Universitatea Politehnica Timișoara

TEZĂ DE DOCTORAT

Conducător științific:
Prof. univ. dr. habil. ing. Cătălin Daniel CĂLEANU

Autor:
ing. Bogdan Ilie Sighencea

Timișoara
2023

Utilizarea rețelelor neuronale profunde în predicția deplasării participanților la traficul rutier

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea Politehnica Timișoara
în domeniul Electronica, Telecomunicații și Tehnologii Informaționale
de către

Ing. Bogdan Ilie SIGHENCEA

Președintele comisiei: Prof. univ. dr. ing. Dan LASCU
Conducător științific: Prof. univ. dr. ing. Cătălin Daniel CĂLEANU
Referenți științifici: Prof. univ. dr. Daniela ZAHARIE
Prof. univ. dr. ing. Corneliu Nicolae FLOREA
Prof. univ. dr. ing. Codruța Orniana ANCUȚI

Ziua susținerii tezei: 28 septembrie 2023.

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Inginerie Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică și Telecomunicații
8. Inginerie Industrială
9. Inginerie mecanică
10. Știința Calculatoarelor
11. Știința și Ingineria Materialelor
12. Ingineria Sistemelor
13. Inginerie Energetică
14. Calculatoare și Tehnologia Informației
15. Ingineria Materialelor
16. Inginerie și Management
17. Arhitectură
18. Inginerie Civilă și Instalații
19. Inginerie Electronică, Telecomunicații și Tehnologii Informaționale

Universitatea Politehnica Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul Şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300223 Timișoara, Bd. Vasile Pârvan 2B
Tel./fax 0256 404677
e-mail: editura@upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activității mele în cadrul Departamentului de Electronică Aplicată al Universității Politehnica Timișoara.

În primul rând, doresc să exprim mulțumiri deosebite conducătorului meu de doctorat, domnul profesor Cătălin Daniel CĂLEANU, pentru supravegherea, îndrumarea sa, suportul, efortul, meticulozitatea, sesiunile de brainstorming, răbdarea și încurajarea pe care mi le-a oferit necontenit pe toată perioada de student doctorand, și care m-au determinat să cresc gradul de cercetare. Cu toate acestea, am reușit să pun în practică noi idei și soluții în domeniul vederii artificiale.

De asemenea, doresc să mulțumesc comisiei de îndrumare formată din domnii profesori: Prof. dr. ing. Cosmin ANCUȚI, Conf. dr. ing. Georgiana SIMION și S.l. dr. ing. Radu MÎRȘU, pentru că au fost implicați în procesul de revizie a fiecărui raport de cercetare și a lucrărilor științifice pe care le-am elaborat. Totodată, comentariile dumnealor m-au ajutat să elaborez cât mai bine lucrările propuse spre publicare.

În al doilea rând, doresc să îi mulțumesc domnului Lector Ion Rareș STANCIU, pentru că a fost alături de mine în toate momentele bune dar și mai puțin bune pe care le-am avut pe toată perioada stagiului de pregătire pentru doctorat. Îi mulțumesc pentru că a împărtășit cu mine cunoștințele și experiență sa bogată, acumulată de-a lungul timpului în cariera sa.

Într-o notă personală doresc să mulțumesc familiei mele, care m-a sprijinit în fiecare moment al vieții mele și mi-au transmis o putere de muncă și concentrare incredibil de mare, având mereu încredere în mine și în decizile pe care le-am luat.

În final doresc, de asemenea, să le mulțumesc stundenților mei, cărora le-am predat în ultimii 3 ani, laboratorul Calculatoare Numerice și Electronică Digitală, pentru implicarea și relația avută cu aceștia, motivându-mă să lucrez și mai mult pentru studiu și cercetare.

Timişoara, 28 septembrie 2023 Bogdan-Ilie SIGHENCEA

Destinatarii dedicaţiei.

SIGHENCEA Bogdan Ilie

Utilizarea rețelelor neuronale profunde în predicția deplasării participanților la traficul rutier

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 2023, 103 pagini, 49 figuri, 10 tabele.

ISSN:
ISBN:

Cuvinte cheie: predicția traiectoriei pietonilor, rețele neuronale profunde, industria auto, senzori.

Rezumat:

În ultimele decenii, producătorii de automobile au lucrat în mod constant la îmbunătățirea experienței de conducere și a face vehiculele rutiere mai sigure prin dezvoltarea tehnologiilor de asistență pentru șofer. Pentru a evalua amploarea progreselor în tehnologia de asistență a șoferului, au fost definite șase niveluri de autonomie de către „Society of Automotive Engineers" (SAE).

Vehiculele autonome (VA) au potențialul de a transforma lumea așa cum o cunoaștem, revoluționând transportul, făcându-l mai rapid, mai sigur și mai puțin intensiv în lucru. Este important ca sistemele AV să perceapă cu acuratețe și să reacționeze în siguranță la diverse scenarii de conducere din lumea reală. Acest lucru necesită ca sistemele de percepție AV să înțeleagă comportamentul participanților la trafic din jur (de exemplu, vehicule, pietoni și bicicliști) și să prezică cu exactitate traiectoriile și comportamentele lor viitoare [1].

Conform ultimului raport pentru accidentele rutiere publicat de către „European Road Safety Observatory" [2], aproape 24 000 de oameni au murit în accidentele rutiere din UE în anul 2019, dintre care 4668 sunt pietoni. Din acest total, 729 de pietoni care au murit în accidentele rutiere sunt din Romania. În ceea ce privește statisticile la nivel mondial, datele sunt mai îngrijorătoare. Raportul Global privind Siguranța Rutieră [3] publicat de Organizația Mondială a Sănătății (OMS), indică faptul că peste 1,3 milioane de oameni au murit în accidente rutiere în întreaga lume în anul 2017. Aproximativ 23% dintre aceste decese este reprezentat de către pietoni.

Această teză are ca și obiectiv predicția traiectoriei pietonilor în diferite scenarii folosind rețele neuronale profunde. Acest domeniu fiind de mare succes dar și foarte studiat de către cercetători datorită dezvoltării senzorilor optici (exemplu camera RGB, Radar, LiDAR etc.) dar și apariția de noi arhitecturi de învățare profundă.

Cuprins

Notații, abrevieri, acronime

Acronim	Descriere
2D	Două dimensiuni
3D	Trei Dimensiuni
ADE	Average Displacement Error
ANN	Artificial Neural Network (rețea neuronală artificială)
AWG	Average (în medie)
BoN	Best of N (cel mai bun N)
BPTT	BackPropagation Through Time (propagarea înapoi în timp
CAGR	Compound Annual Growth Rate (rata de creștere anuală compusă)
CFAR	Constant false alarm rate
CNN	Convolutional Neural Network (rețea neuronală convoluțională)
DL	Deep Learning (învățare profundă)
DNN	Deep Neural Network (Rețea neuronală profundă)
D-STGCN	Dynamic Pedestrian Trajectory Prediction Using Spatio-Temporal Graph Convolutional Networks (predicția dinamică a traiectoriei pietonilor unilizând rețele convoluționale cu graf spațio-temporal)
EMD	Estimated Median Error (eroarea Medie de Deplasare)
ERD	Encoder Recurent Decoder
FC	Fully Connected (ful conectat)
FDE	Final Displacement Error (eroare de deplasare finală)
FMCW	Frequency-Modulated Continuous Wave (undă continuă modulată în frecvență)
GAN	Generative adversarial networks (rețele adversative generative)
GAT	Graph Attention neTworks (rețele de atenție grafică)
GCNN	Graph Convolutional Neural Networks (rețele neuronale convoluționale grafice)
GNN	Graph Neural Network (rețea neuronală de tip graf)
GRU	Gated Recurrent Unit (unitate recurentă închisă)
ITS	Intelligent Transportation System (Sisteme Inteligente de Transport)
KDE	Kernel Density Estimate (estimarea densității nucleului)
KF	Kalman Fillter (Filtrul Kalman)
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute (Institutul de Tehnologie din Karlsruhe și Institutul Tehnologic Toyota)
KMC	K-Means Clustering (K-inseamnă grupare)
KPCA	Kernel Principal Component Analysis (analiza componentelor principale ale nucleului

LiDAR	Light Detection And Ranging (detecția și variația luminii)
LSTM	Long short-term memory (memorie lungă pe termen scurt)
MANN	Memory Augmented Neural Networks (rețele neuronale augumentate cu memorie)
MATF	Multi-Agent Tensor Fusion (fuziune tensorială multi-agent)
MCTF	Multi-Camera Trajectory Forecasting (predicția traiectoriei cu mai multe camere)
MDL	Mixture Density Layer (stratul de densitate a amestecului)
MDP	Markov Decision Process (procesul decizional Markov)
MIMO	Multiple Input Multiple Output (intrări multiple ieșiri multiple)
MLP	Multi-Layer Perceptron (perceptron multistrat)
MLR	Multinomial Logistic Regression (regresie logistică multinomială)
MPI-IS	Institutul Max Planck pentru Informatică
NIR	Near infrared (infraroșu apropiat)
NLL	Negative Log-Likelihood (probabilitate logaritmică negativă)
NLP	Natural Language Processing (Procesare a Limbajului Natural)
OCDE	Organizației pentru Cooperare și Dezvoltare Economică
OMS	Organizația Mondială a Sănătății
PTP	Predestrian Trajectory Prediction (predicția traiectoriei pietonului)
PTPCNN	Predicția traiectoriei pietonilor folosind CNN
ReLU	Rectified Linear Unit (unitate liniară rectificată)
RNA	Rețea Neuronală Artificială
SAE	Society of Automotive Engineers (societatea inginerilor de automobile)
SAP	Stanford Aerial Pedestrian (vedere pietonală la Universitatea Stanford)
SDD	Stanford Drone Dataset (setul de date Stanford)
SGD	Stochastic Gradient Descent (coborârea gradientului stocastic)
SOTA	State-of-the-Art (tehnologie de ultimă generație)
STAP	Space-time adaptive processing (procesare adaptivă spațiu-timp)
ST-GCNN	Spatio-Temporal Graph Convolutional Network (Rețeaua neurală de convoluție spațio-temporală pe graf)
SVC	Surround View Camera (cameră cu vedere apropiată)
SVM	Support Vector Machine (suport mașină vectorială)
TBPTT	Truncated Backpropagation Through Time (propagare trunchiată înapoi în timp)
TCN	Temporal Convolutional Networks (Rețeaua de convoluție temporală)
TreeGNN	Rețea neuronală de tip graf transpusă multi direcție
TXP-CNN	Time-Extrapolator Convolutional Neural Network (Rețeaua neurală de convoluție cu extrapolare temporală)
UE	Uniuniea Europeană
VA	Vehicul Autonom
VIS	Visible (vizibil)
WNMF	Warwick-NTU Multi-camera Forcasting (predicția cu mai multe camere)

Lista de figuri

Lista de tabele

1. INTRODUCERE

În ultimele decenii, producătorii de automobile au lucrat în mod constant la îmbunătățirea experienței de conducere și a face vehiculele rutiere mai sigure prin dezvoltarea tehnologiilor de asistență pentru șofer. Pentru a evalua amploarea progreselor în tehnologia de asistență a șoferului, au fost definite șase niveluri de autonomie de către „Society of Automotive Engineers" (SAE). Aceste niveluri variază de la 0, care corespunde condusului complet manual, până la 5 complet autonom, care este scopul final al cercetărilor recente realizate atât de industria auto, cât și de lumea academică.

Vehiculele autonome (VA) au potențialul de a transforma lumea așa cum o cunoaștem, revoluționând transportul, făcându-l mai rapid, mai sigur și mai puțin intensiv în lucru. Este important ca sistemele AV să perceapă cu acuratețe și să reacționeze în siguranță la diverse scenarii de conducere din lumea reală. Acest lucru necesită ca sistemele de percepție AV să înțeleagă comportamentul participanților la trafic din jur (de exemplu, vehicule, pietoni și bicicliști) și să prezică cu exactitate traiectoriile și comportamentele lor viitoare [1]. Pentru predicții pe termen scurt, poate fi acceptabil să se utilizeze abordări bazate pe fizică pură. Cu toate acestea, deoarece scenariile viitoare sunt necunoscute, un sistem de predicție pe termen lung este esențial pentru a permite nu numai modelarea interacțiunii între diferiți agenți, ci și pentru a identifica regiunile traversabile definite de traseele rutiere și de conformitatea cu regulile de circulație.

Conform ultimului raport pentru accidentele rutiere publicat de către „European Road Safety Observatory" [2], aproape 24 000 de oameni au murit în accidentele rutiere din UE în anul 2019, dintre care 4668 sunt pietoni. Din acest total, 729 de pietoni care au murit în accidentele rutiere sunt din Romania. În ceea ce privește statisticile la nivel mondial, datele sunt mai îngrijorătoare. Raportul Global privind Siguranța Rutieră [3] publicat de Organizația Mondială a Sănătății (OMS), indică faptul că peste 1,3 milioane de oameni au murit în accidente rutiere în întreaga lume în anul 2017. Aproximativ 23% dintre aceste decese este reprezentat de către pietoni. Pe de altă parte, în țările Organizației pentru Cooperare și Dezvoltare Economică (OCDE), peste 20 000 de pietoni își pierd viața anual. Decesele pietonilor reprezintă între 8% și 37% din totalul deceselor din trafic, în funcție de țară și an [4]. Cele mai multe dintre aceste accidente tragice au loc în zone aglomerate de la trecerile de pietoni, cu vizibilitate redusă din cauza atenției scăzute a șoferului și/sau a oboselii.

1.1. Motivație

Potrivit [5], numărul accidentelor cu pietoni în vârstă este influențat de factori multipli din mediul localităților. Ca rezultat, reducerea (sau eliminarea) acestor coliziuni este o preocupare importantă de siguranță. În aceste situații, a ajuta șoferul include prezicerea comportamentului pietonilor. Acest lucru ajută la reducerea efectului diverșilor factori care ar putea afecta negativ siguranța în trafic (cum ar fi oboseala, vizibilitatea slabă, distracția cognitivă accidentală etc.).

Într-un eventual impact, pietonii nu au practic nici-o protecție. Prin urmare, reducerea (eliminarea) acestor impacturi este o problemă cu importantă mare. Ajutarea șoferului în astfel de condiții include prezicerea traiectoriei și/sau comportamentul pietonului și atenuarea erorilor consecvente ale șoferului (de exemplu, oboseală, gândirea cognitivă) și include dezvoltarea de noi tehnologii pentru a reduce numărul de accidente (cu până la 93,5%, conform [6]).

Noi, ca oameni, luăm decizii intuitive importante bazate pe secvențele de acțiuni și interacțiuni cu alte persoane din scenă pentru a obține o navigare sigură și lină. Această intuiție permite mișcări care sunt foarte dinamice, deoarece putem decide ce traseu să luăm într-o manieră foarte dinamică. Această informație simplă, dar valoroasă, este crucială pentru a decide următorul pas care trebuie făcut. Cu toate acestea, odată cu apariția Deep Learning, sunt dezvoltați algoritmi avansați care citesc instinctele pietonilor și permit acționarea. Sunt explorate diferite metode, de la predicția traiectoriei la analiza comportamentală [7]. În același timp, sunt examinate și diferite modalități de introducere de la imagini la date din norul de puncte (Figura 1.1.1). În această teză de doctorat, investigăm ideea utilizării imaginilor monocular RGB ca informații de bază pentru a prezice traiectoria pietonilor.

Figura 1.1.1. Analiza mediului exterior prin detectarea, clasificarea, urmărirea și predicția traiectoriei a agenților participanți la trafic (pietoni, bicicliști și vehicule). Sursa: selfdrivingcars.mit.edu.

Motivația soluțiilor investigate în teză se bazează pe observația că astăzi soluțiile ce folosesc rețele neuronale profunde furnizează rezultatele de top în diverse sarcini de procesare a semnalelor (text, voce, imagini, video etc.) inclusiv în sarcinile de predicție și percepție din domeniul automotive.

1.2. Obiective

Această teză are ca și obiectiv predicția traiectoriei pietonilor în diferite scenarii folosind rețele neuronale profunde. Acest domeniu fiind de mare succes dar și foarte studiat de către cercetători datorită dezvoltării senzorilor optici (exemplu camera RGB, Radar, LiDAR etc.) dar și apariția de noi arhitecturi deep learning. Pentru atingerea obiectivului propus, au fost definite mai multe sarcini de lucru:

- Analiza stadiului curent în domeniu predicției traiectoriei a pietonilor, precum și analiza tuturor arhitecturilor existente folosind rețele neuronale profunde.
- Identificarea și implementarea de noi soluții deep learning (hibrid) care aduc îmbunătățiri substanțiale la soluțiile existente.
- Aplicarea modelelor dezvoltate asupra celor mai cunoscute baze de date pentru acest domeniu, și depășirea soluțiilor state-of-the-art la nivel de performanță și acuratețe.
- Testarea fezabilității și limitele metodelor propuse într-un mod extins în condiții ideale, utilizând baze de date din lumea reală.
- Măsurarea influenței modelării a patru dinamici pietonale diferite, adică statul pe loc, pornirea, oprirea și mersul pe jos. Aceste dinamici permit definirea adecvată a schimbărilor efectuate de pietoni în scenarii reale.
- Dezvoltarea unei metode de predicție a traseul pietonilor, aplicând noi modele neuronale (ex. CNN, LSTM și GNN).

1.3. Structura lucrării

Având în vedere obiectivele propuse, teza este structurată în 7 capitole prezentate succint în cele ce urmează. Capitolul 2 se referă la elemente de inteligență artificială cu accent pe conceptul de rețea neuronală artificială. Sunt prezentate concepte legate de arhitectura și modalitatea de instruire aferente rețelelor neuronale profunde, în special modele folosite în problema predicției traiectoriei pietonilor (PTP). Analiza stadiului actual (state-of-the-art) al realizărilor aferente PTP este efectuată pe parcursul Capitolului 3.

Rezolvarea cu succes a problemei PTP depinde de informația senzorială disponibilă sistemului de predicție. Din acest motiv a fost ales să se analizeze caracteristicile și performanțele diverselor tipuri de senzori în cadrul Capitolului 4 al tezei. Soluțiile propuse sunt prezentate împreună cu rezultatele experimentale aferente în cadrul capitolelor 5 respectiv 6. Lucrarea se încheie cu un capitol destinat concluziilor și posibilelor direcții de dezvoltare ulterioară. De asemenea lucrarea conține o secțiune de referințe bibliografice. În figura de mai jos se poate urmării întreaga structură a lucrării împărțită pe capitole și subcapitole.

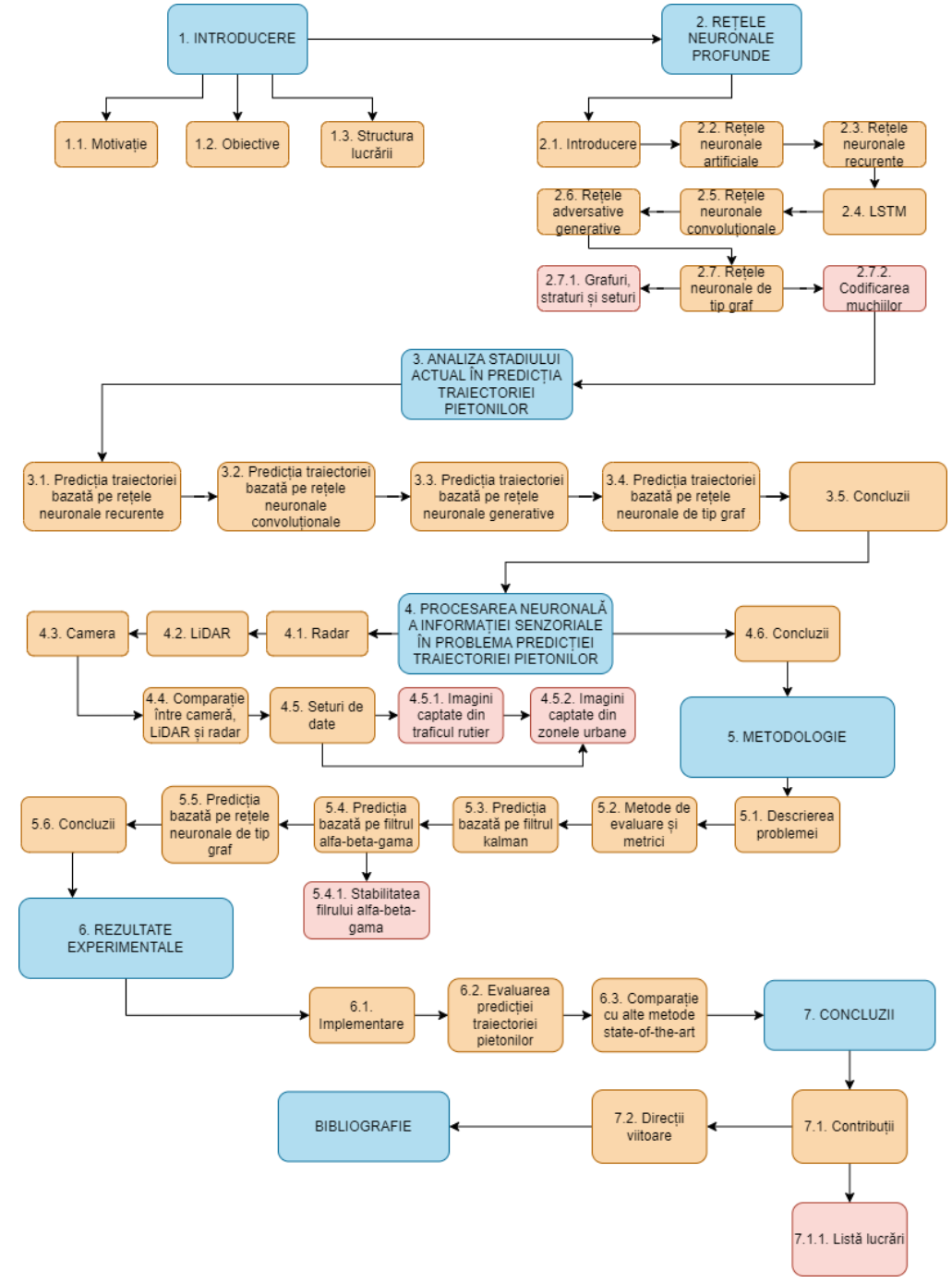

Figura 1.3.1. Structura lucrării pe subcapitole.

2. REȚELE NEURONALE PROFUNDE

2.1. Introducere

Pentru a rezolva problema PTP, în ultimii ani, au fost propuse mai multe metode bazate pe învățarea profundă în literatura de specialitate. Această secțiune detaliază cele mai utilizate metode din această zonă, clasificate în funcție de tipul arhitecturii DNN. Metodele identificate de predicție a traiectoriei pietonale bazate pe învățarea profundă au utilizat în mare parte trei structuri arhitecturale, după cum urmează.

Cu costuri mai reduse de calcul și comunicare mai rapidă, care oferă acces nelimitat la informație și o mai bună înțelegere a lumii fizice din jurul nostru, luarea deciziilor puternic automate, cum ar fi inteligența artificială, devine tehnologia motor a secolului XXI. Cu toate acestea, inteligența artificială a fost elementul de bază al multor aplicații, cum ar fi mașinile autonome, asistenții digitali și imagistica medicală, doar pentru a enumera câteva. Cu toate acestea, oare să existe o lipsă de înțelegere corectă a acestei tehnologii critice. În plus, datorită imensului hype din jurul acestei tehnologii, există multe neînțelegeri în terminologie. Această interpretare greșită poate fi observată în principal atunci când termenii inteligență artificială, învățare automată și învățare profundă sunt schimbați tot timpul. Deși termenii par echivalenți, sensul fiecărui termen variază, iar aceasta secțiune își propune să articuleze clar diferențele dintre inteligența artificială, învățarea automata și învățarea profundă.

Igor Aizenberg și colegii săi au inventat pentru prima dată termenul DL în anul 2000 [8]. Algoritmii de învățare profundă sunt o subcategorie a algoritmilor de învățare automată care imită procesul de învățarea al oamenilor prin învățarea mașinii prin intermediul exemplelor. Algoritmii de învățare profundă utilizează structuri de învățare complexe cu mai multe straturi cunoscute sub numele de rețele neurale, care învață o reprezentare implicita a datelor brute in mod autonom pentru a produce rezultatul dorit. Cu alte cuvinte, pentru a face ca algoritmii tradiționali de învățare automată să funcționeze, este necesară o etapă esențială, dar extrem de complicată, cunoscută sub numele de extragerea de caracteristici, care trebuie realizată manual de către experții din domeniul pentru care algoritmii să funcționeze. Pe de altă parte, algoritmii de învățare profundă învață aceste caracteristici extrase automat în timp ce structurile de învățare din acești algoritmi se optimizează pentru a obține cea mai bună reprezentare abstractă posibilă a datelor de intrare.

Din acest motiv, învățarea profundă devine deosebit de utilă, deoarece majoritatea datelor din lume sunt neorganizate (adică există în diferite formate). O altă diferență importantă între învățarea automată și învățarea profundă este că aceasta din urmă se adaptează mai bine la cantități mari de date. Cu alte cuvinte, acuratețea algoritmilor de învățare profundă tinde să crească odată cu creșterea cantității de date, în timp ce algoritmii tradiționali de învățare automată încetează să se îmbunătățească după un punct de saturare. Din aceste motive, toate progresele recente în inteligența artificială pot fi atribuite algoritmilor de învățare profundă.

Diferitele modele de traiectorie de mișcare (cu mai multe origini și destinații) și interacțiunea umană dinamică sunt cheia pentru un model de predicție a traiectoriei sub circumstanțe complexe. Cele mai multe metode existente bazate pe învățarea profundă depind puternic de scenarii specifice, deoarece efectuează predicția

traiectoriei folosind coordonate absolute. În realitate, traiectoria de mișcare este o mișcare relativă care coincide cu timpul, iar interacțiunea umană este o mișcare relativă între pietoni. Acest lucru motivează construirea unui model de predicție a traiectoriei pentru mișcarea relativă atât a traiectoriei de mișcare, cât și a interacțiunii umane.

Diferite arhitecturi de învățare profundă au fost proiectate pentru a aborda sarcini specifice în diferite domenii. De exemplu, RNA recurente și mecanismele de atenție au fost utilizate în principal pentru modelarea limbajului în procesarea naturală a limbajului. Iar rețelele convoluționale sunt utilizare extensiv pentru a rezolva problema clasificării imaginilor și recunoașterea obiectelor în domeniul viziunii artificiale. Aceste arhitecturi au fost, de asemenea, extinse și adaptate în alte domenii, cum ar fi previziunile financiare și înțelegerea climei. Ele sunt, de asemenea, utilizare în domeniul condusului autonom și în mod specific, pentru prognozarea mișcării.

2.2. Rețelele neuronale artificiale

Cea mai simplă definiție a unei rețele neuronale artificiale (RNA), conform Dr. Robert Hecht-Nielsen, inventatorul primului neuro-calculator, este "un sistem informatic format din mai multe elemente de procesare simple, puternic interconectate, care procesează informații prin răspunsul lor dinamic la intrările externe"[9].

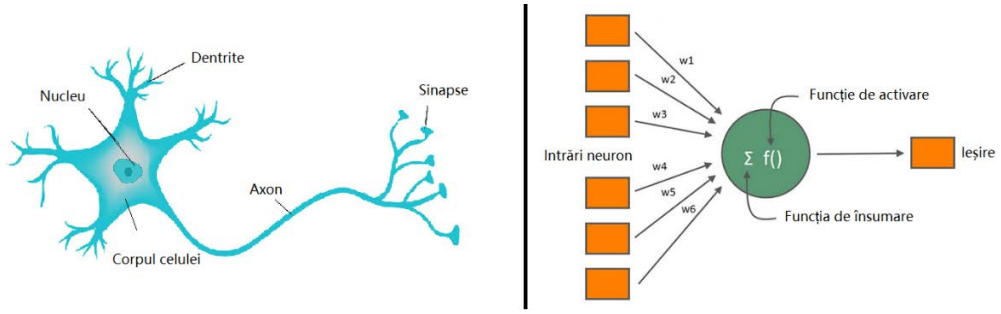

Figura 2.2.1. Comparație între neuronul biologic și neuronul artificial.

Rețelele neuronale artificiale au fost create inițial ca o încercare de a imita neuroni biologici din creierul uman, astfel încât, la fel ca creierul uman care este format din neuroni biologici care procesează impulsurile electrice primite de la neuroni adiacenți și le transmit mai departe, rețelele neuronale artificiale sunt formate din mai multe straturi de noduri (cunoscute șu sub numele de perceptron) care procesează intrările multiple pe care le primesc pentru a produce ieșirea. Ieșirea din acest nod simplu (sau perceptron) poate fi reprezentată ca Ecuația 2.2.1.

$$y = w^T x + b \tag{2.2.1}$$

După cum se poate observa din Ecuația 2.2.1, datele de intrare la nod sunt definite ca $\boldsymbol{x}$. Intrarea este primită direct din setul de date pe care rețeaua neuronală se antrenează sau ca ieșire din nodul precedent reprezentat ca $\boldsymbol{y}_{n-1}$. Ieșirea din rețea este reprezentată ca y, care este suma intrărilor ponderate reprezentate ca $\mathbf{w}^{\mathrm{T}}\mathbf{x}$ și a

abaterii corective **b**. Parametrii greutate w și abaterea corectivă b sunt reglabili, iar valorile finale determină performanța întregii rețele.

Rețeaua neuronală reprezentată în Figura 2.2.1 este un exemplu de rețea feedforward. Rețelele feedforward sunt rețele neuronale în care conexiunile între noduri nu formează un circuit. Rețelele feedforward sunt considerate unul dintre cele mai simple tipuri de arhitecturi de rețele neuronale, fiind modele esențiale pentru învățarea profundă (deep learning) [10]. Deoarece într-o rețea feedforward nu există conexiuni de feedback, acestea sunt considerate de tip serie temporală. Deoarece majoritatea datelor disponibile astăzi sunt foarte neorganizate și non-liniar separabile, adăugarea unei sau mai multor straturi în rețeaua neurală ar permite acesteia să rezolve probleme în care datele sunt non-liniar separabile. Numărul total de straturi într-o rețea neuronală definește "adâncimea" rețelei neurale.

Până acum, am descris trecerea înainte, ceea ce înseamnă că, dată o intrare și greutățile, se calculează ieșirea. După ce antrenamentul este finalizat, rulăm doar trecerea înainte pentru a face predicțiile. Dar mai întâi trebuie să antrenăm modelul pentru a învăța efectiv greutățile, iar procedura de antrenare funcționează în felul următor:

- Inițializăm greutățile pentru toate nodurile în mod aleatoriu. Există metode inteligente de inițializare pe care le vom explora într-un alt articol.
- Pentru fiecare exemplu de antrenare, efectuăm o trecere înainte folosind greutățile curente și calculăm ieșirea fiecărui nod de la stânga la dreapta. Ieșirea finală este valoarea ultimului nod.
- Comparăm ieșirea finală cu ținta reală din datele de antrenare și măsurăm eroarea folosind o funcție de pierdere.
- Efectuăm o trecere înapoi de la dreapta la stânga și propagăm eroarea către fiecare nod individual folosind backpropagation. Calculăm contribuția fiecărei greutăți la eroare și ajustăm greutățile corespunzătoare folosind gradientul descendent. Propagăm gradientul de eroare înapoi începând cu ultimul strat.

2.3. Rețele neuronale recurente

Rețelele neuronale recurente (RNN) au câștigat o popularitate enormă în ultimii ani datorită capacității lor de a manipula date secvențiale nestructurate. Spre deosebire de rețelele neuronale feed-forward, cum ar fi cele convenționale, în care intrările și ieșirile sunt independente, rețelele neuronale recurente să obțină cunoștințe din intrările anterioare pentru a modifica intrarea și ieșirea curentă. Cu alte cuvinte, din ieșirea rețelei neuronale recurente depinde de informațiile istorice din secvență. Calculațiile derivate din intrarea anterioară sunt introduse înapoi în rețea, ceea ce este crucial în învățarea relațiilor neliniare dintre mai multe variabile de calitate a apei. În plus, rețelele neuronale recurente sunt computațional foarte lente. Din aceste motive, sunt necesare specializări suplimentare în rețelele neuronale recurente pentru a procesa secvențe lungi. Un exemplu de rețea neuronală recurentă este reprezentat în Figura 2.3.1.

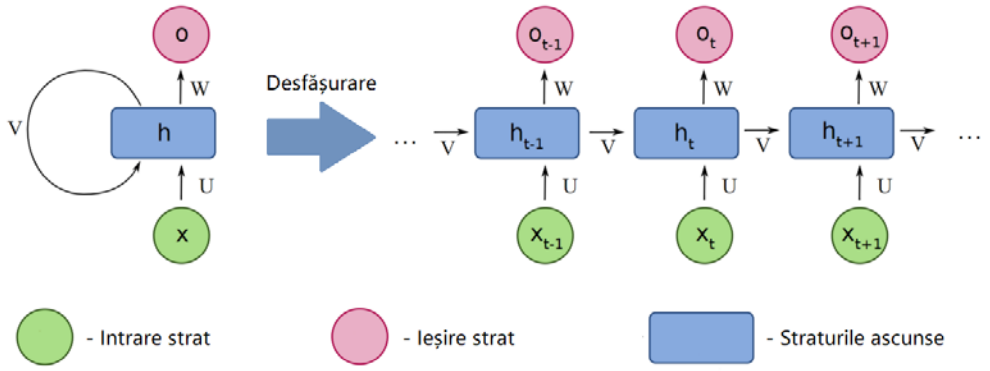

Figura 2.3.1. Prezintă un exemplu de cum poate fi structurat un RNN simplu.

Pentru a antrena o rețea neuronală cu propagare înainte (în limba engleză feedforward) folosind metoda de propagare înapoi (în limba engleză backpropagation), este necesar să se calculeze derivatele acesteia și să se aplice regula derivatei în lanț. În RNN-uri, gradienții depind nu numai de intrarea la un singur moment de timp, ci și de pașii de timp anteriori. Pentru a antrena RNN-uri, se folosește o extensie a algoritmului obișnuit de de propagare înapoi, numită „BackPropagation Through Time" (BPTT) [11]. BPTT desfășoară rețeaua neuronală recurentă în timp, apoi folosește algoritmul de propagare înapoi, având în vedere că parametrii rețelei sunt aceiași la fiecare pas de timp. În practică, se folosește „Truncated Backpropagation Through Time" (TBPTT) [12], care elimină istoricul folosit pentru a reduce nevoia urmărire înapoi prin întreaga secvență de intrare. Această tehnică este utilizată chiar cu riscul favorizării dependențelor pe termen scurt.

Când gradienții sunt propagați înapoi în timp până la stratul inițial, aceștia trec prin numeroase înmulțiri de matrice din cauza regulii de lanț. Un gradient cu o amplitudine inițială mică va avea tendința să se estompeze (gradient de dispariție) în timpul acestui proces și nu va avea nicio influenta asupra învățării, în timp ce un gradient cu o amplitudine inițială mare va avea tendința să crească prea mult (gradient de explozie) și, de obicei, va cauza probleme de condiționare numerică sau va avea o influenta disproporțională asupra învățării.

În plus față de aceste probleme de inițializare și învățare, regularizarea RNN-urilor se dovedește a fi dificila. RNN-urile necesită scheme specifice de regularizare, deoarece posedă în mod natural un bias inductiv mai puternic decât rețelele feedforward datorită definiției lor recurente. Cele mai faimoase tehnici de regularizare pentru RNN includ zoneout [13], „variational dropout" [14], „recurrent dropout" [15], dropconnect la greutățile „hidden-to-hidden" cu o metodă de antrenare a gradientului stocastic mediat [16] și normalizarea batch-urilor recurente [17].

În general, intrarea într-un model RNN este o secvență de lungime variabilă $x = \{x_1, x_2, \ldots, x_T\}$ unde $x_i \in \mathbb{R}^d$ și d reprezintă dimensiunea lui x_i. La fiecare pas de timp, RNN își menține starea sa internă ascunsă h, ceea ce duce la o secvență ascunsă a lui $\{h_1, h_2, \ldots, h_k\}$. Operația unui RNN la pasul de timp t poate fi formulată în Ecuația (2.3.1):

$$h_t = f(ux_t + vh_{t-1}), \tag{1.3.1}$$

unde *f()* este o funcție de activare, *u* este matricea de greutăți convenționale între un strat de intrare *x* si un strat ascuns *h*, iar *v* este matricea între un strat ascuns *h* și el însuși la pasul de timp adiacent.
Ieșirea RNN este calculată de Ecuația:

$$o_t = wh_t, \tag{2.3.2}$$

unde w este matricea de greutăți intre stratul ascuns *h* și ieșirea *o*.

Așa cum este reprezentat în Figura 2.3.1, structura modelului RNN în timp poate fi exprimată ca o rețea neurală profundă cu un strat la fiecare pas de timp. Deoarece această buclă de feedback are loc la fiecare pas de timp în serie, fiecare stare ascunsă conține urme nu numai ale stării ascunse anterioare, ci și ale tuturor celor care au procedat h_{t-1} pe cât timp memoria poate persista. Comparativ cu rețeaua neuronală tradițională „feed-forward", structura recurentă în RNN poate păstra informația secvențială în multe etape de timp pe măsură ce se înaintează pentru a afecta procesarea fiecărui nou exemplu.

Datele colectate în perioade succesive de timp sunt denumite serii temporale. Prognoza de mișcare poate fi abordată ca o problem- de regresie sau clasificare a seriilor temporale. RNN-urile și variantele lor sunt principala cauză a progreselor semnificative în modelarea și generarea seriilor temporale. Acestea au prezentat rezultate promițătoare în domenii diverse, cum ar fi procesarea limbajului natural și recunoașterea vocală. Prin urmare, abordările bazate pe RNN au fost utilizate și în sarcinile de predicție a manevrelor și traiectoriilor. Spre deosebire de alte rețele neurale, acestea iau în considerare informația secvențială și modelează dependența dintre intrări. Acestea acționează prin efectuarea acelorași operații pentru fiecare element de intrare dintr-o secvență, luând în considerare și calculul elementului de intrare anterior.

2.4. LSTM

Memoria pe termen lung (în literatura de limbă engleză „Long Short-Term Memory" - LSTM) este o variabilă specială a RNN folosită în mod frecvent pentru modelarea seriilor de timp. Aceasta rezolvă problema gradientului care dispare în cazul RNN-urilor fiind caracterizată prin capacitatea sporită de a învăța dependențe pe termen lung. LSTM este formată dintr-un lanț de module ale rețelelor neuronale. Elementul-cheie este memoria celulei care trece prin întregul lanț LSTM și stochează informațiile relevante despre secvența de intrare anterioară. Mecanismele de control al fluxului de informații între intrare, ieșire și memoria celulei sunt utilizate pentru a guverna funcționarea LSTM-ului.

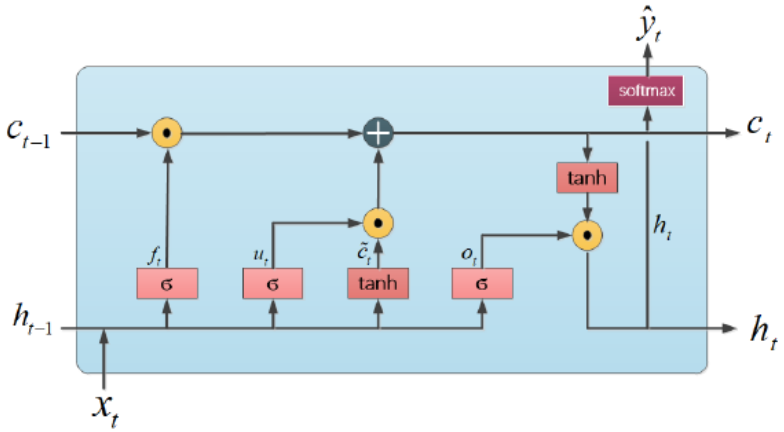

Figura 2.4.1. Reprezentarea fiecărui modul LSTM [18]

LSTM a fost introdusă de Hochreiter și Schmidhuber în 1997 [18] și este capabilă să învețe dependențe pe termen lung. LSTM conține stări ale celulei pentru a-și aminti informațiile din secvența de intrare și porți pentru a permite opțional informația să treacă prin intrare, starea celulei și ieșire. Ca o formă specială de RNN-uri, LSTM are, de asemenea, forma unui lanț de module repetate ale rețelelor neuronale. Fiecare modul LSTM, ilustrat în Figura 2.4.1, funcționează prin următoarele ecuații:

$$u_t = \sigma\left(W_u\left[h_{t-1,}\, x_t\right] + b_u\right), \tag{2.4.1}$$

$$f_t = \sigma\left(W_f\left[h_{t-1,}\, x_t\right] + b_f\right), \tag{2.4.2}$$

$$o_t = \sigma\left(W_o\left[h_{t-1,}\, x_t\right] + b_o\right), \tag{2.4.3}$$

$$\tilde{c}_t = tanh\left(W_c\left[h_{t-1,}\, x_t\right] + b_c\right), \tag{2.4.4}$$

$$c_t = u_t \odot \tilde{c}_t + f_t \odot c_{t-1}, \tag{2.4.5}$$

$$h_t = o_t \odot tanh(c_t), \tag{2.4.6}$$

În ecuațiile de mai sus, x_t reprezintă vectorul de intrare la momentul t; $h_{t-1,}$ și $h_{t,}$ denotă stările ascunse la momentul de timp t-1 și t; c_{t-1} și c_t sunt stările celulei la momentul de timp t-1 și t, în timp ce $\tilde{c}_t$ este o stare a celulei candidat; operatorul $\odot$ reprezintă înmulțirea element-cu-element; W_u, W_f , W_o , W_c sunt matricile de greutate utilizate pentru a calcula vectorul porții de actualizare u_t, vectorul porții de uitare f_t,

și vectorul porții de ieșire o_t; b_u, b_f , b_o, b_c sunt vectorii de bias; σ denotă funcția sigmoidă.

LSTM funcționează prin actualizarea memoriei celei pe baza vectorului de intrare, a memoriei celulei anterioare și a vectorului stării ascunse. Ea generează starea ascunsă folosind memoria celulei. Pentru a face acest lucru, porțile (compuse dintr-un strat de rețea neuronală sigmoid și o înmulțire a elementelor corespondente) primesc valorile vectorului de intrare și a vectorului anterior al stării ascunse pentru a genera vectorii de intrare, de uitare și de ieșire subsecvenți, conform ecuațiilor (2.4.1), (2.4.2) și (2.4.3). Apoi, vectorii porții de intrare și de eliminare actualizează memoria celulei prin înlăturarea informațiilor inutile din memoria celulei anterioare sau prin adăugarea informației bazate pe vectorul de intrare și vectorul stării ascunse anterioare (Ecuația 2.4.5). În cele din urmă, vectorii porții de ieșire filtrează starea celulei pentru a genera starea ascunsă curentă (Ecuația 2.4.6).

Metoda socială LSTM a avut o importanță mare în domeniul prezicerii traiectoriei pietonilor, deoarece a demonstrat că tehnicile de învățare profundă aplicate prezicerii traiectoriei pietonilor pot depăși metodele bazate pe dinamică. Începând din 2016, au fost publicate numeroase lucrări care abordează prezicerea traiectoriei pietonilor folosind rețele neuronale profunde. Aceste lucrări introduc noi arhitecturi și tehnici pentru a îmbunătăți performanțele în prezicerea traiectoriei pietonilor. Acestea sunt prezentate în secțiunile următoare. În prezent, diferit de perioada anterioară anului 2016, metodele folosind învățarea profundă reprezintă principala abordare în prezicerea traiectoriei pietonilor.

2.5. Rețele neuronale convoluționale

Rețeaua neuronală convoluțională (CNN) este una de tip profundă (DNNs), care are o performanță ridicată, fapt demonstrat în multe domenii, cum ar fi clasificarea și recunoașterea obiectelor (de exemplu cifre scrise de mână, litere și fețe). CNN are arhitectura tipică prezentată în Figura 2.5.1 și conține un număr mare de structuri de tip convoluțional, de tip non-liniaritate, de reducere a dimensiunii, de tip abandonare, de normalizare în lot și de tip complet conectate. Rezultatul antrenării parametrilor rețelei face ca aceste caracteristici să includă cea mai semnificativă informație discriminativă necesară pentru identificarea robustă a obiectelor.

Rolul CNN-ului este de a extrage caracteristici semnificative din datele de intrare prin aplicarea succesivă a straturilor de operații convoluționale și de punere în comun. Componenta de bază a unei CNN este stratul convoluțional. Convoluția este o operație matematică care îmbină două seturi de informații. O operație de convoluție se realizează pe intrare folosind un nucleu cunoscut și sub numele de filtru de convoluție pentru a produce o hartă de caracteristici. Atât intrarea cât și filtrul de convoluție sunt vectori sau matrice. În procesul de convoluție nucleul este deplasat peste intrare (cu un număr de poziții) în fiecare locație. Numărul de poziții cu care nucleul se deplasează după fiecare operație se notează cu *S*. În fiecare locație, se realizează o înmulțire element cu element între nucleu și intrare și rezultatele sunt adunate. Acest lucru calculează valoarea hărții de caracteristici pentru acea locație particulară. Deplasarea nucleului peste întreaga intrare calculează harta completă de caracteristici.

Există și CNN-uri 3D, în cazul în care nucleele se mișcă prin trei dimensiuni ale datelor (înălțime, lungime și adâncime) și produc hărți de activare 3D. Un astfel de exemplu sunt imaginile color (în care imaginea rezultat este formată dintr-o sumă

ponderată a imaginilor de verde, roșu și albastru). Un alt exemplu este reprezentat de procesarea datelor de tip nor de puncte.

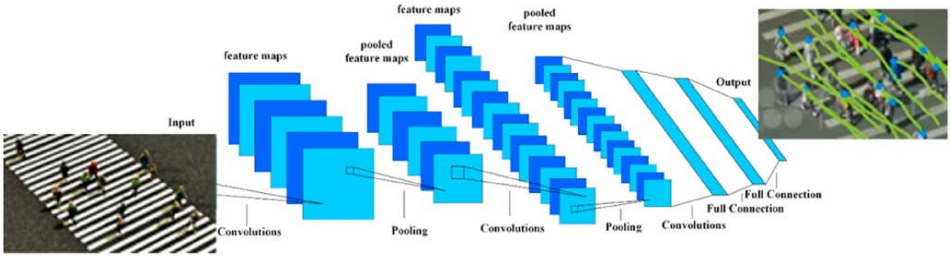

Figura 2.5.1. Prezintă arhitectura tipică a unei rețele neuronale convoluționale (CNN): stratul de intrare, multiple straturi de convoluție cu funcții de activare ReLU, straturi de (max) pooling, aplatizare (flatten), straturi complet conectate și straturi SoftMax de ieșire. [30].

Într-un model complet de CNN, după fiecare strat de convoluție urmează un strat „Rectified Linear Unit"" (ReLU) care va aplica o funcție de activare elementară. Un exemplu de astfel de funcţie este max(0,x), care aduce valorile negative la zero. Acest lucru lasă dimensiunea volumului variabilelor de intrare neschimbată. Pentru reducerea dimensiunilor, se apelează la o operație de punere în comun (în literatura de limba engleză, „pooling"). Aceasta reduce numărul de parametri, ceea ce scurtează timpul de antrenament și evită supra adaptarea (în literatura de limba engleză, overfitting-ul). Straturile de punere în comun reduc dimensiunea fiecărei hărți de caracteristici independent, modificând atât înălțimea cât și lățimea, dar menținând adâncimea intactă. Cele două tipuri principale de punere în comun utilizate în CNN-uri sunt de tip maxim (în literatura de limba engleză, „max pooling") și mediu (în literatura de limba engleză, average pooling). Ambele timpuri de straturi de punere în comun returnează, valoarea maximă respectiv, valoarea medie din partea caracteristicilor acoperite de nucleu.

Straturile complet-conectate finale (în literatura de limba engleză fully connected, FC) calculează scorurile (probabilități) ale fiecărei clase. La fel ca în cazul rețelelor obișnuite, fiecare nod din aceste straturi va fi conectat la toți neuronii din stratul anterior. Aceste straturi pot învăța să aproximeze diferite funcții, dar vor avea un număr mare de parametri (deoarece conțin multe conexiuni). Antrenarea lor durează un timp mai lung decât straturile de convoluție (și implică un consum de energie mai mare). Un strat FC de dimensiune N este reprezentat de Ecuația (2.5.1).

$$o = W * h + b \tag{2.5.1}$$

În Ecuația (2.5.1) s-a notat cu *h* vectorul de intrare de dimensiune *H*, cu *W o matrice de ponderi de dimensiune* $H * N$ și cu *b* vectorul de polarizare de dimensiune *N.*

CNN au fost larg utilizate în multe sisteme de recunoaștere a imaginilor. O astfel de rețea constă în mod obișnuit din straturi convoluționale, straturi de reducere a eșantioanelor și câteva straturi complet conectate. Mai multe filtre de kernel sunt folosite pentru a detecta caracteristicile imaginii. Straturile de reducere a eșantioanelor au capacitatea de a micșora dimensiunile pentru a obține o

caracteristică de rezoluție inferioară. Ulterior, hărțile de caracteristici finale se conectează la straturi complet conectate. În final, modelul este antrenat pentru a reduce eroarea dintre ieșirea rețelei și ieșirea țintă prin propagarea înapoi. CNN poate detecta informații de caracteristici ascunse din variabilele de intrare datorită filtrelor de kernel reutilizate.

CNN-urile sunt folosite în principal pentru reprezentarea caracteristicilor imaginii și domenii conexe, cum ar fi clasificarea lor, segmentarea și detecția obiectelor [19], [20], [21]. Diverse arhitecturi de CNN-uri sunt aplicate în mod frecvent în construirea algoritmilor, printre acestea menționându-se AlexNet, VGGNet, GoogLeNet, ResNet, etc. [22], [23]. În timp ce primele straturi convoluționale capturează caracteristicile de nivel scăzut, cum ar fi culoarea, marginile, etc., straturile mai profunde extrag, de asemenea, caracteristicile de nivel înalt. Acestea sunt printre primele CNN de tip profund.

Rețelele neuronale convoluționale sunt adoptate într-o varietate de aplicații, cum ar fi recunoașterea facială, etichetarea scenelor, clasificarea imaginilor, recunoașterea acțiunilor, estimarea poziției umane, segmentarea semantică etc. Recent acestea au găsit aplicații în domeniul automotive, mai precis în vehiculele cu conducere autonomă. În acest domeniu, informația de intrare provine de la o varietate de senzori, camera, radar, și LiDAR. RNN funcționează bine pentru segmentarea și clasificarea imaginilor. Cu toate acestea, pentru alte aplicații, rezultate mai bune sunt obținute prin combinarea lor cu alte rețele neuronale, cum ar fi LSTM. Acesta este cazul pentru predicția traiectoriei.

2.6. Rețele adversative generative

Reţelele adversative generative (în literatura de limbă engleză, Generative Adversarial Networks – GAN) [24] reprezintă o metodă de a genera date noi și sintetic asemănătoare cu cele de antrenament. GAN reprezintă o modalitate de a aborda aspectul multimodal al problemei de estimare a traiectoriei pietonilor. În acest caz predicția corectă a locației acestora în viitor pe baza pozițiilor anterioare, este o provocare. De exemplu, drumul se poate împărți în două și traiectoriile care merg într-o direcție sau alta sunt ambele posibile.

În scopul urmării, rețeaua GAN reduce fragmentarea care apare de obicei în multe modele convenționale de predicție a traiectoriei și reduce necesitatea de a calcula caracteristici costisitoare de apariție. Arhitectura este compusă din două părţi care concurează între ele. Observațiile candidat sunt produse și actualizate de componenta generativă; cele mai puțin actualizate sunt eliminate. Rețelele GAN nu învață explicit distribuția datelor, dar sunt capabile să genereze exemple realiste noi din distribuția modelului. Pentru a procesa și clasifica secvențele candidat, este utilizată concomitent componenta LSTM cu un model generativ-discriminativ. Această metodă poate duce la modele de înaltă precizie ale comportamentului uman, în special a celui de grup. În comparație cu soluțiile bazate anterior pe CNN, implementarea GAN este considerabil mai ușoară. Recent, mulți autori au aplicat această arhitectură pentru a obține multe modalități în rezultatul predicției. Acest lucru este detaliat în paragraful următor.

Componenta generator primește ca intrare zgomot aleatoriu (vector aleatoriu) și generează o probe noi, în timp ce rețeaua discriminatorului primește atât probe reale, cât și cele generate, ea trebuind să determine dacă proba este falsă sau nu. Obiectivul generatorului este de a induce în eroare discriminatorul, iar cel al discriminatorului este de a eticheta corect datele pe care le primește. Antrenarea

modelelor bazate pe GAN permite generatorului să învețe să genereze exemple plauzibile. Acest lucru poate fi detectat atunci când discriminatorul este păcălit de aproximativ 50% dintre exemplele generate. Prin urmare, în cadrul unui GAN, un model generator și un model discriminator sunt antrenate simultan (Figura 2.6.1). În procesul de antrenare, cele două părți ale rețelei folosesc un algoritm numit minmax.

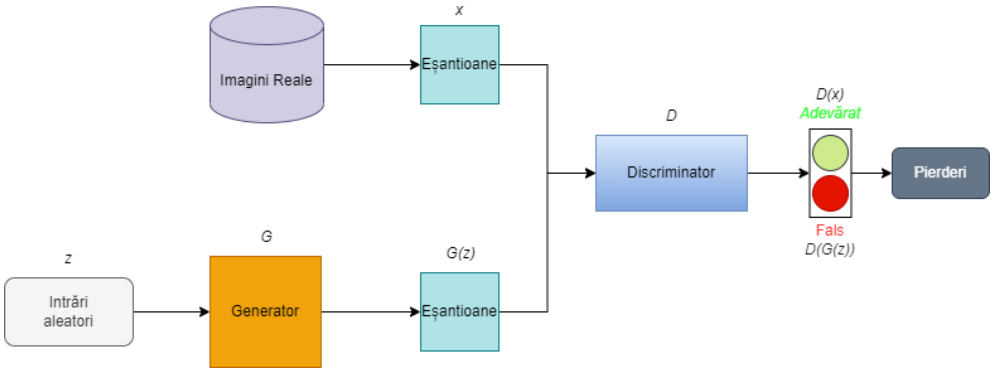

Figura 2.6.1. Structura arhitecturii GAN

Așa cum a fost menționat anterior, scopul generatorului este să inducă în eroare discriminatorul cât mai mult posibil, astfel încât acesta să eticheteze imaginile generate ca fiind adevărate. Pentru a măsura acest lucru se introduce o funcție de pierderi. Aceasta este reprezentată de Ecuația (2.6.1). Este important de amintit că rezultatele funcției de pierderi trebuie minimizate. În cazul generatorului, ar trebui să încerce să minimizeze diferența dintre 1, eticheta pentru datele reale, și evaluarea discriminatorului pentru datele fals generate.

$$L_G = \text{Error}\big(D\big(G(z)\big), 1\big) \tag{2.6.1}$$

Așa cum a fost menționat mai sus, obiectivul discriminatorului este de a eticheta drept false sau adevărate imaginile generate și punctele de date empirice ca fiind adevărate. Prin urmare, se poate considera o funcție de pierdere reprezentată de Ecuația (2.6.2). Aceasta folosește o notație generică și nespecifică pentru eroare pentru a face referire la o anumită funcție care arată distanța sau diferența dintre cele două seturi de parametri funcționali.

$$L_D = \text{Error}(D(x), 1) + \text{Error}\big(D\big(G(z)\big), 0\big) \tag{2.6.2}$$

2.7. Rețele neuronale de tip graf

Rețelele neuronale grafice (în literatura de limbă engleză “Graph Neural Networks – GNN”) sunt o clasă relativ nouă care utilizează structura și proprietățile grafurilor. Grafurile sunt considerate un tip specific de structură de date care reprezintă relațiile (cunoscute și sub numele de muchii) dintre colecții de entități (numite și noduri). Modelele de învățare profundă, precum rețelele neuronale

convoluționale, iau de obicei ca intrare matrice sau aranjamente de tip grilă. Odată ce toate proprietățile grafurilor sunt reprezentate într-un format compatibil cu modelele de învățare profundă, rețelele neuronale cu organizare de tip graf pot fi utilizate pentru a efectua o transformare optimizată a atributelor care păstrează simetriile grafului. Cu alte cuvinte, rețelele neuronale cu organizare de tip graf sunt o clasă care acceptă un graf ca intrare, cu informații încărcate în nodurile, marginile și contextul global care transformă progresiv aceste încorporări fără a schimba conectivitatea grafului de intrare [25]. În figura 2.7.1 se poate vedea un exemplu simplu al unei rețele neuronale de tip graf.

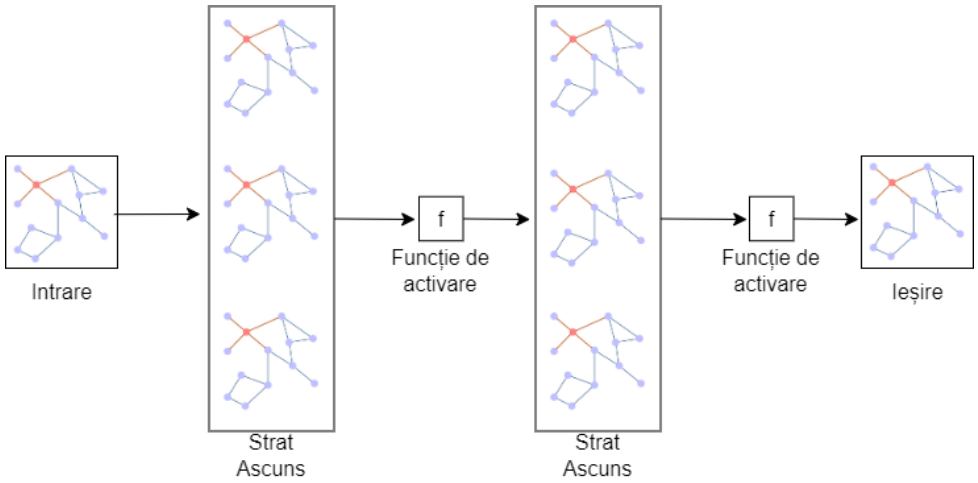

Figura 2.7.1. Un exemplu simplu de GNN.

Modelele de învățare profundă, precum CNN, iau de obicei ca intrare matrici rectangulare. Prin urmare, grafurile nu sunt ușor de reprezentat într-un format compatibil cu modelele de învățare profundă. Una dintre cele mai mari provocări cu structurile de date grafice este reprezentarea conectivității între noduri.

2.7.1. Grafuri, straturi și seturi

Un graf G se poate defini ca o structură de date (cunoscută în literatura de limba engleză sub numele de tupla) formată din două mulțimi, una de noduri V respective una de muchii $\mathcal{E}$: $G = (V, \mathcal{E})$. Se consideră numărul de noduri sau dimensiunea grafului notată $|V| = n$ iar $|gE| = m$ să fie totalul de muchii. De asemenea, se notează numărul de muchii $\mathcal{E} = \{(u_i, u_j) | u_i \in V, u_j \in V, \text{există o conexiune între } u_i \text{ și } u_j\}$.

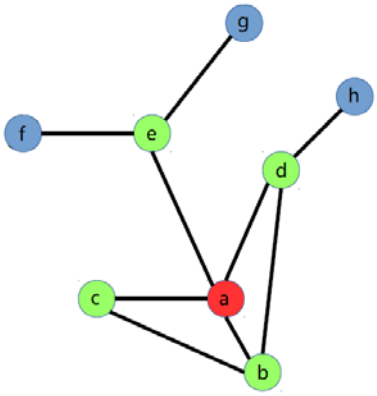

Figura 2.7.1.1. Exemplu de vecinătate într-un graf. Vecinătatea nodului a este egală cu $\mathcal{N}_a = \{b, c, d, e\}$. Nodurile h, f și g sunt considerate vecini de gradul 2 ai nodului a.

Pentru un nod *u*, se definește vecinătatea sa de gradul 1, notată $\mathcal{N}_u$ ca fiind mulțimea nodurilor care prezintă puncte de legătură ale muchiilor atașate nodului u: $\mathcal{N}_u = \{v: (u, v) \in \mathcal{E} \; or \; (u, v) \in \mathcal{E}\}$. Definiția de vecinătate este fundamentală pentru modelele de învățare a grafurilor, deoarece formalizează căile prin care informația se propagă. În Figura 2.7.1.1 prezentăm un exemplu simplu al vecinătății de gradul 1. Astfel, nodurile *b*, *c*, *d* și *e*, sunt vecini de gradul 1 pentru nodul *a* (pentru fiecare dintre acestea este necesar un singur salt pornind de la nodul *a*). Nodurile *h, f și g* sunt considerate vecini de gradul 2, deoarece, pornind de la nodul *a*, avem nevoie de două opriri (sau salturi) pentru a ajunge la aceste noduri.

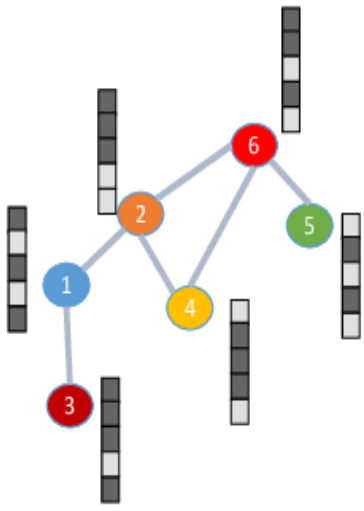

Figura 2.7.1.2. Graful cu atribute. Fiecare nod are un vector atribut și o etichetă.

Din considerente de simplificare, în multe modele de învățare a grafurilor, presupunem că nu există auto-conexiuni, adică nu există muchii care să pornească și să ajungă la același nod. De asemenea, presupunem că există cel mult o singură muchie între noduri și că pentru fiecare muchie $(u_i, u_j) \in g\mathrm{E}$, există și muchia inversă $(u_j, u_i) \in \mathcal{E}$. Graful pentru care acest lucru este adevărat se numește, graf nedirecționat.

De multe ori, modelarea unei rețele din lumea reală printr-un graf nu este suficientă datorită informațiilor contextuale pe care componentele rețelei le poartă. De exemplu, într-o rețea socială, unde nodurile se referă la utilizatori și muchiile se referă la conexiunile dintre ei, informațiile tipice care trebuie procesate pot fi postări, comentarii sau reacții ale unui utilizator. Pentru codificarea acestor informații, definim matricele de atribute $X_v \in \mathbb{R}^{n \times d_v}\ and\ X_e \in \mathbb{R}^{m \times d_e}$, unde d_v, d_e reprezintă dimensiunile vectorilor de caracteristici ale nodurilor și ale muchiilor, respectiv. În funcție de sarcina utilizatorului, graful poate conține, de asemenea, informații despre etichete la nivel de nod, la nivel de muchie sau la nivel general (de graf). Acestea sunt reprezentate de vectorii $Y_v \in \mathbb{N}^n, Y_e \in \mathbb{N}^m, Y_G \in \mathbb{N}$. Un exemplu tipic de graf atribuit este vizualizat în Figura 2.7.1.2 unde avem vectori de atribute ale nodurilor și etichetele nodurilor (reprezentate prin culorile asociate vectorilor de atribute).

2.7.2. Codificarea muchiilor

Cea mai simplă codificare a existenței unei muchii într-un graf este cea binară. Aceasta duce la definirea matricei de adiacență $A \in \{0,1\}^{n \times n}$, unde $A_{ij} = 1$ dacă și numai dacă $(i,j) \in \mathcal{E}$. Gradele grafului G pot fi acum reprezentate prin $D = Diag(A1_n)$, unde 1_n este un vector cu toate elementele unitare de dimensiune *n*. În cazul unui graf nedirecționat, matricea A este simetrică. Date fiind matricele A și D, se poate defini matricea Laplacian $L = D - A \cdot A$ și variantele sale de normalizare: Laplacianul normalizat simetric $L_{sym} = I_n - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ și Laplacianul normalizat pentru mișcare aleatorie $L_{rw} = I_n - D^{-1}A$. Deoarece L_{sym} este o matrice reală simetrică (și pozitiv semi-definită), se poate descompune în matricea sa de vectori proprii $U \in \mathbb{R}^{n \times n}$ cea de valori proprii $\Lambda = Diag(\lambda_1, \ldots, \lambda_n)$, unde $\lambda_1, \ldots, \lambda_n$ sunt valorile proprii ordonate ale lui L_{sym} așa cum se vede în Ecuația (2.7.2.1).

$$L_{sym} = U \Lambda U \tag{2.7.2.1}$$

Având această descompunere a informațiilor de adiacență, se poate utiliza un tip de convoluție pentru a formaliza propagarea informațiilor într-un cadru spectral.
O caracteristică structurală crucială a grafurilor este că nu există nicio presupunere cu privire la ordinea nodurilor lor. Acest lucru înseamnă că orice operație aplicată pe un graf nu ar trebui să presupună o anumită ordine a nodurilor să depindă de aceasta. Această proprietate se numește invarianță la permutare deoarece indiferent de schimbarea nodurilor, funcțiile care acționează pe graf ar trebui să rămână neschimbate. Mai precis, considerând o matrice **P** de dimensiune *n* x *n*, unde fiecare rând și fiecare coloană conțin exact un element unitar. **P** este numită matrice de permutare deoarece dacă ea este înmulțită cu o altă matrice *n* x *n* (de exemplu, adiacența *A* sau laplacianul *L*), va permuta rândurile și coloanele (și, în consecință, etichetele nodului unui graf). De asemenea, o funcție f este numită invariantă la permutare dacă $f(PX) = f(X)$ pentru toate matricele de permutare **P**.

Chiar dacă proprietatea de invarianță la permutări este dorită la nivel global al unui graf (de exemplu, agregând informațiile din întregul set de noduri), majoritatea modelelor de antrenament pe grafuri constau în învățarea reprezentărilor la nivel de nod. Fiind dat un model *M*, învățăm reprezentări *H=M(X)*, unde fiecare rând corespunde informației unui nod. Așa cum descriu autorii în [26], rândurile din H ar trebui să fie aliniate cu rândurile din *X* și, astfel, o matrice de permutare **P** care acționează asupra lui *X* ar trebui să acționeze similar și asupra lui *H*. Această observație creează necesitatea unei alte proprietăți, și anume echivarianța permutării, adică: $f(PX) = Pf(X)$. În formele standard ale rețelelor neuronale grafice, reprezentarea nodului se bazează pe funcții de agregare a vecinătății și, pentru a furniza reprezentări valide, aceste funcții de agregare trebuie să fie echivalente permutării [27], [28], [29].

3. ANALIZA STADIULUI ACTUAL ÎN PREDICȚIA TRAIECTORIEI PIETONILOR

Acest capitol prezintă stadiul evoluției studiilor și metodelor de lucru legate de problematica de predicție a traiectoriei pietonilor. Am propus o taxonomie a metodelor pentru o mai bună înțelegere și clasificare a diferitelor abordări ale problemei [30].

Pentru a rezolva problema PTP, în ultimii ani au fost introduse în literatura de specialitate mai multe metode bazate pe o învățarea aprofundată. Acest capitol detaliază cele mai utilizate metode din domeniu, clasificate în funcție de tipul arhitecturii DNN. Metodele identificate de predicție a traiectoriei pietonilor, bazate pe metode aprofundate de studiu au folosit în principal patru structuri arhitecturale. Acestea vor fi descrise în cele ce urmează. O corespondență între arhitecturile de rețelele neuronale folosite în literatură și problema PTP se poate observa în Figura 3.1.

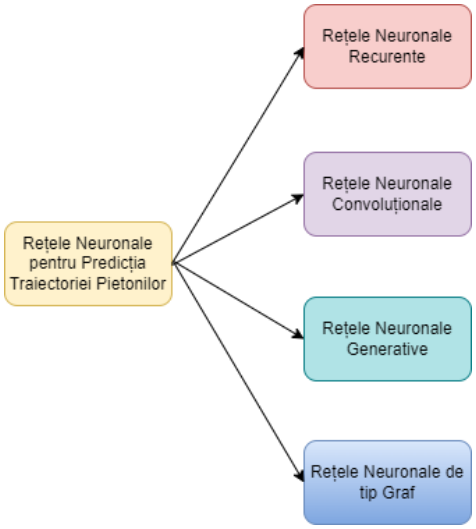

Figura 3.1. Cele mai utilizate rețele neuronale profunde pentru predicția traiectoriei pietonilor.

Arhitecturile utilizate la predicția traiectoriei pietonilor includ rețelele neuronale recurente, cele convoluționale, cele generative dar și cele de tip graf. Estimarea traiectoriei cu acestea este prezentată în paragrafele următoare.

3.1. Predicția traiectoriei bazată pe rețele neuronale recurente

Demonstrând primele rezultate pozitive într-un domeniu de procesare a limbajului natural (NLP) prin modelarea caracteristicilor de date latente, LSTM este, de asemenea, utilizat pentru predicția traiectoriei pietonilor. De exemplu, în [31], Sun și colaboratorii au folosit modelul LSTM pentru a înțelege evoluția contextului și a activității oamenilor din eșantionul țintă de observare pe termen lung (în speță, de la câteva zile la câteva săptămâni).

Pentru a prezice poziția unei persoane, într-un sistem bazat pe captarea mișcării și, de asemenea, în înregistrări video, Fragkiadaki și colaboratorii [32] au propus o metodă care se bazează pe rețele neuronale recurente, folosind arhitectura codificator-recurent-decodor (în limba engleză „Encoder Recurent Decoder" – ERD). Arhitectura ERD este o extensie a modelului LSTM care încorporează rețele de codificare (în limba engleză „Encoder") și decodare (în limba engleză „Decode") neliniare înainte și după straturi recurente. Codificatorul transformă datele de intrare într-o reprezentare, iar decodorul transcrie ieșirea secvențelor recurente în forma vizuală dorită. În acest fel, arhitectura propusă poate estima poziția viitoare a pietonilor prin analiza poziției persoanei.

Alahi și colaboratorii [33] au propus un model social LSTM care sa prezică traiectoriile comune ale unui grup de pietoni într-un spaţiu continuu. Considerând faptul că persoanele din grup se pot influența reciproc, LSTM-ul distribuie informația stocată în stare ascunsă cu pietonii din apropiere, creând un sistem social de tip punere în comun ("pooling") a informației. S-a raportat că modelul lor depășește rezultatele metodelor actuale pe câteva seturi de date. În munca lor, autorii au antrenat un model LSTM pentru fiecare traiectorie. Acesta este cunoscut şi sub numele de "social LSTM". Acest model a fost testat pe seturile de date ETH [34] şi UCY [35].

S. Dai și colaboratorii [36] au propus un model de predicție a traiectoriei spaţio-temporală a pietonilor bazat pe LSTM. În opinia lor, rețelele LSTM nu pot descrie simultan interacțiunile spațiale dintre diferite vehicule. În plus, au subliniat faptul că rezultatele modelelor LSTM sunt afectate de problema dispariției gradientului. Autorii au introdus o conexiune între intrarea și ieșirea a două straturi consecutive pentru a controla dispariția gradientului și pentru a rezolva predicția traiectoriei în trafic. Performanța modelului propus a fost testată pe seturile de date I-80 și US-101. S-a raportat că modelul propus reușește să estimeze traiectoriile cu o acuratețe mai mare decât alte modele de ultimă generație.

L. Xin și colaboratorii [37] au dezvoltat de ceea ce ei numesc "long-horizon trajectory prediction of surrounding vehicles". Metoda lor (o arhitectură profundă de rețea neuronale bazată pe conceptul "conștient de intenție" LSTM (în literatura de limbă engleză "intention-aware" LSTM) este raportat că înțelege la nivel înalt și spaţio-temporal caracteristicile din comportamentul unui șofer. Pentru antrenamentul rețelei utilizate de aceștia s-a folosit setul de date NGSIM. Rezultatul testelor pe autostradă a evidențiat o estimare superioară față de alte metode. Erorile de predicție longitudinală şi laterală sunt mai mici de 5.77 m, respectiv 0.49 m.

In [38], Lee și colaboratorii au generat un cadru de predicție a traiectoriei numită "DEep Stochastic Inverse-optimal-control RNN Encoder" (DESIRE), care funcționează pentru diverși agenți care interacționează folosind rețele neuronale profunde. Cu scopul de a genera o ipotetică traiectorie viitoare, a fost folosit un auto-codificator condițional variațional. După acestea, a fost folosit un model RNN să clasifice și să aprecieze aceste caracteristici bazate pe modul de control invers optim, luând în considerare contextul din trafic. Această metodă ține cont de natura

multimodală a predicției și estimează viitorul rezultat potențial. Pentru creșterea acurateței estimării, autorii au folosit și un algoritm de feedback. Performanța modelului a fost evaluată folosind seturile de date KITTI [39] și Stanford Drone [40].

O metodă ierarhică a fost dezvoltată de Zheng și colaboratorii [41]. Aceasta abordare consideră obiective pe termen lung și pe termen scurt. Soluția este bazată pe rețele neuronale convoluționale recurente să prezică atât micro-acțiuni (mișcare relativă), cât și macro-obiective (obiective intermediare). Aceste DNN-uri au fost generate individual folosind învățarea supervizată, împreună cu un modul de atenție, și la sfârșit, au fost reglate în comun. Aceasta metodă a fost extinsă de către Zhan și colaboratorii [42] folosind RNN-uri variaționale.

Martinez și colaboratorii [43] au descris o abordare, care se bazează pe RNN cu o arhitectură de tip unitate recurentă cu poartă (în literatura de limbă engleză "Gated Recurrent Unit" – GRU), permițând instruirea unui singur model pe întregul corp uman fără nevoia unui strat de codificare spațială. În loc să lucreze cu unghiurile absolute umane, s-a preferat modelarea componentelor cinematice. Autorii au propus ceea ce ei numesc "residual architecture", care modelează derivate de mișcare de ordinul întâi. A fost raportată creșterea acurateței dar și o predicție lină.

Hug și colaboratorii [44] au propus o structură de tip LSTM cu un model cu strat cu densitate mixtă (în literatura de limbă engleză, "Mixture Density Layer" – MDL) combinat cu o metodă de filtrare a particulelor pentru predicția multimodală a traiectoriei pietonilor. Implementarea lor (în TensorFlow) a folosit calcule vectorizate. Modelul lor a fost testat în câteva intersecții de forma "T". Autorii au folosit scene de pe setul de date "Stanford Drone" în experimente. Au fost utilizate scenarii în care predicatori de probabilitate maximă ar eșua din cauza incapacității lor de a furniza ipoteze multiple. Astfel, aceste scenarii includ intersecții de tip sens giratoriu.

În [45] a fost produs un model de predicție pe termen lung folosind RNN-uri. Arhitectura encoder-decoder prezice în comun mișcarea ego-ului la fel ca traiectoria oamenilor. Autorii au susținut că modelul lor poate estima traiectoriile pietonilor la orizonturi de timp dorite. Instruirea și evaluarea performanței au fost efectuate folosind setul de date "Cityscapes" [46].

In [47], T. Salzmann și colaboratorii au propus o metodă care prognozează traiectorii viitoare condiționale ale unui număr general de agenți (pietoni, vehicule) cu clase semantice diferite, incluzând în același timp date eterogene. Pentru a codifica interacțiunile agenților, a fost dezvoltat un sistem în care fiecare agent are o clasă semantică numită autovehicul, autobuz sau pieton și oferă informații despre succesiunile pozițiilor lor, cu dimensiunea contextului, rezoluția spațială și canalele semantice. Autorii au testat modelul propus pe seturile de date ETH, UCY, nuScenes [48].

Xue și colaboratorii [49] prezintă o schema ierarhică bazată pe LSTM pentru predicția traiectoriei pietonilor în scenarii aglomerate. Autorii au pornit de la cazul pietonilor care se deplasează în locuri aglomerate și au ținut cont de obstacolele și amenajările care pot afecta traiectoria celor aflați în mișcare. Trei codificatoare LSTM au fost folosite pentru 3 clase: pietonul (captează fiecare informație a traiectoriei individuale), social (culege informații despre vecinătate), și cadru (înregistrează informații despre obstacole). Pentru a testa arhitectura propusă autorii au folosit seturile de date ETH, UCY și Town center [50].

O prezentare comparativă cu cele mai bune soluții din literatură a rezultatelor RNN pentru PTP este prezentată în Tabelul 3.1.1.

Tabel 3.1.1. Comparație a rezultatelor RNN pentru PTP

Metodă	Bază de date	Rezultate
Social LSTM [33]	ETH	ADE: 0.50 FDE: 1.07 NL-ADE: 0.25
	UCY	ADE: 0.27 FDE: 0.77 NL-ADE: 0.16
Trajectron++ [47]	ETH	ADE: 0.71 FDE: 1.66 KDE NLL: 1.31
	nuScenes	FDE: 0.01 (1 s) KDE NLL: -5.58 (1 s)
LSTM – Bayesian [45]	Cityscapes	MSE: 695 L: 3.97
DESIRE [38]	KITTI	Eroare în metri / rata la 1 m de prag: 0.27 / 0.04
	Stanford Drone	Eroare în pixeli 1/5 din rezoluție: 1.29
SS – LSTM [49]	ETH	ADE: 0.95 FDE: 0.23
	UCY	ADE: 0.81 FDE: 0.13
	Town Centre	ADE: 29.01 (0.8 s) FDE: 36.88 (0.8 s)

3.2. Predicția traiectoriei bazată pe rețele neuronale convoluționale

CNN reprezintă un tip de DNN-uri, ce prezintă o performanță excelentă în multe domenii, cum ar fi clasificarea obiectelor și recunoașterea acestora, e.g., cifre scrise de mână, litere și fețe. Arhitectura tipică a CNN se poate vedea în Figura 2.5.1. Această conține un număr mare de straturi convoluționale, straturi neliniare, de punere în comun, de normalizare și complet conectate. Ca un rezultat al optimizării, CNN-urile sunt capabile să înțeleagă caracteristicile unui obiect. Alegerea potrivită a arhitecturii rețelei și a parametrilor face ca aceste caracteristici să includă cele mai semnificative informații cerute pentru identificarea robustă a obiectelor vizate.

Rehder și colaboratorii [51] au propus o metoda care deduce estimează poziții viitoare ale pietonului din imagini folosind și pozițiile cunoscute. Autorii utilizează o rețea de densitate mixtă. Cu scopul de a dezvolta o predicție a traiectoriei, cum ar fi, planificarea mișcării orientate spre obiective, aceștia au folosit două arhitecturi: o rețea înainte-înapoi (în literatura de limbă engleză, "forward–backward") și o rețea tip Markov Decision Process (MDP). Imaginea și poziția pietonilor reprezintă datele de intrare pentru această arhitectură. Procesarea imaginii se realizează printr-o rețea CNN. Concatenarea vectorului de poziție și ieșirea CNN-ului reprezintă datele de

intrare pentru o rețea LSTM. Aceasta face la ieșire o predicție a destinației viitoare probabile a pietonului folosind harta distribuției probabilităților. Pentru a antrenare și evaluarea rezultatelor în lumea reală, autorii au adunat videoclipuri stereo cu etichetări manuale ale pietonilor din mai multe mașini în zone urbane și rezidențiale.

In [52], S. Hoermann și colaboratorii au propus o metodă care combină două rețele: CNN pentru mișcarea pe termen lung și Bayesian pentru estimarea mediul dinamic actual ca intrare. Analiza scenelor se bazează pe o zona de predicție 360° într-o singură rețea neuronală, cu excepția rețelei care realizează segmentarea zonelor statice și dinamice. Folosind celule dinamice rarefiate ("sparse"), autorii au creat o funcție de pierdere bazată pe contracararea pixelilor dezechilibrați din diverse categorii. Ei au demonstrat abilitatea rețelei de a prezice scenarii complexe cu diferite categorii de participanți la trafic (pietonii, ciclişti și autovehicule). În plus, rețeaua poate identifica diferite tipuri de acțiuni în trafic, e.g., viraj stânga sau dreapta și interacțiuni dintre participanți la trafic.

Zhao și colaboratorii [53] au propus rețeaua Multi-Agent Tensor Fusion (MATF) cu o arhitectura codificator-decodificator. Abordarea centrala în spațiu folosește o rețea flexibila care poate fi programata din imaginile contextuale ale mediului cu traiectoriile secvențiale ale agenților, păstrând relațiile spațiale între caracteristici și capturând interacțiuni dintre agenți. Modelele lor codifica traiectoriile trecute ale fiecărui agent independent si decodifica recurent traiectoriile viitoare ale multiplilor agenți, folosind pierderea contradictorie pentru a studia predicții stocastice. Autorii au folosit pe seturile de date ETH-UCY, Stanford Drone Dataset și NGSIM.

In [54], Yi și colaboratorii au propus un model "Behavior-CNN" care este instruit cu date din scene video aglomerate. Autorii au dezvoltat un model de comportament al unui pieton capabil să prezică viitorul traseului parcurs pe jos dar și destinația. Acest model poate deduce caracteristicile de mișcare ale pietonilor. Pentru a creste acuratețea urmăririi pietonilor, modelul poate oferi informații importante bazate pe predicția traseului pietonului.

Doellinger și colaboratorii [55] au folosit CNN sa prezică harți de ocupare medie a oamenilor aflați in mers chiar și în cazuri in care informațiile despre traiectorie nu sunt disponibile. S-a raportat ca metoda lor funcționează mai bine decât alte câteva metode de baza. Autorii au pus in funcțiune un robot mobil pentru a înregistra imagini si a crea un set de date. Munca lor a demonstrat ca distribuțiile ocupației umane pot fi folosite pentru a găsi poziții de așteptare.

In [56], Marchetti și colaboratorii au prezentat modelul numit MANTRA. Acesta este bazat pe rețeaua neuronala cu memorie expandată (în literatura de limbă engleză „Memory Augmented Neural Networks" – MANN). Modelul utilizează conexiunea dintre poziția trecuta și cea viitoare a pietonului si reține cele mai semnificative probe. MANTRA este capabilă sa actualizeze reprezentarea interna a probelor de mișcare în învățarea online. Prin urmare, pe măsura ce sunt colectate alte probe noi, modelul se îmbunătățește. Autorii au condus testarea cercetării pe trei seturi de date de trafic disponibile: KITTI [39], Oxford RobotCar şi Cityscapes [46].

Wang și colaboratorii [57] au prezentat o metodă care se referă la analiza interacțiunilor spațiale dintre diferite obiecte și medii din scenă cu privire la predicția traiectoriei pietonilor. Autorii au combinat poziții ale subiecților și informații 2D-3D despre dimensiunea acestora într-un model pentru a le estima intențiile. Ei au adoptat metoda estimării adâncimii imaginii monoculare pentru a extrage instantaneu și repetitiv harta de adâncime a imaginii din jurul pietonului.

O prezentare comparativă cu cele mai bune soluții din literatură a rezultatelor CNN pentru PTP este prezentată în Tabelul 3.2.1.

Tabel 3.2.1. Comparație a rezultatelor CNN pentru PTP

Metodă	**Bază de date**	**Rezultate**
MATF [53]	ETH	ADE (Determinis-tic): 0.64 ADE (Stochastic): 0.48 FDE (Deterministic): 1.26 FDE (Stochastic): 0.90
	Stanford Drone	ADE (Determinis-tic): 30.75 ADE (Stochastic): 22.59 FDE (Deterministic): 65.90 FDE (Stochastic): 33.53
MANTRA [56]	KITTI	ADE: 0.16 (1s) FDE: 0.25 (1s)
	Cityscapes	ADE: 0.49 FDE: 0.79
	Oxford RobotCar	ADE: 0.31 (1s) FDE: 0.35 (1s)
MI – CNN [57]	MOT16	ADE: 18.25 FDE: 21.70
	MOT20	ADE: 16.63 FDE: 19.34

3.3. Predicția traiectoriei bazată pe rețele neuronale generative

În ceea ce privește urmărirea, GAN reduce întreruperile care de obicei apar la multe modele convenționale de prezicere a traiectoriei. Ea atenuează necesitatea de a calcula caracteristici de aspect costisitoare din punct de vedere computațional. Observațiile sunt produse și actualizate de o componentă generativă; prin urmare, cele mai puțin actualizate caracteristici vor fi eliminate. Cu scopul de a procesa și a clasifica secvențele candidate, se folosește concomitent (în același model), o componentă LSTM cu un model generativ-discriminativ. Această metodă poate conduce la modele de înaltă precizie în modelarea comportamentului uman, în special a celui de grup. În ultimul timp, mulți autori au folosit arhitectura GAN pentru a realiza o abordare de tip multimodal în rezultatul de predicție.

Fernando și colaboratorii [58] au trecut fiecare cadru printr-un generator GAN. Ca răspuns, acesta returnează o hartă a probabilității pentru fiecare pixel. Aceasta a fost segmentată în continuare. Predicția a fost făcută atât pentru traiectoriile scurte (folosit pentru asocierea datelor), cât și pentru cele lungi (folosit pentru actualizarea traiectoriei obiectelor) perspective pe termen.

Un model ce consideră interacțiunea între membri grupului (în literatura de limbă engleză „socially-aware" GAN) cu RNN-uri a fost propus de Gupta și colaboratorii [59] pentru predicția mișcării pietonilor în medii dinamice. Autorii au pornit de la ideea că pietonii se influențează reciproc în mod uniform. Aceștia au inclus impactul tuturor agenților din cadru-imagine dar și contextul acestuia. Estimările plauzibile ale traiectoriilor ce consideră interacțiunile pietonilor au fost prezise prin antrenarea modulului discriminator adversial. A fost folosită o arhitectură de tip codificator-decodificator. A fost adoptat un nou mecanism de eșantionare pentru agregarea de informații. Pentru antrenament autorii au folosit seturile de date ETH si UCY. Ca metrici, ei au folosit „Average Displacement Error" (ADE) si „Final Displacement Error" (FDE), cu o metodologie de evaluare similară cu [33], pentru 8 (3:2 s) și 12 (4.8 s) pași de timp.

Kosaraju și colaboratorii [60] au urmat aceeași idee a interacțiunii sociale găsită în [58] și [59]. Aceștia au considerat contextul scenei și comportamentul multimodal al pietonilor și au propus rețele de atenție de tip graf care codifică acești factori. Mai mult, arhitectura GAN va estima traiectoriile pietonilor. Pentru a testa această soluție, autorii au folosit seturile de date ETH si UCY pentru că acestea conțin informații etichetate despre traiectoriile pietonilor și interacțiunea pietonilor în cadre publice.

Amirian și colaboratorii [61] au propus o metodă care se bazează pe Info-GAN [62] pentru predicția traiectoriei pietonilor într-un interval de câteva secunde în viitor. Funcția de pierderi L2 a fost înlocuită de funcția de cost bazată pe entropie, din cauza impactului negativ asupra capacității de generalizare a rețelei. Rezultatele au fost raportate pentru seturile de date ETH si UCY.

Sadeghian și colaboratorii [63] au dezvoltat un model interpretabil de predicție a traiectoriei pietonilor bazat pe GAN, numit SoPhie care combină un mecanism de atenție socială cu atenție fizică. Stratul de ieșire generează traiectorii posibile din punct de vedere social și fizic folosind un LSTM bazat pe GAN.

O prezentare comparativă cu cele mai bune soluții din literatură a rezultatelor GAN pentru PTP este prezentată în Tabelul 3.3.1.

Tabel 3.3.1. Comparație a rezultatelor GAN pentru PTP

Metodă	Bază de date	Rezultate
DGMMPT [58]	Town Centre	MOTA: 42.5 MOTP: 69.8
Social GAN [59]	ETH	ADE: 0.39 / 0.58 FDE: 0.78 / 1.18
Social-BiGAT [60]	ETH	ADE: 0.69 FDE: 1.29
Social Ways [61]	ETH	ADE: 0.39 FDE: 0.64
	UCY	ADE: 0.55 FDE: 1.31

3.4. Predicția traiectoriei bazată pe rețele neuronale de tip graf

Rețeaua neuronală de tip graf este un model de învățare profundă care este aplicat direct pe arhitectura grafului. În contextul GNN-ului, cele mai multe grafuri sunt atribuite (cu atribute de noduri și margini, și/sau caracteristici globale). În înțelegerea reprezentării orientate pe grafuri folosind GNN-uri, există trei elemente principale: noduri, margini, și actualizări globale [64].

„Graph Convolutional Neural Networks" (GCNNs) pot fi împărțite în metode spectrale [65], [66], [67] și spațiale [68], [69]. Primele utilizează o reprezentare spectrală a grafurilor pentru a crea convoluții, întrucât cea din urmă definește convoluții de pe graf, lucrând pe un grup de vecini apropiați din punct de vedere spațial. Filtrele utilizate de metodele spectrale sunt definite de o bază proprie laplaciană, care este determinată cu ajutorul structurii grafice. Ca rezultat, un model instruit pe o structură particulară nu poate fi transferată imediat spre un alt graf diferit din punct de vedere structural. În orice caz, modelarea interacțiunii sociale umane necesită un graf variabil în timp. În concluzie, metodele spectrale sunt ineficiente în prezicerea traiectoriei pietonilor. Prin urmare, metoda propusă aparține categoriei soluțiilor spațiale.

Velickovi *et al*. [70] a introdus așa numitele „Graph Attention neTworks "(GATs). Acest lucru permite integrarea unei arhitecturi de tip "self-attention-based" în orice tip de structură de date caracterizată ca un grafic. Aceste rețele îmbunătățesc caracteristicile GCNN [71] prin permisiunea ca modelul să atribuie în mod direct fiecărui nod de rețea o valoare dinamică. Kosajaru *et al*. [60] a propus "social-BIGAT", care generează interacțiuni importante. Agentul format înțelege reprezentări ale interacțiunii sociale atât din dimensiunea temporară, cât și din cea socială, argumentând că reprezentarea timpului și a elementelor sociale separat pot duce la o soluție suboptimă.

Mohamed *et al*. [72] a propus o abordare prin modelarea interacțiunilor dintre pietoni ca o reprezentare tip graf folosită pentru extragerea de caracteristici semnificative. Acestea conțin informații despre reprezentarea compactă a istoricului traiectoriei pietonului observat. Pentru a prezice traiectoriile viitoare a tuturor pietonilor, autorii au creat un strat secundar („time–extrapolator CNN") cu intrarea T

x P x N (P- dimensiunea poziției pietonului, N-numărul pietonilor, T- intervalul de timp). TXP–CNN lucrează direct asupra dimensiunii temporale a înglobării grafului și îl extinde atât cât este necesar pentru predicție. Pentru a evalua această metodă, autorii au folosit seturile de date ETH și UCY. Abordarea lor („Social-STGCNN") a îmbunătățit metoda STGAT [73] prin obținerea datelor traiectoriei de la fiecare pieton, modelarea interacțiunilor dintre pietoni cu ajutorul unui GNN și calcularea unui grafic ponderat utilizând GAT. Vemula și colaboratorii [74], au propus pentru estimarea traiectoriei pietonilor un grafic spațio-temporar cu noduri, acestea reprezentând oamenii din scenă.

Li și colaboratorii [75] au conceput „Conditional Neural Generative System" (CNGS) pentru a estima traiectoriile viitoare ale vehiculelor pentru călătorie inteligentă și sigură. Din cauza unei metode de învățare nesupervizată și a unei programări dinamice estimarea precisă cu această soluție constituie o provocare.

O prezentare comparativă cu cele mai bune soluții din literatură a rezultatelor GNN pentru PTP este prezentată în Tabelul 3.4.

Tabel 3.4.1. Comparație a rezultatelor GNN pentru PTP.

Metodă	**Bază de date**	**Rezultate**
Social – STGCNN [72]	ETH	ADE: 0.64 FDE: 1.11
	UCY	ADE: 0.44 FDE: 0.79
STGAT [73]	ETH	ADE: 0.50 FDE: 0.89
	UCY	ADE: 0.38 FDE: 0.79
CGNS [75]	ETH	ADE 0.66 FDE 1.16
	UCY	ADE: 0.38 FDE: 0.84
	Stanford Drone	ADE:15.84 FDE: 25.17

3.5. Concluzii

În cadrul acestui capitol s-au examinat lucrările reprezentative din domeniu ("State-of-the-Art", SOTA), cu precădere metodele de predicție care folosesc rețele neuronale profunde. În legătură cu tematica prezentului capitol am publicat lucrarea [30] în care am se analizează cele mai recente soluții bazate pe învățare profundă pentru problema predicție traiectoriei pietonilor împreună cu senzorii utilizați și metodologiile de procesare aferente. Se efectuează totodată o prezentare generală a seturilor de date disponibile, a indicatorilor de performanță utilizați în procesul de evaluare. Lucrarea expune și problemele care nu sunt încă rezolvate precum si potențiale noi direcții de cercetare.

4. PROCESAREA NEURONALĂ A INFORMAȚIEI SENZORIALE ÎN PROBLEMA PREDICȚIEI TRAIECTORIEI PIETONILOR

Un vehicul autonom (VA) reprezintă un ansamblu echipat cu senzori (LiDAR, radare, camere etc.) și sisteme capabile să-l controleze automat astfel încât să poată rula pe drum fără intervenție umană. Pentru a îndeplini astfel de sarcini, sistemul de control trebuie să poată detecta obiecte în apropierea vehiculului și să estimeze traiectoriile lor viitoare și parametri cinematici ai mișcării acestora [76].

Senzorii și sistemele care procesează informațiile primite oferă asistență șoferilor, semnalizându-le diferite circumstanțe pentru minimizarea riscul de expunere. Acestea pot chiar automatiza sarcinile de conducere pentru a elimina erorile umane [77]. Pentru a colecta informații din mediul exterior, VA utilizează senzori numiți exteroceptivi. Procesarea informațiilor primite duce la recunoașterea altor participanții la trafic (pietoni, vehicule, etc.) și a obiectelor din apropiere. În ultimii ani, senzorii externi ai VA au câștigat importanță, în special datorită dezvoltării procesării imaginilor și camerelor, deoarece aceste sisteme permit o gamă largă de aplicații [78].

Conform [79], asistența la conducere autonomă se bazează în principal pe sisteme legate de procesarea imaginilor provenite de la camere. LiDAR reprezintă cel mai necesar senzor în sistemele auto. În contrast cu camerele, acesta este caracterizat de o detecție omnidirecțională și nu este afectat de condițiile de lumină. Un sfert din piața senzorilor auto este reprezentat de senzorii ultrasonici și radar, în timp ce alți senzori exteroceptivi, cum ar fi microfoanele, acumulează 18% din piață (vezi Figura 4.1).

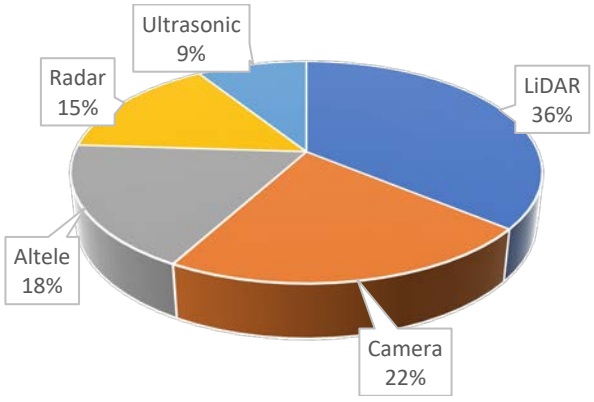

Figura 4.1. Predicția creșterii pieței senzorilor auto (rata de creștere anuală compusă (CAGR), 2017-2022) pe senzorii exteroceptivi, conform [79].

Un set de senzori amplasați pe vehicule captează date care sunt ulterior procesate pentru a obține o reprezentare digitală a mediului înconjurător. Această informație este folosită pentru evaluarea riscurilor precum și planificarea și prezicerea traiectoriilor, ceea ce conduce la controlul mișcării vehiculului.

Percepția oferă mijloacele prin care vehiculul poate obține informații referitoare la ce se întâmplă în mediul său de operare. Într-un scenariu de conducere autonomă, există numeroși participanți la trafic în jurul vehiculului: pietoni, bicicliști, motocicliști, alte vehicule, etc. O provocare majoră constă în a percepe toate aceste elemente într-un mod continuu și precis, fără a avea alarme false (raportarea de obstacole inexistente) sau lipsa de detecție (adică omisiunea de obstacole reale). În acest scop, se pot utiliza mai mulți senzori exteroceptivi activi și pasivi, printre care diferite tipuri de camere video, LiDAR și RADAR. Algoritmii de detecție de ultimă generație oferă și clasificare obiectelor percepute. În prezent, aceștia pot detecta obiecte utilizând date de la o gamă extinsă de senzori [80], [81], [82].

Cu toate acestea, nu există o tehnologie de senzori unică capabilă să ofere informații spațio-temporale precise și complete despre tot ceea ce înconjoară vehiculul, fiecare având propriile avantaje și dezavantaje [83]. Soluțiile aplicate în prezent combină astfel diferite tipuri de senzori prin intermediul unui proces de interconectare a mai multor senzori (fuziune). Astfel este posibilă exploatarea caracteristicilor lor majore într-un mod concludent, reducând astfel incertitudinea asociată fiecăruia dintre ei. Ca exemplu, Figura 4.2 prezintă configurarea senzorilor de pe un autoturism model Renault Zoe, care este utilizat pentru înregistrarea setului de date nuScenes [48]. Acesta utilizează 6 camere, 5 RADAR-uri și un LiDAR pentru a înregistra date de senzori sincronizate pentru cercetarea și dezvoltarea algoritmilor de percepție și navigație.

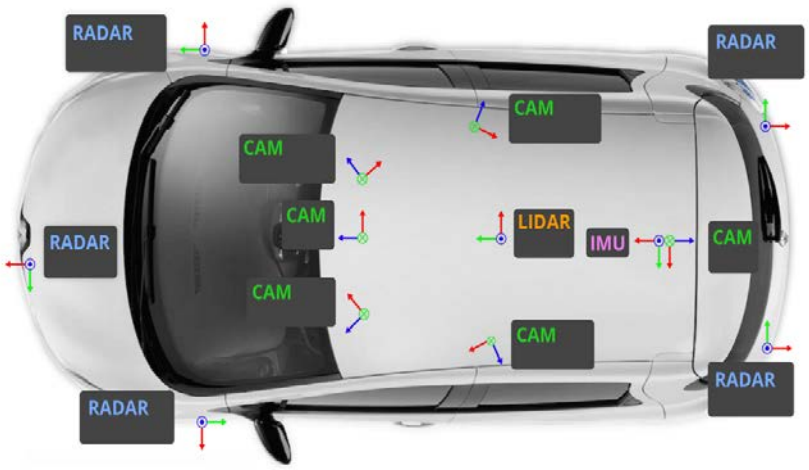

Figura 4.2. Configurarea senzorilor pe un Renault Zoe folosit pentru alcătuirea setului de date NuScenes [48].

4.1. Radar

Scopul principal al sistemelor radar auto este de a determina țintele de interes (de exemplu, pietoni, mașini sau bicicliști) și de a estima dimensiunea, mișcarea, distanța, viteza relativă și direcția acestora în raport cu radarul [84]. Folosind undele electromagnetice reflectate, care sunt transmise și recepționate, radarul monitorizează întreaga imagine a mediului. Până în prezent, datorită unor inconveniente, datele radar au fost utilizate doar în câteva cazuri de predicție a traiectoriei pietonilor.

Figura 4.1.1. Radarul AGD326 este un detector de pietoni de 24 GHz care poate fi utilizat pentru optimizarea etapei de traversare. (Sursă imagine: www.agd-systems.com; accesat la 30 octombrie 2021).

Sistemele radar implică transmiterea unui puls de undă radio, care poate ricoșa pe țintele din fața vehiculului. S-a observat o problemă referitoare la interferența diferitelor informații datorită pulsului reflectat care ajunge mai târziu la senzor.

Schimbarea frecvenței săvârșită datorită efectului Doppler poate îmbunătăți măsurarea vitezei relative a obiectelor în mișcare (de exemplu, a pietonilor). Sistemele radar auto funcționează în mod obișnuit la benzile de frecvență de 24, 79 și 77 GHz (pentru cele mai recente generații de radar) și pot acoperi unghiuri între 9° și 150° [85]. Radarul poate funcționa în condiții nefavorabile (de exemplu, ploaie, praf, zăpadă sau ceață) [86] și poate acoperi trei intervale de distanță: lung (10-250 m), mediu (1-100 m) și scurt (0,15-30 m). Distanța joacă un rol crucial. Din această cauză, pentru determinarea acesteia, se folosește întârzierea în timp a călătoriei dus-întors a undelor electromagnetice care trebuie să circule către și de la țintă.

Pentru a asigura o direcționare controlată a undei emise și o distincție a obiectelor în funcție de viteză și distanță, ultimele modele de radare auto utilizează tehnologia "*Frequency-Modulated Continuous Wave*" (FMCW) împreună cu formarea digitală a fasciculelor ("digital beamforming") [87]. Estimarea direcției se bazează pe obținerea datelor referitoare la undele reflectate din diferite dimensiuni, obținute prin combinarea variabilelor de frecvență, spațiu-timp (figura 4.1.1). De exemplu, în [88],

a fost propusă o metodă pentru a învăța și a prezice dinamica țintelor în mișcare (pietoni), prin aplicarea datelor măsurate de radarul de rezoluție FMCW cu o frecvență rapidă, direct modelelor neuronale de tip LSTM și CNN. În arhitectura propusă, datele radar au fost măsurate și transformate în imagini de distanță și viteză prin transformata Fourier bidimensională. Metoda propusă a fost testată utilizând datele obținute de radarul FMCW cu impuls rapid al Hyundai Mobis.

Mecanismul radarului auto este influențat de unele informații reflectate nedorit, împreună cu undele reflectate de la țintele de interes. Această cantitate de date, sub formă de zgomot sau perturbație, ce poate rezulta dintr-o reflecție de pe pereți, deșeuri de pe drum sau balustrade, duce la o percepere greșită a mediului înconjurător. Prin urmare, au fost dezvoltați diferiți algoritmi pentru a atenua efectul acestor perturbații. "*Space-time adaptive processing*" (STAP) [89], precum și "*Constant false alarm rate*" (CFAR) [90], [91] reprezintă doi algoritmi importanți care pot fi utilizați în această problemă. Pentru o detectare precisă a țintei de interes în prezența zgomotului, trebuie să se ia în considerare valoarea de referință, care trebuie stabilită corect, în funcție de cantitatea de zgomot din sistem (pe măsură ce zgomotul se extinde, trebuie stabilită o valoare mai mare).

Ding și colaboratorii [92] au propus o tehnică pentru extragerea traiectoriilor micro-Doppler ale pietonilor din bruiaj radar de undă continuă. Aceștia au utilizat CFAR pentru a estima, în timp real, parametrii de zgomot și pentru a ajusta limita de filtrare. După această etapă, detectarea falsă a pietonilor poate fi mult suprimată și unele semnale sunt mai ușor detectate. Pentru "eliminarea zgomotului", autorii au utilizat algoritmul "CLEAN" [93], în care multiple componente ale ecoului radar de undă continuă sunt extrase secvențial, fiecare parametru este estimat, iar componentele mai puternice sunt înlăturate.

Efectul Doppler este în esență schimbarea frecvenței emise de un generator în mișcare percepută de un receptor static. Se poate defini „efect micro-Doppler" modificarea frecvenței datorită mișcărilor membrelor unui pieton [94].

Acest concept important este cunoscut și ca „semnătura micro-Doppler" care definește un model periodic urmat de viteză în timp datorită mișcării periodice a membrelor. Pentru a detecta în mod specific mersul pe jos al pietonilor, sunt folosiți diferiți algoritmi, care pot include extragerea informației și identificarea naturii acesteia. În [95], autorii au dezvoltat o metodă ce folosește o combinație de radare auto și Doppler pentru a detecta componentele de mișcare ale pietonilor prin aplicarea distribuției Gauss și a unui filtru Kalman. Prin analizarea spectrogramei Fourier a frecvenței Doppler, se poate detecta mișcarea umană periodică. Pentru a testa metoda, autorii au colectat patru seturi de date din diferite medii.

În [96] a fost propusă o altă metodă de predicție a comportamentului de deplasare ale pietonilor. S-au utilizat măsurători radar simultane și senzori de captură a mișcării pentru înregistrarea digitală a mișcărilor pentru fiecare parte a corpului. Pentru detecție, autorii au folosit un algoritm CFAR în care fiecare celulă Doppler este estimată printr-o limită de detecție. De asemenea rezultatele și caracteristicile specifice ale comportamentului de mișcare sunt furnizate de la un singur pieton.

Dubey și colaboratorii [97] au prezentat un cadru Bayesian pentru a integra modalitățile de mișcare și apariție ale pietonilor în sistem. Pentru a distinge și învăța caracteristicile pentru fiecare clasă, ei au creat o învățare metrică de distanță printr-un vector latent de caracteristici. Traiectoriile pietonilor au fost determinate prin

interpolarea punctelor individuale presupunând viteze constante prin combinarea sistemelor de urmărire și clasificare. Pentru a genera diferite scenarii, autorii au folosit o aplicație MATLAB numită "*driving scenario designer*" care permite proiectarea scenariilor de conducere. Autorii lucrării [98] au prezentat o metodă de estimare a direcției de mișcare a pietonilor în scenarii complexe folosind semnătura *micro-Doppler* obținută prin radarul „Multiple Input Multiple Output" (MIMO) auto. Această metodă observă pietonii dintr-un singur unghi și extrage informații despre direcția de mișcare din semnătura *micro-Doppler* utilizând metode de regresie. Pentru a testa metoda propusă, autorii au folosit simulări de scenarii auto, unde pietonul este observat în mai mulți senzori radar de intrare/ieșire.

Odată cu creșterea cererii pe piață a tehnologiei de asistență pentru conducerea autonomă, radarul auto face pași importanți către a deveni o soluție mai robustă pentru problema prezicerii traiectoriei pietonilor. Această transformare implică toate aspectele radarului auto, inclusiv conceptul de sistem, modulația și procesarea semnalului.

4.2. LiDAR

Au fost dezvoltați senzori "Light Detection And Ranging" (LiDAR), care se bazează pe reflexia laser pentru detectarea obiectelor din jurul vehiculului. Este cunoscut faptul că senzorii LiDAR emit impulsuri periodice de lumină, de exemplu, la fiecare 30 ns. Fasciculul de lumină emis de senzor are o lungime de undă tipică de 905 nm și este coaxial cu componenta de lumină reflectată [99]. Tehnologia LiDAR are o mare precizie datorită acțiunii circulare și verticale, care permite obținerea de modele 3D - ilustrații spațiale ale coordonatelor obținute prin înregistrarea distanței și direcției impulsurilor de lumină care se întorc, sub forma de puncte de date, care apoi sunt organizate în grupări de puncte. Senzorii LiDAR facilitează o colectare inovatoare a datelor la nivelul traiectoriei în cazul condițiilor de trafic mixt. Utilizând grupări de puncte 3D [100], acești senzori sunt capabili să raporteze locația precisă a obiectelor și pot acoperi unghiuri de până la 42° în ceea ce privește gama verticală de vizibilitate și 360° în ceea ce privește gama orizontală din jurul vehiculului, fără a fi influențați de condițiile de lumină (vezi Figura 4.2.1). Un avantaj important al tehnologiei LiDAR este sensibilitatea redusă la condițiile de lumină și atmosferice. Senzorii LiDAR care au un preț mic (începând de la 100$) sunt caracterizați de un singur fascicul de lumină și de un consum redus de energie (începând de la 8W). Între timp, cele mai noi modele de senzori LiDAR au o rezoluție mai bună a grupării de puncte, utilizând matrice laser (până la 128 de fascicule). Comparativ cu alți senzori, cum ar fi camerele digitale, LiDAR pot duce la o percepție mai bună în toate condițiile de iluminare, ceea ce îl face remarcabil în cazul vehiculelor autonome. Deși acuratețea datelor este redusă de condițiile meteorologice nefavorabile, cum ar fi ploaia sau ceața, în condiții meteorologice moderate, senzorul LiDAR poate fi utilizat corespunzător în aplicații cu frecvență mare (de exemplu, crearea unui strat de percepție în cazul unui vehicul autonom).

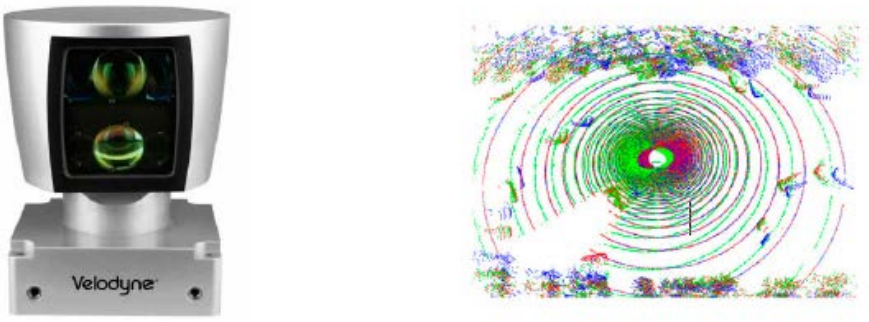

Figura 4.2.1. A Senzorul 3D-LiDAR poate fi utilizat pentru raze scurte, medii, telescopice sau combinații (dual scurtă, dual medie). Aici, un senzor Velodyne HDL-64E și gruparea de puncte generat. (Sursa: www.velodynelidar.com).

Sistemul LiDAR poate funcționa în toate condițiile de iluminare, fiind capabil să genereze hărți locale pentru un vehicul. Acestea pot fi utile în estimările de comportament, în ceea ce privește mediul și vehiculele din jur. Mai exact, predicțiile de comportament în mediul înconjurător au un rol critic în planificarea căii predictive a unui vehicul autonom; de exemplu, se poate prezice posibilitatea unui vehicul din față de a face un anumit viraj.

În prezent, senzorii LiDAR sunt utilizați în principal pentru detectarea obstacolelor, utilizatorilor de drum și marcajelor de bandă [101], [102], [103] și [104]. Utilizând grupări de puncte intense, senzorii LiDAR de la bordul vehiculului pot crea o descriere completă a obiectelor, în timp ce senzorii LiDAR de pe marginea drumului furnizează puncte de date vagi. Figura 4.2.2 prezintă un exemplu de detecție a grupărilor de puncte LiDAR a pietonilor. Caracteristicile datelor depind de distanța dintre LiDAR și pieton.

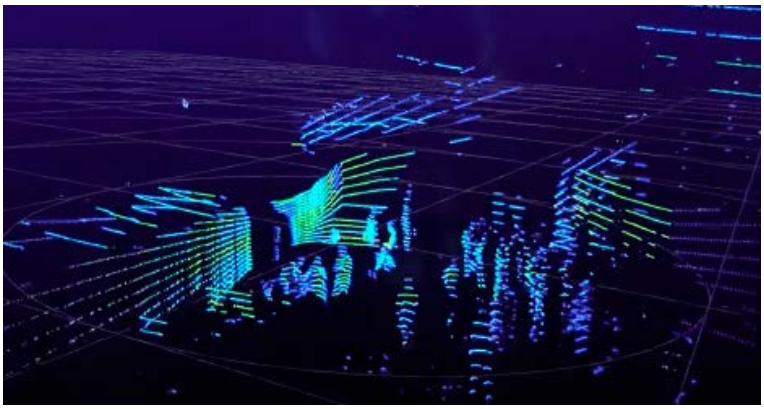

Figura 4.2.2. Date LiDAR ale pietonilor, capturate cu ajutorul senzorului Velodyne HDL. (Sursă imagine www.velodynelidar.com)

Informația de adâncime furnizată de LiDAR a fost utilizată direct de mai mulți cercetători pentru a grupa punctele, estimând locația pietonilor ca o regiune încadrată 3D. Rezoluția angulară verticală și orizontală care caracterizează acest tip de senzor influențează densitatea grupărilor de puncte. Figura 4.2.3 prezintă forma ansamblului de puncte din scanările 3D LiDAR ale pietonilor la distanțe diferite între pieton și senzor. Proiecțiile XYZ sunt calculate dintr-un eșantion care are suficiente puncte.

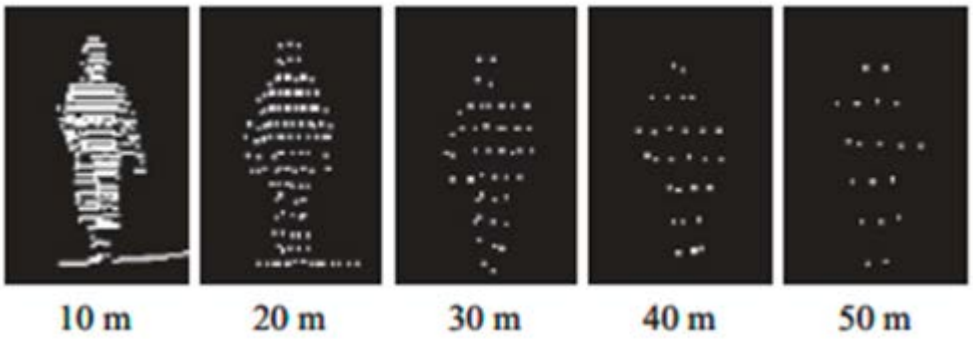

Figura 4.2.3. Caracteristicile grupărilor de puncte 3D pentru pietoni la diferite distanțe.

În [105] autorii au dezvoltat un sistem complet de procesare a datelor LiDAR la marginea drumului utilizând date într-un sistem de coordonate cartezian 3D pentru a prezice traiectoriile viitoare ale pietonilor în timp real. Traiectoria viitoare avea informații despre poziția XYZ, numărul total de puncte de date, distanța dintre LiDAR și pieton, viteza, ID-ul de urmărire, numărul de cadre și o etichetă asociată fiecărui pieton. Ei au folosit un model de senzor LiDAR VLP-16 de la compania Velodyne Lidar din San Jose, CA, SUA, cu 16 lasere rotative orizontal și un motor intern în coordonatele XYZ. Pentru a clasifica datele secvenței de la senzor, cum ar fi traiectoriile clasificărilor bazate pe caracteristici, autorul a folosit algoritmul „Naïve Bayes" [106] aplicat la intrarea modelului pentru diferite intervale de probabilitate și o combinație optimă de caracteristici.

Pentru a extrage traiectoriile pietonilor din datele LiDAR, este necesară o procedură de preprocesare a datelor pentru a efectua filtrarea în fundal [107], gruparea [108], clasificarea [109] precum și urmărirea obiectelor [110], [111], așa cum este ilustrat în Figura 4.2.4.

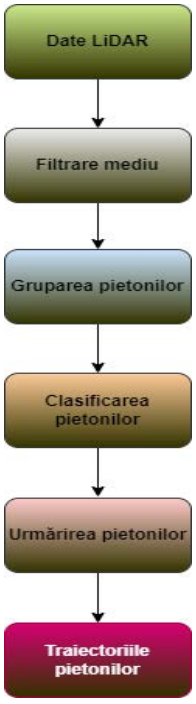

Figura 4.2.4. Diagrama de procesare a datelor

Bu și colaboratorii [112] au propus o metodă care poate efectua estimarea 3D a pietonilor pe baza datelor LiDAR 2D și a camerei monoculare. Această metodă constă din trei sub-rețele (rețeaua de orientare, rețeaua de propunere regională și „PredictorNet") pentru a efectua predicții mai precise cu casete delimitatoare. Rețeaua de orientare redimensionează datele pentru a determina unghiurile de orientare. Rețeaua de propunere regională preia hărțile de caracteristici și intrările de la rețeaua de orientare și generează casete delimitatoare neorientate ale pietonilor. „PredictorNet" utilizează harta de caracteristici a pietonilor obținută din rețelele anterioare pentru a face o predicție și o clasificare finală.

În [113], Völz și colaboratorii au prezentat diferite arhitecturi care pot fi utilizate pentru a identifica intențiile pietonilor de a traversa strada la o trecere de pietoni dată. Aceștia au introdus o arhitectură densă de rețele neuronale pentru a clasifica intențiile pietonilor pe baza caracteristicilor de la mai multe intervale de timp, ajungând la o acuratețe de validare de 96,21%. Pentru a analiza mai precis caracteristicile seriei temporale, au utilizat rețele neuronale recurente, care au permis alimentarea datelor înapoi în rețelele neuronale dense, ajungând la o acuratețe de validare de 95,77%. Folosind rețele neuronale convoluționale, caracteristicile imaginii sunt extrase prin convoluția filtrelor antrenate de-a lungul imaginii din LiDAR, iar aceste caracteristici sunt folosite pentru a clasifica datele. Pentru a implementa aceste arhitecturi în Python, autorii au folosit mediile de implementare DNN Theano [114] și Lasagne [115].

O soluție interesantă pentru estimarea online a poziției, vitezei și accelerării pietonilor a fost propusă de Mohammadbagher și colaboratorii [116]. Pentru a

identifica pietonii în imaginea capturată de LiDAR, au folosit o arhitectură de rețea neuronală profundă bazată pe detectarea obiectelor (YOLOv3 Pytorch). Pentru a localiza vehiculul ego și, prin urmare, pietonul de interes în imagine, s-au folosit informațiile de odometrie capturate de la senzorii GPS/IMU. Autorii au testat modelul în două experimente și în diferite scenarii de imagine.

În timp ce sistemele LiDAR de pe extremități sunt capabile să funcționeze independent, senzorii LiDAR de pe bordul mașinii au nevoie de alte componente, cum ar fi radarul sau camerele, pentru a sprijini sistemele de conducere autonomă. Costurile ridicate, împreună cu aplicațiile limitate referitoare la implementarea senzorilor LiDAR de pe extremități sunt responsabile pentru utilizarea lor limitată, chiar dacă acest tip de senzor poate furniza o colecție de date în timp real și la nivel de traiectorie. În [117], autorii au propus un subsistem pentru gestionarea pietonilor în trecerile de pietoni prin aplicarea directă a metodelor de învățare profundă asupra datelor provenite din fuziunea cameră video - senzorii LiDAR. Pentru a detecta pietonul, autorii au folosit o rețea neuronală convoluțională. Pentru identificarea pozițiile pietonilor în imaginile LiDAR, au utilizat datele din ansamblul de puncte LiDAR.

Cu toate acestea, o implementare extinsă a senzorilor LiDAR va fi posibilă în curând datorită progresului recent în tehnologia LiDAR și datele disponibile public (de exemplu, concursul de predicții nuScenes și „Lyft Motion Prediction" pentru vehicule autonome). Luând în considerare faptul că tehnologia LiDAR de pe extremități nu poate folosi direct metodele aplicate procesării datelor LiDAR de la bordul vehiculului, este crucial să se analizeze conceptele fundamentale ale tehnologiei LiDAR, inclusiv strategiile de instalare și tehnici eficiente privind procesarea datelor online și offline.

4.3. Camera

Scopul senzorului de vedere este multifuncțional: acesta poate fi utilizat pentru monitorizarea șoferului și a pasagerilor, dar și pentru detectarea obiectelor (semne de circulație, alte vehicule) din mediul înconjurător [118]. Noile concepte sunt reprezentate de fuzionarea informațiilor din interior și exterior și de sistemul de vedere panoramică („Surround View Camera" – SVC), constând din patru camere – față, spate și pe oglinzile retrovizoare exterioare, fiind capabil, printre alte sarcini, să recunoască pietonii din apropiere într-un stadiu incipient (vezi Figura 4.3.1).

Camerele bazate pe tehnologia CMOS pot opera în multiple benzi spectrale, de exemplu, vizibile (VIS), infraroșu-apropiat (NIR) sau infraroșu scurt/lung, fiecare oferind caracteristici utile în diverse scenarii de trafic (zi, noapte, ceață, zăpadă, etc.). Acestea pot fi clasificate ulterior în funcție de rezoluția lor, câmpul vizual sau numărul de camere video (mono, vedere stereo, SVC) [119].

Principalul avantaj al acestui tip de senzor constă în prețul redus și consumul de energie redus, în timp ce dezavantajele sunt legate de dependența performanței de condițiile de iluminare și trafic.

Figura 4.3.1. Detectarea pietonilor cu ajutorul camerelor SVC. O cameră de vedere „surround Valeo 360" ar putea oferi o vedere tridimensională a mediului. (Sursa imagine: www.valeo.com/en/360-vue/ și www.fordclubsweden.se; accesat pe 12 martie 2021).

În ciuda cantității crescute de informații furnizate, există un interes tot mai mare în analizarea comportamentului factorilor (oameni, vehicule) din date video provenite de la cameră. În acest context, problema predicției traiectoriei/traseului este prezentată în principal în literatură în două situații distincte: (1) când datele video sunt furnizate de camerele unui sistem de supraveghere - terestru sau aerian - și (2) când intrarea provine din sistemul senzorial al unui autovehicul. Abordarea dintâi constă în principal în aplicații de securitate, în timp ce cea de-a doua se referă la siguranța activă a unui autovehicul. Cu toate acestea, principiile de funcționare prezentate în aceste două cazuri sunt interschimbabile, adică o metodă utilizată în predicția traiectoriei pentru un sistem de supraveghere poate fi utilă pentru o aplicație auto. De asemenea, predicția locației viitoare din cameră ar putea fi realizată din mai multe puncte de vedere: de aproape sau de departe și o vedere oculară-izometrică. De interes deosebit este cazul predicției traiectoriei mișcării pietonilor din perspectiva oculară la distanță mare [120].

Majoritatea abordărilor inițiale au utilizat paradigme clasice/statistice și funcții dinamice create manual pentru a estima riscul de coliziune dintre vehicul și pieton. Un caz tipic este utilizarea filtrării Kalman [121] pentru utilizatorii vulnerabili de drumuri. Un astfel de exemplu este lucrarea lui Keller și colaboratorii [122] în care se detectează mișcarea de oprire utilizând două abordări de filtrare Kalman, versus două metode bazate pe stereo-viziune utilizând un flux optic dens. O estimare a traiectoriei pietonului bazată pe fișiere video înregistrate este prezentată în [123]. Ea bazează pe abordări statistice (modele Gauss mixte / extragerea planului îndepărtat pentru segmentare, binarizare Otsu, siluetă și extracție de schelet) în care caracteristicile mișcării, cum ar fi poziția, viteza și accelerația, sunt calculate de la scheletul uman. Pentru ultimul pas al prezicerii destinației pietonului, au fost testate mai multe modele, cum ar fi *„Multinomial Logistic Regression" (MLR)* şi *„Multi-Layer Perceptron" (MLP)*. „S*upport Vector Machine"* (SVM) a oferit cea mai bună valoare mediană a metricii aria de sub curbă („Area Under the Curve" , AUC), de 87%.

Lucrările anterioare au luat în considerare prezicerea traiectoriilor folosind o singură cameră. Există și alte abordări care prezic traiectoriile pietonilor pe baza mai multor camere care nu se suprapun, de exemplu lucrarea [124]. Aici, *„Multi-Camera Trajectory Forecasting"* (MCTF) se efectuează folosind multiple metrici, cum ar fi

distanța cea mai scurtă în lumea reală sau traiectoria cea mai similară. O rețea LSTM și un modul „*Gated Recurrent Unit*" (GRU), ambele cu 128 de unități ascunse, oferă cele mai bune rezultate: 74,4% și, respectiv, 75,1% cea mai bună acuratețe utilizând baza de date „*Warwick-NTU Multi-camera Forcasting*" (WNMF).

Pentru imaginile prelevate de la un senzor de tip cameră, învățarea profundă a devenit metoda de ultimă oră pentru tipurile de date 2D și 3D, așa cum este rezumat mai jos. Majoritatea abordărilor bazate pe camere sunt formulate ca previziuni ale mișcării ego, de exemplu, [125], unde problema traiectoriei vehiculului este rezolvată prin segmentarea semantică a datelor furnizate de o singură cameră monocromă. Autorii propun o soluție completă bazată pe FlowNet [126], AtrousCNN [127] și Spatial Pyramid Pooling (SPP) din Deeplab [128] și au obținut, folosind setul de date KITTI [39], o acuratețe de 89,00% și IoU de 72,25% pentru un orizont de predicție de 5 secunde.

În Loukkal și colaboratorii [129] au subliniat importanța practică a unui sistem în care sunt folosite doar camere monoculare. Au propus o arhitectură bazată pe două soluții de rețele neuronale profunde în care, mai întâi, s-a realizat o transformare de la imaginea camerei la o hartă a grilei de ocupare din vederea de sus. Apoi, o a doua etapă a efectuat planificarea mișcării folosind o configurație LSTM codificator-decodor. Rezultatele raportate au arătat o ADE de 0,78 pentru modelul holistic complet, comparativ cu setul de date nuScenes [48].

Alte abordări care efectuează predicții privind locația și traiectoria viitoare a unei persoane folosind camere montate pe vehicule sunt prezentate în [45] și [130]. În mod similar, camerele purtabile au fost utilizate în [131], [132]. Autorii din [132] au descris un sistem bazat pe codificator-decodificator LSTM care utilizează locațiile și pozițiile persoanei vizate și datele de măsurare inerțială (IMU) de la o cameră portabilă egocentrică (GoPro Hero 7 Black) ca intrare și poate prezice locația și traiectoria mișcării viitoare a persoanei vizate.

În [133], autorii fac o observație importantă: "este mult mai eficient să se învețe și să se prezică traiectoriile pietonilor în spațiul tridimensional, deoarece mișcarea umană are loc în lumea fizică tridimensională". Soluția lor a folosit o cameră stereo pentru a furniza informații 3D. Utilizând o rețea neuronală adversarială pentru estimarea poziției („Twin PoseGAN"), autorii au estimat poziția. Soluția lor poate fi considerată o extensie a „Social GAN"din domeniul 2D în domeniul 3D.

4.4. Comparație între cameră, LiDAR şi radar

În domeniul „automotive", cele mai noi tendințe combină informațiile provenite de la mai mulți senzori. De exemplu, Meyer și Kusch [134], au utilizat informații provenite de la senzorul radar Astyx 6455 HiRes, de la camera Point Grey Blackfly și de la modulul LiDAR Velodyne VLP-16 împreună cu CNN. Zhang și colaboratorii [135] au propus, de asemenea, un sistem de senzori montat pe un vehicul care include un LiDAR Velodyne HDL-32E, un sistem de navigație inerțial (OxTs Inertial + GNSS/INS suite) și o cameră Mako folosite pentru prezicerea nivelului de risc al coliziunii cu pietoni. O rețea LSTM a fost utilizată pentru a estima traiectoria pietonilor pe baza unor durate scurte (în medie 3,23 secunde). S-au utilizat 36 de pietoni pentru a colecta date ce reprezintă traiectoriile. Autorii au raportat o eroare medie de deplasare de 0,5074 m. Pentru clasificarea nivelului de risc, autorii au propus o combinație de K-Means Clustering (KMC), Kernel Principal Component

Analysis (KPCA) și o Kernel SVM. Pentru o imagine relativă la aplicațiile în care pot fi utilizați acești senzori auto, în raportul [136], a fost prezentată o retrospectivă a fiecărui senzor auto bazată pe aspecte de performanță (vezi Tabelul 4.4.1).

Tabel 4.4.1. Rezumat al performanței al fiecărui senzor auto (radar, LiDAR și cameră) prin evidențierea avantajelor și dezavantajelor acestora în diferite sarcini [136].

Performanța	**Radar**	**LiDAR**	**Camera**	**Fuziune (Radar+LiDAR +Camera)**	**Metrică**
Detecția Obiectelor	Înaltă	Înaltă	Mediu	Înaltă	Acuratețe
Clasificarea Obiectelor	Slabă	Mediu	Înaltă	Înaltă	Acuratețe
Estimarea Distanței	Înaltă	Înaltă	Mediu	Înaltă	Acuratețe
Detecția Marginilor	Slabă	Înaltă	Înaltă	Înaltă	Sensibilitate
Urmărirea Liniilor	Slabă	Slabă	Înaltă	Înaltă	Linearitate
Vizibilitate	Înaltă	Mediu	Mediu	Înaltă	Rezoluție
Performanță în condiții meteo rele	Înaltă	Mediu	Slabă	Înaltă	Acuratețe
Iluminare	Înaltă	Înaltă	Mediu	Înaltă	Sensibilitate

4.5. Seturi de date

Seturile de date mari sunt resurse importante pentru antrenarea și testarea modelelor DNN. Pentru asta, date din acestea sunt etichetate. Într-un scenariu posibil, pietonul poate fi instruit să efectueze acțiuni predefinite (oprire, traversare pe trecere de pietoni, traversare, etc.), dar nu este limitat la acestea. Din păcate, numărul de date colectate este limitat. Se pot identifica două provocări majore în seturile de date disponibile: deoarece sunt predefinite, acestea nu cuprind toate informațiile (pietonul înregistrat este un "actor", variabilitatea acțiunilor sale este inexistentă). O doua provocare este reprezentată de faptul că în viața reală, acțiunile pietonilor sunt variabile, ele fiind determinate de evenimente aleatorii (sosirea autobuzului, semaforul își schimbă culoarea, etc.). Scenariile din viața reală au fost utilizate pentru modelele de nivel scăzut, cum ar fi detecția și urmărirea. Din păcate, ele nu oferă datele necesare pentru modelele de nivel superior (de exemplu, interacțiunile sociale).

Pentru a testa sistemele de predicție a traiectoriei pietonale, cercetătorii utilizează în mod obișnuit mai multe seturi de date. Acestea furnizează imagini ale pietonilor din diferite scenarii (promenadă, treceri de pietoni, trotuare, etc.). În acestea oamenii se mișcă în diferite direcții. Majoritatea detectoarelor utilizează imagini color [137], deși abordările bazate pe viziune nu pot colecta și furniza același nivel de informații în condiții de lumină scăzută sau noapte.

Pentru a compara performanța predicției, există mai multe metrici utilizate: Eroarea de Deplasare Finală (în literatura de limbă engleză „Final Displacement Error" – FDE) și Eroarea Medie de Deplasare (în literatura de limbă engleză „Estimated Median Error" – EMD) sunt aplicate la atribuțiile standardizate de predicție. EMD poate fi definită ca fiind media erorii pătratice medii, care este calculată între locația prevăzută a traiectoriei și cea reală. Acest lucru este calculat în fiecare interval de timp pe o perioadă de 5 secunde. Se știe că o metodă care oferă valori EMD mici are o abatere redusă față de adevăr. Acest lucru o face convenabilă. Eroarea pătratică medie între predicția traiectoriei și cea adevărată în ultimul interval de timp reprezintă eroarea de deplasare finală. Metodele cu FDE mic sugerează predicții mai bune pe termen lung.

Cu toate că au fost utilizate diferite seturi de date în același mod, comparațiile de performanță generează ocazional dispute, rămânând greu de subliniat importanța unei bune performanțe pe un set de date sau o secvență specifică, în ceea ce privește un algoritm de predicție. Înregistrarea datelor poate fi realizată cu unul sau mai multe tipuri diferite de senzori: camere monoculare, camere stereo, radar, camere RGB-D, LiDAR sau o combinație între aceștia.

4.5.1. Imagini captate din traficul rutier

Disponibilitatea seturilor de date ample și precise este importantă pentru obținerea unor performanțe bune în ceea ce privesc metodele bazate pe date. În această secțiune, vor fi prezentate principalele seturi de date folosite pentru predicția traiectoriilor pietonilor în trafic.

Setul de date KITTI Vision Benchmark (KITTI) [39]: KITTI este unul dintre cele mai cunoscute seturi de date când vine vorba de cercetarea vehiculelor autonome. Problematica asupra cărora se concentrează acest set de date cuprinde: stereo, flux optic, odometrie vizuală, detectare obiecte 3D și urmărire 3D. În acest scop, s-a echipat un vehicul cu două camere video color și alb-negru de înaltă rezoluție. Informații precise de referință pentru obiectele percepute sunt furnizate utilizând un scaner Velodyne 3D-LiDAR și un sistem de localizare pe bază de GPS. Seturile de date sunt capturate în orașul Karlsruhe (Germania), în zone rurale dar și pe autostrăzi. Până la 15 vehicule și un număr de 30 de pietoni sunt vizibili în imagini. În total, KITTI cuprinde aproximativ 6 ore de date colectate în scenarii de trafic. El conține hărți de adâncime și segmentare semantică, precum și grupări de puncte LiDAR. Acestea permit compararea rezultatelor obținute prin diferite abordări utilizând aceleași intrări. Incluzând peste o sută de metode clasificate, reperul de detecție a pietonilor KITTI promovează o evoluție semnificativă în domeniul vehiculelor autonome. El furnizează o infrastructură care permite testarea și compararea diferitelor abordări pentru detecția și urmărirea pietonilor. Exemple pentru datele etichetate din acest set de date și vehiculul folosit pentru a le înregistra sunt prezentate în Figura 4.5.1.1.

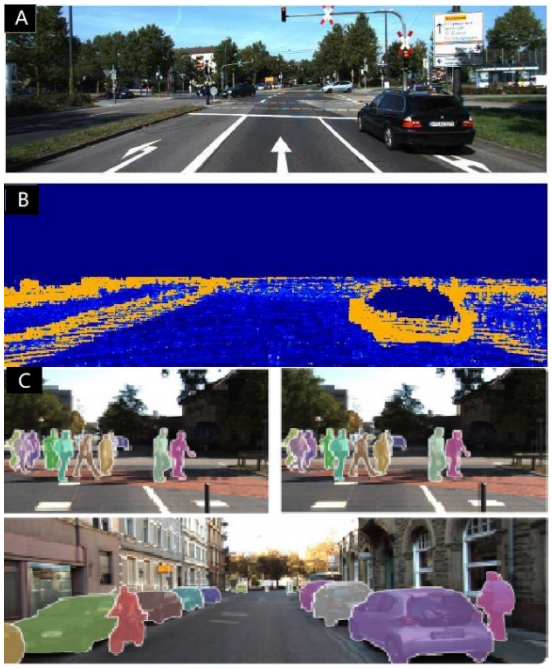

Figura 4.5.1.1. Setul de date Kitti: Vehiculul folosit pentru a înregistra setul de date și unele date adnotate ale camerei și 3D LiDAR de grupări de puncte [39].

Setul de date NuScenes [48] este primul set de date la scară largă care furnizează date din întregul ansamblu de senzori al unui vehicul autonom (adică 6 camere, 1 LiDAR, 5 radar, GPS, IMU). În 2019, setul complet de date cu 1.000 de scene a fost lansat. Acesta include aproximativ 1,4 milioane de imagini de cameră, 390.000 de scanări LiDAR, 1,4 milioane de scanări radar și 1,4 milioane de cutii încadrate pentru obiecte în 40.000 de cadre cheie. Trăsături suplimentare cu o hartă extinsă, mai multe date sunt eliberate secvențial. Acestea includ 1000 de scene de conducere în Boston și Singapore, două orașe cunoscute pentru traficul dens și situațiile de conducere extrem de provocatoare. Perioada fiecărei scene este de 20 de secunde. Acestea sunt selectate manual pentru a arăta un set divers și interesant de manevre de conducere, situații de trafic și comportamente neașteptate. Pentru a facilita sarcinile comune de viziune artificială, cum ar fi detectarea și urmărirea obiectelor, se notează 23 de clase de obiecte încadrate 3D la 2Hz pe întregul set de date. O extensie a setului de date, potrivită pentru sarcinile de predicție a traiectoriei, a fost publicată la sfârșitul anului 2020. Exemple pentru datele etichetate în acest set de date sunt prezentate în Figura 4.5.1.2.

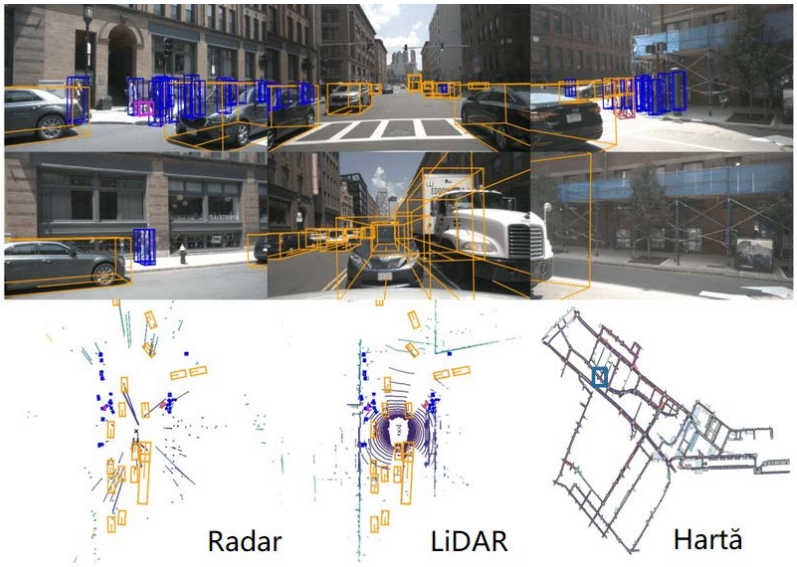

Figura 4.5.1.2. Set de date NuScenes: imagini adnotate ale camerei, RADAR, LiDAR și date de hărți din setul de date NuScenes [48].

Setul de date „ApolloScape Open" [138] este un set de date larg, care susține antrenarea și evaluarea algoritmilor și sistemelor de conducere autonomă bazate pe viziune. Include diferite sarcini, cum ar fi: reconstrucție 3D, auto-localizare, segmentare semantică, segmentare a instanțelor, prezicerea traiectoriei, detectarea și urmărirea obiectelor. Sistemul lor de achiziție constă din două scanere cu laser, până la șase camere video și un sistem combinat IMU/GNSS. Organizat în seturi separate sub numele de setul de date „Trajectory", setul de date „3D Perception LiDAR Object Detection"și setul de date „Tracking", include aproximativ 100.000 de cadre de imagine, 80.000 de grupări de puncte LiDAR și traiectorii de 1000 km în trafic urban. „ApolloScape" include condiții și densități de trafic variate, care includ multe scenarii dificile în care vehiculele, bicicliștii și pietonii se mișcă unul printre celălalt.

Setul de date Argoverse [139] este un alt set de date mare, colectat în orașele Miami și Pittsburgh din Statele Unite ale Americii. Înregistrările au fost făcute în diferite condiții meteorologice și în diferite momente ale zilei. Acestea conțin imagini „surround view" obținute cu ajutorul a 7 camere și un sistem de viziune stereo. Au fost utilizați doi senzori Velodyne LiDAR cu 32 de fascicule, plasați vertical, pentru a furniza grupări de puncte LiDAR cu 64 de fascicule. Imaginile frontale sincronizate cu ansamblele de puncte corespunzătoare din setul original de date Argoverse au fost extrase. S-a permis o toleranță de 51 de milisecunde între scanările LiDAR și imaginile corespunzătoare. Setul rezultat furnizează seturi pentru antrenare (aproximativ 13 k imagini), validare (peste 5 k imagini) și testare (mai mult de 4 k imagini de testare).

Setul de date Cityscapes [46] a fost dezvoltat de Daimler AG R&D, Institutul Max Planck pentru Informatică (MPI-IS) și Grupul de Inferențe Vizuală TU Darmstadt, din Germania. Setul de date Cityscapes se concentrează pe înțelegerea semantică a scenelor urbane, motiv pentru care conține imagini color stereo de vedere.

Diversitatea imaginilor este foarte mare: 50 de orașe, anotimpuri diferite (primăvară, vară, toamnă), condiții meteorologice diverse și dinamici diferite ale scenelor. Există 5000 de imagini cu etichetări detaliate și 20.000 de imagini cu adnotări grosiere ale informației semantice.

Setul de date „Caltech Pedestrian" [140] este unul dintre cele mai utilizate seturi de date pentru predicția traiectoriei pietonilor. Conține imagini color etichete sub forma de chenare de încadrare pentru pietoni. Setul de date „Caltech Pedestrian" conține 10 ore de material video cu rezoluție de 640 x 480, și 30 de cadre pe secundă, luate dintr-un vehicul de conducere din mediu urban. Aproximativ 250.000 de cadre au fost etichetate. Ele conțin segmente de aproximativ 137 de minute și un total de 350.000 de chenare de încadrare folosite pentru a marca aproximativ 2300 de pietoni.

Setul de date „Daimler" [141] furnizează informații privind detectarea pietonilor. Acesta conține informații referitoare la intenția lor. Acesta ia în considerare patru tipuri diferite de mișcare a pietonilor: începutul mersului, oprirea, traversarea și înclinarea. Fiecare secvență video conține etichetele menționate anterior referitoare la intențiile pietonilor.

Setul de date "Lyft Level 5" [142] conține peste 18.000 de imagini cu vedere frontală. Aproximativ 12.600 de imagini au fost utilizate în scopul antrenării modelelor DNN. În timp ce 3.000 dintre ele formează un subset de validare, restul de 3.000 de imagini au fost selectate pentru testare. Datele au fost înregistrate în jurul orașului Palo Alto, SUA, în timpul zilei și în condiții meteorologice bune. Acest set de date oferă grupări de puncte pentru fiecare imagine, iar doi senzori LiDAR pentru plafon cu 40 (sau 64) de fascicule și doi senzori LiDAR cu 40 de fascicule au fost utilizați. În plus, sunt disponibile cinci imagini de cameră care acoperă un unghi de 360°.

Setul de date Waymo [143] conține peste 122 de mii de imagini de antrenament, aproximativ 30 de mii de imagini de validare și aproximativ 40 de mii de imagini de testare. Acesta este împărțit în 12.000, 3000 și 3000 pentru antrenare, validare și respectiv testare. Setul de date conține imagini înregistrate în Phoenix, Mountain View și San Francisco în diferite momente ale zilei și în diverse condiții meteorologice. Setul de date Waymo conține, de asemenea, ansamblul de puncte LiDAR pentru fiecare imagine. Pentru înregistrarea acestor date s-au folosit cinci senzori LiDAR și mai multe camere.

Setul de date KAIST [144] folosește imagini provenite de la camere termice (majoritatea seturilor de date utilizează doar imagini color). Acest set de date le combină cu date colectate folosind camere obișnuite. Scopul este de a îmbunătăți antrenarea rețelelor neuronale.

Setul de date PIE [145] constă în peste 6 ore de înregistrări video capturate cu o cameră monoculară calibrată de tip dashboard Waylens Horizon, echipată cu o lentilă cu unghi larg de 157°. PIE conține videoclipuri HD (1920 x 1080 px) care rulează cu 30 cadre pe secundă. Peste 300.000 de cadre video etichetate, dezvăluind 1842 de exemple de pietoni, fac din PIE cel mai mare set de date public disponibil adecvat pentru comportamentul pietonilor în trafic.

Setul de date inD [146], care a fost obținut utilizând o dronă, conține peste 11.000 de traiectorii ale participanților la trafic, în general cei motorizați. Scenariile sunt bazate pe mobilitate urbană, incluzând scene de intersecții rutiere sau sensuri giratorii. O motivare similară referitoare la observarea spațiilor dintre mașini și alți utilizatori de drum a fost realizată de Ko-PER [147]. Folosind videoclipuri și scanări

laser pot fi furnizate traiectoriile vehiculelor și pietonilor într-o anumită intersecție rutieră.

Setul de date H3D [148] include peste 27.000 de imagini care reprezintă 160 de scene aglomerate, cu un total de 1,1 milioane de chenare 3D etichetate. Este utilizat un unghi de vizualizare de 360° pentru a marca obiectele (pentru comparație, în setul de date KITTI, doar obiectele din vizualizarea frontală sunt marcate).

Setul de date TRAF [130] ia în considerare categoriile de mașini, autobuze, camioane, pietoni, scutere, motociclete și animale. Setul de date conține 13 vehicule motorizate, 5 pietoni și 2 biciclete pe cadru. Etichetarea a fost efectuată urmând un protocol strict, iar fiecare fișier video rezultat conține coordonate spațiale în pixeli, un ID al agentului precum și tipul acestuia. Unghiul de vedere al camerei (față/sus) este luat în considerare pentru a categorisi imaginile din setul de date. Sunt luate în considerare momentele diferite ale zilei (zi, seară și noapte) și nivelurile diferite de dificultate.

4.5.2. Imagini captate din zonele urbane

Așa cum imaginile captate din traficul rutier ajută la dezvoltarea și cercetarea de noi metode pentru a rezolva problema prezicerii traiectoriei pietonilor, în ultimi ani au apărut în literatura de specialitate și baze de date captate în zonele urban pietonale. În acest subcapitol sunt prezentate cele mai utilizate astfel de baze de date pentru PTP.

Seturile de date ETH [34] și UCY [35] sunt cele mai utilizate seturi de date în această zonă, bazate pe videoclipuri de supraveghere a pietonilor care se deplasează (pe trotuar) și sunt etichetate cu coordonatele lor de locație. Setul de date UCY furnizează direcții ale privirii subiecților, utilizate pentru a captura unghiul de vizualizare al pietonului (vezi figura 4.5.2.1). Aceste două seturi de date includ cinci scene, trei dintre ele fiind de la UCY (numite Univ, Zara1 și Zara2) și două de la ETH (numite ETH și Hotel). În total, ele conțin peste 1600 de traiectorii ale pietonilor, locațiile acestora fiind etichetate la fiecare 0,4 secunde. Pentru a realiza antrenarea și testarea, se folosește metoda de validare încrucișată "leave-one-out", care presupune că modelul trebuie să fie antrenat pe patru scene diferite, dar trebuie să fie testat pe a cincea. Acest proces trebuie repetat o dată pentru fiecare scenă, ceea ce înseamnă de cinci ori. De acum înainte, aceste două seturi de date vor fi menționate ca setul de date ETH-UCY, luând în considerare faptul că sunt utilizate împreună.

Figura 4.5.2.1. Imagini urbane captate în orașul Zurich din diferite locații.

Un alt set de date demn de menționat este Stanford Aerial Pedestrian (SAP), cunoscut uneori și sub numele de Stanford Drone (SD) [40]. Imaginile (vezi figura 4.5.4) din acest set de date furnizează o vedere de sus a scenelor (parcare, intersecții și alei pietonale din campus), imaginile fiind înregistrate de drone. Etichetele setului de date (reprezentate cu roșu și roz în figura 4.5.2.2) includ și clase pentru obiecte. La fiecare 0,4 secunde, locația pietonului este etichetată pentru un singur cadru. Autorii au împărțit datele în seturi de antrenare și testare, acesta din urmă fiind disponibil doar pentru locația observată. Setul de date poate furniza traiectoriile a aproximativ 19.000 de pietoni în zona unui campus universitar cu interacțiuni între aceștia (pietoni, cicliști, mașini și autobuze).

Figura 4.5.2.2. Imagini urbane captate cu drona în campusul Universității Stanford California.

Setul de date VIRAT Video [149] este un set de date de supraveghere video la scară mare, proiectat pentru a evalua performanța algoritmilor de recunoaștere de evenimente folosind scene realiste. Include date din camere fixe și drone. După acest set de date, în 2018 a apărut ActEV/VIRAT [150], o versiune îmbunătățită a VIRAT, conținând mai multe etichetări și videoclipuri. Acesta implică peste 12 ore de înregistrări, pentru 12 scene cu 455 de videoclipuri la 30 cadre pe secundă. Majoritatea videoclipurilor sunt caracterizate de o rezoluție ridicată de 1920 x 1080.

Setul de date ATC [151] se bazează pe etichetări furnizate de 49 de senzori 3D pentru un interval de 92 de zile, în ceea ce privește traiectoriile pietonilor în zona unui centru comercial.

Setul de date Town-Centre [50] a fost dezvoltat cu scopul de a urmări vizual, (folosind filmări video) un centru aglomerat al orașului. Se referă la aproximativ 2000 de pietoni care se deplasează, definiți prin comportamente naturale. În special, setul de date PETS'2009 [152] conține 11 secvențe care sunt capturate de opt camere monoculare, incluzând date furnizate de pietoni și care au diferite niveluri de aglomerări.

4.6. Concluzii

În cadrul acestui capitol sunt examinate lucrările reprezentative din domeniu ("State-of-the-art", SOTA), cu precădere metodele de predicție care folosesc rețele neuronale profunde împreună cu cei mai utilizați senzori (camera, LiDAR și radar) pentru conducerea autonomă a vehiculelor. S-au prezentat comparativ performanțele fiecărui senzor auto (radar, LiDAR și cameră) prin evidențierea avantajelor și dezavantajelor acestora în diferite sarcini, de exemplu detecția și clasificarea obiectelor, estimarea distanței, detecția marcajelor rutiere și a limitelor drumului etc. Au fost identificate cele mai importante seturi de date disponibile public, folosite de către cercetători în implementarea soluțiilor de tip PTP.

În legătură cu tematica prezentului capitol este publicată lucrarea [30] în care este realizată și o comparație cu privire la acești senzori care sunt punctele tari și slabe a fiecăruia. Totodată sunt alese cele mai utilizate seturi de date în PTP și au sunt prezentate locațiile unde s-au realizat acestea (traficul rutier sau zonele urbane).

5. METODOLOGIE

5.1. Descrierea problemei

Obiectivul principal al prezicerii traiectoriei este de a estima pozițiile viitoare ale unui grup de *N* pietoni într-o scenă reală. Acest lucru se bazează pe pozițiile lor anterioare și curente pe o reprezentare de tip hartă pe parcursul unui interval de timp, începând de la momentul T_o (așa cum se poate observa în Figura 5.1.1) după T_p iterații în timp. Poziția actuală a fiecărui pieton într-o scenă este reprezentată de coordonata reală X = (x, y) la iterația *i* aceasta fiind notată cu $X_t^i = (x_t^i, y_t^i)$, cu $t \in \{1, \ldots, T_o\}$. Aici (x_t^i, y_t^i) sunt variabile aleatoare care descriu distribuția de probabilitate a poziției pietonului *n* la momentul *t* în cadrul scenei (*n* este dependent de scenă). Traiectoria viitoare reală/adevărată este notat $Traj_{obs} = \{Y_t^i = (x_t^i, y_t^i)\}$, unde $i \in \{1, \ldots, n\}$, $T_o + 1 \leq t \leq T_p$.

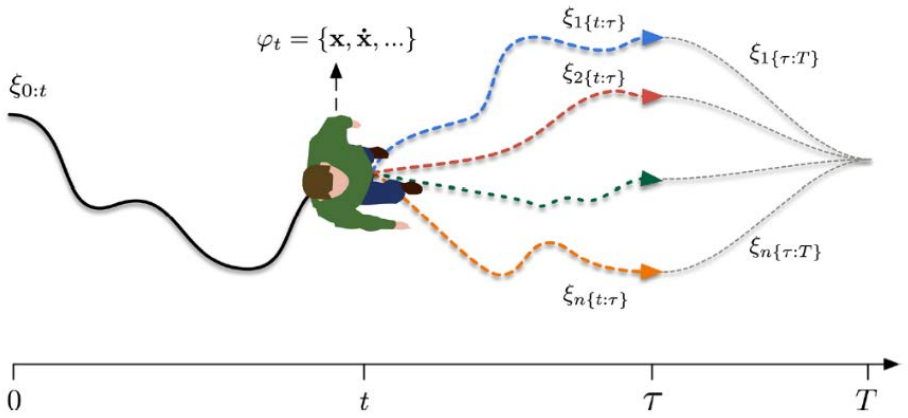

Figura 5.1.1. Distribuția temporală a poziției fiecărui pieton începând de la momentul T_o până la T_p. Aici se pot menționa trei tipuri de poziții: observate, reale-viitoare și prezise [160].

Pozițiile prezise, notate $Traj_{pred} = \{\hat{Y}_t^i = (\hat{x}_t^i, \hat{y}_t^i)\}$, $i \in \{1, \ldots, n\}$, $T_o + 1 \leq t \leq T_p$ reprezintă o serie de variabile aleatoare. Acestea sunt rezultatul unei presupuneri că poziția pietonului *i* la momentul *t* urmează o distribuție gaussiană bidimensională, notată $\hat{Y}_t^i \sim \mathcal{N}(\mu_t^i, \sigma_t^i, \rho_t^i)$. În cadrul acestei distribuții $\mu_t^i = (\mu_x, \mu_y)_t^i$ reprezintă centrul grupului de pietoni la momentul *t*. Deviația standard a distribuției este reprezentată de $\sigma_t^i = (\sigma_x, \sigma_y)_t^i$, iar coeficientul de corelație este notat ρ_t^i. Pentru a obține predicția traiectoriei, arhitectura propusă prezice parametrii distribuției gaussiene $(\mu_x, \mu_y, \sigma_x, \sigma_y, \rho)_t^i$.

Pentru a estima pozițiile la momentele $T_o + 1 \leq t \leq T_p$ pentru fiecare participant la trafic se folosesc locațiile observate la momentele de timp $1 \leq t \leq T_o$. Pentru estimarea traiectoriei se folosește o arhitectură de rețea neuronală. Pentru a antrena

parametrii modelului, se utilizează funcția de pierdere a logaritmului negativ al probabilității, așa cum este prezentată în Ecuația (5.1.1).

$$L^i(W) = -\sum_{t=1}^{T_p} log(f(x_t^i, y_t^i | \mu_x, \mu_y, \sigma_x, \sigma_y, \rho)) \tag{5.1.1}$$

În Ecuația (5.1.1), *W* reprezintă parametrii rețelei antrenate. Valoarea pierderii este minimizată pentru a obține performanțele optime ale rețelei.

5.2. Metode de evaluare și metrici

În cercetarea în domeniul vederii artificiale, traiectoriile sunt adesea descrise prin statistici de mișcare, cum ar fi numărul de coliziuni, accelerația medie, viteza medie și distanța totală parcursă [61]. Pentru fiecare participant la trafic, există opt etape de observare (de 3,2 secunde). Pentru toate seturile de date, se folosesc douăsprezece etape (4,8 secunde) pentru a reprezenta pozițiile reale viitoare. Pentru evaluarea diferenței poziției dintre traiectoria reală și cea estimată se folosesc normele euclidiene *L2*.

Cu o gamă atât de largă de abordări și rezultate, poate fi dificil de evaluat progresul în domeniu. Chiar formularea întrebării despre performanță poate introduce polarizări în favoarea anumitor metode. De exemplu, formularea următoare exclude abordările generative sau probabilistice: dată fiind o predicție a traiectoriei $\{\hat{p}_1, \ldots, \hat{p}_T\}$ și a poziției observate $\{p_1, \ldots, p_T\}$, cum se evaluează cât de "aproape" este estimarea de traiectoria reală? Vom începe cu această întrebare, chiar dacă exclude anumite clase de metode.

Figura 5.2.1 (a) ilustrează una dintre cele mai comune modalități de a compara direct traiectoriile alăturate, adică de a măsura cât de departe este pentru fiecare *t* și apoi de a calcula media acestor distanțe pentru a obține eroarea medie pe durata prognozei. Aceasta este cunoscută în mod obișnuit ca Eroarea Medie de Deplasare („Average Displacement Error" - ADE) și este raportată de obicei în unități de lungime, de exemplu metri:

$$ADE = \frac{\sum_{n=0}^{N} \sum_{t=0}^{T_p} \|\hat{p}_t^n - p_t^n\|_2}{N \times T_p} \tag{5.2.1}$$

De multe ori, este posibil ca interesul să fie reprezentat de eroarea punctului final al traiectoriei subiectului, ilustrată în Figura 5.2.1 (b) (în special, se compară doar punctele $\hat{p}_3$ și p_3. Aceasta furnizează o măsură a erorii metodei la sfârșitul orizontului de predicție și este denumită frecvent eroarea de deplasare finală („Final Displacement Error" - FDE). Si ea este de obicei raportată în unități de lungime:

$$FDE = \frac{\sum_{n=0}^{N} \|\hat{p}_t^n - p_t^n\|_2}{N}, t = T_p, \tag{5.2.2}$$

unde *N* reprezintă numărul de pietoni, T_p numărul pașilor de timp prezis, iar p_t^n și $\hat{p}_t^n$ sunt rezultatul real și rezultatul prezis la pasul de timp *t*.

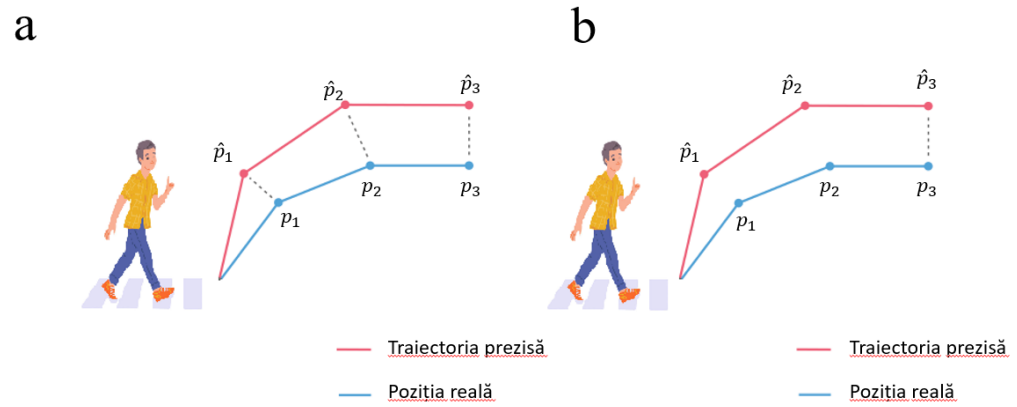

Figura 5.2.1. Ilustrații ale metricilor. (a) Eroare de deplasare medie (ADE), (b) Eroare deplasare finală (FDE).

ADE și FDE sunt cele două metrici principale utilizate pentru evaluarea regresorilor deterministi. În timp ce aceste metrici sunt naturale pentru obiectivul propus, ușor de implementat și interpretabile, în general nu reușesc să captureze nuanțele metodelor mai sofisticate (vezi mai multe detalii mai jos).

După cum s-a menționat mai devreme, sistemele critice pentru siguranță trebuie să ia în considerare multe posibile rezultate viitoare, ideal cu probabilitățile fiecăruia de a se produce. Astfel, luarea deciziilor se face în condiții de siguranță, considerând o gamă întreagă de poziții posibile. În acest context, ADE și FDE sunt insuficiente deoarece se concentrează pe evaluarea unei singure traiectorii deterministe estimate. Aceasta ridică următoarea întrebare: Cum se evaluează abordările generative care produc simultan mai multe estimări sau chiar distribuții complete asupra pozițiilor viitoare?

Dată fiind traiectoria viitoare observată $\{p_1,\ldots,p_T\}$ și o metodă care produce o distribuție analitică pentru fiecare pas de timp viitor, cum se evaluează cât de "bune" sunt distribuțiile în raport cu adevărul? O abordare comună este de a evalua probabilitatea adevărului sub distribuțiile estimate, reprezentată în Figura 5.2.2 (a). În special, se calculează probabilitatea pozițiilor viitoare adevărate sub distribuțiile estimate asociate, se face o medie și se neagă rezultatul (deoarece metricile de eroare sunt în general minimizate), obținându-se „Negative Log-Likelihood” (NLL),

$$NLL\big(\hat{p}, w(y|x)\big) = -\frac{1}{T}\sum_{t=1}^{T} log\, w(p_t|x) \tag{5.2.3}$$

În timp ce NLL ține cont de forma completă a distribuției prezise, nu doar de medie și varianță, acesta poate evalua doar metodele de prezicere a traiectoriilor care produc distribuții de ieșire analitice (acele funcții de probabilitate ale căror calcul este

eficient). Ca rezultat, este necesară o altă metrică pentru metodele care pot produce doar distribuții empirice, cum ar fi abordările bazate pe GAN.

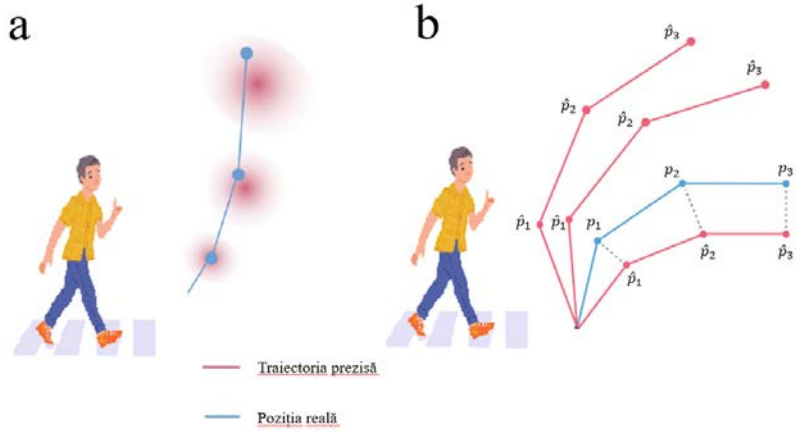

Figura 5.2.2. Ilustrații ale metricilor (a) Negative Log-Likelihood (NLL) și (b) Best-of-N (BoN) ADE.

Pentru metodele din care rezultă distribuții empirice, evaluarea se face prin colectare de traiectorii estimate $\{\hat{p}_1,\ldots,\hat{p}_T\}$. O idee inițială pentru evaluarea performanței acestor metode este de a genera *N* predicții din model și de a calcula performanța celei mai bune. Această abordare este denumită de obicei Best-of-N (BoN), împreună cu metrica de performanță asociată. De exemplu, în Figura 5.2.2 (b) este prezentată metrica Best of *N* ADE, unde *N* = 3 și se măsoară ADE al celei mai bune prognoze, adică prognoza cu cel mai mic ADE. Aceasta este principala metrică folosită de metodele generative care produc distribuții empirice, cum ar fi abordările bazate pe GAN. Ideea din spatele acestei scheme de evaluare este de a identifica dacă traiectoria reală se apropie de prognozele generate de câteva eșantioane din model (*N* este de obicei ales să fie mic, de exemplu, 20). Implicit, această metrică de evaluare selectează un eșantion ca fiind cea mai bună predicție și apoi o evaluează cu ajutorul metricilor ADE/FDE menționate anterior. Cu toate acestea, această abordare nu este potrivită pentru conducerea autonomă deoarece necesită cunoașterea viitorului și nu este clar cum se corelează performanța BoN cu lumea reală.

O metrică posibil a fi utilizată este „Kernel Density Estimate" (KDE). Acesta este un instrument statistic apropiat de o funcție de densitate de probabilitate obținută pe baza unui set de eșantioane. Ilustrată în Figura 5.2.3, începe prin adunarea mai multor traiectorii ($\sim 10^3$, pentru a obține un set reprezentativ de rezultate ale metodelor comparate). Apoi, se ajustează o estimare de densitate a nucleului [153], [154] la fiecare pas al prognozei pentru a obține o funcție de densitate de probabilitate a pozițiilor eșantionate la fiecare pas. Aceasta se utilizează pentru calculul mediei log-probabilității traiectoriei reale. Această metrică denumită „Negative Log-Likelihood" este bazată pe KDE (KDE-NLL) și este raportată în unități logaritmice.

KDE-NLL nu prezintă aceleași dezavantaje ca BoN, deoarece: (1) metodele cu rezultate foarte diferite vor genera KDE-uri foarte diferite; (2) nu necesită estimări în timpul evaluării. În plus, estimează corect NLL al unei metode fără a face presupuneri cu privire la structura distribuției rezultatelor metodei.

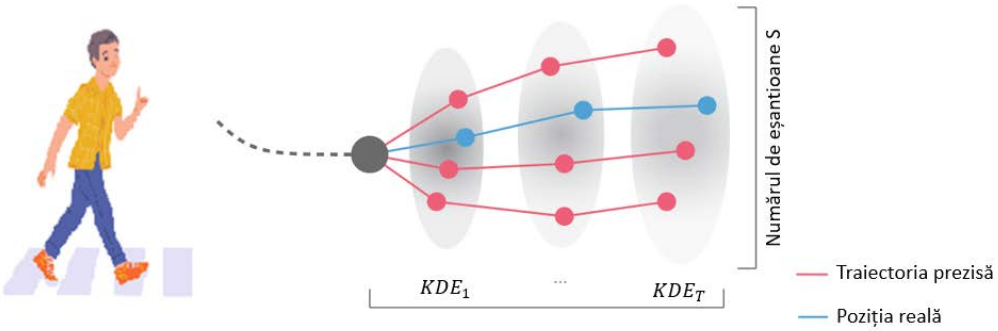

Figura 5.2.3. Metrica Kernel Density Estimate bazată Negative Log-Likelihood (KDE-NLL) utilizează KDE-uri la fiecare pas de timp pentru a calcula probabilitatea traiectoriei, realizând o medie în timp pentru a obține o valoare.

5.3. Predicția bazată pe filtrul Kalman

Filtrul Kalman (KF) este un filtru bun pentru reducerea zgomotului și netezirea datelor într-un proces cu zgomot aleatoriu liniar, spre exemplu mișcările unui avion în timpul zborului sau aterizării. Cu toate acestea, deplasarea unui pieton tinde să fie un proces aleatoriu neliniar, de aceea se poate utiliza un filtru Kalman extins, deoarece acesta poate filtra zgomotele dintr-un proces neliniar. KF extins are doi parametri de stare, *x* și *p*; *x* este definit ca stare și include informații despre poziție și viteză; *p* este definit ca matricea de covarianță care va fi actualizată cu starea *x* anterioară [155].

Filtrul Kalman este utilizat pentru a estima stările recursive într-un sistem liniar și va folosi fiecare măsurătoare și starea precedentă prezisă pentru a obține o nouă estimare a stării.

Filtrul Kalman are două faze principale: faza de predicție și faza de actualizare. În prima dintre ele, filtrul va estima starea curentă utilizând starea anterioară. Ecuațiile de predicție sunt prezentate mai jos [156]:

$$x_{k|prediction} = ax_{k-1|update} \tag{5.3.1}$$

$$p_{k|prediction} = ap_{k-1|update}a \tag{5.3.2}$$

În ecuațiile (5.3.1) și (5.3.2), variabila *x* reprezintă starea estimată, iar *k* reprezintă numărul de stare. $x_{k|prediction}$ este a *k*-a stare estimată din prima fază, iar $x_{k-1|update}$ reprezintă a $(k-1)$-a stare actualizată din faza de actualizare. Varibila *p* reprezintă covarianța stării *x*. $p_{k|prediction}$ este estimarea p a stării *k* din prima fază și $p_{k-1|update}$ este p_k a stării prezise. Valoarea $p_k-1|update$ reprezintă covarianța actualizată al (k − 1) a stării din faza de actualizare. Variabila a este o constantă. Pentru faza de actualizare, filtrul va actualiza starea curentă utilizând noua măsurătoare reală. Ecuațiile de actualizare sunt prezentate mai jos:

$$g_k = \frac{p_{k|prediction}}{p_{k|prediction} + r} \tag{5.3.3}$$

$$x_{k|update} = x_{k|prediction} + g_k(z_k - x_{k|prediction}) \tag{5.3.4}$$

$$p_{k|update} = (1 - g_k)p_{k|prediction} \tag{5.3.5}$$

Variabila g_k este câștigul curent, unde *r* reprezintă zgomotul mediu al datelor de intrare, iar z_k reprezintă măsurătoarea observată.

5.4. Predicția bazată pe filtrul alfa-beta-gama

În cea mai mare parte a timpului, detectarea oricărei stări particulare a unui sistem implică efectuarea măsurătorilor și compararea acestora cu una sau mai multe valori limită. Dacă poziția pietonului este parametrul de interes, valoarea măsurată periodic va fi comparată cu valoarea limită. Această abordare este utilizată aproape peste tot pentru declanșarea anumitor situații [157].

O metodă ușor de înțeles, aplicată în domeniul vederii artificiale pentru determinarea pragului optim în prelucrarea imaginilor începe cu o valoare particulară. De obicei, pragul este setat inițial la valoarea medie din imagine. Pixelii imaginii sunt împărțiți în două seturi pe baza comparației cu această valoare. Se calculează valorile medii pentru ambele seturi. Valoarea pragului pentru următoarea iterație este media acestor valori. Imaginea este apoi împărțită din nou folosind rezultatul. Acești pași se repetă până când diferența dintre cele două valori de prag (limită) este suficient de mică.

Un exemplu de caz real în care se observă şi măsurătorile și pragul se poate vedea în Figura 5.4.1. În acest caz, abscisa reprezintă timpul. Ordinata poate reprezenta orice valoare de interes (inclusiv procent). Valorile măsurate depășesc pragul la momentul *t1*. După depășirea pragului, sistemul efectuează semnalizarea. Uneori se utilizează și un factor de încredere. Într-un caz posibil, încrederea ține cont de numărul de valori peste prag (cu cât este mai mare acest număr, cu atât este mai mare încrederea).

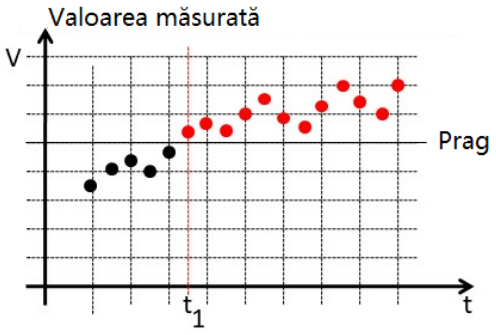

Figura 5.4.1. Valoarea măsurată care depășește pragul: Valorile 'V' sub și peste prag.

Măsurătorile sunt în mod obișnuit afectate de erori (semnalele ale căror valori sunt măsurate, sunt afectate de zgomot). Modalitățile de minimizare a erorilor de măsurare sunt descrise în literatură. Dacă pragul este aproape de valorile măsurate, zgomotul existent poate determina declanșarea sistemului în starea pragului superior. Aceasta poate fi observată în Figura 5.4.2. În astfel de cazuri pot apărea oscilații și sistemul poate deveni instabil. Comportamentul oscilant este nedorit deoarece poate cauza deteriorarea sistemului.

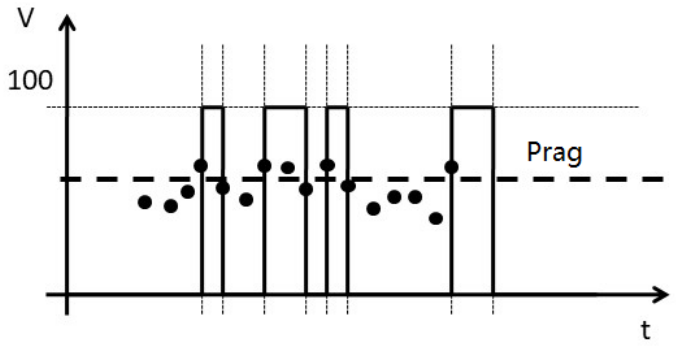

Figura 5.4.2. Valoarea măsurată care depășește pragul: Detectări false în prezența zgomotului.

Se poate utiliza o histereză (fenomenul în care valoarea unei proprietăți fizice rămâne în urmă modificărilor efectului care o provoacă) pentru a evita problema prezentată în Figura 5.4.2. În acest scop, sunt definite două praguri. Creșterea valorii măsurate va depăși în timp pragul superior, determinând sistemul să semnaleze o stare nouă. Scăderea valorilor va ajunge la pragul inferior și în cele din urmă va coborî sub acesta. În momentul în care se întâmplă acest lucru, sistemul va semnala acest fapt. Această situație poate fi observată în Figura 5.4.3. Utilizarea histerezei face sistemul mai puțin sensibil la zgomot. Se poate utiliza o valoare și o compensație pentru a calcula cele două praguri.

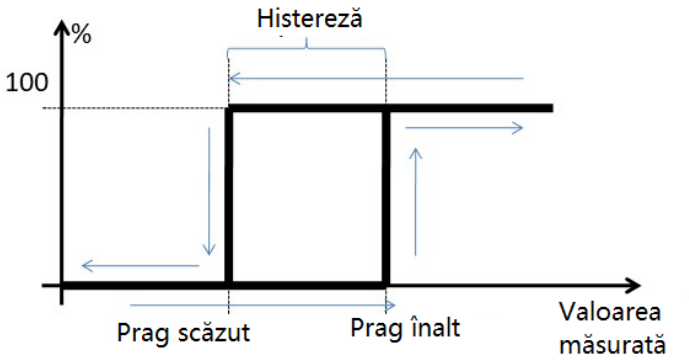

Figura 5.4.3. Utilizarea histerezei pentru eliminarea detectărilor false.

Sistemul poate rămâne într-o astfel de situație pe termen nelimitat (această apariție poate fi cauzată de mediul înconjurător. De exemplu, în domeniul auto, imaginea furnizată de camera de bord este afectată de condiții meteorologice precum ploaie, zăpadă, condens, etc.). Cazurile ambelor praguri pot fi observate în Figura 5.4.4 (valorile se află imediat deasupra pragului superior) și Figura 5.4.5 (valorile se află imediat sub pragul superior). Schimbarea valorilor pragurilor pentru a atenua această problemă nu garantează că o astfel de situație nu se va mai întâmpla. Este important de menționat că există cazuri (de exemplu, în domeniul vederii artificiale) în care nu se poate decide dacă sistemul funcționează corect prin vizualizarea imaginilor/înregistrărilor.

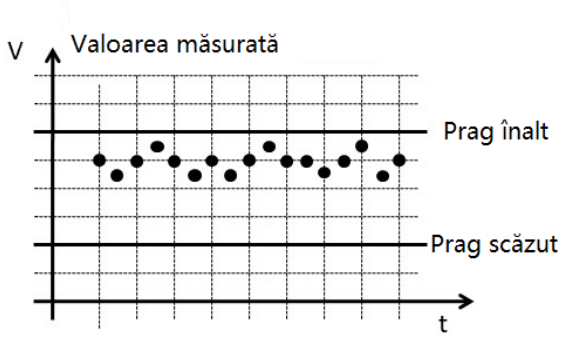

Figura 5.4.4. Limitele pragului, problema "sub pragul superior".

Una dintre modalităţile de corecţie a acestui fel de situaţii constă în utilizarea unui filtru de urmărire. Un posibil algoritm candidat este filtrul Alfa-Beta-Gama. Acesta este un membru al familei de filtre Alfa-Beta. Fitrul urmăreşte bine, erorile apar mai ales la maximele şi minimele locale.

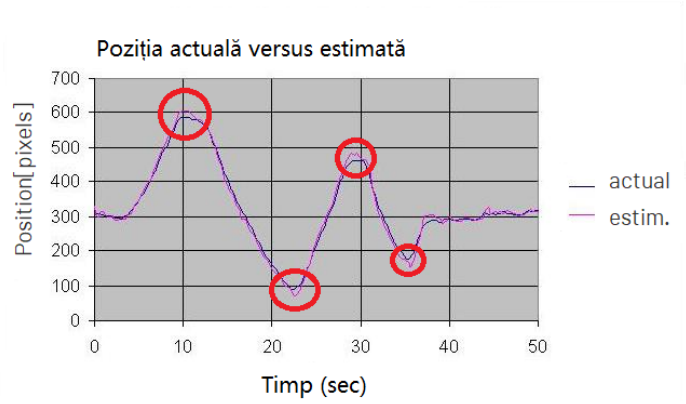

Figura 5.4.5. Predicția poziției (albastru - actual, roșu - predicție filtrată) și erori de schimbare a semnului primei derivată.

Acesta este chiar cazul necesar a fi corectat (figura 5.4.5). Acest lucru sugerează faptul că un astfel de filtru ar putea fi folosit la rezolvarea unei astfel de probleme.

Filtrul estimează prima derivată a valorii și iterația următoare a valorii în pasul de predicție. Când valorile estimate devin cunoscute, estimările vor fi netezite (ajustate) în al doilea pas [158].

Ecuațiile (23) și (24) reprezintă de fapt pasul de estimare. În Ecuația (5.4.1), valoarea prezisă pentru iterarea *k+1* este denumită $x_p(k+1)$, $x_s(k)$ reprezintă valoarea corectată pentru iterarea *k*, în timp ce valoarea corectată a primei derivate pentru iterarea *k* este denumită $v_s(k)$. Timpul de eșantionare este reprezentat de litera *T*.

$$x_p(k+1) = x_s(k) + T \cdot v_s(k) + \frac{T^2}{2} \cdot a_s(k) \tag{5.4.1}$$

$$v_p(k+1) = v_s(k) + T \cdot a_s(k) \tag{5.4.2}$$

În ecuațiile (5.4.1) și (5.4.2), $a_s(k)$ reprezintă valoarea rotunjită a celei de-a doua derivate pentru iterația *k* (restul notărilor rămânând aceleași).

Corecția (rotunjită) este dată de ecuațiile (5.4.3), (5.4.4) și (5.4.5) și are loc într-un mod similar cu cel al filtrului anterior. Atât variabila cât și prima sa derivată sunt corectate. Estimările valorii și primei sale derivate sunt corectate în al doilea pas (reprezentat de ecuaţiile 5.4.3, 5.4.4 şi 5.4.5).

$$x_s(k) = x_p(k) + \alpha \cdot \left(x_o(k) - x_p(k)\right) \tag{5.4.3}$$

$$v_s(k) = v_p(k) + \frac{\beta}{T} \cdot \left(x_o(k) - x_p(k)\right) \tag{5.4.4}$$

$$a_s(k) = a_s(k-1) + \left(\frac{\gamma}{2 \cdot T^2}\right) \cdot \left(x_o(k) - x_p(k)\right) \tag{5.4.5}$$

5.4.1. Stabilitatea filtrului alfa-beta-gama

Filtrul Alfa-Beta-Gama nu este un algoritm stabil în orice condiţii. Pentru a studia stabilitatea, trebuie să obținem funcția de transfer a filtrului. Se va aplica așa-numita transformare *Z* pentru a determina funcția de transfer. Astfel, ecuațiile de predicție vor deveni după cum este arătat în ecuațiile (5.4.1.1) și (5.4.1.2).

$$zX_p(z) - zx_p(0) = X_s(z) + TV_s(z) + \frac{1}{2}T^2A_s(z) \tag{5.4.1.1}$$

$$zV_p(z) - zv_p(0) = V_s(z) + TA_s(z) \tag{5.4.1.2}$$

Corecția este exprimată matematic în ecuațiile (5.4.1.3), (5.4.1.4) și (5.4.1.5). În Ecuația (32), termenul *1/z* consideră că $a_s(k-1)$ reprezintă valoarea

anterioară a accelerației. În cele din urmă, funcția de transfer a filtrului este dată de Ecuația (5.4.1.6).

$$X_s(z) = X_p(z) + \alpha \cdot \left[X_o(s) - X_p(z)\right] \tag{5.4.1.3}$$

$$V_s(z) = V_p(z) + \frac{\beta}{T} \cdot \left[X_o(s) - X_p(z)\right] \tag{5.4.1.4}$$

$$A_s(z) = \frac{1}{z} \cdot A_s(z) + \left(\frac{\gamma}{2 \cdot T^2}\right) \cdot \left[X_o(s) - X_p(z)\right] \tag{5.4.1.5}$$

După cum s-a menționat anterior, valoarea prezisă (denumită 'poziția' țintei) poate fi orice măsură de interes (de exemplu, distanța) cu valorile senzorului folosite pentru a calcula estimările. De asemenea, se poate deduce o altă funcție de transfer care corelează 'viteza' prezisă cu poziția observată pentru filtru.

$$G_{P\alpha\beta\gamma}(z) = \frac{X_p(z)}{X_o(z)} = \frac{\left(\alpha+\beta+\frac{\gamma}{4}\right)\cdot z^2+\left(-2\cdot\alpha-\beta+\frac{\gamma}{4}\right)\cdot z+\alpha}{z^3+\left(\alpha+\beta+\frac{\gamma}{4}-3\right)\cdot z^2+\left(-2\cdot\alpha-\beta+\frac{\gamma}{4}+3\right)\cdot z+\alpha-1} \tag{5.4.1.6}$$

Este necesar un criteriu pentru a examina stabilitatea filtrului. Criteriul lui Jury este utilizat pentru a studia stabilitatea filtrului. Studiul este prezentat doar pentru *α - β - γ* (deoarece va fi efectuat în același mod și pentru celelalte).

Criteriul de stabilitate Routh-Hurwitz nu poate fi implementat direct în planul *z* deoarece granița sa diferă de cea a planului *s* [159]. În schimb, testul de stabilitate Jury este o procedură similară pentru sistemele discrete. Să luăm în considerare următoarea ecuație (5.4.1.7).

$$G(z) = a_n \cdot z^n + a_{n-1} \cdot z^{n-1} + \cdots + a_1 \cdot z + a_0 \tag{5.4.1.7}$$

Testul de stabilitate Jury este format conform prezentării din Tabelul 5.4.1.1.

Tabel 5.4.1.1. Criteriul de stabilitate Jury pentru un sistem cu o funcție de transfer discretă și o ecuație caracteristică (34).

z^0	z^1	…	z^{n-1}	z^n
a_0	a_1	$\cdots$	a_{n-1}	a_n
a_n	a_{n-1}	$\cdots$	a_1	a_0
b_0	b_1	$\cdots$	b_{n-1}	
b_{n-1}	b_{n-2}	$\cdots$	b_1	
c_0	c_1	$\cdots$		
c_{n-2}	c_{n-3}	$\cdots$		
…	…	…	…	…
a_0	m_1	m_2		

Pentru un sistem de ordinul doi, tabelul criteriului de stabilitate Jury conține un singur rând. Odată cu creșterea ordinului sistemului, se adaugă două linii suplimentare în tabel (liniile cu numere pare au aceleași elemente ca și cele anterioare, dar aranjate în ordine inversă). Liniile cu numere impare au elementele definite așa cum se poate observa în ecuațiile (5.4.1.8) și (5.4.1.9).

$$b_k = \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix} \tag{5.4.1.8}$$

$$c_k = \begin{vmatrix} b_0 & b_{n-1-k} \\ b_{n-1} & b_k \end{vmatrix} \tag{5.4.1.9}$$

Polinomul *G(z)* definit de (5.4.1.10) și (5.4.1.11) nu are rădăcini în afara cercului unitate (sau pe granița sa) dacă următoarele *n-1* constrângeri sunt satisfăcute. Condițiile exprimate de (5.4.1.12) și (5.4.1.13) sunt uneori numite "condiții necesare de stabilitate". Condițiile exprimate în (5.4.1.12), (5.4.1.13) și (5.4.1.14) sunt uneori denumite "condiții suficiente de stabilitate".

Coeficienții polinomului caracteristic pentru poziția filtrului pot fi văzuți în Tabelul 5.4.1.2.

$$G(1) > 0 \tag{5.4.1.10}$$

$$(-1)^n \cdot G(-1) > 0 \tag{5.4.1.11}$$

$$|a_0| < |a_n| \tag{5.4.1.12}$$

$$|b_{n-1}| < |b_0| \tag{5.4.1.13}$$

$$|c_{n-2}| < |c_0| \cdots |m_2| < |m_0| \tag{5.4.1.14}$$

Tabel 5.4.1.2. Criteriul de stabilitate Jury pentru filtrul α – β – γ.

$\mathbf{z^0}$	$\mathbf{z^1}$	$\mathbf{z^2}$	$\mathbf{z^3}$
$\alpha - 1$	$-2 \cdot \alpha - \beta + \frac{\gamma}{4} + 3$	$\alpha + \beta + \frac{\gamma}{4} - 3$	1
1	$\alpha + \beta + \frac{\gamma}{4} - 3$	$-2 \cdot \alpha - \beta + \frac{\gamma}{4} + 3$	$\alpha - 1$
$\alpha(\alpha - 2)$	$\alpha\left(4 - 2 \cdot \alpha - \beta + \frac{\gamma}{4}\right) - \frac{1}{2}$	$\alpha\left(\alpha + \beta - 2 + \frac{\gamma}{4}\right) - \frac{1}{2}$	

Prima condiție de stabilitate necesară este:

$$1 + \alpha + \beta + \frac{\gamma}{4} - 3 - 2 \cdot \alpha - \beta + \frac{\gamma}{4} + 3 + \alpha - 1 = \frac{\gamma}{2} > 0 \tag{5.4.1.15}$$

Expresia (5.4.1.15) restricționează parametrul γ la valori pozitive. A doua condiție necesară de stabilitate este reprezentată de Ecuația (5.4.1.16).

$$2 \cdot \alpha + \beta < 4 \tag{5.4.1.16}$$

Există două condiții suficiente pentru stabilitatea filtrului. Deoarece parametrul γ al filtrului este 1, Ecuația (5.4.1.12) se traduce în (5.4.1.17).

$$|\alpha - 1| < 1 \tag{5.4.1.17}$$

Ecuația (5.4.1.17) va duce la (5.4.1.18).

$$0 < \alpha < 2 \tag{5.4.1.18}$$

A 2-a condiție este exprimată de Ecuația (5.4.1.19).

$$|\alpha \cdot (\alpha - 2)| > \left|\alpha \cdot (\alpha - 2) + \alpha \cdot \left(\beta + \frac{\gamma}{4}\right) - \frac{\gamma}{2}\right| \tag{5.4.1.19}$$

Deoarece parametrul α este întotdeauna pozitiv și (α - 2) este, de asemenea, negativ (conform ecuației (5.4.1.18)), expresia (5.4.1.19) se transformă conform (5.4.1.20).

$$\alpha \cdot \left(\beta + \frac{\gamma}{2}\right) - \frac{\gamma}{2} > 0 \tag{5.4.1.20}$$

Expresia (5.4.1.20) conduce la o constrângere pentru parametrul γ, prezentată în (5.4.1.21).

$$\gamma < \frac{4 \cdot \alpha \cdot \beta}{2 - \alpha} \tag{5.4.1.21}$$

5.5. Predicția bazată pe rețelele neuronale de tip graf

În cadrul acestui subcapitol este furnizată o descriere a aspectelor cheie ale metodei de estimare propuse. Ulterior, va fi discutată în detaliu concepția fiecărui modul, inclusiv crearea generală a modelului. Scopul principal este de a prezice traiectoriile viitoare pentru numeroși factori care interacționează folosind istoricul pozițiilor și informațiile de context. Metoda de predicție este, de asemenea, extensibilă la cadrele de urmărire multi-țintă [160].

Metoda constă în două componente: rețeaua neurală de convoluție spațio-temporală pe graf (ST-GCNN) și rețeaua neurală de convoluție cu extrapolare temporală (TXP-CNN). Prima componentă utilizează convoluția pentru a extrage caracteristici. Aceste caracteristici oferă o descriere concisă a istoricului traiectoriilor observate. Acestea sunt introduse în TXP-CNN, care le folosește pentru a prezice pozițiile tuturor pietonilor din grup și pentru a extrapola traiectoriile viitoare. Diagrama de ansamblu a acestei metode este ilustrată în Figura 5.5.1.

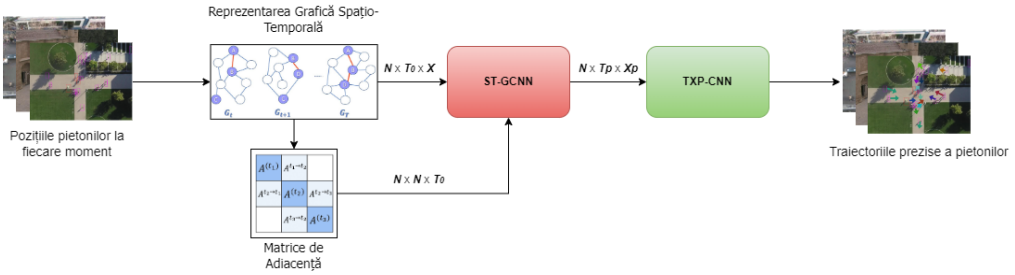

Figura 5.5.1. Arhitectura generală a metodei propuse. Prin optimizarea dimensiunii stratului, este posibilă obținerea unei precizii sporite în predicția traiectoriei [160].

Graful este format din pozițiile relative ale traiectoriilor pietonilor și capturează interacțiunile spațiale dintre pietoni. Luând în considerare *N* pietoni la momentul t, traiectoriile lor sunt reprezentate sub forma unui graf $G_t = (V_t, E_t)$. Aici, $V_t = \{\{v_t^n | \forall_i \in \{1,\dots,N\}\}$ este un set de noduri care corespund pozițiilor pietonilor. Fiecare locație observată (x_t^n, y_t^n) este un punct al v_t^n. $\{e_t^{nj} | \forall n, j \in \{1,\dots,N\}\}$ reprezintă setul de muchii în cadrul grafului G_t. Vectorul e_t^{nj} evidențiază muchia între v_t^n și v_t^j (*n* și *j* denotă pietoni) și este reflectată printr-o matrice de adiacență A_t^m de dimensiune *N x N* pentru fiecare moment. A_t^m este definită în Ecuația (5.5.1).

$$A_t^m(n,j) = \begin{cases} 1/\left\|v_t^n - v_t^j\right\|_2 & if\ \left\|v_t^n - v_t^j\right\|_2 \neq 0, \\ 0 & Otherwise \end{cases} \tag{5.5.1}$$

Prima componentă a arhitecturii este ST-GCNN. Este folosită pentru a încorpora reprezentarea obiectului [72]. Ea are rolul de a extrage înglobarea spațio-temporală din graficul de intrare. Următorul pas constă în normalizarea matricei A_s care reprezintă topologia rețelei la pasul de timp t, pentru generarea matricei Laplacian, după cum se poate observa în Ecuația (5.5.2):

$$f(V(\iota), A) = \sigma\left(\Lambda_t^{-\frac{1}{2}} \hat{A}_t^s \Lambda_t^{-\frac{1}{2}} V(\iota) W(\iota)\right). \tag{5.5.2}$$

În Ecuaţia (5.5.2), legăturile nodului sunt luate în considerare prin adăugarea matricei identitate I la matricea de adiacență A_t^s, iar Λ_t este matricea diagonală a gradului cu componente care reflectă suma pe rânduri a matricei $A_t^s + I$.

ST-GCNN este specificat în continuare prin convoluția hărții de viteză $V(\iota)$ cu kernelul $W(\iota)$ la stratul ι sub Laplacianul grafic $\hat{A}_t^s$, după cum se poate observa în Ecuația (5.5.3).

$$f(V(\iota), A) = \sigma\left(\Lambda_t^{-\frac{1}{2}} \hat{A}_t^s \Lambda_t^{-\frac{1}{2}} V(\iota) W(\iota)\right). \tag{5.5.3}$$

A doua componentă utilizată este TXP-CNN. Aceasta lucrează direct pe dimensiunea temporală a înglobării grafice $\hat{V}$ care crește pentru a satisface cerințele

de precizie în predicție. Cea de a doua componentă are mai puțini parametri decât unitățile recurente, deoarece se bazează pe operații de convoluție în spațiul caracteristicilor. Este important de menționat că stratul TXP-CNN nu este invariant la permutare, deoarece variațiile în înglobarea grafică conduc la rezultate diferite. Cu toate acestea, predicțiile sunt invariante dacă ordinea pietonilor este alterată [161].

Obiectivul principal al extrapolatorului temporal al rețelei este de a efectua convoluții temporale în istoricul traiectoriei pentru a detecta locațiile viitoare. Acest lucru se datorează faptului că rețeaua de convoluție temporală (TCN) [162] este considerată un sistem mai puternic și mai eficient pentru învățarea dependențelor temporale decât arhitecturile recurente. Fiecare strat temporal este interconectat în mod rezidual cu cel anterior.

Întregul proces de predicție a traiectoriei pietonilor este ilustrat în Algoritmul 1.

Algoritmul 1 Rețele Convoluționale cu Graf Dinamic

Date de intrare: coordonatele pietonilor $X = (x, y)$;

Date de ieșire: metricele de evaluare, eroare medie de deplasare (ADE) și eroarea deplasării finale (FDE);

for $t \in [1, T_o]$ **do**
    reprezintă traiectoriile drept un graf: $G_t = (V_t, E_t)$;
    calculează traiectoriile viitoare; $Traj_{obs} = Y_t^i = \left(x_t^i, y_t^i\right)$
end for
creează distanță de distribuție N x N matrice de adiacență A_t^m utilizând ecuația (5.5.1);
generează matricea laplaciană utilizând ecuația (5.5.2);
for fiecare $i \in 1, \ldots, n$ **do**
    for all $t \in [1, T_o]$ **do**
        distribuția probabilității pentru traiectoria prezisă: $\hat{Y}_t^i = (\hat{x}_t^i, \hat{y}_t^i) \sim \mathcal{N}(\mu_t^i, \sigma_t^i, \rho_t^i)$;
    end for
end for
colectează toate locațiile prezise si cele reale pentru fiecare pieton;
calculează ADE și FDE cu formula de la ecuația (5.2.1) și (5.2.2);
return ADE și FDE

Scopul principal al prezicerii traiectoriei pietonilor este de a estima pozițiile viitoare ale unui grup de N pietoni în funcție de istoricul lor de mișcare. Dată o condiție de trafic, x_s^t reprezintă coordonata spațială a pietonului s la pasul de timp t. Pentru a obține coordonatele N pietonilor, de la pasul de timp 1 la T_o, este posibil să se obțină traiectoriile observate, denumite $\boldsymbol{W} = \{x_s\}_{s=1}^N$, unde $x_s = \{x_s^t,\}_{t=1}^{T_o}$. Datorită multi-modalității traiectoriei viitoare, există Z traiectorii viitoare, denumite $\boldsymbol{P} = \left\{\boldsymbol{\rho}_j\right\}_{j=1}^Z$. Poziția adevărată este $\widehat{\boldsymbol{\rho}} = \{y_s\}_{s=1}^N$, unde $y_s = \{x_s^t\}_{t=T_{o+1}}^{T_p}, \widehat{\boldsymbol{\rho}} \in \boldsymbol{P}$. Într-un scenariu real de trafic, traiectoria este modificată nu doar de intenția pietonilor, ci și de interacțiunea

dintre pietoni la fiecare pas de timp t, reprezentată de $\boldsymbol{Z} = \{Z_t\}_{t=1}^{T_o}$. Pe scurt, metoda este împărțită în două părți. Pe baza traiectoriei observate $\boldsymbol{W}$ și a interacțiunii $\boldsymbol{Z}$, modelul prezice mai întâi toate traiectoriile viitoare social acceptabile $\boldsymbol{P}$. Fiecare traiectorie calculată are o probabilitate asociată. La final, traiectoriile cu cea mai mare probabilitate sunt folosite pentru a genera traiectoria finală [163]. Metoda este descrisă matematic prin Ecuația (5.5.4).

$$c(\hat{\boldsymbol{\rho}}|\boldsymbol{W},\boldsymbol{Z}) = \sum_{\rho \in P} c(\boldsymbol{\rho}|\boldsymbol{W},\boldsymbol{Z}) c(\hat{\boldsymbol{\rho}}|\boldsymbol{\rho},\boldsymbol{W},\boldsymbol{Z}), \tag{5.5.4}$$

În Ecuaţia (5.5.4), variabila c (.|.) este o distribuție condiționată discretă deoarece $\boldsymbol{P}$ este reprezentat printr-un arbore.

Pe de altă parte, soluția încorporează $\boldsymbol{P}$ într-un spațiu structurat, care este definit în mod particular de un arbore ternar (*ternary tree*). Deoarece arborele ternar nu este afectat de modalitatea medie a datelor, fiecare traiectorie conținută în arbore poate menține propriul său comportament de deplasare, oferind o bună interpretabilitate și evitând modalitatea frecventă. În plus, în comparație cu eșantionarea repetată, procedura de selecție ar putea produce rezultate stabile și previzibile. Toate datele neprelucrate sunt obținute din scenarii reale și sunt utilizate ca date de intrare pentru model [164].

Pasul principal al soluției constă în construirea unui arbore de traiectorie, care este legat de generarea unui arbore ternar, așa cum s-a discutat anterior. La fiecare pas de timp, întregul proces poate fi văzut ca o împărțire recursivă în trei direcții. Datorită dependenței temporale a traiectoriei, vectorul de viteză al traiectoriei observate este utilizat pentru a obține direcția împărțirii înainte (mers înainte). În mod special, direcțiile împărțirii în stânga (viraj la stânga) și dreapta (viraj la dreapta) sunt obținute prin rotații pozitive și negative ale vectorului de viteză cu un unghi specific, respectiv.

Figura 5.5.2 prezintă arhitectura generală a metodei optimizate. Mai exact, mai întâi este creat arborele de traiectorie pentru a furniza spațiul discret structurat $\boldsymbol{P}_{coarse}$. Similar, interacțiunea spațială și traiectoria observată sunt codificate una după alta pentru a obține codificarea interacțiunii și codificarea observată.

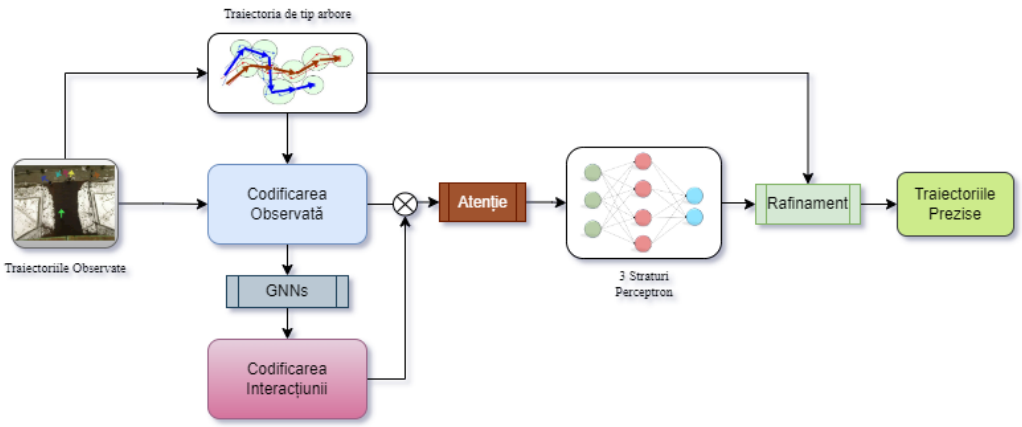

Figura 5.5.2. Arhitectura generală a metodei propuse [163].

GCN este folosit pentru a extrage codificarea observată și codificarea interacțiunii din traiectoria observată [169]. Apoi, arborele și interacțiunea sunt combinate printr-un mecanism de atenție pentru a evalua fiecare cale în arborele de traiectorii grosiere, iar vectorul de încredere obținut este îmbunătățit prin supervizare cu ajutorul etichetei, care este obținută prin comparația între fiecare cale și traiectoria grosieră adevărată. După aceasta, $\boldsymbol{P}_{coarse}$ este rapid optimizată prin calea cu cea mai mare probabilitate pentru a obține traiectoria grosieră estimată, supervizată de traiectoria grosieră adevărată. În final, se obţin traiectoriile estimate grosieră (pentru rafinare) şi detaliată.

După cum este arătat în Figura 5.5.2, având locația traiectoriilor observate x_i (linie cu săgeți verzi) pentru pietonul *i*, se pot obține vitezele corespunzătoare notate $v_i = \{v_i^t\}_{t=1}^{T_o}$, prin deplasarea de la un pas de timp la următorul pas de timp, presupunând că pietonul stă pe loc la primul pas de timp, adică $\{v_i^1\}_{i=1}^{N} = 0$. Deoarece complexitatea arborelui crește exponential cu creșterea adâncimii *d*, de exemplu, presupunând lungimea prezisă $T = T_P - T_O = 12$, se va genera un arbore ternar cu *d* = 12. Acesta va avea 312 căi dacă se face împărțirea la fiecare pas de timp.

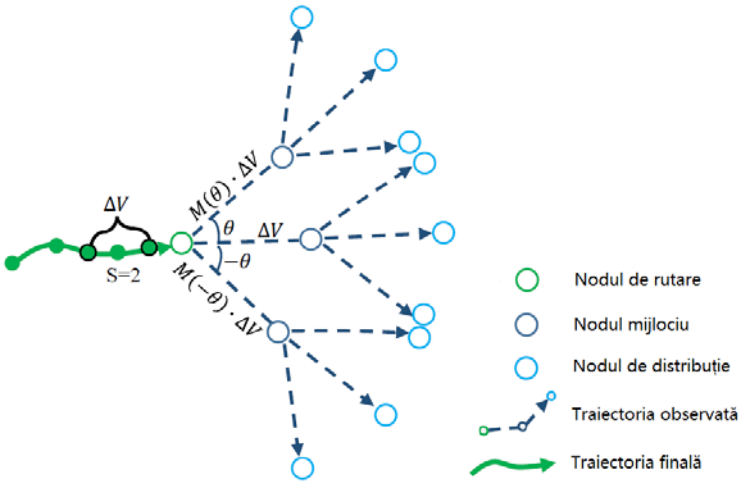

Figura 5.5.3. Un exemplu de generare a arborelui de traiectorii.

Pentru a echilibra complexitatea și acoperirea arborelui, acesta se împarte în mai multe etape temporale în loc să se facă împărțiri la fiecare pas de timp. Se stabilește un interval temporal specific $(1 \le Z \le T)$, iar viteza de ordin înalt (ΔV), obținută prin suma tuturor vectorilor de viteză în ultimele interacţiuni *Z* din traiectoria observată, este considerată direcția de împărțire în sensul de înaintare. Unghiul (θ) este utilizat pentru a genera direcțiile împărțirii în stânga și dreapta. În cele din urmă, se va genera un arbore ternar cu traiectorii grosiere, cu adâncimea $d = \lfloor T / Z \rfloor$ după un proces recursiv. Pentru acesta, fiecare cale de la rădăcină la frunză reprezintă o posibilă traiectorie viitoare grosieră. Totalitatea acestora constituie spațiul discret structurat grosier $\boldsymbol{P}_{coarse}$.

În Figura 5.5.3 se poate vedea exemplu de generare a arborelui de traiectorii grosiere cu adâncimea d = 2 și intervalul Z = 2. Viteza de ordin înalt ΔV, adică

deplasarea într-un interval temporal Z, este considerată direcția de împărțire înainte. Direcția de împărțire în stânga este obținută prin înmulțirea între matricea de rotație M cu unghiul (θ) și ΔV, în timp ce direcția de împărțire în dreapta este obținută prin înmulțirea între matricea de rotație M cu unghiul $-(\theta)$ și ΔV. Arborele de traiectorii grosiere este generat recursiv la fiecare împărțire, unde fiecare cale de la rădăcină la frunză reprezintă o posibilă traiectorie viitoare grosieră.

Traiectoria viitoare nu este afectată doar de informațiile interne de mișcare, ci și de interacţiunile cu alți pietoni. În [163] atenţia principală cade pe evaluarea eficacității arborelui pentru prezicerea traiectoriei pietonilor. Pentru a codifica traiectoria observată este folosit un multilayer perceptron (MLP). Ultimul pas al metodei este de a rafina traiectoria estimată grosieră pentru a obține o una detaliată.

O altă abordare utilizată în lucrarea [161] a fost implementarea unei arhitecturi de tip codificator-decodor așa cum se poate vedea în Figura 5.5.4. Numit "Spatio-Temporal Graph Convolutional Network" (ST-GCNN), este utilizată pentru modelarea interacțiunilor temporale ale traiectoriilor pietonale observate în timp. Aici, o rețea de grafice temporale este utilizată pentru a modela interacțiunile temporale ale unui singur pieton în pași de timp diverşi. Graficul temporal al pietonului N este creat și reprezintă pozițiile relative ale acestuia la diferite eșantioane de timp.

Decodorul conține un alt modul, și anume "Time-Extrapolator Convolutional Neural Network" (TXP-CNN), care este folosit pentru a prezice pașii următori. TXP-CNN lucrează direct la dimensiunea temporală a încorporării graficului și o mărește ca o necesitate pentru predicție. TXP-CNN are mai puțini parametri decât unitățile recurente, deoarece se bazează pe operații de convoluție asupra entităților. O caracteristică specială ce privește stratul TXP-CNN este că nu este un invariant la permutare, deoarece modificările în încorporarea graficului chiar înainte de TXP-CNN conduc la rezultate diferite.

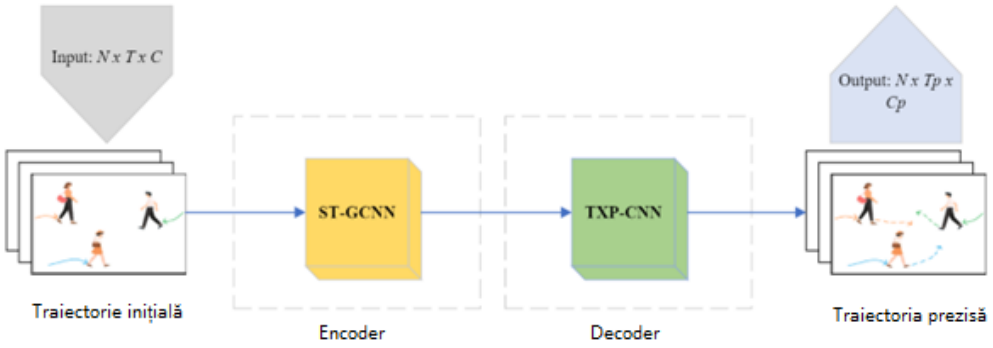

Figura 5.5.4. Arhitectura de tip codificator-decodor [161].

5.6. Concluzii

În cadrul acestui capitol au fost descrise metodele folosite de-a lungul cercetării pentru predicția traiectoriei pietonilor. În legătură cu tematica prezentului capitol am publicat lucrările [157], [158] pentru predicția bazată pe filtrul alfa-beta-gama. S-a efectuat o analogie între filtrul Kalman și filtrul alfa-beta-gama cu scopul de a identifica acurateței metodei. Pentru predicția bazată pe rețele neuronale de tip graf am publicat lucrările [160], [161]. Aici au fost prezentate metodele și arhitectura soluțiilor de predicție folosind rețeaua neurală de convoluție spațio-temporală pe graf (ST-GCNN) și rețeaua neurală de convoluție cu extrapolare temporală (TXP-CNN). Pentru evaluarea metodelor și comparația acestora cu alte metode din literatură, au fost prezentate cele mai reprezentative metrici din domeniu.

6. REZULTATE EXPERIMENTALE

În acest capitol, sunt prezentate experimentele și evaluările efectuate în ceea ce privește predicția traiectoriei pietonilor. Așa cum a fost menționat în Capitolul 5, experimentele au fost realizate folosind metode de predicție a traiectoriei pietonilor bazate în principal pe folosirea rețelelor neuronale profunde. Evaluăm performanța metodelor noastre în comparație cu metode similare de predicție a traiectoriei bazate pe rețele neuronale profunde.

Predicția traiectoriei joacă un rol în estimarea potențialului de risc al pietonilor, deoarece poate fi utilizată și pentru prezicerea traversării, ceea ce este necesar pentru prevenirea coliziunilor. Pentru a evalua aplicabilitatea metodelor de predicție a traiectoriei, evaluăm performanța metodelor pe care le-am propus (vezi Secțiunea 5.2), în mod cantitativ și calitativ, utilizând seturile de date ETH [34], UCY [35] și Stanford Drone [40], deoarece acestea sunt larg utilizate în literatură, disponibile public, și utilizează coordonate din lumea reală. Utilizarea metodelor bazate pe date necesită disponibilitatea unor date de calitate în cantitate suficientă. Mai exact, în predicția traiectoriei pietonilor, datele disponibile pot fi în două formate diferite: în coordonate de imagine sau în coordonate din lumea reală. Coordonatele de imagine înseamnă că fiecare pieton este reprezentat cu pixelii pe care îi ocupă în imaginea camerei, în timp ce coordonatele din lumea reală înseamnă că fiecare pieton este reprezentat prin poziția sa în metri, cu originea într-un punct arbitrar al lumii. Evaluările calitative sunt efectuate prin vizualizarea traiectoriilor generate.

Se va analiza calitativ modul în care metodele pe care le-am propus capturează interacțiunile sociale între pietoni și le iau în considerare atunci când prezic distribuțiile. Prezentăm cazuri în care D-STGCN [160], PTPCNN [161] și TreeGNN [163] prezic cu succes traiectorii fără coliziuni între pietoni care vin din unghiuri diferite, mențin mersul paralel și prezic corect rezultatul situațiilor în care o persoană se întâlnește cu un grup de pietoni.

6.1. Implementare

Pentru implementarea soluțiilor propuse, am utilizat framework-ul de învățare PyTorch. Pentru antrenare și evaluare, am folosit un GPU Nvidia GeForce GTX 1080 cu 8GB de memorie RAM pe un desktop care rulează Ubuntu versiunea 20.04 (vezi Figura 6.1.1). Modelele a fost antrenat timp de 250 de epoci, cu o dimensiune a lotului setată la 128. Algoritmul „Stochastic Gradient Descent" (SGD) a fost utilizat ca optimizator pentru rețeaua neurală. Rata de învățare inițială a fost setată la 0,01, în timp ce descreșterea a fost setată la 0,002 după 150 de epoci, iar PRELU [165] a fost utilizat ca funcție de activare. Toți hiperparametri au fost determinați empiric, urmând o abordare de încercare și eroare. Fiecare experiment are aceleași setări vizavi de hiperparametri atât în etapele de antrenare, cât și în cele de testare. Datele de antrenament au fost selectate în mod aleatoriu. Pentru evaluare, toate modelele au utilizat 20 de eșantioane (K): 8 cadre ca intrare și au prezis următoarele 12 cadre.

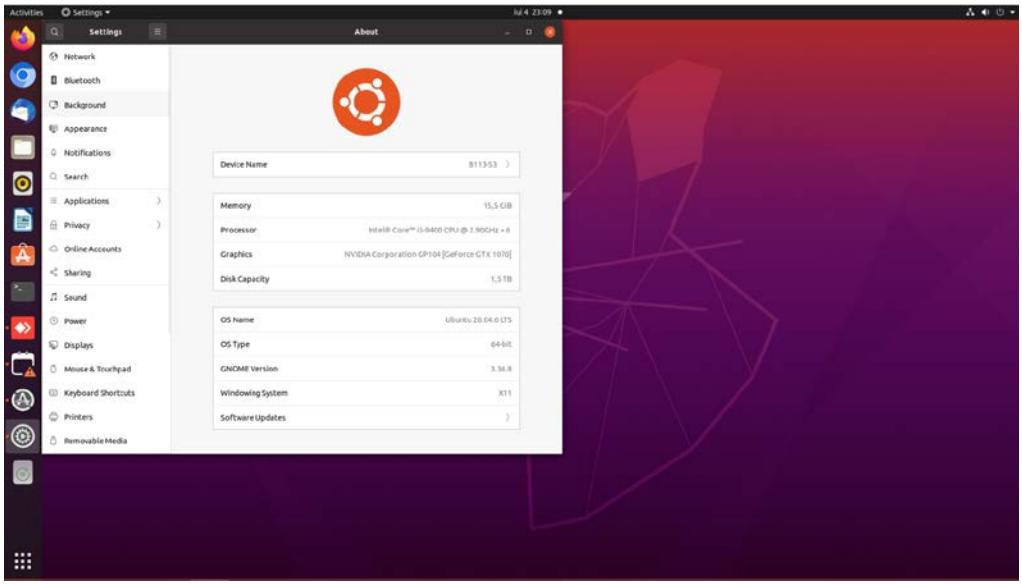

Figura 6.1.1. Ansamblu software și hardware folosit pentru antrenarea modelelor neuronale de predicție a traiectoriei.

6.2. Evaluarea predicției traiectoriei pietonilor

D-STGCN [160] este creat prin îmbinarea mai multor straturi de ST-GCNN și TXP-CNN. În general, selecția dimensiunii rețelei poate avea un impact major asupra rezultatelor ideale. Pentru a identifica arhitectura cea mai eficientă, au fost testate mai multe combinații de straturi ST-GCNN și TXP-CNN pe baza metodei de proiectare a experimentului (DOE), similar cu [166].

Fiecare experiment a fost efectuat cu aceleași setări de hiperparametri în fazele de testare și antrenare. Având în vedere complexitatea modelului, numărul maxim de straturi ST-GCNN și TXP-CNN este stabilit la 4. Metricile descrise în Subsecțiunea 5.2 au fost utilizate în scopuri de evaluare. Rezultatele empirice sunt detaliate în Tabel 6.2.1.

Tabel 6.2.1. Rezultate ale diferitelor combinații de parametri aplicate pe seturile de date ETH, UCY și SDD. Rezultatele sunt prezentate în termeni de metrici medii ADE/FDE. Coloana AWG reprezintă rezultatele medii ADE/FDE pentru toate scenele seturilor de date ETH-UCY. Cele mai bune rezultate sunt indicate în bold. Rezultatele numerice mai mici sunt mai bune.

ST-GCNN	TXP-CNN	SDD	ETH	HOTEL	UNIV	ZARA1	ZARA2	AWG
1	1	17.03/ 28.32	0.68/ 1.32	0.52/ 0.86	0.47/ 0.83	0.41/ 0.61	0.34/ 0.54	0.48/ 0.83
1	2	20.47/ 36.21	0.72/ 1.23	0.54/ 1.01	0.47/ 0.83	0.38/ 0.64	0.32/ 0.50	0.48/ 0.84
1	3	18.98/ 29.46	**0.63**/ **1.03**	0.40/ 0.65	0.50/ 0.89	0.37/ 0.60	0.32/ 0.50	**0.44**/ **0.73**
1	4	19.69/ 34.28	0.74/ 1.26	**0.37**/ **0.58**	0.47/ 0.85	**0.35**/ 0.57	**0.29**/ **0.48**	0.44/ 0.74
2	1	15.82/ **25.50**	0.69/ 1.33	0.58/ 0.91	**0.46**/ **0.78**	0.38/ **0.56**	0.34/ 0.51	0.49/ 0.81
2	2	18.65/ 31.05	0.99/ 1.80	0.52/ 0.93	0.52/ 0.96	0.37/ 0.58	0.37/ 0.57	0.55/ 0.96
2	3	24.32/ 44.00	0.77/ 1.50	0.43/ 0.72	0.50/ 0.92	0.36/ 0.59	0.36/ 0.56	0.48/ 0.85
2	4	17.56/ 31.53	0.81/ 1.64	0.67/ 1.23	0.50/ 0.90	0.38/ 0.62	0.34/ 0.55	0.54/ 0.98
3	1	25.09/ 29.93	0.69/ 1.34	0.53/ 0.91	0.62/ 1.10	0.42/ 0.70	0.39/ 0.58	0.53/ 0.92
3	2	**15.18**/ 25.93	0.70/ 1.26	0.58/ 0.97	0.57/ 1.03	0.44/ 0.67	0.35/ 0.53	0.52/ 0.89
3	3	43.46/ 62.67	0.77/ 1.34	0.66/ 1.22	0.49/ 0.88	0.40/ 0.60	0.39/ 0.56	0.54/ 0.92
3	4	23.86/ 42.04	0.73/ 1.42	0.48/ 0.78	0.51/ 0.96	0.44/ 0.70	0.32/ 0.53	0.49/ 0.82
4	1	18.65/ 31.84	0.99/ 1.74	0.53/ 0.74	0.57/ 0.95	0.50/ 0.86	0.35/ 0.53	0.58/ 0.96
4	2	21.54/ 31.21	0.94/ 1.94	0.64/ 1.03	0.53/ 0.84	0.51/ 0.89	0.35/ 0.51	0.59/ 1.04
4	3	19.18/ 33.29	0.75/ 1.29	1.13/ 2.04	0.53/ 0.98	0.39/ 0.63	0.39/ 0.55	0.63/ 1.09
4	4	26.88/ 43.56	0.71/ 1.25	0.89/ 1.56	0.56/ 0.99	0.45/ 0.69	0.35/ 0.54	0.59/ 1.00

Această studiu a concluzionat că cea mai bună arhitectură de model constă din:

- Un strat ST-GCNN și trei straturi TXP-CNN pentru seturile de date ETH-UCY.
- Trei straturi ST-GCNN și două straturi TXP-CNN pentru setul de date SDD.

Pe măsură ce numărul de straturi D-STGCN crește, performanța modelului scade (adică combinația de 3 și 4 straturi). Acest comportament este cauzat de lipsa datelor de vizualizare a scenei pentru intrările modelului.

Predicția metodei propuse este analizată calitativ pentru cinci scene reale diferite din seturile de date ETH și UCY. Rezultatele vizualizărilor sunt ilustrate în secțiunea următoare. Graficele prezentate aici arată că pietonii acordă atenție împrejurimilor lor. Oamenii acordă mai multă atenție oamenilor din apropiere și obstacolelor din fața lor decât celor din spatele lor. Schimbările de context în spatele unui pieton sau în fața lui au un impact redus asupra deciziilor de mișcare viitoare.

Pozițiile observate ale pietonului (traiectorie cunoscută cu opt puncte) sunt reprezentate de segmentele albastre între două locații consecutive. Traiectoria reală (observată) este reprezentată în verde (douăsprezece poziții). Predicția, pornind de la poziția nouă (pentru următoarele douăsprezece puncte), este reprezentată în roșu. Toate locațiile sunt ilustrate într-un grafic 2D (coordonate x, y) în metri.

Figura 6.2.1 reprezintă o scenă de traversare a drumului. Aceasta include interacțiuni ale pietonilor și un scenariu de evitare a obstacolelor în scena ETH. Această figură arată comportamentul algoritmului de predicție a traseului în absența informațiilor din interacțiunile sociale.

Figura 6.2.2 prezintă evaluarea calitativă a predicției modelului pentru interacțiunea pietonilor în scenariul HOTEL. De exemplu, soluții precum S-GAN-P [59] și Sophie [63] utilizează informații sociale simple pentru colectarea datelor; prin urmare, rezultatele predicției deviază semnificativ de la traiectoria reală. Modelul propus capturează informații sociale pe termen lung folosind graficul spațial-temporal; astfel, rezultatele sunt mai susceptibile să corespundă traiectoriilor reale ale viitorului.

Figura 6.2.3 corespunde scenei UNIV, în care subiectul merge singur pe campus fără interferențe. Modelul propus adună informații despre interacțiunea spațio-temporală între subiect și mediul înconjurător utilizând graficul spațial-temporal și furnizează diverse greutăți de impact pe baza acestor informații. Ca rezultat, traiectoria prezisă este destul de apropiată de traiectoria reală a viitorului.

Figura 6.2.4 reprezintă scena ZARA1 în care predicția traiectoriei eșuează. Pietonul analizat merge în mod normal în direcția dreaptă, dar este deviat temporar în direcția de alți pietoni care trec în apropiere. Rezultatul așteptat al modelului pentru această situație nu este satisfăcător, ținând cont de eroarea punctului final (FDE).

În final, în Figura 6.2.5 se poate vedea cel mai bun rezultat obținut cu modelul nostru în scena ZARA2. Aici avem un scenariu de traversare în care pietonii sunt considerați că merg împreună cu alți doi pe parcursul întregii scene. Figura 6.2.5 arată că, luând în considerare interacțiunea spațială, se obțin traiectorii mai adecvate social.

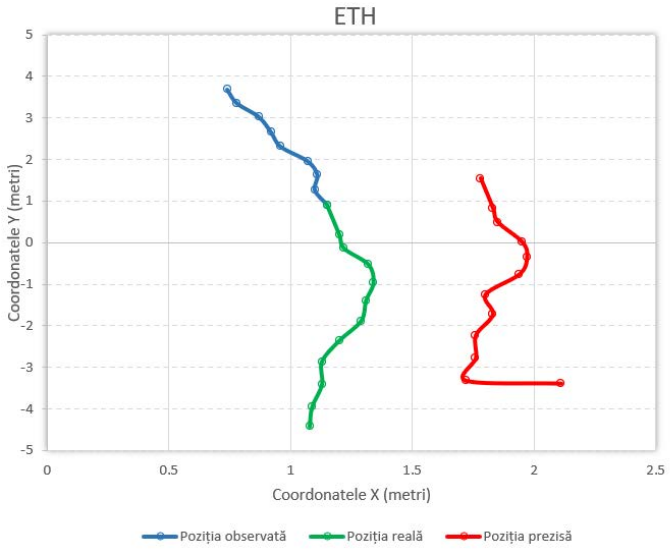

Figura 6.2.1. Rezultate scena ETH [160]

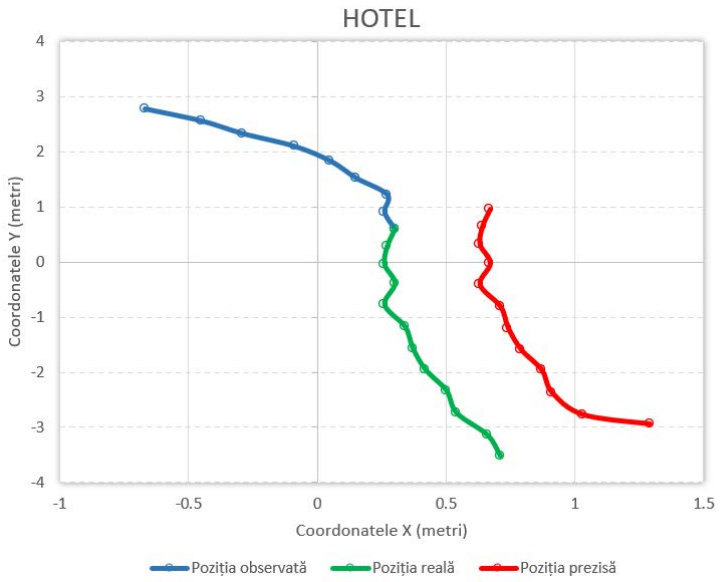

Figura 6.2.2. Rezultate scena HOTEL [160].

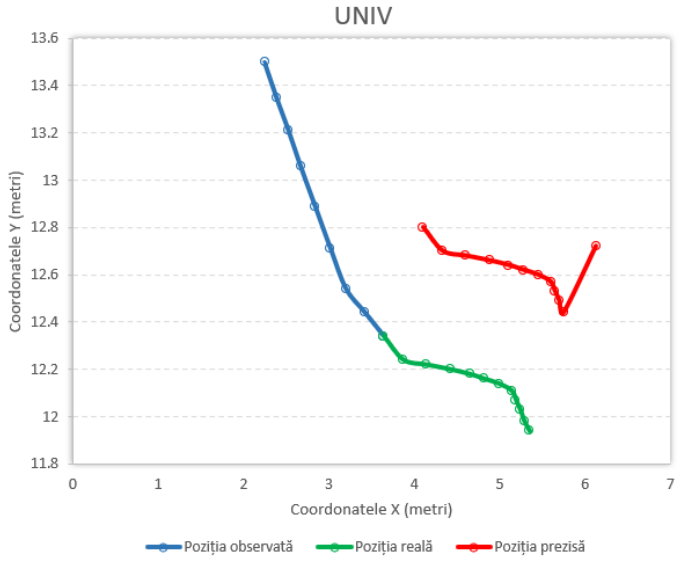

Figura 6.2.3. Rezultate scena UNIV [160].

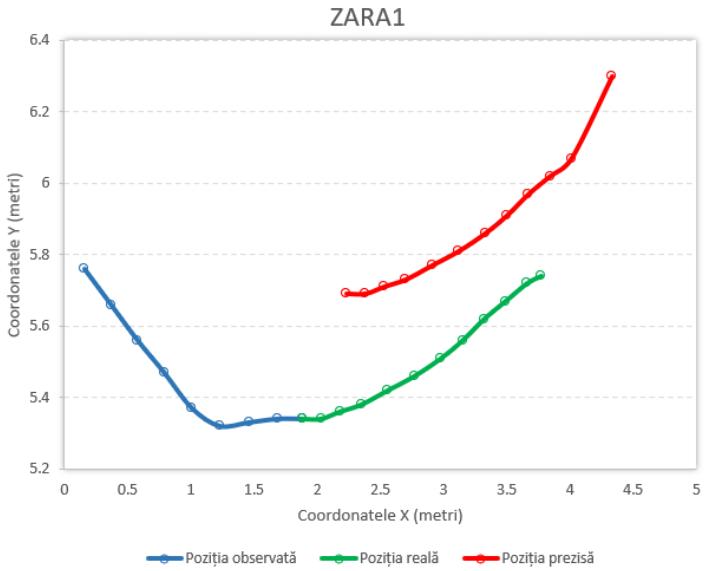

Figura 6.2.4. Rezultate scena ZARA1 [160].

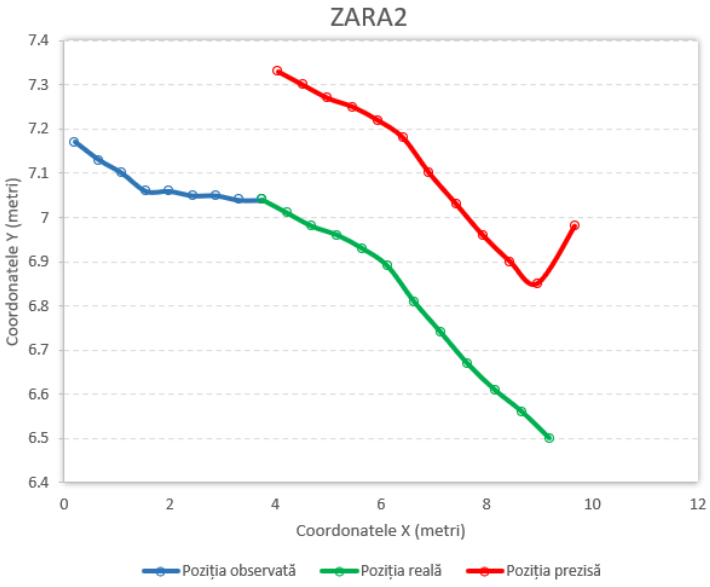

Figura 6.2.5. Rezultate scena ZARA2 [160].

Pentru [161], s-a efectuat un studiu de ablație pe seturile de date ETH și UCY pentru a valida performanța ST-GCNN și TXP-CNN bazate pe atenție în model. În ambele faze de antrenare și testare, fiecare experiment este definit cu aceleași valori de hiperparametri. Pentru a determina cea mai eficientă arhitectură, am testat diferite combinații de straturi ST-GCNN și TXP-CNN. Conform studiului nostru de ablație, cel mai bun model are un strat ST-GCNN și patru straturi TXP-CNN. În plus, odată cu creșterea numărului de straturi ST-GCNN, performanța modelului scade. Investigăm dacă creșterea sau scăderea dimensiunii modelului îmbunătățește eficacitatea învățării dintr-un număr mai mic de eșantioane de date. Datele de antrenament au fost alese în mod aleatoriu. Folosim aceleași date pentru toate modelele generate.

Cu structura propusă, cu un strat ST-GCNN și patru straturi TXP-CNN, pe setul de date ETH cu scena HOTEL, obținem cele mai bune rezultate pentru metricile noastre ADE/FDE (vezi Figura 6.2.6).

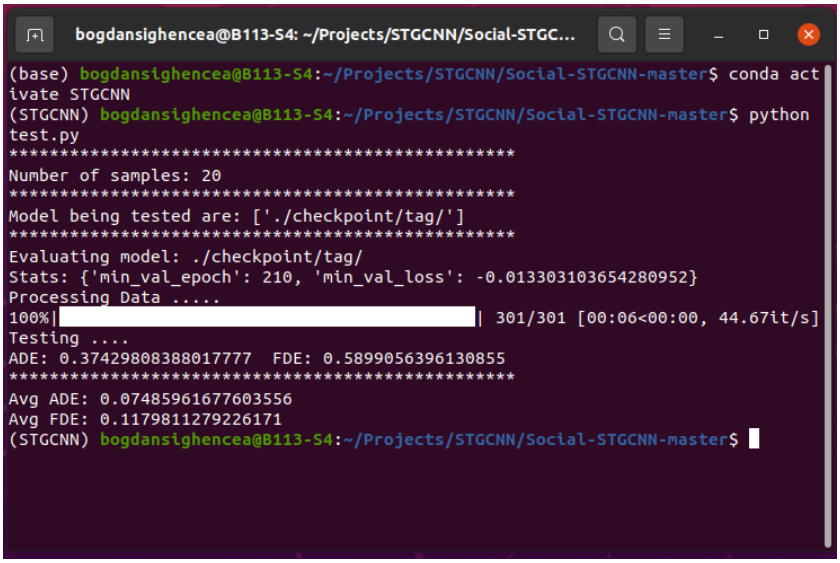

Figura 6.2.6. Cele mai bune rezultate ADE/FDE pentru arhitectura propusă pe setul de date ETH cu scena HOTEL.

A fost antrenată o altă structură de arhitectură cu două straturi ST-GCNN și un strat TXP-CNN, și s-a obținut pe setul de date UCY cu scena UNIV cele mai bune rezultate pentru metricile ADE/FDE (vezi Figura 6.2.7).

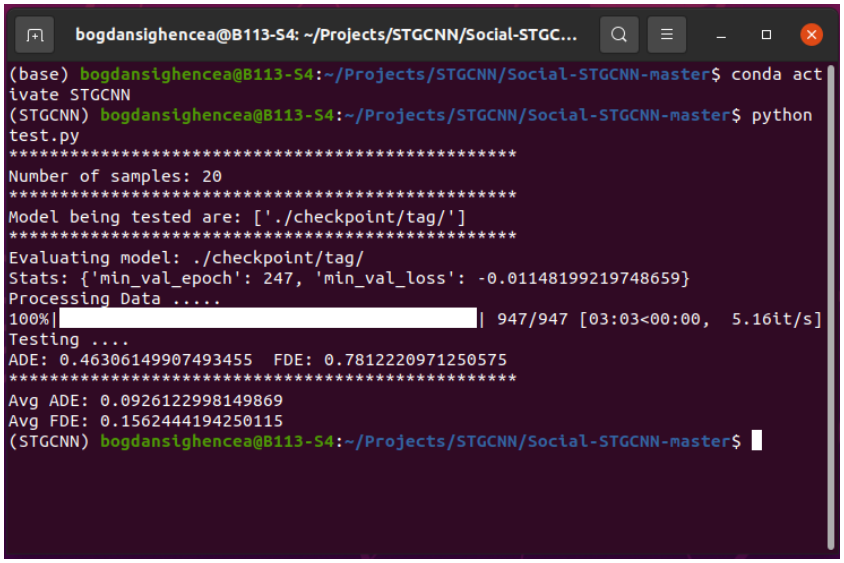

Figura 6.2.7. Cele mai bune rezultate ADE/FDE pentru arhitectura propusă pe setul de date UCY cu scena UNIV.

O altă metodă a fost prezentată în [163]. Aplicabilitatea metodei se referă în primul rând la traseul în arbore, care oferă o bună explicație pentru comportamentele viitoare de mișcare, cum ar fi mersul direct și apoi virarea la dreapta. Pentru a demonstra că abordarea poate selecta traseul cel mai apropiat în corelație cu realitatea, au fost create statistici în setul de testare care înregistrează rata de selectare a traseului cel mai apropiat în diferite estimări de tip "cel mai bun din K", unde K_{max}=20 reprezintă numărul de pași de timp viitor prezis. Am efectuat un experiment special pentru a demonstra că arborele este capabil să prezică traiectoria pietonilor fără niciun fel de antrenament. Fiecare experiment este stabilit cu setări identice de hiperparametri atât în etapele de antrenare, cât și în cele de testare. Rezultatele obținute sunt prezentate în Tabelul 6.2.2 pentru seturile de date ETH, UCY și SDD. Datele de antrenament au fost selectate în mod aleatoriu. Pentru evaluare, toate modelele au utilizat cel mai bun K din cele 20 de valori posibile. Modelele au folosit 8 cadre ca intrare și au prezis următoarele 12 cadre.

Tabel 6.2.2. Rezultate pe seturile de date ETH, UCY și SDD în funcție de numărul de pași de timp viitor prezis. rezultatele sunt raportate în termeni de metrici medii ADE/FDE. fontul îngroșat indică cel mai bun rezultat obținut.

K	ETH		UCY			SDD
	ETH	***HOTEL***	***UNIV***	***ZARA1***	***ZARA2***	***Toate scenele***
1	1.23/2.34	**2.23**/3.61	0.86/1.57	0.51/0.92	0.44/0.74	31.25/54.97
5	0.89/1.63	2.8/2.38	0.63/1.15	0.34/0.59	0.3/0.49	21.94/38.36
10	0.71/1.31	2.73/2.11	0.51/0.89	0.3/0.48	0.27/0.42	18.52/31.53
15	0.67/1.14	2.7/1.98	0.44/0.74	0.28/0.44	0.25/0.39	16.77/27.1
20	**0.64/1.07**	2.69/**1.85**	**0.41/0.67**	**0.27/0.43**	**0.24/0.38**	**15.84/25.17**

Rezultatele vizualizărilor sunt ilustrate în secțiunea următoare pentru scene similare. Graficele prezentate în continuare (Figura 6.2.8. - Figura 6.2.9.) arată că pietonii acordă atenție împrejurimilor/vecinătății lor.

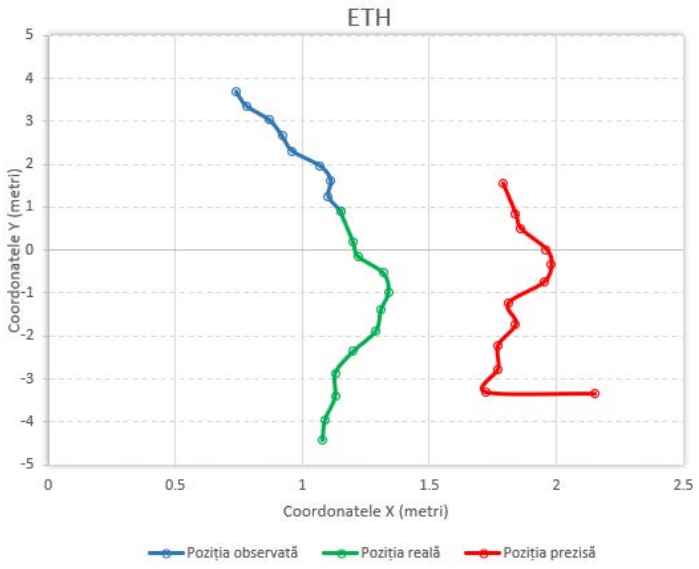

Figura 6.2.10. Rezultate scena ETH [163].

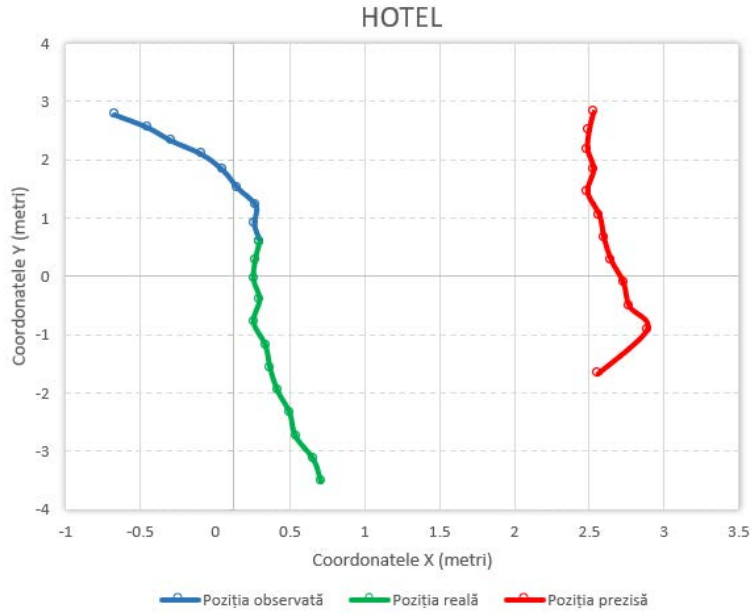

Figura 6.2.11. Rezultate scena HOTEL [163].

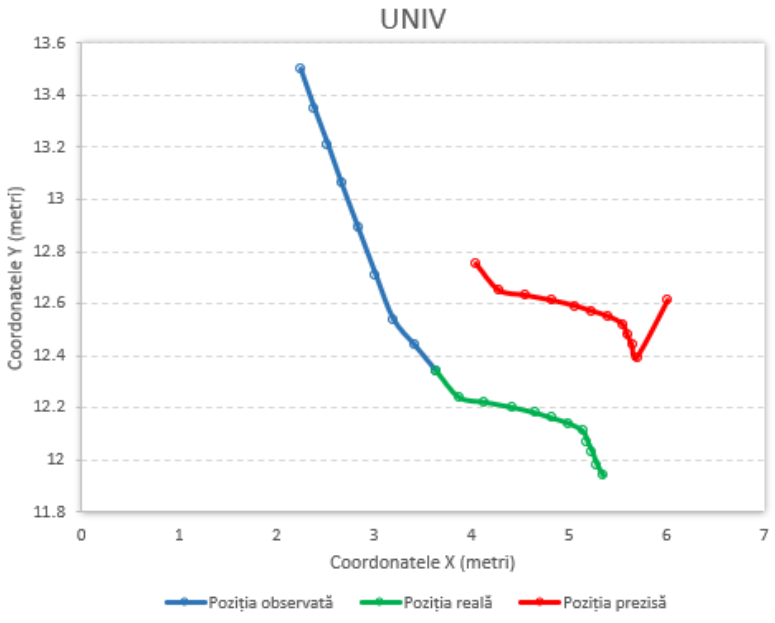

Figura 6.2.12. Rezultate scena UNIV [163].

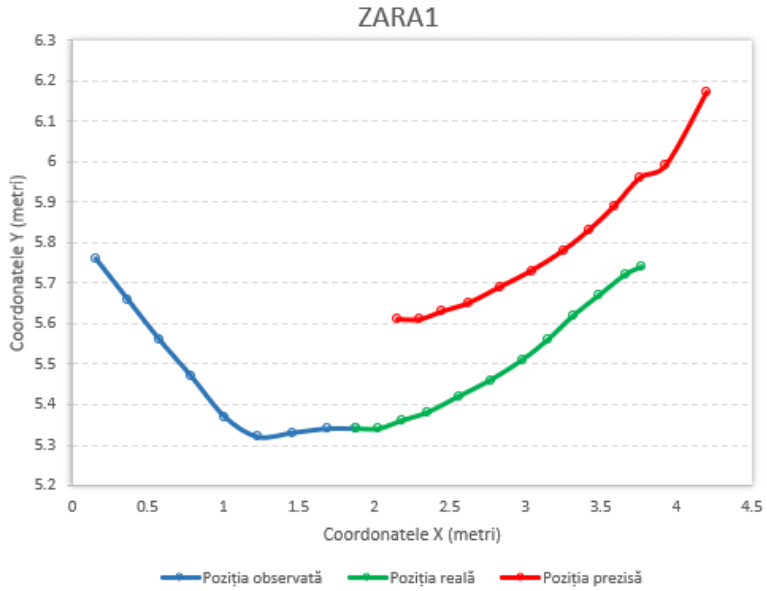

Figura 6.2.13. Rezultate scena ZARA1 [163].

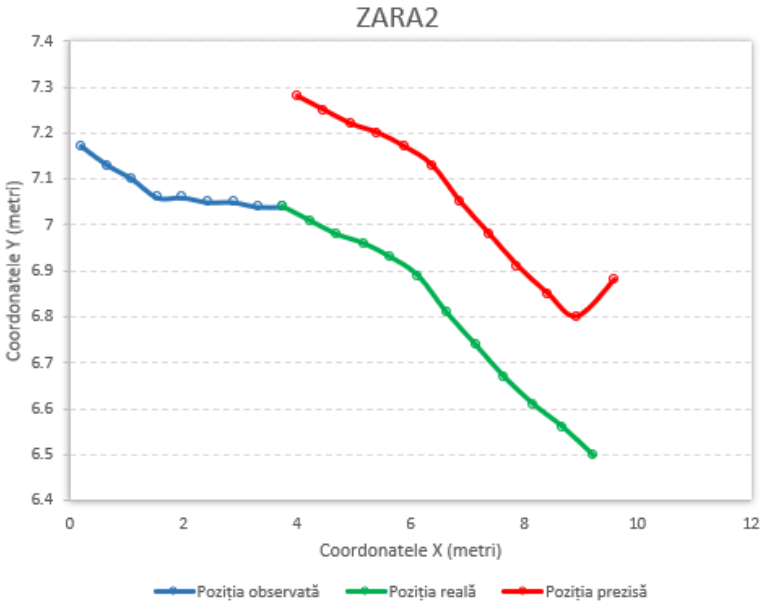

Figura 6.2.14. Rezultate scena ZARA2 [163].

6.3. Comparație cu alte metode de top din literatura de specialitate

Tabelul 6.3.1 prezintă comparația între performanțele celor mai bune metodele identificate în literatura de specialitate și cea propusă de mine în cadrul tezei de doctorat. Rezultatele sunt raportate pe seturile de date ETH, UCY și SDD în termeni de metrici de evaluare ADE/FDE. Se constată că în cele mai multe situații, metoda propusă depășește alte abordări de ultimă generație. În ceea ce privește metrica ADE, metodele dezvoltate au redus eroarea cu 3% pentru seturile de date ETH-UCY în comparație cu soluția de ultimă generație SR-LSTM-2 [167] și au redus eroarea cu 5% pentru setul de date SDD în comparație cu metoda de ultimă generație CGNS [75]. În ceea ce privește metrica FDE, abordarea prezentată a redus eroarea cu 10% în comparație cu metoda de referință de ultimă generație Social-STGCN [72]. În mod surprinzător, D-STGCN, care nu utilizează date de imagine ale scenei, depășește metode care o fac, cum ar fi SoPhie [63], STGAT [73], sau PIF [168].

Tabel 6.3.1. Rezultate cantitative ale metodelor de ultimă generație pentru seturile de date ETH, UCY și SDD definite în termenii metricilor ADE/FDE. Coloana AWG reprezintă rezultatele medii între scenele seturilor de date ETH-UCY. n/a înseamnă că lucrările respective nu au furnizat rezultate detaliate cu setul de date SDD.

Metode	SDD	ETH	HOTEL	UNIV	ZARA1	ZARA2	AWG
S-LSTM [33]	31.19/ 56.97	1.09/ 2.35	0.79/ 1.76	0.67/ 1.40	0.47/ 1.00	0.56/ 1.17	0.71/ 1.53
Social-STGCNN [72]	n/a	0.64/ 1.11	0.49/ 0.85	0.44/ 0.79	0.34/ 0.53	0.30/ 0.48	0.44/ 0.75
SR-LSTM [169]	n/a	0.63/ 1.25	0.37/ 0.74	0.51/ 1.10	0.41/ 0.90	0.32/ 0.70	0.44/ 0.93
SR-LSTM-2 [167]	n/a	0.58/ 1.13	0.31/ 0.62	0.50/ 1.10	0.41/ 0.90	0.33/ 0.73	0.43/ 0.89
S-GAN-P [59]	27.23/ 41.44	0.87/ 1.62	0.67/ 1.37	0.76/ 1.52	0.35/ 0.68	0.42/ 0.84	0.61/ 1.20
S-Ways [61]	n/a	**0.39**/ **0.64**	0.39/ 0.66	0.55/ 1.31	0.44/ 0.64	0.51/ 0.92	0.45 /0.83
SoPhie [63]	16.27/ 29.38	0.70/ 1.43	0.76/ 1.67	0.54/ 1.24	0.30/ 0.63	0.38/ 0.78	0.53/ 1.15
SSALVM (20) [170]	n/a	0.61/ 1.09	0.28/ 0.51	0.59/ 1.24	0.30/ 0.64	0.37/ 0.78	0.43/ 0.85
MATF-GAN [53]	27.82/5 9.31	1.33/ 2.49	0.51/ 0.95	0.56/ 1.19	0.44/ 0.93	0.34/ 0.73	0.64/ 1.26
PIF [168]	n/a	0.73/ 1.65	**0.30**/ 0.59	0.60/ 1.27	0.38/ 0.81	0.31/ 0.68	0.46/ 1.00
Social BiGAT[60]	n/a	0.69/ 1.29	0.49/ 1.01	0.55/ 1.32	0.30/ 0.62	0.36/ 0.75	0.47/ 0.99
STGAT [73]	n/a	0.65/ 1.12	0.35/ 0.66	0.52/ 1.10	0.34/ 0.69	0.29/ 0.60	0.43/ 0.83
CGNS [75]	15.84/**2 5.17**	0.62/ 1.40	0.70/ 0.93	0.48/ 1.22	0.32/ 0.59	0.35/ 0.71	0.49/ 0.97
Metodă propusă [160]	**15.18**/ 25.50	0.63/ 1.03	0.37/ **0.58**	0.46/ **0.78**	0.35/ 0.56	0.29/ 0.48	**0.42**/ **0.68**
Metodă propusă [163]	15.84/ **25.17**	0.64/ 1.07	2.29/ 1.85	**0.41**/ 0.67	**0.27**/ **0.43**	**0.24**/ **0.38**	0.77/ 0.88

7. CONCLUZII

Predicția traiectoriei pietonilor este unul dintre numeroasele domenii în care apariția rețelelor neuronale profunde au schimbat complet abordarea problemei. Înainte de 2016, modelele bazate pe dinamica fizică erau singura modalitate de a prezice cu exactitate traiectoriile viitoare ale pietonilor, dar în anul 2016 s-a demonstrat că învățarea profundă ar putea face acest lucru mai bine [33]. De atunci, au fost propuse diverse arhitecturi noi cu rezultate din ce în ce mai bune. Modelele bazate pe date, cum ar fi modelele folosind rețele neuronale convoluționale, recurente sau de tip graf, sunt foarte sensibile la cantitatea și calitatea datelor de antrenament și la modul în care aceste date sunt apoi integrate modelului.

În această teză, au fost analizate mai multe aspecte, printre care date, senzori și metodele actuale de ultimă generație aplicate pentru problema de predicție a traiectoriei pietonilor. Chiar dacă procesul de dezvoltare a unor soluții fiabile a fost foarte laborios până acum, este totuși necesar să se depună un efort mai mare pentru a realiza un sistem care să asigure securitatea pietonilor pe străzi. Lucrările selectate au fost examinate cu privire la mai mulți factori comuni, pentru a asigura o comparație simplă pentru cei interesați de acest subiect dar și pentru cititorii acestei teze.

Prin utilizarea abordărilor de învățare profundă, sistemele actuale sunt capabile să rezolve mai bine problema PTP. Aceste metode presupun o serie de locații pentru pietoni în ultimele câteva secunde și produc o serie de locații viitoare. În Capitolul 2 sunt identificate și descrise, ca și principiu de funcționare, arhitecturile DNN (RNN, CNN, LSTM, GAN și GNN) cele mai potrivite pentru problema predicției traiectoriei pietonilor. Aceste abordări nu sunt exclusive și sunt adesea utilizate în combinație hibridă.

Deși metodele PTP actuale au fost îmbunătățite considerabil, acestea pot fi încă actualizate pentru aplicații mai bune în lumea reală. Tehnicile de predicție care implică traiectoriile ca procese separate și nu se bazează pe modelarea specifică a interacțiunilor sociale pot fi destul de eficiente pe anumite seturi de date particulare în care traiectoriile sunt caracterizate de o energie de coliziune scăzută. În cadrul Capitolului 3 sunt prezentate cele mai bune soluții existente în literatura de specialitate ce folosesc rețele neuronale profunde pentru abordarea problemei PTP.

În Capitolul 4 sunt prezentate cele mai populare tehnologii de senzori folosite pentru problema PTP (Radar, LiDAR și Camera Video); este analizată și posibilitatea fuziunii informației primite de la acești senzori. În ceea ce privește tipul de informații dobândite, fiecare senzor este caracterizat de diferite puncte forte și puncte slabe. Pentru a obține percepția lumii reale, acești senzori sunt montați în interiorul sau exteriorul vehiculul pentru captarea traficului sau în diferite locații stradale în scopul supravegherii. Acestea oferă informații valoroase despre mișcarea și poziția pietonilor. Tot în cadrul acestui capitol sunt expuse cele mai noi seturi de date, care au un nivel crescut de densitate și o interacțiune mai mare între agenții participanți la trafic, în funcție de zonele unde au fost captate acestea. Aceste seturi de date ar putea oferi informații mai precise cu privire la calitatea algoritmilor de predicție deja propuși.

Capitolul 5 conține descrierea arhitecturii tuturor metodelor și modelelor neuronale publicate pentru a soluționa problema PTP, începând cu descrierea problemei și a dinamicii pietonale. Metricile (ADE și FDE) utilizate pentru compararea metodelor propuse cu alte metode sunt prezentate detaliat. Capitolul 6 se referă la rezultatele experimentale ale metodelor propuse și oferă compararea acestora cu cele mai bune soluții existente pentru PTP precum și vizualizarea diferitelor traiectorii prezise în diferite scene ale ETH, UCY și SDD.

7.1. Contribuții

Prezentul subcapitol subliniază contribuțiile aduse la rezolvarea cu acuratețe crescută a problemei predicției traiectoriei pietonilor.

În lucrarea [160] am abordată o nouă tehnică pentru predicția traiectoriei pietonilor, folosind rețele neuronale de tip GNN. Această soluție introduce o abordare bazată pe două componente: o rețea neuronală cu graf spațial (SGNN) pentru modelarea interacțiunii și o rețea neuronală cu graf temporal (TGNN) pentru extracția caracteristicilor de mișcare. SGNN utilizează o metodă de atenție pentru a colecta periodic interacțiuni spațiale între toți pietonii. TGNN folosește și o metodă de atenție, pentru a colecta modelul de mișcare temporală al fiecărui pieton. Cu o dimensiune variabilă mai mică (date și model) și o rată de predicție mai bună, D-STGCN este mai compactă și mai eficientă decât alte soluții SOTA, oferind rezultate experimentale mai bune luând în considerare valorile erorii medii de deplasare (ADE) și erorii finale de deplasare (FDE).

O altă soluție pe care am publicat-o [163] prezintă o abordare bazată pe arbori GNN pentru a face față provocării de predicție multimodală. Arborele este proiectat pe baza datelor observate și este, de asemenea, utilizat pentru a prezice traiectoriile viitoare. În comparație cu abordările anterioare care utilizează variabile latente implicite pentru a descrie posibile traiectorii viitoare, comportamentele de mișcare pot fi reprezentate direct cu o abordare de tip arbore (de exemplu, mergeți drept și apoi virați la stânga) și, astfel, sunt oferite de către model traiectorii mai potrivite din punct de vedere social.

Pentru a obține informații despre interacțiune dintre pietoni dar și dintre pietoni și mediul înconjurător am propusă o soluție [161] bazată pe două componente (codificator și decodificator). Această soluție a fost implementată folosind GNN + CNN și au fost obținute rezultate foarte bune în termenii ADE și FDE pentru PTP.

În cadrul acestui articol [30] au fost analizate cele mai recente soluții bazate pe învățare profundă pentru problema predicție traiectoriei pietonilor împreună cu senzorii utilizați (camera, LiDAR și radar) și metodologiile de procesare aferente. S-au prezentat comparativ performanțele fiecărui senzor auto (radar, LiDAR și cameră) prin evidențierea avantajelor și dezavantajelor acestora în diferite sarcini, de exemplu detecția și clasificarea obiectelor, estimarea distanței, detecția marcajelor rutiere și a limitelor drumului etc. Au fost identificate cele mai importante seturi de date disponibile public, folosite de către cercetători în implementarea soluțiilor de tip PTP și a indicatorilor de performanță utilizați în procesul de evaluare.

În cadrul articolelor pe care le-am publicat [157], [158] au fost descrise metodele pentru predicția bazată pe filtrul alfa-beta-gama. S-a efectuat o analogie între filtrul Kalman și filtrul alfa-beta-gama cu scopul de a identifica acurateței metodei. Rezultatele au arătat că utilizarea unor astfel de arhitecturi pot oferii o soluție fiabilă pentru PTP.

7.1.1. Listă lucrări

- Sighencea B.I.; Stanciu I.R.; Căleanu C.D., "D-STGCN: Dynamic Pedestrian Trajectory Prediction Using Spatio-Temporal Graph Convolutional Networks", Electronics 2023, 12(3), 611, WOS:000933823000001.

- Sighencea B.I, Stanciu R.I., Șorândaru C., Căleanu C.D., "The Alpha-Beta Family of Filters to Solve the Threshold Problem: A Comparison", Mathematics 2022; 10(6):880, WOS:000774078800001.

- Sighencea B.I., "Pedestrian Trajectory Prediction Based on Tree Method using Graph Neural Networks", 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Hagenberg / Linz, Austria, 2022, pp. 245-249.

- Sighencea B.I., Stanciu R.I. and Căleanu C.D., "Pedestrian Trajectory Prediction in Graph Representation Using Convolutional Neural Networks," IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timișoara, Romania, 2022, pp. 000243-000248.

- Sighencea B.I, Stanciu R.I., Căleanu C.D, "A Review of Deep Learning-Based Methods for Pedestrian Trajectory Prediction", Sensors 2021; 21(22):7543, WOS:000778251600009.

- Sighencea B.I., Stanciu R.I. and Sorandaru C., "Using the α-β-γ Filter to solve the Threshold Problem, IEEE EUROCON 2021 - 19th International Conference on Smart Technologies, Lviv, Ukraine, 2021, pp. 45-50, WOS:000728121700008.

7.2. Direcții viitoare

Din rezultatele și concluziile prezentate în teză, pot fi propuse mai multe linii viitoare de lucru. Acestea corespund diferitelor aspecte care nu au fost rezolvate sau necesită o analiză suplimentară pentru a îmbunătăți performanța:

- Trebuie luat în considerare un număr mai mare de secvențe, deoarece mașinile, bicicliștii sau chiar persoanele în vârstă nu sunt incluse în seturile de date ETH, UCY și Stanford Drone. După cum se susține în [171], pietonii în vârstă aleg decizii mai periculoase decât tinerii, chiar dacă în mod normal durează mai mult timp pentru a le lua.

- Testarea tuturor algoritmilor cu diferite tipuri de caracteristici sau combinarea acestora poate îmbunătăți performanța metodelor propuse în această teză. De exemplu, pot fi utilizate și caracteristicile de mișcare obținute prin intermediul fluxului optic sau al imaginilor istoricului mișcării în locul deplasărilor pietonale extrase din pozițiile corpului. În plus, la un nivel superior, combinația de informații bazate pe context, împreună cu o evaluare a situației și o analiză a limbajului caroseriei pietonilor ar permite dezvoltarea unui sistem automat de frânare de urgență mai fiabil. Astfel, înțelegerea

scenei, detectarea pietonilor și algoritmii de predicție sunt linii interesante de cercetare în PTP.

- Pentru a obține siluete pietonale mai precise, ar putea fi dezvoltate abordări de captare a mișcării fără marcaj bazate pe CNN, cum ar fi algoritmul propus în [172], în locul algoritmului bazat pe intrări de tip coordonate în imagine.

- Crearea unui set extins de date privind situațiile pietonale reale ar face posibilă compararea diferitelor abordări în condiții similare. Metodologia de identificare a evenimentelor propusă în această teză ar ajuta cercetătorii să determine diferitele activități pietonale.

- Testarea algoritmilor în vehiculele aflate în mișcare în timp real. Pentru a face acest lucru, ego-mișcarea ar trebui să fie compensată la fiecare pas de arhitectură hardware de înaltă performanță.

BIBLIOGRAFIE

[1] S. Lefèvre et al., „A survey on motion prediction and risk assessment for intelligent vehicles", Robomech J, 2014.
[2] European Road Safety Observatory, „Annual Accident Report", 2020.
[3] WHO, „Global Status Report on Road Safety", 2018.
[4] ITF, Pedestrian Safety, Urban Space and Health, 2012.
[5] D. Gálvez-Pérez et al., "The Influence of Built Environment Factors on Elderly Pedestrian Road Safety in Cities: The Experience of Madrid", Int. J. Environ. Res. Public Health, Vol. 19, 2022.
[6] T. Winkle, „Safety benefits of automated vehicles: Extended findings from accident research for development, validation, and testing", Autonomous Driving. Springer, 2016.
[7] S. Ahmed et al., „Pedestrian and cyclist detection and intent estimation for autonomous vehicles: A survey", Applied Sciences, Vol. 9, 2019.
[8] I. N. Aizenberg, N. N. Aizenberg, and J. P. Vandewalle, "Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications"., Kluwer Academic Publishers, 2000.
[9] K. Gurney, An Introduction to Neural Networks. USA: Taylor & Francis, Inc., 1997.
[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[11] J. P. Werbos,. "Backpropagation through time: what it does and how to do it". In: Proceedings of the IEEE 78.10, pp. 1550–1560, 1990.
[12] J. Herbert, Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach. Vol. 5. GMD-Forschungszentrum Informationstechnik Bonn, 2002.
[13] D. Krueger, et al., "Zoneout: Regularizing rnns by randomly preserving hidden activations". In: arXiv preprint arXiv:1606.01305, 2016.
[14] G. Yarin and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks". In: Advances in neural information processing systems, pp. 1019–1027. 2016.
[15] S. Stanislau, A. Severyn, and E. Barth, "Recurrent dropout without memory loss". In: arXiv preprint arXiv:1603.05118, 2016.
[16] M. Stephen, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models". In: arXiv preprint arXiv:1708.02182, 2017.
[17] C. Tim, et al., "Recurrent batch normalization". In: arXiv preprint arXiv:1603.09025, 2016.
[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput. 9, 1735–1780, 1997.
[19] J. Gu, et al., „Recent Advances in Convolutional Neural Networks. Pattern Recognit". 77, 354–377, 2018.
[20] S. Ren, et al., „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In Advances in Neural Information Processing Systems; Neural Information Processing Systems Foundation, Inc.: Montreal, QC, Canada, pp. 91–99, 2015.
[21] A. Krizhevsky, et al., „Imagenet classification with deep convolutional neural networks". In Advances in Neural Information Processing Systems; Neural Information Processing Systems Foundation, Inc.: Montreal, QC, Canada, pp. 1097–1105, 2012.

[22] G. Jiuxiang, et al., „Recent advances in convolutional neural networks", Pattern Recognition, Volume 77, Pages 354-377, 2018.
[23] A. Karpathy and F.F. Li, „Cs231n: Convolutional neural networks for visual recognition", 2015.
[24] I. J. Goodfellow et al. "Generative Adversarial Nets". In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS14. MIT Press, pp. 2672–2680, 2014.
[25] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A gentle introduction to graph neural networks," Distill, 2021.
[26] M. M. Bronstein et al. "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges". In: CoRR abs/2104.13478, arXiv: 2104.13478, 2021.
[27] H. Maron et al. "Invariant and Equivariant Graph Networks". In: International Conference on Learning Representations. 2019.
[28] H. Maron et al. "Provably powerful graph networks". In: Advances in Neural Information Processing Systems, pp. 2156–2167, 2019.
[29] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," CoRR, vol. abs/1812.08434, 2018.
[30] B. I. Sighencea, R. I. Stanciu and C. D. Căleanu, „A Review of Deep Learning-Based Methods for Pedestrian Trajectory Prediction". Sensors, 21(22):7543, 2021.
[31] L. Sun, et al., „3DOF Pedestrian Trajectory Prediction Learned from Long-Term Autonomous Mobile Robot Deployment Data". In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, pp. 5942–5948, 21–25 May 2018.
[32] K. Fragkiadaki, et al., „Recurrent Network Models for Human Dynamics". In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, pp. 4346–4354, 7–13 December 2015.
[33] A. Alahi et al., „Social LSTM: Human Trajectory Prediction in Crowded Spaces". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 961–971, 30 June 2016.
[34] S. Pellegrini, „You'll never walk alone: Modeling social behavior for multi-target tracking". In Proceedings of the IEEE 12th International Conference on Computer Vision, pp. 261–268, 27 September–4 October 2009.
[35] A. Lerner, „Crowds by example". Comput. Graph. Forum, 26, 655–664, 2007.
[36] S. Dai, „Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction". IEEE Access, 7, 38287–38296, 2019.
[37] L. Xin, „Intention aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks". In Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 1441–1446, 4–7 November 2018.
[38] N.Lee, et al., „DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2165–2174, 21–26 July 2017.
[39] A. Geiger, et al., „Are we ready for autonomous driving? The KITTI vision benchmark suite". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361, 16–21 June 2012.
[40] A. Robicquet, et al. „Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes". In Computer Vision–ECCV, Springer, Volume 9912, 2016.

[41] S. Zheng, Y. Yue and J. Hobbs, „Generating long-term trajectories using deep hierarchical networks". In Proceedings of the Thirtieth Conference on Neural Information Processing Systems (NIPS), 5–10 December 2016.
[42] E. Zhan, et al. „Generative multi-agent behavioral cloning". In Proceedings of the 35th International Conference on Machine Learning, 10–15 July 2018.
[43] J. Martinez, et al. „On Human Motion Prediction Using Recurrent Neural Networks". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4674–4683, 21–26 July 2017.
[44] R. Hug, et al. „Particle-based Pedestrian Path Prediction using LSTM-MDL Models". In Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2684–2691, 4–7 November 2018.
[45] A. Bhattacharyya, et al. „Long-Term On-board Prediction of People in Traffic Scenes under Uncertainty". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4194–4202, 18–23 June 2018.
[46] M. Cordts, et al. „The Cityscapes Dataset for Semantic Urban Scene Understanding". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3213–3223, 27–30 June 2016.
[47] T. Salzmann, et al. „Trajectron++: Dynamically Feasible Trajectory Forecasting with Heterogeneous Data". In Computer Vision–ECCV, Proceedings of the 16th European Conference, Springer, Volume 12363, 23–28 August 2020.
[48] H. Caesar, et al. „nuscenes: A multimodal dataset for autonomous driving". CoRR, abs/1903.11027, 2019.
[49] H. Xue, et al. „SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction". In Proceedings of the IEEEWinter Conference on Applications of Computer Vision (WACV), pp. 1186–1194, 12–15 March 2018.
[50] B. Benfold and I. Reid, „Guiding visual surveillance by tracking human attention". In Proceedings of the British Machine Vision Conference (BMVC), 7–10 September 2009.
[51] E. Rehder, et al., "Pedestrian prediction by planning using deep neural networks". In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 21–25 May 2018.
[52] S. Hoermann, et al., „Dynamic Occupancy Grid Prediction for Urban Autonomous Driving: A Deep Learning Approach with Fully Automatic Labeling". In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2056–2063, 21–25 May 2018.
[53] T. Zhao, et al., „Multi-Agent Tensor Fusion for Contextual Trajectory Prediction". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12118–12126, 15–21 June 2019.
[54] S. Yi, et al., „Pedestrian Behavior Understanding and Prediction with Deep Neural Networks". In Computer Vision–ECCV, Proceedings of the Amsterdam, The Netherlands, Springer, Volume 9905, 11–14 October 2016.
[55] J, Doellinger, et al., „Predicting Occupancy Distributions of Walking Humans with Convolutional Neural Networks". IEEE Robot. Autom. Lett., 3, 1522–1528, 2018.
[56] F. Marchetti, et al., „MANTRA: Memory Augmented Networks for Multiple Trajectory Prediction". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7141–7150, 16–18 June 2020.
[57] R. Wang, et al., „Multi-information-based convolutional neural network with attention mechanism for pedestrian trajectory prediction". Image Vis. Comput. 107, 104110, 2021.

[58] T. Fernando, et al., „Tracking by Prediction: A Deep Generative Model for Mutli-person Localisation and Tracking". In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1122–1132, 12–15 March 2018.
[59] A. Gupta, et al., „Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2255–2264, 18–23 June 2018.
[60] V. Kosaraju, et al., „Social-BiGAT: Multimodal trajectory forecasting using Bicycle-GAN and graph attention networks". In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Volume 32, 8–14 December 2019.
[61] J. Amirian, et al., „Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2964–2972, 16–17 June 2019.
[62] X. Chen, et. Al., „Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In Proceedings of the Advances in neural information processing systems, pp. 2172–2180, 5–10 December 2016.
[63] A. Sadeghian, et al., „SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1349–1358, 15–21 June 2019.
[64] P. W. Battaglia, et al., „Relational inductive biases, deep learning, and graph networks". arXiv:1806.01261, 2018.
[65] J. Bruna, et al., „Spectral networks and locally connected networks on graphs", arXiv:1312.6203, 2013.
[66] M. Defferrard, et al., „Convolutional neural networks on graphs with fast localized spectral filtering". In Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 3844–3852, 5–10 December 2016.
[67] T. Kipf, et al., „Semi-supervised classification with graph convolutional networks". In Proceedings of the International Conference on Learning Representations, 24–26 April 2017.
[68] W. Hamilton, et al., „Inductive representation learning on large graphs". In Proceedings of the Annual Conference on Advances in Neural Information Processing Systems, pp. 1024–1034, 4–9 December 2017.
[69] S. Yan, et al., „Spatial temporal graph convolutional networks for skeleton-based action recognition". In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 7444–7452, 2–7 February 2018.
[70] P. Velickovic, et al., „Graph attention networks". In Proceedings of the 6th International Conference on Learning Representations (ICLR), 30 April–3 May 2018.
[71] P. Isola, et al., „Image-to-Image Translation with Conditional Adversarial Networks". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976, 21–26 July 2017.
[72] A. Mohamed, et al., „Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14412–14420, 16–18 June 2020.

[73] Y. Huang, et al., „STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction". In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 6271–6280, 27 October–2 November 2019.
[74] A. Vemula, et al., „Social attention: Modeling attention in human crowds". In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 4601–4607, 21–26 May 2018.
[75] J. Li, et al., „Conditional Generative Neural System for Probabilistic Trajectory Prediction". In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6150–6156, 4–8 November 2019.
[76] J. Ziegler, et al., „Making Bertha Drive—An Autonomous Journey on a Historic Route". IEEE Intell. Transp. Syst. Mag., 6, 8–20, 2014.
[77] C. Guo, et al., „Cooperation between driver and automated driving system: Implementation and evaluation". Transp. Res. Part F Traffic Psychol. Behav., 61, 314–325, 2019.
[78] F. M. Ortiz, et al., „Vehicle Telematics via Exteroceptive Sensors: A Survey". arXiv:2008.12632, 2020.
[79] Yole Developpement. MEMS and Sensors for Automotive: Market & Technology Report. 2017. Disponibil online: https://bit.ly/2X5pL70 (accessed on 23 July 2021).
[80] A. Lang, et al., „Pointpillars: Fast encoders for object detection from point clouds", 2018.
[81] X. Chen, et al., „Multi-view 3d object detection network for autonomous driving". IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[82] Y. Zhou, et al., „Voxelnet: End-to-end learning for point cloud based 3d object detection". IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[83] J. Kocic, et al., „Sensors and sensor fusion in autonomous vehicles", 2018.
[84] H. Sjafrie, „Introduction to Self-Driving Vehicle Technology", Chapman and Hall/CRC, 2019.
[85] H. H, Meinel, "Evolving automotive radar: From the very beginnings into the future". In Proceedings of the 8th European Conference on Antennas and Propagation (EuCAP 2014), pp. 3107–3114, 6–11 April 2014.
[86] G. Reina, et al., „Radar Sensing for Intelligent Vehicles in Urban Environments". Sensors, 15, 14661–14678, 2015.
[87] J. Hasch, „Driving towards 2020: Automotive radar technology trends". In Proceedings of the IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), pp. 1–4, 27–29 April 2015.
[88] S. Kim, et al., „Moving Target Classification in Automotive Radar Systems Using Convolutional Recurrent Neural Networks". In Proceedings of the 26th European Signal Processing Conference (EUSIPCO), pp. 1482–1486, 3–7 September 2018.
[89] M. Wicks, et al., „Space-time adaptive processing: A knowledge-based perspective for airborne radar". IEEE Signal Process. Mag., 23, 51–65, 2006.
[90] M. A. Richards, et al., „Principles of Modern Radar", Scitech Publishing: Raleigh, 2010.
[91] H. Rohling, „Radar CFAR Thresholding in Clutter and Multiple Target Situations". IEEE Trans. Aerosp. Electron. Syst., AES-19, 608–621, 1983.
[92] Y. Ding, et al., „Micro-Doppler Trajectory Estimation of Pedestrians Using a Continuous-Wave Radar". IEEE Trans. Geosci. Remote Sens., 52, 5807–5819, 2014.

[93] K. Kulpa, „The CLEAN type algorithms for radar signal processing". In Proceedings of the Microwaves, Radar and Remote Sensing Symposium, pp. 152–157, 22–24 September 2008.
[94] V.C. Chen, et al., „Micro-Doppler effect in radar: Phenomenon, model, and simulation study". IEEE Trans. Aerosp. Electron. Syst., 42, 2–21, 2006
[95] J. Ahtiainen, et al., „Radar based detection and tracking of a walking human". IFAC Proc. Vol. 43, 437–442, 2020.
[96] P. Held, et al., „Radar-Based Analysis of Pedestrian Micro-Doppler Signatures Using Motion Capture Sensors". In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Changshu, pp. 787–793, 26–30 June 2018.
[97] A. Dubey, et al., „A Bayesian Framework for Integrated Deep Metric Learning and Tracking of Vulnerable Road Users Using Automotive Radars". IEEE Access, 9, 68758–68777, 2021.
[98] P. Khomchuk, et al., „Pedestrian motion direction estimation using simulated automotive MIMO radar". IEEE Trans. Aerosp. Electron. Syst., 52, 1132–1145, 2016.
[99] M. Gilmartin, "INTRODUCTION TO AUTONOMOUS MOBILE ROBOTS, by Roland Siegwart and Illah R. Nourbakhsh", MIT Press, xiii+ 321 pp., ISBN 0-262-19502-X. Robotica 2005, 23, 271–272. 2005.
[100] C. Zou, et al., „Learning motion field of LiDAR point cloud with convolutional networks". Pattern Recognit. Lett., 125, 514–520, 2019.
[101] B. Li, et al., „Vehicle detection from 3D Lidar using fully convolutional network". In Robotics: Science and Systems, Proceedings of the 2016 Robotics: Science and Systems Conference, 18–22 June 2016.
[102] H. Wang, et al., „Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle". Robot. Auton. Syst., 88, 71–78, 2017.
[103] H. Wang, et al., A 64-line Lidar-based Road obstacle sensing algorithm for intelligent vehicles". Sci. Program, 6385104, 2018.
[104] J. Jung, et al., „Efficient and robust lane marking extraction from mobile Lidar point clouds". J. Photogramm. Remote Sens., 147, 1–18, 2019.
[105] J. Zhao, et al., „Probabilistic Prediction of Pedestrian Crossing Intention Using Roadside LiDAR Data". IEEE Access, 7, 93781–93790, 2019.
[106] D. D. Lewis, „Naive (Bayes) at forty: The independence assumption in information retrieval". In Proceedings of the European Conference on Machine Learning, Chemnitz, Germany, Springer, pp. 4–15., 21–23 April 1998.
[107] J. Wu, et al., „Automatic Background Filtering Method for Roadside LiDAR Data". Transp. Res. Rec., 2672, 106–114, 2018.
[108] K. Liu, et al., „Pedestrian Detection with Lidar Point Clouds Based on Single Template Matching". Electronics, 8, 780, 2019.
[109] G. Melotti, et al., „CNN-LIDAR pedestrian classification: Combining range and reflectance data". In Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp. 1–6, 12–14 September 2018.
[110] J. Wang, et al., „LIDAR and vision based pedestrian detection and tracking system". In Proceedings of the IEEE International Conference on Progress in Informatics and Computing (PIC), pp. 118–122, 18–20 December 2015.
[111] K. Granström, et al., „Pedestrian tracking using Velodyne data—Stochastic optimization for extended object tracking". In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 39–46, 11–14 June 2017.
[112] F. Bu, et al., „Pedestrian Planar LiDAR Pose (PPLP) Network for Oriented Pedestrian Detection Based on Planar LiDAR and Monocular Images". IEEE Robot. Autom. Lett., 5, 1626–1633, 2020.

[113] B. Völz, et al., „A data-driven approach for pedestrian intention estimation”. In Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 2607–2612, 1–4 November 2016.
[114] F. Bastien, et al., „Theano: New features and speed improvements”. In Proceedings of the Twenty-Sixth Conference on Neural Information Processing Systems Workshop, 3–8 December 2012.
[115] S. Dieleman, et al., „Lasagne: First Release”. Disponinbil online: https://zenodo.org/record/27878#.YY8dFMozY2w (accesat în 7 Noiembrie 2021).
[116] E. Mohammadbagher, et al., “A. Real-time Pedestrian Localization and State Estimation Using Moving Horizon Estimation”. In Proceedings of the IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–7, 20–23 September 2020.
[117] R. Guidolini, et al., „Handling Pedestrians in Crosswalks Using Deep Neural Networks in the IARA Autonomous Car”. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), pp. 1–8, 8–13 July 2018.
[118] J. W. Miller, et al., „Camera performance considerations for automotive applications”. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 26 April–1 May 2004.
[119] M. Gressmann, et al., „Surround view pedestrian detection using heterogeneous classifier cascades”. In Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1317–1324, 5–7 October 2011.
[120] Y. Cai, et al., „Pedestrian Motion Trajectory Prediction in Intelligent Driving from Far Shot First-Person Perspective Video”. IEEE Trans. Intell. Transp. Syst., pp. 1–16, 2021.
[121] Y. Bar-Shalom, et al., „Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software”, John Wiley & Sons: Hoboken, 2001.
[122] C. G. Keller, et al., „Will the Pedestrian Cross? A Study on Pedestrian Path Prediction”. IEEE Trans. Intell. Transp. Syst., 15, 494–506, 2013.
[123] P. Afsar, et al., „Automatic human trajectory destination prediction from video”. Expert Syst. Appl., 110, 41–51, 2018.
[124] O. Styles, et al., „Multi-Camera Trajectory Forecasting: Pedestrian Trajectory Prediction in a Network of Cameras”. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern RecognitionWorkshops (CVPRW), pp. 4379–4382, 14–19 June 2020.
[125] Y. Sun, et al., „See the Future: A Semantic Segmentation Network Predicting Ego-Vehicle Trajectory with a Single Monocular Camera”. IEEE Robot. Autom. Lett., 5, 3066–3073, 2020.
[126] A. Dosovitskiy, et al., „FlowNet: Learning Optical Flow with Convolutional Networks”. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2758–2766, 7–13 December 2015.
[127] X. U. Zhou, et al., „ACNN: A Full Resolution DCNN for Medical Image Segmentation”. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 8455–8461, 31 May–31 August 2020.
[128] L. C. Chen, et al., „DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. IEEE Trans. Pattern Anal. Mach. Intell., 40, 834–848, 2017.
[129] A. Loukkal, et al., „Driving among Flatmobiles: Bird-Eye-View occupancy grids from a monocular camera for holistic trajectory planning”. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 51–60, 5–9 January 2021.

[130] R. Chandra, et al., „Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions”. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8483–8492, 15–20 June 2019.
[131] T. Yagi, et al., „Future person localization in first-person videos”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7593–7602, 18–23 June 2018.
[132] J. Qiu, et al., „Indoor Future Person Localization from an Egocentric Wearable Camera”. In Proceedings of the IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS), 27 September–1 October 2021.
[133] J. Zhong, et al., „Pedestrian Motion Trajectory Prediction with Stereo-Based 3D Deep Pose Estimation and Trajectory Learning”. IEEE Access, 8, 23480–23486, 2020.
[134] M. Meyer and G. Kuschk, „Deep Learning Based 3D Object Detection for Automotive Radar and Camera”. In Proceedings of the 16th European Radar Conference (EuRAD), pp. 133–136, 2–4 October 2019.
[135] Z. Zhang, et al., „Prediction of Pedestrian Risky Level for Intelligent Vehicles”. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 169–174, 23 June 2020.
[136] WCP [Woodside Capital Partners]. Beyond the Headlights: ADAS and Autonomous Sensing. 2016. Disponibil online: https://secureservercdn.net/198.71.233.189/fzs.2d0.myftpupload.com/wp-content/uploads/2016/12/20160927-Auto-Vision-Systems-Report_FINAL.pdf (accesat în 5 Noiembrie 2021).
[137] S. Hwang, et al., “Multispectral pedestrian detection: Benchmark dataset and baseline”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1037–1045, 7–12 June 2015.
[138] P. Wang, et al., „The apolloscape open dataset for autonomous driving and its application”. IEEE transactions on pattern analysis and machine intelligence, 2019.
[139] M. F. Chang, et al., „Argoverse: 3D Tracking and Forecasting with Rich Maps”. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8740–8749, 15–21 June 2019.
[140] P. Dollar, et al., „Pedestrian detection: A benchmark”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 304–311, 20–25 June 2009.
[141] N. Schneider, et al., „Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study”. In Proceedings of the German Conference on Pattern Recognition, Springer, 2013; Volume 8142, 3–6 September 2013.
[142] R. Kesten, et al., „Lyft Level 5 av Dataset”. Disponibil online: https://level5.lyft.com/dataset (accesat în 19 Mai 2020).
[143] P. Sun, et al., “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2443–2451, 14–19 June 2020.
[144] Y. Choi, et al., „KAIST Multi-Spectral Day/Night Data Set for Autonomous and Assisted Driving”. IEEE Trans. Intell. Transp. Syst., 19, 934–948, 2018.
[145] A. Rasouli, et al., „PIE: A Large-Scale Dataset and Models for Pedestrian Intention Estimation and Trajectory Prediction”. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6261–6270, 27 October–3 November 2019.

[146] J. Bock, et al., „The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections". In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 1929–1934, 19 October–13 November 2020.
[147] E. Strigel, et al., „The Ko-PER intersection laserscanner and video dataset". In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1900–1901, 8–11 October 2014.
[148] F. Yu, et al., „BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2633–2642, 16–18 June 2020.
[149] S. Oh, et al., „A large-scale benchmark dataset for event recognition in surveillance video". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3153–3160, 20–25 June 2011.
[150] G. Awad, et al., „Benchmarking video activity detection video captioning and matching video storytelling linking and video search". In Proceedings of the Trecvid, Gaithersburg, 13 November 2018.
[151] S. Oh, et al., „A large-scale benchmark dataset for event recognition in surveillance video". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3153–3160, 20–25 June 2011.
[152] J. Ferryman, et al., „PETS2009: Dataset and challenge". In Proceedings of the Twelfth IEEE InternationalWorkshop on Performance Evaluation of Tracking and Surveillance, pp. 1–6, 7–9 December 2009.
[153] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," Annals of Mathematical Statistics, vol. 27, no. 3, pp. 832–837, 1956.
[154] E. Parzen, "On estimation of a probability density function and mode," Annals of Mathematical Statistics, vol. 33, no. 3, pp. 1065–1076, 1962.
[155] N. Thacker and T. Lacey, "Tutorial: The kalman filter," Imaging Science and Biomedical Engineering Division - Tina Memo no. 1006-002. pp. 133–140, 1998.
[156] D. L. Simon, "The Extended Kalman Filter: An Interactive Tutorial." 2016 [Online], Disponibil la: https://home.wlu.edu/~levys/kalman_tutorial/
[157] B. I. Sighencea, I. Rares Stanciu and C. Sorandaru, "Using the α-β-γ Filter to solve the Threshold Problem," IEEE EUROCON 2021 - 19th International Conference on Smart Technologies, pp. 45-50, 2021.
[158] B. I. Sighencea, R. I. Stanciu, C. Șorândaru and C. D. Căleanu, „The Alpha-Beta Family of Filters to Solve the Threshold Problem: A Comparison". Mathematics, 10, 880, 2022.
[159] H. Ying-Qing, et al., „Application of adaptive α-β filtering algorithm to electronic image stabilization". International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), pp. 322-325, 2011.
[160] B. I. Sighencea, I. R. Stanciu and C. D. Căleanu, „D-STGCN: Dynamic Pedestrian Trajectory Prediction Using Spatio-Temporal Graph Convolutional Networks". Electronics, 12, 611, 2023.
[161] B. I. Sighencea, R. I. Stanciu and C. D. Căleanu, "Pedestrian Trajectory Prediction in Graph Representation Using Convolutional Neural Networks," 2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2022, pp. 000243-000248.
[162] S. Bai, et al., „An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling". arXiv preprint 2018.
[163] B. I. Sighencea, "Pedestrian Trajectory Prediction Based on Tree Method using Graph Neural Networks," 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 245-249, 2022.

[164] L. Shi, et al., "Social Interpretable Tree for Pedestrian Trajectory Prediction", 36th AAAI Conference on Artificial Intelligence, Feb-Mar. 2022.
[165] K. He, et al., „Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034, 7–13 December 2015.
[166] R. Chai, et al., „Multiobjective Overtaking Maneuver Planning for Autonomous Ground Vehicles". IEEE Trans. Cybern., 51, 4035–4049, 2021.
[167] P. Zhang, et al. „Social-Aware Pedestrian Trajectory Prediction via States Refinement LSTM". IEEE Trans. Pattern Anal. Mach. Intell. 2022.
[168] J. Liang, et al., "Peeking into the Future: Predicting Future Person Activities and Locations in Videos". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2960–2963, 16–17 June 2019.
[169] P. Zhang, et al., „SR-LSTM: State Refinement for LSTM Towards Pedestrian Trajectory Prediction". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12077–12086, 15–20 June 2019.
[170] D. Berenguer, et al., „Context-Aware Human Trajectories Prediction via Latent Variational Model". IEEE Trans. Circuits Syst. Video Technol. 31, 1876–1889, 2021.
[171] J. Oxley, et al., "Crossing roads safely: an experimental study of age differences in gap selection by pedestrians. Accident; analysis and prevention" 37, 962–971, 5 September 2005.
[172] Elhayek, A., de Aguiar, E., Jain, A., Thompson, J., Pishchulin, L., Andriluka, M. Bregler, et al., „Marconiconvnet-based marker-less motion capture in outdoor and indoor scenes". IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 501–514, 3 March 2017.