

Documentatie Proiect

Scopul proiectului:

Scopul proiectului consta in realizarea unei aplicatii care, fiind furnizata cu o poza a unei masini, aplicatia sa poata identifica si citi numarul de inmatriculare al acesteia.

Pasii procesului sunt urmatoarii:

1. Conversie BGR-Grayscale
2. Blur Gaussian
3. Filtru Median
4. Conversie Binar
5. Detectarea muchiilor
6. Identificarea obiectelor conexe distincte
7. Incadrarea obiectelor in dreptunghiuri
8. Identificarea numarului de inmatriculare

Pasi de implementare:

Imaginea de intrare, furnizata aplicatiei este colora. De asemenea, putem presupune ca un numar de inmatriculare este asezat in poza cat de cat perpendicular.

Ca si prim pas de procesare, vom transforma imaginea din BGR in Grayscale, calculand media canalelor B, G si R. De asemenea, din imaginea initiala color vom mai crea o masca de albastru, in care retinem doar pixelii unde canalul de albastru este semnificativ mai intens decat cele de rosu si verde. Aceasta masca este utila deoarece stim ca numarul de inmatriculare are o bara albastra in stanga lui, lucru util in identificarea ulterioara a acestuia din poza initiala.

Urmeaza doua operatii de convolutie: una cu filtru Gaussian una cu filtru median.

Convolutia presupune aplicarea unui sablon (kernel) peste fiecare pixel al imaginii. Pentru fiecare pozitie, se face produsul dintre valorile kernelului si valorile pixelilor vecini, iar rezultatul se aduna si se atribuie pixelului central.

Filtrul Gaussian este folosit in prealabil unui algoritm de detectie a muchiilor si are ca scop netezirea imaginii, lucru realizat prin eliminarea variatiilor bruste de intensitate pe zone mici, pastrand totusi trecerile graduale(naturale) de intensitate.

Filtrul median este poate usor redundant, insa in situatii in care calitatea imaginii initiale lasa de dorit, filtrarea mediana reduce zgomotul de tip Salt & Pepper, lucru, de asemenea util in prealabil detectiei muchiilor.

Ultimul pas inaintea detectiei muchiilor este binarizarea imaginii, Binarizarea se realizeaza cu calcularea automata a unui threshold in mod bimodal. Metoda bimodala presupune ca histograma imaginii are doua clase dominante de pixeli (fundal si obiect). Algoritmul incepe cu un prag initial si imparte histograma in doua grupuri: pixeli sub si peste prag. Se calculeaza mediile pentru fiecare grup, iar noul prag devine media celor doua medii. Procesul se repeta iterativ pana cand pragul se stabilizeaza (diferenta dintre iteratii este sub o eroare prestabilita). Astfel se obtine un prag optim de separare intre obiect si fundal.

Detectia muchiilor din imaginea binarizata se realizeaza folosind algoritmul lui Canny. Acesta functioneaza prin aplicarea unui Kernel de tip Sobel, care calculeaza gradientul de intensitate si directia acestuia in fiecare punct, urmand ca doar muchiile formate din maxime locale sau muchiile conectate la muchii maxime sa fie pastrate.

Pe imaginea cu muchiile evidentiata aplicam ulterior un algoritm de identificare a obiectelor conexe, care incepe si parcurge imaginea pixel cu pixel, pana cand se loveste de un pixel obiect, moment in care continua sa parcurga doar pixeli obiect din vecinatatea de 8 a pixelului prim detectat, pana cand nu mai exista. Dupa ce avem un obiect complet, continuam iterarea pixel cu pixel pana cand nu mai avem pixeli obiect nevizitati.

Aceste obiecte se stocheaza ca o lista de liste de pixeli. Pe fiecare din elementele listei mari, le parcurgem ulterior pentru stabilirea coordonatelor si dimensiunii dreptunghiului circumscris.

Cu lista de dreptunghiuri circumscrise mai avem acum 2 verificari de facut pentru identificarea unui numar de inmatriculare. Conform realitatii, un numar de inmatriculare are dimensiunea 530x110 mm, insemnand ca aspect ratio-ul este de 4,81. Pentru a permite mici variatii a perspectivei din care este vazut numarul de inmatriculare, am stabilit, prin testare, pragul de >4.0 si <5.2 pentru aspect ratio pentru a trece testul. De asemenea, verificam ca dreptunghiul are o suprafata de minim 3000 pixeli pentru a elimina artefactele.

In momentul de fata avem un dreptunghi care respecta conditiile de dimensiune pentru a fi un numar de inmatriculare. Testul definitiv se face folosind masca de albastru stocata la inceput, verificand daca in stanga dreptunghiului circumscris se afla pixeli obiect in masca de albastru.

Pentru validarea procesului s-au folosit poze din realitate, cu autovehicule amplasate perpendicular cu aparatul foto.



Se poate observa cum, in urma pasilor de procesare, numarul de inmatriculare este incadrat corect in poza de output.

Avem aici alt exemplu:

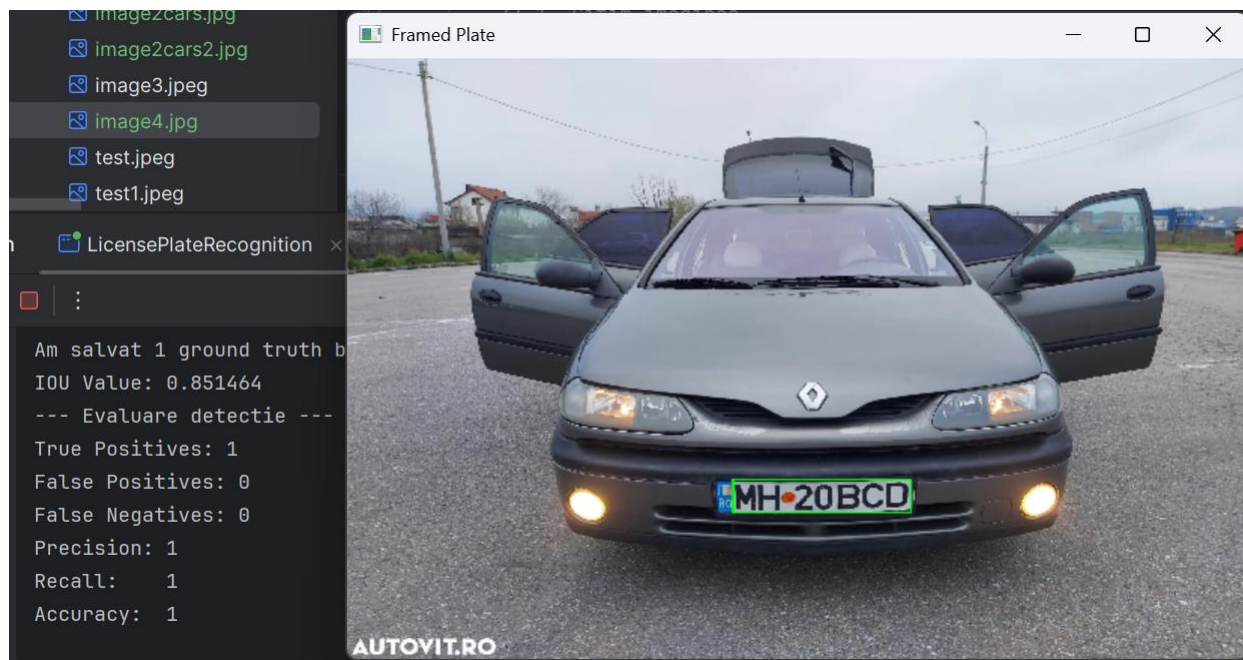


Testare:

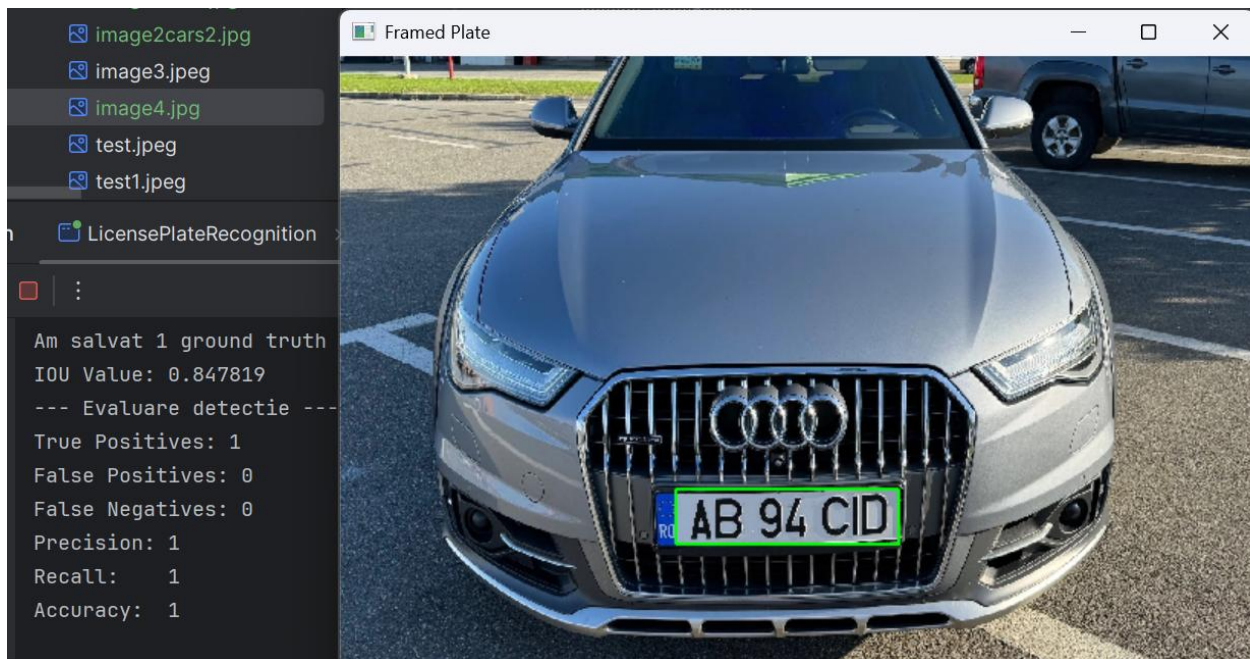
În cazul acestui proiect, ideea de testare este de a compara output-ul pipeline-ului dezvoltat de mine cu un ground truth (încadrarea manuală a numărului de înmatriculare), prin tehnica IoU.

IoU (intersection over union) este raportul între reuniunea ariei celor 2 dreptunghiuri împărțită la intersecția acestora. Pe scurt, asta rezultă în rata de suprapunere, exprimată numeric.

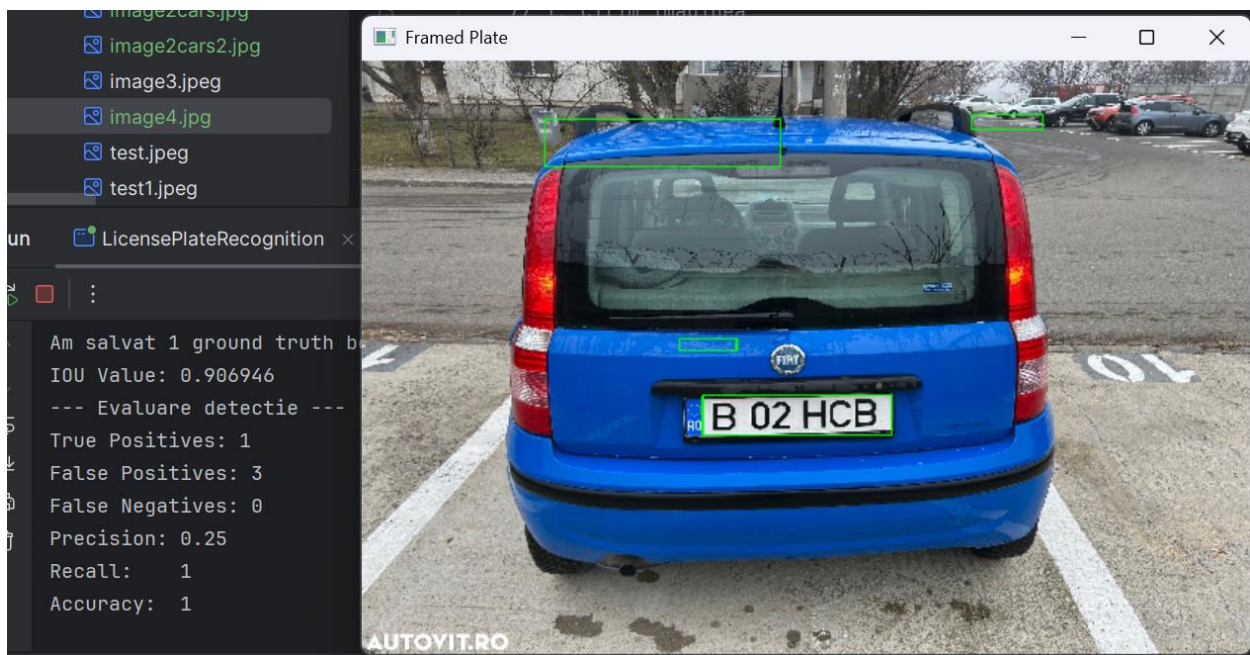
Alegerea ground truth-ului este gestionată tot de aplicație. În momentul rularii aplicației, se va deschide o fereastră cu imaginea originală în care, cu mouse-ul, se încadrează numărul sau numerele de înmatriculare în rectangle-uri. Punctul de start și dimensiunea rectangle-urilor sunt salvate în fișierul ground_truth.txt. Mai apoi, pe măsura ce algoritmul începe să găsească dreptunghiuri care respectă parametrii de aspect ratio și arie, acesta compară coordonatele dreptunghiului detectat cu setul de coordonate din ground_truth.txt și în funcție de asta, afișează în consolă niște metrice precum valoare efectivă a IoU, true positive, false positive, true negative etc.



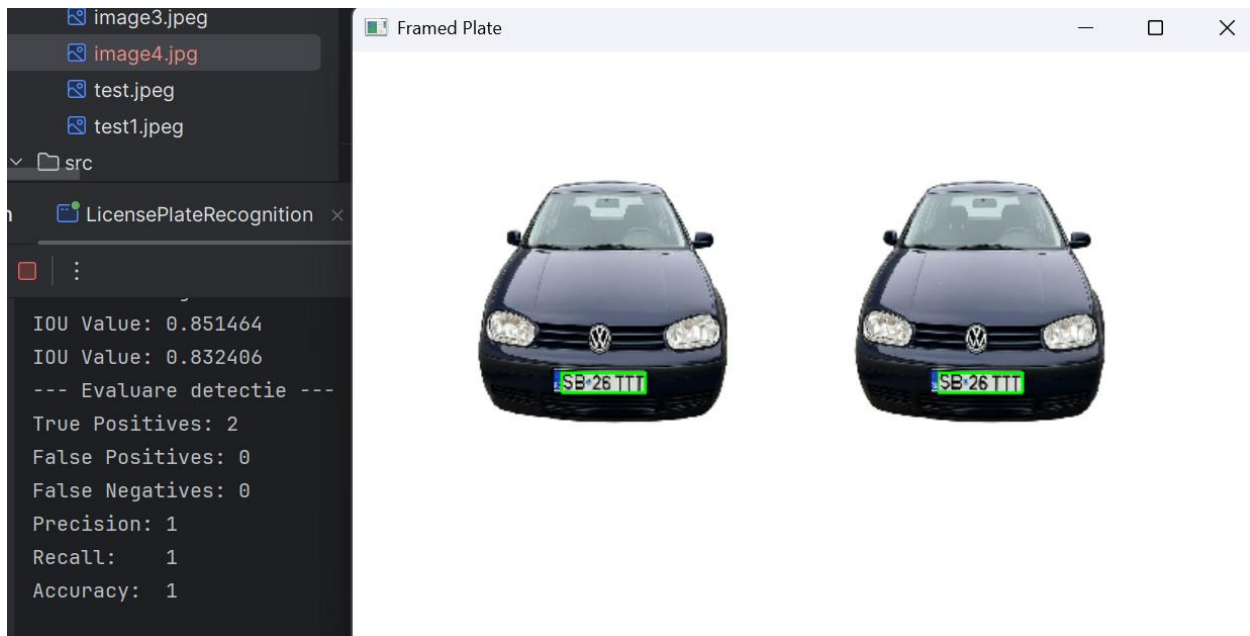
Avem aici un prim exemplu unde se vede că detectia e realizată cu succes. Valoarea IoU poate fi diferită de la rulare la rulare, deoarece ground truth-ul este manual, deci survine eroarea umană.



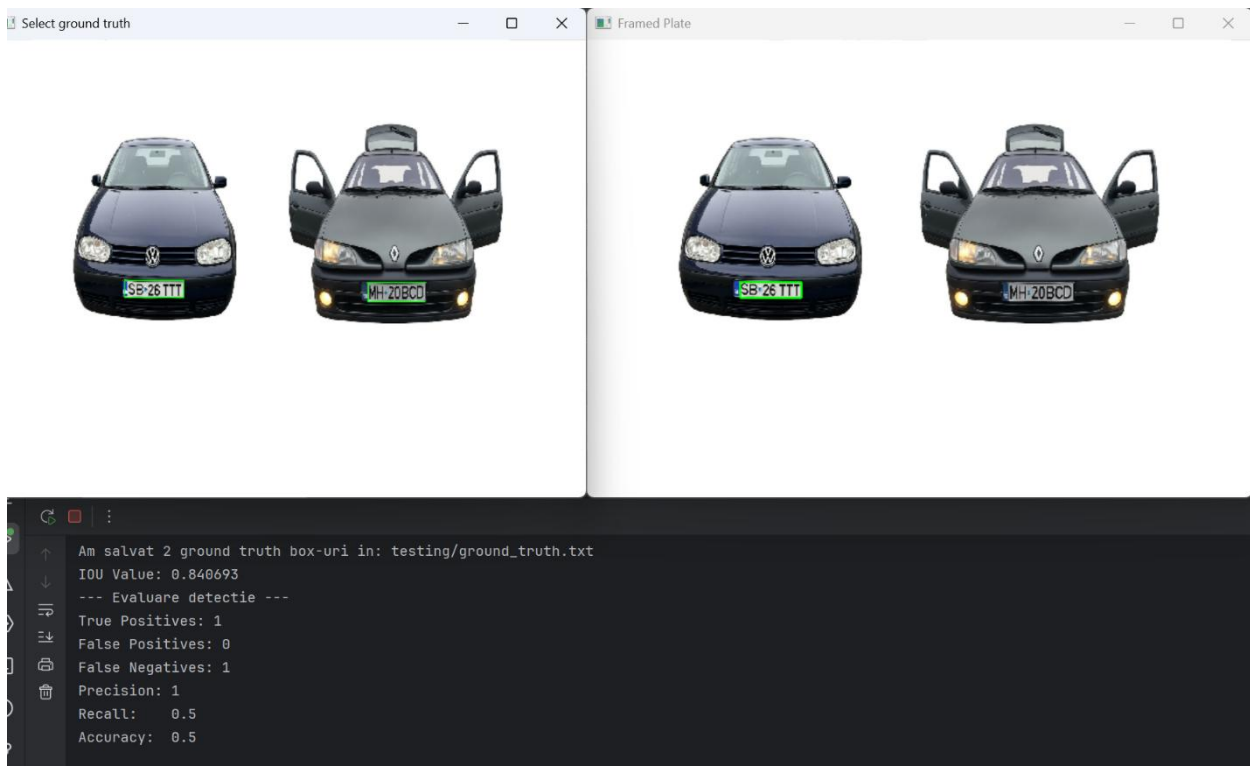
Aici avem un al doi-lea exepmlu pozitiv.



Aici avem un exeplu unde precizia este de doar 25%. Motivul pentru care artefactele sunt detectate ca potientale numere de inmatriculare este pentru ca, in acest caz particular, masca de albastru nu extrage doar dreptunghiul din stanga numarului de inmatriculare din cauza culorii masinii.

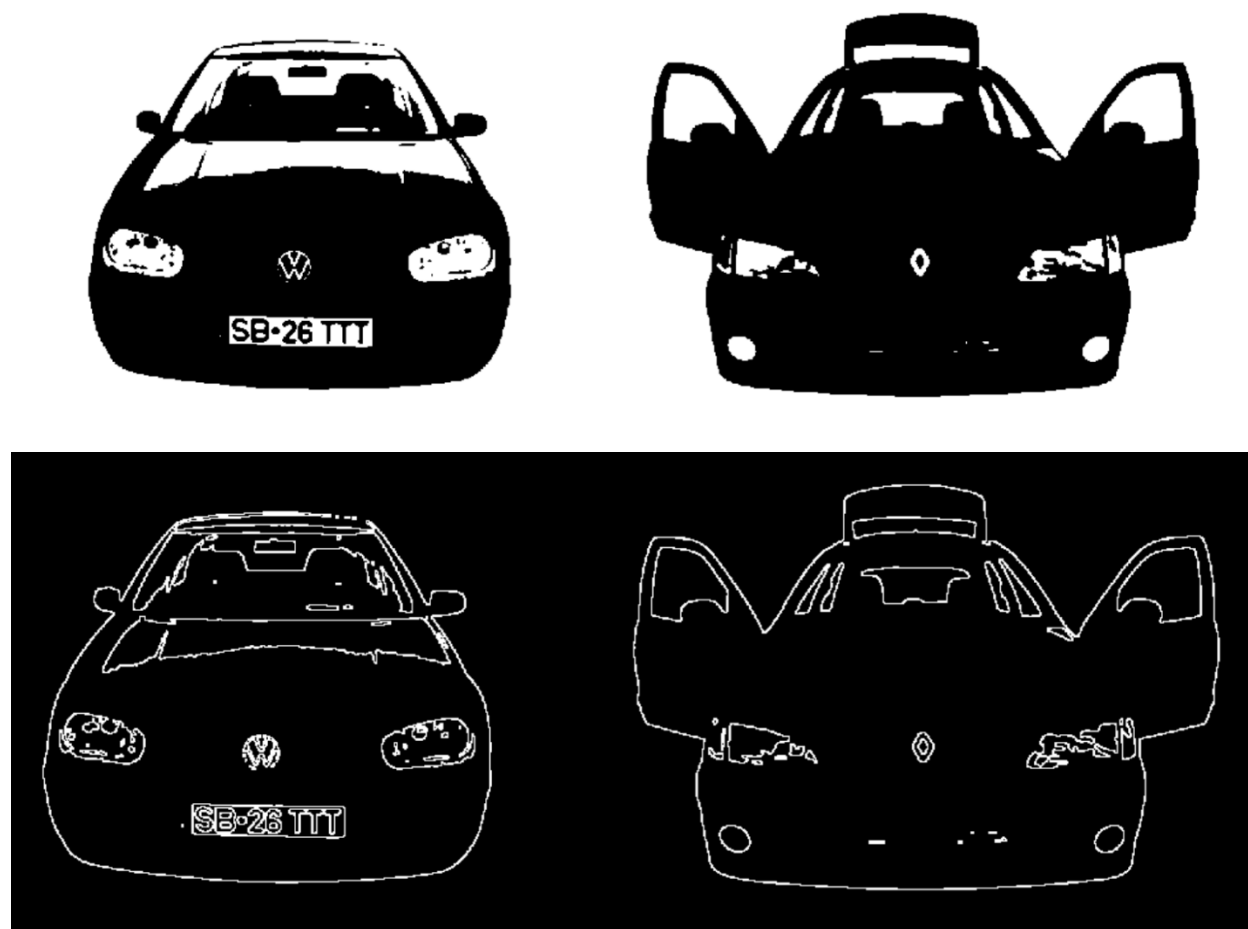


Aici avem un exemplu unde programul identifica corect 2 numere de inmatriculare. Acest exemplu este putin nedrept, din cauza faptului ca avem un fundal uni, care reduce riscul de artefacte dupa binarizare si Canny.



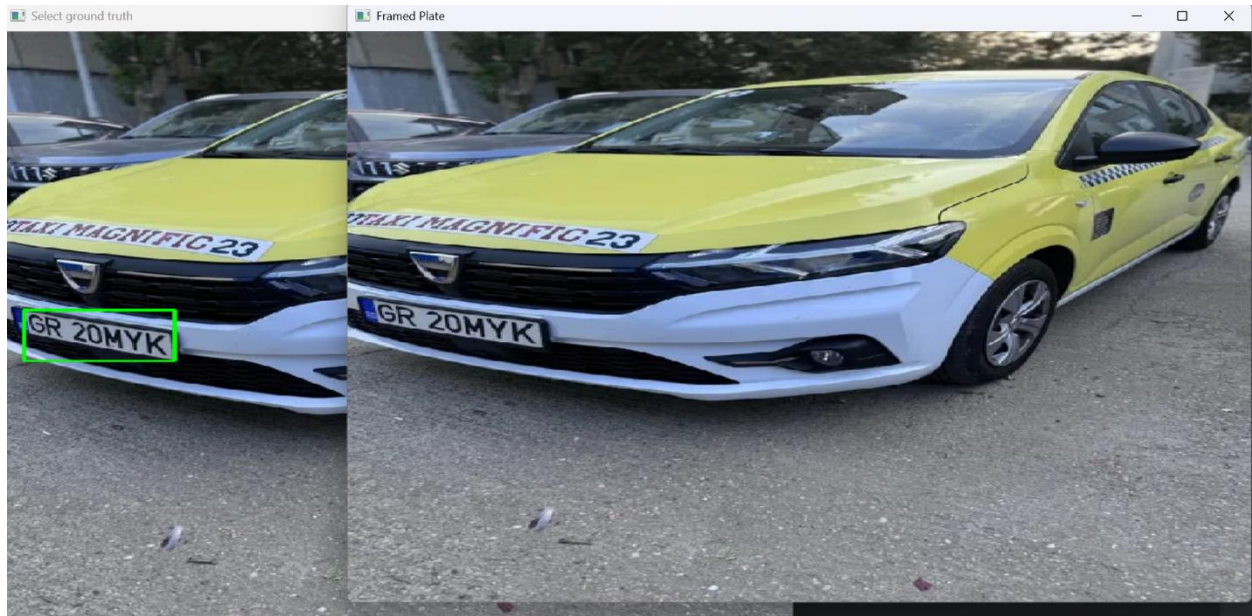
Acum un contra exemplu cu 2 masini. Aici, programul nu detecteaza al 2-lea numar de inmatriculare din cauza ca cele 2 masini au culori de saturatii diferite, lucru care afecteaza

calculul thresholdului bimodal de la pasul de binarizare, lucru care afecteaza si obiectele conexe detectate de Canny.



Mai jos avem exepu cu aceeaasi masina, cand este singura in poza.





Aici putem vedea ca algoritmul nu functioneaza. Asta este in cauza faptului ca algoritmul nu tine cont de perspectiva. In cazul asta, din cauza perspectivei, aspect ratio-ul este in afara parametrilor setati. Acest lucru se poate remedia cu perspective transform eventual.