

Traffic Manager

Student: Mihai Andrei Gherghinescu
Supervizor: Lect. Dr. Todor Ivașcu

21 iunie 2023

Introducere

Una dintre principalele cauze ale congestiilor in trafic sunt intersectiile. Acest fenomen este in special observat in zonele urbane unde prezenta acestora este abundenta. Pentru a minimiza timpul pierdut atat cat si siguranta soferilor au fost dezvoltate sisteme de trafic inteligente(ITS). Acest concept a fost reinventat dealungul timpului iar in prezent este cuprins in notiunea de "smart city". Urmeaza sa prezint un scurt istoric al evolutiei ITS cat si sa prezint contributia proprie prin prezentarea unui nou tip de sistem.

Dealungul timpului au fost concepute numeroase abordari ale gestionarii traficului. Noi o să prezentăm unele dintre cele mai recente abordări care au dovenit să aducă îmbunătățiri asupra traficului:

- **Sisteme bazate pe detectia de obiecte**

- Sisteme bazate pe senzori
- Sisteme care sincronizează traficul
- Sisteme bazate pe logica fuzzy
- Sisteme bazate pe DSRC

Privire de ansamblu asupra tehnologiei dezvoltate

Dealungul timpului au fost concepute numeroase abordari ale gestionarii traficului. Noi o să prezentăm unele dintre cele mai recente abordări care au dovenit să aducă îmbunătățiri asupra traficului:

- Sisteme bazate pe detectia de obiecte
- **Sisteme bazate pe senzori**
- Sisteme care sincronizează traficul
- Sisteme bazate pe logica fuzzy
- Sisteme bazate pe DSRC

Privire de ansamblu asupra tehnologiei dezvoltate

Dealungul timpului au fost concepute numeroase abordari ale gestionarii traficului. Noi o să prezintăm unele dintre cele mai recente abordări care au dovenit să aducă îmbunătățiri asupra traficului:

- Sisteme bazate pe detectia de obiecte
- Sisteme bazate pe senzori
- **Sisteme care sincronizează traficul**
- Sisteme bazate pe logica fuzzy
- Sisteme bazate pe DSRC

Privire de ansamblu asupra tehnologiei dezvoltate

Dealungul timpului au fost concepute numeroase abordari ale gestionarii traficului. Noi o să prezintăm unele dintre cele mai recente abordări care au dovenit să aducă îmbunătățiri asupra traficului:

- Sisteme bazate pe detectia de obiecte
- Sisteme bazate pe senzori
- Sisteme care sincronizează traficul
- **Sisteme bazate pe logica fuzzy**
- Sisteme bazate pe DSRC

Privire de ansamblu asupra tehnologiei dezvoltate

Dealungul timpului au fost concepute numeroase abordari ale gestionarii traficului. Noi o să prezentăm unele dintre cele mai recente abordări care au dovenit să aducă îmbunătățiri asupra traficului:

- Sisteme bazate pe detectia de obiecte
- Sisteme bazate pe senzori
- Sisteme care sincronizează traficul
- Sisteme bazate pe logica fuzzy
- **Sisteme bazate pe DSRC**

Sisteme bazate pe detectia de obiecte

Sistemele se bazeaza pe determinarea numarul de masini ce asteapta in trafic folosind camere (Fig 1). Metoda se bazeaza pe algoritmi de segmentare a imaginilor si detectie de obiecte.

Cu toate acestea, tehniciile folosite pentru a rezolva problema s-au dovedit a fi ineficiente în timp real, datorita complexitatii computationale a algoritmilor de procesare a imaginilor, astfel sistemul nu a putut tine pasul cu vehiculele ce se deplasau la viteze mari. De asemenea, in conditii meteo neprielnice, acuratetea acestora scade drastic.

Sisteme bazate pe detectia de obiecte

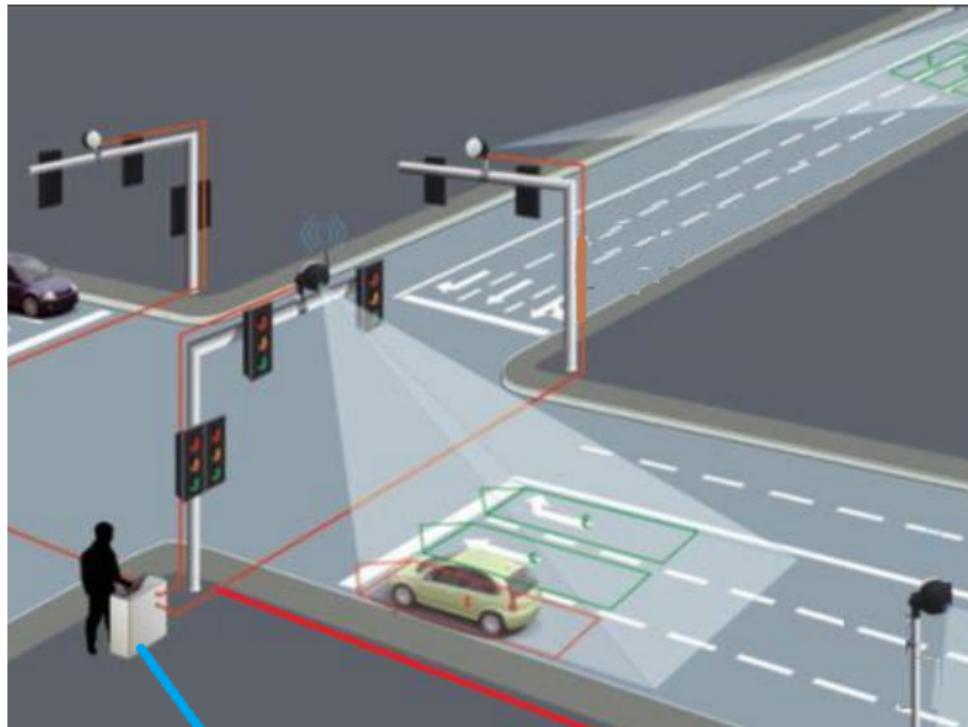


Figura: Sisteme bazate pe detectie de obiecte (Sursa imagine ©)

Sisteme bazate pe senzori

O alta modalitate de a gestiona traficul este aceea bazata pe senzori. Aceasta presupune monitorizarea sosiri si plecari vehiculelor prin intermediul datelor GPS. Astfel, am putea folosi tehnologia încorporată pentru a înregistra datele GPS și a le trimite la sistemul de monitorizare a traficului prin GSM/GPRS (Fig 2).

Dezavantajele acestei metode sunt faptul că implică costuri de implementare foarte mari, iar unele vehicule nu pot fi urmărite folosind sisteme de detectare radio. Această problemă poate fi abordată și cu ajutorul senzorilor de drum, dar ar necesita costuri chiar mai mari deoarece acestia ar trebui înlocuiti destul de des, datorita uzurii parti carosabile și a construcțiilor.

Sisteme bazate pe senzori

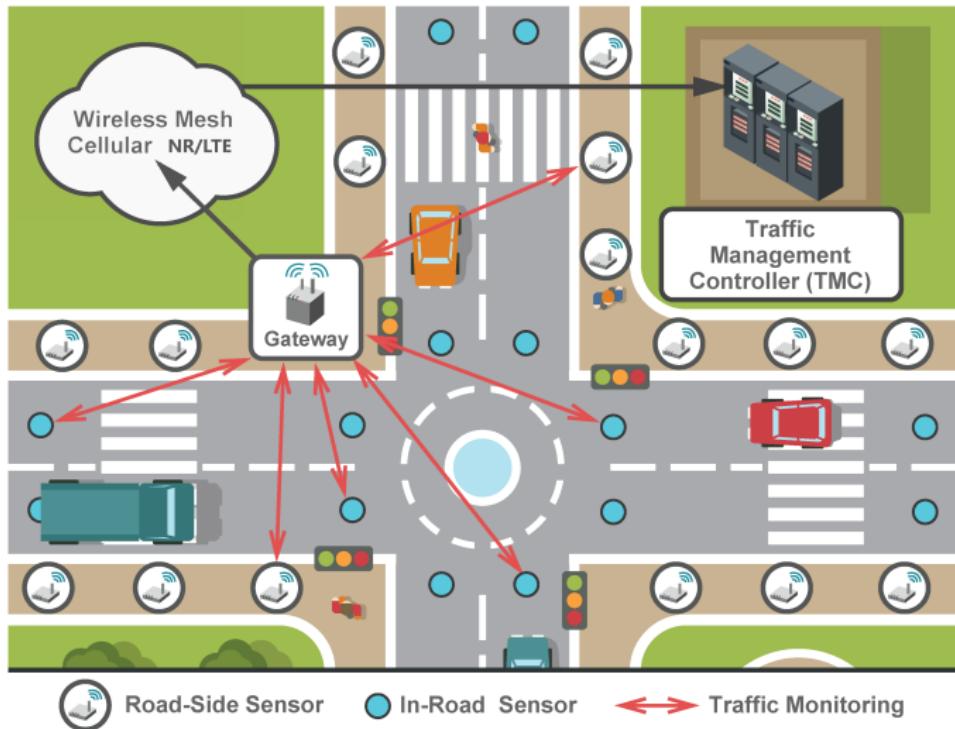


Figura: Sisteme bazate pe senzori (Sursa imagine ©)

Sisteme care sincronizeaza traficul

Sistemele de sincronizare a semafoarelor (TSC) urmăresc să minimizeze numărul de apariții STOP și GO prin adaptarea stării semafoarelor în intersecții. Această tehnică are ca și constrângere deplasarea la viteza constantă a vehiculelor, dând posibilitatea acestora să treaca prin un lanț de intersecții fără oprire. La fel ca majoritatea altor sisteme, aceasta colectează date despre traficul curent dar nu depinde de o anumită metodă specifică. Când vehiculele parcurg una dintre intersecții durata de verde crește, la fel și cea de roșu pentru drumurile adiacente, dând posibilitatea curgerii constante a traficului.

Principalele dezavantaje ale acestei metode este că în scenarii reale conducători nu vor respecta condiția de deplasare constantă în normele legale de viteza, ducând la scenarii neprevăzute, precum așteptarea prelungită a rutelor secundare. De asemenea, metoda presupune prioritizarea unei rute principale, în cazul în care două sau mai multe rute principale se intersecțează traficul va fi desincronizat.

Sisteme bazate pe logica fuzzy

Sistemele bazate pe logica fuzzy(FITS) au fost concepute initial cu intentia de a imita un politist ce gestioneaza traficul dintr-o intersecție. Acestea iau starea traficului și aplică reguli fuzzy pentru a gestiona traficul. Astfel, starea traficului va fi reprezentată de valori între 0 și 1. De exemplu, durata timpului de verde poate fi modelată pe baza setului fuzzy care include 4 stări: "none" (1), "scurt" (2), "moderat" (3), "lung" (4). În ciuda beneficiilor acestei abordări, acesta necesită o perioadă îndelungată de training, acesta facându-se real-time datorită faptului că traficul este mult prea variadic și nu poate fi prezis. De asemenea, acest training va trebui realizat pentru fiecare intersecție în parte, iar rezultatele nu coincid întotdeauna cu așteptările, traficul putând chiar fi ingreunat din cauza acestuia.

Sisteme bazate pe logica fuzzy

$$\text{"none"} - f(q) = \max(\min((10 - q)/10, 2), 0) \quad (1)$$

$$\text{"scurt"} - f(q) = \max(\min(q/10, (20 - q)/10), 0) \quad (2)$$

$$\text{"moderat"} - f(q) = \max(\min(q/5, (30 - q)/5), 0) \quad (3)$$

$$\text{"lung"} - f(q) = \max(\min((50 - q)/10, q/2), 0) \quad (4)$$

Sisteme bazate pe DSRC

DSRC sunt canale de comunicație fără fir unidirectionale sau bidirectionale special concepute pentru utilizare în automobile, care sunt utilizate în cea mai mare parte de către ITS pentru a comunica cu alte vehicule sau cu tehnologia infrastructurii. Acestea funcționează pe banda de 5,9 GHz a spectrului de frecvențe radio și sunt eficiente pe distanțe scurte și medii. Virtual Traffic Light(VTL) este o abordare inspirată din biologie a controlului traficului care se bazează pe comunicare între vehicule (V2V) print utilizarea mesajelor DSRC. Ne putem imagina mașinile ca fiind "routere în mișcare", iar intersecțiile fiind "routere staționare". Ori de câte ori o cale de transport este supraincarcată, una dintre rutile alternative este blocată, astfel abordarea seamănă mult cu tehniciile provenite din retelele de date.

Principalul dezavantaj este faptul că în momentul de față sunt multe vehicule care nu suportă acest tip de tehnologie. Astfel infrastructura traficului nu permite lansarea sistemului.

Sisteme bazate pe DSRC



Figura: Sisteme bazate pe DSRC (Sursa imagine ©)

Motivatie, scopuri si obiective

Ce ne-a determinat pe noi sa concepem acest nou tip de sistem este experienta in traficul din Timisoara, Romania. Credem ca sistemul de trafic de aici se bazeaza pe sisteme de sincronizare a semafoarelor, deoarece de cele mai multe ori, atunci cand puteti prinde semaforul verde la un anumita intersecție, le veti prinde si pe restul ce urmeaza. Ce este problematic si credem ca este problema principala a soferilor sunt orele de varf, cand traficul nu este gestionat bine si de multe ori va trebui sa astepti la acelasi semafor pana la 3 sau ba chiar 4 cicluri de rosu si verde. Problema prezentata este datorita faptului ca din cauza volumului mare de trafic, masinile nu vor mai putea circula cu o viteza constanta iar traficul in sine va fi desincronizat. Aceasta este doar un exemplu de trafic gestionat prost, dar aceasta problema persista la nivel global, netinand cont de tipul de sistem folosit, mereu vom ajunge la ambuteiaje. Principalele noastre obiective sunt crearea unui sistem:

- **Performant si accesibil**
- Adaptabil la orice conditie de trafic
- Scalabil la nivel global

Motivatie, scopuri si obiective

Ce ne-a determinat pe noi sa concepem acest nou tip de sistem este experienta in traficul din Timisoara, Romania. Credem ca sistemul de trafic de aici se bazeaza pe sisteme de sincronizare a semafoarelor, deoarece de cele mai multe ori, atunci cand puteti prinde semaforul verde la un anumita intersecție, le veti prinde si pe restul ce urmeaza. Ce este problematic si credem ca este problema principala a soferilor sunt orele de varf, cand traficul nu este gestionat bine si de multe ori va trebui sa astepti la acelasi semafor pana la 3 sau ba chiar 4 cicluri de rosu si verde. Problema prezentata este datorita faptului ca din cauza volumului mare de trafic, masinile nu vor mai putea circula cu o viteza constanta iar traficul in sine va fi desincronizat. Aceasta este doar un exemplu de trafic gestionat prost, dar aceasta problema persista la nivel global, netinand cont de tipul de sistem folosit, mereu vom ajunge la ambuteiaje. Principalele noastre obiective sunt crearea unui sistem:

- Performant si accesibil
- **Adaptabil la orice conditie de trafic**
- Scalabil la nivel global

Motivatie, scopuri si obiective

Ce ne-a determinat pe noi sa concepem acest nou tip de sistem este experienta in traficul din Timisoara, Romania. Credem ca sistemul de trafic de aici se bazeaza pe sisteme de sincronizare a semafoarelor, deoarece de cele mai multe ori, atunci cand puteti prinde semaforul verde la un anumita intersecție, le veti prinde si pe restul ce urmeaza. Ce este problematic si credem ca este problema principala a soferilor sunt orele de varf, cand traficul nu este gestionat bine si de multe ori va trebui sa astepti la acelasi semafor pana la 3 sau ba chiar 4 cicluri de rosu si verde. Problema prezentata este datorita faptului ca din cauza volumului mare de trafic, masinile nu vor mai putea circula cu o viteza constanta iar traficul in sine va fi desincronizat. Aceasta este doar un exemplu de trafic gestionat prost, dar aceasta problema persista la nivel global, netinand cont de tipul de sistem folosit, mereu vom ajunge la ambuteiaje. Principalele noastre obiective sunt crearea unui sistem:

- Performant si accesibil
- Adaptabil la orice conditie de trafic
- **Scalabil la nivel global**

O noua varianta flexibila si economica de a gestiona traficul

Credem că viitorul gestionari traficului se va baza pe semnale asemanatoare cu cele DSRC. Deocamdată, infrastructura nu oferă o modalitate de a implementa acest tip de sisteme, asa ca am decis să dezvoltăm o nouă abordare care să combină și să preia cele mai bune caracteristici din toate tehnologiile descrise mai sus și să ofere o migrare ușoară. Pentru asta am conceput un sistem IPC alcătuit din 2 tipuri de servere și 2 tipuri de clienti:

- **Servere:**

- ① Junction Main Server(JMS) - legat de intersecție și semafoare
- ② Proxy - servere regionale ce asigură conectivitatea

- **Clienti:**

- Traffic Observer(TO) - legat de către o cameră pe fiecare direcție
- Vehicle Tracker(VT) - instalat direct pe autovehicul

O noua varianta flexibila si economica de a gestiona traficul

Credem că viitorul gestionari traficului se va baza pe semnale asemanatoare cu cele DSRC. Deocamdată, infrastructura nu oferă o modalitate de a implementa acest tip de sisteme, asa ca am decis să dezvoltăm o nouă abordare care să combină și să preia cele mai bune caracteristici din toate tehnologiile descrise mai sus și să ofere o migrare ușoară. Pentru asta am conceput un sistem IPC alcătuit din 2 tipuri de servere și 2 tipuri de clienti:

- Servere:

- ① Junction Main Server(JMS) - legat de intersecție și semafoare
- ② Proxy - servere regionale ce asigură conectivitatea

- Clienti:

- Traffic Observer(TO) - legat de către o cameră pe fiecare direcție
- Vehicle Tracker(VT) - instalat direct pe autovehicul

O noua varianta flexibila si economica de a gestiona traficul

Credem că viitorul gestionari traficului se va baza pe semnale asemanatoare cu cele DSRC. Deocamdată, infrastructura nu oferă o modalitate de a implementa acest tip de sisteme, asa ca am decis să dezvoltăm o nouă abordare care să combină și să preia cele mai bune caracteristici din toate tehnologiile descrise mai sus și să ofere o migrare ușoară. Pentru asta am conceput un sistem IPC alcătuit din 2 tipuri de servere și 2 tipuri de clienti:

- Servere:

- ① Junction Main Server(JMS) - legat de intersecție și semafoare
- ② Proxy - servere regionale ce asigură conectivitatea

- Clienti:

- Traffic Observer(TO) - legat de către o cameră pe fiecare direcție
- Vehicle Tracker(VT) - instalat direct pe autovehicul

O noua varianta flexibila si economica de a gestiona traficul

Credem că viitorul gestionari traficului se va baza pe semnale asemanatoare cu cele DSRC. Deocamdată, infrastructura nu oferă o modalitate de a implementa acest tip de sisteme, asa ca am decis să dezvoltăm o nouă abordare care să combină și să preia cele mai bune caracteristici din toate tehnologiile descrise mai sus și să ofere o migrare ușoară. Pentru asta am conceput un sistem IPC alcătuit din 2 tipuri de servere și 2 tipuri de clienti:

- Servere:

- ① Junction Main Server(JMS) - legat de intersecție și semafoare
- ② Proxy - servere regionale ce asigură conectivitatea

- Clienti:

- ① Traffic Observer(TO) - legat de către o cameră pe fiecare direcție
- ② Vehicle Tracker(VT) - instalat direct pe autovehicul

O noua varianta flexibila si economica de a gestiona traficul

Credem că viitorul gestionari traficului se va baza pe semnale asemanatoare cu cele DSRC. Deocamdată, infrastructura nu oferă o modalitate de a implementa acest tip de sisteme, asa ca am decis să dezvoltăm o nouă abordare care să combină și să preia cele mai bune caracteristici din toate tehnologiile descrise mai sus și să ofere o migrare ușoară. Pentru asta am conceput un sistem IPC alcătuit din 2 tipuri de servere și 2 tipuri de clienti:

- Servere:
 - ① Junction Main Server(JMS) - legat de intersecție și semafoare
 - ② Proxy - servere regionale ce asigură conectivitatea
- Clienti:
 - ① Traffic Observer(TO) - legat de către o cameră pe fiecare direcție
 - ② Vehicle Tracker(VT) - instalat direct pe autovehicul

O noua varianta flexibila si economica de a gestiona traficul

Credem că viitorul gestionari traficului se va baza pe semnale asemanatoare cu cele DSRC. Deocamdată, infrastructura nu oferă o modalitate de a implementa acest tip de sisteme, asa ca am decis să dezvoltăm o nouă abordare care să combină și să preia cele mai bune caracteristici din toate tehnologiile descrise mai sus și să ofere o migrare ușoară. Pentru asta am conceput un sistem IPC alcătuit din 2 tipuri de servere și 2 tipuri de clienti:

- Servere:
 - ① Junction Main Server(JMS) - legat de intersecție și semafoare
 - ② Proxy - servere regionale ce asigură conectivitatea
- Clienti:
 - ① Traffic Observer(TO) - legat de către o cameră pe fiecare direcție
 - ② **Vehicle Tracker(VT) - instalat direct pe autovehicul**

Vehicle tracker

Clientul respectiv se va folosi de componentele deja instalate pe o masina anume: GPS si Radio. Astfel acesta va determina pozitia geografica atat cat si directia de mers si prin intermediul Proxy-ului se va conecta la urmatoarea intersectie, semnaland intentia de traversare. Odata ce clientul respectiv a trecut de intersectie, el se va deconecta si va interoga ultimul proxy cunoscut. Astfel, vom avea o metoda rapida si neperturbabila de a transmite date catre semafor, putand determina usor starea traficului in orice moment.

Prima problema care trebuie rezolvata este cum anume se va determina care este urmatoarea intersectie la care trebuie sa se conecteze clientul. Pentru a rezolva aceasta dilema am conceput un tip de server care actioneaza ca si un proxy. El va fii amplasat la nivel regional si are rolul de a redirectiona clienti instalati pe vehicule catre serverul instalat pe intersectia corespunzatoare. Fiecare proxy are propriul sau set de date ce includ intersectiile acoperite atat si o lista de noduri cu proxy-urile interconectate. Acest lucru ne permite scalarea sistemului pe orizontala, astfel sistemul capatand posibilitatea de a fi lansat la nivel global.

Traffic observer

Deoarece vrem ca sistemul nostru sa poata fi folosit inainte ca o noua infrastructura pe baza de DSRC sa fie dezvoltata, am fost nevoiti sa adaugam o solutie alternativa pentru determinarea starii traficului. Astfel am decis sa cream un nou tip de client, ce se bazeaza pe detectia de obiecte, anume un client ce poate detecta masinile. Acesta este amplasat pe fiecare drum ce trece prin intersectie si necesita prezenta unei camere. Odata instalat, el va actiona asemanator cu sistemele bazate pe detectie de obiecte descrise mai sus. Aceasta se poate dezactiva oricand, avand ca si scop doar permiterea unei tranzitii mai usoare catre sistemele DSRC odata ce avem infrastructura necesara.

JMS sunt serverele principale, amplasate si legate direct la semafoare ce au rolul de a gestiona traficul.

În ceea ce privește algoritmul principal pentru managementul traficului, avem 8 faze ale traficului (Fig. 4). Scenariul cel mai simplu al traficului este acela in care cele 8 faze ale traficului vor fi parcurse intr-un mod circular pornind de la starea 1. Acest ciclu reprezinta modul in care se desfasoara traficul in momentul de fata, dar noi vrem sa il depasim, sa putem sa sarim de la o stare a traficului la alta "aleator" in orice moment.

Junction main server

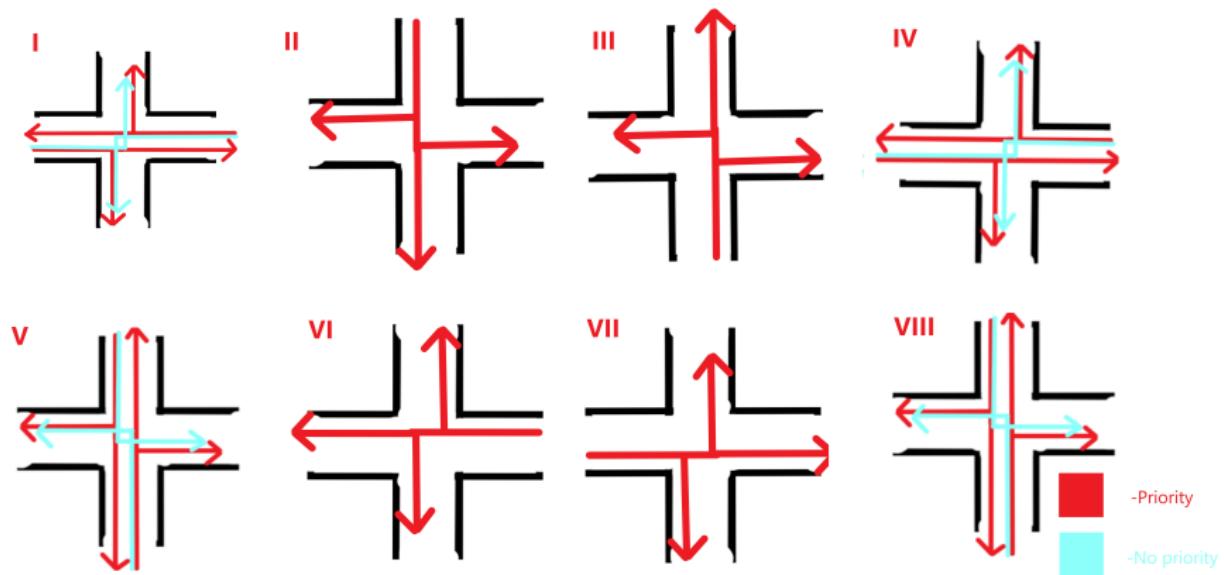


Figura: Fazele traficului

Pentru a determina urmatoare starea de trafic o sa avem cronometre asociate fiecarei punct cardinal (E, W, N si S). In prima faza cronometrele vor fi setate la o anumita valoarea, dar timpul de asteptare va scadea dinamic, odata cu aparitia vehiculelor. De asemenea vom avea un timer asociat si semnalului verde al semaforului. Odata ce acesta expira, ne vom calcula urmatoarea stare a traficului. In caz ca niciun timer nu a expirat, vom urma flow-ul normal al traficului, in caz contrar vom sari la starea corespunzatoare (Fig. 5). Putem insa avea si conflicte, de exemplu nu putem lasa vehicule sa traverseze atat din N cat si din E. Aceste scenarii le-am numit "scenarii de conflict" si dorim sa le evitam pe cat posibil. In cazul nefericit al aparitiei unui astfel de eveniment, am decis sa tratam scenariul folosind regula FIFO(Fig. 6).

Junction main server

CURRENT STATE:

TIMER_E: 120 sec

TIMER_W: 120 sec

TIMER_N: 120 sec

TIMER_S: 120 sec

END GREEN
LIGHT PHAZE

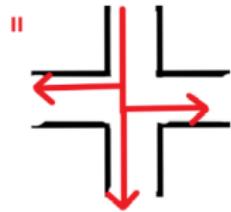
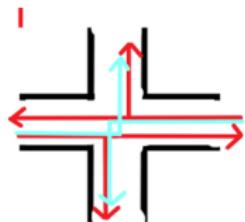
CURRENT STATE:

TIMER_E: 120 sec

TIMER_W: 120 sec

TIMER_N: 10 sec

TIMER_S: 5 sec



NO TIMER EXPIRED CONTINUE NORMAL FLOW

CURRENT STATE:

TIMER_E: 40 sec

TIMER_W: 10 sec

TIMER_N: 120 sec

TIMER_S: 0 sec

S TIMER EXPIRED SO WE MOVED
TO CORRESPONDING PHAZE: PHAZE III

END GREEN
LIGHT PHAZE

CURRENT STATE:

TIMER_E: 0 sec

TIMER_W: 0 sec

TIMER_N: 20 sec

TIMER_S: 120 sec

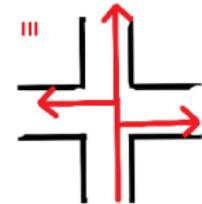


Figura: Exemplu flow al traficului

Junction main server

CURRENT STATE:

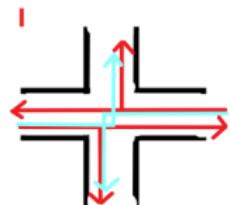
TIMER_E: 120 sec

TIMER_W: 120 sec

TIMER_N: 120 sec

TIMER_S: 120 sec

END GREEN
LIGHT PHASE



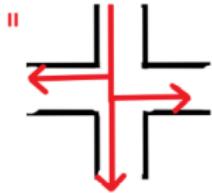
CURRENT STATE:

TIMER_E: 120 sec

TIMER_W: 120 sec

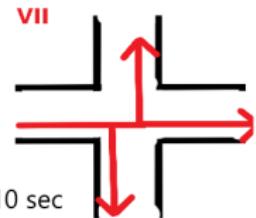
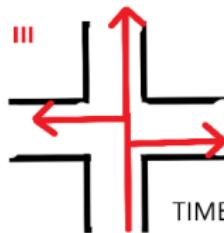
TIMER_N: 10 sec

TIMER_S: 5 sec



NO TIMER EXPIRED CONTINUE NORMAL FLOW

POSSIBLE STATES



TIMER_E: 10 sec
TIMER_W: 0 sec
TIMER_N: 120 sec
TIMER_S: 0 sec

WE HAVE BOTH TIMER FOR E AND S EXPIRED
BUT NO CORRESPONDING PHASE.

Figura: Scenariu de conflict

Detalii de implementare

Întregul sistem a fost dezvoltat folosind C++17, Python, Boost, OpenSSL, GLFW, MSVC WinAPI, OpenCV, Tensorflow, YoloV8 și MySQL. Sistemul în sine este tratat ca un proiect mare și împărțit în mai multe submodule:

- librari
 - Common
 - CarDetector
 - IPC
 - GUIGLFW
- executable
 - servere
 - Proxy
 - JunctionMainServer
 - ObjectDetectionServer
 - clienti
 - VehicleTracker
 - TrafficObserver
 - mediu de testare

Ideea principală a submodulului Common este să acționeze ca o bibliotecă ajutatoare. Oferă soluții la următoarele probleme frecvent întâlnite precum:

- Probleme de concurenta
- Parsarea datele produse de GPS
- Interactiuni cu baza de date
- Parsarea fisierelor de configurare
- Crearea unor actiuni planificate
- Logarea sincronă în scenarii multifir de execuție

Pentru a stabili comunicarea între toate executabilele, un sistem IPC nou a fost dezvoltat, ce se foloseste de Boost Asio pentru operatiile cu socket-uri. Logica în sine rulează pe 4 fire de executie: unul pentru menținerea contextului în funcțiune, unul pentru citire, unul pentru scriere și unul pentru notificarea progresului.

Clientul are 4 metode principale, toate fiind rulate sincron: "connect", "disconnect", "send", "waitForAnswer". De fiecare dată când a se stabilește o nouă conexiune cu serverul, un nou obiect "Connexion" va fi creat care va fi responsabil de citire, scriere și furnizarea de mesaje. "Connexion" este responsabilă pentru citirea și scrierea mesajelor. Acționează ca un punct de mijloc între client și server. Aceasta are ca și obiect să partajeze o coadă de mesaje primite, astfel încât să poată notifica proprietarul că a sosit un nou mesaj.

Serverul rulează pe 2 fire de execuție, unul pentru gestionarea operațiilor boost asio și unul pentru procesarea mesajelor primite. Are o listă de clienți deja conectați, iar conexiunea în sine este singura operație care este efectuată în mod asincron. Are 3 metode principale care pot fi suprascrisе și care definesc comportamentul efectiv al serverului: "onClientConnect", "onClientDisconnect" și "onMessage", care sunt callback-uri către evenimente IPC. Dacă, de exemplu, un integrator ar dori să utilizeze framework-ul nostru IPC, tot ce ar trebui să facă ar fi să suprascrăste aceste callback-uri furnizate.

Pentru a defini structura mesajelor, am creat și o clasă specifică. Antetul mesajului conține dimensiunea acestuia, dacă are prioritate (este de la un vehicul de urgență) și tipul mesajului care este trimis, precum și un ID. Datele salvate în corpul mesajului sunt în format de octeți iar adăugarea/extragerea de date funcționează ca și o stivă.

Pentru a putea vizualiza datele reale, am creat un modul GUI. Acesta este doar în scopuri de demonstrație/testare și poate fi dezactivat folosind un flag. Pentru a realiza acest lucru, am utilizat GLFW, mai exact am pre-desenat intersecția și am legat-o de cronometrele intersectiei. În ceea ce privește mașinile, am avut o coadă de acțiuni în așteptare și ori de câte ori o mașină era detectată de JMS, acțiunea de desenare era pusă în coadă. Acțiunea în sine poate de asemenea adăuga la randul ei o altă acțiune, dacă mașina nu a depășit încă fereastra. Deoarece dorim ca scena să acționeze ca o intersecție, atunci când suntem pe punctul de a traversa intersecția în timpul desenării, am întârziat-o atât timp cat lumina roșie persistă (Fig 7).

GUIGLFW

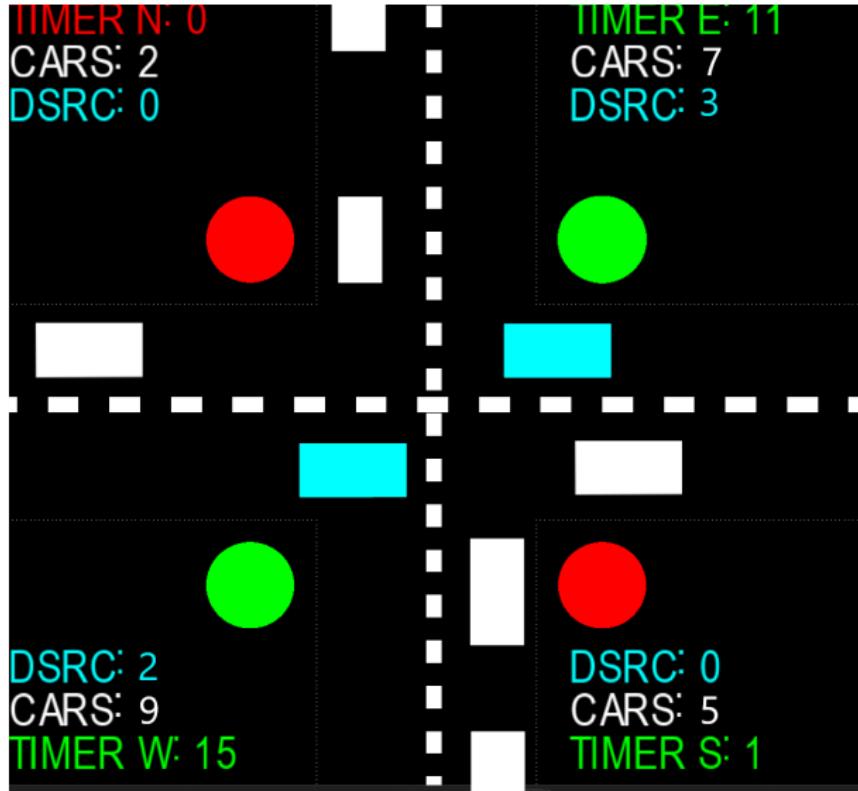


Figura: GUI IMS
Traffic Manager

Car Detector

Pentru a putea conecta la camerele de trafic rutier și pentru a determina numărul de vehicule care intră/ies, am creat w rappere peste biblioteca OpenCV. Întregul proces funcționează în felul următor: începe prin detectarea obiectelor în mișcare într-un cadru dat, încearcă să prezică poziția lor următoare și odată ce acestea traversează linii imaginare, acestea sunt înregistrate dacă s-au dovedit a fi mașini. Pentru a îmbunătăți şansele de înregistrare efectivă a trecerii unei mașini, am considerat mașinile ca obiecte punctiforme, luând în considerare doar punctul central al dreptunghiului delimitator ca fiind mașini reale. Clasificarea mașinilor în sine nu este realizată de modulul dat datorită incompatibilității actuale dintre Tensorflow și OpenCV în C++, ci de către ObjectDetectionServer scris în Python. Modulul CarDetector acționează doar ca un client și trimite octetii reali ai imaginii decupate ce reprezinta obiectele în mișcare către server și așteaptă un răspuns.

Object Detection Server

Pentru a putea detecta dacă sunt prezente mașini, am încercat să utilizăm două arhitecturi diferite: una bazată pe YOLOv8 și una utilizând API-ul de detecție a obiectelor TensorFlow, în timp ce am folosit același set de date pentru antrenament. În urma unei analize indelungate, ce poate fi regasită în lucrare, am determinat că modelul antrenat folosind Tensorflow este mult mai potrivit pentru scenariul nostru. Am folosit SSD MobileNet, un tip de rețea neuronala convolutionară ce a fost preantrenată și este folosită deseori pe sisteme cu resurse limitate. Acuratetea medie a modelului antrenat de noi este de 60%, iar acesta este încărcat de către un server simplist scris în Python.

Proxy-ul este un wrapper peste implementarea serverului care asigură conexiunea între VT și JMS. Fiecare proxy este conectat la propria sa bază de date, care conține o listă de intersecții, proxy-uri și zona acoperită de acestea. Modul de funcționare este destul de simplu: dacă un client interoghează proxy-ul pentru următoarea intersecție, acesta caută în baza de date. Dacă găsește o intersecție potrivită, trimit datele de conexiune, în caz contrar redirecționează vehiculul către proxy-ul cel mai apropiat. Acest proces se repetă până când ajunge în cele din urmă la o intersecție validă sau mașina ieșe din zona de acoperire (de exemplu, dacă mașina se află undeva pe o navă în mijlocul mării, nu există niciun motiv să o conectăm la o intersecție).

Proxy

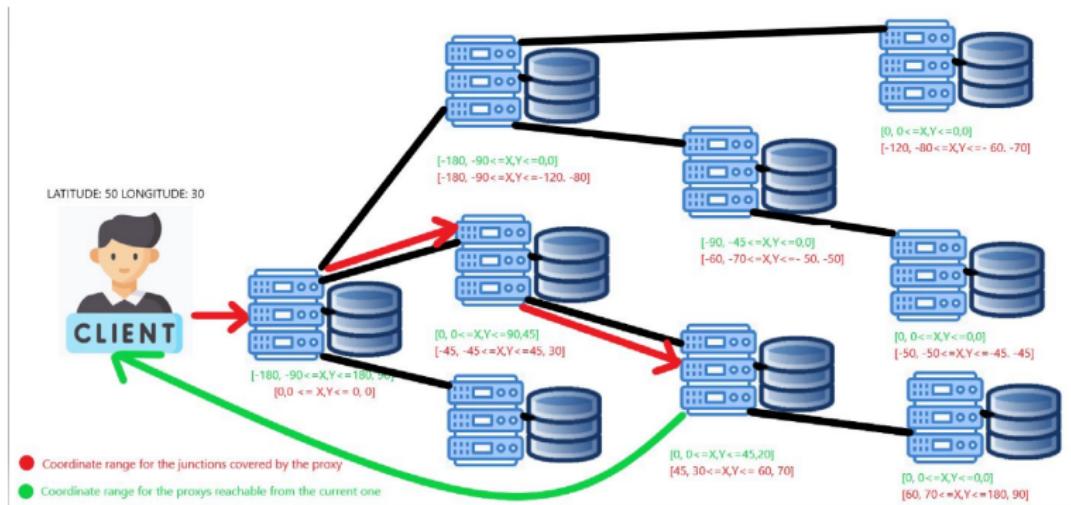


Figura: Flow-ul de interogare a proxy-urilor

Executabilul este un wrapper peste implementarea serverului si gestionează stările traficului. Acesta urmărește vehiculele care urmeaza sa treaca prin intersectie și cronometrează fiecare stare de trafic. Implementarea în sine este concepută ca un state machine, în care stările sunt reprezentate de stările traficului, iar evenimentele sunt reprezentate de expirarea cronometrelor. Fiecare stare are asociat un cronometru care scade în mod normal sau când este detectat un vehicul care urmeaza sa treaca prin intersectie. Ori de câte ori ne aflăm într-o anumită stare, cronometrul corespunzător este oprit și apoi resetat, iar durata acestuia este recalculată in functie de trafic. Modul in care update-ul cronometrelor influenteaza traficul se poate vedea in figura 9

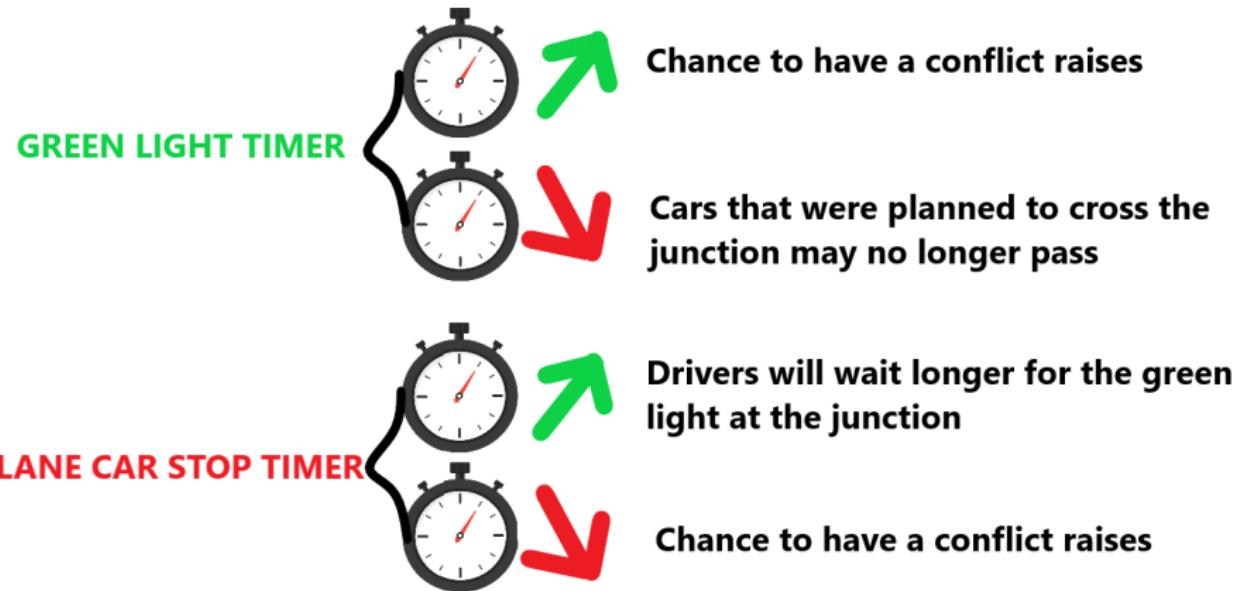


Figura: Impactul cresterii/scaderii duratii cronometrului

Principiul de baza pe care incercam sa il obtinem este un timp minim de asteptare fara a intra in stari de conflict. Pentru acest lucru am retinut o medie a vehiculelor ce au trecut de intersectie in timpul lumini verzi a semaforului si daca suntem sau nu in stare de conflict. Din cauza complexitatii ridicate a algoritmilor de detectie a obiectelor, pentru a numara vehiculele ce au trecut de semafor, ne vom baza doar pe mesajele primite de la VTs. Astfel avem 4 scenarii posibile iar modul de tratare se poate vedea in figura 10



Figura: Faulty Scenarios Handling

TrafficObserver

Executabilul este un wrapper peste client ce incarca functionalitatile modulului de detectie de masini (Fig 11) si trimite periodic mesaje la server. De asemenea, din cauza volumului mare de mesaje trimise si potentialelor exploit-uri, am adaugat o parola de conectare la server si un schimb de chei publice, pentru a impiedica eventualele atacuri cibernetice.

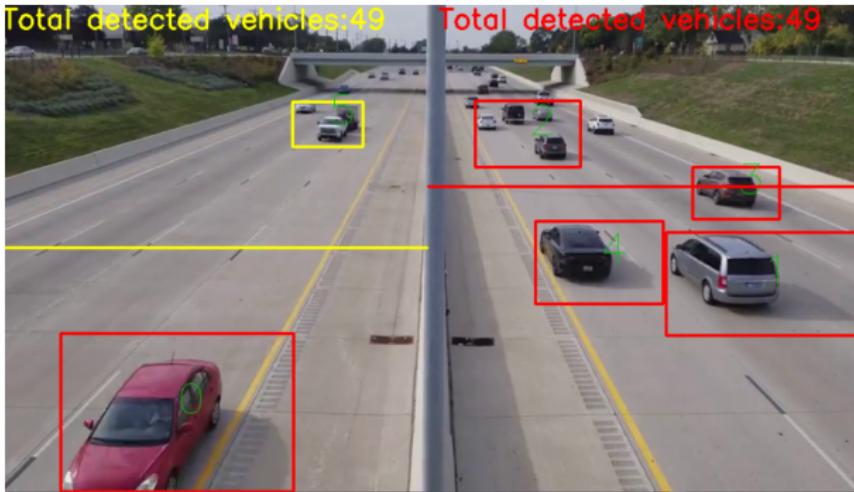


Figura: Exemplu detectie masini

Executabilul este un wrapper peste client. La pornire acesta incepe sa parseze datele geografice dintr-un stream de date (in momentul testari dintr-un fisier). Astfel, din formatul NMEA al datelor, anume din mesajele de tip GPGLL, GPGGA si GPRM se extrag coordonatele geografice curente. Pentru a afla de asemnea directia acesta citeste doua coordonate geografice distincte consecutive si determina aproximativ directia predominanta. De exemplu daca vehiculul merge spre NE dar acesta se deplaseaza mai mult spre N, directia acestuia va fi N. Odata ce am obtinut coordonatele geografice cat si directia, acestea vor fi trimise la proxim pentru a determina urmatoare intersectie. Clientul se va conecta la server-ul corespunzator, va astepta pana cand acesta va trece de intersectie si va relua procesul de la capat.

Mediu de testare

Pentru a putea testa întregul nostru sistem, am creat un executabil care să emuleze condițiile de trafic și să ruleze sistemul nostru. Executabilul primește mai multe fișiere de configurare ca intrare și poate rula simultan mai multe servere (JMS/Proxy-uri) și mai mulți clienți de orice tip, în orice fel de combinație. Pentru a emula mișcarea clientilor instalati pe vehicule, am generat date GPS cu ajutorul NMEA GEN, iar pentru a simula inputul camerelor am colectat mai multe videoclipuri.

Concluzii, defecte ale sistemului si directii viitoare

În această teză, prin analiza cercetărilor existente, exemplelor de implementare și tendințelor emergente, am obținut idei valoroase cu privire la eficacitatea și potențialul sistemelor de gestionare a traficului. Cu datele colectate, am oferit o nouă modalitate alternativă de gestionare a traficului prin definirea și implementarea unui nou sistem de trafic, pe care credem că va îmbunătăți semnificativ fluxul de trafic. Aceasta se poate adapta la condițiile de trafic, poate fi lansat la nivel global și reprezintă o punte de trecere către sistemele DSRC. De asemenea acesta este accesibil deoarece nu necesită prezența tuturor componentelor hardware și/sau software pentru o bună funcționare, iar majoritatea cerințelor sunt deja indeplinite.

Concluzii, defecte ale sistemului si directii viitoare

Unul din principalele defecte ale sistemului este faptul ca exista inca scenarii de exploit in sistem. Ca si imbunatatire, trebuie stabilita o limita maxima de incercari de conectare, atat si creat un "blacklist" pentru IP-urile userilor maliciosi.

De asemenea se poate integra un algoritm care aproximeaza zonele ocupate ale carosabilului, dar asta ar presupune o complexitate destul de ridicata si o ingreunare a sistemului, fiind discutabil daca se merita sau nu acest tradeoff de performanta pentru o mai buna securitate.

Un alt defect care se poate observa, este faptul ca, ca si orice alt sistem bazat pe detectie de obiecte, conditiile meteo ii vor afecta performanta. Ca si rezolvare la aceasta problema, se poate incerca detectia de obiecte pe baza altor tipuri de imagini, mai putin perturbabile de meteo, cum ar fi imaginile radio care au dovedit a fi un suport mai bun de antrenare a modelului, cel putin in conditii de ceata. O ultima imbunatatire ce ar trebui adusa asupra sistemului este crearea unui model nou specializat doar pe detectia de masini.