Universitatea Tehnică ”Gheorghe Asachi” Iași

Facultatea de Automatică și Calculatoare

Detecția plăcuțelor de înmatriculare

**Studenți:**

Chelarașu Elena-Denisa, 1308B

Miron Alexandru, 1308B

**Professor:**

Marius Gavrilescu

**Github:**

[Prelucrarea Imaginilor - Proiect](https://github.com/Marioalexsan/PI-Proiect)

**Introducere**

Interesul pentru metodele de prelucrare a imaginilor digitale își are rădăcinile în două idei:

* îmbunătățirea informației vizuale pentru interpretarea umană
* procesarea imaginilor pentru stocare, transmitere și ulterioara detecție a informației de către algoritmi specializați

În zilele noastre, procesarea imaginilor este folosită în foarte multe domenii. Spre exemplu, procesarea imaginilor se aplică în medicină (detecția cancerului, detecția organelor, vaselor sangvine, etc.), automotive (detecția plăcilor de înatriculare, automobile self-driving, etc.), și alte domenii.

În cadrul acestui proiect, ne propunem să realizăm detecția plăcuțelor de înmatriculare în imagini. Unele aplicări ale acestui algoritm sunt camerele pentru viteză, parcări private cu număr fix de locuri, vinietă, taxa de pod automată, etc.

**OpenCV**

OpenCV este o librărie open source, orientată spre vederea computerizată în timp real. Librăria oferă diverse toolkituri pentru lucrul cu 2D și 3D, recunoașterea fețelor, gesturilor, obiectelor, urmărirea mișcărilor, facilități pentru realizarea interfețelor grafice, și altele.

În cadrul proiectului, vom folosi OpenCV pentru facilitarea operațiilor de filtrare, de detecție a contururilor, și de editări ale imaginilor.

**Optical Character Recognition (OCR)**

OCR reprezintă translatarea imaginilor ce conțin text printat, scris de mână sau tipărit în text editabil.

OCR este un domeniu de cercetare în recunoașterea modelelor, inteligența artificială și vederea mecanică. Recunoașterea optică a caracterelor (folosind metode optice ca oglinzi și lentile) și recunoașterea digitală a caracterelor (folosind scanere și algoritmi pe calculator) au fost, inițial, considerate domenii diferite. Deoarece puține aplicații folosesc tehnici optice, termenul OCR include și procesarea digitală a documentelor.

Programele inițiale necesitau învățarea caracterelor (exemple ale fiecărui caracter) pentru a identifica un font specific. Astăzi există programe „inteligente” care au un grad mai mare de acuratețe, putând identifica majoritatea fonturilor. Unele programe sunt chiar capabile de a aranja textul pe coloane, imaginile și elementele non-textuale în pagină aproape identic cu sursa originală.

**Scurtă descriere a aplicației**

Aplicația va consta dintr-o interfață grafică cu mai multe elemente:

* Un buton pentru selectarea imaginii spre analiză - se va deschide o fereastră pentru căutare
* Un panou în care imaginea selectată va fi afișată; plăcuțele de înmatriculare detectate vor fi încadrate cu cutii roșii
* Un panou cu toate plăcuțele detectate, tăiate din imagine
* Un panou cu descrierea text a fiecărei plăcuțe detectate:
  + Descrierea poziționării pe imagine
  + Descrierea numărului de înmatriculare detectat (caracterele necunoscute se vor marca cu ”?”)
  + Descrierea țării de origine detectată (cod țară)
  + (Bonus) Validarea numărului de înmatriculare

**Obiective inițiale**

* Detecția plăcuțelor de înmatriculare
  + Plăcuțele vor fi încadrate cu o cutie verde pe imagine și vor fi separate într-o imagine aparte
* Citirea numărului de înmatriculare (total sau parțial - după condiții foto)
  + Textul recunoscut va fi afișat în cadrul interfeței
* Detecția țării ce corespunde plăcuței
* ... și alte informații ce pot apărea pe plăcuțe, în funcție de resursele disponibile

**Obiective realizate în etapa intermediară**

* Detecția plăcuțelor de înmatriculare cu încadrarea contururilor viabile în cutii verzi
* Segmentarea literelor de pe plăcuță
* Detecția literelor prin prelucrare manuală a plăcuței de înmatriculare (fară OCR)

**Obiective ce urmează a fi realizate**

**Sarcini**

* Detecția și segmentarea plăcuțelor de pe imagine
  + S-au segmentat plăcuțele din imagine cu ajutorul algoritmului Canny și a funcției findContours()
  + S-a realizat un algoritm pentru simplificarea contururilor și selecția contururilor dreptunghi pe baza unor criterii
* Detecția și segmentarea caracterelor de pe plăcuță
  + Folosing plăcuța segmentată, s-a aplicat algoritmul Canny și funcția findContours(), s-au extras contururile
  + Pentru fiecare literă s-a determinat un “bounding-box” (cutii)
  + S-a calculat mediana înălțimii și lățimii cutiilor
  + s-au eliminat contururile care aveau o înălțime cu o deviație mai mare de 10% față de înălțimea mediană și lățimea cu 10% față de lățimea mediană
  + s-a presupus că literele formează majoritatea contururilor din plăcuță și că literele au aproximativ aceeași înălțime (sunt scrise cu același font)
* Recunoașterea numărului de înmatriculare de pe plăcuță
  + S-a folosit un algoritm pentru calcularea distanței între două imagini
  + Acesta calculează distanța pătratică dintre pixelii celor două imagini
  + Aplicând acest algoritm între litera segmentată și fiecare literă din fontul de referință ( Mittelschrift ), selectăm litera cu cea mai mica distanță ca fiind litera recunoscută ca fiind cea corectă
  + Eventual vom calcula și orientarea conturului literelor pe regiuni ca un criteriu adițional
* Interfața grafică a aplicației
* Detecția tipului de plăcuță (țară de origine, etc.)
* Afișarea datelor pe ecran

Sarcinile vor fi făcute împreună până ce se va prezenta o oportunitate de a lucra separat.

**Algoritm aplicat pentru detecție**

* Citirea imaginii
* Aducerea imaginii în format grayscale
* Aplicarea unui filtru de netezire pentru reducerea zgomotului
* Transformarea imaginii în binar
* Aplicarea unor filtre de accentuare a contururilor
* Detecția contururilor
  + Contur din 4 segmente
  + Suprafață corespunzătoare unei plăci
  + Laturi aproximativ paralele (pentru poze cu unghi)
  + Laturi aproximativ drepte (pentru poze cu unghi)
* Decuparea plăcii rezultante
* Folosim OCR pentru detecția caracterelor din plăcuță
* Afișarea textului pe ecran

**Tehnologii folosite**

Aplicația va fi realizată în C++, folosind mediul de dezvoltare Visual Studio 2017 / 2019. Pentru prelucrarea imaginilor se va folosi OpenCV, iar pentru GUI - librăria Qt sau modulul HighH GUI de la OpenCV.

**Exemple din implementația preliminară**

”Plăcuțele” detectate sunt reprezentate printr-un dreptunghi verde!





Caz nefavorabil, unde pot fi eșuări:



**Bibliografie**

[Wikipedia: OCR](https://en.wikipedia.org/wiki/Optical_character_recognition)

[OpenCV Documentation](https://docs.opencv.org/)

[Wikipedia: Canny Edge Detector](https://en.wikipedia.org/wiki/Canny_edge_detector)