

# PROIECTAREA ȘI IMPLEMENTAREA UNUI SISTEM DE AVERTIZARE ANTICOLIZIUNE BAZAT PE INTELIGENȚĂ ARTIFICIALĂ

**Autori:** Cătălin<sup>1</sup> CUCUTIANU, Emil<sup>1</sup> LĂZĂRESCU  
catalin.cucutianu@gmail.com      emil\_lazarescu@yahoo.ro

**Coordonatori:** Conf.univ.dr.ing. **Olimpiu STOICUȚA**<sup>2</sup>, Șef lucr.dr.ing. **Cosmin RUS**<sup>2</sup>, Șef lucr.dr.ing. **Marius RÎSTEIU**<sup>2</sup>

<sup>1</sup>Universitatea din Petroșani, Facultatea I.M.E., Master specializarea: TTIA anul 1

<sup>2</sup>Universitatea din Petroșani, Facultatea I.M.E., Departamentul:A.C.I.E.E.

## Rezumat:

În această lucrare ne-am propus sa proiectam un sistem de avertizare anticoliziune folosind fluxul video de la orice camera de video. Pentru aceasta ne vom folosi de librăria de inteligență artificială de ultimă generație YOLOV8, vom crea un set de date pentru a antrena modelul de AI să recunoască într-un mod precis un reper pe care îl vom alege, după care vom integra acest model într-o interfață grafică prin care putem seta parametrii detecției și vom fi avertizați în cazul pericolului de coliziune. Un aspect interesant al acestei lucrări îl va reprezenta abordarea neconvențională în privința modului de construire a setului de date și a reperului și modului de procesare a informației generate de AI pentru a identifica pericolul de coliziune iminentă.

## Cuvinte cheie:

AI, automotive, safety, YOLOV8, realtime

### 1. Introducere

În lumea de astăzi, siguranța rutieră a devenit o problemă de importanță maximă. Odată cu creșterea numărului de vehicule de pe drumuri, riscul de coliziuni și accidente a crescut semnificativ. Cu toate acestea, progresele în tehnologie au deschis calea pentru soluții inovatoare pentru a aborda această problemă. Un astfel de sistem revoluționar este Sistemul Anticoliziune bazat pe analiza imaginilor video cu ajutorul Inteligenței Artificiale. Această tehnologie de ultimă generație combină inteligența artificială și algoritmi de viziune computerizată pentru a detecta și preveni potențialele coliziuni pe drumuri. Prin analiza datelor în timp real de la camere, acest sistem poate identifica cu precizie vehiculele oferind o abordare proactivă pentru evitarea accidentelor. În acest articol, vom explora principiile de funcționare, avantajele și perspectivele viitoare ale acestui sistem anticolidiție revoluționar, evidențiind potențialul său de a transforma siguranța rutieră și de a salva numeroase vieți.

Un alt aspect demn de luat în considerare este și ușurința cu care un sistem poate fi integrat, nu doar în proiectele noi, ci și în autoturismele care există deja. Din acest motiv am decis să creăm un sistem plug and play care poate fi conectat la orice camera video care are semnal de ieșire, cum ar fi camerele de supraveghere auto, versiunile mai noi de GoPro sau chiar telefoanele mobile în combinație cu anumite softuri.

### 2. Alegerea subsistemelor

Deoarece ne-am propus nu doar să facem un concept fictiv, ci să și implementăm un “proof of concept”, am ales să folosim doar ce avem la îndemână: un laptop cu accelerator grafic capabil cu nuclee CUDA, camera web a laptop-ului și un telefon pentru a capta un video de test. Având aceste componente fizice, alegerea componentelor software a fost următoarea:

1. Librăria YOLOV8: antrenare și inferență a modelului de AI
2. Aplicația LabelMe: etichetarea imaginilor pentru a putea crea setul de date necesar antrenării AI-lui
3. Utilitarul labelme2yolo: conversia imaginilor etichetate într-un format recunoscut de YOLOV8
4. Anaconda Navigator: gestionarea pachetelor de Python în medii independente
5. Mediile de dezvoltare VS Code și Spyder (inclusiv ca și pachet în Anaconda Navigator)
6. Librăria de Python Custom Tkinter: generarea fără efort a unei interfețe relativ moderne

### 3. Arhitectura sistemului

Ca și limbaj de programare am ales Python datorită faptului că este ușor de înțeles și nu necesită compilare pentru a testa modificările din program.

Arhitectura aplicației este una simplă dar extensibilă, separând aplicația principală de interfață, captura și procesarea imaginilor, cat și de logica de urmărire a obiectelor. Astfel, aplicația principală încarcă macheta interfeței grafice, apoi asociază fiecare element al interfeței cu funcția potrivită din celelalte module.

Fluxul de imagini este captat de către camera video sau poate fi încărcat dintr-un fișier, apoi este procesat în modulul de manipulare de imagini, care poate fi configurat în timp real de către aplicația principală prin comenzi primite prin intermediul modulului de interfață grafică. Apoi acest flux grafic este trimis către modelul de inteligență artificială (YOLOV8) care realizează detecția propriu-zisă, iar aceste informații sunt extrase și trimise către modulul de selector de obiecte, care decide ce obiect să fie în focus și calculează rata de apropiere pe baza variației în timp a suprafeței detectate. Informația astfel obținută este folosită de către modulul principal pentru a avertiza utilizatorul în cazul unui pericol de coliziune iminentă. În această lucrare avertizarea este sub formă vizuală dar este extrem de simplu de adaptat, în cazul unei eventuale dezvoltări, fiind posibil ca avertizarea să fie și auditivă, sau s-ar putea conecta chiar la sistemul de frânare a autovehiculului

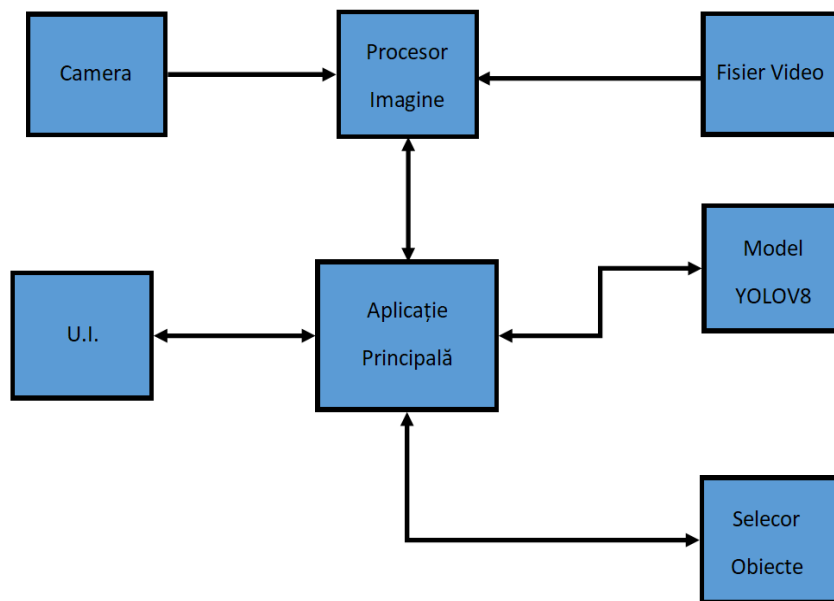


Fig. 1. Arhitectura aplicației

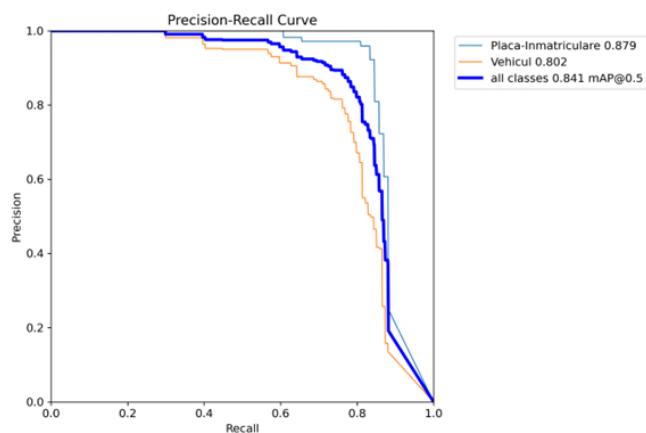
### 4. YOLOV8: creierul la baza sistemului

YOLOV8 reprezintă un model de inteligență artificială conceput să fie rapid, robust și flexibil. De asemenea este ușor de folosit, acest lucru făcându-l potrivit pentru o gamă largă de aplicații. Printre capacitățile acestuia se numără: detecția și urmărirea obiectelor, segmentarea imaginilor, clasificarea obiectelor și estimarea posturii.

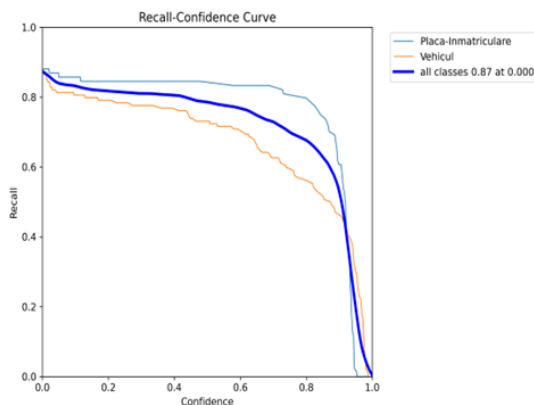
Deși există o serie de modele pregătite pentru a fi folosite, acestea sunt de uz general și nu sunt potrivite pentru aplicația noastră. Pentru aceasta vom avea nevoie să antrenăm un model nou specializat nevoilor noastre, și anume să detecteze cu precizie foarte mare plăcuțele de înmatriculare.

#### 4.1 Pregătirea setului de date și antrenarea modelului de inteligență artificială

Pentru a construi setul de date necesar antrenării modelului am folosit setul “Car License Plate Detection” disponibil sub licența publică pe site-ul kagle.com. Acesta constă dintr-o serie de autoturisme în care sunt adnotate autoturismele și plăcuțele de înmatriculare. În urma experimentelor inițiale cu acest set nemodificat am obținut rezultate nesatisfăcătoare, după cum se observă în fig. 2 și mai ales fig. 3, unde observăm un grad mare de detecții ratate chiar și la nivele de siguranță (confidence) sub 50%.

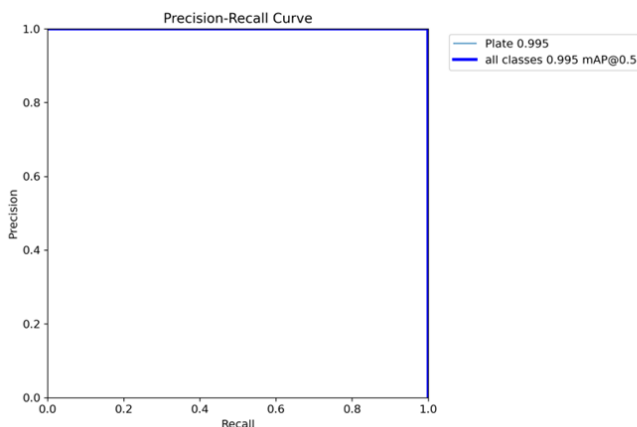


**Fig. 2.** Curba precision-recall

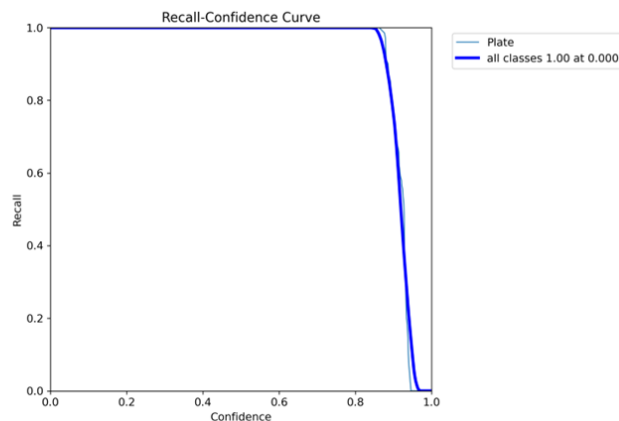


**Fig. 3.** Curba recall-confidence

După aceea am folosit o abordare complet neconvențională, și anume am scos la imprimantă un subset de imagini, apoi am fotografiat paginile cu ajutorul camerei web de la diferite distanțe, am aplicat un filtru CLAHE (Contrast Limited Adaptive Histogram Equalization) după care am adnotat setul de date folosind aplicația LabelMe, obținând rezultate superioare:



**Fig. 4.** Curba precision-recall



**Fig. 5.** Curba recall-confidence

Deși setul de date rezulat pare mai uniform, rezultatele observate sunt excelente, și după cum se va observa și în testele cu video-uri independente, gradul de recunoaștere și stabilitatea detecției sunt aproape perfecte. Totuși, în cadrul aplicației noastre cel mai important parametru este recall - procentul de repere existente care sunt detectate, precizia (gradul de corectitudine a detecției) fiind mai puțin importantă deoarece algoritmul nostru trebuie sa detecteze reperul principal cu orice preț, iar detecțiile accidentale sunt filtrate.

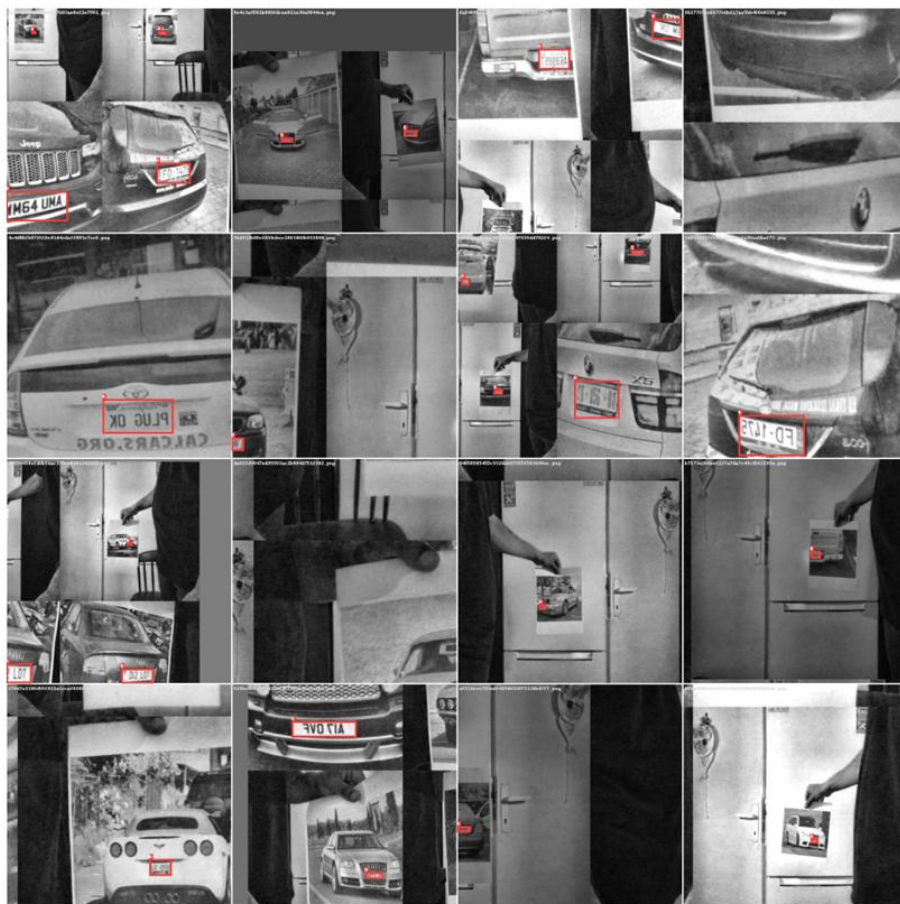
#### 4.2 Detalii despre procesul de antrenare a rețelei neuronale

Odată construit setul de date, procesul de antrenare este extrem de simplu, doar executând următoarea linie de comandă:

```
yolo task=detect mode=train model=yolov8m.pt imgsz=640 data=C:\Users\cct8blj\z\datasets\carplate_norm.yaml epochs=500 batch=16 deterministic=false cache=True overlap_mask=False name=yolov8n_normplate
```

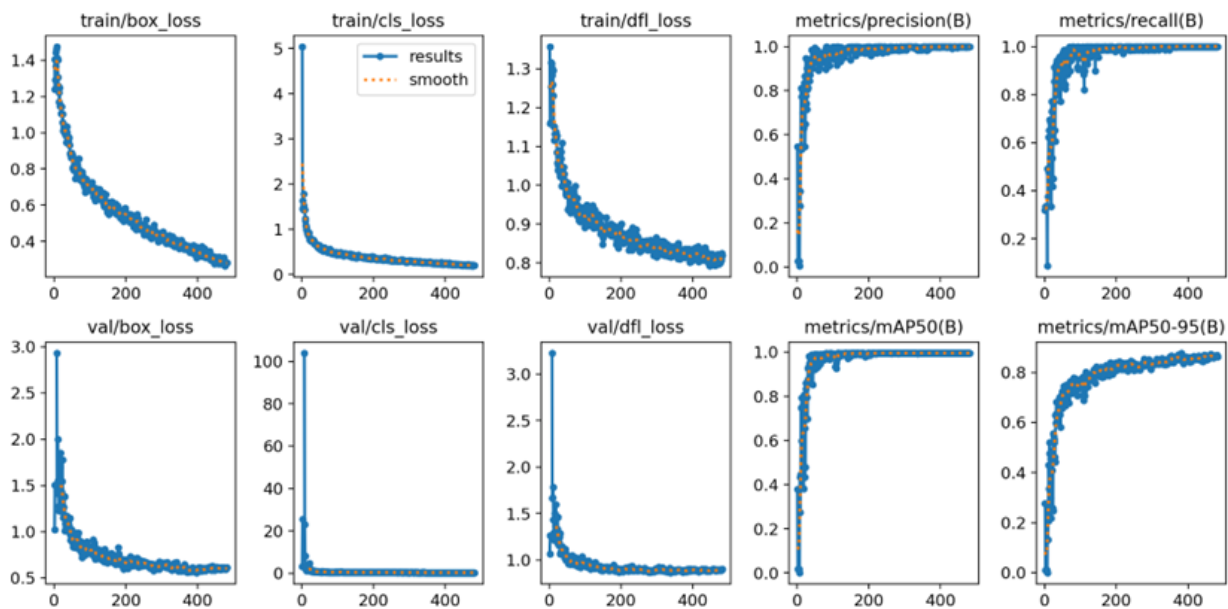
Prin aceasta am specificat task-ul detect (detectie dreptunghi de încadrare), modul train (mod antrenare), am ales modelul general yolov8m.pt ca și punct de pornire, dimensiunea imaginilor de 640x640, locația unde se află fișierul care descrie setul de date, numărul de epoci (tregeri complete prin setul de date), precum și alți parametri auxiliari.

Imaginile care sunt trimise spre procesul de antrenare arată în felul următor:



**Fig. 6.** Lot (batch) de imagini folosite pentru antrenare

Se observă atât monul neconvențional de obținere a imaginilor cât și faptul că algoritmul de antrenare YOLOV8 manipulează în mod automat imaginile transformându-le prin mozaicare, decalare în spațiu și modificare a luminozității / contrastului pentru a genera un model rezistent variații de poziție și de condiții de iluminare.



**Fig.7.** Rezultatele unei sesiuni de antrenare

Din fig.7 deducem că după 500 de epoci de antrenare, deși precizia este excelentă, încă mai poate fi îmbunătățită prin creșterea numărului de iterații.

### 5. Interfața aplicației

În fig.8 este prezentată aplicația de detecție cu setările implicite în timp ce este simulat un test real folosind o filmare realizată din interiorul autovehiculului, iar în fig. 9 se observă modul în care utilizatorul este avertizat în cazul unui pericol de coliziune iminentă. Opțiunile din categoria Capture Zone influențează suprafața din imaginea completă care este decupată, iar cele din categoria Capture Mode influențează modul în care sunt preprocesate culorile și luminozitatea, după cum se observă în fig.9 unde este activat modul CLAHE Mono: Contrast Limited Adaptive Histogram Equalisation converted to Monochrome, acest mod fiind foarte util în condiții de iluminare slabă îmbunătățind extrem de mult nivelul de detalii și contrast local.

Setările Exposure +/- și ISO +/- sunt utile în cazul folosirii unei camere live pentru ajustarea timpului de expunere și a sensibilității.

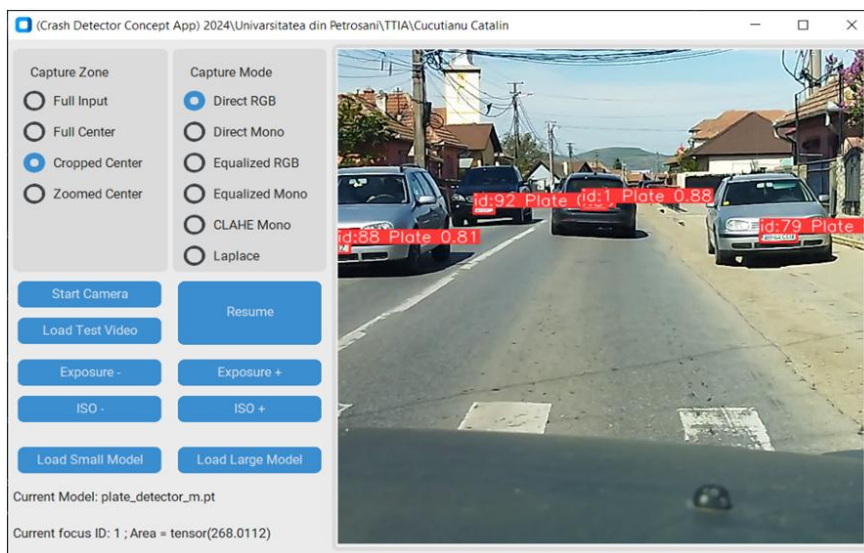


Fig. 8. Interfața aplicației

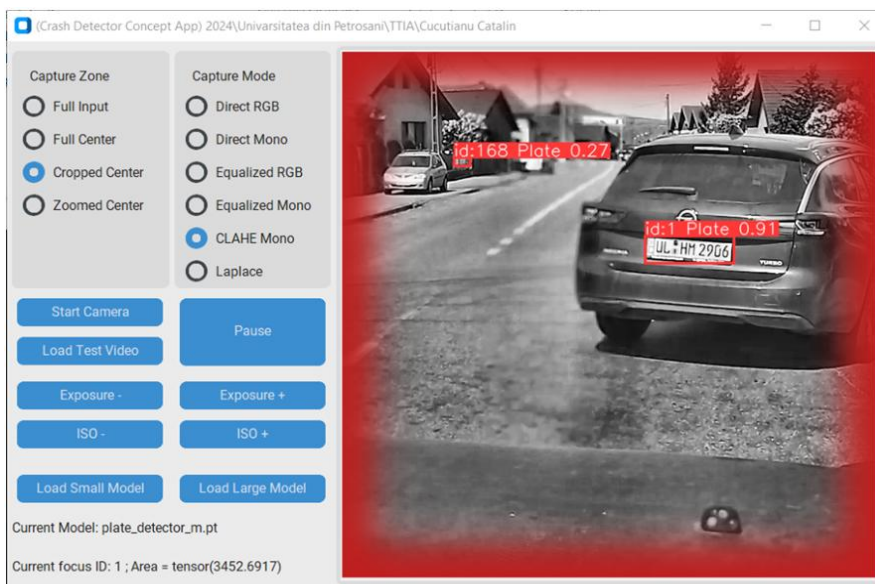


Fig.9. Avertisment

## 6. Concluzii

Domeniul siguranței auto este unul de maximă importanță și nici o ramură a acestuia nu poate fi subestimată. În acest context am proiectat și implementat cu succes un sistem de detectare timpurie a coliziunilor bazat pe computer vision capabil să funcționeze în timp real. Scopurile îndeplinite în aceasta lucrare sunt:

- Construirea unui set de date pentru detecția variației distanței față de un automobil bazat pe informații din domeniul public.

- Antrenarea unei rețele neuronale capabile să detecteze cu precizie și siguranță extrem de mare un reper care există pe toate autovehiculele: plăcuța de înmatriculare.

- Încapsularea rețelei neuronale într-o aplicație modulară și extensibilă.

În acest demers am observat importanța majoră a calității datelor folosite pentru antrenarea unui AI, în principal consistența adnotărilor și diversitatea scenariilor, aceasta având un rol mai mare decât complexitatea modelului folosit. Ca și relație empirică, cele mai bune rezultate se obțin echilibrând dimensiunea setului de date, a modelului și a numărului de epoci de antrenare, cu mențiunea că modelele mai mari sunt predispuse la fenomenul de “over fitting” dacă nu este furnizat un set de date suficient de mare și variat.

## Referințe

1. Hussain, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines***2023**,*11*, 677. <https://doi.org/10.3390/machines11070677>
2. M. Safaldin, N. Zaghdien and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," in *IEEE Access*, doi: 10.1109/ACCESS.2024.3393835. keywords: {YOLO;Feature extraction;Real-time systems;Detectors;Computer architecture;Object recognition;Task analysis;Deep learning;Deep learning;localization;object detection;segmentation;Yolo},
3. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics***2023**,*12*, 2323. <https://doi.org/10.3390/electronics12102323>
4. Wang, X.; Gao, H.; Jia, Z.; Li, Z. BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. *Sensors***2023**,*23*, 8361. <https://doi.org/10.3390/s23208361>
5. [License Plates Dataset \(roboflow.com\)](https://roboflow.com) - set de date sub licență CC BY 4.0