
<i>Vizualizáció</i>	<i>Eredmények, BBox-ok és vonalak megjelenítése a kimeneti videón.</i>	<i>OpenCV (cv2), cvzone</i>

2.1. Objektum Detektálás: YOLOv8

Az alkalmazás **YOLOv8s** modellt használ a járművek felismerésére. A YOLO (You Only Look Once) egy rendkívül gyors, end-to-end mélytanulási architektúra, amely egyetlen hálózati futtatással adja meg a detektált objektumok helyét és osztályát.

- Bemenet:** Videóképkocka.
- Kimenet:** A detektált objektumok listája, mindegyikhez:
 - Határ-doboz koordinátái: (x1, y1, x2, y2)
 - Konfidencia Pontszám (Score): A detektálás valószínűsége.
 - Osztály (Class ID): Pl. 2 (car), 7 (truck).

2.2. Objektum Követés: SORT

A **SORT** algoritmus célja, hogy az egymást követő képkockákon detektált objektumokat összerendelje (Data Association).

- **Kalman-szűrő:** Előre jelzi az objektumok következő pozícióját a korábbi mozgásuk alapján. Ez segít a detektálás hiánya esetén is megtartani a nyomvonalaat.
- **IoU-alapú Asszociáció:** Az előre jelzett pozíciót a tényleges detektálásokkal egy **Intersection over Union (IoU)** nevű metrika alapján veti össze. Ez a metrika adja meg a két doboz átfedésének mértékét.
- **Track ID:** minden sikeresen követett objektum egyedi ID-t kap.

A SORT bemenete a YOLO-detektálás, formátuma: N x 5 numpy tömb: [x1, y1, x2, y2, score]. (Bewley, 2023)

3. Számlálási Logika Részletes Elemzése

A számlálási mechanizmus a követési azonosítók (**Track ID-k**) két, előre definiált horizontális vonalhoz viszonyított elhelyezkedésén alapul.

3.1. Zóna Paraméterek

Paraméter	Érték	Leírás
Vörös Vonal	$y = 250$	<i>A felső virtuális vonal a számlálási zónában.</i>
Kék Vonal	$y = 300$	<i>Az alsó virtuális vonal a számlálási zónában.</i>
Tolerancia (offset)	15 pixel	<i>Egy objektum akkor tekinthető a vonalon áthaladónak, ha a középpontjának y koordinátája a vonal + offset tartományában van</i>

3.2. Áthaladás Detektálása

Az áthaladás regisztrálása két lépésben történik, a jármű középpontjának (cx, cy) koordinátája alapján.

- **Irány meghatározása (Lépés 1: Előkészítés):** A rendszer rögzíti, ha egy ID átlép egy vonalat.
 - Le irány: Ha a jármű először a Vörös Vonalnál halad át ($y=250+{\text{offset}}$), a rendszer tárolja az ID-t a down szótárban.
 - Fel irány: Ha a jármű először a Kék Vonalnál halad át ($y=300+{\text{offset}}$), a rendszer tárolja az ID-t a up szótárban.
- **Számlálás (Lépés 2: Befejezés):** Csak akkor történik meg a számlálás, ha az ID már rögzítve van az 1. lépésben, és áthalad a második vonalon.
 - Le Felé Számlálás: Ha egy ID benne van a down szótárban ÉS áthalad a Kék Vonalnál ($y=300+{\text{offset}}$), akkor egyszer számláljuk.
 - Fel Felé Számlálás: Ha egy ID benne van a up szótárban ÉS áthalad a Vörös Vonalnál ($y=250+{\text{offset}}$), akkor egyszer számláljuk.

3.3. Ismételt Számlálás Elkerülése

A **Track ID** egyszeri számlálását a **halmazok (set)** használata biztosítja:

- counter_down_set
- counter_up_set

Amikor egy áthaladás sikeresen regisztrálásra kerül (pl. lefelé), az ID bekerül a megfelelő halmazba. A további képkockákon az „if id not in counter_down_set:” feltétel megakadályozza, hogy ugyanazt az ID-t újra beleszámítsák, még akkor is, ha az objektum középpontja ideiglenesen visszakerül a vonal +-offset tartományába.

4. Használati Útmutató

4.1. Telepítés és Futtatás

4.1.1. Környezet Előkészítése (Windows PowerShell)

1. Virtuális környezet létrehozása és aktiválása:

```
python -m venv .venv  
.venv\Scripts\Activate
```

2. **Függőségek telepítése:** Előfeltétel: rendelkezésre kell állnia egy requirements.txt fájlnak, amely tartalmazza az ultralytics, opencv-python, cvzone, numpy, és pandas könyvtárakat.
pip install -r requirements.txt

4.1.2. Futtatás

- Bemeneti videó megadása.
- python main.py
- A megjelenített videóablakból a q billentyűvel lehet kilépni.

4.2. Fontos Konfigurációs Pontok (main.py)

A rendszer finomhangolásához a következő kulcsparaméterek módosítása szükséges:

- **Modell és Bemenet:**

```
model = ultralytics.YOLO('yolov8s.pt') # Modellfájl útvonala  
cap = cv2.VideoCapture('videos/vehicle-counting.mp4')
```

- **Követett Osztályok:** A megadott kódban a selected_classes = "truck"
- **SORT Tracker Paraméterei:** A követés stabilitásának kulcsa.

```
tracker = Sort(max_age=20, min_hits=3, iou_threshold=0.3)  


- max_age: Hány képkocka után töröljük a nyomvonalaat, ha nem detektáljuk az objektumot.
- min_hits: Hány detektálás szükséges a nyomonalat végreleges megerősítéséhez.
- iou_threshold: Az asszociációhoz szükséges minimális IoU érték.

```

- **Számlálási Zóna:**

```
red_line_y = 250  
blue_line_y = 300  
offset = 15
```

Irodalomjegyzék

Bewley, A. (2023. november 28.). *sort*. Forrás: abewley/sort:
<https://github.com/abewley/sort>