

Ministerul Educației și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

**RAPORT**

**Lucrarea de laborator nr.6**

**la Inteligența Artificială**

*Tema: Sisteme video inteligente de supraveghere și control bazate pe recunoaștere automată a formelor și imaginilor.*

Grupa academică: TI-214  
A efectuat: Reguș Ruslan

**A verificat: Mariana Rusu**

Chișinău 2024

Recunoașterea automată a formelor și imaginilor reprezintă o ramură importantă a inteligenței artificiale, cu aplicații variate în domenii precum supravegherea video, asistența medicală, vehicule autonome, recunoașterea facială și multe altele. Această tehnologie implică dezvoltarea algoritmilor și a sistemelor capabile să identifice și să clasifice obiecte sau forme în imagini digitale sau în fluxuri video.

**Metode Generale de Recunoaștere și Clasificare a Imaginilor:**

1. **Extracția de Caracteristici (Feature Extraction):** Această etapă implică transformarea imaginii într-o reprezentare numerică care poate fi prelucrată mai departe. Caracteristicile pot fi extrase folosind diverse tehnici, cum ar fi:

* **HOG (Histogram of Oriented Gradients):** Măsoară distribuția gradientului direcțiilor într-o imagine, furnizând informații despre textură și formă.
* **SIFT (Scale-Invariant Feature Transform):** Detectează și descrie caracteristici locale invariante la rotație, scalare și iluminare.
* **CNN (Convolutional Neural Networks):** O abordare bazată pe învățarea profundă care extrage automat caracteristici relevante din imagini.

1. **Clasificarea Imaginilor:**

* **Mașini cu Vectori Suport (SVM):** O metodă de învățare supervizată utilizată pentru clasificare, care găsește un hiperplan optim pentru a separa clasele în spațiul caracteristic.
* **Rețele Neuronale Convoluționale (CNN):** O abordare puternică bazată pe învățarea profundă, care este capabilă să clasifice imagini cu performanțe deosebite.

1. **Post-procesare:**

* **Filtrare și Reducerea Zgomotului:** Îmbunătățește precizia recunoașterii prin eliminarea sau reducerea informațiilor irelevante din imagine.
* **Fuziunea Datelor:** Integrarea mai multor surse de informații pentru a obține rezultate mai precise și robuste.

**Analiza Algoritmilor de Recunoaștere Automată a Formelor și Imaginilor depind de careva factori:**

1. Eficiență și Performanță;
2. Rezistență la Variabilitate;
3. Generalizare;
4. Interpretabilitate;
5. Etica și Confidențialitate.

from collections import defaultdict

import cv2

import numpy as np

from ultralytics import YOLO

# Încărcarea modelului preantrenat

model = YOLO('yolov8n.pt')

# Definirea căii către fișierul video

video\_path = "C:/Users/schio/OneDrive/Desktop/Anul 3/sem 2/IA/lab 6/test2.mp4"

cap = cv2.VideoCapture(video\_path)

# Inițializarea unui defaultdict pentru stocarea istoricului urmăriii

track\_history = defaultdict(lambda: [])

# Inițializarea buclei de prelucrare a videoului

while cap.isOpened():

# Citirea frame-ului din video

success, frame = cap.read()

# Dacă frame-ul a fost citit cu succes

if success:

# Rularea modelului într-un mod persistent între cadre

results = model.track(frame, persist=True)

# Obținerea conturilor obiectelor și id-urile acestora

boxes = results[0].boxes.xywh.cpu()

track\_ids = results[0].boxes.id.int().cpu().tolist()

# Vizualizarea rezultatului prelucrării în formă de cadru

annotated\_frame = results[0].plot()

# Trasarea traiectoriilor de mișcare a obiectelor

for box, track\_id in zip(boxes, track\_ids):

# Extragerea coordonatelor și dimensiunilor box-ului

x, y, w, h = box

# Obținerea istoricul urmăririi

track = track\_history[track\_id]

# Adăugarea coordonatelor centrului boxului

track.append((float(x), float(y)))

# Dacă lungimea căii de urmărire este mai mare decât 30

if len(track) > 30:

# Se șterg cele mai vechi valori

track.pop(0)

# Trasarea liniilor de urmărire

points = np.hstack(track).astype(np.int32).reshape((-1, 1, 2))

cv2.polylines(annotated\_frame, [points], isClosed=False, color=(75, 245, 2), thickness=4, lineType= cv2.LINE\_AA)

# Afișarea frame-ului

cv2.imshow("YOLOv8 Tracking", annotated\_frame)

# Redimensionarea ferestrei

cv2.resizeWindow('YOLOv8 Tracking', 1920, 1080)

# Închiderea ferestrei dacă tasta q este apăsată

if cv2.waitKey(1) & 0xFF == ord("q"):

break

else:

# Oprirea buclei dacă s-a ajuns la sfârșitul video-ului

break

# Eliberează toate resursele ocupate și încjide fereastra

cap.release()

cv2.destroyAllWindows()

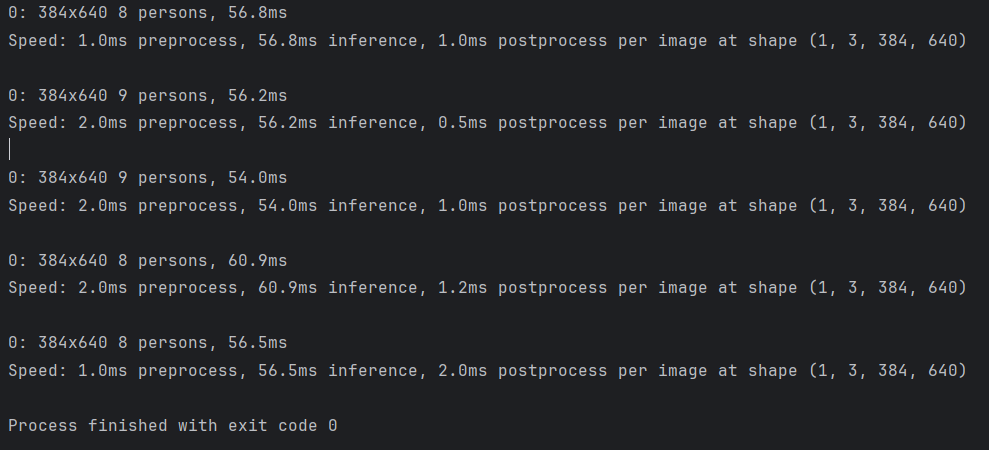
****

Figura 1 – Execuția în console

Pe parcursul rulării codului în consolă putem vedea rezumatul procesării fiecărui cadrul separat. Ni se afișează următoarele informații: dimensiunea obiectului, numărul de obiecte în cadru și denumirea acestora, timpul de procesare, timpul de preprocesare, timpul de inferență (timpul de detectare a obiectelor în cadrul frame-ului), dimensiunea figurii. A fost folosit modelul YOLO preantrenat (You Only Look Once).



Figura 2 – Execuția codului din fereastră

Acesta este doar un frame din cadrul unui video. Se observă că modelul preantrenat determină obiectele și trasează un box împrejurul lor, deasupra căruia se indică precizia că obiectul este determinat corect.

**Acest model poate recunoaște automobilele**

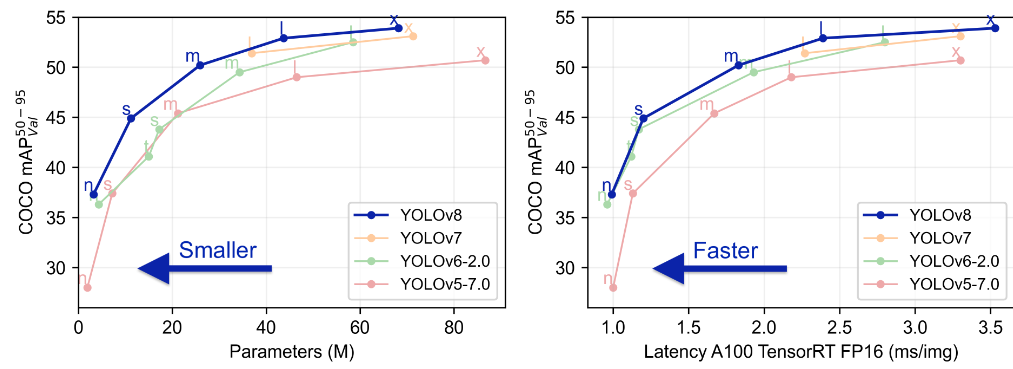


Figura 3 – Exactitatea modelului YOLOv8 în comparație  
cu cele din generațiile mai vechi



Figura 4 – Capacitățile modelului YOLOv8

**Concluzie**

În timpul lucrului la proiectul de laborator, am dezvoltat un sistem de detectare și urmărire a obiectelor într-un videoclip, folosind modelul YOLOv8. Acest sistem operează printr-o buclă de procesare a cadrelor video, identificând obiectele și evidențiindu-le traiectoriile în timpul mișcării. Prin optimizarea funcțiilor de urmărire și de desenare a liniilor, am reușit să realizăm o vizualizare și urmărire a obiectelor în timp real.

Proiectul oferă o demonstrație practică a aplicării modelelor de învățare profundă în domeniul detecției și urmăririi obiectelor în contextul analizei de video. Această abordare nu se limitează doar la această aplicație specifică, ci poate fi extinsă pentru a fi utilizată în diverse domenii, cum ar fi monitorizarea traficului, securitatea sau analiza comportamentului în timp real.