

# Jármű detektáló rendszer élő kamerakép alapján

**py-snake**

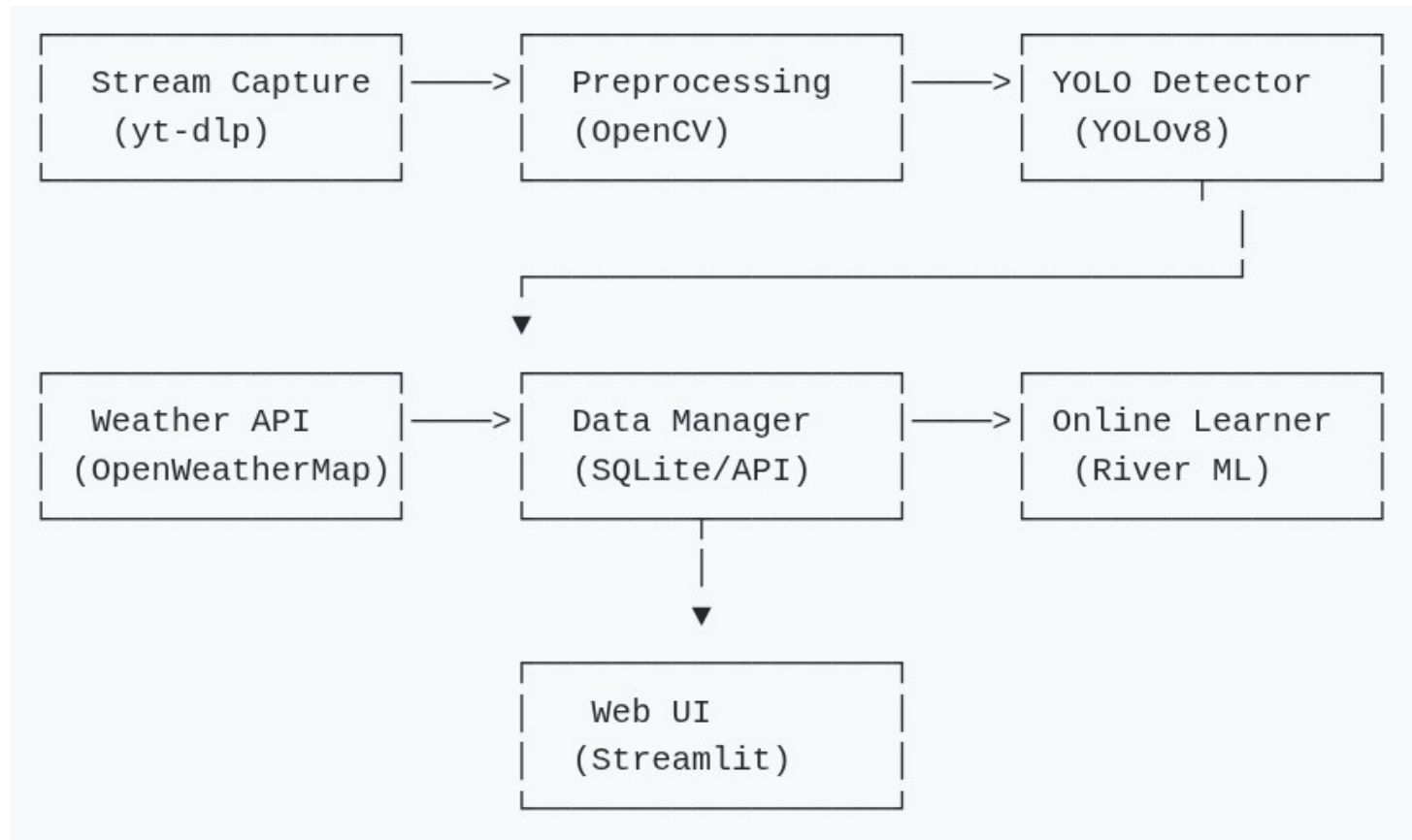
# Projekt célja

- Internetes közvetítés alapján forgalom figyelés
- Úton haladó járművek detektálása és számlálása
- Kamerakép javítása nappali és éjjeli használatához
- Web-es felület a kezeléshez, kép figyeléséhez
- Online tanulás, előrejelzés

# Felépítés

- Docker konéterekekre épül
- Python alapú programok
- Kamerakép: YouTube-ról
- Képfeldolgozás: OpenCV
- Detektálás: YOLOv8
- Web-es felület

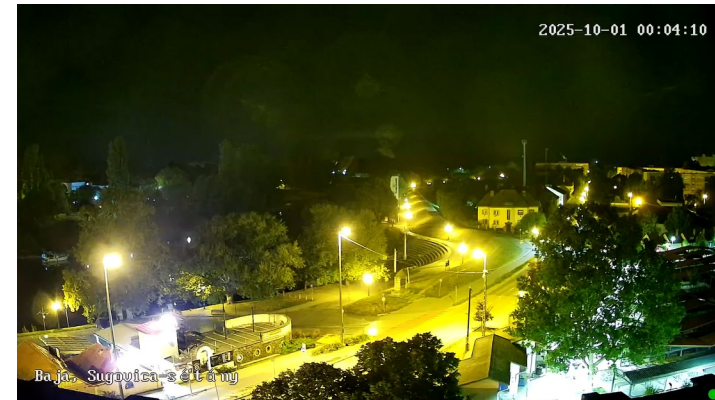
# Architektúra



# Képfeldolgozás

- Hasznos terület kivágása
- Adaptív javítás nappali és éjjeli fényviszonyokhoz
- Képjavítás:
  - Fényerő
  - Kontraszt
  - Gamma
- Háttér kivonás

# Képek



# Detektált kép

## Performance

Preprocessing

24.7 ms

↑ avg: 26.1 ms

YOLO Detection

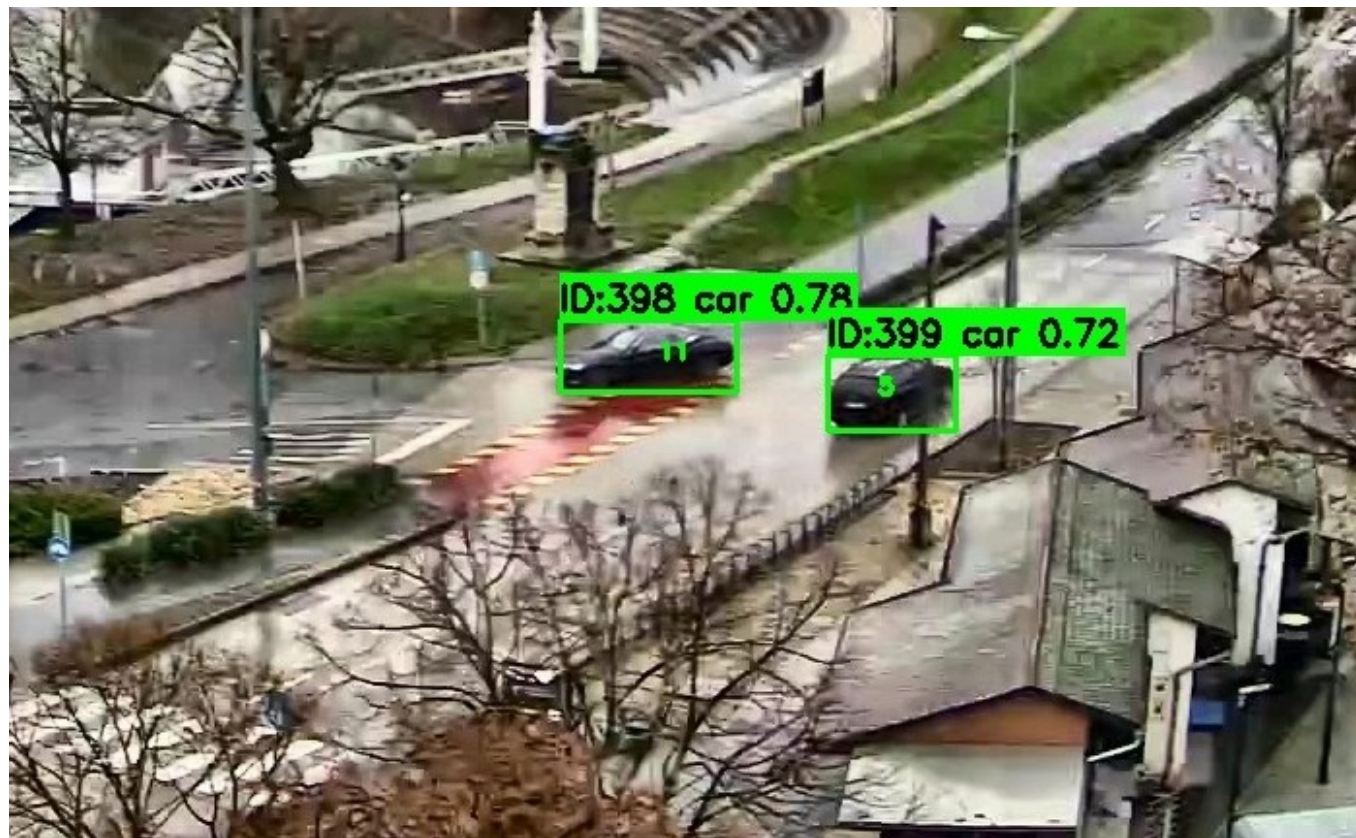
75.3 ms

↑ avg: 74.2 ms

Overall

100.0 ms

↑ 10.0 FPS



# Dashboard

## Stream Controls

☐ Auto-refresh

Refresh rate (FPS)



Refresh Now

## Background Subtraction

☐ Enable Background Subtraction

## Performance

Preprocessing

26.4 ms

↑ avg: 25.7 ms

YOLO Detection

80.3 ms

↑ avg: 72.5 ms

Overall

106.7 ms

↑ 9.4 FPS

## Flowium - Live Stream Test

[Live Stream](#) [Crop Configuration](#) [Analytics](#) [Predictions](#)

Testing YouTube stream capture with yt-dlp

### Raw Stream Capture



Raw - 13:36:50

Full Resolution: 1920 x 1080

### Preprocessed View



Processed - 13:36:50

### Vehicle Detection



Detected - 13:36:50

Total Vehicles: 0

## System Status

Preprocessing Service

Frames Available

YOLO Detector

## Recent Detections (Last 10)

# Feldolgozott képek

## Performance

### Preprocessing

20.6 ms

↑ avg: 25.6 ms

### YOLO Detection

73.0 ms

↑ avg: 77.3 ms

### Overall

93.7 ms

↑ 10.7 FPS

## Raw Stream Capture



Raw - 11:41:31

Full Resolution: 1920 x 1080

## Preprocessed View



Processed - 11:41:31

## Vehicle Detection



Detected - 11:41:31

Total Vehicles: 1

# Teljesítmény

- Feldolgozás:
- 30 FPS kamera -> 5 FPS feldolgozás -> 3-10 FPS detektálás
- Pytorch alapú .pt modell: 200-300ms egy kép
- ONNX formátum: 80-100ms egy kép
- Online tanulás
- Valós időben tanul, nincs nagy számítási igény

# Korlátok

- Nem teljesen ideális kép: távoli, csak kis területre van szükség
- Csak CPU áll rendelkezésre: korlátozott lehetőségek, csökkenteni kell a felbontást és az FPS-t a valósidejű feldolgozáshoz
- CPU miatt legkisebb YOLO modell használható
- nano, small 3x több idő

# Kiegészítő lehetőségek

- Adatbázis, eddigi forgalom tárolása
- Online tanuló modell valósidejű tanuláshoz és becslésekhez a jövőre nézve
- Időjárási viszonyok figyelembevétele

# Köszönöm a figyelmet!