

Járművek trajektóriáinak előrejelzése machine learning modellekkel

Péter Bence Mérnökinformatika BSc 6. félév

Témavezetők:
Dr. Horváth András egyetemi docens
Agg Áron PhD hallgató

Széchenyi István Egyetem

TMDK 2023. április 26.

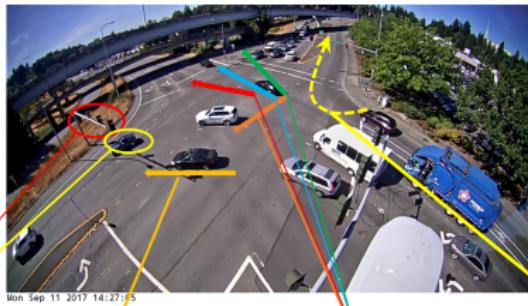
Tartalom

- 1 Bevezetés
- 2 Objektumdetektálás
- 3 Trajektória
- 4 Objektumkövetés
- 5 Machine Learning
 - Unsupervised learning (Nem felügyelt tanulás)
 - Supervised learning (Felügyelt tanulás)
- 6 Tipikus viselkedési formák meghatározása: Klaszterezés (Unsupervised)
 - Adatok tisztítása
 - Zajos vs Szűrt
 - OPTICS vs KMeans, BIRCH
 - OPTICS vs DBSCAN
- 7 Klasszifikáció (Supervised)
 - Klasszifikációs algoritmusok
- 8 Megjelenítő alkalmazás
- 9 Összefoglalás
- 10 Vége

Bevezetés

- Adathalmaz:

- ▶ 5 különböző helyszín amerikai városokból
- ▶ 20 órányi felvétel



yolov7

Objektumok
megkülönböztetése
Rendőrlámpa vs.
autó

DeepSORT



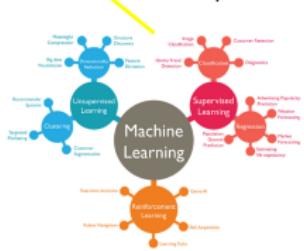
Objektum mozgásának követése

Where is the Deep Learning
all of this?

yolov7

Trajektóriák
megkülönböztetése:
klasszifikálás
Unsupervised

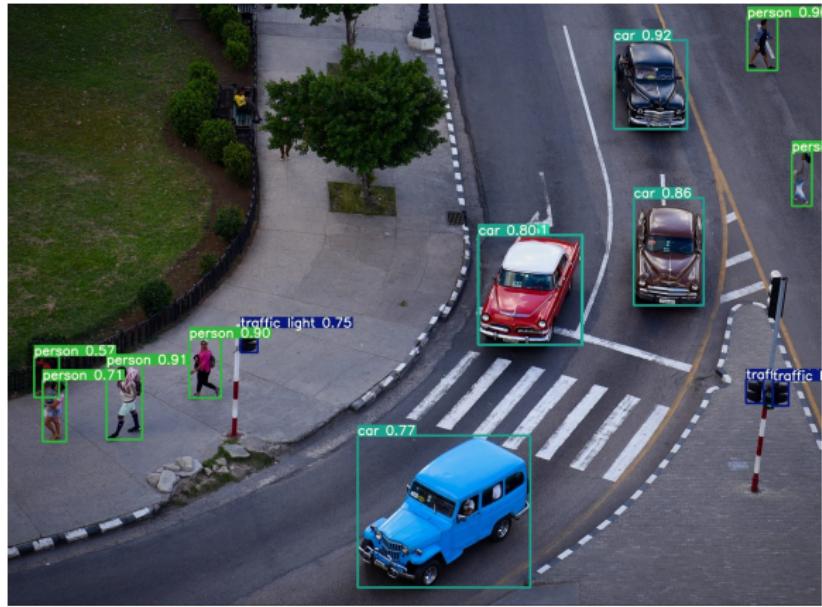
Trajektóriák
előrejelzése (predikció)
a klasszifikációk alapján
Supervised



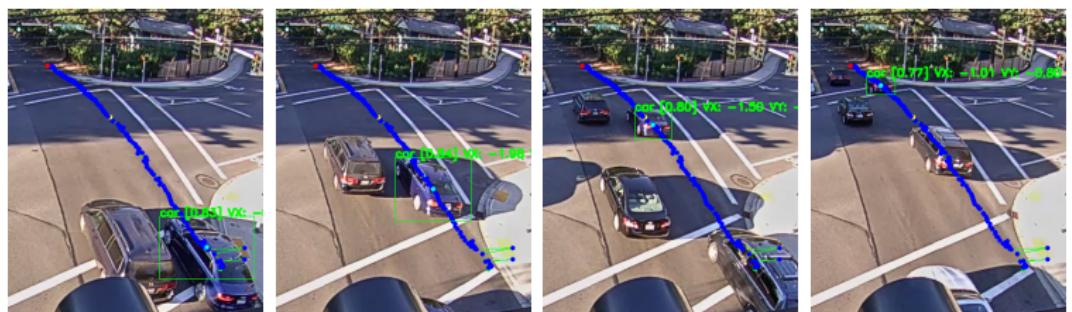
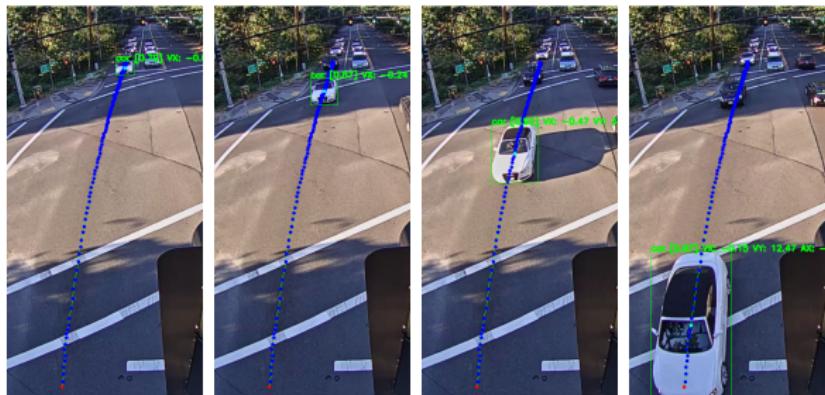
Objektumdetektálás

YOLOv7

- A világban százezrek használják
- Nagy pontosság, valós időben egy erős videókártyán

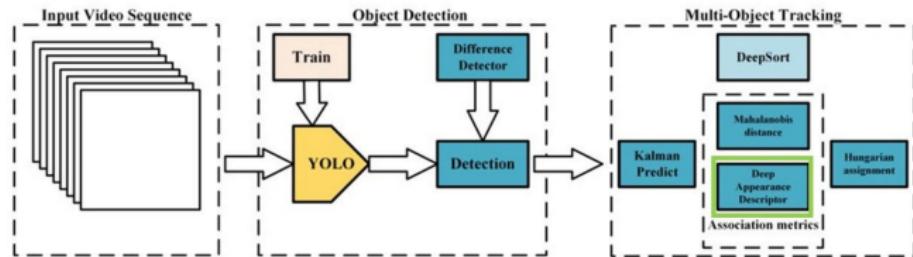


Trajektória



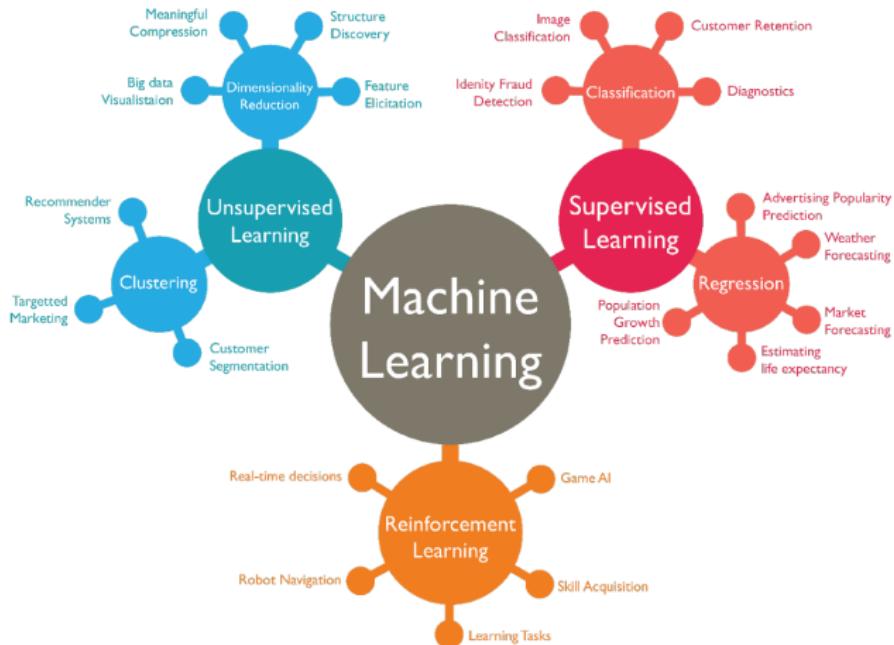
DeepSORT

Where is the Deep Learning
in all of this?



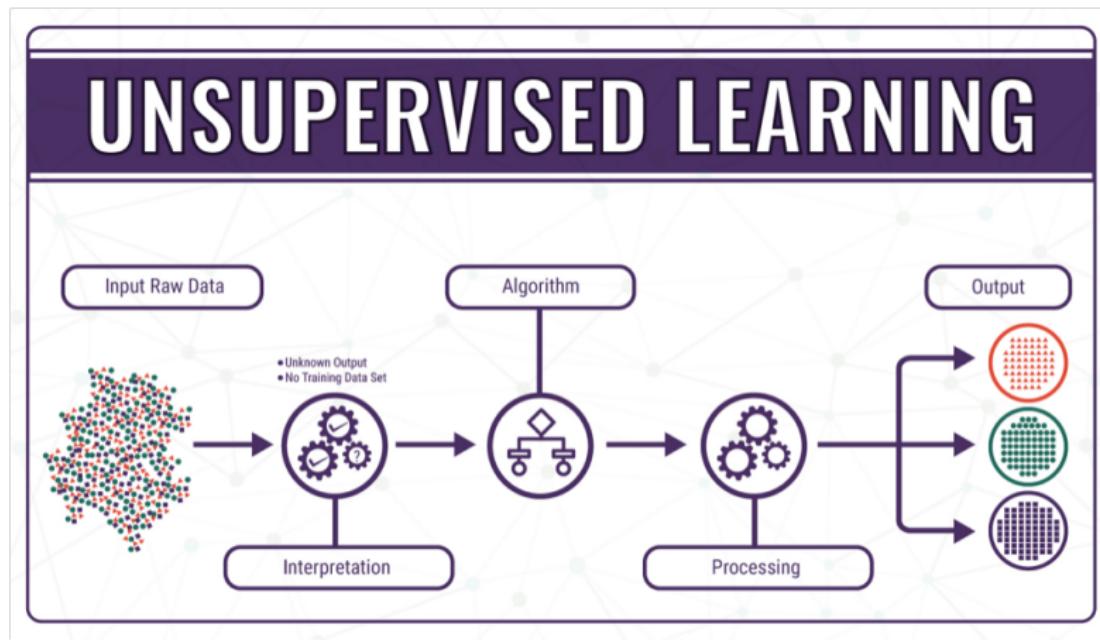
- YOLO detektálások összefűzése trajektóriákká

Machine Learning



Unsupervised learning (Nem felügyelt tanulás)

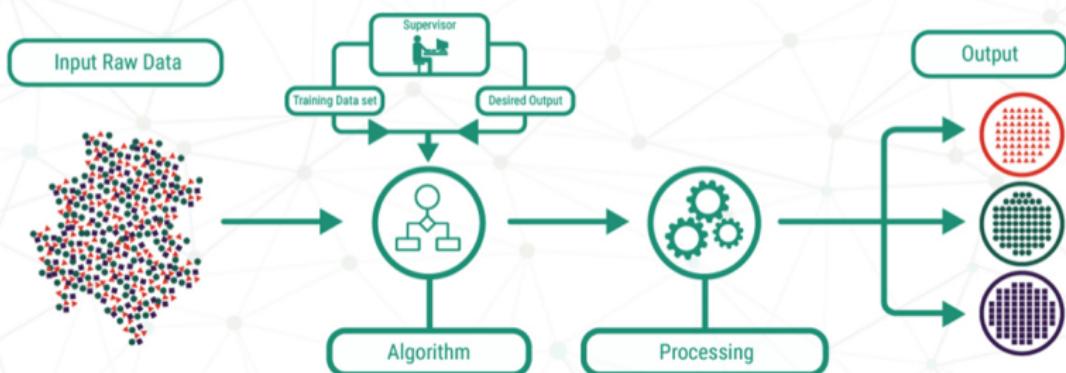
- Nincsenek előre meghatározott osztályok
- Az algoritmus saját maga próbálja csoportokba rendezni a adatokat



Supervised learning (Felügyelt tanulás)

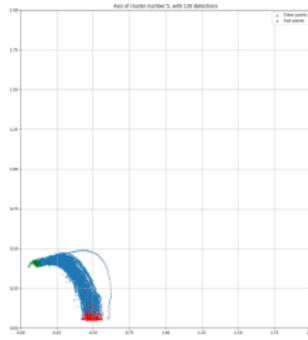
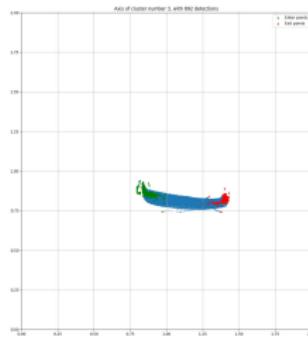
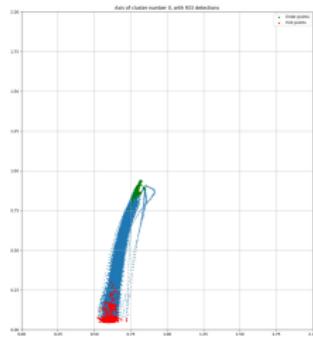
- Előre meghatározott osztályok alapján
- Az algoritmust tanítani kell példa adatokkal

Supervised Learning



Tipikus viselkedési formák meghatározása: Klaszterezés (Unsupervised)

- Bemeneti adatok: gyűjtött trajektóriák objektumdetektálás és követés segítségével
- Feature vektorok: [bemeneti x,y koordináta, kimeneti x,y koordináta]



Adatok tisztítása

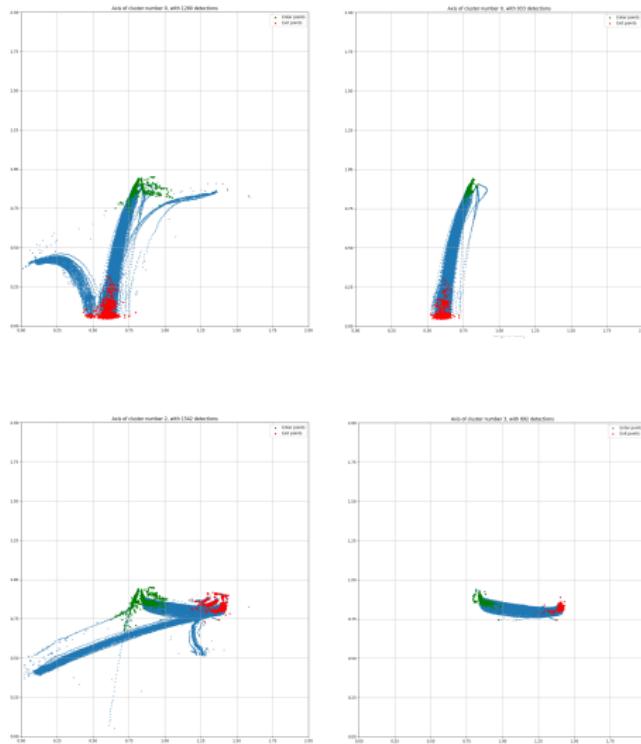
- Több különböző zajforrás

- ▶ YOLO késői detektálás és követés (a járművet már csak akkor veszi észre amikor bennt van a kereszteződésben)
- ▶ YOLO fals detektálás (rendőrlámpát vagy táblát autónak néz)
- ▶ DeepSORT áttapadások (egymáshoz közel elhaladó járművek identitása felcserélődik)

- Megoldások

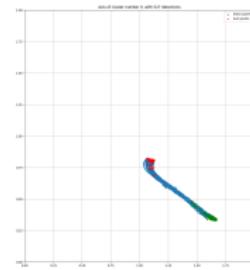
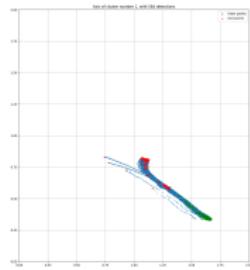
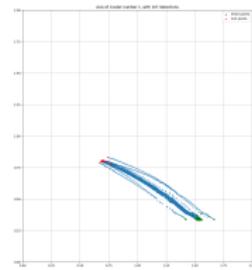
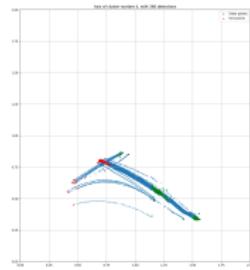
- ▶ Kép relatív széleinek megtalálása min-max kereséssel (az utakat nem biztos, hogy a kép szélétől széléig látjuk, hanem a kép közepétől kezdődve, ezt okozhatja egy nagy épület)
- ▶ Szűrés trajektóriák kezdő és végpontjainak euklideszi távolsága alapján
- ▶ Szűrés trajektóriák egymást követő detektálásai közötti távolságokkal

Zajos vs Szűrt

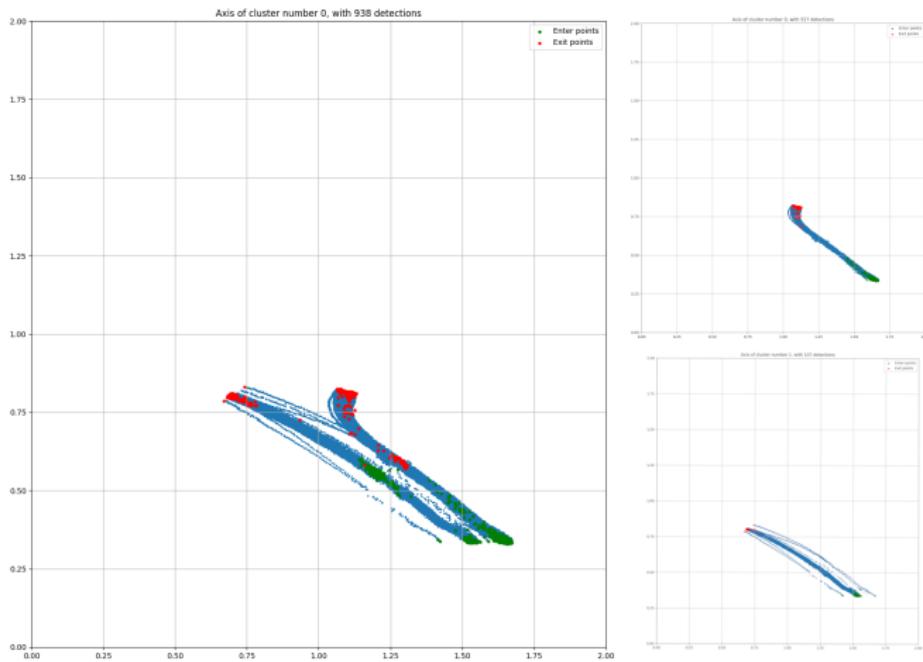


OPTICS vs KMeans, BIRCH

- Klaszterezés nem triviális feladat
- Több algoritmus tesztelése
- KMeans, BIRCH, DBSCAN, OPTICS
- OPTICS adta a legjobb eredményeket



OPTICS vs DBSCAN



Klasszifikáció (Supervised)

- Megfigyelt autó besorolása a klaszterezés alapján meghatározott csoportba.
- Feature vektorok: trajektóriák felosztása több kisebb részre, amik valós idejű futásból származó trajektóriákhoz hasonlítanak
 - ▶ v1: [kezdőkoordinátája, sebessége, középső koordinátája, utolsó koordinátája, sebessége]
 - ▶ v7: [kezdőkoordinátája ·1, sebessége ·100, utolsó koordinátája ·2, sebessége ·200]
- Adathalmaz dúsítása:
 - ▶ Cél: kiegyenlítettebb darabszámok
 - ▶ egy trajektóriából több feature vektor generálása, tanító adathalmaz megnövelése
- Trajektóriák hossza: 15 és 30 hosszúságú trajektóriák tesztelése
- Eredmény: hossz növeléssel nem történik pontosságban javulás

Klasszifikációs algoritmusok

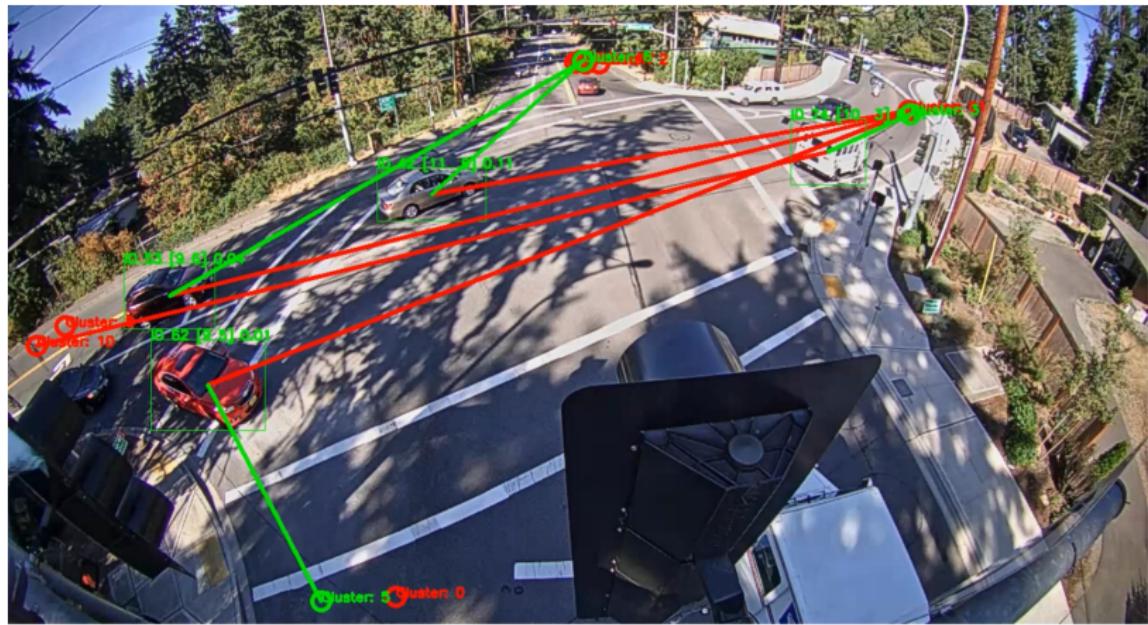
- K-NearestNeighbors, SupportVectorMachine, DecisionTree
- 5 különböző helyszín adatain mért pontosságok átlaga
- Top1: legvalószínűbb előrejelzés tényleg az elvárt osztály
- Top2: 2 legvalószínűbb előrejelzésben benne van az elvárt osztály

Average Accuracy Feature Vector v1			
	Balanced	Top 1	Top 2
KNN	94.16%	96.71%	99.46%
SVM	81.65%	90.82%	98.05%
DT	93.11%	94.88%	96.03%

Average Accuracy Feature Vector v7			
	Balanced	Top 1	Top 2
KNN	92.08%	95.61%	98.66%
SVM	88.72%	93.86%	98.92%
DT	89.46%	93.30%	94.55%

Megjelenítő alkalmazás

- Saját fejlesztésű megjeleítő alkalmazás
- Zöld: legvalószínűbb előrejelzés
- Piros: második legvalószínűbb előrejelzés



Összefoglalás

- YOLOv7 és DeepSORT detektáló alkalmazásba integrálása
- SQL Adatbázis architektúra tervezése, implementálása
- Klaszterezés: Scikit-Learn
- Klasszifikáció: Scikit-Learn
- Klasszifikációs Modellek tárolása: Pickle, Joblib
- Megjelenítő alkalmazás
- Fejlesztői környezet: Linux, Python, Codium
- Saját forráskód mérete: 7460 sor
- Implementáció megtalálható itt:
https://github.com/Pecneb/computer_vision_research

Vége

Köszönöm a figyelmet

