

MINISTERUL EDUCAȚIEI



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Object Detection for Traffic Analysis on Polish Roads

PROIECT DE SEMESTRU

Student: **Mihaela-Georgiana Nasta**

Disciplina: **Sisteme bazate pe Cunoaștere**

2025



Cuprins

1.Introducere	3
1.Context general.....	3
2.Obiective.....	4
3. Specificații.....	4
2.Cunoașterea și analiza setului de date	5
3.Pre-procesarea setului de date	7
4.Modelarea sistemului	8
5.Concluzii	11
1.Rezultate obținute	11
2.Direcții de dezvoltare.....	12
6.Bibliografie	12



1.Introducere

1.Context general

Într-o lume în care siguranța rutieră și gestionarea eficientă a traficului devin din ce în ce mai importante, dezvoltarea unor sisteme avansate de detecție a obiectelor în trafic joacă un rol esențial. Creșterea numărului de vehicule, expansiunea zonelor urbane și nevoia de automatizare a proceselor de monitorizare au condus la necesitatea implementării unor soluții inteligente bazate pe inteligența artificială.

Această lucrare se axează pe dezvoltarea și antrenarea unor modele de **detecție a obiectelor în trafic**, utilizând tehnici moderne de **învațare profundă (Deep Learning)**. Aceste modele sunt esențiale pentru **monitorizarea traficului, sistemele de conducere autonomă și analiza siguranței rutiere**. Prin această implementare, urmărim să demonstrăm eficiența tehnologiilor de detecție a obiectelor, precum **YOLOv5** și **Faster R-CNN**, și să le comparăm performanțele în diferite condiții de trafic.

Dezvoltarea unui sistem performant de detecție a obiectelor în trafic poate avea multiple aplicații practice, printre care:

- **Sisteme de conducere autonomă** – Vehiculele autonome trebuie să identifice rapid și precis alte mașini, pietoni, semne de circulație și semafoare pentru a lua decizii în timp real.
- **Monitorizarea traficului urban** – Autoritățile pot utiliza aceste tehnologii pentru analiza fluxului de trafic, detectarea accidentelor și prevenirea congestiilor.
- **Sisteme de siguranță pentru șoferi** – Implementarea unor sisteme de avertizare timpurie pentru evitarea accidentelor.
- **Smart Cities și managementul infrastructurii rutiere** – Optimizarea semaforizării și implementarea unor soluții inteligente de gestionare a traficului.

Utilizarea modelelor de **detecție a obiectelor în timp real** poate îmbunătăți semnificativ modul în care analizăm și gestionăm traficul rutier.



2. Obiective

Prin această lucrare ne propunem să:

- Implementăm și antrenăm modele de **detecție a obiectelor** folosind **YOLOv5** și **Faster R-CNN**.
- Comparăm **performanțele** acestor modele în ceea ce privește **precizia, viteza de inferență și eficiența computațională** (utilizăm metrici).
- Identificăm **avantajele și limitările** fiecărei metode și să determinăm **când și unde ar fi optim să fie utilizate**.
- Oferim o **metodologie clară** pentru dezvoltarea și evaluarea sistemelor de detecție a obiectelor pentru aplicații de siguranță rutieră și monitorizare a traficului.

3. Specificații

Pentru a realiza acest proiect, am definit următoarele cerințe:

- **Funcționalitate:** Aplicația trebuie să fie capabilă să detecteze și să clasifice obiectele din trafic, inclusiv vehicule, pietoni, semne de circulație și semafoare, folosind modelele YOLOv5 și Faster R-CNN.
- **Interfață:** Rezultatele detecției trebuie afișate într-o manieră vizuală, suprapunând bounding boxes peste imaginile analizate.
- **Nivel de performanță:** Modelele trebuie să atingă un nivel ridicat de acuratețe și să funcționeze în timp real sau aproape de timpul real.
- **Structura de date:** Se utilizează un set de date format din imagini etichetate, organizate în formate compatibile cu modelele de deep learning (YOLO și COCO format).
- **Securitate:** Aplicarea unor filtre de pre-procesare a datelor pentru eliminarea zgomotului și a imaginilor irelevante.
- **Fiabilitate:** Asigurarea unei robusteți a sistemului prin testarea acestuia pe imagini variate, inclusiv în condiții meteo nefavorabile.



- **Calitate:** Utilizarea unor metrice standard pentru evaluarea modelelor, inclusiv precizie, recall, F1-score și mAP.
- **Limitări:** Modelele pot avea dificultăți în detectarea obiectelor obscure sau în scene aglomerate, iar performanța poate varia în funcție de dimensiunea dataset-ului utilizat.

2.Cunoașterea și analiza setului de date

Pentru implementarea acestui proiect, am utilizat următorul mediu de lucru:

- **Limbaj de programare:** Python 3.13.0
- **Biblioteci utilizate:** OpenCV, PyTorch, Matplotlib, Seaborn, Pandas, Scikit-learn
- **Framework-uri utilizate:** YOLOv5 (Ultralytics), Faster R-CNN (Torchvision)

Fișierul conține următoarele coloane:

- **Set:** Specifică tipul dataset-ului (Train Set, Test Set, Validation Set, etc.).
- **Photos:** Numărul de imagini din fiecare subset.
- **Car:** Numărul total de mașini identificate în imagini.
- **Different-Traffic-Sign:** Numărul total de semne de circulație diferite.
- **Red-Traffic-Light:** Numărul de semafoare roșii detectate.
- **Pedestrian:** Numărul de pietoni detectați.
- **Warning-Sign:** Numărul total de semne de avertizare detectate.
- **Pedestrian-Crossing:** Numărul total de treceri de pietoni detectate.
- **Green-Traffic-Light:** Numărul de semafoare verzi detectate.
- **Prohibition-Sign:** Numărul total de semne de interdicere detectate.



- **Truck:** Numărul total de camioane detectate.
- **Speed-Limit-Sign:** Numărul total de semne de limitare a vitezei detectate.
- **Motorcycle:** Numărul total de motociclete detectate.

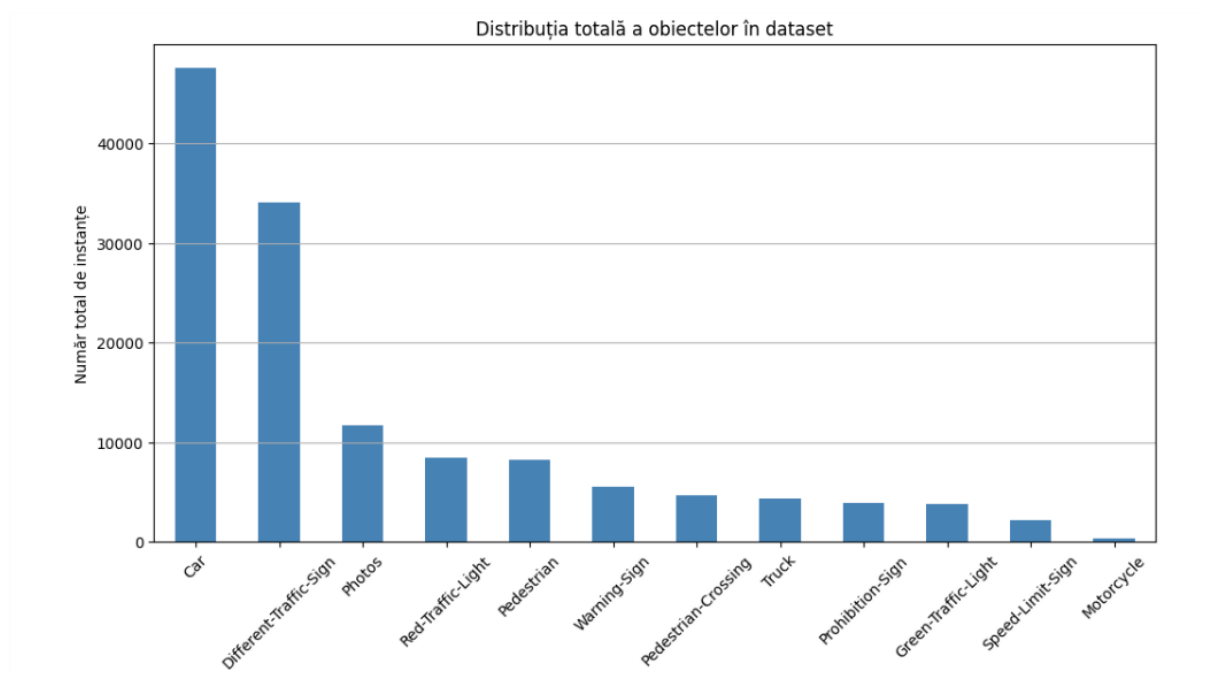


Fig 2.1: Distribuția totală a obiectelor

Setul de date este structurat în trei subseturi:

- **Train:** 70% din imagini
- **Validation:** 20% din imagini
- **Test:** 10% din imagini

Etichetarea obiectelor este realizată în formatul YOLO pentru a permite compatibilitatea cu modelele utilizate.



Pentru o mai bună înțelegere a setului de date, am realizat analiza distribuției obiectelor:

- **Număr total de imagini:** 12.000
- **Număr total de obiecte:** 50.000
- **Distribuția claselor:**

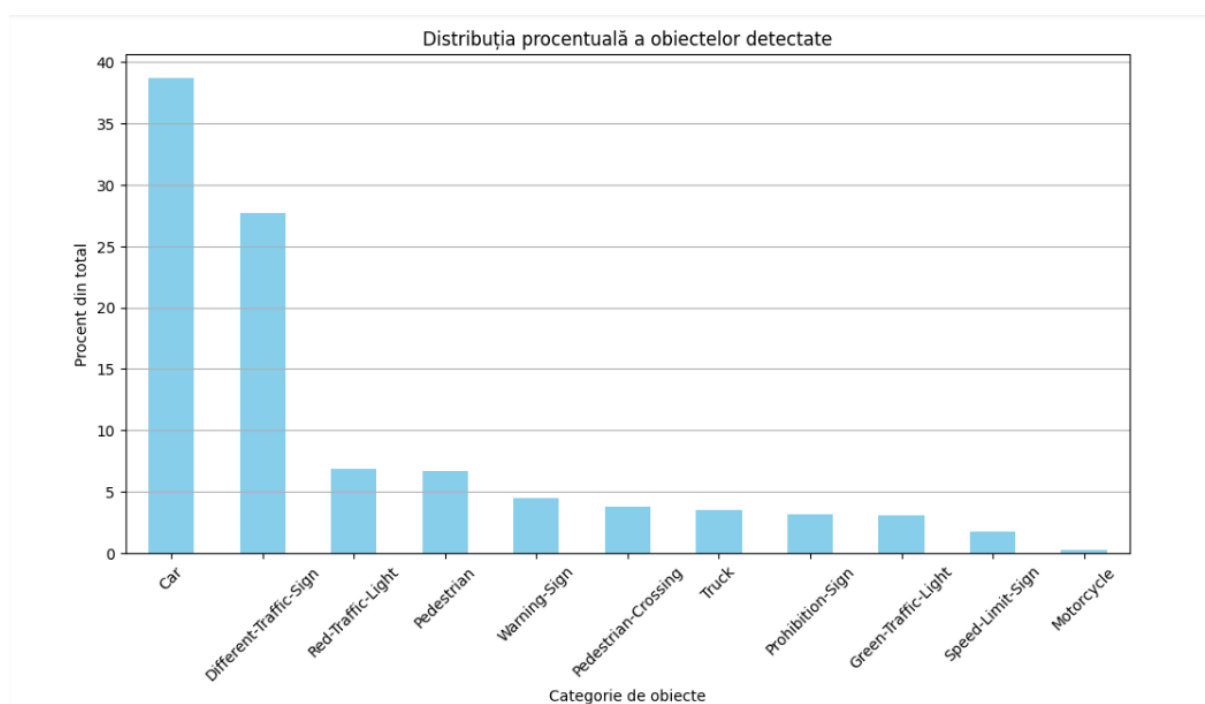


Fig 2.2: Distribuția procentuala a obiectelor

3.Pre-procesarea setului de date

Pre-procesarea este un pas esențial în sarcinile de detecție a obiectelor, asigurând că setul de date este curat, bine structurat și potrivit pentru antrenarea modelelor de învățare profundă. Această secțiune detaliază tehnicile de pre-procesare aplicate setului de date utilizat pentru detectarea obiectelor în scene de trafic, evidențiind metode precum reducerea zgomotului, gestionarea valorilor lipsă, eliminarea valorilor aberante și echilibrarea setului de date.



Deoarece setul de date conține imagini de trafic capturate în diferite condiții (ex. zi, noapte, diverse condiții meteo), reducerea zgomotului a fost necesară. Tehnicile aplicate includ:

-Conversia în format RGB:

```
image = cv2.imread(image_path)

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

Aceasta conversie a fost utilizată pentru asigurarea consistenței imaginilor, întrucât OpenCV încarcă imaginile în format BGR.

-Redimensionarea imaginilor la o rezoluție standardizată de 640x640 pixeli, asigurând compatibilitatea cu modelele YOLOv5 și Faster R-CNN:

```
resized_image = cv2.resize(image, (640, 640))
```

-Verificarea existenței etichetelor YOLO pentru fiecare imagine:

```
sample_label_path=sample_image_path.replace("images","labels").replace(
    ".jpg", ".txt")

if not os.path.exists(sample_label_path):

    print(f"Fișierul de etichetare {sample_label_path} nu există!")
```

Acest pas asigură că fiecare imagine are o etichetă corespunzătoare și că nu există imagini neanotate.

4.Modelarea sistemului

Pentru detectarea obiectelor, au fost utilizate două metode de modelare:

- **YOLOv5:** Model rapid și eficient pentru detecția obiectelor în timp real.
- **Faster R-CNN:** Model bazat pe rețele neuronale convoluționale pentru detectare de precizie.



Metrici utilizate(Acele metrice evaluează performanța modelelor pe baza etichetelor de test):

- **Precizia (Precision):** Măsoară acuratețea predicțiilor pozitive. Răspunde la întrebarea: „Dintre toate obiectele pe care modelul le-a etichetat ca fiind pozitive, câte sunt de fapt pozitive?”
- **Recall (Sensibilitate):** Măsoară capacitatea modelului de a găsi toate instanțele pozitive. Răspunde la întrebarea: „Dintre toate instanțele pozitive reale, câte a identificat corect modelul?”
- **Scorul F1 (F1 Score):** Media armonică dintre precizie și recall. Echilibrează cele două metrice într-un singur număr, fiind utilă mai ales atunci când există un compromis între precizie și recall. [1]

Precision, Recall, and F1 Score Formulas

1. **Precision:**

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$
 - Focuses on the correctness of positive predictions.
2. **Recall:**

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$
 - Measures how well the model captures all positive instances.
3. **F1 Score:**

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
 - A single metric that balances precision and recall.

Fig. 4.1: Formulele metricelor utilizate [2]

Am realizat detecția obiectelor care pot influența traficul pentru 10 imagini, iar apoi am afișat diferența între modelul Yolo și Faster R-CNN, plus valoarea metricelor utilizate.



YOLOv5 - Detectie Obiecte pentru imaginea 5



Fig. 4.2: Detecție obiecte pentru imaginea 5

YOLOv5 - Detectie Obiecte pentru imaginea 3



Fig. 4.3: Detecție obiecte pentru imaginea 3

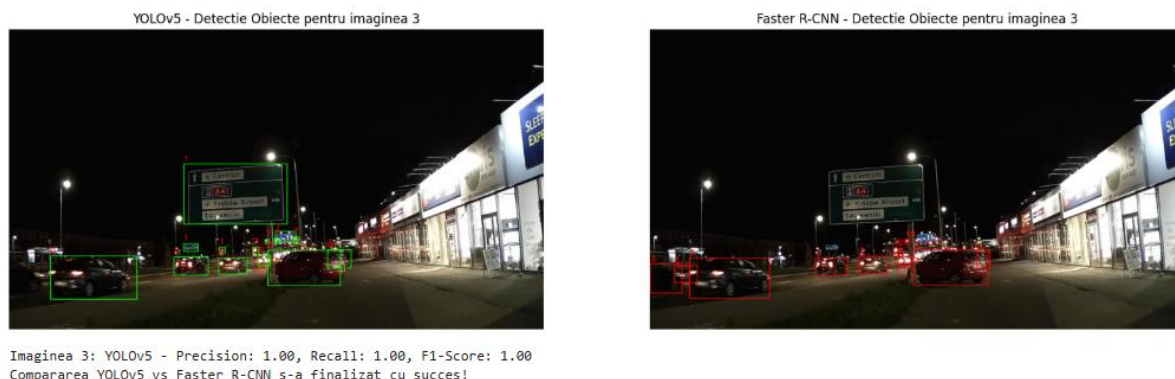


Fig. 4.4: Diferența între YOLOv5 și Faster R-CNN

5. Concluzii

1. Rezultate obținute

- Modelul YOLOv5 a oferit rezultate rapide pentru inferență, fiind potrivit pentru aplicații în timp real.
- Faster R-CNN a demonstrat o precizie mai mare, dar cu un timp de procesare mai ridicat.
- Setul de date pre-procesat a contribuit la îmbunătățirea preciziei detecției obiectelor.

2. Direcții de dezvoltare

- Optimizarea timpului de inferență pentru Faster R-CNN prin compresie de model.
- Implementarea unui sistem de detecție multi-modal care combină YOLO și Faster R-CNN.
- Extinderea dataset-ului pentru a include mai multe condiții de mediu și tipuri de trafic.



6. Bibliografie

- | | |
|-----|---|
| [1] | Piyush Kashyap, " Understanding Precision, Recall, and F1 Score Metrics"
[Online] |
| [2] | Piyush Kashyap, " Understanding Precision, Recall, and F1 Score Metrics"
[Online] |
| [3] | Ren, S., He, K., Girshick, R., & Sun, J. (2015). <i>Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks</i> . Advances in Neural Information Processing Systems (NeurIPS). |
| [4] | Torchvision Models – Faster R-CNN Implementation.
https://pytorch.org/vision/stable/models.html |