



**Universitatea
Transilvania
din Brașov**
FACULTATEA DE MATEMATICĂ
ȘI INFORMATICĂ

Programul de studii:
Informatică Aplicată

Lucrare de licență Detectia numerelor de inmatriculare

Absolvent: Doloiu Mihai Alexandru

Coordonator științific: Lect. Dr. Vlad Monescu

Brasov, 2024

Cuprins

1	Introducere	3
1.1	Scopul lucrării	3
1.2	Motivația alegerii temei	3
1.3	Descrierea capitolelor	3
2	Medii si concepte de programare	3
2.1	Limbajul de programare C++	3
2.2	Limbajul de programare Java	3
2.3	Rețele neurale artificiale	4
2.4	Rețele neurale convoluționale	4
2.5	Biblioteca OpenCV	4
2.6	Biblioteca YOLOv5	5
2.7	Framework-ul Qt	5
2.8	Framework-ul Spring	6
2.9	Platforma CMake	7
3	Detecția plăcuțelor de înmatriculare folosind OpenCV	8
3.1	Preprocesarea imaginii	8
3.1.1	Redimensionarea imaginii	8
3.1.2	Conversia imaginii in grayscale	8
3.1.3	Reducerea zgomotului	8
3.1.4	Operatia de Opening	8
3.1.5	Subtracting	8
3.1.6	Thresholding	8
3.2	Postprocesarea imaginii	8
3.3	Rezultate	8
4	Detecția plăcuțelor de înmatriculare folosind rețele neurale convoluționale	9
4.1	Setul de imagini utilizat	9
4.2	Antrenarea modelului	9
4.3	Interferența in C++	9
4.4	Rezultate	9
5	Recunoașterea caracterelor din placuța de înmatriculare	9
5.1	Preprocesarea imaginii	9
5.1.1	Undistortioning	9
5.1.2	Skew Correction	9
5.2	Rezultate	9
6	Modelarea entităților	9
7	Arhitectura aplicațiilor	9
8	Prezentarea aplicației C++	9

9	Prezentarea aplicației Web	9
10	Securitate	9
11	Concluzii și dezvoltări ulterioare	9

Listă de figuri

1	Structura directorului sursă al proiectului principal.	7
---	--	---

1 Introducere

1.1 Scopul lucrării

1.2 Motivația alegerii temei

1.3 Descrierea capitolelor

2 Medii si concepte de programare

2.1 Limbajul de programare C++

Inițial denumit "C cu clase", C++ este un limbaj de programare de nivel înalt, ce este considerat a fi o extensie și o îmbunătățire a limbajului C. Acesta păstrează toate caracteristicile limbajului C și aduce concepte specifice programării orientate pe obiecte, precum clase, obiecte, încapsulare, moștenire, polimorfism, gestionare de excepții și multe altele. Fiind un limbaj de programare orientat pe obiecte, accentul este pus mai mult pe "obiecte" și nu pe manipularea lor. Pe lângă caracteristica menționată anterior, C++ mai dispune și de alte caracteristici la care se numără:

- Viteza de compilare - deoarece C++ este un limbaj compilat, însemnând că codul sursă al acestuia este tradus direct în cod mașină, programul rezultat beneficiază de performanțe superioare în comparație cu alte limbaje de programare.
- Suport pentru pointeri - deși în ziua de astăzi acest lucru este indisponibil în multe alte limbaje de programare, C++ oferă o modalitate de a manipula și gestiona direct memoria cu ajutorul pointerilor.
- Portabilitate - fiind unul dintre cele mai utilizate limbaje din lume, codul C++ poate fi scris astfel încât să ruleze pe mai multe platforme hardware și sisteme de operare.

Având în vedere caracteristicile menționate, acestea contribuie la eficiența și adaptabilitatea sa, facilitând dezvoltarea aplicațiilor pe diverse platforme și în diverse contexte. Astfel, fiind unul dintre cele mai importante și puternice limbaje de programare, am decis ca implementarea codului sursă al aplicației pentru detectarea și recunoașterea numerelor de înmatriculare, dar și pentru gestionarea bazei de date a acesteia, să fie realizată în C++.

2.2 Limbajul de programare Java

Java, la fel ca și C++, este tot un limbaj de programare de nivel înalt, orientat pe obiecte, dar bazat pe clase, unde fiecare fișier este de fapt o clasă. A fost dezvoltat pentru a oferi portabilitate, performanță și securitate, fiind creat cu scopul de a fi independent de platformă, eficient și simplu. Atât Java, cât și C++, prezintă numeroase similarități din punct de vedere al sintaxei limbajului, însă limbajul Java este considerat a fi mai ușor de folosit. Printre principalele caracteristici ale limbajului Java se numără:

- Programarea orientată pe obiecte - Java, la fel ca și C++, dispune de conceptele specifice programării orientate pe obiecte, îmbunătățind practicitatea limbajului de programare prin oferirea unei modalități mai eficiente de structurare a codului, promovând reutilizarea, flexibilitatea și modularitatea.
- Independent de platformă - programele create în Java pot fi rulate pe orice platformă hardware și sistem de operare datorită faptului că codul sursă este tradus într-un format intermediar numit "Java Bytecode" care permite rularea programului oriunde se dispune de o mașină virtuală Java instalată (*JVM*¹). JVM-ul încarcă, verifică și execută Java Bytecode-ul, asigurând totodată gestionarea memoriei, gestionarea excepțiilor, etc.
- Securitate - asupra securității s-a pus un accent deosebit, fiind asigurată prin diverse mecanisme, precum Sandboxing, care constă în izolarea resurselor periculoase ale sistemului de operare prin rularea aplicației Java într-un mediu controlat. Un alt mecanism este Garbage Collection, care previne scurgerile de memorie și depășirile de buffer.

În prezent, Java este utilizat în diverse aplicații și site-uri web, datorită numeroaselor avantaje pe care le oferă. Pentru site-ul web prezentat în această lucrare, implementarea acestuia va fi realizată cu ajutorul limbajului de programare Java și a *framework-ului*² Spring.

Prin urmare, limbajul de programare Java se remarcă prin portabilitate, securitate și performanță. Datorită implementării în mod robust a conceptelor fundamentale ale programării orientate pe obiect, se facilitează crearea de cod bine structurat și ușor de întreținut.

2.3 Rețele neurale artificiale

Salut

2.4 Rețele neurale convoluționale

Salut

2.5 Biblioteca OpenCV

*OpenCV*³ este o bibliotecă open-source specializată în vedere computerizată, procesarea imaginilor și învățare automată, deținând peste 2500 de algoritmi ce pot fi utilizați în diverse aplicații, precum recunoașterea facială, detectarea obiectelor, analiza medicală. Aceasta a fost scrisă în limbajul de programare C++, însă este compatibilă și cu alte limbaje de programare, inclusiv C++, Python, Java și Matlab.

Pentru procesarea de imagini, OpenCV oferă o gamă variată de funcții pentru manipularea

¹Java Virtual Machine - Mașină virtuală Java

²Framework - set de biblioteci, unelte și convenții de programare care oferă o infrastructură pentru dezvoltarea rapidă și eficientă a aplicațiilor.

³OpenCV - Open Source Computer Vision Library.

imaginilor, precum conversia de la un spațiu de culoare la altul, ajustarea luminii și a contrastului, transformări geometrice, filtrare și altele. De asemenea, în cadrul bibliotecii, există un modul numit OpenCV *DNN*⁴, dedicat utilizării și implementării DNN-urilor în aplicații de computer vision. Oferă funcționalități puternice în implementarea inferenței modelelor DNN și suporta integrarea cu diferite framework-uri, precum TensorFlow, Caffe, Yolo, Darknet și altele.

În proiectul C++, pentru detecția și recunoașterea numerelor de înmatriculare ale mașinilor, s-a utilizat versiunea 4.8.0 de OpenCV. Biblioteca a fost utilizată de la începutul proiectului, în timpul detecției plăcuței de înmatriculare și a literelor de pe aceasta, unde s-au folosit algoritmi adecvați, până la final, la recunoașterea caracterelor, unde s-a folosit un set de date ce conține toate caracterele de pe plăcuțele din România și s-a realizat template matching.

2.6 Biblioteca YOLOv5

*YOLOv5*⁵, dezvoltată de către echipa Ultralytics, este o bibliotecă de Deep Learning și un framework pentru detecția diferitelor obiecte în imagini și videoclipuri. Ca structură, se bazează pe o abordare de tip "You Only Look Once", însemnând că în loc ca procesul de detecție să fie împărțit pe etape separate, cum ar fi detecția, clasificarea și localizarea, acesta este tratat ca o singură rețea neurală, abordare ce permite furnizarea predicțiilor în timp real. Astfel, biblioteca YOLO este cunoscută pentru eficiența sa.

Biblioteca permite utilizatorilor să antreneze modele folosind seturi de date proprii, folosind una dintre arhitecturile variate pe care biblioteca le furnizează, cum ar fi YOLOv5s (Small), YOLOv5m (Medium), YOLOv5l (Large), și realizarea inferenței (detecție de obiecte) pe imagini noi.

În proiectul dezvoltat pentru aceasta lucrare de diplomă, detecția plăcuțelor de înmatriculare se poate realiza, la alegerea utilizatorului, atât prin folosirea algoritmilor de procesare de imagini din OpenCV, cât și prin folosirea unui model antrenat cu ajutorul acestei biblioteci.

2.7 Framework-ul Qt

Qt este un framework utilizat pentru crearea de aplicații cross-platform, precum serverele, care rulează pe diverse platforme software și hardware, cum ar fi Windows, Linux, OS X, Android, iOS, și altele. Preprocesorul *MOC*⁶ extinde limbajul C++ prin adăugarea de noi caracteristici necesare în dezvoltarea aplicațiilor cu interfață grafică, cum ar fi semnalele și sloturile. Există numeroase aplicații populare cu interfață grafică realizate cu ajutorul framework-ului Qt și folosite de milioane de utilizatori din întreaga lume, printre care se numără Skype, Google Earth și browser-ul Opera.

Proiectarea și crearea de *GUI*⁷ se realizează cu ajutorul unelei grafice incluse în framework,

⁴DNN - Deep Neural Networks.

⁵YOLOv5 - You Only Look Once, versiunea 5.

⁶MOC - Meta-Object Compiler.

⁷GUI - Graphical User Interface

Qt Designer. Elementele grafice sunt atașate codului prin folosirea mecanismului de sloturi și a semnalelor Qt. Qt Designer generează automat codul sursă asociat cu interfața, odată ce aceasta este proiectată. Acest lucru elimină nevoia programatorilor de a scrie manual codul pentru fiecare element din interfață, economisind timp și evitând potențialele erori. Proiectarea interfețelor grafice ale aplicațiilor se realizează foarte ușor, având posibilitatea adăugării și configurării a diferitelor elemente GUI, cum ar fi butoanele, etichetele, textbox-uri și altele, utilizând pur și simplu drag-and-drop. Utilizarea Qt Designer implică patru etape de bază:

- Alegerea propriei interfețe și a obiectelor dorite;
- Așezarea obiectelor pe interfața;
- Conectarea semnalelor la sloturile corespunzătoare;
- Vizualizarea interfeței.

Pentru realizarea interfeței grafice a aplicației principale, responsabilă pentru detecția și recunoașterea automată a numerelor de înmatriculare ale mașinilor, s-a folosit framework-ul Qt. Meniul acesteia oferă utilizatorului o metodă ușoară și intuitivă de configurare a camerelor video și a locurilor de parcare, folosite pentru monitorizarea traficului din interiorul unei parări.

2.8 Framework-ul Spring

Spring este un framework open-source și cross-platform, ce oferă un model cuprinzător de programare și configurare a aplicațiilor moderne de întreprindere bazate pe limbajul de programare Java. Deoarece se axează pe suportul infrastructural al aplicațiilor, echipele de dezvoltare pot dedica mai mult timp logicii la nivel de aplicație, fără legături inutile cu medii de implementare specifice. Printre principalele caracteristici ale framework-ului se numără:

- Inversiunea de control - containerul Spring preia responsabilitatea gestionării obiectelor și dependențelor, oferind astfel o structură modulară și ușor de gestionat.
- Spring Data - o abordare simplificată în ceea ce privește lucrul cu baze de date relaționale și non-relaționale.
- Spring Web - furnizează suport pentru dezvoltarea aplicațiilor web, oferind funcționalități precum gestionarea cererilor HTTP, manipularea parametrilor de solicitare, și facilitarea construirii de aplicații web robuste și scalabile.
- Spring Security - se ocupă de aspectele legate de securitatea aplicațiilor Java, furnizând suport pentru autentificare, autorizare și alte aspecte de securitate.

Pentru realizarea aplicației secundare, ce constă într-un site web, s-a folosit framework-ul Spring. Datorită componentei Spring Security, utilizatorii își pot crea un cont asociat numărului de înmatriculare corespunzător mașinii detectate la intrarea din parcare. Odată

conectați, aceștia pot vizualiza informații, precum timpul petrecut în parcare, ora intrării, locurile de parcare unde a fost localizată mașina, și transmiterea unor întrebări în cadrul unui chat și a eventualelor plângeri.

2.9 Platforma CMake

CMake este un software open-source și cross-platform pentru automatizarea proceselor de build, testing, packaging și instalare a programelor prin utilizarea unei metode independente de compilator. CMake generează fișierele de compilare ale unui alt sistem, prin urmare acesta nu este un sistem de compilare.

Pentru generarea fișierelor de compilare, se vor utiliza fișiere numite "CMakeLists.txt" ce vor fi plasate în fiecare director sursă. Acest lucru funcționează prin producerea unui mediu de construire nativ unde se vor crea bibliotecile, se vor genera pachetele, se va compila codul sursă și ulterior va construi executabile.

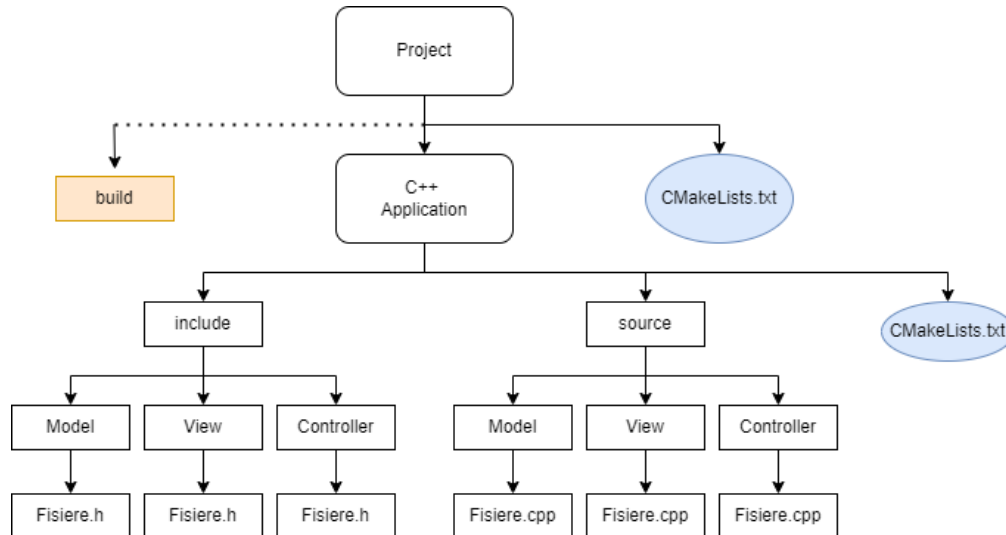


Figura 1: Structura directorului sursă al proiectului principal.

În figura 1 este evidențiată structura directorului sursă al proiectului, precum și utilizarea fișierului "CMakeLists.txt". După ce proiectul este generat și configurat, se creează directorul "build", unde vor fi prezente toate informațiile necesare rulării proiectului.

3 Detecția plăcuțelor de înmatriculare folosind OpenCV

3.1 Preprocesarea imaginii

3.1.1 Redimensionarea imaginii

3.1.2 Conversia imaginii in grayscale

3.1.3 Reducerea zgomotului

3.1.4 Operatia de Opening

3.1.5 Subtracting

3.1.6 Thresholding

3.2 Postprocesarea imaginii

3.3 Rezultate

- 4 Detecția plăcuțelor de înmatriculare folosind rețele neurale convoluționale
 - 4.1 Setul de imagini utilizat
 - 4.2 Antrenarea modelului
 - 4.3 Interferența în C++
 - 4.4 Rezultate
- 5 Recunoașterea caracterelor din placuța de înmatriculare
 - 5.1 Preprocesarea imaginii
 - 5.1.1 Undistortioning
 - 5.1.2 Skew Correction
 - 5.2 Rezultate
- 6 Modelarea entităților
- 7 Arhitectura aplicațiilor
- 8 Prezentarea aplicației C++
- 9 Prezentarea aplicației Web
- 10 Securitate
- 11 Concluzii și dezvoltări ulterioare