**UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

# PROIECT

## SISTEME BAZATE PE CUNOAȘTERE

# Cruise Control Adaptiv

**Student:** Kovacs Erik

Anul III, Semestrul I

**Coordonator:** Conf. Dr. Ing. Roxana Rusu Both

# CUPRINS

# 1. INTRODUCERE

Prezentul proiect își propune dezvoltarea unui sistem autonom de navigație pentru vehicule, bazat pe tehnici avansate de învățare automată și viziune computerizată. Sistemul implementează două componente principale: un modul de segmentare semantică pentru identificarea elementelor din mediul rutier și un modul de detecție a obiectelor pentru recunoașterea semnelor de circulație și participanților la trafic.

Contextul acestui proiect se înscrie în tendința actuală de automatizare a vehiculelor și dezvoltare a sistemelor ADAS (Advanced Driver Assistance Systems). Vehiculele autonome și semiautonome necesită capacitatea de a percepe și interpreta mediul înconjurător în timp real, adaptându-și comportamentul în funcție de condițiile de trafic, vreme și obstacole.

## 1.1 Obiectivele Proiectului

Obiectivele principale ale proiectului sunt:

- Dezvoltarea unui sistem de segmentare semantică pentru identificarea elementelor din scenă (drum, trotuare, clădiri, vegetație, vehicule)
- Implementarea unui modul de detecție a obiectelor pentru recunoașterea semnelor rutiere și participanților la trafic
- Crearea unui sistem de decizie care adaptează viteza și comportamentul vehiculului în funcție de condițiile detectate
- Simularea unui vehicul autonom care răspunde la semne, semafoare, proximitatea altor vehicule și condițiile meteo

## 1.2 Structura Documentației

Documentația este structurată în următoarele capitole: fundamentarea teoretică a metodelor utilizate, arhitectura sistemului, detalii de implementare, rezultate experimentale și concluzii. Fiecare capitol prezintă aspecte tehnice detaliate, însoțite de rezultate vizuale și metrici de performanță.

# 2. FUNDAMENTARE TEORETICĂ

## 2.1 Segmentarea Semantică

Segmentarea semantică este o sarcină fundamentală în viziunea computerizată, care constă în clasificarea fiecărui pixel dintr-o imagine într-una dintre clasele predefinite. Spre deosebire de clasificarea la nivel de imagine sau detecția obiectelor (care produc bounding boxes), segmentarea semantică oferă o înțelegere precisă, la nivel de pixel, a scenei.

### 2.1.1 Arhitectura FCN (Fully Convolutional Network)

În acest proiect, s-a utilizat arhitectura FCN-ResNet50, care combină un backbone ResNet50 (pentru extracția caracteristicilor) cu straturi de deconvoluție pentru upsampling. FCN elimină straturile fully-connected tradiționale, păstrând doar convoluții, ceea ce permite procesarea imaginilor de orice dimensiune și menținerea informației spațiale.

Avantajele FCN pentru segmentarea semantică:

- Procesare end-to-end la nivel de pixel
- Capacitate de transfer learning din modele pre-antrenate (ResNet50)
- Skip connections pentru păstrarea detaliilor fine
- Performanță bună pe CPU prin optimizări arhitecturale

## 2.2 Detecția Obiectelor cu YOLO

YOLO (You Only Look Once) este o familie de modele de detecție a obiectelor bazate pe rețele neuronale convoluționale, care abordează problema detecției ca pe o sarcină de regresie unică, spre deosebire de metodele clasice care utilizează region proposals.

### *2.2.1 YOLOv8 - Caracteristici*

YOLOv8 reprezintă ultima versiune a familiei YOLO la momentul dezvoltării acestui proiect. Principalele caracteristici includ:

- Arhitectură CSPDarknet pentru extracție de caracteristici
- Path Aggregation Network (PANet) pentru fuziunea multi-scară
- Anchor-free detection pentru simplificare și performanță
- Procesare în timp real și acuratețe îmbunătățită

## *2.3 Funcția de Pierdere și Optimizare*

Pentru antrenarea modelului de segmentare semantică, s-a utilizat Cross-Entropy Loss cu ignore_index pentru a exclude pixelii necunoscuți din calcul. Optimizatorul Adam a fost ales pentru convergență rapidă și stabilitate.

Metricile de evaluare utilizate:

- Pixel Accuracy: procentul de pixeli corect clasificați
- Mean Intersection over Union (mIoU): măsoară suprapunerea dintre predicții și ground truth
- IoU per clasă: permite identificarea claselor cu performanță scăzută

# 3. ARHITECTURA SISTEMULUI

Sistemul dezvoltat este compus din două module principale care funcționează complementar: modulul de segmentare semantică și modulul de detecție a obiectelor. Acestea sunt integrate într-un simulator care ia decizii de conducere în timp real.

## 3.1 Modulul de Segmentare Semantică

Acest modul utilizează FCN-ResNet50 antrenat pe dataset-ul CARLA pentru clasificarea la nivel de pixel. Sistemul identifică 12 clase semantice:

- Traffic Sign (semne de circulație)
- Building (clădiri)
- Fence (garduri)
- Pedestrian (pietoni)
- Pole (stâlpi)
- Road Line (marcaje rutiere)
- Road (drum)
- Sidewalk (trotuar)
- Vegetation (vegetație)
- Car (mașini)
- Wall (ziduri)
- Unlabeled (zone neclasificate)

### 3.1.1 Procesarea Datelor

Imaginile de intrare sunt redimensionate la 96×192 pixeli pentru procesare rapidă pe CPU. Măștile RGB sunt convertite în indici de clasă folosind un lookup table (LUT) pentru eficiență maximă. Această abordare reduce timpul de conversie de la ordinul milisecundelor la microsecunde.

## 3.2 Modulul de Detecție a Obiectelor

Modulul YOLO detectează obiecte specifice relevante pentru conducerea autonomă: vehicule, pietoni, biciclete, semafoare, semne de viteză, semne stop. Modelul returnează bounding boxes, confidence scores și clase pentru fiecare obiect detectat.

### 3.2.1 Calcul Proximitate Vehicul

Pentru estimarea distanței față de vehiculul din față, sistemul calculează raportul dintre aria bounding box-ului și aria totală a imaginii (Lead Vehicle Area Ratio). Acest proxy simplu dar eficient permite decizii de frânare fără necesitatea senzorilor de distanță:

- Area Ratio < 0.08: Distanță sigură, mențin viteza
- Area Ratio 0.08-0.15: Vehicul aproape, inițiez frânare
- Area Ratio > 0.15: Vehicul foarte aproape, STOP complet

## 3.3 Sistemul de Decizie

Sistemul de decizie combină informațiile de la ambele module și date despre condițiile meteo pentru a determina acțiunea optimă. Logica de decizie este organizată ierarhic, cu prioritate pentru siguranță:

- Prioritate 1: Oprire pentru vehicul prea aproape (Area Ratio > 0.15)
- Prioritate 2: Frânare de urgență pentru Stop/Semafor roșu
- Prioritate 3: Respectare limite de viteză
- Prioritate 4: Adaptare viteză la condițiile meteo
- Prioritate 5: Cruise control normal

### 3.3.1 Adaptare la Condițiile Meteo

Viteza de bază (50 km/h) este ajustată prin multiplicatori în funcție de vreme:

- Sunny/Clear: 100% (50 km/h)
- Cloudy/Overcast: 90% (45 km/h)
- Night: 85% (42 km/h)
- Wet: 80% (40 km/h)
- Rainy: 75% (38 km/h)
- Fog/Foggy: 65% (32 km/h)

# 4. IMPLEMENTARE

## 4.1 Dataset și Preprocesare

S-a utilizat dataset-ul CARLA Semantic Segmentation, care conține 28 de videoclipuri înregistrate în diverse condiții meteo și scenarii de conducere. Dataset-ul a fost împărțit la nivel de video pentru a evita data leakage:

- Training set: 70% (19 videoclipuri)
- Validation set: 15% (4 videoclipuri)
- Test set: 15% (5 videoclipuri)

Pentru antrenare rapidă pe CPU, imaginile au fost redimensionate la 96×192 pixeli folosind BILINEAR pentru imagini și NEAREST pentru măști (pentru a păstra clasele exacte). S-a aplicat augmentare de date prin horizontal flip cu probabilitate 50%.

## 4.2 Antrenarea Modelului de Segmentare

Configurația de antrenare:

- Model: FCN-ResNet50 (torchvision.models.segmentation)
- Epoci: 2 (optimizat pentru CPU)
- Batch size: 1
- Optimizer: Adam (lr=1e-4)
- Loss: CrossEntropyLoss cu ignore_index=255
- Device: CPU (Intel i7 12th gen)

Pentru a accelera procesul, validarea în timpul antrenării a fost limitată la 150 de batch-uri, reducând timpul per epoch fără a compromite monitorizarea convergentului.

## 4.3 Antrenarea Modelului YOLO

Pentru detecția obiectelor, s-a utilizat YOLOv8n (varianta nano pentru rapiditate) antrenat pe aceleași date CARLA. Modelul a fost fine-tunat pentru a recunoaște clase specifice conducerii autonome: car, truck, bus, pedestrian, bicycle, traffic light, stop sign, speed limit signs.

## 4.4 Simulatorul Autonom

Simulatorul încarcă frame-uri aleatorii din dataset, aplică ambele modele (segmentare + detecție), calculează metricile de proximitate și afișează decizia în timp real. Interfața vizuală include:

- Bounding boxes colorate pentru obiecte detectate
- Overlay semi-transparent cu informații de acțiune
- Metrici în timp real: Lead Vehicle Area Ratio, viteză recomandată, condiții meteo
- Lista obiectelor detectate de senzori

# 5. REZULTATE EXPERIMENTALE

## 5.1 Performanța Segmentării Semantice

Rezultatele finale pe setul de test:

- Test Loss: 0.2616
- Pixel Accuracy: 90.71%
- Mean IoU: 48.99%

**Figura 1: Rezultate Test - Metrici de Performanță**

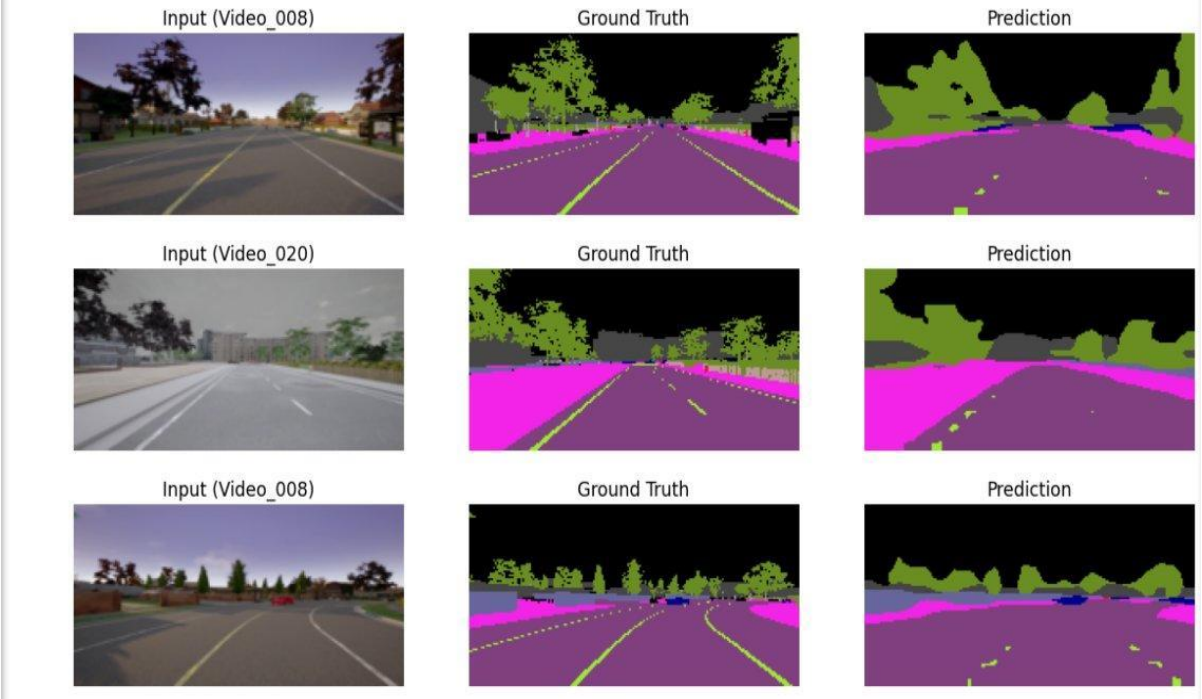
```
TEST RESULTS
  test_loss: 0.26164002581112505
  pixel_acc: 0.9070691038758203
  mIoU: 0.48997115226552684

IoU per class:
  00 Traffic Sign          IoU=0.0000
  01 Building              IoU=0.8015
  02 Fence                 IoU=0.1646
  03 Pedestrian            IoU=0.2340
  04 Pole                  IoU=0.0383
  05 Road Line             IoU=0.2521
  06 Road                  IoU=0.9092
  07 Sidewalk              IoU=0.6505
  08 Vegetation            IoU=0.5933
  09 Car                   IoU=0.7837
  10 Wall                  IoU=0.5275
  11 Unlabeled             IoU=0.9249
```

Analizând IoU per clasă, observăm performanțe variabile:

- Clase cu IoU ridicat (>70%): Car (78.37%), Road (90.92%), Unlabeled (92.49%)
- Clase cu IoU mediu (40-70%): Building (80.15%), Sidewalk (65.05%), Vegetation (59.33%), Wall (52.75%)
- Clase cu IoU scăzut (<40%): Road Line (25.21%), Pedestrian (23.40%), Fence (16.46%), Pole (3.83%), Traffic Sign (0%)

IoU-ul scăzut pentru Traffic Sign și Pole se datorează dimensiunii reduse a acestor obiecte în imagine și rezoluției mici folosite pentru antrenare (96×192). Cu toate acestea, clasele critice pentru navigație (Road, Car) au performanțe excelente.

**Figura 2: Exemple de Segmentare Semantică**




## *5.2 Evoluția Antrenării*

**Figura 3: Curbe de Loss (Train vs Validation)**

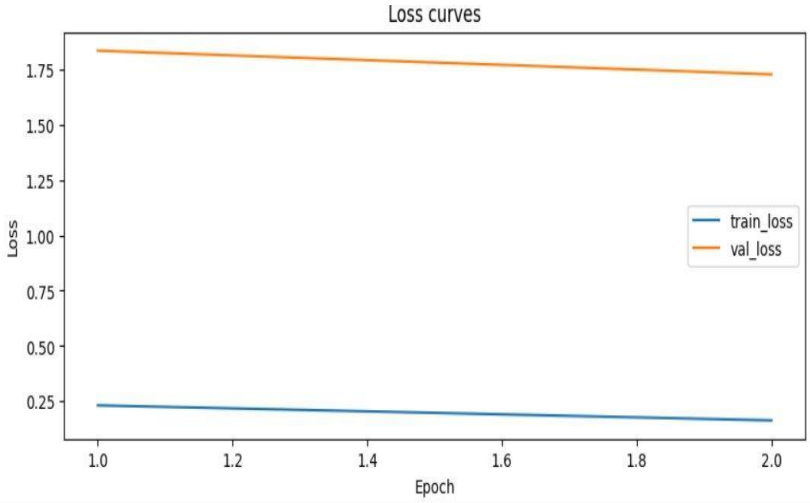
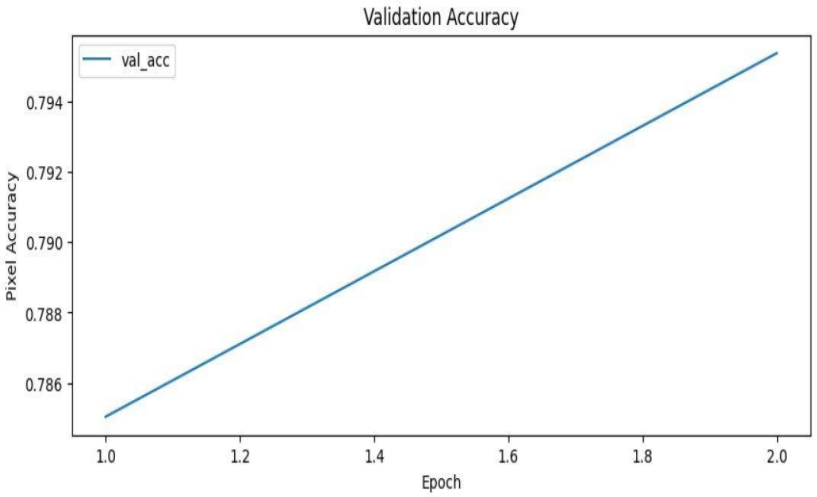


**Figura 4: Evoluția Validation Accuracy**



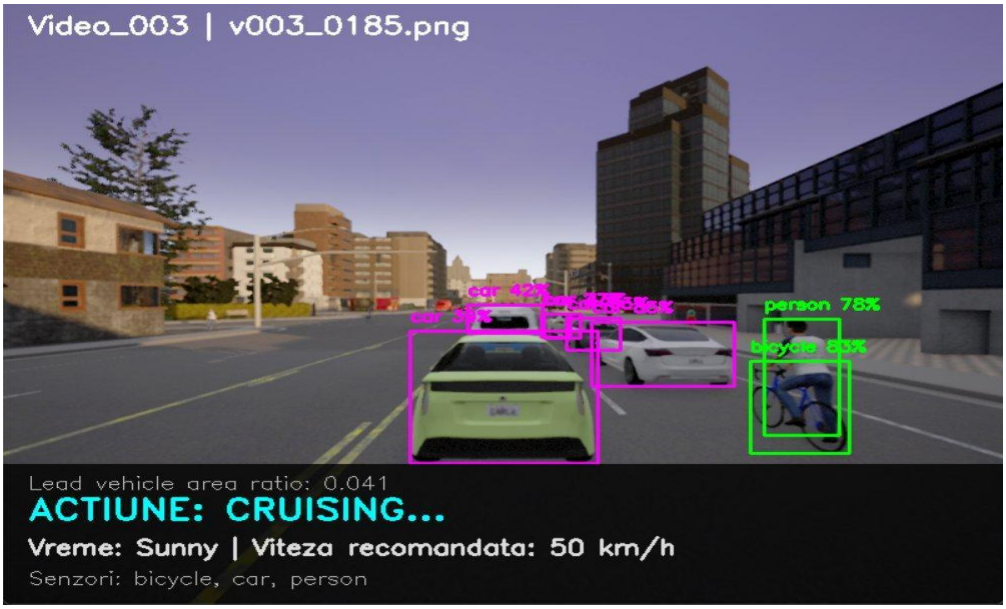

Curbele de antrenare arată convergență clară în cele 2 epoci. Validation accuracy crește constant de la 78.5% la 79.5%, indicând că modelul învață fără overfitting semnificativ. Diferența mică între train loss și validation loss sugerează că modelul generalizează bine.

## *5.3 Performanța Sistemului de Decizie*

Simulatorul a fost testat pe diverse scenarii din dataset, demonstrând capacitatea de a răspunde corect la:

### *5.3.1 Scenariul CRUISING*

**Figura 5: Cruising Normal - Condiții Sunny**




În condiții normale (sunny, fără obstacole apropiate), sistemul menține viteza recomandată de 50 km/h. Lead vehicle area ratio este 0.041, sub pragul de frânare (0.08), confirmând distanță sigură.


## *5.3.2 Scenariul RAINY - Adaptare la Vreme*

**Figura 7: Adaptare la Ploaie**


În condiții de ploaie, sistemul reduce automat viteza recomandată la 38 km/h (75% din viteza de bază). Această ajustare îmbunătățește siguranța pe suprafețe alunecoase.

### *5.3.3 Scenariul BRAKING - Vehicul Prea Aproape*

**Figura 8: Frânare - Vehicul Aproape**


Când lead vehicle area ratio atinge 0.109 (între 0.08 și 0.15), sistemul inițiază frânare și reduce viteza recomandată la 22 km/h. Acțiunea este marcată cu culoare portocalie pentru a indica urgența medie.

**Figura 9: Frânare - Sunny**

### *5.3.4 Scenariul STOPPED - Oprire Completă*

**Figura 10: Oprire Totală - Vehicul Foarte Aproape**


Când lead vehicle area ratio depășește 0.15 (în acest caz 0.239), sistemul oprește complet vehiculul (0 km/h) și afișează STOPPED - VEHICLE AHEAD cu roșu, indicând urgență maximă. Această reacție previne coliziunile în situații critice.

# 6. CONCLUZII ȘI DEZVOLTĂRI VIITOARE

## 6.1 Concluzii

Proiectul a demonstrat cu succes implementarea unui sistem autonom de navigație bazat pe viziune computerizată și învățare automată. Principalele realizări includ:

- Dezvoltarea unui sistem de segmentare semantică cu 90.71% pixel accuracy pe dataset CARLA
- Implementarea unui modul de detecție YOLO pentru recunoașterea obiectelor rutiere
- Crearea unui sistem de decizie ierarhic care prioritizează siguranța
- Adaptarea comportamentului la condițiile meteo și proximitatea vehiculelor
- Optimizarea pentru rulare pe CPU fără GPU (antrenare în 20-30 minute)

Sistemul răspunde corect la diverse scenarii: cruising normal, adaptare la vreme, frânare preventivă și oprire de urgență. Metrica Lead Vehicle Area Ratio s-a dovedit un proxy eficient pentru estimarea distanței fără senzori dedicați.

## 6.2 Limitări Identificate

- IoU scăzut pentru obiecte mici (Traffic Sign, Pole) din cauza rezoluției reduse (96×192)
- Doar 2 epoci de antrenare - performanța ar putea fi îmbunătățită cu mai multe epoci
- Estimarea distanței bazată pe aria bounding box-ului nu este robustă la schimbări de perspectivă
- Sistemul nu integrează date de la senzori fizici (radar, lidar, GPS)

## 6.3 Dezvoltări Viitoare

Pentru îmbunătățirea sistemului, se propun următoarele direcții:

- Antrenare cu rezoluție mai mare (256×512) pe GPU pentru îmbunătățirea IoU la obiecte mici
- Extinderea la 10-20 epoci pentru convergență completă
- Implementarea unui modul de estimare 3D a distanței folosind depth estimation
- Integrare cu planificare de traiectorie pentru manevre complexe (schimbare bandă, intersecții)
- Testare în simulatorul CARLA complet pentru validare în scenarii dinamice
- Adăugarea unui modul de predicție a traiectoriei pentru vehicule și pietoni
- Utilizarea unor arhitecturi mai avansate (DeepLabV3+, HRNet) pentru segmentare

## *6.4 Impactul și Aplicabilitatea*

Acest proiect demonstrează viabilitatea sistemelor de conducere autonomă bazate pe viziune computerizată, chiar și cu resurse computaționale limitate (CPU). Abordarea poate fi extinsă pentru sisteme ADAS reale, unde asistența șoferului este crucială pentru siguranță. Rezultatele confirmă că învățarea automată poate oferi percepție robustă a mediului rutier, esențială pentru viitorul mobilității autonome.