

# Segmentarea participanților la trafic

Oprica Andrei

Informatică Aplicată

Anul 3

Proiectului îi spune segmentarea participanților la trafic, însă am ajuns doar la performanța de a detecta mașiniile ce circulă într-o anumită zonă de interes ce este selectată în argumentele funcției ce se ocupă cu procesarea și afișarea. Și camera ce filmează trebuie să filmeze dintr-un punct fix. Pentru videoclipurile ce le am testat programul a detectat toate vehiculele din zona de interes, însă am avut câteva erori, acestea constând în detectarea nu foarte corectă a vehiculelor (adică detecta, dar putea găsi vehiculul și separat îi găsea și parbrizul, acesta ar fi un exemplu). Din această cauză nu pot nici să fac tracking pe vehicule, adică am reușit să încerc să fac tracking, dar pentru că programul detectează în unele cazuri și mașina și separat și parbrizul acesteia, astfel programul crede că aceeași mașină sunt altele diferite.

În realizarea acestui proiect am utilizat python și am utilizat librăria cv2 (OpenCV).

Partea de funcționalitate a proiectului constă în două fișiere .py: Primul este un fișier denumit tracker.py în care am implementat o clasă denumită EuclideanDistTracker ce reține un dicționar în care v-or fi liste cu punctele participanților și un numărător care indică identificatorul participantului. Această clasă conține și o metodă update() ce calculează distanța dintre punctele a două liste și în funcție de asta știe dacă este același participant sau este altul.

Al doilea fișier este un simplu fișier main.py în care import librăria cv2 și fișierul tracker. Apoi folosesc o funcție ce primește cinci argumente: un string cu calea videoclipului ce trebuie analizat și patru int-uri reprezentând regiunea de interes pe care o analizăm. În funcție creez un obiect de tipul EuclideanDistTracker, un obiect ce extrage fundalul pentru detectarea participanților. Apoi deschid videoclipul și îl iau pe cadre, din fiecare cadru îmi selectez regiunea de interes pentru a avea o

detectie cât mai corectă. Asupra regiunii de interes aplic detectarea obiectelor, urmat de două threshold-uri (primul pentru a elimina umbrele create de obiecte și a doua pentru a obține o definiție mai bună, prin eliminarea reflexiilor), urmat de un filtru bilateral în încercarea de a reduce reflexiile. După am căutat contururile cu ajutorul unei funcții oferite de cv2 și am căutat în acestea, iar dacă aveau o arie destul de mare le adaug la detecții. Până aici am făcut detecție și a mers bine. În continuare am încercat să urmăresc participanții adăugând un identificator fiecărei detecții. Aici am avut probleme din cauza reflexiilor și a detecției întrerupe a participanților astfel rezulta că același participant i se schimba numărul de identificare. Prin urmare am scos partea de numerotare a participanților. După sunt afișate cadrele pe rând. Această funcție este apelată din main cu cele 5 argumente și poate fi întreruptă prin apăsarea tastei escape.

Pentru KPI's am testat proiectul pentru cinci cazuri videoclipuri. Pentru fiecare caz pot acorda maxim 20%. Astfel am avut următoarele rezultate:

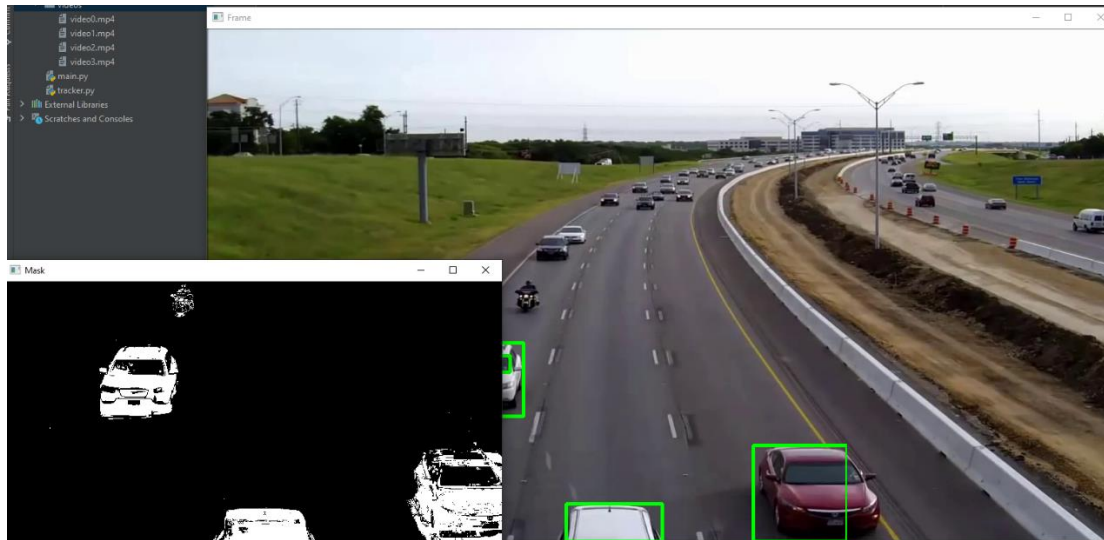
- Pentru primul caz am avut lumină normală și destul de multe obiecte (vehicule) privite din față.
  - În acest caz am detectat foarte bine, a detectat toate obiectele. Au fost foarte puține erori ca detectarea a aceluiași obiect de două ori.
- Acestei părți îi acord 20%.
- Pentru al doilea caz am avut lumina puternică, dar și umbre; obiecte văzute de deasupra-față.

- Aici au fost detectate toate obiectele, masca nu a detectat perfect, un vehicul a fost detectat de două ori.
- Acestei părți îi acord 15%.
- Pentru al treilea caz am avut lumina puternică, umbră și mai multe vehicule privite din față.
- În acest caz am detectat foarte bine, a detectat toate obiectele. Au fost foarte puține erori ca detectarea a aceluiași obiect de două ori.
- Acestei părți îi acord 15%.
- Pentru al patrulea caz am avut lumină normală, dar foarte multe vehicule.
- Aici au fost detectate aproape toate obiectele, dar a fost detectat și nimic, de asemenea unele dintre vehicule au fost detectate de două ori.
- Acestei părți îi acord 10%.
- Pentru al cincilea caz am avut lumină puternică și vehiculele privite din spate.
- Aici au fost detectate toate obiectele, masca nu a detectat perfect, un vehicul a fost detectat de două ori.
- Acestei părți îi acord 15%.

Concluzionând am un punctaj total de 75% de corectitate a programului pentru testele ce le-am utilizat pentru a vedea performanța programului.

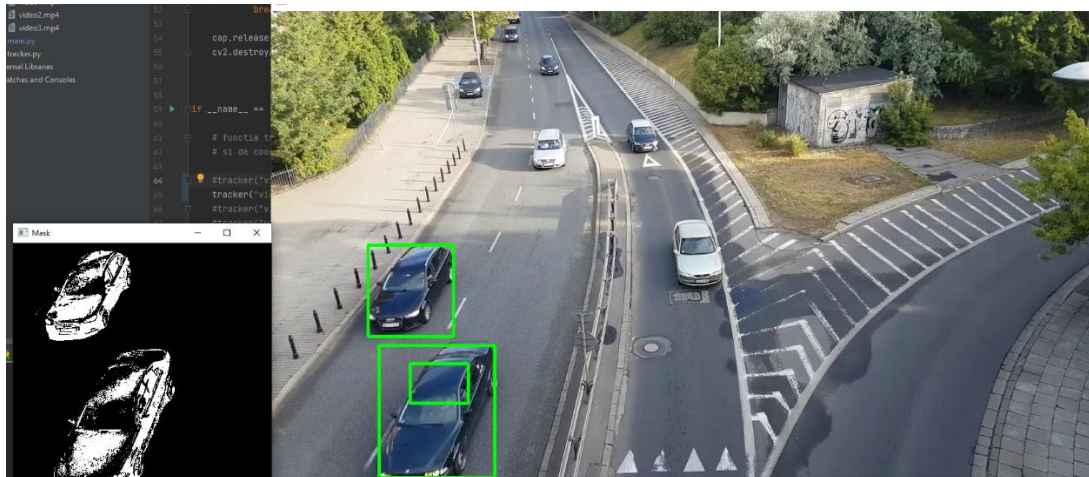
În cele ce urmează voi prezenta câteva rezultate a programului.

Primul exemplu:



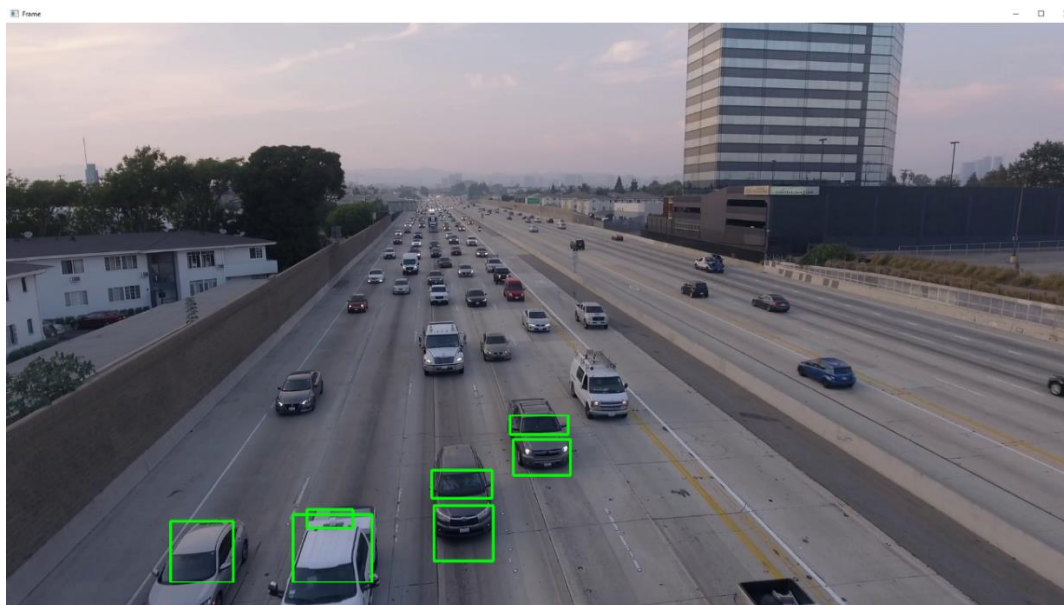
Partea alb-negru este masca pe care au fost aplicate threshold-urile și filtrul. De asemenea este zona de interes pe care se caută detecția. După cum se poate observa programul a detectat mașina albă din stânga, dar a detectat separat și parbrizul acesteia, ceea ce este puțin eronat.

Al doilea exemplu:



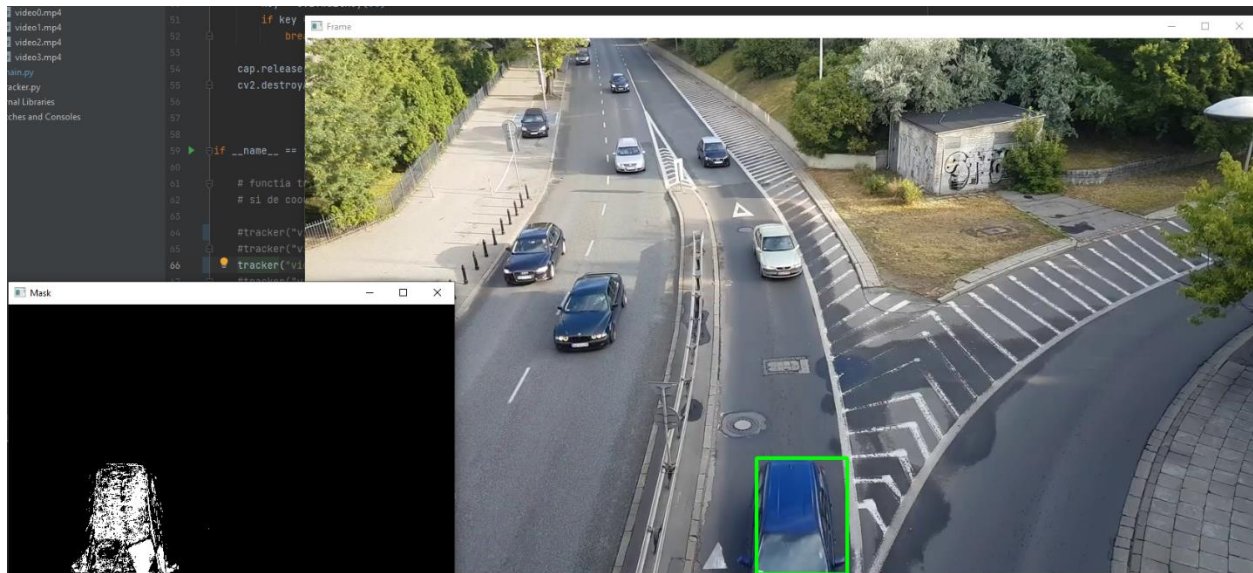
Și în cazul acestui exemplu se poate observa, chiar mai bine, eroarea de la primul exemplu, acesta fiind detectarea parbrizului mașinii din dreapta în cazul acestui videoclip.

Al treilea exemplu:



În acest exemplu se poate observa mai bine fiind mai mulți participanți.

Al patrulea exemplu:



Acest exemplu este tot videoclipul de la exemplul al doilea, dar cu o altă regiune de interes. În acest exemplu programul a detectat bine.

Implementarea programului poate fi găsită la adresa:  
<https://github.com/AndreiOprica/appCV>.

În primă fază am urmărit un tutorial pentru a crea clasa EuclideanDistTracker și pentru a putea citi și afișa videoclipul apoi am folosit câteva măști pentru a încerca să optimizez programul pentru datele mele de intrare.

# Bibliografie

- Tutorialul urmarit:

<https://youtu.be/O3b8IVF93jU>

- Câteva dintre videoclipurile pe care am adaptat programul:

- <https://youtu.be/MNn9qKG2UFI>
- <https://youtu.be/PJ5xXXcfuTc>
- <https://youtu.be/QuUxHIVUoaY>

- Documentație:

- [https://docs.opencv.org/4.x/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html)
- [https://docs.opencv.org/3.4/d4/d86/group\\_imgproc\\_filter.html](https://docs.opencv.org/3.4/d4/d86/group_imgproc_filter.html)
- [https://docs.opencv.org/4.x/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html)