

Detecția plăcuțelor de înmatriculare

*STICRIX-RMA06

1st Ciobanu Eduard-Tarciziu
1308A

Univ. Tehnica „Gheorghe Asachi”
Iasi, Romania
ciobanueduard@yahoo.com

2nd Sandu Cristi
1308A

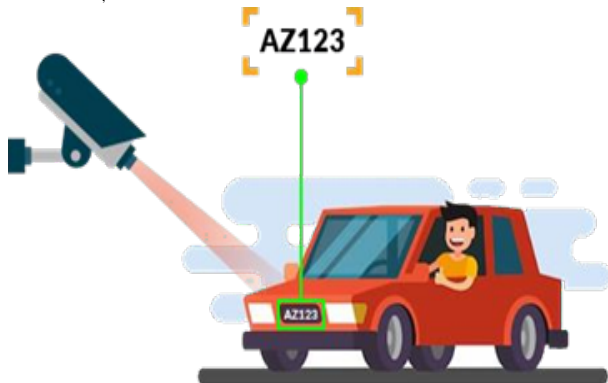
Univ. Tehnica „Gheorghe Asachi”
Iasi, Romania
sanducristi@yahoo.com

I. INTRODUCERE

Acest document este un model și instrucțiuni pentru \LaTeX .
Vă rugăm să respectați limitele paginilor conferinței.

II. DESCRIEREA TEMEI

Sistemele de recunoaștere automată a numerelor de înmatriculare montate pe autovehicule sunt considerate în general costisitoare, dar odată cu aglomerarea traficului urban și extraurban, extinderea parcurilor auto și, mai ales, digitalizarea și interconectarea la nivelul societății, ele au devenit necesare, atât din perspectiva îmbunătățirii siguranței publice și private, cât și din perspectiva îmbunătățirii calității serviciilor și a nivelului de confort.



Din punct de vedere al eficienței timpului, este mult mai favorabil să se folosească o recunoaștere automată a numerelor de înmatriculare și nu una manuală, care necesită

Ne propunem ca în acest proiect să implementăm un sistem de detectare a plăcuțelor de înmatriculare. Scopul acestui sistem este de a verifica valabilitatea asigurării auto corespunzătoare numărului de înmatriculare identificat și de a aplica amenzi asupra conducătorilor auto care nu au plătit.

Astfel, se urmărește eficientizarea traficului, a parcărilor și creșterea securității.



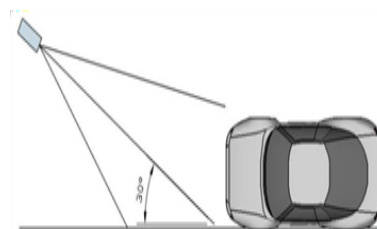
Vom folosi un clip video pentru detectarea numărului de înmatriculare, care ulterior va fi transmis către un site în vederea verificării asigurării auto(RCA).

Pentru eficiența funcției de recunoaștere a numerelor de înmatriculare, prescurtată și LPR, se poate utiliza o cameră care suportă această funcționalitate, instalată într-o anumită manieră, astfel încât imaginile cu plăcuțe să fie captate corect.

Pot exista diferențe în cerințele de amplasare a camerelor și numărul necesar de pixeli. Înainte de alegerea locului de instalare, utilizatorul trebuie să ia în considerare instrucțiuni obligatorii: deschiderea camerei, distanța ce trebuie respectată din locul camerei instalate până la punctul de detecție al plăcuțelor și dimensiunea corespunzătoare a imaginii plăcuței.

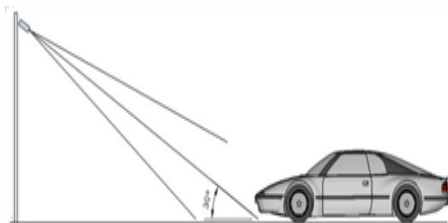
Iată câteva cerințe comune de instalare pentru camerele cu funcție LPR:

- Lățimea imaginii plăcuței de înmatriculare să fie între 130-300 pixeli (optim: 150-200 pixeli)
- Unghi vedere vertical maxim: 30°

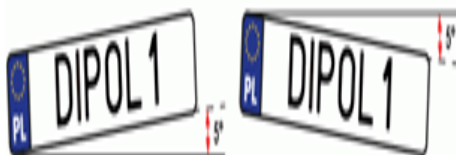


- Unghi vedere orizontal maxim: 30°

Identify applicable funding agency here. If none, delete this.



- Diferența unghiulară dintre marginea orizontală a imaginii și marginea orizontală a plăcuței să nu depășească $\pm 5^\circ$



III. NOȚIUNI DE BAZĂ

- Vom folosi notebook-ul Python Jupyter pentru a construi proiectul nostru, împreună cu două software open-source pentru a face magia să se întâmple, și anume OpenCV și TesseractOCR. Înainte de a continua, iată pașii pentru a configura aceste

OPENCV

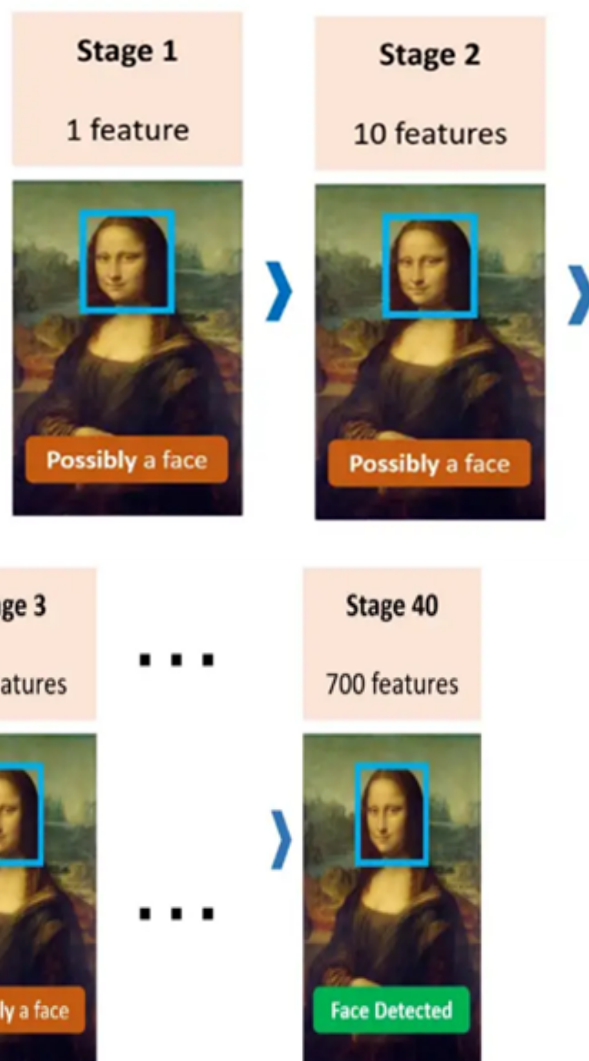
- OpenCV (Open Source Computer Vision Library) este o bibliotecă de software open-source pentru viziune pe computer și învățare automată. Se concentrează în principal pe procesarea imaginilor, captarea și analiza video, inclusiv funcții precum detectarea feței și detectarea obiectelor și ajută la furnizarea unei infrastructuri comune pentru aplicațiile de viziune computerizată.

- Pe lângă instalarea bibliotecii OpenCV, un alt lucru important de preluat este fișierul XML Haar Cascade .

- Să vorbim mai întâi despre teoria din spatele cascadelor Haar, deoarece este un concept important. În 2001, Paul Viola și Michael Jones au venit cu tehnica de detectare a obiectelor folosind clasificatoare în cascadă Haar bazate pe caracteristici. Este o abordare bazată pe învățarea automată (care implică AdaBoost) în care o funcție în cascadă este antrenată din multe imagini pozitive și negative. Extrage valori numerice pentru caracteristici (de exemplu, margini, linii) în mod eficient cu conceptul de imagine integrală (sau tabel cu suprafețe însumate), care depășește modalitatea implicită de calcul grea de a scădea sumele de pixeli din mai multe regiuni ale unei imagini întregi.

- În plus, folosește ” Cascada de clasificatori ”. Aceasta înseamnă că, în loc să aplici sute de clasificatori pentru multe caracteristici din imagine dintr-o dată (ceea ce este foarte ineficient), clasificatorii sunt aplicați unul câte unul. • Luați, de exemplu, o imagine a unui chip uman. Dacă primul clasificator pentru caracteristica ” ochi” a eșuat (adică nu reușește să detecteze ochii umani în imagine), algoritmul nu se deranjează să aplice clasificatorii ulterioare (de exemplu, nas, gură etc.). În schimb, se oprește și declară că nu este detectată nicio față.

- Pe de altă parte, dacă această primă caracteristică ”ochi” este detectată, algoritmul va aplica a doua etapă de clasificare a caracteristicilor și va continua cu procesul de clasificare. În final, dacă imaginea trece de toate etapele de clasificare, algoritmul poate declara că este prezentă o regiune a feței.



- OpenCV vine de fapt cu fișiere XML pre-antrenate ale diferitelor cascade Haar, unde fiecare fișier XML conține setul de caracteristici. Vom folosi fișierul Haar Cascade XML care conține funcțiile pentru plăcuțele auto rusești

IV. TESSERACTOCR

TesseractOCR este un motor de recunoaștere optică a caracterelor (OCR) open-source. Este recunoscut ca fiind unul dintre cele mai populare și mai precise motoare OCR open-source. Faptul distractiv este că acest motor a început de fapt ca un software proprietar dezvoltat de Hewlett Packard, dar a fost ulterior open-source în 2005, iar Google a sponsorizat de atunci dezvoltarea acestuia.

V. DETECTAREA PLĂCUTELOR AUTO CU OPENCV ȘI HAAR CASCADE

Acum este timpul să introducem setul de caracteristici Haar Cascade (fișier XML) pentru plăcuțele auto rusești, folosind funcția CascadeClassifier a OpenCV. Apoi, folosim detectMultiScale metoda CascadeClassifier pentru a rula detectarea.

Să vorbim pe scurt despre detectMultiScale metoda OpenCV. Metoda ne permite să detectăm obiecte de diferite dimensiuni în imaginea de intrare și returnează o listă de limite dreptunghiulare unde sunt detectate obiectele. Pentru fiecare dreptunghi vor fi returnate 4 valori, care corespund, respectiv:

- coordonata x a colțului din stânga jos al dreptunghiului (x)
- Coordonata y a colțului din stânga jos al dreptunghiului (y)
- lățimea dreptunghiului (w)
- înălțimea dreptunghiului (h)

Parametrii cheie implicați în detectMultiScale funcție sunt scaleFactor și minNeighbors .

• scaleFactor specifică cât de mult este redusă dimensiunea imaginii la fiecare scară a imaginii (ca parte a piramidei de scară , care este o reprezentare pe mai multe scară a unei imaginii). În esență, atunci când modelele de detectare a obiectelor sunt antrenate, acestea sunt antrenate să detecteze obiecte (adică plăcuțele de mașină în cazul nostru) de o dimensiune fixă și ar putea rata plăcuțele de mașină care sunt mai mari sau mai mici decât se aștepta. Ca parte a piramidei scării, imaginea este redimensionată de mai multe ori în speranța că o plăcuță de mașină va ajunge să aibă o dimensiune "detectabilă". Am folosit factorul de scară implicit de 1,1 , ceea ce înseamnă că OpenCV va reduce imaginea cu 10 la suta pentru a încerca să se potrivească mai bine cu plăcuțele mașinii.

• minNeighbors ne permite să specificăm câți vecini ar trebui să aibă fiecare dreptunghi candidat pentru ca dreptunghiul candidat să fie reținut. În termeni mai simpli, acest parametru influențează calitatea obiectelor detectate. O valoare mai mare are ca rezultat mai puține detecții, dar detecțiile vin cu o calitate și o acuratețe mai ridicate. Aceasta înseamnă că o valoare mai mare poate ajuta de fapt la reducerea numărului de fals pozitive.

VI. RECUNOAȘTEREA ȘI EXTRAȚIA NUMERELOR PLĂCUTELOR CU TESSERACTOCR

Extragem plăcuța de înmatriculare a mașinii și mărim imaginea Pentru a asigura succesul funcției OCR, trebuie să efectuăm o serie de pași de procesare a imaginii . Să începem prin a izola plăcuța de înmatriculare a mașinii ca imagine. Facem asta prin configurarea unei funcții similare cu ceea ce am făcut mai devreme cu detectarea plăcuței auto, doar că de data aceasta vom extrage și returnăm regiunea de interes (placă auto) ca o nouă imagine.

REFERENCES

- [1] Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs arxiv.org/pdf/1601.05610v1.pdf
- [2] An Efficient Approach For Automatic License Plate Recognition System
- [3] A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector arxiv.org/pdf/1802.09567v6.pdf
- [4] Deep Learning Methods for Automatic Number Plate Recognition System
- [5] Automatic License Plate Detection using KNN and Convolutional Neural Network
- [6] YOLO -Based Three-Stage Network for Bangla License Plate Recognition in Dhaka Metropolitan City
- [7] <https://www.alsecuritycameras.com/blog/what-is-lpr-technology/>
- [8] <https://www.alsecuritycameras.com/blog/what-is-lpr-technology/>
- [9] <https://docs.video.avasecurity.com/Products/aware/lpr/lpr-cam-consider.htm>.
- [10] <https://www.elko.ro/stiri/recunoasterea-automata-a-numerelor-de-inmatriculare-auto>