

# OBJECT TRACKING

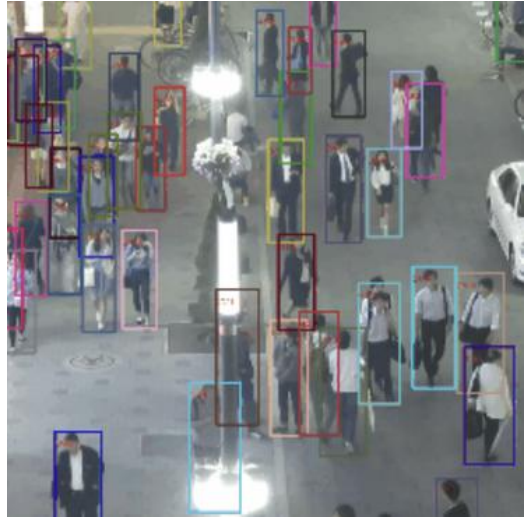
---

Tremurici Rebeca

An 3

## Descriere proiect

Urmărirea obiectelor este un proces de urmărire a unui obiect în cadre video succesive. Utilizăm algoritmi de urmărire pentru că sunt mai rapizi decât algoritmi de detecție. Atunci când urmăriți un obiect care a fost detectat în cadrul anterior, știți multe despre aspectul obiectului. OpenCV este o bibliotecă renumită pentru urmărirea unui obiect. Este o bibliotecă care vizează viziunea computerizată în timp real. Urmărirea și localizarea obiectelor în imaginile digitale a devenit una dintre cele mai importante aplicații pentru industrii. Aceasta ușurează munca utilizatorului, economisește



timp și realizează paralelismul. Aceasta nu este o tehnică nouă, ci o îmbunătățire a tehnicilor existente.

Un exemplu de aplicație este un sistem de supraveghere video și securitate, în care pot fi detectate acțiuni suspecte. Alte exemple sunt monitorizarea traficului pe autostrăzi și, de asemenea, analiza mișcării jucătorilor într-un meci de fotbal! În acest ultim exemplu, este posibil să urmăriți traseul complet pe care l-a urmat jucătorul în timpul meciului.

## Object Tracking vs. Object detection

Urmărirea obiectelor se referă la capacitatea de a estima sau de a prezice poziția unui obiect țintă în fiecare cadru consecutiv al unei înregistrări video, odată ce poziția inițială a obiectului țintă este definită.

Pe de altă parte, detectarea obiectului este procesul de detectare a unui obiect țintă într-o imagine sau într-un singur cadru video. Detectarea obiectelor va funcționa numai dacă imaginea țintă este vizibilă pe baza datelor de intrare. În cazul în care obiectul țintă este ascuns de orice interferență, acesta nu va putea fi detectat.

Urmărirea obiectului este antrenată pentru a urmări traiectoria obiectului în ciuda ocluziilor.

## Tipuri de urmărire a obiectelor

Există două tipuri de urmărire a obiectelor:

- urmărirea imaginilor
- urmărirea video

## **1. Urmărirea imaginilor**

Urmărirea imaginilor este destinată detectării imaginilor bidimensionale de interes într-o intrare dată. Imaginea respectivă este apoi urmărită în mod continuu pe măsură ce se deplasează în decor.

Urmărirea imaginilor este ideală pentru seturile de date cu imagini foarte contrastante (de exemplu, alb-negru), asimetrie, puține modele și multiple diferențe identificabile între imaginea de interes și alte imagini din setul de imagini.

Urmărirea imaginilor se bazează pe viziunea computerizată pentru a detecta și a mări imaginile după ce țintele imaginii sunt predeterminate.

## **2. Urmărirea video**

Urmărirea video este sarcina de a urmări un obiect în mișcare într-un videoclip.

Ideea urmăririi video este de a asocia sau de a stabili o relație între obiectele țintă așa cum apar în fiecare cadru video. Cu alte cuvinte, urmărirea video constă în analiza secvențială a cadrelor video și în corelarea locației anterioare a obiectului cu locația prezentă prin predicție și crearea unei cutii de delimitare în jurul acestuia.

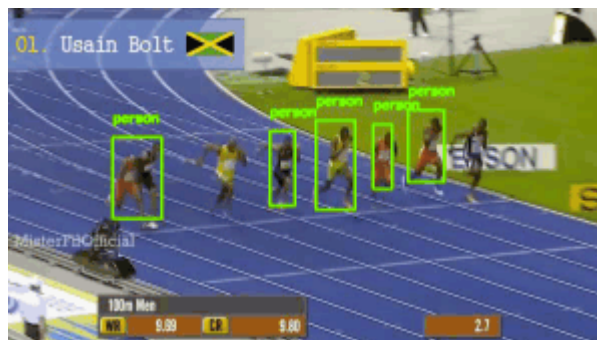
Urmărirea video este utilizată pe scară largă în monitorizarea traficului, în mașinile care se conduc singure și în domeniul securității, deoarece poate procesa imagini în timp real.

În realizarea proiectului am folosit 7 algoritmi (Boosting, MIL, KCF, CSRT, MedianFlow, TLD, MOSSE) pe care i-am folosit în două tipuri de tracking, și anume: single tracking și multiple tracking. Am folosit același videoclip pentru toți algoritmii pentru a vedea care dintre ei sunt mai eficienți în detectarea obiectelor.

În cazul urmăririi unui singur obiect, caseta de delimitare a țintei din primul cadru este furnizată urmăritorului. Scopul urmăritorului este apoi de a localiza aceeași țintă în toate celelalte cadre. SOT aparține categoriei de urmărire fără detecție, deoarece prima cutie delimitatoare este furnizată manual urmăritorului. Aceasta înseamnă că sistemele de urmărire a unui singur obiect ar trebui să poată urmări orice obiect care le este dat, chiar și un obiect pentru care nu a fost antrenat niciun model de clasificare disponibil.



În cazul urmăririi obiectelor multiple, după cum indică și numele, există mai multe obiecte de urmărit. Algoritmul de urmărire trebuie, în primul rând, să determine numărul de obiecte din fiecare cadru și, în al doilea rând, să țină evidența identității fiecărui obiect de la un cadru la altul.



## **BOOSTING :**

Algoritmul Boosting în sine nu poate nici să învețe, nici să prezică nimic, deoarece este construit pe un alt algoritm (slab). Termenul "Boosting" se referă la o familie de algoritmi care transformă învățătorii slabi în învățători puternici pentru a minimiza erorile de instruire. În cadrul acestei metode, se selectează un eșantion aleatoriu de date, căruia i se aplică un model și apoi se antrenează secvențial - adică fiecare model încearcă să compenseze punctele slabe ale predecesorului său. La fiecare iterație, regulile slabe de la fiecare clasificator individual sunt combinate pentru a forma o singură regulă de predicție puternică.

Pentru a înțelege Boosting, este esențial să recunoaștem că boosting este un algoritm generic mai degrabă decât un model specific. Boosting are nevoie să specificați un model slab (de exemplu, regresie, arbori de decizie superficiali etc.) și apoi îl îmbunătățește.

## MIL (Multiple Instance Learning)

Multiple Instance Learning (MIL) este propusă ca o variantă a învățării supravegheate pentru probleme cu cunoștințe incomplete despre etichetele exemplilor de instruire. În învățarea supravegheată, fiecărei instanțe de instruire i se atribuie o etichetă discretă sau cu valoare reală. În comparație, în MIL, etichetele sunt atribuite doar la *saci de instanțe*. În cazul binar, un sac este etichetat pozitiv dacă cel puțin o instanță din acel sac este pozitivă, iar sacul este etichetat negativ dacă toate instanțele din el sunt negative. Nu există etichete pentru instanțele individuale. Scopul MIL este de a clasifica saci sau instanțe nevăzute pe baza *sacilor* etichetați ca date de instruire.

## KCF (Kernelized Correlation Filter)

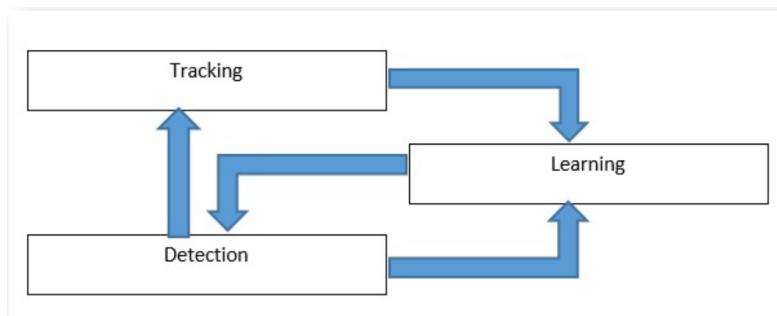
Componenta de bază a celor mai multe dispozitive moderne de urmărire este un clasificator discriminativ, care are sarcina de a face distincția între țintă și mediul înconjurător. Pentru a face față schimbărilor naturale ale imaginii, acest clasificator este de obicei antrenat cu imagini traduse și scalate. Astfel de seturi de eșantioane sunt pline de redundanțe - toți pixelii care se suprapun sunt constrânși să fie identici.

Pentru regresia cu kernel, derivăm un nou filtru de corelație kernelizat (KCF), care, spre deosebire de alte filtre kernel algoritmi de tip kernel are exact aceeași complexitate ca și omologul său liniar. Bazându-ne pe acesta, propunem, de asemenea, o extensie rapidă multi-canal a filtrelor de corelație liniare, prin intermediul unui nucleu liniar, pe care îl numim Filtru de corelație dublă (DCF).

## TLD (Tracking-Learning-Detection)

Urmărirea tradițională a obiectelor este ușor afectată de deformare, modificări de scară, modificări de iluminare, modificări parțiale ocluzii și așa mai departe. TLD (Tracking-Learning-Detection) este o metodă de algoritm clasic eficient în urmărirea pe termen lung care poate rezolva bine aceste probleme. Între timp, performanța în timp real a sistemului trebuie să fie luată în considerare în timp ce se află în situația reală.

Obiectul de interes este definit de o cutie delimitatoare într-un singur cadru. TLD urmărește simultan obiectul, îi învață aspectul și îl detectează ori de câte ori apare în videoclip.



### **MedianFlow**

Este descris un algoritm îmbunătățit al fluxului de mijloc utilizat pentru urmărirea vizuală a obiectelor. Îmbunătățirea constă în selectarea adaptivă a mărimii ferestrei de deschidere și a numărului de niveluri ale piramidei la estimarea fluxului optic. Aceasta poate crește eficiența urmăririi în comparație cu algoritmul de bază, în special în cazul obiectelor mici și cu contrast redus. Analiza fluxului median îmbunătățit a fost realizată pe secvențe video reale. Rezultatele obținute arată versatilitatea și robustețea computațională a algoritmului.

### **MOSSE (Minimum Output Sum of Squared) Error)**

Filtrul MOSSE este un filtru de corelație stabil care poate fi inițializat pe un singur cadru al unui videoclip. Filtrul MOSSE se adaptează în funcție de schimbările de aspect ale obiectului în timpul urmăririi. Urmărirea cu ajutorul filtrului MOSSE nu depinde de schimbările de iluminare, de transformările nerigide, de postură și de scară. Această lucrare prezintă pașii sistematici pentru construirea filtrului și aplicarea acestuia ca urmăritor în urmărirea vizuală.

### **CSRT (Channel and Spatial Reliability Tracking)**

CSRT. Acest tracker funcționează prin antrenarea unui filtru de corelație cu caracteristici comprimate. Filtrul este apoi utilizat pentru a căuta zona din jurul ultimei poziții cunoscute a obiectului în cadre succesive. Avantaje: - Mai lent, dar mai precis decât KCF.

Rezultatele experimentale au demonstrat că trackerul CSRT prezintă rezultate mai bune de urmărire prin integrarea modelului de detectare a obiectelor, mai degrabă decât prin utilizarea algoritmului de urmărire sau a filtrului în sine.

Concluzie:

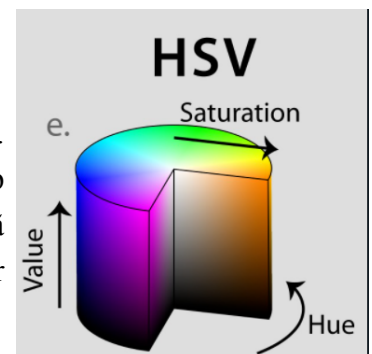
- CSRT: precizie bună, dar este mai lent decât ceilalți
- KCF: precizie nu atât de bună, dar este rapid
- MOSSE: este cel mai rapid

Un alt algoritm pe care l-am folosit este Opticalflow.

## Opticalflow

Fluxul optic este un câmp vectorial între două imagini, care arată modul în care pixelii unui obiect din prima imagine pot fi mutați pentru a forma același obiect în a doua imagine. Este un tip de învățare prin corespondență, deoarece, dacă se cunosc pixelii corespunzători ai unui obiect, se poate calcula câmpul fluxului optic.

Fluxul optic este o tehnică utilizată pentru a descrie mișcarea imaginii. De obicei, se aplică la o serie de imagini care au un mic pas de timp între ele, de exemplu, cadre video. Fluxul optic calculează o viteză pentru punctele din imagini și oferă o estimare a locului în care s-ar putea afla punctele în următoarea secvență de imagini.



Ne concentrăm pe obținerea unui vector de deplasare pentru obiectul care urmează să fie urmărit de-a lungul cadrelor. Urmărirea cu ajutorul fluxului optic se bazează pe trei ipoteze importante:

1. Consistența luminozității: Se presupune că luminozitatea din jurul unei regiuni mici rămâne aproape constantă, deși locația regiunii se poate schimba.
2. Coerența spațială: Punctele vecine din scenă aparțin, de obicei, aceleiași suprafețe și, prin urmare, au mișcări similare.
3. Persistența temporală: Mișcarea unei zone are o schimbare treptată.
4. Mișcare limitată: Punctele nu se deplasează foarte mult sau în mod dezordonat.

# Optical flow - sparse

Intensity predictor  
indica movementul predictor

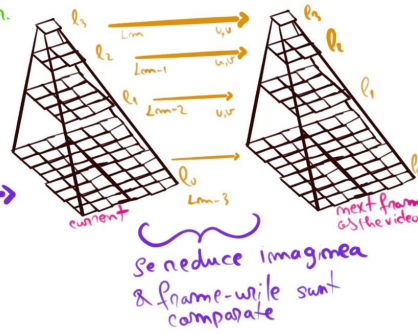
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$R = \det M = \frac{1}{4} (\text{trace } M)^2$  purpose: găsirea  
direcției vectorilor.  
 $\det M = \lambda_1 \lambda_2$  highest &  
 $\text{trace } M = \lambda_1 + \lambda_2$  2nd highest values

- Shi și Tomasi au modificat the Harris Corner.

$$R = \min(\lambda_1, \lambda_2)$$

OpenCV method  $\Delta$  KLT (Kornhuber, Lucas, Tomasi)  
 — for smaller images of the same frame





## **Bibliografie:**

<https://viso.ai/deep-learning/object-tracking/>

<https://paperswithcode.com/task/object-tracking>

<https://www.v7labs.com/blog/object-tracking-guide>

[https://docs.opencv.org/4.x/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html)

<https://nanonets.com/blog/object-tracking-deepsort/>

<https://www.scitepress.org/Papers/2020/91838/91838.pdf>

[https://machinelearningknowledge.ai/learn-object-tracking-in-opencv-python-with-code-examples/  
#i\\_KCF\\_Object\\_Tracking](https://machinelearningknowledge.ai/learn-object-tracking-in-opencv-python-with-code-examples/#i_KCF_Object_Tracking)

<https://broutonlab.com/blog/opencv-object-tracking>

<https://www.sciencedirect.com/topics/engineering/optical-flow>

<https://learnopencv.com/optical-flow-in-opencv/>