PN-III-P2-2.1-PED-2019-4995

PETRA

Detecția și urmărirea persoanelor pentru roboți sociali și mașini autonome

(People detection and **tra**cking for social robots and autonomous cars)

Etapa II - 2021

Raport științific și tehnic

Cuprins

1. Introducere 1
2. Problema predicției traiectoriilor persoanelor 2
3. Predicția traiectoriilor persoanelor prin transformări de perspectivă 3
   3.1. Proiectarea și implementarea arhitecturii 3
   3.2. Modulul de segmentare a imaginilor 4
   3.3. Modulul de transformare a perspectivei 5
   3.4. Modulul de predicție a traiectoriei persoanelor 7
   3.5. Evaluarea sistemului propus 10
4. Model generic de predicție a traiectoriilor persoanelor 15
   4.1 Abordare și arhitecturi utilizate 15
   4.2 Metrici de evaluare și rezultate 18
5. Concluzii 20

Bibliografie 20

1. Introducere

Detecția, urmărirea și recunoașterea persoanelor este o capacitate valoroasă pentru aplicațiile de vedere computerizată. Cu toate acestea, aceste sarcini sunt destul de dificil de realizat în mod autonom și, deși s-au obținut rezultate semnificative, acestea sunt încă o provocare tehnologică majoră. Urmărirea persoanelor, spre deosebire de alte sarcini de recunoaștere și interpretare, este dificilă atât din punctul de vedere al recunoașterii și predicției traiectoriei, cât și din cel al identificării instanțelor reale. Persoane parțial vizibile, reflexii în oglinzi sau ferestre și obiecte care seamănă foarte mult între ele, toate impun ambiguități intrinseci, astfel încât chiar și oamenii ar putea să nu fie de acord cu o aceeași soluție. Mai mult, stabilirea unor metrici de evaluare cu parametri liberi și definiții ambigue (cum se întâlnește des în literatură) duc adesea la rezultate cantitative contradictorii.

Obiectivul principal al proiectului PETRA este dezvoltarea unei platforme software pentru dezvoltarea de aplicații care necesită detecția și urmărirea persoanelor în medii reale. Scopul este acela de a proiecta și implementa o platformă și un set de algoritmi asociați care să poată fi ușor utilizați și integrați în aplicații care necesită atât detecția și urmărirea persoanelor de către roboții sociali în spații închise, cât și efectuarea aceleiași sarcini în spații deschise, în particular detecția și urmărirea pietonilor.

În cadrul primei etape a proiectului s-au analizat diferite arhitecturi de rețele neurale adânci pentru detecția de persoane și pentru urmărirea de persoane, atât pentru spații închise (indoor) – caz potrivit pentru aplicații ale roboților sociali, cât și spații deschise (outdoor) – caz potrivit pentru pietoni în aplicații de conducere autonomă. S-au comparat și evaluat aceste arhitecturi și s-au propus arhitecturi pentru detecția și urmărirea persoanelor. S-au investigat diferite seturi de date existente în domeniul public, s-au comparat aceste seturi de date din punct de vedere al performanțelor realizate pe diferite arhitecturi adânci și s-a colectat propriul set de date pentru detecția pietonilor și a mașinilor.

Obiectivul Etapei II (2021) a proiectului a fost acela de a realiza soluții noi pentru predicția traiectoriei persoanelor în scopul urmăririi mișcării acestora și utilă pentru a putea estima unde se vor duce persoanele în perioada următoare, ce vor face, și eventual a lua diverse decizii în cazurile care impun efectuarea unor acțiuni (de exemplu în cazul pietonilor dintr-un scenariu de conducere autonomă, ce decizii trebuie să ia șoferul automat). Pentru realizarea acestui obiectiv am propus, implementat și evaluat *două abordări*: (i) o metodă de extragere a informațiilor globale ale scenei din imaginile de vizualizare 2D de la nivelul liniei orizontului, prin combinarea unei transformări de perspectivă de la vedere la nivelul ochiului (*eye-level view*) la vedere la nivel de pasăre (*bird-level view*) , cu o tehnică de segmentare care determină parametrii pentru transformare, și (ii) un model generic de predicție a secvențelor video, din care deducem traiectoria. În ambele metode, au fost incluse metode de detecție a persoanelor din secvențele video utilizate.

2. Problema predicției traiectoriilor persoanelor

Urmărirea persoanelor este un subiect amplu studiat de către cercetători în prezent, deoarece are o aplicabilitate largă. Odată cu dezvoltarea dispozitivelor autonome, rezultatele precise sunt imperative în realizarea unui obiectiv specific sau prevenirea evenimentelor nedorite. Fie că vorbim de vehicule autonome, roboți, supraveghere sau case inteligente, a putea urmări și recunoaște aceeași persoană printr-o secvență de imagini, precum și a distinge între mai mulți indivizi este o componentă importantă. Predicția traiectoriei oamenilor este una dintre metodele introduse ca soluție la problema de urmărire.

Urmărirea oamenilor și predicția traiectoriei sunt tehnici care încearcă să prezică mișcările umane pentru a putea reidentifica aceiași indivizi și pentru a prezice comportamentul lor imediat următor. Predicția traiectoriei oamenilor este capacitatea de a analiza comportamentul unei persoane și de a deduce mișcările viitoare pe baza observațiilor. Deși poate părea simplu, modelarea procesului complex din spatele comportamentului uman este un lucru complex.

Oamenii au capacitatea naturală de a interpreta mișcarea, de a planifica traiectorii și de a evita coliziunile pe baza percepției vizuale a împrejurimilor. Un dispozitiv autonom care trebuie să traverseze un mediu trebuie să analizeze fiecare persoană și agent din jurul lui, să deducă traiectorii și să prezică căi acceptabile din punct de vedere social pentru a nu încurca sau bloca ceilalți participanți. Predicția traiectoriei reprezintă estimarea pozițiilor viitoare ale unei persoane pe baza observațiilor din trecut, care pot fi reprezentate prin coordonate în imagini sau coordonate în spațiu. Un sistem simplu de predicție a traiectoriei folosește o rețea neuronală recurentă aplicată pe pozițiile anterioare, în timp ce tehnici mai inovatoare iau în considerare

factori suplimentari. Întrucât traiectoria unei persoane este influențată de mișcarea celorlalți oameni din cartier, de obstacolele din mediu și de punctele de interes din scenă, tehnicile noi încorporează influențele sociale dintre oameni și informații despre dispunerea scenei împreună cu pozițiile persoanei urmărite.

Dacă inițial metodele legate de predicția traiectoriei pietonilor erau bazate pe diverse tehnici matematice, pe măsură ce tehnologia a evoluat și au apărut rețelele neurale, majoritatea sistemelor de predicție a traiectoriei au început să folosească cel puțin metode mixte, care combină rețele neurale cu diverse metode matematice (modele Markov, modele ce folosesc mixturi Gaussiene, etc), ulterior s-a trecut la folosirea predominantă a tehnicilor legate de rețele neurale.

Pentru predicția traiectoriei, metodele inițiale foloseau modele Markov sau mixturi Gaussiene. Algoritmii moderni sunt bazați, în general, pe rețele recurente de tip LSTM cu diverse îmbunătățiri. Există și modele de tip encoder-decoder folosite pentru această problemă. Deși modelele recurente sunt cele mai folosite pentru predicția pietonilor, metode recente folosesc rețele generative (GAN), dar și rețele de tip VAE (variational autoencoder). Ambele rețele sunt, în general, folosite cu succes pentru generarea unor imagini noi, inclusiv pentru predicția de frame-uri în video-uri.

De asemenea, este utilă studierea și a altor rețele care fac predicție de traiectorie, cum ar fi cele care au fost construite pentru detecția mașinilor, metodele putând fi ușor adaptate pentru detecția pietonilor. Există diverse modele specializate pentru predicția traiectoriei pietonilor ce folosesc rețele de tip LSTM [1-3], de asemenea rețele ce folosesc alte unități recurente precum GRU (Gated Recurrent Units) specializate pentru predicția poziției pietonilor [4], dar și metode ce folosesc rețele de tip encoder-decoder [5], inclusiv pentru detecția pietonilor care traversează strada [6].Unul dintre cele mai citate articole pentru detecția traiectoriei pietonilor este [7], ce folosește un model de tip generativ, dar și o tehnică ce ia în considerare doar traiectorii acceptate din punct de vedere social – Social GAN (de exemplu se ignoră traiectorii care pot lovi alte persoane). Această metodă, cea a folosirii elementelor sociale, a fost ulterior des aplicată în predicția traiectoriilor. O altă metodă interesantă [11] ce folosește rețele de tip generativ analizează interacțiunea dintre pietoni și mașini.

3. Predicția traiectoriilor persoanelor prin transformări de perspectivă

Pentru a include atât influența socială, cât și informațiile despre scenă, am implementat un sistem care prezice traiectorii oamenilor incluzând date vizuale. Sistemul conține o metodă de extragere a informațiilor globale ale scenei din imaginile de vizualizare 2D de la nivelul liniei orizontului. Metoda combină o transformare în perspectivă matematică de la vedere la nivelul ochiului (*eye-level view*) la vedere la nivel de pasăre (*bird-level view*) , cu o tehnică de segmentare care determină parametrii pentru transformare. Folosim o rețea generativă adversială care conține o componentă socială pentru a integra date despre poziția persoanei în scenă, poziția în imagine și schema obstacolelor din scena cu scopul de a îmbunătăți rezultatele. Evaluăm sistemul pe mai multe seturi de date luând în considerare mai multe metrici folosite în sistemele de predicție a traiectoriei: metricile ADE și FDE aplicate atât pe distanțe între pixeli, cât și pe distanțe în spațiul scenei.

3.1. Proiectarea și implementarea arhitecturii

O metodă eficientă pentru a realiza urmărirea persoanelor în imagini este predicția traiectoriilor persoanelor. Dacă știm poziția curentă a unei persoane, știm conform predicției unde ar trebui să se afle persoana la pasul următor și avem detecția unei persoane în aceea poziție, atunci putem spune cu o probabilitate destul de mare că persoana detectată este aceeași cu cea

pentru care avem predicția. Predicția traiectoriilor este un subiect complex, care are nevoie de mai multe tipuri de date pentru a avea o precizie bună. Este bine cunoscut faptul că modul în care se mișcă un om este direct influențat de obstacolele pe care le are în față și oamenii pe lângă care trece. În arhitectura definită mai jos, sistemul de predicție a traiectoriilor înglobează informații despre mișcarea individului pentru care facem predicția, modul în care arată obstacolele în mediu și pozițiile celorlalți participanți.

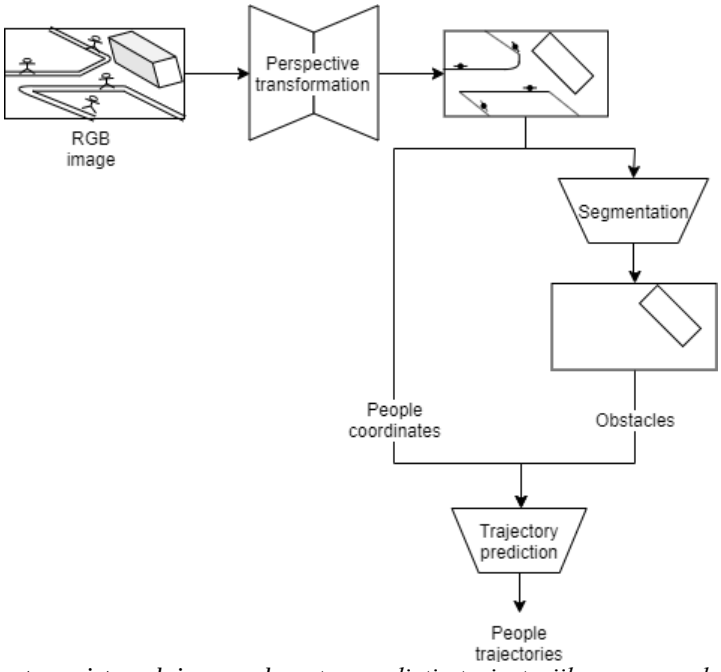

Fig. 1. Arhitectura sistemului general pentru predicția traiectoriilor persoanelor în imagini.

Aceasta arhitectură are 3 componente principale:

- modulul de segmentare a imaginilor
- modulul de transformare a perspectivei,
- modulul de predicție a traiectoriilor.

Pentru fiecare modul am detaliat modalitatea de implementare.

3.2. Modulul de segmentare a imaginilor

Imaginile din seturile de date folosite sunt imagini color preluate din perspective diferite și medii diferite. Pentru a avea o interpretare semantică a datelor redate în imagini, sistemul introduce o componentă de segmentare care să ofere o interpretare abstractă a informațiilor.

Componenta de segmentare semantică a imaginilor extrage informații despre obiectele și clasele din imagini. Această componentă are două atribuții specifice, respectiv segmentarea regiunilor și clasificarea obiectelor. Capacitatea de a segmenta regiuni în imagini presupune identificarea pixelilor care sunt asociați într-o imagine și care denotă același obiect. Capacitatea de a clasifica obiectele din imagini presupune atribuirea unor etichete fiecărei regiuni determinate prin segmentarea regiunilor pentru a indica clasa obiectului. Imaginea de mai jos este o ilustrare a unei segmentări semantice aplicate asupra unei imagini din setul de date, în care fiecare culoare denotă un tip de obiect diferit sau o parte a peisajului.

În arhitectura introdusă componenta de segmentare semantică are două obiective. Unul constă în scopul principal al sistemului, extragerea obiectelor din imagini RGB, în timp ce al doilea este reprezentat de calculul parametrilor pentru transformarea perspectivei. Atât modul de reprezentare al scenei, cu obstacole și zone posibile de parcurs, cât și punctul de referință al

imaginii care este cerut pentru calculul transformării perspectivei imaginii este extras folosind datele procesate de sistemul de segmentare.

Fig. 2. Segmentare semantică aplicată asupra unei imagini color. Imaginea din stânga este imaginea de referință, iar imaginea din dreapta reprezintă segmentarea asociată. Fiecare culoare din imaginea din dreapta reprezintă o clasa predefinită (ex.: vișiniu reprezintă persoană, verde copaci). O regiune este reprezentată de o zonă din imagine în care există pixeli de aceeași culoare conectați.

În cadrul sistemului am integrat o rețea neuronală pentru a determina toate clasele de obiecte din imagine. Rețeaua de segmentare folosită este *ResNet50* [8] combinată cu modulul *Pyramid Pooling* [9]. Această rețea oferă posibilitatea de a calcula masca de segmentare a imaginii. Rețeaua a fost antrenată pe setul de date de analiză a scenei *MIT ADE20K* [10,11], deoarece include o varietate de clase, clase de obiecte care pot fi găsite atât în interior, cât și în exterior. Varietatea claselor a fost un punct important în alegerea rețelei și a setului de antrenare, deoarece un sistem de urmărire a persoanelor, cum ar fi un robot de asistență de interior sau o mașină autonomă, trebuie să genereze rezultate precise în toate condițiile.

3.3. Modulul de transformare a perspectivei

Acest modul acre ca scop transformare imaginilor din perspectiva liniei orizontului în perspectivă de ansamblu. Obiectivul principal al metodei propuse pentru transformarea imaginilor este de a prelua imagini color, din unghiuri care simulează vederea de la nivelul ochilor unui om (eye-view), și de a prezice și înțelege aspectul global al mediului, practic simularea unei viziuni de sus a mediului (bird-view). Imaginile sunt convertite în date care simulează vederea de sus folosind o transformare a perspectivei și apoi sunt trecute printr-o tehnică de segmentare pentru a obține informațiile despre scenă.

Transformarea perspectivei imaginii este un algoritm matematic care modifică virtual unghiul de vizualizare al unei imagini. Simulează o schimbare a unghiului camerei folosind o rearanjare a pixelilor. Fiecare pixel din imaginea originală este poziționat la noi coordonate în imaginea rezultată folosind o matrice de omografie. Matricea de omografie determină unghiul transformării perspectivei. Valorile matricei de omografie sunt calculate pe baza punctului de întâlnire la infinit (vanishing point). O exemplificare vizuală a conceptului este prezentată în imaginea de mai jos.

Punctul de întâlnire la infinit dintr-o imagine este punctul în care liniile paralele converg atunci când sunt văzute în perspectivă. De obicei, punctul de întâlnire la infinit este calculat pe baza liniilor detectate în imagine, cum ar fi benzile de drum sau colțurile de perete, dar aceste tipuri de caracteristici nu sunt întotdeauna disponibile în imagini. În funcție de mediul în care este poziționată camera, pot exista imagini care nu oferă o structură clară a scenei. Pentru a depăși această problemă, sistem folosește tehnica de segmentare definită mai sus pentru a identifica un punct, pe care îl considerăm punct de referință necesar transformării perspectivei, în locul punctului de întâlnire la infinit.

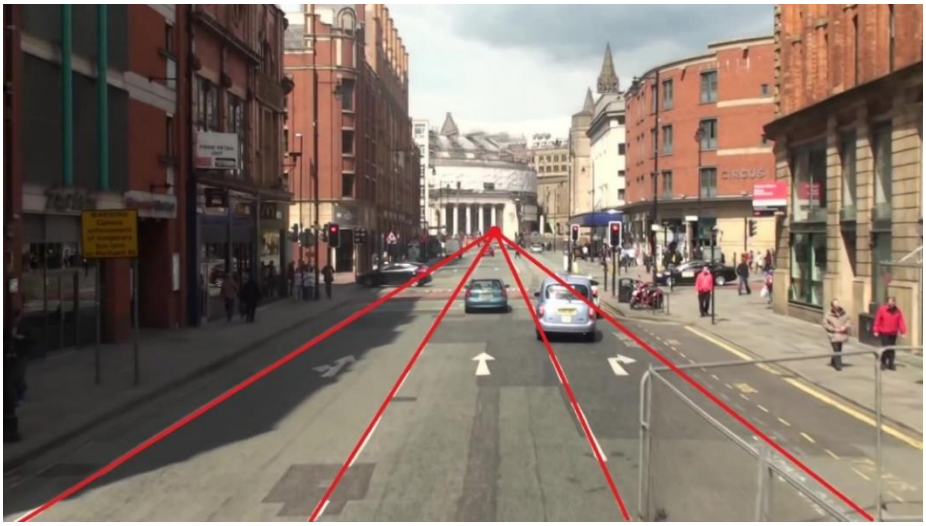

Fig. 3. Punctul de întâlnire la infinit a liniilor paralele văzute într-o imagine din perspectiva liniei orizontului (eye-level view).

Punctul de referință din imagine pe baza căruia se realizează transformarea de perspectivă a imaginilor este punctul care determină nivelul cel mai de sus al solului. Există un număr fix de clase posibile pentru nivelul solului care sunt analizate pentru găsirea punctului de referință. În setul de date folosit pentru antrenarea modelului de segmentare există mai multe clase care definesc solul: șosea, trotuar, pământ, etc.. Pe baza rezultatului tehnicii de segmentare semantică, nivelul solului poate fi diferențiat de restul scenei, indiferent de tipul de clasă identificat al terenului. Căutarea punctului de referință depinde de unghiul transformării perspectivei dorite. Întrucât scopul metodei propuse este de a oferi o vedere generală a scenei din fața camerei, transformarea în perspectivă pe care o realizăm este simetrică față de centrul imaginii. În consecință, căutarea punctului de referință se face de-a lungul liniei mediane. Căutarea se face folosind o ”fereastră” (kernel) 2D care se aplică de-a lungul liniei mediane pentru o mai bună precizie a punctului de referință.

Matricea de omografie necesară pentru transformarea perspectivei unei imagini este calculată pe baza a 8 puncte: 4 calculate relativ la punctul de referință estimat din imaginea originală și 4 puncte determinate în noua imagine care reprezintă poziția estimată a primelor puncte. Deoarece avem nevoie de o transformare centrală simetrică în perspectivă, primele 4 puncte au fost alese de-a lungul liniilor determinate de punctul de referință și punctele centrale-jos din partea stângă și dreaptă a imaginii. Poziția precisă este determinată de intersecția dintre liniile menționate anterior și două linii orizontale plasate sub punctul de referință. Pentru a include cei mai apropiați pixeli de cameră în noua imagine, una dintre liniile orizontale ar trebui să fie în partea de jos a imaginii, în timp ce a doua ar trebui să fie cât mai aproape de punctul de referință pe cât de multe date trebuie să fie incluse în noua imagine. Modalitatea de calculul a celor 4 puncte din imaginea originală este exemplificat în figura de mai jos.

Celelalte 4 puncte necesare matricei de omografie sunt de fapt o translație aplicată punctelor identificate în imaginea originală, în noua imagine transformată . Translația generează un dreptunghi situat în centrul de jos al noii imagini. Mărimea dreptunghiului este determinată de punctele de sus din imaginea originală. Suprafața imaginii originale care o să apară în imaginea transformată, depinde de dimensiunea dreptunghiului. Pentru a permite mai multor pixeli și implicit mai multor informații din imaginea originală să apară în noua imagine, se poate aplica o reducere a dimensiunii dreptunghiului. Procesul de transformare a perspectivei unei imagini poate fi văzut în figura de mai jos.

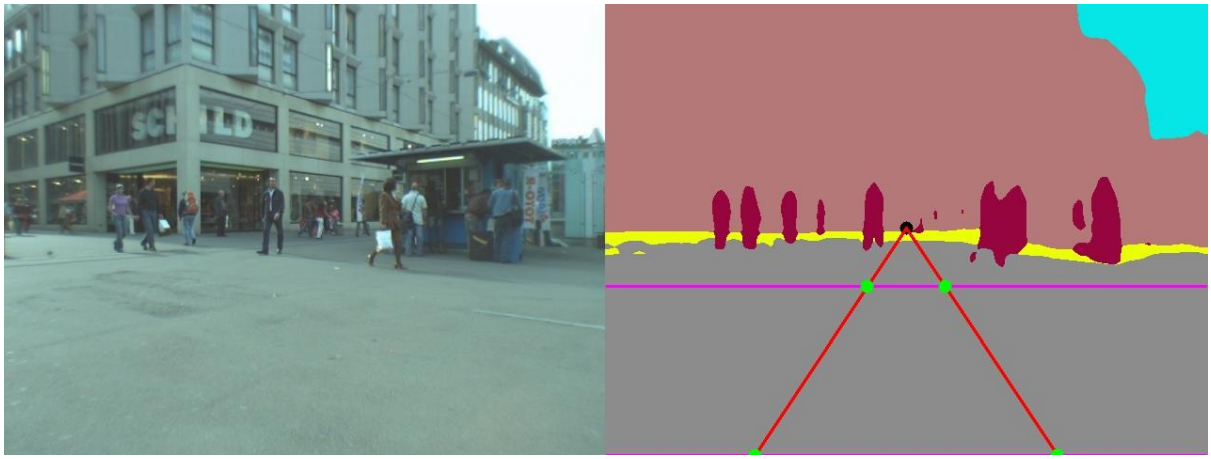

Fig. 4. Punctele estimate în imaginea originală. Punctul negru reprezintă punctul de referință, în timp ce punctele verzi reprezintă primele 4 puncte necesare pentru matricea de omografie.

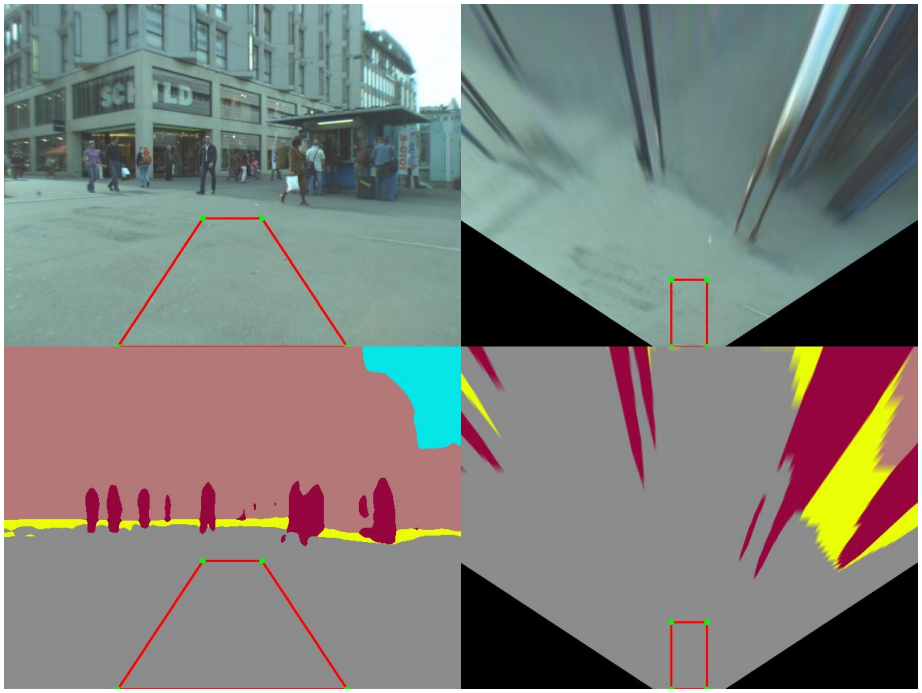

Fig. 5. Transformarea perspectivei unei imagini în perspectivă de ansamblu. Partea de sus prezintă transformarea imaginii originale, iar cea de jos a imaginii segmentate.

3.4. Modulul de predicție a traiectoriei persoanelor

Modulul pentru predicția traiectoriei este partea cea mai complexă a arhitecturii propuse. Modelul propus generează predicția traiectoriei pe baza unei combinații între coordonatele reale ale unei persoane și scena în care au fost înregistrate coordonatele. Coordonatele următoare ale unei persoane sunt generate pe baza coordonatelor observate anterior.

Traiectoria unei persoane constă într-o succesiune de tupluri, formate din coordonate din lumea reală și imaginea corespunzătoare, în raport cu timpul.

$$x = (coord_x, coord_y, image)$$

Formularea problemei din spatele modelului de predicției a traiectoriei este reprezentată de capacitatea unui sistem de a prezice traiectoria imediat următoare a unei persoane, X_{pred} pe baza traiectoriei observate a unei persoane X_{obs}. X_{obs} reprezintă traiectoria observată a unei persoane, de la momentul de timp t_1 la t_o, unde *o* reprezintă lungimea datelor observate.

$$X_{obs} = [x_1, x_2, \dots, x_o]$$

X_{pred} reprezintă succesiunea de poziții ale persoanei de-a lungul traiectoriei prezise, care continuă imediat după traiectoria observată, de la t_1 la t_p, unde p este lungimea predicției.

$$X_{pred} = [x_{o+1}, x_{o+2}, \dots, x_{o+p}]$$

Întrucât complexitatea acestei componente este una mai mare, am definit o arhitectură special pentru această componentă. Arhitectura este compusă în principal din două module: un modul pentru înțelegerea scenei și un modul pentru generarea traiectoriei prezise. Structura arhitecturii este prezentată în figura de mai jos.

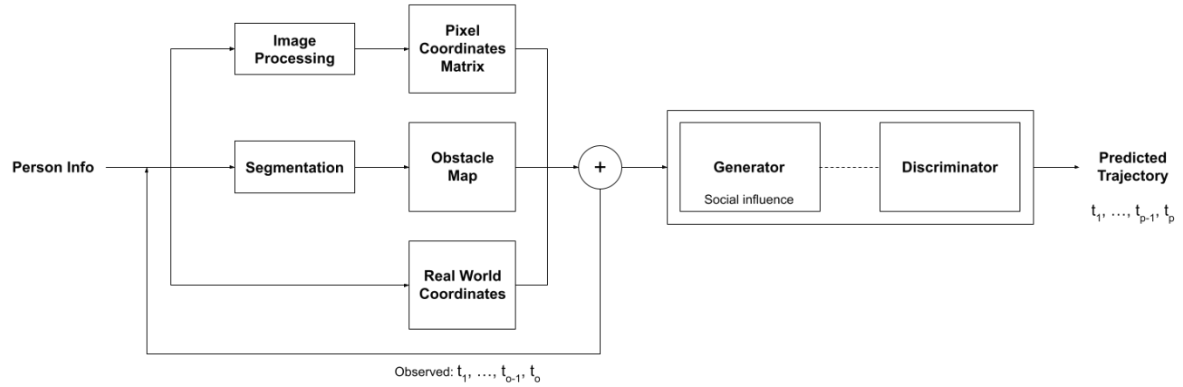

Fig. 6. Arhitectura componentei de predicție a traiectoriilor din sistemul general.

Informația asociată unei persoane care este transmisă componentei de predicție a traiectoriilor este compusă dintr-o secvență de coordonate din lumea reală reprezentând poziția persoanei la fiecare pas de timp, alături de imaginile RGB ale scenei dobândite la acel moment exact. Fiecare imagine prezintă un unghi de vedere de sus al scenei pentru o mai bună înțelegere a obstacolelor din mediu.

Pe baza informațiilor asociate cu o persoană, sistemul calculează două informații suplimentare: coordonatele pixelilor persoanei și harta obstacolelor din mediu. Folosind aceste informații putem calcula o matrice a coordonatelor pixelilor și o hartă a obstacolelor.

Matricea coordonatelor pixelilor este o imagine care reprezintă vizual poziția persoanei urmărite. Pentru fiecare pereche de coordonate din lumea reală ale unei persoane se deduce poziția imaginii în funcție de poziția pixelilor. Pentru fiecare set de date pe care îl folosim în sistemul nostru care nu include coordonatele pixelilor, aplicăm o procesare prealabilă pentru a determina proporția și corelația dintre coordonatele reale și persoanele din imagine. Determinăm suprafața scenei care este vizibilă în imagine calculând lungimea în metri atât a lățimii imaginii, cât și a înălțimii. Calculul se bazează pe coordonatele reale ale persoanelor care apar la marginea imaginilor și pe dimensiunea imaginilor din setul de date. Acesta este un calcul unic care trebuie făcut doar o dată înainte de faza de învățare a metodei. Pe baza dimensiunii zonei din imagine, aplicăm o etapă de procesare a imaginii pentru a calcula coordonatele pixelilor persoanei dintr-o imagine.

Matricea coordonatelor pixelilor este o matrice pătratică, formată din valori de 0 și o singură valoarea de 1. Valoarea de 1 corespunde poziției persoanei în scenă. Motivația din spatele alegerii unei matrice pătratice este pentru a avea proprietatea de omogenitate între toate dimensiunile de imagini pe care sistemul le poate primi. Indiferent de dimensiunea imaginilor dintr-un set de date, calculăm o matrice de tip grilă pătrată. Împărțim imaginea originală într-un număr fix de pătrate (orizontal și vertical) și atribuim 1 pentru grila în care se află persoana și 0 în rest. Un exemplu pentru cum ar arăta matricele de coordonate ale pixelilor pentru o persoană este în figura de mai jos.

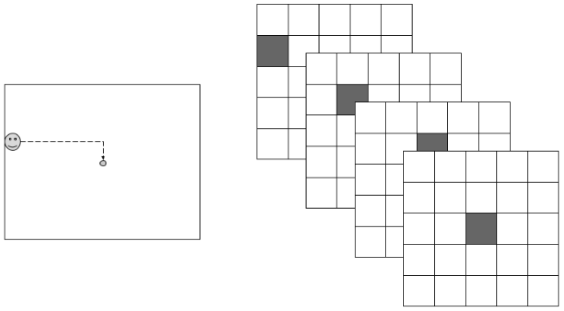

Fig. 7. Construirea matricelor de coordonate ale pixelilor pe baza traiectoriei unei persoane. Traiectoria persoanei din imaginea din stânga, care este arătată prin linia punctată, este reprezentată prin secvența de matrice din dreapta, unde fiecare fiecare pătrat negru reprezintă poziția în scenă a persoanei la un moment de timp.

Poziția persoanelor în imagine este integrată cu modul în care arată scena: dispunerea obstacolelor, dimensiunea lor și zonele pe care un pieton poate merge. Pentru a avea o înțelegere semantică a scenei, sistemul construiește o hartă a obstacolelor. Harta obstacolelor integrează forma scenei în ceea ce privește obstacolele și zonele posibile de mers. Pentru a genera harta obstacolelor ne folosim de rețeaua de segmentare descrisă anterior. Întrucât setul de date utilizat pentru antrenarea rețelei de segmentare are un număr mare de clase, am simplificat generarea măștii de segmentare grupând clasele în trei grupe principale: *obstacol*, *zonă de mers preferată* și *zonă de mers posibilă*. Rețeaua de segmentare va genera o mască cu segmentarea scenei pentru o imagine luând în considerare doar aceste trei clase. Pentru a potrivi dimensiunile rezultatului întors de rețeaua de segmentare cu matricea de coordonate ale pixelilor, am aplicat aceeași transformare a imaginii în matrice de tip grilă. Am împărțit masca de segmentare într-un număr fix de pătrate pe ambele axe, orizontal și vertical, și am atribuit fiecărei grile valoarea clasei cu cel mai mare număr de apariții în acel pătrat.

Datele procesate anterior sunt agregate într-un singur volum de informații și folosite mai departe în a doua parte a modulului. Pentru fiecare poziție observată, calculăm o matrice de coordonate a pixelilor și o hartă a obstacolelor pe care apoi le combinăm într-o singură intrare. Adăugăm aceste informații împreună cu coordonatele reale în scenă ale fiecărei poziții din traiectoria observată a persoanei și dăm aceste date ca intrare pentru modulul de generare a traiectoriei următoare.

Structura modulului de generare a traiectoriei următoare este compus dintr-o rețea generativă adversială [12], formată dintr-un *generator* și un *discriminator*. Generatorul prezice mai multe traiectorii posibile, iar discriminatorul penalizează traiectoriile greșite. În cadrul generatorului este integrat și un strat care permite schimbul de informații între persoanele din scenă. Fiecare persoana are asociat în cadrul generatorului o rețea recurentă, respectiv un *LSTM* [13]. După codificarea traiectoriilor fiecărei persoane din scena folosind această rețea recurentă, codificarea rezultată este trecută printr-un strat de conectare (*pooling*) care asigură că o persoană are informații despre persoanele din vecinătate. *Social-GAN*, respectiv modulul de predicție a traiectoriei următoare funcționează similar cu arhitectura prezentată în [7]. Principala diferență dintre [7] și sistemul nostru este faptul că sistemul nostru prezice atât coordonatele reale în scenă, cât și coordonatele pixelilor din imagine pentru o persoană. Funcția de penalizare (*loss*) utilizată pentru antrenarea rețelei este obținută prin combinarea diferențelor dintre coordonatele reale în scenă și coordonatele prezise, respectiv poziția reală a pixelilor din imagine și poziția pixelilor prezisă.

3.5. Evaluarea sistemului propus

Evaluarea sistemului a fost făcută atât pe seturi de date ce prezintă imagini din interior (*indoor*), cât și pe seturi de date ce conțin imagini din exterior (*outdoor*). În general seturile de date existente pentru astfel de probleme sunt făcute în exterior. În testarea sistemului, pe lângă seturile de date clasice de predicție a traiectoriei persoanelor, UCY **[14]** și ETH **[15]**, am considerat două seturi de date care conțin imagini din interior, respectiv MOT17 **[16]** și JRDB **[17]**. Ambele seturi prezintă o colecție de filmări din diferite locații, din diferite părți ale unui oraș, respectiv din campusul facultății Standford. Filmările prezintă imagini din ambele tipuri de scene, interior și exterior.

Evaluarea componentei de schimbare a perspectivei este făcută doar pe ultimele două seturi de date. Acest lucru se datorează faptului că ultimele două seturi sunt văzute de la nivelul unui robot autonom, în timp ce primele două conțin imagini preluate de sus. Componenta de schimbare de perspectivă se aplică doar atunci când dorim să obținem o imagine cu o privire de ansamblu, deși imaginea nu prezintă scena decât din perspectiva orizontului. Deoarece nu există un set de date dedicat pentru această problemă pentru a calcula o metrică de evaluare obiectivă pentru rezultatele metodei propuse, în această secțiune raportăm rezultatele vizual. Imaginile de mai jos prezintă rezultatele acestei componente pe cele două seturi de date. În fiecare din figurile de mai jos, imaginea din stânga reprezintă imaginea originală, iar cea din dreapta imaginea transformată și segmentată.

Evaluarea componentei de schimbarea a perspectivei imaginilor din exterior

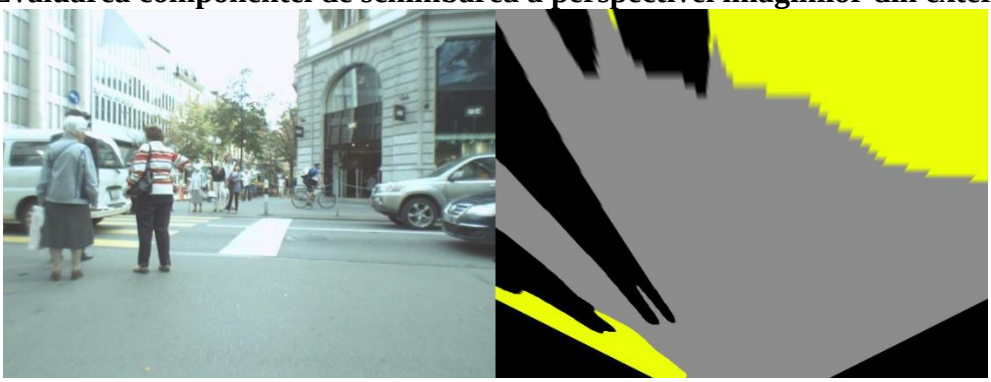

Fig. 8. Imagine din exterior din setul de date MOT17 [16]

Fig. 9. Imagine din exterior din setul de date MOT17 [16].

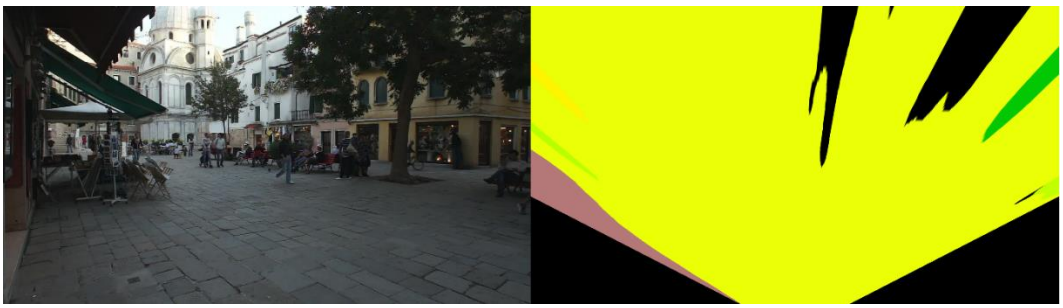

Fig. 10. Imagine din exterior din setul de date MOT17 [16].

Fig. 11. Imagine din exterior din setul de date JRDB [17].

Fig. 12. Imagine din exterior din setul de date JRDB [17].

Fig. 13. Imagine din exterior din setul de date JRDB [17].

Evaluarea componentei de schimbarea a perspectivei imaginilor din interior

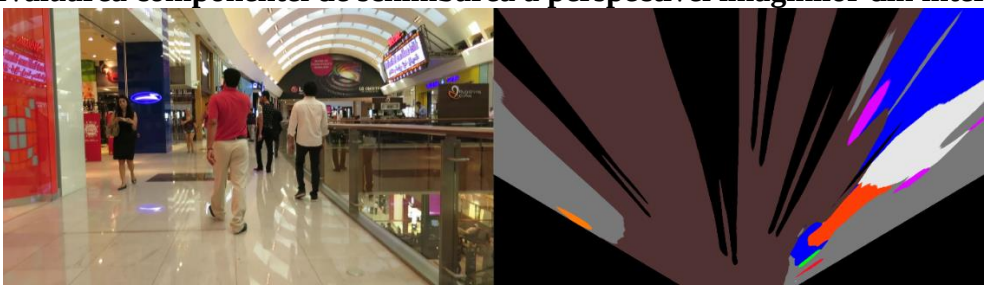

Fig. 14. Imagine din interior din setul de date MOT17 [16].

Fig. 15. Imagine din interior din setul de date MOT17 [16].

Fig. 16. Imagine din interior din setul de date JRDB [17].

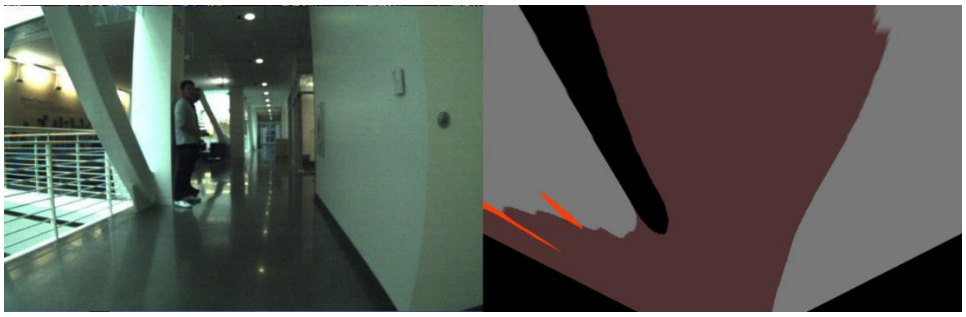

Fig. 17. Imagine din interior din setul de date JRDB [17].

Pentru a obține o evaluare calitativă și cantitativă a sistemului nostru de predicție a traiectoriei, am testat modelul de predicție a traiectoriei pe mai multe seturi de date. Experimentele noastre au fost efectuate pe seturile de date clasice pentru predicția traiectoriei UCY și ETH. Seturile de date conțin pozițiile scenei și imaginile asociate. Coordonatele imaginii oamenilor au fost deduse așa cum a fost descris în capitolul anterior. Pentru a obține o evaluare corectă, pentru testare am considerat aceleași două scenarii pentru predicția traiectoriei: observăm 8 poziții consecutive din traiectoria unei persoane și generăm predicția următoarelor poziții din traiectoria persoanei pentru următoarele 8 și respectiv 12 poziții.

Rezultatele experimentelor noastre sunt raportate în tabelele de mai jos. Primul tabel conține rezultatele predicțiilor pentru 8 poziții prezise, iar cel de-al doilea pentru 12 poziții. În ambele tabele am raportat erorile luând în considerare metricile ADE (Average Displacement Error) și FDE (Final Displacement Error), care reprezintă abaterea medie, respectiv abaterea finală de la traiectoria corectă. Valorile raportate în această evaluare conțin pe lângă erorile calculate relativ la coordonatele reale în scenă ale persoanelor, și erorile relativ la poziția în imagine. Valorile obținute au la bază valoarea distanței euclidiene calculată între poziția de referință și poziția prezisă.

Set de date	**Poziția în scenă**		**Poziția în imagine**	
	ADE (m)	**FDE (m)**	**ADE (pixeli)**	**FDE (pixeli)**
ZARA 1 **[14]**	0.72	1.24	17.56	35.69
ZARA 2 **[14]**	0.85	1.30	16.69	28.99
Students03 **[14]**	1.12	1.90	27.47	55.27
Hotel **[15]**	1.62	2.83	18.78	35.55
ETH **[15]**	2.55	4.42	33.69	59.49

Tabel 1. Valorile metricilor ADE și FDE obținute prin predicția a 8 poziții consecutive.

Set de date	**Poziția în scenă**		**Poziția în imagine**	
	ADE (m)	**FDE (m)**	**ADE (pixeli)**	**FDE (pixeli)**
ZARA 1 **[14]**	0.75	1.33	23.14	49.37
ZARA 2 **[14]**	0.87	1.57	20.66	43.73
Students03 **[14]**	1.54	2.80	49.32	99.68
Hotel **[15]**	2.01	3.55	23.35	48.17
ETH **[15]**	3.01	4.74	36.82	68.44

Tabel 2. Valorile metricilor ADE și FDE obținute prin predicția a 12 poziții consecutive.

Așa cum se poate observa, valorile obținute pentru predicția unei traiectorii de 8 poziții este mai precisă decât predicția unei traiectorii de 12 poziții. Pentru a avea o interpretare vizuală mai clară a modului în care funcționează sistemul de predicție a traiectoriei, am raportat imaginile de mai jos. În imagini se poate vedea omul urmărit, traiectoria reală pe care o are omul și traiectoria generată de sistemul propus. Fiecare om din imagine este urmărit separat, mai exact se generează câte o traiectorie pentru fiecare om din imagine la un anumit moment de timp. Figurile prezintă traiectoria unui singur om din imagine pentru o vizualizare mai clară a rezultatului. Punctele roșii reprezintă traiectoria observată pe baza cărora se generează rezultatul, punctele verzi traiectoria reală, iar cele albastre traiectoria prezisă.

Fig. 18. Rezultatul obținut pentru o persoană din setul de date UCY [14].

Traiectoria din figura 18 este foarte similară cu traiectoria reală a persoanei. Primele puncte prezise au o eroare aproape de 0, poziția finală fiind cea care dă eroarea traiectoriei. În cazul traiectoriei prezise omul își continuă direcția înainte, în timp ce în traiectoria reală omul coboară spre centrul trotuarului.

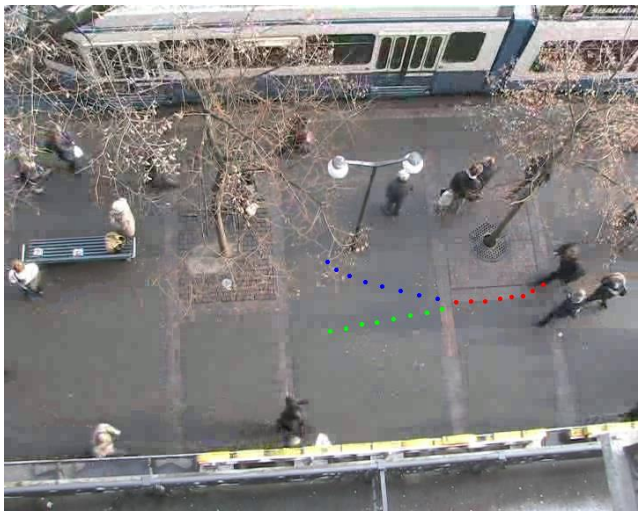

Fig. 19. Rezultatul obținut pentru o persoană din setul de date ETH [15].

În figura 19 traiectoria diverge față de valoarea corectă încă de la început. Deși acest lucru generează o eroare mare în evaluarea conform metricilor ADE și FDE, analizând rezultatul putem observa că traiectoria prezisă poate fi una validă. Conform traiectoriei prezise omul ocolește obstacolul (felinarul) și se îndreaptă către tramvai. Chiar dacă în cazul acestei persoane traiectoria prezisă nu corespunde cu traiectoria corectă, considerăm faptul că traiectoria generată nu este greșită, ci este influențată de obiectele detectate în imagine.

Fig. 20. Rezultatul obținut pentru o persoană din setul de date JRDB [11].

Figura 20 prezintă rezultatul obținut atunci când sistemul rulează pe imagini din interior. Omul urmărit în acest scenariu, omul de lângă ușă, merge spre ușa din dreapta. Traiectoria prezisă de sistemul nostru este foarte aproape de traiectoria finală, diferența făcându-se în termeni de viteză. Chiar dacă omul urmează traiectoria spre ușa din dreapta, el încetinește la urcatul scărilor, în timp ce sistemul nostru prezice că își va menține velocitatea.

Fig. 21. Rezultatul obținut pentru o persoană din setul de date ETH [9].

Un scenariu diferit pe care l-am raportat în figura 21 este cazul în care omul nu se mișcă de pe loc. În acest caz toate cele 3 tipuri de puncte, roșii, verzi și albastre sunt concentrate în poziția omului pe care l-am încercuit cu galben. Analizând acest exemplu și exemplele anterioare putem observa faptul că sistemul este capabil să distingă între diferitele comportamente ale oamenilor, prezicând corect atât cazurile în care persoanele se mișcă în scenă, cât și cazurile în care sunt statice.

Sistemul de predicție a traiectoriei încorporează informațiile primite de rețeaua de segmentare și modulul de schimbare a perspectivei în generarea rezultatelor. Metoda noastră propune un sistem de predicție a traiectoriei bazat pe imagini, care generează rezultatul atât pe baza imaginilor, cât și pe baza coordonatelor reale. Reprezentăm mișcarea indivizilor în scenă și configurația scenei ca imagini de tip grilă generate pe baza coordonatelor oamenilor ca pixeli

și pe baza rezultatului generat de rețeaua de segmentare. Aceste informații împreună cu coordonatele reale în scenă le dăm ca intrare pentru o rețea adversială generativă care conține un strat de pooling pentru a genera traiectoria prezisă. Acest strat de pooling înglobează influența socială a participanților în scenă în mișcarea unui individ.

Ne-am evaluat abordarea pe mai multe seturi de date, atât cu imagini din interior, cât și cu imagini din exterior, prin analiza valorilor obținute pe metricile ADE și FDE pe ambele tipuri de coordonate, coordonatele scenei în metri și coordonatele imaginii în pixeli. Pentru o vizualizare mai clară a rezultatelor am generat vizual traiectoriile obținute în diferite scenarii.

4. Model generic de predicție a traiectoriilor persoanelor

4.1 Abordare și arhitecturi utilizate

A doua abordare propusă și implementată este o metoda nouă, care nu a mai fost investigată în literatura de specialitate. Deoarece modelele bazate pe rețele neurale ce sunt gândite special pentru detecția traiectoriei pietonilor necesită foarte multe date cu care să fie antrenate (un element specific rețelelor neurale), dezvoltarea acestor rețele este strâns legată de seturile de date existente. Deși există câteva seturi de date specializate pentru detecția pietonilor, acestea nu sunt suficiente, conținând doar câteva scene din diferite orașe sau campusuri universitare.

Ideea de la care am plecat în realizarea acestui proiect este că putem folosi un model generic – acela de predicție de video, pentru a deduce traiectoria.

Modelele care fac predicție de video sunt diferite de cele care fac predicția traiectoriei, deoarece ele încearcă să prezică în întregime următoarele frame care vor exista în cadrul unui video, pornind de la imaginile din video deja existent. Deși această problemă este mult mai dificilă decât cea inițială, necesitând generarea unei întregi imagini, nu doar predicția traiectoriei câtorva pietoni, care pot fi ușor detectate folosind algoritmii de detecție existenți, care au acuratețe foarte bună, are avantajul că putem utiliza aceste viitoare imagini pentru a extrage pozițiile elementelor în mișcare din ele și, implicit, viitoarele traiectorii, putând folosi ca date de antrenare orice video existent pe internet ce conține pietoni.

Pentru modelul propus de noi, pe lângă modelul de bază, de generare de imagini, am folosit și o rețea de detecție a pietonilor, pentru a putea vedea ce pietoni există în imaginile din video, dar și în imaginile prezise de model.

În plus, pentru a oferi datelor noastre rezultate mai bune, am folosit încă o rețea de segmentare care să detecteze drumul. Detecția drumului este utilă pentru a putea ajusta datele legate de traiectoria pietonilor – se poate observa că în general un pieton tinde să își păstreze poziția relativă la un drum (mai puțin dacă vrea să îl traverseze), așa că putem ajusta poziția prezisă și în funcție de locația în care se află pietonul relativ la drum.

Pentru metoda originala propusă pentru detecția traiectoriilor, am pornit de la modelul din [18]. Modelul este unul dezvoltat pentru a prezice nu doar traiectoria pietonilor, cât și a altor agenți din trafic, cum ar fi mașini, bicicliști, etc, modelul fiind construit pentru mașini autonome.

Legat de predicția de video, există două tipuri de a face acest lucru. Unii algoritmi încearcă doar să construiască niște clipuri ce pot părea reale, pornind de la niște date existente, dar fără legatură cu niște date reale. Aceste modele fac generare de video, dar fără a încerca să prezică frame-uri ulterioare pentru niște clipuri reale. Pe de altă parte, există modele care au ca scop predicția unor frame-uri reale pentru niște clipuri analizate. Există inclusiv modele care abordează ambele probleme – generarea de frame-uri ce pot constitui un clip real dar și predicția de frame-uri pornind de la un clip deja existent. Deoarece această problemă nu este deloc una

simplă, multe arhitecturi s-au limitat doar la a genera niște clipuri cu o calitate scăzută, cu puțini pixeli și puține elemente (cum ar fi o singură persoană care se mișcă, se joacă golf) sau chiar ceva și mai simplu, generarea unor clipuri ce conțin doar niște cifre.

Cu toate acestea, pe măsură ce a evoluat cercetarea în acest domeniu, au început să apară arhitecturi care încearcă se prezică următoarele imagini din niște clipuri reale, inclusiv cu imagini din stradă, cu oameni și cu mașini. Și aici există o distincție dintre arhitecturi care prezic un singur frame în viitor și arhitecturi care prezic mai multe frame-uri, totuși se pot extinde ușor arhitecturile care prezic un singur frame, considerăm frame-ul nou generat ca făcând parte din cele reale și apoi se aplică din nou rețeaua asupra noului clip.

Problema predicției de video este una nouă și nu se poate aborda decât folosind rețele, fiind prea complexă pentru a putea fi modelată clasic, matematic. Deoarece problema este una dificilă, în general nu se folosește un singur tip de rețea, cum ar fi o rețea de tip LSTM, ci se preferă combinarea acestora. O combinație folosită este de rețele recurente de tip LSTM cu rețele convoluționale, în modele de tip CNN-LSTM [19]. Uneori, se folosesc și arhitecturi de tip encoder-decoder, cum ar fi în [20,21].

Pentru aplicația noastră am folosit 3 modele generative. Unul dintre ele, Prednet [22], folosește rețele de tip CNN-LSTM. Arhitectura acestuia se poate vedea în figura 22.

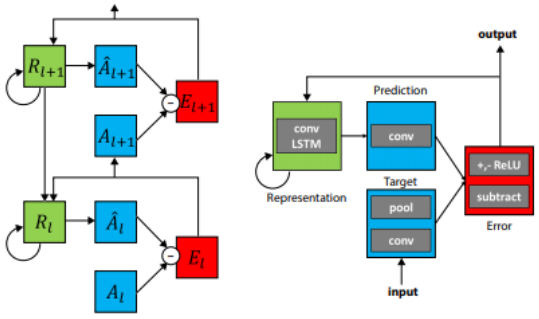

Fig. 22. Arhitectura Prednet [16]

Deși arhitecturile de tip CNN-LSTM sunt destul de utilizate, majoritatea arhitecturilor generative folosesc rețele dedicate pentru asta, cum ar fi GAN sau VAE. Un model foarte interesant ce folosește GAN și este folosit pentru a prezice acțiunile oamenilor este [24]. Față de rețelele GAN clasice, se folosesc 2 rețele de discriminare, una pentru a incoropora date legate de timp și alta pentru a incorpora date legate de spațiu. Modelul abordează atât problema generării, cât și a predicției de video. Celelalte două modele folosite de noi sunt Seg2Vid [24], o arhitectură ce se bazează pe rețele de tip VAE și segmentare, și a cărui arhitectură se poate vedea în figura 23, respectiv SAVP [25], un model ce folosește ambele rețele atât VAE, cât și GAN. Arhitectura lui se poate vedea în figura 24.

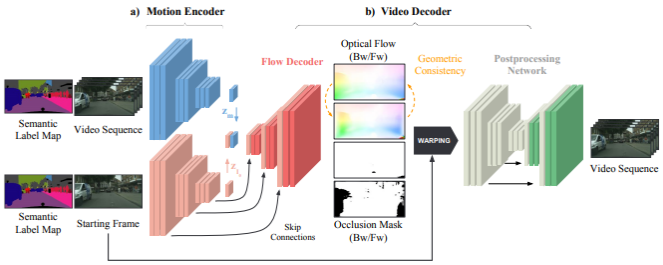

Fig. 23. Arhitectura Seg2Vid [18]

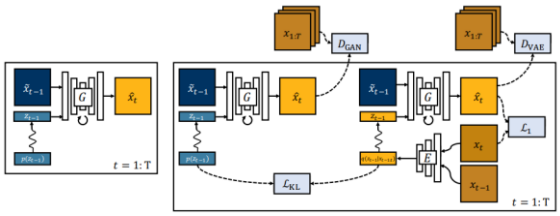

Fig. 24. Arhitectura SAVP [19]

Așa cum menționam în paragrafele precedente, o mare problemă pentru predicția de traiectorie este că trebuie să existe seturi de date diverse ce trebuie adnotate manual cu traiectoria, spre deosebire de un model ce face predicție de video, ce se poate antrena folosind orice clip existent pe internet ce conține oameni (plus o rețea care detectează acei oameni, rețelele existente având performanțe destul de bune).

Există, totuși, câteva seturi de date folosite pentru traiectoria pietonilor. Cele mai cunoscute sunt Trajnet [26], ce conține peste 3000 de traiectorii pentru pietoni, setul de date de la ETH [27], ce are 750 de traiectorii din două locații, DUT [28], un set de date asemănător cu cel folosit de noi, înregistrat într-un campus, ce conține 2000 de traiectorii. De asemenea există câteva seturi de date mai mari, cum ar fi setul celor de la Stanford [29], cu peste 20.000 de persoane filmate cu o dronă, sau setul celor de la WildTrack [30], ce conține atât traiectorii, cât și informații legate de urmărirea persoanelor (tracking). În general traiectoriile sunt dependente și de urmărirea persoanelor, pentru că în momentul în care discutăm de o traiectorie ne referim la traiectoria unei singure persoane, de aceea problemele legate de predicția traiectoriei presupun inclusiv o parte în care se face urmărirea unei persoane pe parcursul a mai multor cadre, ca să știm că este vorba de aceeași persoană.

După cum se poate observa, seturile de date existente conțin, în general, puține traiectorii, iar cele care conțin mai multe date nu sunt neapărat la fel de exacte, de aceea problemele legate de predicția traiectoriei suferă de lipsa datelor.

Acesta a fost motivul principal pentru care am decis să încercăm o metodă nouă – predicția de traiectorii folosind un sistem mai complex, ce folosește rețele pentru predicția de video, detecția pietonilor, segmentarea drumului, dar și estimarea distanței pentru testare. Arhitectura modelului nostru se poate vedea în figura 25.

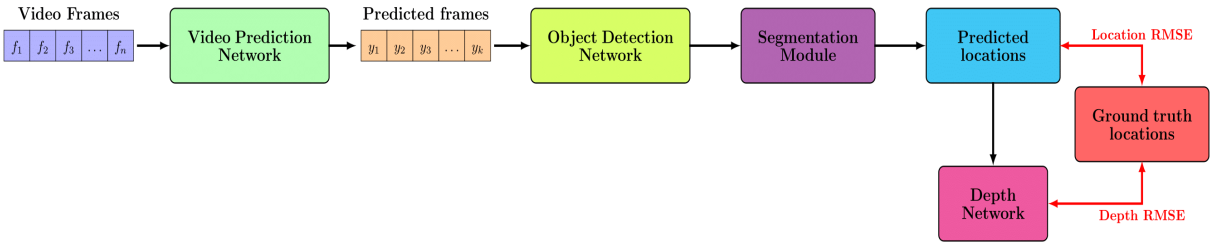

Fig. 25. Arhitectura propusă

În cadrul proiectului am colectat un set de date pentru conducerea autonomă, format din 13001 cadre, (set de date POLI). Imaginile conțin în principal mașini și pietoni, dar și câteva biciclete și indicatoare, un număr total de 60227 obiecte, din care 41064 mașini și 14576 persoane. Pentru generarea ground truth, am adnotat manual fișierul imagini folosind CVAT, un instrument pentru generarea ground truth. În setul nostru de date, peste 90% dintre obiecte sunt fie mașină,

fie persoană. Pentru experimentele noastre din cadrul acestei abordări am folosit diverse imagini din setul de date POLI, înregistrate în perioade mai aglomerate în care am putut înregistra multe persoane în cadrul unui clip.

4.2 Metrici de evaluare și rezultate

Pentru **validarea rezultatelor**, am calculat folosind metrica RMSE (abaterea pătratică medie), care are următoarea formulă, considerând dreptunghiurile care încadrează pietonii și având x_{i1p} coordonata x din stânga prezisă, x_{i1r} coordonata x din stânga reală, x_{i2p} coordonata x din dreapta prezisă, x_{i2r} coordonata x din dreapta reală, respectiv y_{i1p} coordonata y din stânga prezisă, y_{i1r} coordonata y din stânga reală, y_{i2p} coordonata y din dreapta prezisă, y_{i2r} coordonata y din dreapta reală, și N numărul de pietoni:

$$RMSE = \sqrt{\frac{\sum_i^N (x_{i1p}-x_{i1r})^2+(x_{i2p}-x_{i2r})^2+(y_{i1p}-y_{i1r})^2+(y_{i2p}-y_{i2r})^2}{N}} \quad (1)$$

În plus, pentru a avea **încă o metodă de evaluare**, am folosit inclusiv o rețea ce calculează adâncimea pixelilor, ce corespunde distanței față de camera de supraveghere. Deoarece abaterea medie pătratică legată de poziția prezisă furnizează detalii ce depind de dimensiunea imaginilor înregistrate și de coordonatele pixelilor prezișsi, o altă metodă bună de evaluare folosește calculul abaterii medie pătratice dintre diferența dintre distanța prezisă pentru un pieton și distanța reală față de acel pieton. Putem modela această eroare folosind următoarea formulă, unde N1 reprezintă numărul de pixeli ce formează imaginea pietonului prezis și N2 numărul de pixeli ce formeaza imaginea pietonului din imaginea reală:

$$RMSE_{depth} = \sqrt{\frac{\sum_j^N \left(\frac{\sum_i^{N1} ValoarePixelAdâncimePrezis_i}{N1} - \frac{\sum_i^{N2} ValoarePixelAdâncimeReal_i}{N2}\right)^2}{N}} \quad (2)$$

Pentru prezicerea distanței pixelilor din imagine, pentru modelul nostru, am folosit YOLO v4 [31] ca rețea de detecție a pietonilor, FCN [32] pentru detecția drumului și Monodepth [3] pentru estimarea distanței față de pietoni.

Am realizat diverse experimente, variind timpul zilei în care au fost filmate datele (pe timp de zi, la răsărit/ apus și noaptea) și analizând abaterea medie pătratică pentru locațiile prezise și pentru adâncimea prezisă în toate cele 3 cazuri, cât și media acestora. De asemenea, am făcut diverse experimente, variind între date obținute de rețele și date reale, astfel:

- utilizarea unei rețele de detecție a pietonilor comparată cu adnotarea manuală a acestora
- utilizarea unei rețele de detecție a drumului comparată cu adnotarea manuală a drumului
- folosirea rețelei de estimare a adâncimii pe datele prezise comparativ cu folosirea rețelei de segmentare pe datele reale.

Aceste date au fost utile în realizarea limitărilor modelului nostru, dar și pentru a analiza care ar fi o limită superioară legată de calitatea detecțiilor traiectoriilor, considerând că celelalte rețele ar funcționa în mod ideal (cu excepția rețelei principale, ce încearcă sa prezică viitoare cadre în cadrul unui video). Pentru experimentele noastre, am considerat clipuri de 35 de cadre, în care primele 30 au fost considerate știute și ultimele 5 au trebuit să fie prezise. De asemenea, am putut compara rezultatele obținute de noi cu cele ale modelului specializat pentru predicție de traiectorie. Din testele făcute de noi, am observat că rețelele generative strică destul de mult calitatea imaginii, astfel că pietonii sunt mai greu de recunoscut de rețelele de detecție în imaginile prezise, în special noaptea. Din acest punct de vedere, o relevanță mai mare au avut-o

rezultatele în care am folosit adnotarea manuală a pietonilor. Un aspect pozitiv a fost că rețeaua de detecție a drumului a adus aproape aceleași îmbunătățiri asupra modelului ca și adnotarea manuală. Îmbunătățirea dată de segmentare a fost chiar și de 30%. Între abaterea medie pătratică a adâncimii pe datele reale și pe cele prezise diferența a fost, de asemenea, foarte mică. Comparativ cu rezultatele obținute de rețeaua de predicție a traiectoriei, modelul nostru a fost undeva la 40% din calitatea acestuia, ceea ce era de așteptat, considerând că folosim un model pentru generare de video la bază. Cu toate acestea, pe viitor sperăm să putem îmbunătăți aceste rezultate. Câteva dintre rezultatele noastre legate de abaterea medie pătratică pot fi văzute în tabelul 1 (pentru locație) și tabelul 2 (pentru adâncime).

	Zi	**Apus**	**Noapte**	**Medie**
PredNet				
Fără segmentare	280.01	193.69	46.45	269.98
Cu segmentare	275.45	158.58	46.40	263.69
SAVP				
Fără segmentare	615.78	568.45	543.74	561.84
Cu segmentare	497.73	478.46	323.96	393.00
Seg2Vid				
Fără segmentare	320.88	278.37	103.58	310.92
Cu segmentare	306.36	196.98	102.15	289.27

Tabel 1. Abatere medie pătratică a locației (RMSE)

	Zi	**Apus**	**Noapte**	**Medie**
PredNet				
Fără segmentare	69.69	20.21	6.87	65.08
Cu segmentare	65.27	19.58	5.94	61.76
SAVP				
Fără segmentare	38.66	29.47	32.69	33.45
Cu segmentare	34.95	31.80	36.41	36.09
Seg2Vid				
Fără segmentare	62.51	25.72	20.26	56.57
Cu segmentare	64.10	25.91	19.60	58.75

Tabel 2. Abatere medie pătratică a adâncimii ($RMSE_{depth}$)

Un alt aspect interesant este că datele legate de segmentare au fost folosite pentru a micșora abaterea pătratică medie în ceea ce privește locația, nu și adâncimea, de aceea per total eroarea legată de adâncime crește un pic când folosim segmentare, cu toate acestea nu se depărtează

semnificativ de mult, ceea ce face modelul nostru unul sustenabil – segmentarea îmbunătățește semnificativ eroarea legată de locație și lasă aproape la fel eroarea legată de adâncime.
În general, cel mai bun model folosit a fost SAVP, atât pentru erorile legate de adâncime, cât și pentru cele legate de locație, cu excepția cazului când am folosit datele obținute de rețeaua YOLO, când cel mai bun model a fost Prednet. De asemenea, cele mai bune rezultate au fost obținute noaptea, pentru că au fost detectați puțini pietoni.

5. Concluzii

În cadrul Etapei II (2021) a proiectului am realziat soluții noi pentru predicția traiectoriei persoanelor în scopul urmăririi mișcării acestora. Am propus, implementat și evaluat *două abordări*: (i) o metodă de extragere a informațiilor globale ale scenei din imaginile de vizualizare 2D de la nivelul liniei orizontului, prin combinarea unei transformări de perspectivă de la vedere la nivelul ochiului (*eye-level view*) la vedere la nivel de pasăre (*bird-level view*) , cu o tehnică de segmentare care determină parametrii pentru transformare, și (ii) un model generic de predicție a secvențelor video, din care deducem traiectoria. În ambele metode, au realizat evaluări pe diferite seturi de date.
În Etapa III (2022) vom realiza integrarea soluțiilor în platformă, testarea extinsă, raportarea metodologiei testării și a performanțelor obținute.

Bibliografie

[1] H. Xue, D. Q. Huynh, and M. Reynolds. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In Applications of Computer Vision (WACV). 2018 IEEE Winter Conference on, pages 1186–1194. IEEE. 2018.

[2] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 12085–12094. 2019.

[3] Huynh Manh and Gita Alaghband. Scene-lstm: A model for human trajectory prediction. arXiv preprint arXiv:1808.04018. 2018.

[4] Martinez, J., Black, M. J., and Romero, J., "On human motion prediction using recurrent neural networks," in [Conference on Computer Vision and Pattern Recognition (CVPR)], 4674–4683, IEEE. 2017

[5] Wu J, Woo H, Tamura Y, Moro A, Massaroli S, Yamashita A and Asama H 2019 Pedestrian trajectory prediction using BiRNN encoder–decoder framework Adv. Robot. 2019

[6] P. Xue, J. Liu, S. Chen, Z. Zhou, Y. Huo and N. Zheng, "Crossing-Road Pedestrian Trajectory Prediction via Encoder-Decoder LSTM," 2019 IEEE Intelligent Transportation Systems Conference (ITSC). 2019.

[7] Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S. & Alahi, A. Social gan: Socially acceptable trajectories with generative adversarial networks, The IEEE Conference on Computer Vision and Pattern Recognition. 2018

[8] K. He et al. "Deep Residual Learning for Image Recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[9] Hengshuang Zhao et al. "Pyramid Scene Parsing Network", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] Bolei Zhou et al. "Scene Parsing through ADE20K Dataset", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] Bolei Zhou et al. "Semantic Understanding of Scenes through the ADE20KDataset", in *CoRRabs/1608.05442, arXiv:1608.05442*, 2016.

[12] Ian J. Goodfellow et. al. "Generative Adversarial Networks", in *arXiv: 1406.2661*, 2014.

[13] Sepp Hochreiter et. al. “Long Short-term Memory. Neural computation”, in *10.1162/neco.1997.9.8.1735*, pp. 1735-80, 1997.

[14] Alon Lerner et. al. “Crowds by example”. in *Computer graphics forum 26.3,* pp. 655–664, Sept. 2007.

[15] S. Pellegrini et al. “You’ll never walk alone: Modelling social behavior for multi-target tracking”, in *2009 IEEE 12th International Conference on Computer Vision*, pp. 261–268, Sept. 2009.

[16] A. Milan et al. “MOT16: A Benchmark for Multi-Object Tracking”, in *arXiv:1603.00831*, Mar. 2016.

[17] R. Martin-Martin et al. “JRDB: A dataset and benchmark for visual perception for navigation in human environments,” in *CoRR, vol. abs/1910.11792*, Oct. 2019.

[18] Chandra, R., Bhattacharya, U., Bera, A. & Manocha, D. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8483–8492). 2019

[19] M. Hosseini, A. S. Maida, M. Hosseini, and G. Raju, “Inception-inspired lstm for next-frame video prediction,” 2019.

[20] Harini Kannan Dumitru Erhan Quoc V. Le Honglak Lee Ruben Villegas, Arkanath Pathak. High fidelity video prediction with large stochastic recurrent neural networks. In NeurIPS. 2018

[21] Kalchbrenner, N., Oord, A. v. d., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A. & Kavukcuoglu, K. Video pixel networks. ArXiv:1610.00527. 2016

[22]Lotter, W., Kreiman, G. & Cox, D.Deep predictive coding networks for video prediction and unsupervised learning, arXiv:1605.08104 . 2016

[23] Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient video generation on complex datasets. arXiv preprint arXiv:1907.06571. 2019.

[24] Pan, J., Wang, C., Jia, X., Shao, J., Sheng, L., Yan, J. & Wang, X. (2019). Video generation from single semantic label map. ArXiv:1903.04480

[25] Lee, A.X., Zhang, R., Ebert, F., Abbeel, P., Finn, C. & Levine, S (2018). Stochastic adversarial video prediction. ArXiv:1804.01523

[26] S. Becker, R. Hug, W. Hubner, and M. Arens. An Evaluation of Trajectory Prediction Approaches and Notes on the TrajNet Benchmark. CoRR, abs/1805.07663. 2018.

[27] S. Pellegrini, A. Ess, and L. V. Gool, “Improving data association by joint modeling of pedestrian trajectories and groupings,” in ECCV. 2010.

[28] D. Yang, L. Li, K. Redmill, and U. Ozg ¨ uner, “Top-view Trajectories: ¨ A Pedestrian Dataset of Vehicle-Crowd Interaction from Controlled Experiments and Crowded Campus,” arXiv:1902.00487 [cs], Feb. 2019, arXiv: 1902.00487

[29] A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese,*Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes*in European Conference on Computer Vision (ECCV). 2016.

[30] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. M. Bagautdinov, L. Lettry, P. Fua, L. V. Gool, and F. Fleuret, “Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5030–5039. 2018.

[31] Bochkovskiy. A., Wang, C, & Liao, H.M. YOLOv4: Optimal speed and accuracy of object detection, arXiv:2004.10934. 2020

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR. 2015

[33] Godard, C., Mac Aodha, O., Brostow, G, Digging into self-supervised monocular depth estimation, arXiv preprint arXiv:1806.01260. 2018.