UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Scene compliant spatio-temporal multi-modal multi-agent long-term trajectory forecasting

Daniela Alves Ridel

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP SÃO CARLOS

Daniela Alves Ridel

Scene compliant spatio-temporal multi-modal multi-agent long-term trajectory forecasting

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
October 2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R544s

Ridel, Daniela Alves
Scene compliant spatio-temporal multi-modal multi-agent long-term trajectory forecasting / Daniela Alves Ridel; orientador Denis Fernando Wolf. -- São Carlos, 2021.
106 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2021.

1. Multimodal Trajectory Forecasting. 2. Convolutional Neural Networks. 3. Machine Learning. I. Wolf, Denis Fernando, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Daniela Alves Ridel

Predição multimodal de trajetórias de longo prazo de múltiplos tipos de agentes adaptável a cena

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutora em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
Outubro de 2021

Dedicated to my parents Rosinei and Ridel,
to my sister Denise (in memorian),
to my grandparents Durvalino and Zoraide, and
Hugo and Lucia (in memorian).

ACKNOWLEDGEMENTS

A Ph.D. is indeed a long journey, where either the journey and the arrival are important. The journey became smoother when you find great people on the path, I was lucky to find all those to whom I write here.

The first one, the one that made my life possible and has always been guiding me, thank you God for everything, I know without you nothing of this would be possible. Rosinei and Ridel thank you for providing a good environment where I was able to solely focus on studying. I acknowledge you for pushing me towards my best and for pointing to me the importance of studying and achieving good grades. I am grateful for you trusting in the girl you have raised, and for letting me pursue the world outside of my comfort zone.

Another thing that it takes to make a great journey is having the right guidance. Denis Wolf, thank you for accepting me into your group, guiding me through this doctorate, and providing a nice environment where creative ideas could emerge. I am grateful for all the discussions and the calmness in giving directions. Thank you for allowing us to use all lab types of equipment and platforms, that was very important for us to get hands-on experience when developing intelligence for self-driving cars.

Moving to a completely new city might be frightening. I am thankful to all my friends from ICMC-USP that were a family for me during all this time: Stevão Andrade, Lucas Tsusui, Raphael Rocha, Lucas Pagliosa, Lina Garcés, Valéria Carvalho, Jadson Castro, Adam Moreira, Rafael Mantovani, Rafael Montanari, Mariane Neiva, Sr. Orivaldo, and Brauner Oliveira. I am grateful for all discussions, demo preparation, and birthday parties I shared with the members of the Mobile Robotics Lab and the Laboratory of Critical Embedded Systems: Prof. Denis Wolf, Prof. Kalinka, Prof. Fernando Osório, Tiemi Nakamura, Patrick Shinzato, Luis Rosero, Caio Mendes, Tiago dos Santos, Iago Gomes, Carlos Massera, Ricardo Horita, Alberto Horita, Kelen Vivaldini, Lucas Nunes, Carlos Braile, Laercio, Alberto Hata, Diego Bruno, Francisco Alencar, Raphael Berri, Jefferson Souza, Victor Hugo Sillerico, Jean Amaro, Mariana Rodrigues, Isadora Ferrão, and Viviana Romero.

Stepping out of our comfort zone might be challenging. I am grateful for the Erasmus Mundus scholarship for the opportunity to develop research in Germany at the Institute of Measurement and Control Systems (MRT) Karlsruhe Institute of Technology (KIT), under supervision of Professor Christoph Stiller and Martin Lauer. I thank Eike Rehder and Jan-Hendrik Pauls for the discussions and support in broadly understanding the research history in pedestrian path prediction. I also would like to thank all my KIT friends for the coffee and lunch

break discussions, hikings, skiings, and segelfliegen flights: Annika Meyer, Ole Salscheider, Tilman Kühner, Jannik Quehl, Maximilian Naumann, Sahin Tas, Piotr Orzechowski, Julius Kümmerle, Christoph Burger, Fabian Poggenhans, and Sascha Wirges.

I am forever grateful to Fulbright for the best experience of my whole life. A special thanks to Taynara and Carol. It's indescribable the feeling of being awarded a scholarship that cares so much for us. I cannot place into words how much I have grown during my first year being a Fulbrighter. Thank you, Fulbright - Doctoral Dissertation Research Award for the opportunity to research at University of California, San Diego (UCSD), Laboratory for Intelligent and Safe Automobiles (LISA) in the United States of America, under supervision of Professor Mohan Trivedi. I am grateful for had crossed paths with Nachiket Deo and Akshay Rangesh, thank you for the great deep learning discussions. Much obliged Nasha Meoli, Bowen Zhang, and all LISA team members for the great conversations.

This paragraph is dedicated to all the amazing women in my life. My mother Rosinei, grandmothers Zoraide and Lucia, aunts (Marinês, Marina, Maura, and Meire), have always been a good source of inspiration throughout my life. The doctorate process is broader than only research, we do not only grow the Science, as Scientists, but we also grow as people, as humans. Professor Solange Resende made me think a lot about being our lives' main actors. I could realize a lot about life and our responsibilities while I was her Professor Assistant. Thank you, Professor Kalinka, for being so enthusiastic about empowering young girls through Computer Science, and for being an inspiration to all the Technovation Hackday and Summer School mentors and volunteers. Louise Poubel, thank you for the great job you do at Open Robotics and for being such a great inspiration for all the girls in Robotics. Muazma, thank you for being my mentor and for helping me to set my goals and working hard to reach them. I am grateful to Microsoft for pairing me with you. Michelle Paison and Rheina Agosa, thank you for taking such good care of all the women in computing awardees during our stay in Houston and Seattle. Thank you for the astonishing opportunity to cross paths with these amazing women that I am sure will be developing great things in the near future: Cassandra Oduola, Rebecca Houston, Veronica Lewis, Mónica Ceisel, Pooja Nagpal, Diana Torres, Shannen Bravo-Brown, Chineye Emeghara, and Elizabeth Lin.

Thank you to all Professors and employees from ICMC-USP for providing such a great environment that everyone wants to come back to. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

All journeys one day meet their destination, clearing the way to new paths that worth be sought, hence keeping human beings in their life-long search for the purpose of our own unique existence.

“Luck is where preparation meets opportunity.”
(Randy Pausch, The Last Lecture)

RESUMO

ALVES RIDEL, D. **Predição multimodal de trajetórias de longo prazo de múltiplos tipos de agentes adaptável a cena**. 2021. 106 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

A previsão de movimentação humana de longo prazo é uma tarefa desafiadora devido à não linearidade, multimodalidade e incerteza inerente nas trajetórias futuras. Esse tipo de previsão é importante para garantir a segurança no contexto de veículos autônomos, especialmente quando eles se deslocam dentro de centros urbanos onde ciclistas e pedestres podem ser vistos com mais frequência. Ao prever as trajetórias dos agentes ao seu redor, o veículo autônomo pode planejar rotas mais seguras e evitar possíveis colisões. Trabalhos prévios usaram diferentes tipos de informações de entrada, dependendo do tipo de agente (carros, pedestres ou ciclistas), a duração da trajetória prevista (longo ou curto prazo) e a quantidade de trajetórias previstas (unimodal ou multimodal). Trabalhos relacionados normalmente ou dependem de mapas de alta definição, ou processam a cena e as trajetórias como recursos desconexos, portanto, a inferência espacial do contexto nas trajetórias futuras é perdida. Nesta tese é proposta uma nova abordagem para a previsão de trajetórias que alinha as informações de entrada no espaço e no tempo usando o mesmo frame de referência centrado no agente. Alinhando essas informações conseguimos utilizar o poder das redes neurais convolucionais para computar os caminhos mais prováveis e forçar o modelo a compreender a cena. O modelo proposto aprende automaticamente o contexto da cena e prevê vários caminhos que são plausíveis de acordo com as informações de entrada. A abordagem proposta atingiu resultados competitivos quando comparado ao estado da arte no *Stanford Drone Dataset* (SDD) para predição de trajetórias de longo prazo, usando cinco trajetórias previstas. Para aplicações críticas, como carros autônomos, é importante prever várias trajetórias futuras possíveis para cada agente-alvo, pois assim é abrangido uma gama mais ampla de possíveis futuros, aumentando a segurança de veículos autônomos. Nesse sentido, a previsão de trajetórias é uma tarefa crucial a ser desenvolvida e incluída no pipeline de carros autônomos.

Palavras-chave: Predição multimodal de trajetórias, Redes neurais convolucionais, Aprendizado de máquina.

ABSTRACT

ALVES RIDEL, D. **Scene compliant spatio-temporal multi-modal multi-agent long-term trajectory forecasting**. 2021. 106 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

Predicting long-term human motion is challenging due to the non-linearity, multi-modality, and inherent uncertainty in future trajectories. Such type of prediction is important to ensure safety in the context of self-driving vehicles, especially when driving inside cities where vulnerable road agents, as cyclists and pedestrians, might be more commonly seen. By predicting the trajectories of surrounding agents, the self-driving car can plan safer routes and avoid possible collisions. Prior studies have used different types of input information depending on the type of agent (cars, pedestrians, or cyclists), the length of the predicted trajectory (long or short-term), and the number of predicted trajectories (unimodal or multimodal). Related work either rely on high-definition maps or processes scene and past trajectories as disconnected features, therefore the spatial inference of context in future trajectories is lost. We propose a new approach to trajectory forecasting that aligns the input information in space and time in an agent-centered manner. By aligning the input information we can take advantage of convolutional neural networks to compute the most plausible paths. Our model automatically learns and enforces scene context and therefore can predict multiple plausible paths according to the input information. The proposed approach achieved competitive results compared to the state-of-the-art in the Stanford Drone Dataset (SDD) for long-term trajectory forecasting, using five predicted trajectories. For critical applications, like self-driving cars, it is important to predict several possible future trajectories of each target agent, as it covers a broader range of possible futures, increasing self-driving car safety. Accordingly, the prediction of trajectories is a crucial task to be developed and included in the self-driving cars pipeline.

Keywords: Multimodal Trajectory Forecasting, Convolutional Neural Networks, Machine Learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbf{X}^{p}$ — Set of past trajectories data

$\mathbf{X}_{i}^{p}$ — A past trajectory (array)

$\mathbf{Y}_{i}^{f}$ — A ground truth future trajectory

$\hat{\mathbf{Y}}_{i}^{f}$ — Set of K predicted trajectories

$\hat{\mathbf{Y}}_{i,k}^{f}$ — k-predicted trajectory estimated by the model

τ^{p} — Past trajectory (frame of reference grid)

S_t — BEV RGB image (frame of reference grid)

O_t — Surrounding agents static location (frame of reference grid)

$\|x\|_2$ — L^2norm of x

$\mathbb{R}^2$ — denotes a plane

K — Number of predicted trajectories

N — Width and height of the image sample

T — t+t_f

t — Current time-step

t_p — Past time window length

t_f — Future time window length

CONTENTS

CHAPTER

1

INTRODUCTION

Road traffic injury is the eighth leading cause of death in the globe, and the primary leading cause of death among young adults (World Health Organization, 2018). Self-driving cars arise as a possible solution for such a problem, as they can be equipped with sensors, and execute algorithms providing a nondisruptive 360º of environmental awareness. Autonomous cars can also improve traffic flow and fuel usage, and provide mobility to impaired people. Alongside such technology arriving in the core of cities, several other challenges emerge. One of such challenges is how to provide safety for surrounding agents moving in the same scene as the ego-vehicle. An approach to ensure the safety of vulnerable road agents is to predict their future steps, therefore allowing self-driving cars to perform evasive maneuvres to avoid collisions.

As part inherent of humans' motion they are constantly adapting their paths regarding goals they want to reach, obstacles they want to avoid, and rules they are obligated to obey. When humans navigate in urban spaces, they might be walking, cycling, skating, or driving. These are just a few examples of types of transportation commonly used by humans. The type of transportation used by a person characterizes his/her pattern of motion. Therefore the person's trajectory is very correlated to the scene. This suggests that scene semantic information is an important cue when dealing with different patterns of human motion. Imagine a scenario where a pedestrian is walking straight, in a few meters ahead there is a wall. An algorithm that solely relies on dynamics would not be able to predict that the pedestrian will make a turn (CUI *et al.*, 2019).

When looking at humans and predicting their behaviors inside cities a common pipeline is first detecting them in 2D/3D images, then tracking them among consecutive images (video), by assigning a unique identifier, and then finally predicting their future behavior. The behavior prediction task was tackled in the literature in many forms, by classifying among many possible motion patterns (SCHNEIDER; GAVRILA, 2013; KOEHLER *et al.*, 2013; BONNIN *et al.*, 2014; VÖLZ *et al.*, 2015; HASHIMOTO *et al.*, 2015b; KWAK; KO; NAM, 2017) by predicting one future trajectory (QUINTERO *et al.*, 2015; GOLDHAMMER *et al.*, 2015; KOOIJ *et al.*,

2014; FERGUSON *et al.*, 2015; SCHULZ; STIEFELHAGEN, 2015a), or by predicting many possible trajectories (GUPTA *et al.*, 2018; SADEGHIAN *et al.*, 2019; AMIRIAN; HAYET; PETTRÉ, 2019; LEE *et al.*, 2017; DEO; TRIVEDI, 2019; CUI *et al.*, 2019; Zyner; Worrall; Nebot, 2019).

Studies that focus on predicting pedestrian behavior using classification, try to predict among a set of motion types as crossing, stopping, bending in, and starting. Alternately, the studies that focus on trajectory forecasting, estimate the exact locations the pedestrian will walk through in the future. This thesis focuses on the latter, but instead of estimating one possible future trajectory, we estimate many possible trajectories for each target agent, such a task is commonly represented in the literature as multi-modal trajectory forecasting. Our work also focuses on long-term trajectory and we do not restrict our approach to solely a type of agent (as pedestrians or cars). As each possible future trajectory draws a different contingency plan scenario, the trajectory prediction is an important task that allows self-driving cars to prepare themselves for emergency action to be taken, planning safer routes, without the need for a full stop.

Classification of motion types is usually associated with short-term predictions. Predicting pedestrian future paths can be associated with either short or long-term predictions. The time window (horizon) is an important factor while deciding the forecasting approach to be used. Studies that comprise long-term predictions usually draw information from static cameras and aim at predicting either the pedestrians' final destination or the path followed (KARASEV *et al.*, 2016; KITANI *et al.*, 2012; DEO; TRIVEDI, 2017), while studies on short-term focus more on the body and head orientation. Despite no consensus regarding the range of time for an approach to be considered short and long term, some authors say that short-term (KOOIJ *et al.*, 2014; SCHMIDT; FÄRBER, 2009; FÄRBER, 2016; BONNIN *et al.*, 2014) usually predict pedestrians' position up to the next 2.5 seconds.

The motion speed, behavior, and path preferences usually vary according to the different types of locomotion. Cars are usually bounded by lanes and their direction. Cyclists may exhibit a different behavior by not complying with traffic rules, like moving in the wrong way or entering a roundabout by the wrong side. Pedestrians are likely to walk on the sidewalk, they might shorten their paths by walking through the grass, they can cross the street in a diagonal. Pedestrians can decide to quickly change direction, which makes their long-term predictions a challenging task (FERGUSON *et al.*, 2015; GANDHI; TRIVEDI, 2008). All those differences create an extra challenge when forecasting different agents' trajectories in the same given model.

The agent's past positions are another meaningful piece of information, as they can help to understand the direction the person is moving towards. An agent's past trajectory can also restrict the space of probable future positions, as generally, a person does not return to a preceding position. A high probable path for one person may have a low probability to another just based on the direction both of them are walking towards.

Humans have a prior knowledge of the world that makes it easier for us to learn a new task as driving (LECUN, 2020). We understand the gravity and the consequences of driving outside the path on a cliff, we can simulate other people's behavior, we can project the future, think about the consequences of it, and act accordingly in the present to prevent damages. A model that comprises several factors that may correlate, in an ideal manner, with future trajectory prediction is still missing in the literature. Related work using Deep Learning (DL) also lacks in the ability to incorporate context cues and providing a qualitative explanation, and necessary ablations studies. The correspondence of scene features and future locations is a harder task to be achieved in a higher-dimensional space, and this might be the reason why current approaches lack diversity and compliance with the scene in their predicted trajectories.

1.1 Problem statement and objective

We assume we have access to a set of BEV Red Green Blue (RGB) images I. Such images I can be obtained through an Unmanned Aerial Vehicle (UAV), smart city infrastructure (e .g., camera in a traffic light or at the top of a building), or the projection of a camera mounted on an Autonomous Ground Vehicle (AGV). We assume we also have access to a robust tracker and detector that provide detections and track identification for all target agents in the scenes. From the 2D bounding boxes and identifications given by the detector and tracker module, we estimate the agent position at time t as the 2D bounding box center discrete position $[x,y] \in \mathbb{R}^2$. The set $\mathbf{X}^p$ contain all past trajectories $\mathbf{X}_i^p$,

$$\mathbf{X}^p = \{\mathbf{X_i}^p, \cdots, \mathbf{X_M}^p\}, \tag{1.1}$$

where M is the size of the dataset of trajectories and i is the trajectory id. Each trajectory $\mathbf{X}_i^p$ has a set of consecutive discrete coordinates $\mathbf{x} = [x,y]$ comprising the agent consecutive positions from time $t - t_p$ to time t:

$$\mathbf{X}_i^p = \left[\mathbf{x}^{t-t_p}, \cdots, \mathbf{x}^{t-\Delta_t}, \mathbf{x}^t\right], \tag{1.2}$$

where t is considered the last observed position of the trajectory i, t_p is the length of the past time window, and Δ_t is the interval between two consecutive observations.

The ground truth future trajectory $\mathbf{Y}_i^f$ of the agent i, ranges from $t + \Delta_t$ to $t + t_f$, where t_f is the length of the future time window:

$$\mathbf{Y}_i^f = \left[\mathbf{x}^{t+\Delta_t}, \cdots, \mathbf{x}^{t+t_f}\right]. \tag{1.3}$$

Unimodal approaches directly try to estimate $\mathbf{Y}_i^f$. Multi-modal approaches, instead, predict a set $\hat{\mathbf{Y}}_i^f$ of possible trajectories:

$$\hat{\mathbf{Y}}_i^f = \left\{ \hat{Y}_{i,1}^f, \cdots, \hat{Y}_{i,K}^f \right\}, \tag{1.4}$$

where K is the number of predicted trajectories, and

$$\hat{\mathbf{Y}}_{i,k}^f = \left[\mathbf{x}^{t+\Delta_t}, \mathbf{x}^{t+2\Delta_t}, \cdots, \mathbf{x}^{t+t_f} \right], \text{with } k \in \{1, 2, ..., K\}. \tag{1.5}$$

First, we pre-process the data in an agent-centric grid approach. From the image I, we center and crop the BEV image around the target agent, resulting in a $N \times N$ RGB image S_t. From the past trajectories $\mathbf{X}^p$ we generate τ^p and O_t. τ^p is the 3-D $N \times N \times t_p$ grid representing the target agent's past trajectory $\mathbf{X}_i^p$. O_t is the one-hot 2D $N \times N$ grid of surrounding agents discrete 2D locations. We use grids to estabilish a spatio-time compliance between scenes and trajectories. Hence, we take advantage of Convolutional Neural Network (CNN)s to learn from the data in a supervised learning manner. Given surrounding agents information O_t, scene data S_t, and target agent's past trajectory τ^p, our goal is generating $\hat{\mathbf{Y}}_i^f$. A direct mapping from the data to trajectories is a difficult task to be achieved directly in an end-to-end manner (LEE *et al.*, 2017; Schöller *et al.*, 2020). Therefore we split the problem into two modules named probability grid generation and trajectory generation. The first module takes as input O_t, S_t, and τ^p and generates grids G^f representing the probability of each grid cell be occupied, by the target agent, at each future time step. The second module takes as input the probability grids (G^f) and generates plausible future trajectories ($\hat{\mathbf{Y}}_i^f$) for the target agent. The proposed approach improved prior work results in the full real-world Stanford Drone Dataset (SDD) dataset using five predicted trajectories. The qualitative results indicate that the proposed approach was able to improve scene compliance for the tested environments.

1.2 Contributions

In this thesis we propose a novel approach to trajectory forecasting that establish a spatio-temporal correspondence between past trajectories and scene context, performing semantic scene segmentation and generating an intermediate probability map that enforces scene compliance in the multi-modal predicted trajectories. The proposed approach compares positively with the state-of-the-art results by improving the displacement error metrics on the complete SDD real-world dataset. The qualitative results show that the predicted trajectories are diverse and in conformity with the observed (past) trajectory and scene. We also provide a set of ablations that experimentally demonstrate the contribution of different networks' setups. As evaluating multi-modal trajectories is still an open problem, as most of the current approaches fail into evaluating the performance of all the predicted trajectories, we also propose a new measure to

complement commonly used prediction metrics. We summarize this thesis contributions, to the best of our knowledge, as follows:

- A new approach for multi-modal trajectory forecasting combining past trajectory, semantic scene, and surrounding agent in a spatially temporal manner. By incorporating a pre-trained semantic segmentation module, our approach can automatically extract scene features, not being dependent on high-definition maps.
- An approach that enforces scene compliance through an intermediate representation that constrains the model on feasible paths. Such intermediate representation also provides a grasp of interpretability by enabling the visualization of the learned representations. Prior approaches have struggled with generating multi-modal trajectories that learn scene features.
- An approach to measuring the precision of the estimated multi-modal trajectories as a complement to the widely used Average Displacement Error (ADE) and Final Displacement Error (FDE) metrics. A problem with the current metrics for multi-modal trajectory prediction is that only the distance of the trajectory that most closely matched the ground truth trajectory is used. That means the assessment of the quality of all the other trajectories is not taken into account. We propose a way of measuring the quality of all estimated trajectories through scene compliance.
- A wide discussion regarding qualitative results, leveraging different scenarios, and failure cases. Jointly with a set of different ablations to comprehend the contribution of each module. Most of the prior studies offer only some form of ranking but do not fully exploit/explain the generated trajectories and failure scenarios.
- An approach that does not restrict the number of surrounding agents, nor is retained to only one agent type. Most approaches that use surrounding agent data as vectors have a limitation on the number of surrounding agents they can represent. Most approaches also focus on a specific type of agent, like cars or pedestrians.

1.3 Thesis Outline

This thesis is structured as follows: Chapter 2 shows a brief history of human motion from the first studies in the area until the most recent ones. We also present how our work improves and differs from prior approaches. Chapter 3 presents the field of Machine Learning, focusing on Deep Learning, Convolutional Neural Networks, and their components. Chapter 4 presents our proposed approach, describing each one of the models that compose our architecture, implementation, and training details. In Chapter 5 we describe the data and metrics. We explain the best practices together with our data pre-processing, and some tips specific to the trajectory

forecasting problem. We present our quantitative and qualitative results, ablations studies, and discussions. Chapter 6 presents a wrap up of the thesis together with discussion of possible improvements. Published papers and journals are shown in the Appendix section.

CHAPTER

2

HUMAN MOTION FORECASTING

The human movement analysis based on vision has been a topic of research in many fields and applications, as games, character animation, surveillance systems, traffic analysis, social interfaces, sign-language translation, and dance choreography (GAVRILA, 1999). Analyzing human's actions is a key component to secure their safety on streets for surveillance or self-driving car applications. For surveillance applications, analyzing humans on cameras is used to detect suspicious behavior, and for self-driving car applications such analysis is important to avoid collisions, allowing the ego-vehicle to perform evasive steering maneuvers (KELLER *et al.*, 2011).

The detection of pedestrians has been investigated in several studies (ENZWEILER; GAVRILA, 2009; R. OMRAN M.; SCHIELE, 2015; ZHANG R. BENENSON; SCHIELE, 2016; GAVRILA; MUNDER, 2007). Most of them use images (BERTOZZI *et al.*, 2015; YE; LIANG; JIAO, 2012), 3D point clouds (MEISSNER; DIETMAYER, 2012; JIN *et al.*, 2011; WENG *et al.*, 2020), or even the fusion of both sets of information (LIN; LEE, 2016; SCHLOSSER; CHOW; KIRA, 2016). Li et al. (LI *et al.*, 2017b) designed a system for concurrently detecting pedestrians and cyclists.

The estimation of humans' intention is even more challenging due to uncertainties regarding their impending motion (FERGUSON *et al.*, 2015). In a fraction of a second, they can decide to move in one of many different possible directions, stop walking abruptly (SCHNEIDER; GAVRILA, 2013; FERGUSON *et al.*, 2015; GANDHI; TRIVEDI, 2008), have their image/point cloud occluded by a variety of obstacles, and be distracted talking over the phone or to other pedestrians. Quintero *et al.* (2015), observed the difference between an effective and a non-effective intervention can depend merely on a few centimeters or a fraction of a second.

Iacoboni *et al.* (2005) analyzed the cerebral activity of people watching others performing some actions and observed some neural cells were activated as soon as an intention had been inferred (before the action was performed). In other words, humans observing other humans'

actions can implicitly understand their intentions. According to Keller, Hermes and Gavrila (2011), algorithms still do not predict pedestrian intentions as well as humans, and pedestrians' behaviors in urban scenarios are not random (VASISHTA; VAUFREYDAZ; SPALANZANI, 2017).

A comparative study of pedestrian path prediction was conducted by Schneider and Gavrila (2013), whereas Ohn-Bar and Trivedi (2016) provides a survey on types of interactions between autonomous vehicles and humans. Shirazi and Morris (2015) reviewed pedestrian, driver, and vehicle behaviors at intersections and analyzed features that distinguish different pedestrian motion patterns. A related approach can be found in (KÖHLER *et al.*, 2015). A more recent body of literature addressing the problem of human motion forecasting can be found in (SHIRAZI; MORRIS, 2015; Ohn-Bar; Trivedi, 2016; RIDEL *et al.*, 2018; RUDENKO *et al.*, 2019).

We split the related work into Classical Approaches (Sec. 2.1), and Deep Learning-based Approaches (Sec. 2.2). In Sec. 2.1 we present the literature on behavior prediction that ranges from 1995 to 2017, most of them focused on pedestrians. In Sec. 2.2, we present the recent studies in the field focused on the advancements of CNN and RNN research. More related to our approach are the methods presented in Sec 2.2, that use deep learning with the scene, past motion, and also incorporate other agents as input to forecast long-term trajectories, by the end of the Section 2.2 we present how our approach differs from prior studies.

2.1 Classical Approaches

Several studies try to model the problem of behavior prediction by classifying human's motion regarding whether they will cross a street or not (FURUHASHI; YAMADA, 2011), classify into several motion types (crossing, stopping, bending in, and starting) or (walking, starting, stopping, and standing) (BERTOZZI *et al.*, 2004; MøGELMOSE; TRIVEDI; MOESLUND, 2015; SCHNEIDER; GAVRILA, 2013; KELLER; GAVRILA, 2014), such studies that aim at classifying are summarized in Table 1. Other approaches try to predict the exact coordinates the pedestrian will walk through in the future, a summary is available in Tab. 2.

Some studies explore the use of pedestrians' contour (KÖHLER *et al.*, 2012; KÖHLER *et al.*, 2015), posture (FURUHASHI; YAMADA, 2011; HARIYONO; JO, 2017) and body language (QUINTERO *et al.*, 2014; QUINTERO *et al.*, 2015) to predict their intentions. Some of the approaches also include pose recognition and body language (FURUHASHI; YAMADA, 2011; QUINTERO *et al.*, 2014; HARIYONO; JO, 2015a; HARIYONO; JO, 2015b). The model proposed by Köhler et al. (KÖHLER *et al.*, 2012; KÖHLER *et al.*, 2015) used a HOG-like descriptor for motion contour pedestrian detection along with a SVM to estimate pedestrians' intentions. Hariyono and Jo (2015b) used pose recognition, lateral speed, orientation, and scene comprehension as input to a neural network to predict actions, as walking, starting off, bending

Table 1 – Classical studies of classification of motion

Study/author	*Sensor/input*	*Method*	*Objective/output*	*Evaluation/dataset*
Schneider and Gavrila (2013)	Stereo-cameras	Comparative study on recursive Bayesian filters Interacting Multiple Model (IMM) and Extended Kalman Filter (EKF)	[crossing, stopping, bending in, starting]	Accuracy of position estimation and path prediction
Koehler *et al.* (2013)	Lidar and two cameras	Interacting Multiple Model Extended Kalman Filter (IMM-EKF), Motion Contour Histogram of Oriented Gradients (MCHOG) and Support Vector Machine (SVM)	[crossing, not crossing]	Receiver Operating Characteristic Curve (ROC)
Bonnin *et al.* (2014)	Image, CAN	MLP	[crossing, not crossing]	False Positive/ True Positive Rate
Völz *et al.* (2015)	Lidar	SVM	[crossing, not crossing]	False Positive/ True Positive Rate/-Classification Accuracy
Hashimoto *et al.* (2015b)	Three monocular cameras	Dynamic Bayesian Network (DBN) and Particle Filter (PF)	[crossing, not crossing]	Mean of the estimated probability of the pedestrian crossing/waiting
Kwak, Ko and Nam (2017)	Night images from thermal camera	Dynamic fuzzy	[Standing Sidewalk, Walking Sidewalk, Walking Crossing and Running-Crossing]	Precision and recall rate. Comparison with Markovian model-based method, DBN and Fuzzy Finite Automata (FFA)
Quintero *et al.* (2015)	3D pedestrian body pose	Balanced Gaussian Process Dynamical Models (B-GPDM) and naive-Bayes	Trajectory prediction and classification of behavior [walking, starting, stopping and standing]	Confusion Matrix and mean error for position

Table 2 – Classical studies of path prediction.

Study/author	***Sensor/input***	***Method***	***Objective/output***	***Evaluation/dataset***
Goldhammer *et al.* (2015)	Camera	MLP	Trajectory prediction	Mean square deviation (RMSD2D) from the predicted position to GT. Comparison with {KF Constant Velocity (CV)
Kooij *et al.* (2014)	Image	DBN	Trajectory prediction	Comparison with Probabilistic Hierarchical Trajectory Matching (PHTM) through Log Likelihood
Ferguson *et al.* (2015)	Lidar	Gaussian Process (GP)	Trajectory prediction	Probability of correct motion pattern and RMS error
Schulz and Stiefelhagen (2015b)	Stereo-cameras	IMM and Latent Dynamic Conditional Random Fields (LDCRF)	Trajectory prediction	Lateral position error
Bock *et al.* (2017)	Pedestrian with sensors	LSTM	Trajectory prediction	Mean displacement error and final displacement error. Comparison with Kalman Filter (KF). Intersection dataset based on infrastructure sensors and information on pedestrian localization. Single pedestrian
Li *et al.* (2017a)	Indoor positioning wifi data	LSTM	Trajectory prediction	Comparison with Gated Recurrent Unit (GRU) and vanilla Recurrent Neural Network (RNN)
Dominguez-Sanchez, Cazorla and Orts-Escolano (2017) (	Stereo-camera	CNN	Pedestrian moving direction	Accuracy among different CNNs
Rehder *et al.* (2018)	Images	CNN, LSTM and path planning	Trajectory and goal prediction	Predicted probability distribution and comparison with a Constant Position (CP)-IMM and CV-IMM

in, and stopping. Regression is used to predict the exact coordinate the pedestrian will occupy in the future. Some classical approaches that are based on path prediction are summarized in Tab. 2.

Karasev *et al.* (2016), modeled pedestrians' intention in a Markov decision-process framework and inferred their state using a Rao-Backwellized filter. They focused on each pedestrian individually and neglected their interactions with other traffic participants. Kitani *et al.* (2012) predicted future actions of pedestrians using noisy visual data and the effects of the physical environment on pedestrians' behavioral choices combining ideas from Control Theory and Markov Decision Processes.

Attempts towards predictions of pedestrians' positions originated from tracking, which is naturally the second step after the detection of an agent. Many studies predicted pedestrians' positions using KF and PF (BERTOZZI *et al.*, 2004; MøGELMOSE; TRIVEDI; MOESLUND, 2015), also performing comparisons between IMM, EKF (SCHNEIDER; GAVRILA, 2013; KELLER; GAVRILA, 2014), and GP, PHTM, KF, and IMM (KELLER; GAVRILA, 2014). In (HARIYONO; SHAHBAZ; JO, 2015), as in similar research initiatives, the direction of a pedestrian walking is estimated according to his/her position within multiple consecutive image frames regarding the distance from the vehicle. Switching dynamics (Linear Dynamical System (LDS)) was used by Kooij, Schneider and Gavrila (2014) towards more accurate path predictions. They established certain actions more likely to occur in the future depending on previous movements and current locations.

Nevertheless, Schmidt and Färber (2009) observed the use of only dynamics would not be sufficient, e .g. a KF tracking a pedestrian walking parallel to the ego-vehicle would always predict pedestrian's future positions to be set further. However, a pedestrian constantly turning his/her head towards the autonomous vehicle and the road is an indication of where he/she intends to go (e. g. the other side of the street). Therefore, an approach that solely relies on pedestrians' dynamics will never predict their intention of crossing a street.

Information on head orientation has been incorporated in estimation methods towards improving the estimation of pedestrians' intentions (GOLDHAMMER *et al.*, 2013; SCHULZ; STIEFELHAGEN, 2015b), which has given rise to research on perfecting the classification of head orientation (REHDER; KLOEDEN, 2015). Several studies (SCHULZ; STIEFELHAGEN, 2015b; SCHULZ; STIEFELHAGEN, 2015a; HASHIMOTO *et al.*, 2015b; HASHIMOTO *et al.*, 2015a; HARIYONO; JO, 2017) use information on pedestrian dynamics coupled with the awareness of the situation, i .e., the possible pedestrian's visualization of a vehicle and a critical situation.

Goldhammer *et al.* (2013) focused on trajectory prediction of pedestrians on crosswalks and estimated gait initiation through a piecewise linear model and a sigmoid model for calculating velocity and inferring a trajectory. They designed an approach (GOLDHAMMER *et al.*, 2015) that uses a MLP network based on head orientation information to predict a continuous trajectory for a 2.5-second future time horizon and motion types (starting and stopping). However, relying

solely on head orientation may not be the best alternative, since pedestrians may be looking at an advertisement or searching for someone; in such moments, their head might not indicate their current direction.

Some studies (CLOUTIER *et al.*, 2017; KOOIJ *et al.*, 2014) have evaluated the influence of the environment on the behavior of pedestrians. Cloutier *et al.* (2017) observed different crossing surface materials and one-way streets were significantly associated with fewer interactions with vehicles, whereas streets with parked vehicles and main streets were associated with more interactions. Several approaches use information from the environment, therefore, relations among environment, autonomous car, and pedestrians are structured (HARIYONO; JO, 2015a; HARIYONO; JO, 2015b; KOOIJ *et al.*, 2014; KIM; OWECHKO; MEDASANI, 2010; SCHULZ; STIEFELHAGEN, 2015b; HASHIMOTO *et al.*, 2015b; HASHIMOTO *et al.*, 2015a; BONNIN *et al.*, 2014; HARIYONO; JO, 2017; GU *et al.*, 2016; VÖLZ *et al.*, 2016).

Bonnin *et al.* (2014) predicted whether a pedestrian would cross a street creating relations among pedestrian, crosswalk, and ego vehicle, and combining two models, namely a standard inner-city model, which is always activated, and a model activated only in crosswalks. Their focus is on cases in which the pedestrian actually crosses the path of the ego-vehicle. DBN was used in (HASHIMOTO *et al.*, 2015a; KOOIJ *et al.*, 2014; HASHIMOTO *et al.*, 2015b). The latter, (HASHIMOTO *et al.*, 2015b), considered external surroundings context, pedestrian behavior, physical movement, and information on a pedestrian being in a group or alone (HASHIMOTO *et al.*, 2015a). Kooij *et al.* (2014) proposed a DBN that captures some factors as latent states that affect a Switching Linear Dynamics System (SLDS). A pedestrian that always intends to cross a street is the subject of the test sequences. The authors used three types of information on top of SLDS, namely 1) minimum distance between pedestrian and ego-vehicle if both keep the same velocity (indicating criticality of the situation); 2) pedestrian's head orientation (awareness); and 3) distance from the pedestrian to the curbside.

Bonnin *et al.* (2014) and Kooij *et al.* (2014) employed almost the same observable features, i .e., distance to curb, distance to ego-vehicle, and head orientation. They did not use information from other pedestrians and cars and focused on short-term predictions.

Some researchers considered decisions made by pedestrians based on social norms commonly followed within a shared common space (HELBING; MOLNÁR, 1995; PELLEGRINI *et al.*, 2009; TAMURA *et al.*, 2012; ZENG *et al.*, 2014). They observed the patterns used by pedestrians in such interactions and identified several norms, e .g., pedestrians maintain some distance from each other, pedestrians avoid others coming towards them, pedestrians can follow the flow of other pedestrians on the scene, etc.

2.2 Deep Learning

The past motion of agents is the simplest cue for forecasting their future motion. Past motion is typically represented using sequences of location coordinates obtained via detection and tracking. A majority of approaches encode such sequences using Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTMs) networks or Gated Recurrent Units (GRUs) (ALAHI *et al.*, 2016; Zyner; Worrall; Nebot, 2019; AMIRIAN; HAYET; PETTRÉ, 2019; GUPTA *et al.*, 2018; HASAN *et al.*, 2018; SADEGHIAN *et al.*, 2019; DEO; TRIVEDI, 2018).

Alternatively, some approaches use temporal convolutional networks for encoding sequences of past locations (LEE *et al.*, 2017; NIKHIL; MORRIS, 2018), allowing for faster run-times. In addition to location coordinates, some approaches also incorporate auxiliary information such as the head pose of pedestrians (HASAN *et al.*, 2018; RIDEL *et al.*, 2019) while encoding past motion.

A number of approaches jointly model the past motion of multiple agents in the scene to capture the interaction between agents (ALAHI *et al.*, 2016; LIANG *et al.*, 2019; LEE *et al.*, 2017; SADEGHIAN *et al.*, 2019; AMIRIAN; HAYET; PETTRÉ, 2019; DEO; TRIVEDI, 2018). This is typically done by pooling the RNN states of individual agents in a *social tensor* (ALAHI *et al.*, 2016; LEE *et al.*, 2017; DEO; TRIVEDI, 2018), using graph neural networks (VEMULA; MUELLING; OH, 2018) or by modeling pairwise distances between agents along with max-pooling (GUPTA *et al.*, 2018; SADEGHIAN *et al.*, 2019; AMIRIAN; HAYET; PETTRÉ, 2019).

Locations of static scene elements such as roads, sidewalks, crosswalks, and obstacles such as buildings and foliage constrain the motion of agents, making them a useful cue for motion forecasting. Most recent approaches use Convolutional Neural Networks (CNNs) to encode the static scene context, either by applying the CNNs to bird's eye view images (SADEGHIAN *et al.*, 2019; SADEGHIAN *et al.*, 2018; LEE *et al.*, 2017), high fidelity maps (CUI *et al.*, 2019; CHOU *et al.*, 2019), or LiDAR point cloud statistics in the bird's eye view (ZENG *et al.*, 2019; RHINEHART; KITANI; VERNAZA, 2018).

Alahi *et al.* (2016) propose a method that applied a social layer over the LSTM network for each pedestrian and implicitly learned such interactions through the sharing of LSTM's hidden states. Lee *et al.* (2017) also incorporate other agents' motion in the trajectory forecasting of the agent being predicted. Other approaches (GUPTA *et al.*, 2018) globally learn such pooling getting features from all agents in the scene.

An inherent difficulty in motion forecasting is its multi-modal nature. There are multiple plausible future trajectories at any given instant due to latent goals of agents and multiple paths to each goal. Regression-based approaches for motion forecasting tend to average these modes, often leading to implausible forecasts. Prior work has addressed this challenge by learning one-to-many mappings. This is most commonly done by sampling generative models such as Generative Adversarial Networks (GANs) (GUPTA *et al.*, 2018; SADEGHIAN *et al.*, 2019;

AMIRIAN; HAYET; PETTRÉ, 2019), Variational Autoencoders (VAEs) (LEE *et al.*, 2017) and invertible models (RHINEHART; KITANI; VERNAZA, 2018). Some approaches sample a stochastic policy obtained using imitation learning or inverse reinforcement learning (Li, 2019; DEO; TRIVEDI, 2019). Other approaches learn mixture models (CUI *et al.*, 2019; Zyner; Worrall; Nebot, 2019; DEO; TRIVEDI, 2018; Deo; Trivedi, 2018). Table 3 provides an overview of the most recent studies performing trajectory forecasting.

2.3 Final Considerations

In this chapter, we have presented from the classical studies in behavior prediction to the most recent ones leveraging the power of deep learning networks. Differently from (CASAS; LUO; URTASUN, 2018; LUO; YANG; URTASUN, 2018; CHOU *et al.*, 2019; DJURIC *et al.*, 2020b; ALAHI *et al.*, 2016; SADEGHIAN *et al.*, 2018), we use multi-modal trajectories that allow the car to be more robust to avoid collisions. We output a fixed number of output trajectories and use the best-of-k prediction loss to train the model similar to (GUPTA *et al.*, 2018; CUI *et al.*, 2019). We also do not condition the trajectory to actor specific types, as in (WANG *et al.*, 2020; CASAS; LUO; URTASUN, 2018; LUO; YANG; URTASUN, 2018; CUI *et al.*, 2019; DJURIC *et al.*, 2020b; ALAHI *et al.*, 2016; GUPTA *et al.*, 2018), nor rely on high-definition maps as in (DJURIC *et al.*, 2020a; NIEDOBA *et al.*, 2019; WANG; ZHANG; YI, 2017; CASAS; LUO; URTASUN, 2018; CUI *et al.*, 2019; CHOU *et al.*, 2019; DJURIC *et al.*, 2020b). High-definition maps are rasterized top-down scenes with road and crosswalk locations, lane directions, observed traffic lights, and signage. Such maps are usually not scalable, being challenging to maintain and store. We pre-train a model to learn how to automatically segment the BEV image into semantics that are useful for the trajectory prediction problem. From the works that automatically learn scene context (SADEGHIAN *et al.*, 2019; SADEGHIAN *et al.*, 2018) we differ from them by representing the scene and the past trajectory in the same frame of reference. We represent the past trajectories of the agents using one-hot 2-D grids, and the underlying scene as a RGB BEV image, with an agent-centric frame of reference. Closest to our approach is the model proposed by Li (2017), which uses a ConvLSTM encoder-decoder trained on a grid-based representation of past motion. However, unlike our model, they do not encode the static scene and surrounding agents. By using the same frame of reference for the input data, and creating an intermediate step that generates a time-expanded probability map, we enforce the usage of scene context for the trajectory forecasting task, improving prior work results.

Table 3 – Deep Learning-based studies

Study/author	*Sensor/input*	*HD maps*	*Semantic segmentation*	*Multi-modal*	*Pedestrians*	*Cars*	*Cyclists*	**Skaters**	*Surrounding actors*	*GANs*	*End point*
Djuric *et al.* (2020a)	Point cloud and HD maps[a]	yes	no	yes	yes	yes	yes	yes	yes	no	no
Niedoba *et al.* (2019)	States[b] and HD maps	yes	no	yes	yes	yes	yes	no	yes	no	no
Wang *et al.* (2020)	States and HD maps	yes	no	yes	no	yes	no	no	yes	yes	no
Casas, Luo and Urtasun (2018)	Point cloud and HD maps	yes	no	no	no	yes	no	no	yes	no	no
Luo, Yang and Urtasun (2018)	Point cloud	no	no	no	no	yes	no	no	yes	no	no
Cui *et al.* (2019)	States and HD maps	yes	no	yes	no	yes	no	no	yes	no	no
Chou *et al.* (2019)	States and HD maps	yes	no	no	yes	no	yes	no	yes	no	no
Djuric *et al.* (2020b)	States and HD maps	yes	no	no	no	yes	no	no	yes	no	no
Sadeghian *et al.* (2019)	static scene context path history of all the agents	no	yes	yes	yes	yes	yes	yes	yes	yes	no
Alahi *et al.* (2016)	past trajectories	no	no	no	yes	no	no	no	yes	no	no
Gupta *et al.* (2018)	past trajectories	no	no	yes	yes	no	no	no	yes	yes	no
Sadeghian *et al.* (2018)	raw top-view images and past trajectory	no	yes	no	yes	yes	yes	yes	no	no	no
Lee *et al.* (2017)	past motion and scene context	no	no	yes	yes	yes	yes	yes	yes	VAE	no

[a] HD map is usually a rasterized top-down scene with road and crosswalk locations, lane directions, observed traffic lights and signage)
[b] Actor states are commonly bbox center, position, velocity, acceleration, heading, and heading change rate.

CHAPTER

3

THEORETICAL BACKGROUND

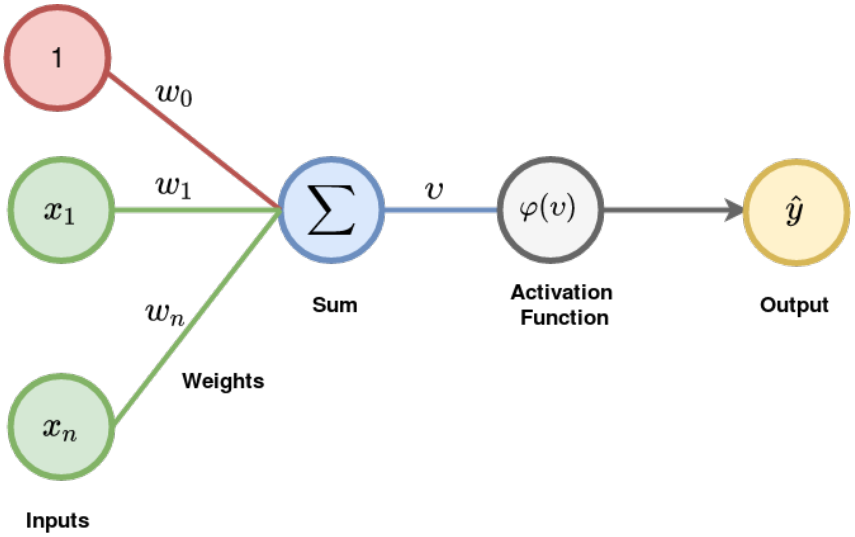
Imagine a task in which given a year, an algorithm has to return the names of the Nobel Prize laureates in Economics. In that scenario, the input and output pair is known. Such information could be stored in a hash table, and the answer could be easily retrieved by a simple request. The rule there is clear, humans can understand and describe the task at hand, an algorithm that was given as input the year would be able to straightforwardly solve that task.

Now, imagine a new task of grouping sets of spoken words according to the owner of the speech. A human listening to the words might be able to somehow group the words with some confidence, but describing each decision might be challenging (GOODFELLOW; BENGIO; COURVILLE, 2016). To solve such tasks that are not easily describable by humans, the algorithms had to become smarter, to discover/infer such criteria. Patterns could be extracted from the data and therefore classified using classical Machine Learning (ML) algorithms.

Machine Learning algorithms are commonly split into four categories (MARSLAND, 2014): Supervised Learning, Unsupervised Learning, Reinforcement Learning, and Evolutionary Learning. In Supervised Learning, the ground truth label for each input sample is provided. These labels contribute to train the model to solve the proposed task. In Unsupervised Learning the labels are not used, the algorithm group similar examples by computing distance measurements among the input data. Reinforcement Learning is often considered a semi-supervised learning approach, the algorithm gets feedback for its estimated output, but it does not have access to the step-by-step process to correct each one of its choices. The algorithm has to find the correct answer in a try and error approach, trying different possibilities until converging to the correct answer. Evolutionary Learning is based on the theory of evolution, in which sets of individuals mutate characteristics until evolving to a set where an acceptable fitness value is reached.

Supervised Learning tasks are usually described as a set of data $(\mathbf{x_i}, \mathbf{y_i})$, where $\mathbf{x_i}$ is the input and $\mathbf{y_i}$ is the ground truth output, i is an index representing the different samples inside the set. Given an input $\mathbf{x_i}$, the algorithm will output an estimated answer $\hat{\mathbf{y}}_\mathbf{i}$, the algorithm is

Figure 1 – The Perceptron.



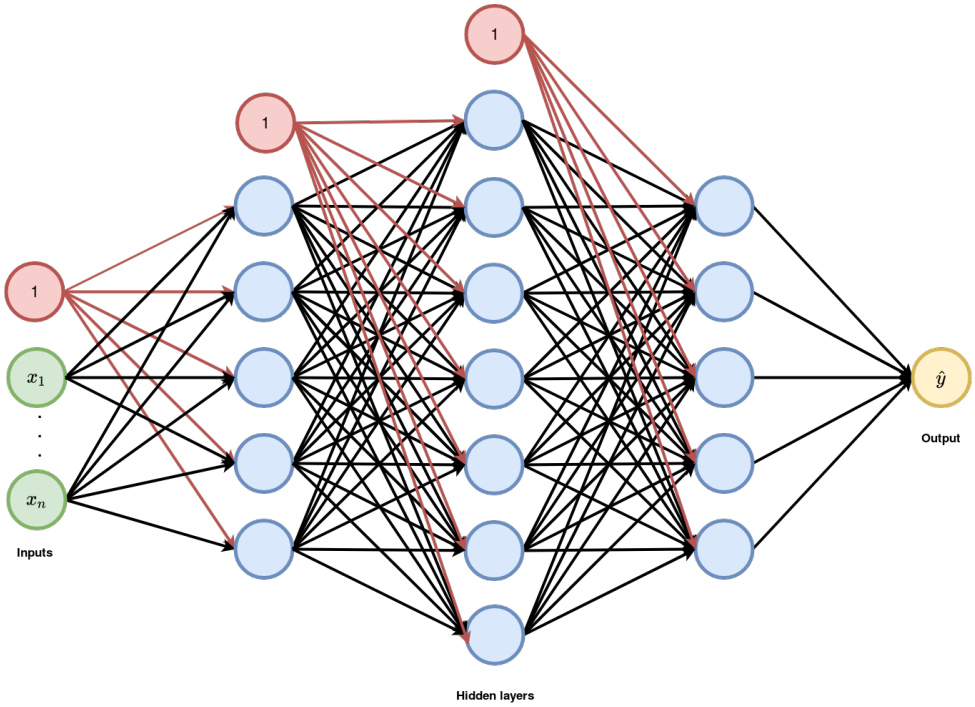
Source: Adapted from (AMINI, 2020)

then able to measure the distance between its estimation $\hat{\mathbf{y}}_\mathbf{i}$ and the ground truth answer $\mathbf{y}_\mathbf{i}$, using such information to improve its estimate in the next iteration. According to Mitchell (1997), a computer program is said to learn from experience E with respect to a task T and performance measure P, if its performance at the task T improves with experience E. The performance measure P is an important factor because it is the main source of information for the algorithm to learn.

The concept of learning might be challenging to be explained, as the human brain learning mechanisms are still not fully understood. The first studies modeling the human brain date back to the 40's (MCCULLOCH; PITTS, 1943). Hebb's theory (HEBB, 1949) supports the fact that the strength of a synapse connection between neurons gets stronger when they fire simultaneously. Neurons have a set of dendrites that connects with the axons of other neurons. The amount of input information that will be delivered to the soma is regulated through the synapses' strength. The soma is responsible to process all those signals, and the axons redistribute the result to other neural cells. In Fig. 1 we illustrate the Perceptron model proposed by Rosenblatt (1958). The model is inspired by the biological concept of a human brain neuron, where inputs are represented by green circles ranging from x_1 to x_n, the weights (w_1 to w_n) connect each one of the n inputs to the soma (blue circle). Such inputs are pondered by the weights and summed up plus the bias (w_0). The weights are responsible for controlling how much of each input will pass to the soma. The output υ passes through an activation function that limits the amplitude of the neuron output (HAYKIN, 2009), $\hat{y}$ is the final output of the i-th neuron. Eq. 3.1.

$$\hat{y} = \varphi(\upsilon) = \varphi\left(\left(\sum_{j=1}^{n} w_{kj} x_j\right) + b\right). \tag{3.1}$$

Figure 2 – Example of a MLP with three hidden layers.



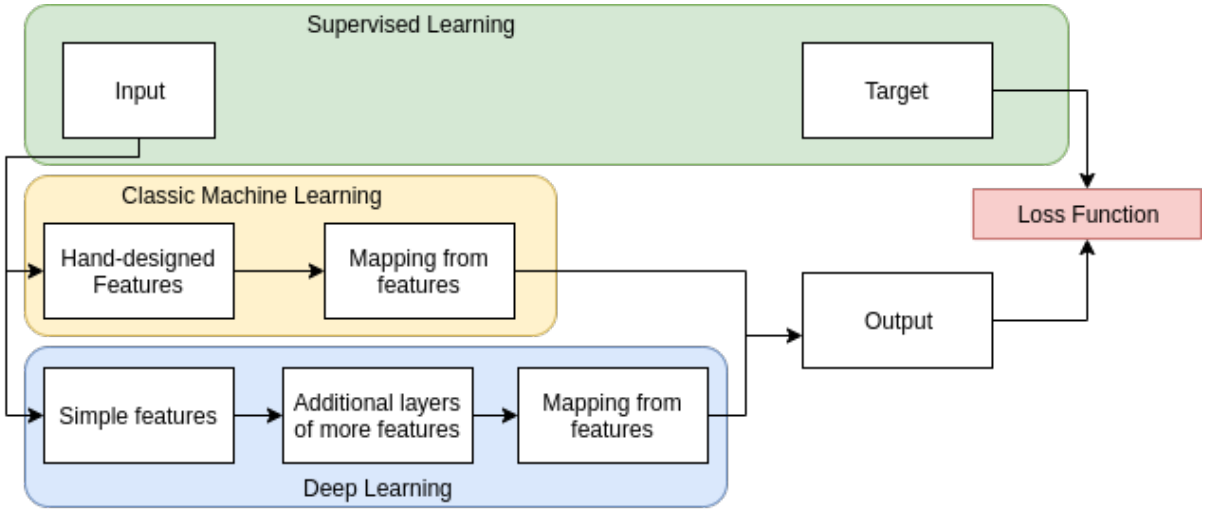
Source: Elaborated by the author.

To train a perceptron means finding the right weights and bias (w_0 to w_n), i .e., creating a linear separation in the amostral space. As in the human brain, a neuron does not work alone but in a network where axons of several neurons are "connected" to others neurons, Multilayer Perceptron (MLPs) (CHURCHLAND; SEJNOWSKI, 1992) became popular, years ago, as they could solve a range of challenging tasks at that moment.

MLP consists of a network in which there are one or more hidden layers between the input and the output layer. The advantage of using more layers is that such networks can create more hyperplanes. While the perceptron maps the inputs directly to the outputs, the MLP extract higher-order statistics from the input (HAYKIN, 2009). Such networks are feed-forward and fully-connected. In a feed-forward network, the information just flows in one direction (forward). Fully-connected means that all the neurons in a layer are connected to all the neurons in the next layer. We show an example of a MLP in Figure 2, the network has one input layer (the neurons here just store the information, not applying any computation), three hidden layers, and one output layer. MLPs have been used for a long time to solve many different tasks such as handwriting recognition, play checkers, and even autonomous driving (POMERLEAU, 1989; MITCHELL, 1997).

According to Goodfellow, Bengio and Courville (2016), ML has the ability of enabling machines to learn from experience and solve tasks by creating a hierarchy of concepts and their relations. The idea of a computer being able to learn from a hierarchy that goes from simpler to more complex concepts is what turned classical ML into DL (GOODFELLOW; BENGIO;

Figure 3 – **Machine Learning**. Differences between classical Machine Learning and Deep Learning.



Adapted from (GOODFELLOW; BENGIO; COURVILLE, 2016).

COURVILLE, 2016). The huge advancement of DL algorithms comes with their ability to automatically extract useful features from data. Formerly Neuroscience was regarded as the main source for understanding and developing models that tried to emulate the human brain. However, according to Goodfellow, Bengio and Courville (2016), DL appeals for a more general idea of multiple levels of knowledge, creating a composition that is not necessarily brain-inspired. Such models became more popular nowadays because of the increase in computational resources, which enabled the size of the models to become larger. Fig. 3 highlights the main differences between classic Machine Learning and Deep Learning, also exemplifying Supervised Learning. The right factorization of knowledge is the key for robustness, whether classical AI has explicitly steps, deep learning is still not fully understood (BENGIO, 2019). The unconscious and conscious ability of the human brain has previously been associated by Bengio (2019) as Machine Learning playing the conscious aspect and the unconscious being Deep Learning. The exploration of explainability in deep networks can help to bridge this gap between Deep Learning and classical learning methods.

3.1 Deep Learning

Recently, with the increase of computational power and the creation of Graphical Processing Units (GPU), the development of deeper networks (larger number of layers) became possible. That enabled the networks to create a larger number of connections to learn complex problems in a way that was not possible before. According to (GOODFELLOW; BENGIO; COURVILLE, 2016) there is not a consensus regarding the number of layers to an architecture to be identified as a deep model. The number of layers in the first architectures LecunNet (LECUN *et al.*, 1998), Alexnet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), and GoogleNet (SZEGEDY *et al.*,

2015) range from 7 to 22 layers, and nowadays there are models that reach 110 layers (HAN; KIM; KIM, 2016). We know that in practice the number of hidden layers can grow very large.

Some advantages of using DL is that the features are learned from the data, i .e., they are not hand-crafted as in classical machine learning techniques. A key achievement attributed to Deep Learning is that the models could learn to select the best features to solve the given task, but the capacity of machine learning algorithms has to be appropriated for the true complexity of the given task, therefore if you have the right model and enough data, the model will perform best (GOODFELLOW; BENGIO; COURVILLE, 2016). The representational capacity of the model is the name given to the selection of the algorithm family chosen to solve the task. The process of designing a DL architecture is a vital task to the success of the problem. The model selection by the engineer behind the network is a very important step because the model specifies which family of functions the learning algorithm can choose from when learning the parameters of such model (GOODFELLOW; BENGIO; COURVILLE, 2016). The model will perform coordinately better if we set the preferences that are aligned with the learning problem we ask the algorithm to solve (GOODFELLOW; BENGIO; COURVILLE, 2016). Therefore the capacity of the model depends not only on the selection of the appropriate set of algorithm family but also on the training procedure to set the best parameters for such model.

The loss function (also known as performance measure, or objective function) is used to evaluate how good (or bad) the current iteration performed over the current network parameters. This value is also used to backpropagate the error and update the weights. The task of minimizing some loss function $f(x, \theta)$ by changing θ is refereed as optimization. Most Deep Learning approaches involve some type of optimization to select the best parameters of the model. In most cases, finding the exact set of parameters to solve the task (global minimum) is a difficult optimization problem (GOODFELLOW; BENGIO; COURVILLE, 2016), as the loss space is non-convex. Because of that, a set of best practices exists to ensure the parameters will not be extremely optimized to the train data and perform poorly on the test set (overfitting). The opposite is also a problem, underfitting occurs when the model is not able to make the training error small (GOODFELLOW; BENGIO; COURVILLE, 2016).

Preventing overfitting is a key component of this optimization problem, this means finding the right time to stop training. Techniques as data augmentation, early stopping, dropout, and batch normalization come at hand to improve generalization. Increasing your input data (Data Augmentation) is often used, especially when the dataset is small, where modifications in the data are performed to generate new input samples. Early stopping is a method where the training is stopped when the validation set is at its lowest error, in practice, you save a copy of the model parameters only when the validation error is decreased. When the training finishes you return the stored parameters, and not the current iteration parameters (GOODFELLOW; BENGIO; COURVILLE, 2016). Dropout regularizes units to be not only a good feature but a feature that is good in many contexts (GOODFELLOW; BENGIO; COURVILLE, 2016).

Dropout is commonly used as a regularization method. It randomly inactivates some input data with probability p using samples from a Bernoulli distribution[1]. In contrast to training, during testing all input information is used.

Batch normalization (Ioffe and Szegedy, 2015) is a method of adaptive reparametrization (GOODFELLOW; BENGIO; COURVILLE, 2016). When optimizing deep neural networks, the update of weights in backpropagation steps assumes that the prior layers will not be modified, however in fact they are modified, and that affects the current layer being updated. Batch normalization aims at producing a zero mean unit variance over all the batch in intermediate layers of a network, therefore improving optimization:

$$y = \frac{x - E[x]]}{\sqrt{Var[x] + \varepsilon]}} * \gamma + \beta, \tag{3.2}$$

where ε is a small positive value imposed to avoid zero, the variables γ e β are learned parameters that enable the model to recover the original values, the new parametrization is easier to be learned using gradient descent (GOODFELLOW; BENGIO; COURVILLE, 2016).

In general, most of the DL approaches training procedure find a local minima that is good enough to solve the problem (GOODFELLOW; BENGIO; COURVILLE, 2016). In the real world, the best set of parameters is never found because the loss space of functions demands a huge set of parameters to be set, and is still very poorly understood (CHOROMANSKA *et al.*, 2015). Many of the approaches usually settle for a loss value that is low but not mandatory minimal in any formal sense, in other words, local minima is often chosen as an acceptable answer as far as they map to significant low values of the loss function (GOODFELLOW; BENGIO; COURVILLE, 2016). Choromanska *et al.* (2015) noticed that a considerable amount of researchers that used deep neural networks and Stochastic Gradient descent to train, obtained consistent results with similar performances after multiple experiments. Concluding that despite multilayer nets have many local minima they are easy to find and they are equivalent when considering the performance in the test set. Nowadays, optimization algorithms such as Adam (KINGMA; BA, 2014) has been widely used and they provide a way to automatically adjust each parameter learning rate. As an area still under development there are many issues when training deep neural networks, and according to (HASTIE; FRIEDMAN, 2010) it can even be considered an art, as the optimization problem is non-convex, it has many local minimum and is unstable unless some guidelines are followed.

While the parameters (the weights, or filters' values of the network) are automatically learned during training through the backpropagation, the hyperparameters are network design choices made by the human behind the network, or autoML/optimization algorithms. Such hyperparameters play a key role because they are defining the architecture being used and it can interfere in the success of your network learning process to ensure generalization. Finding

[1] <https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html>

the right set of hyperparameters might be a challenging task for a beginner in the field. Hastie and Friedman (2010) suggest that it is better to have more than fewer layers, as with proper regularization the excess of weights can be shrunk towards zero, and with too few layers the model can be harmed by not having enough flexibility to capture the nonlinearities. Other authors (BENGIO; DELALLEAU; ROUX, 2005) affirm that if the number of neurons grows too much that can also be a problem (The curse of dimensionality), i.e., if the number of variables increases the number of possible arrangements increases exponentially proportional (GOODFELLOW; BENGIO; COURVILLE, 2016).

Deep learning models are well known to be able to select the best features for a given task, however, the human behind the network can make things easier by crafting an architecture that will make the task easier, and also, pre-processing the input enables the network to converge faster to an answer. Some tricks commonly used in the literature can help to accelerate the training. The community is aware of the benefits of normalizing images before passing them to CNNs, using zero mean and one variation for different types of data, usually in image tasks we use only the zero mean because images already have a fixed relation among its data (because of pixels' grid structure) (KARPATHY, 2016). Techniques as Transfer Learning and Fine Tuning have been demonstrated to help, as usually, the features extracted in the first layers can be similar in many tasks, so the parameters in those layers can be imported from a training step in prior datasets.

3.2 Recurrent Neural Networks

Recurrent Neural Networks are known due to their capacity to maintain states over time and also to discover the contextual relationship between inputs (LI *et al.*, 2017a). The main difference between RNN and feed-forward networks is that the hidden layers, in RNNs, are connected among different time-steps, therefore the network can exchange information from past time-steps. Such RNNs can be used in many different settings as, in many-to-many, many-to-one, or one-to-many. For example, a one-to-many approach can be used for image captioning, where one image is given as input and many words describing the image are the output, and a many-to-one approach can be used for sentiment analysis where many inputs are given (e .g., a small text), and just one output is predicted (e .g., the dominant sentiment in the text)[2].

Recently, given the growth in the computer power capacity and the highlight given to Deep Learning approaches, some Deep RNN variants have been proposed as LSTM (HOCHREITER; SCHMIDHUBER, 1997) and GRU (CHO; MERRIENBOER; BAHDANAU, 2014). The work in Chung *et al.* (2014), Karpathy, Johnson and Li (2015) performs comparisons among RNNs and its variations, and the work Greff *et al.* (2017) performs comparisons among LSTM variants. LSTM has shown good results for applications where the long term affects the current

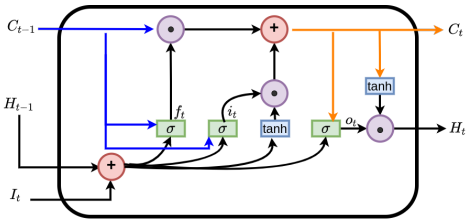
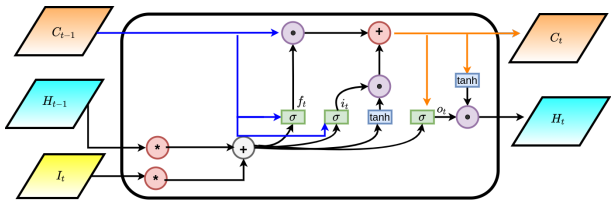
[2] <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

state.

A variation of LSTM was proposed by Cho, Merrienboer and Bahdanau (2014). The GRU is considered a simplified version (GREFF *et al.*, 2017) because it simplifies the number of parameters and might decrease the computation cost, where instead of using the input and forget gate just an update gate is used, and the peephole connections and output activations were excluded. Chung *et al.* (2014) provides a comparison between LSTM and GRU.

The biggest difference from the LSTM to the ConvLSTM (SHI *et al.*, 2015) is that in the latter all the inputs, cell outputs and hidden states are 3D tensors, and instead of fully connections, convolutions are used. In Tab. 4 we illustrate the peephole LSTM variation introduced by (GERS; SCHMIDHUBER, 2000) and ConvLSTM equations as presented in (GRAVES, 2013; SHI *et al.*, 2015).

Table 4 – Comparison between LSTM and ConvLSTM

LSTM	ConvLSTM
$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i)$	$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i)$
$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f)$	$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f)$
$c_t = f_t \circ c_{t-1} + i_t \circ tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$	$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$
$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o)$	$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o)$
$h_t = o_t \circ tanh(c_t)$	$H_t = o_t \circ tanh(C_t)$
	
Adapted from (RAHMAN; SIDDIQUI, 2019)	Adapted from (KIM *et al.*, 2020)

3.3 Convolutional Neural Networks

The initial studies that resulted in the, now widely know, Convolutional Neural Networks started in the 80's with Fukushima (1980) neocognitron architecture mimicking the mammalian visual system (GOODFELLOW; BENGIO; COURVILLE, 2016). Such architecture inspired the LeNet-5 proposed by Lecun *et al.* (1998) with applications in digits recognition. Later, in the 2010s the AlexNet proposed architecture (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) reached state-of-the-art results at the ImageNet competition, bringing the attention of several researchers to the power of CNNs. Nowadays the usage of CNNs is spread all over the globe, and they have already been applied to a diverse range of tasks.

The main difference between CNNs and MLPs is the use of convolution layers (GOODFELLOW; BENGIO; COURVILLE, 2016; PONTI; COSTA, 2017). The name convolution is used because a sliding window passes through the original image convolving it with $k \times k$ filters resulting in new images, Fig. 4. Those filters are composed of the weights the network should learn in the training step. In a fully connected network (as an MLP), generally, all the neurons on a previous layer are connected to all the neurons on the current layer. In a convolutional layer, an output value is given by each filter convolution in each local region of the layer's input image. This "overlapping" region in the image is also known as the receptive field, and the output from the filter convolution in the image is known as the feature map. Therefore, instead of having all the neurons in one layer connected to all the neurons in the next layer (fully connected), the connections between two layers in a CNN are considered to be sparse, if the kernel size is smaller than the input grid size, as shown in Fig. 5.

An important parameter here is the number of pixels the sliding window will skip in the convolution (also know as stride). One can notice that a convolution operation would result in a smaller output image, a common approach to avoid this reduction is the usage of padding. There are several different types of padding, adding zeros to the created borders, copying the borders of the image, and reflecting the image rows values (ReflectionPad, Fig. 6) [3].

Commonly, a CNN is composed of several building blocks, Fig. 7. Each building block receives an input, processes it, and gives an output to the next block. There are building blocks to perform convolutions (e .g., filters for discovering edges), to apply activation functions (e .g., Rectified Linear Units (ReLU)), and to decrease the amount of data (e .g., max-pooling). The ReLU activation function, Eq. 3.3, is used to eliminate the negative numbers that can result after the convolution, this can be interpreted as the positive numbers being the ones that provided a significant response from the filters (MELLO; FERREIRA; PONTI, 2017). By using ReLU, it is possible to keep the non-linearity in the resulted map, i .e., activation functions help to introduce non-linearities in the model. Non-linearities are important because otherwise, the network would only be able to apply linear transformations. A variation of ReLU is the LeakyReLU (Eq. 3.4), where the negative values are close to 0, but not 0.

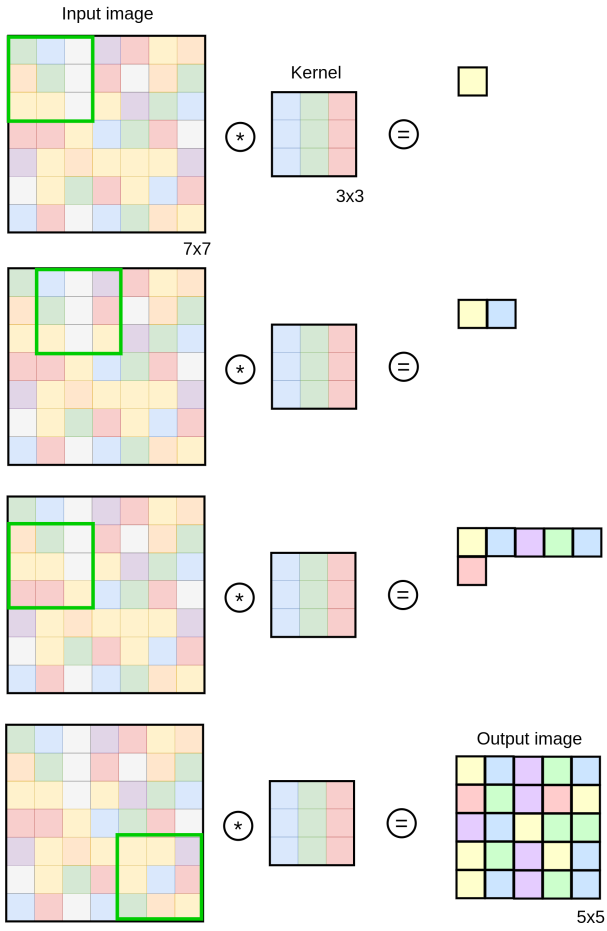
$$\textbf{ReLU} = max(0, \upsilon) \tag{3.3}$$

$$\textbf{LeakyRELU} = \begin{cases} \upsilon, \text{if} \geq 0 \\ \text{negative slope} \times \upsilon, \text{else} \end{cases} \tag{3.4}$$

Pooling layers are usually applied after some convolutional operations (PONTI; COSTA, 2017) and they downsample the amount of data without losing the important information. Max-pooling (Fig. 8) consists of selecting the maximum value among a region in the given input,

[3] <https://pytorch.org/docs/stable/generated/torch.nn.ReflectionPad2d.html>

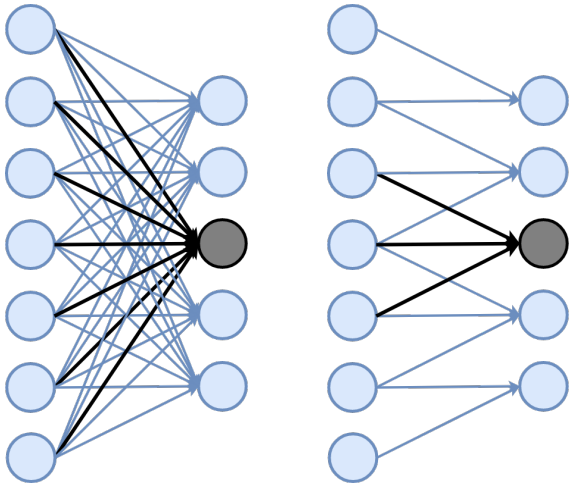
Figure 4 – 2D convolution over an input image. A filter, or kernel, is convolved with each receptive field in the input image, generating an output value. All output values together form then the feature map. The stride value controls the number of pixels that will be skipped in each horizontal and vertical direction, in an approach that resembles a sliding window. At each convolution between the kernel and receptive field (depicted in green), an output value is computed. For simplicity we represent the image and kernel as 2D images, i .e., channel dimension is not displayed.



Source: Elaborated by the author.

therefore decreasing data dimensionality. The implicit idea behind it is that the maximum value can indicate whether a feature was present or not in each image patch.

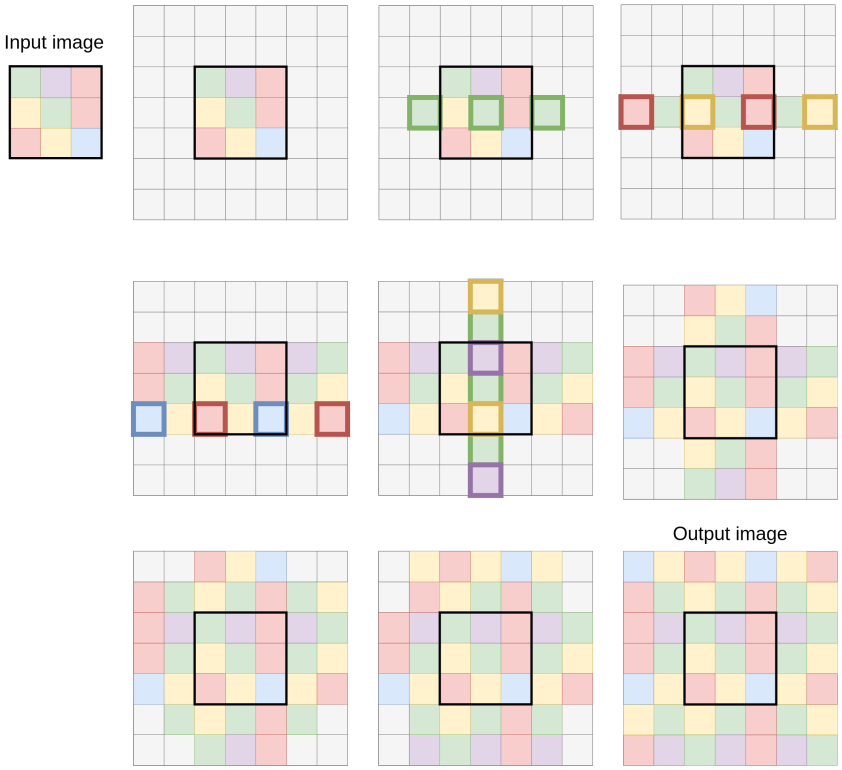
Figure 5 – Difference between fully connected and convolutional layer.



(a) Densely connected (b) Sparsely connected

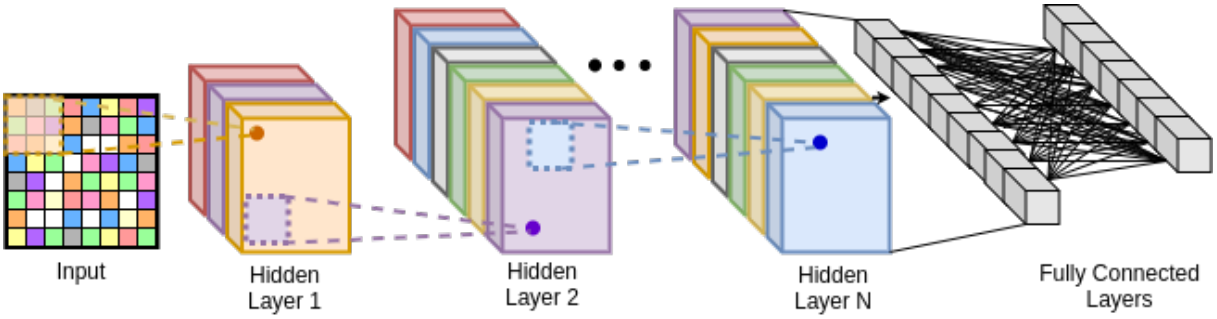
Source: Adapted from (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figure 6 – 2D ReflectionPad: A method for padding the input image reflecting values from the image itself.



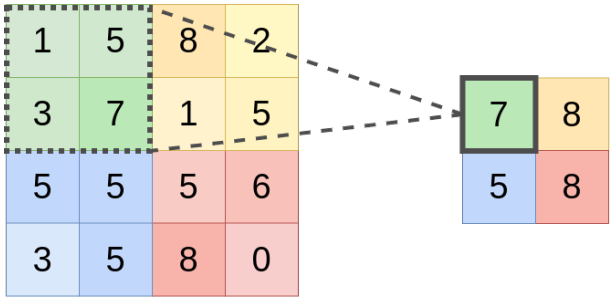
Source: Elaborated by the author.

Figure 7 – Example of Convolutional Neural Network Architecture for image classification.



Source: Elaborated by the author.

Figure 8 – Max-pooling applied to an image reduces its spatial size while selecting the maximum value among a region. Implicitly this can be interpreted as the most important information is whether some feature was detected in that region instead of its exact location.



Source: Elaborated by the author.

Each filter convolution with the input image generates a different feature/output map. Those features maps can then serve as input to another layer, therefore generating a new feature map, in this way the network can learn simpler features (as edges and corners) in the first layers, to more complex representations on the last layers (as faces, cars, and urban scenes). A network with a smaller filter size and a bigger number of layers is preferable over a larger filter size and a smaller number of layers because both of them would reach a similar coverage area however the smaller filter option would request fewer parameters and computations, and allow more nonlinearities (JOHNSON, 2016). Along with the convolutional layers, the network learns how to extract from simple to more robust features in images.

In CNNs for image classification, usually, after many hidden layers all the output grids are flattened into a vector (feature vector) and it is given as input to a fully-connected layer (e .g., MLP), Fig 7. This network is then responsible for performing the classification step, where each neuron in the hidden layer is connected to each value in the layer's input vector. The last layer in this network produces the classification probabilities for the trained classes. Softmax is commonly used when you want a probabilistic answer, the results will be arranged in a 0 to 1

scale, with sum 1:

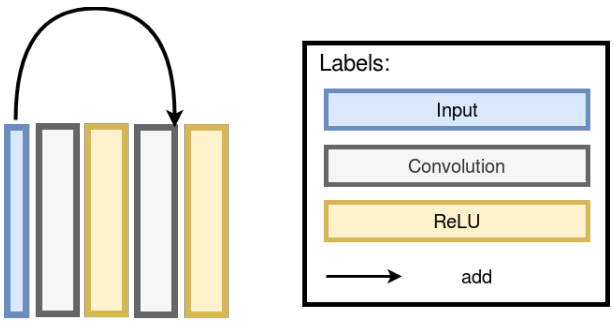
$$Softmax(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)}. \tag{3.5}$$

In classical image processing approaches, the features were hand-crafted by using feature extractors as: corner and edge detector (HARRIS; STEPHENS *et al.*, 1988; SHI; TOMASI, 1994), SIFT (LOWE, 2004), HOG (DALAL; TRIGGS, 2005), SURF (BAY; TUYTELAARS; GOOL, 2006), FAST (ROSTEN; DRUMMOND, 2006), BRIEF (CALONDER *et al.*, 2010), ORB (RUBLEE *et al.*, 2011). In CNNs, the most suitable features are automatically learned from the data, usually CNNs last layers comprise a set of fully connected layers that learns how to classify. Different types of CNNs have been explored since the past years exploring a extensive range of tasks: classify objects (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; SZEGEDY *et al.*, 2015; SIMONYAN; ZISSERMAN, 2015; HOWARD *et al.*, 2017; TAN; LE, 2019; He *et al.*, 2016), classify each pixel in the image (KIRILLOV *et al.*, 2019; SANDLER *et al.*, 2018; RONNEBERGER; FISCHER; BROX, 2015), detect objects in the image (GIRSHICK, 2015; GIRSHICK *et al.*, 2016; REDMON *et al.*, 2016), and instance segmentation (HE *et al.*, 2017; CHEN *et al.*, 2019). Several studies perform joint tasks as: detection and tracking, detection and classification, pixel-wise segmentation and classification, or pixel-wise segmentation with instance segmentation.

3.4 ResNet

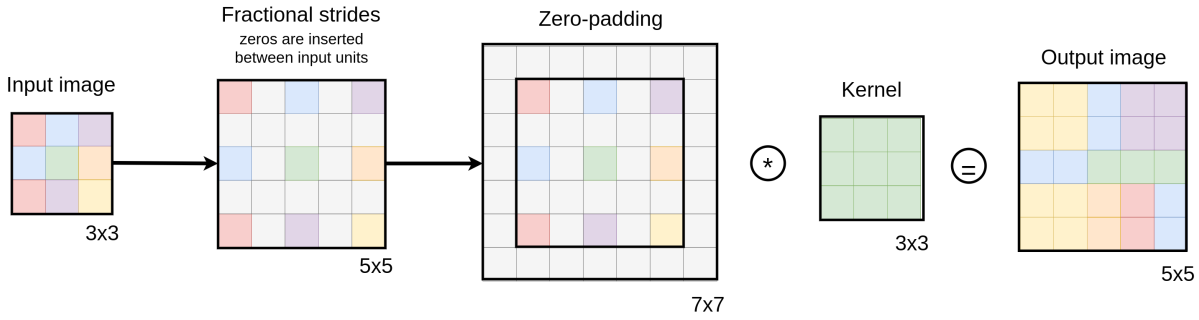
The main purpose of Residual Networks (ResNet) is to facilitate the training of deep networks. He *et al.* (2016) provided evidence that using the proposed residual blocks (Fig. 13) allowed networks to optimize easier than similar ones without the skip connections. These skip connections perform identity mapping into the input and are added together with the processed output. The advantage of this approach is that it does not add parameters. ResNet blocks prevent vanishing gradient problem while also allowing us to just skip layers and implicitly decrease the number of layers necessary for the task at hand.

Figure 9 – Skip connections proposed by (He *et al.*, 2016)



Source: Elaborated by the author.

Figure 10 – A method for upscale input images consists of adding zeros between the input data, optionally adding padding, and performing a convolution with a kernel, resulting in an image with a bigger size.



Source: Elaborated by the author.

3.5 U-Net

The U-Net (RONNEBERGER; FISCHER; BROX, 2015), as the name suggests, consists of a U-shape architecture, where the downsample path is followed by an upsample path, as shown in Fig. 11. The advancement from prior models was the expansive path being symmetric to the downsampling path. By having a large number of feature channels the network can propagate context information to higher resolution layers. The downsample path consists of a set of 3x3 unpadded convolutions followed by ReLU layer and a 2x2 MaxPooling with stride two. While the width and height of the image get smaller because of the unpadded convolutions and MaxPooling operations, the number of feature channels is doubled at each operation. The upsampling path is symmetrical to the downsampling, allowing that in each same size pair, the corresponding feature map in the downsampling path is copied into the upsampling feature map. The upsampling feature map, in each step, is composed of an up convolution and a decrease by half on the feature channels and two 3x3 convolutions followed by ReLU. A final 1x1 convolution is applied to map to the desired output.

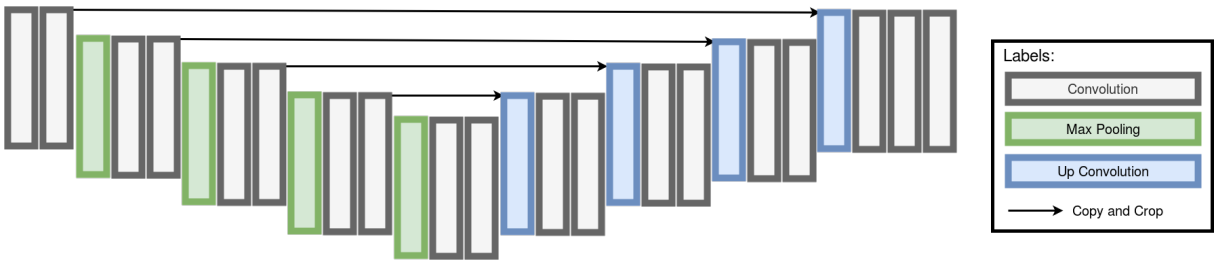
A 2D transposed convolution (Fig. 10) can be used when we want to increase the shape of the input image in a upconv manner, it is also known as a fractionally-strided convolution[4]. In a fractionally-strided convolution, zeros are filled among neighboring pixels in the input, and optionally padding is added around the increased image, reconstructing the spatial resolution. Then a convolution is done over the expanded image and a filter, resulting in an output image of a bigger size.

3.6 Final Considerations

Deep Learning has opened a new era for Machine Learning applications. The allowance of usage of a higher number of layers enabled the network to extract from low to high dimensional

[4] <https://pytorch.org/docs/stable/generated/torch.nn.ConvTranspose2d.html>

Figure 11 – U-Net proposed by (RONNEBERGER; FISCHER; BROX, 2015)



Source: Elaborated by the author.

features, therefore achieving higher accuracy in several datasets to solve a diverse set of problems. The process of designing the DL architecture is a vital task to the success of the problem. Several tricks are well known in the literature to improve training convergence and generalization. LSTMs were able to fix the vanishing gradient problem present in Vanilla RNNs. U-Nets and ResNets were used in several image problems and have demonstrated their effectiveness in several recent years' applications.

CHAPTER

4

PROPOSED APPROACH

Our main task is to generate K plausible trajectories that are diverse, and compliant to the scene. We use supervised learning to predict multiple possible future trajectories given as input the past positions of the agents prior to time t, the scene information, and the static location of surrounding agents. To solve our main task we also propose some intermediate tasks as semantic segmentation and probability grid generation. In this chapter, we explain all the proposed model details.

4.1 Problem Formulation and Notation

We assume that all data is processed to obtain the positions and *ids* of all traffic participants across all the scenes and timesteps. A trajectory is defined as a sequence of x, y discrete positions with respect to time. $\mathbf{X_i^p} = \left[\mathbf{x}^{t-t_p}, \cdots, \mathbf{x}^{t-\Delta_t}, \mathbf{x}^t\right]$ represents an agent's past trajectory until upon time t, where Δ_t is the interval of time between two observations, $\mathbf{x} = [x, y] \in \mathbb{R}^2$ is the 2D discrete coordinate of the agent's position, and t_p is the size of past time window. At inference time, t represents the last observation of an agent's position. The ground truth future trajectory of an agent is represented by $\mathbf{Y_i^f} = \left[\mathbf{x}^{t+\Delta_t}, \cdots, \mathbf{x}^{t+t_f}\right]$ ranging from time $(t+\Delta_t)$ to $(t+t_f)$, where t_f is the length of the future horizon window. The trajectories can be considered a structured output comprising multiple values, where the elements have an important relationship.

Frame of Reference: The frame of reference is centered at each agent being predicted at time t. For each trajectory, we consider the x and y discrete positions of the agent at time t as the center of the grid, adapting the past and future positions according to this reference. Trajectory, scene, and surrounding agents follow the same frame of reference.

Trajectory Representation with Grids (τ^p): To keep the spatio-temporal correspondence among all input information, we transform each trajectory $\mathbf{X_i^p}$ to a grid representation. For each trajectory $\mathbf{X_i^p}$ we generate a $N \times N$ one-hot grid with t_p number of channels. Each grid

channel is populated according to the trajectory x and y discrete positions at each time step.

Scene Representation with Grids (S_t): The scene is represented by a $N \times N$ grid with three channels. Each grid position stores the RGB pixel values of a BEV map of the environment. The scene and trajectory are represented using the same frame of reference.

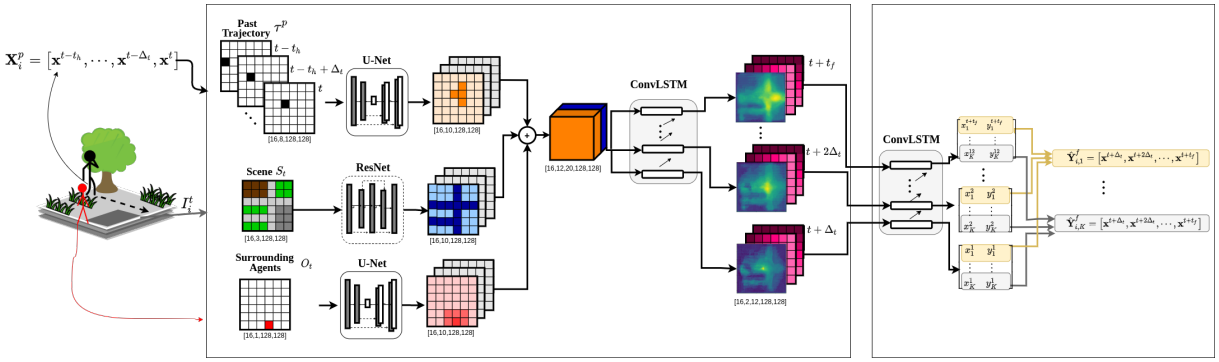
Surrounding Agents Representation with Grids (O_t): We create a grid, in the same reference frame, for representing the position of each surrounding agent occupying the scene at the same time as the agent being predicted. Here we consider the other agents as static obstacles and do not consider their motion.

Given an agent's past trajectory τ^p, surrounding agents current location (O_t), and scene information S_t, we want to predict $\hat{\mathbf{Y}}_i^f = \left[\hat{Y}_{i,1}^f, \cdots, \hat{Y}_{i,K}^f\right]$, with $k \in \{1,2,...,K\}$, where $\hat{\mathbf{Y}}_{i,k}^f = \left[\mathbf{x}^{t+\Delta_t}, \mathbf{x}^{t+2\Delta_t}, \cdots, \mathbf{x}^{t+t_f}\right]$.

4.2 Model

Our network, Fig. 12, comprises two modules. The first one (probability grid generation) takes as input trajectory (τ^p), surrounding agents (O_t), and scene (S_t), and generates the probability grids. These grids store the probability of the agent being in each cell at a determined time-step. The second module (trajectory generating) generates K trajectories ($\hat{\mathbf{Y}}_{i,k}^f$) from the probability grids.

Figure 12 – **Proposed model for scene compliant trajectory forecasting with spatial grids**. A U-Net with skip connections processes the trajectory and surrounding agents grids, and a ResNet processes the scene grid. The concatenation of the outputs from both U-Nets and ResNet are used as input to the ConvLSTM model that outputs the probability grids. The sampling module uses the generated probability grids, and the CoordConv module, to create K possible trajectories ($\hat{\mathbf{Y}}_i^f$) for each target agent i. Numbers inside brackets below figures are either [batch size, number of channels, height, width] or [batch size, number of time steps, number of channels, height, width].



Source: Elaborated by the author.

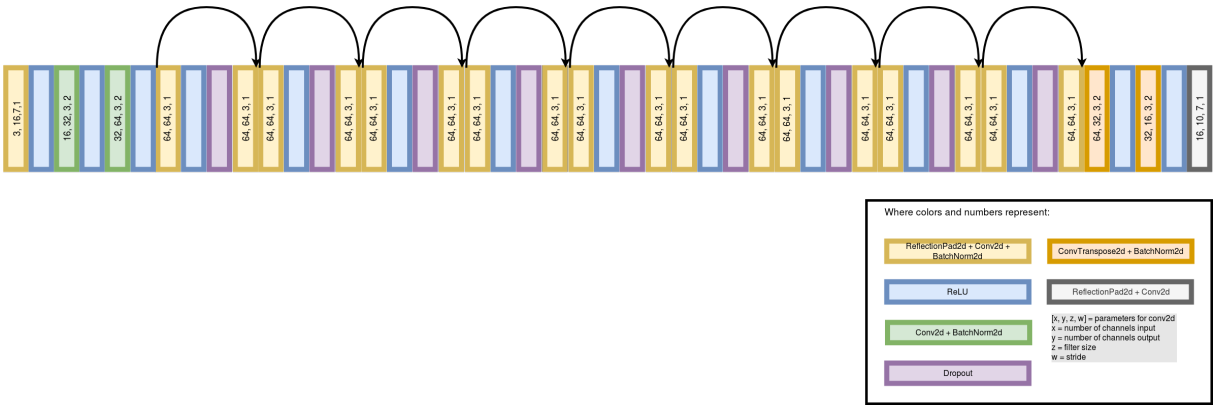
4.3 Probability Grid Generation ($S_t, \tau^p, O_t \rightarrow G^f$)

The Probability Grid Generation module is responsible for encoding the scene (S_t), past trajectory (τ^p), and surrounding agents (O_t) and generating the probability grids G^f. The grids G^f contain the information regarding the probability of the target agent occupying each grid cell at each predicted time step. We describe now each encoding and the process for generating such grids.

4.3.1 Scene Context Encoding

The BEV scene is a grid with depth representing the RGB color channels. A ResNet (He *et al.*, 2016) based encoder-decoder was used to process the scene. Such networks preserve specific features while also reasoning about the global features of the scene. ResNet is also useful to train because it can transform the loss search space into a smother function (LI *et al.*, 2018). We pre-trained the ResNet (Fig. 13) to semantically segment the satellites images from the International Society for Photogrammetry and Remote Sensing (ISPRS) (ROTTENSTEINER *et al.*, 2012) using the Potsdam dataset. Such Dataset provides 38 different satellite scenes with semantic labels for 6 classes: impervious surfaces, buildings, low vegetation, tree, car, and clutter/background. As the dataset does not have a specific label for sidewalks, and we consider that is a piece of important information for our path prediction problem we hand-labeled some images from the training set of SDD (ROBICQUET *et al.*, 2016) dataset and further trained the model to semantically segment such images. We used cross-entropy loss, Eq. 4.1, to train the semantic segmentation network.

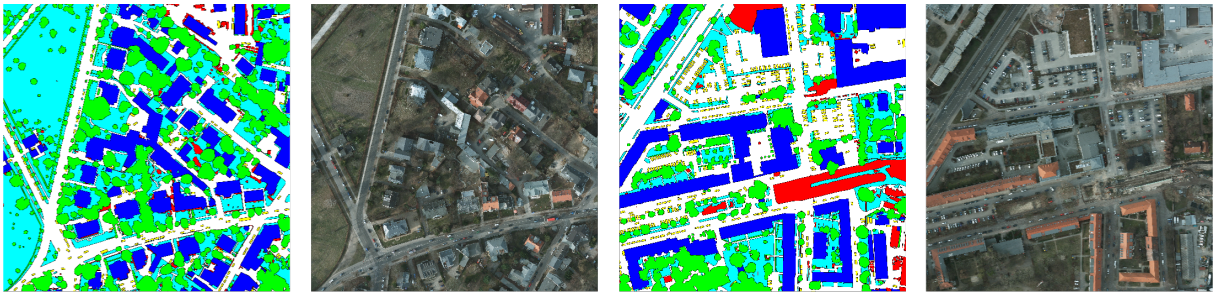
Figure 13 – ResNet-based encoder-decoder architecture used in this proposed approach.



Source: Elaborated by the author.

$$CrossEntropyLoss(x, class) = -log\left(\frac{exp(x[class])}{\sum_j exp(x[j])}\right) = -x[class] + log\left(\sum_j exp(x[j])\right) \tag{4.1}$$

Figure 14 – ISPRS Potsdam Dataset (ROTTENSTEINER *et al.*, 2012)

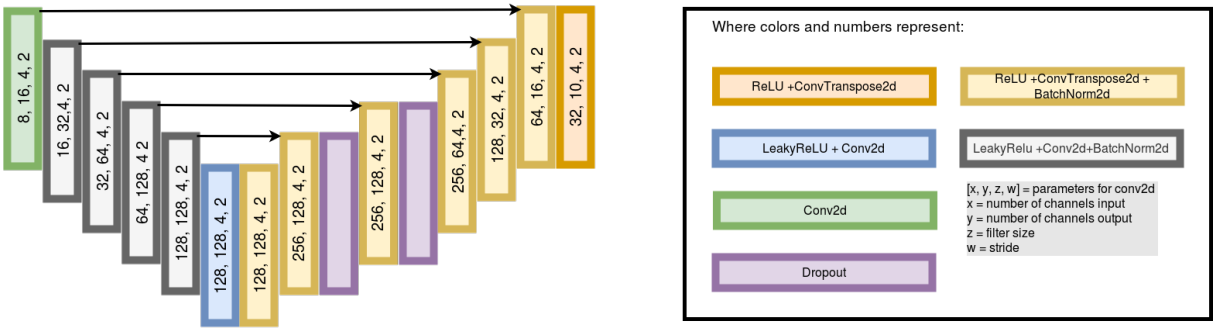


Source: (ROTTENSTEINER *et al.*, 2012).

4.3.2 Past Trajectory Encoding

Both trajectory and scene are represented in grids. Whether for the scene we use grids with depth representing the color channels, we need t_p grids to represent the pedestrian past trajectory. For each past trajectory x, y position, we generate a one-hot grid. The past trajectory grid is processed by a U-Net (RONNEBERGER; FISCHER; BROX, 2015) with skip connections. The choice of such architecture was made because as we are forecasting slow and fast-moving agents in the same network we had to make sure all grid positions would be convoluted to encode the trajectory. As stated in related work the most recent positions of an agent have more influence in his/her future positions than older positions. Such past trajectory information is commonly useful to learn the orientation as in most cases the agents do not return to positions they have already been to, except the cases where they are stopped, where in fact, they do not leave the past position.

Figure 15 – U-Net architecture used in this proposed approach.



Source: Elaborated by the author.

4.3.3 Surrounding Agents Encoding

The other agents on the scene are also represented in a grid. The grid is populated according to each surrounding agent's coordinates at time t. The encoding procedure is similar to the one presented in Sec. 4.3.2 using a U-Net.

4.3.4 Time-expanded Probability Grids

The prediction of long-term future trajectories tends to be more challenging and strongly relate to the scene. Such future trajectories can be more robust and comprise curves to avoid obstacles. Because of this extra challenge we use ConvLSTMs instead of the simple convolution networks. Such architecture can learn more robust trajectories by deciding which features it should use from the prior LSTM cell and the current input, learning what it can forget or remember to reason about the future.

We use weighted cross-entropy loss (4.2) to train the grid generation (first module in Fig. 12). Our model outputs t_f grids with probable agent future positions, each grid corresponds to a specific predicted time-step. The importance of using weighted cross-entropy, instead of cross-entropy, is in the view that the number of populated cells is lower than the empty cells, therefore, by using weighted cross entropy we can give a large weight for the populated cells.

$$WeightedCrossEntropy(x, class) = weight[class]\left(-x[class] + log(\sum_j exp(x[j])\right) \tag{4.2}$$

4.4 Trajectory Generation ($G^f \rightarrow \hat{\mathbf{Y}}_i^f$)

To compare our method with prior approaches we have to extract diverse and cohesive trajectories from the probability grids computed by the first module. The trajectory generation step, the second module in Fig. 12, receives as input the grid maps and outputs K predicted trajectories, $\hat{\mathbf{Y}}_{i,k}^f$ with $k \in \{1, 2, ..., K\}$. We have also concatenated a CoordConv (LIU *et al.*, 2018) to the probability grid to help with the mapping back from grid to coordinates. The ConvLSTM used for the trajectory generation has the same hyperparameters as the ConvLSTM used for grid generation, however, the last layer was replaced for a fully connected layer. The network was trained using Variety loss (GUPTA *et al.*, 2018; LEE *et al.*, 2017; SADEGHIAN *et al.*, 2019) with mADE loss. Variety Loss, also know as best-f-k loss, measures the distances of all predicted trajectories to the GT trajectory, and only the best loss is backward during training. The k used in most approaches are 5, 10, or 20. Prior work has reported that by using such loss they were able to reach more diverse sets of trajectories instead of predicting trajectories that only average among the dataset trajectories.

$$mADE = \min_{k \in \{1,2,...,K\}} \frac{1}{T} \sum_{t=1}^{T} \left\| \mathbf{Y}^{f,t} - \hat{\mathbf{Y}}_{(k)}^{f,t} \right\|_2, \tag{4.3}$$

where $\mathbf{Y}^{f,t}$ is the GT trajectory position at time t, and $\hat{\mathbf{Y}}_{(k)}^{f,t}$ is the position of predicted trajectory k at time t.

4.5 Implementation Details

All implementations were made using PyTorch. We applied a random rotation to all grids during training and we randomly shuffled the batches at every epoch. The images are normalized according to the ImageNet mean and standard deviation. We used early stopping with Adam optimizer (KINGMA; BA, 2014) for both modules. We used a grid size of $N = 128$ because it allows us to fit most of the trajectories after downsampling them by a factor of ten, $t_h = 8$ (3.2 secs), and $t_f = 12$ (4.8 secs) as previously used in related work (SADEGHIAN *et al.*, 2019). The U-Net and ResNet architectures were adapted from Isola *et al.* (2017). We used seven blocks of U-Net with skip connections and ResNet with nine blocks. All convolutions in both architectures follow a Convolution-BatchNorm-ReLU or a Convolution-BatchNorm-Dropout-ReLU sequence, with stride 2. The convolutions down-sample and up-sample by a factor of two. The weights were initialized from a Gaussian distribution with a zero mean and standard deviation of 0.02. All ReLUs used in downsample are LeakyReLUs with slope = 0.2. The dropout rate was 0.5. The ConvLSTM architecture has one layer with input dimension of 20, hidden state with 16 channels, and kernel size of $(11, 11)$.

CHAPTER

5

RESULTS AND DISCUSSION

In this chapter we quantitatively compare our results with different baselines, and ablations. We provide qualitative results to illustrate multiple scenarios and observed behaviors. We believe such qualitative results help us to have a grasp of what is happening inside the network. To conclude the chapter we discuss cases where future work can improve the performance of the current proposed approach.

5.1 Metrics and Data

We conducted experiments on SDD (ROBICQUET *et al.*, 2016) dataset to quantitatively and qualitatively evaluate our approach. According to Hastie and Friedman (2010), if there are enough data available, the best approach is to randomly divide the dataset into three parts: train, validation, and test. Where the training set is used to fit the weights in the model, the validation set is used to select the right architecture and hyperparameter configuration (also known as Model Selection), and the test set is used to evaluate the prediction error in a never seen data, i .e. generalization error. We use the standard train, validation, and test split available in TrajNet[1]. The SDD dataset comprises different scenarios captured by a drone's camera. For each scene, several trajectories are pixel-wise labeled. Those trajectories comprise diverse agents (pedestrians, cyclists, skaters, cars, buses, and carts). Lost positions were excluded from the sets of trajectories, and a new ID was created when the agent re-appeared in the scene.

5.2 Performance measure

Given the GT trajectory $\mathbf{Y_i^f}$ and the K predicted trajectories $\hat{\mathbf{Y}}_{i,k}^{f}$ with $k \in \{1,2,...,K\}$, we compute three metrics to evaluate the proposed method.

[1] <http://trajnet.stanford.edu/>

Table 5 – SDD dataset information.

Scenes	Videos	Bicyclist	Pedestrian	Skateboarder	Cart	Car	Bus
gates	9	51.94	43.36	2.55	0.29	1.08	0.78
little	4	56.04	42.46	0.67	0	0.17	0.67
nexus	12	4.22	64.02	0.60	0.40	29.51	1.25
coupa	4	18.89	80.61	0.17	0.17	0.17	0
bookstore	7	32.89	63.94	1.63	0.34	0.83	0.37
deathCircle	5	56.30	33.13	2.33	3.10	4.71	0.42
quad	4	12.50	87.50	0	0	0	0
hyang	15	27.68	70.01	1.29	0.43	0.50	0.09

Minimum Average Displacement Error (mADE): Minimum value among the average distance between each predicted trajectory and GT.

$$mADE = \min_{k \in \{1,2,\ldots,K\}} \frac{1}{T} \sum_{t=1}^{T} \left\| \mathbf{Y}^{f,t} - \hat{\mathbf{Y}}_{(k)}^{f,t} \right\|_2, \tag{5.1}$$

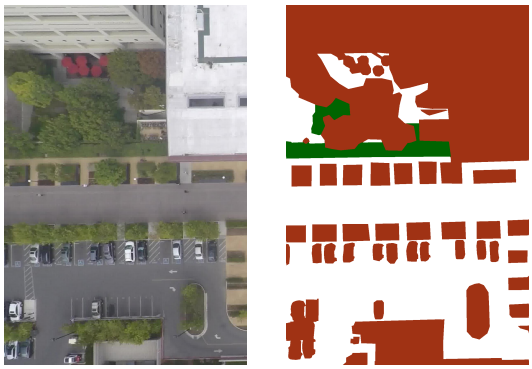
where $\mathbf{Y}^{f,t}$ is the GT trajectory position at time t, $\hat{\mathbf{Y}}_{(k)}^{f,t}$ is the position of predicted trajectory k at time t, and $T = t + t_f$.

Minimum Final Displacement Error (mFDE): Minimum final displacement error between each predicted trajectory and the GT final point T, where $T = t + t_f$.

$$mFDE = \min_{k \in \{1,2,\ldots,K\}} \left\| \mathbf{Y}^{f,T} - \hat{\mathbf{Y}}_{(k)}^{f,T} \right\|_2, \tag{5.2}$$

Correspondence to Scene: Efficiently evaluating multi-modal trajectories is still an open problem. For critical applications, like self-driving cars, learning several modes is important because the car should be able to estimate different plausible scenarios. This makes trajectory forecasting a challenging task because the observations stored in datasets are typically uni-modal. Most of the current approaches using best-of-K (or variety loss) compare each one of the K predicted trajectories with the GT trajectory and consider as the result the predicted path that achieved the closest distance to the GT. A current problem of such an approach is that it only takes into consideration the accuracy of the trajectory that most closely matched the GT, neglecting the other trajectories. To evaluate the performance of our K predicted trajectories, we propose the Correspondence to Scene metric that gives us the percentage of predicted trajectory points that lie on paths, terrain, and obstacles. For each image in the testing set, we hand-labeled, as depicted in Fig.16, the pixels into path (sidewalk, street), terrain (grass, ground), or obstacle (trees, cars, buildings). As such trajectories should not pass through obstacles, the CS metric accesses the precision of all the estimated trajectories.

Figure 16 – **Illustration of manually accomplished semantic labeling.** (a) BEV image (Nexus 5) from SDD dataset, (b) Semantic labeled image, obstacle (red), terrain (green), and path (white).



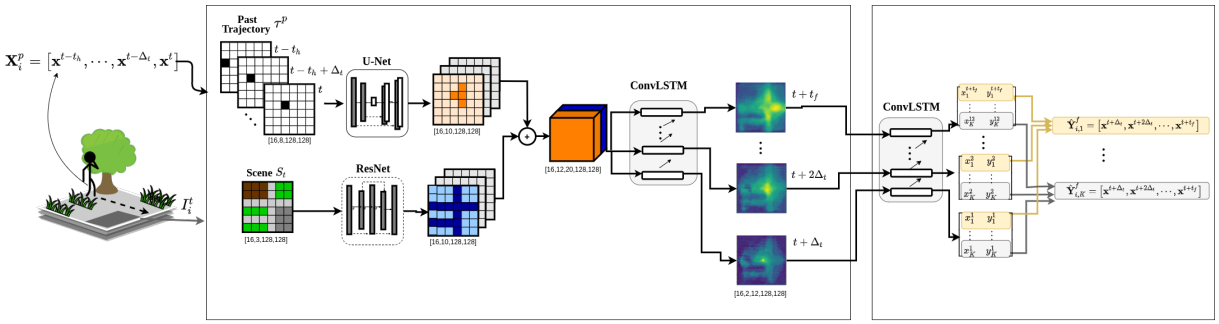
Source: left image (ROBICQUET *et al.*, 2016), and right image elaborated by the author.

5.3 Evaluation

We denominate SCPTSA-CC-PG our proposed model presented in Chapter 4. It uses Scene Context (SC), Past Trajectory (PT), Surrounding Agents (SA), and the Probability Grid (PG) with CoordConv (CC) layer. We use the same baselines and directly report the results from (SADEGHIAN *et al.*, 2019) in Tab. 6. The baselines used for comparison are: Social GAN (S-GAN) (GUPTA *et al.*, 2018): Generative Adversarial Network (GAN) based LSTM encoder-decoder; SoPhie (SADEGHIAN *et al.*, 2019): attentive GAN with social and physical attention mechanisms; Desire (LEE *et al.*, 2017): Inverse Optimal Control (IOC) RNN based encoder-decoder; Social LSTM (S-LSTM) (ALAHI *et al.*, 2016): LSTMs with social pooling layer; CAR-NET (SADEGHIAN *et al.*, 2018): attentive RNN; Social Forces (SF) (Yamaguchi *et al.*, 2011): tracking-based behavioral model; and a Linear Regressor (LR). We also compare the results with two variations of the proposed model. The first variation SCPT-PG (Fig. 17) is the proposed model without Surrounding Agents and CoordConv. The second variation SCPT (Fig. 18), is the first variation without the probability grid generation, i .e. the concatenation of processed scene and past trajectory is directly given to the trajectory sampling. All ablations inputs follow the format presented in Sec. 4.1

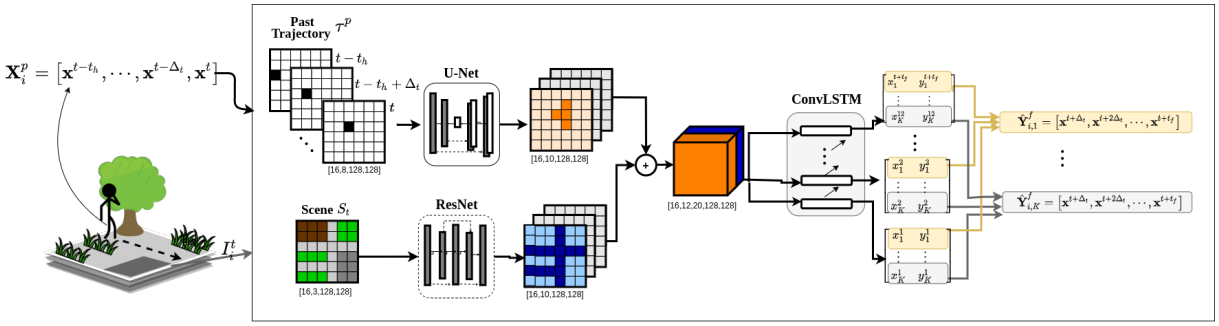
The SCPT performed better on ADE and FDE metrics when compared to SCPT-PG, the opposite happens when comparing the Correspondence to Scene (CS) metrics (Tab. 7), where SCPT-PG outperformed SCPT. The SCPTSA-CC-PG achieved consistent results in both trajectory distance and compliance to scene metrics, therefore bridging the gap between SCPT and SCPT-PG models. We believe the intermediate grid generation step forces the predicted trajectory to have better compliance with the scene, as both methods that used the grid generation achieved better CS results in comparison with the model that did not use the grid generation (SCPT). The grid generation step is also useful to have some grasp of interpretability of the model learning process, as it gives a qualitative understanding of the extracted knowledge from the input information. One observation from the CS table is that some points in the GT lie

Figure 17 – **Proposed model without surrounding agents and CoordConv layer**. U-Net with skip connections processes the trajectory grid and ResNet processes the scene grid. The concatenation of the outputs from both U-Net and ResNet are used as input to the ConvLSTM model that outputs the probability grids. The sampling module uses the generated probability grids to create K trajectories ($\hat{\mathbf{Y}}_i^f$) for each target agent i. Numbers inside brackets below figures are either [batch size, number of channels, height, width] or [batch size, number of time steps, number of channels, height, width].



Source: Elaborated by the author.

Figure 18 – **Proposed model without surrounding agents, CoordConv layer, and intermediate generated probability grid**. U-Net with skip connections processes the trajectory grid and ResNet processes the scene grid. The concatenation of the outputs from both U-Net and ResNet are directly used as input to the ConvLSTM that creates K trajectories ($\hat{\mathbf{Y}}_i^f$) for each target agent i. Numbers inside brackets below figures are either [batch size, number of channels, height, width] or [batch size, number of time steps, number of channels, height, width].



Source: Elaborated by the author.

Table 6 – Quantitative comparative performance analysis of the proposed approach in pixels on Stanford Drone Dataset. Note, results indicate an improvement over relevant state-of-the-art approaches as measured by two commonly used metrics (ADE and FDE).

Method	S-GAN	Sophie	Desire	LR	SF	S-LSTM	CAR-NET	SCPTSA-CC-PG	SCPT-PG	SCPT
K	20	20	5	1	1	1	1	5	5	5
ADE↓ (pixels)	27.24	16.27	19.25	37.11	36.48	31.19	25.72	**14.35**	14.92	**14.35**
FDE↓ (pixels)	41.44	29.38	34.05	63.51	58.14	56.97	51.8	**26.41**	27.89	26.85

Table 7 – Quantitative performance results of forecasted trajectories as compared against the ground truth.

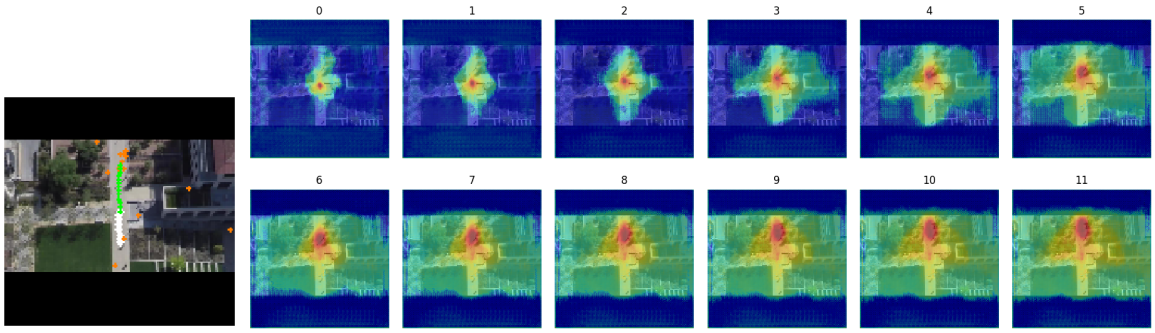
Method	**K**	**% on path** ↑	**% on terrain** ↓	**% on obstacles** ↓	**% out of the image**↓
SCPTSA-CC-PG	5	**86.48**	**5.64**	**7.86**	0.02
SCPT-PG	5	86.35	5.74	7.89	**0.01**
SCPT	5	84.52	7.15	8.20	0.1
GT	1	87.88	4.95	7.16	0.0

in obstacles. In the SDD dataset, there are scenarios where pedestrians are partially walking inside buildings, and as we hand-labeled buildings as obstacles, the trajectories' points will be computed as obstacles even if an indoor path existed. To fully understand the % on obstacles we have to look to both GT and ours results.

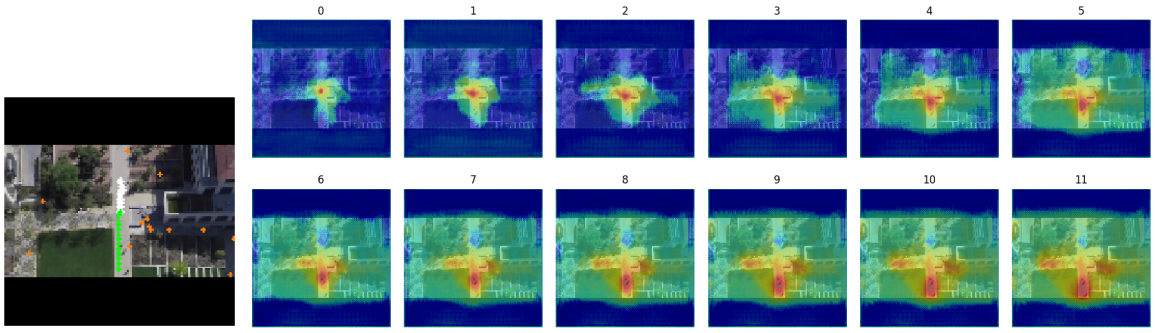
As explained in Sec. 4.3.4, a probability grid is generated for each future time step predicted. In both sets of images (Fig. 19 and Fig. 20) the scenes are similar but the agents' trajectories are different, therefore the generated probability grids exhibit different higher probability distributions. In most of the examples, the probabilities shift according to reasonable path preferences and agent movement orientation. Such probability grids are then fed into the trajectory generation module. Respective grids and trajectories are displayed in figures 21, 22, 23, and 24. In Fig. 21c the probabilities shift from the center of grid 0 towards the top of grid 11, indicating the higher probabilities for the straight path, the trajectories generated by the model (Fig. 21c) are all in the same orientation. In Fig. 22 the probabilities (Fig. 22c) and trajectories (Fig. 22b) lie in the same region. In Fig. 23c the probabilities shift from the center in grid 0 in direction to the paths at the bottom of the image. In Fig. 24a the agent is walking towards the top of the image, in the grid maps (Fig. 24c) the probabilities in grid 0 shift to the top of grid 11, also the generated trajectories follow the same behavior.

In Figures 25, 26, 30, 29, 28, 27, 33, 31, and 32 we depict some of the trajectories generated by the proposed model, and we discuss some points observed in those images. Usually by paying attention to the past trajectory (τ^p) a human would be able to figure it out the length

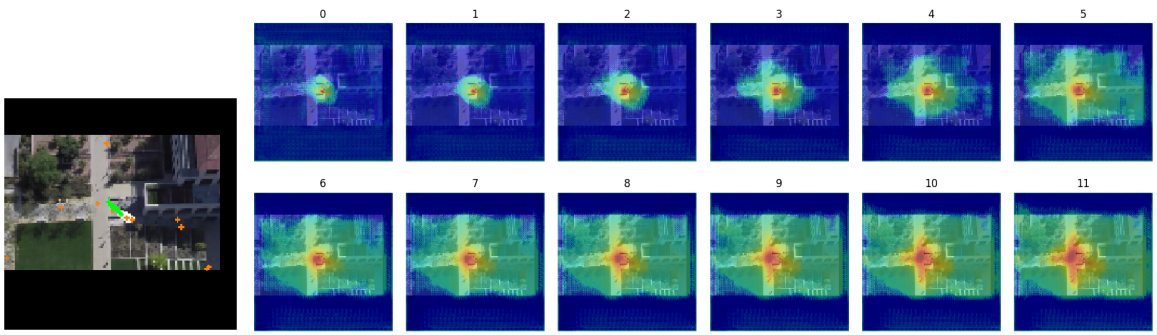
Figure 19 – **Qualitative result of probability grids generated by the proposed method**. The BEV images on the left column (a, c, and e) contain a scenario where an agent's past motion is represented in white, the ground truth future motion is represented in green, and surrounding agents are represented in orange. The set of images on the right column contains the $t_f = 12$ grids generated by the proposed approach according to the respective BEV image on the left. Each grid corresponds to one predicted time step. Each grid's cell stores the probability of the agent occupy that cell at that time-step. Closer to red higher the probability. Same scenes with different trajectories (a, c, e) generate different probability grids (b, d, f). In Figure a the agent is moving to the top of the image, the generated probability grids reflect that behavior. The opposite happens in Figure c, the agent is moving towards the bottom of the image, and in the probability grids (d) the probabilities shift from the center of the image towards three different paths. In figure (e) the agent is walking slower, and this behavior is reflected in the more central location of higher probabilities in the grid maps (f).



(a) (b)



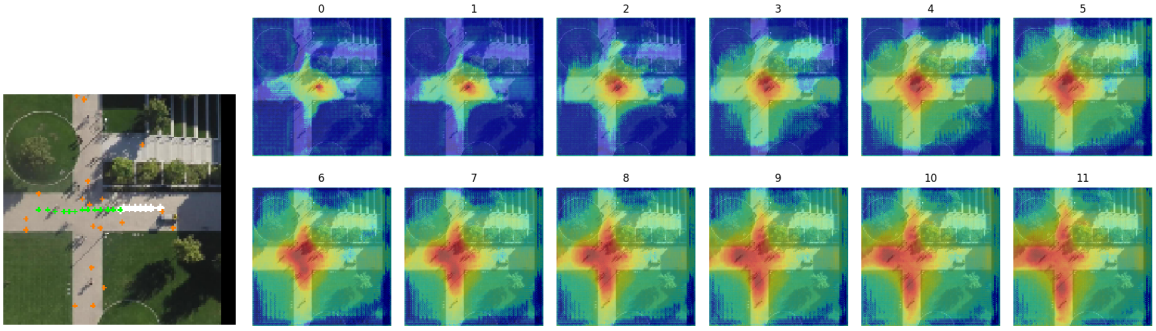
(c) (d)



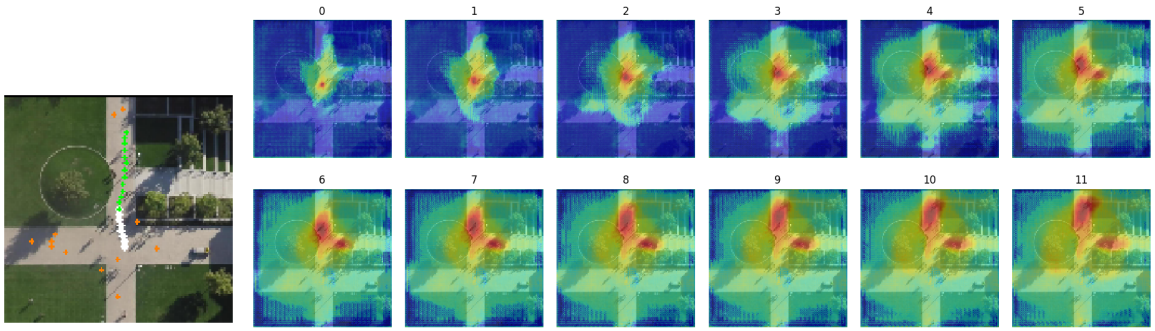
(e) (f)

Source: Elaborated by the author

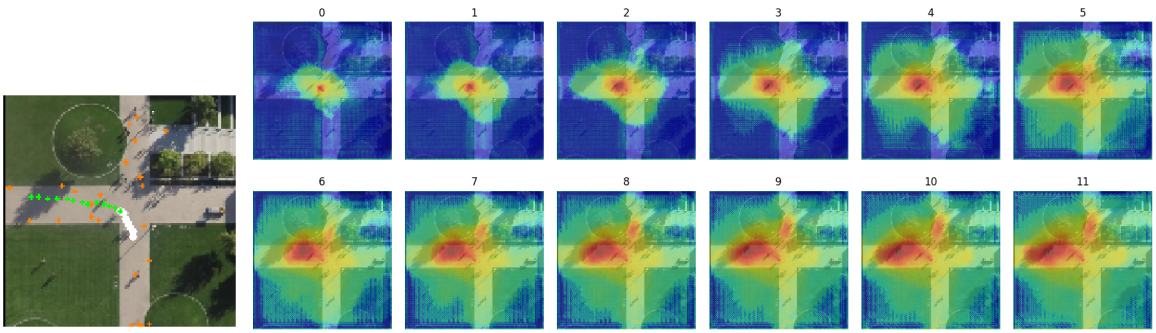
Figure 20 – **Qualitative result of probability grids generated by the proposed method**. The BEV images on the left column (a, c, and e) contain a scenario where an agent's past motion is represented in white, the ground truth future motion is represented in green, and surrounding agents are represented in orange. The set of images on the right column contains the $t_f = 12$ grids generated by the proposed approach according to the respective BEV image on the left. Each grid corresponds to one predicted time-step. Each grid's cell stores the probability of the agent occupy that cell at that time-step. Closer to red higher the probability. In the images (a, c, e) the scenes are similar but the agent's motion is diverse. In Fig. (a) the agent is moving towards the left (according to the viewer's perspective). In (b) the high probable cells comprise the clear paths seen in the image. In (c) the agent is moving towards the top, and in (d) the higher probabilities are towards the paths keep forward and turn to the left. In (e) the agent is moving to the left, and in (f) the probability is higher for the left path, with some small probability of keeping forward. By reasoning about the past trajectory, the model can distinguish different future trajectories even in similar scenes.



(a) (b)



(c) (d)

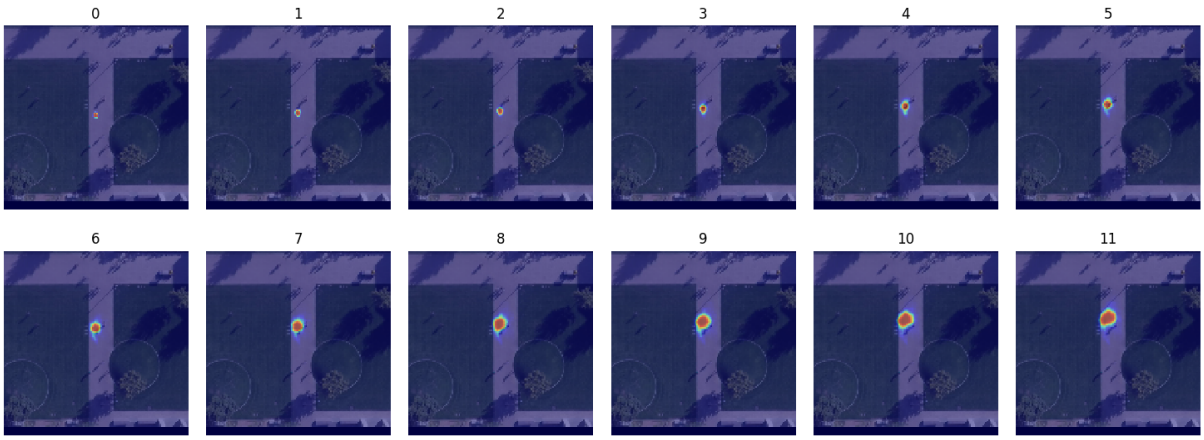


(e) (f)

Source: Elaborated by the author

Figure 21 – **Qualitative result of probability grids generated by the proposed method**. Fig. (a) Agent's past motion is represented in white, the ground truth future motion is represented in green, and orange points represent the surrounding agents, Fig. (b) contains the generated trajectories represented in light blue, dark blue, black, red, and magenta, Fig (c) contains the $t_f = 12$ grids generated by the proposed approach according to the respective image (a). Each grid corresponds to one predicted time-step. Each grid's cell stores the probability of the agent occupy that cell at that time-step, where closer to red higher the probability.

(a) (b)



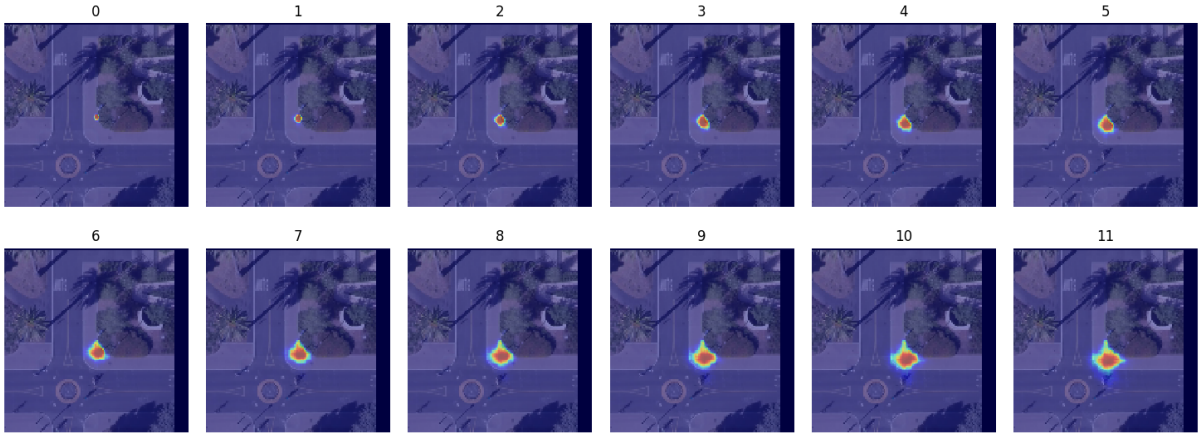
(c)

Source: Elaborated by the author

of the future path ($\mathbf{Y}^f$). In Fig. 25 we exemplify scenes with similar context with predicted trajectories that follow the expected future trajectory size. That means that even if we do not clearly give the class of the object as input to the model, the model was able to implicitly learn different agent's speed, i .e. estimate the future trajectory length based on the past trajectory (τ^p). Agent in Fig. 25 (f) is clearly moving faster then the agents in Figs. 25 (b, d). A challenge arise when the agent change his own speed only in the GT future path (Fig. 26), in these scenes the model fails in generating trajectories with lengths that match the GT trajectories. In some examples the predicted trajectories for a given agent have different lengths, that is somehow helpful when the agent decreases his motion speed in GT (Fig. 27).

Figure 22 – **Qualitative result of probability grids generated by the proposed method**. Fig. (a) Agent's past motion is represented in white, the ground truth future motion is represented in green, and orange points represent the surrounding agents, Fig. (b) contains the generated trajectories represented in light blue, dark blue, black, red, and magenta, Fig (c) contains the $t_f = 12$ grids generated by the proposed approach according to the respective image (a). Each grid corresponds to one predicted time-step. Each grid's cell stores the probability of the agent occupy that cell at that time-step, where closer to red higher the probability.

(a) (b)



(c)

Source: Elaborated by the author

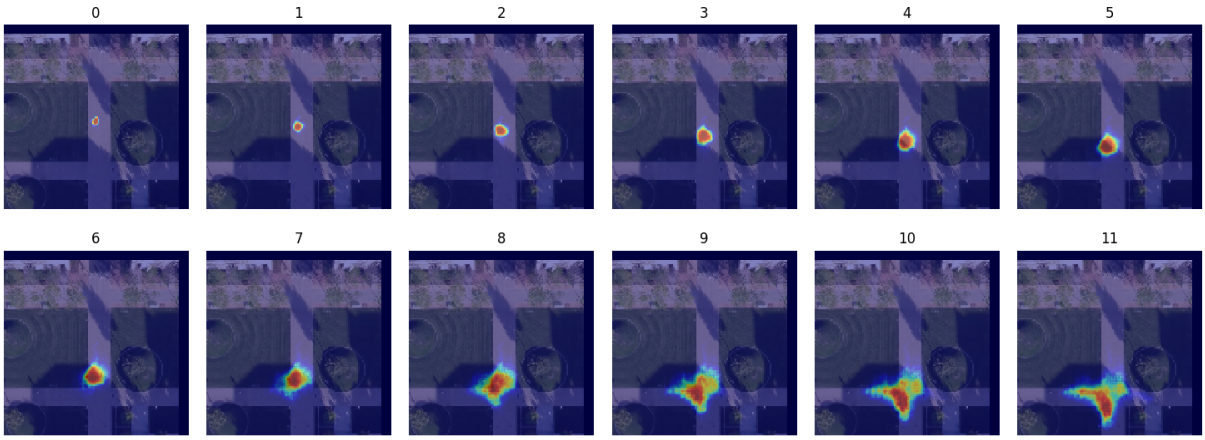
In some examples, the predicted trajectories are more spread or squeezed together. When the predicted trajectories are all squeezed together in the same direction usually there is a clear delimited path as shown in Fig. 28 (a, b, c, d, e, f). In other scenes, the generated trajectories comply with a larger range of possible orientations Fig. 28 (g, h, i, j, k, l).

In Fig. 29, the generated trajectories are in a range of orientations comprising almost all visibly delimited paths. Sometimes, as shown in Fig. 30, the past trajectories also bound the space of possible future trajectories, i .e. the orientation of the agent can enable the generated trajectories to be more assertive by focusing the predicted trajectories in more probable paths.

Papadimitriou, Lassarre and Yannis (2016) discuss that humans when crossing streets

Figure 23 – **Qualitative result of probability grids generated by the proposed method**. Fig. (a) Agent's past motion is represented in white, the ground truth future motion is represented in green, and orange points represent the surrounding agents, Fig. (b) contains the generated trajectories represented in light blue, dark blue, black, red, and magenta, Fig (c) contains the $t_f = 12$ grids generated by the proposed approach according to the respective image (a). Each grid corresponds to one predicted time-step. Each grid's cell store the probability of the agent occupies that cell at that time-step, where closer to red higher the probability.

(a) (b)



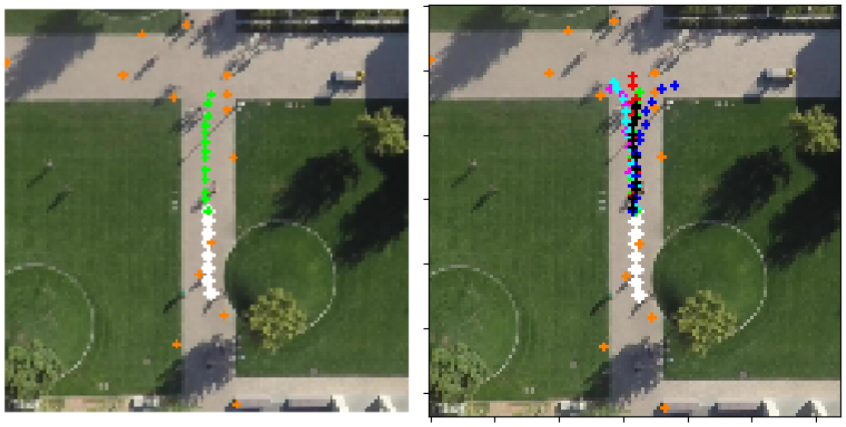
(c)

Source: Elaborated by the author

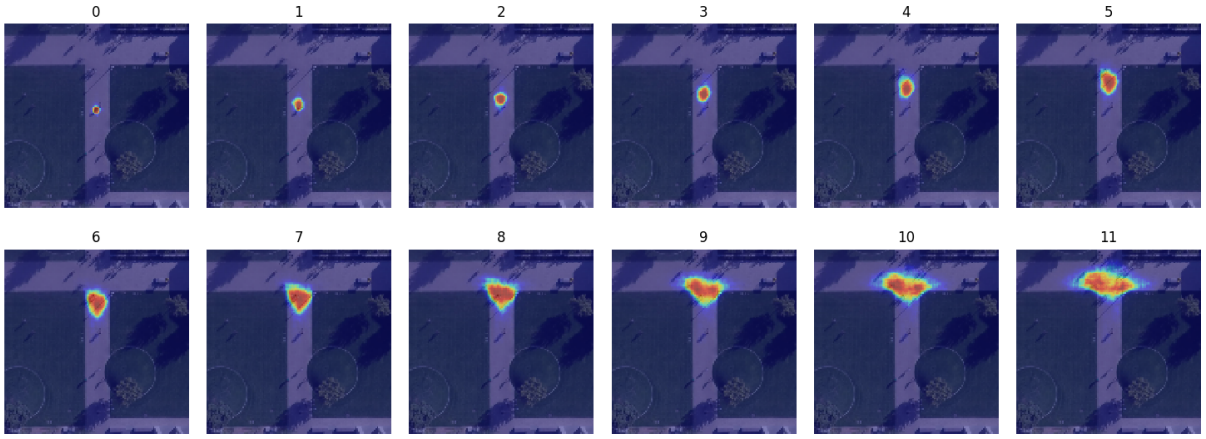
sometimes can present a pattern of crossing the street diagonally. Such behavior was seen in some examples (Fig. 31) especially when the agent was moving near a street. Nearly all predicted trajectories in Fig. 32 (a, b, c, d, e, f, g, h, i) were contouring the roundabout and were plausible extensions of the past trajectory, however in some examples, the roundabout behavior was not followed (Fig. 32 j, k, l), we believe that happens because in the dataset some agents exhibit a jaywalker behavior.

Scenarios where the agent is stopped in the τ^p, and starts to walk in the $\mathbf{Y}^f$, are still a challenge. Such a challenge arises due to the lack of information regarding orientation in τ^p. In Figure 33 (f, g, h, and i) the predicted trajectories did not match the GT trajectory. In some examples, as Fig. 33 (d), all predicted trajectories estimated that the agent would keep stopped,

Figure 24 – **Qualitative result of probability grids generated by the proposed method**. Fig. (a) Agent's past motion is represented in white, the ground truth future motion is represented in green, and orange points represent the surrounding agents, Fig. (b) contains the generated trajectories represented in light blue, dark blue, black, red, and magenta, Fig (c) contains the $t_f = 12$ grids generated by the proposed approach according to the respective image (a). Each grid corresponds to one predicted time-step. Each grid's cell stores the probability of the agent occupy that cell at that time-step, where closer to red higher the probability.



(a) (b)



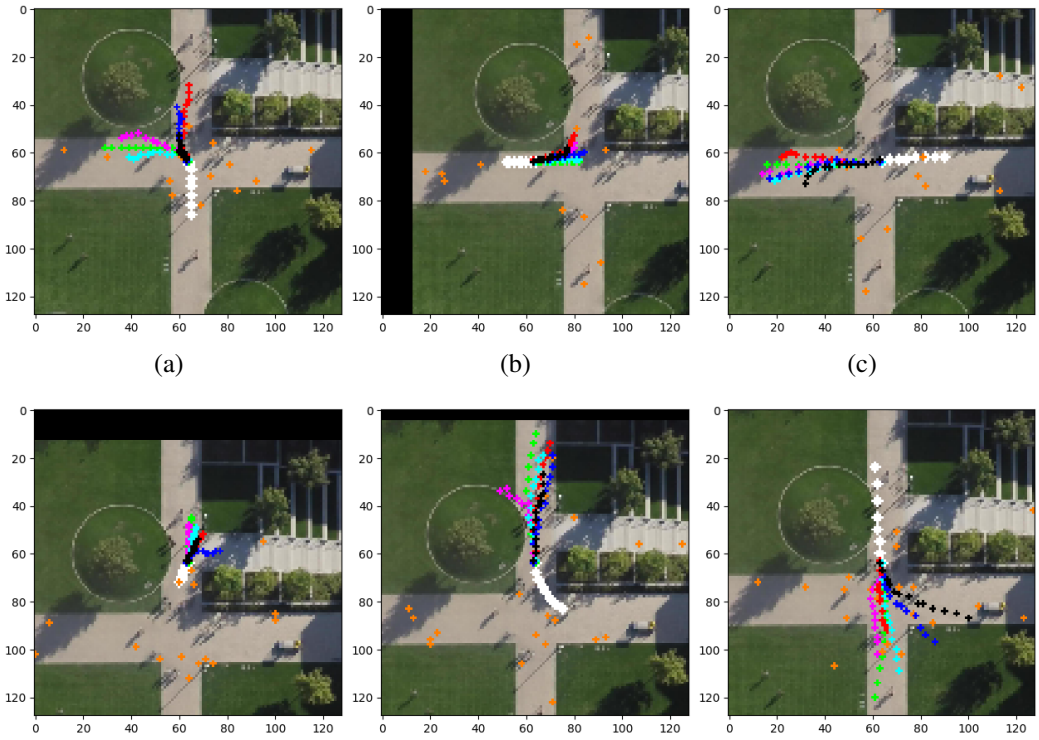
(c)

Source: Elaborated by the author

in the other images in Fig. 33, the predicted trajectories were split in static and guesses of future motion.

By analyzing the results we can notice that scene and past trajectory play a key role in the prediction. We could not reliably notice the aspects in which the usage of surrounding agent information improved the performance, but we can not discard that possibility. The generated probability grids and trajectories help us to get some reasoning about what is happening inside the network. In most of the examples, the forecasted trajectories are feasible continuations from the past trajectory and comply with the scene, therefore, achieving the objectives proposed by this Thesis.

Figure 25 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. An observation is that the future trajectories length varies according to the length of the past trajectory, this implicitly means that the model was able to figure out the different agents' speeds.

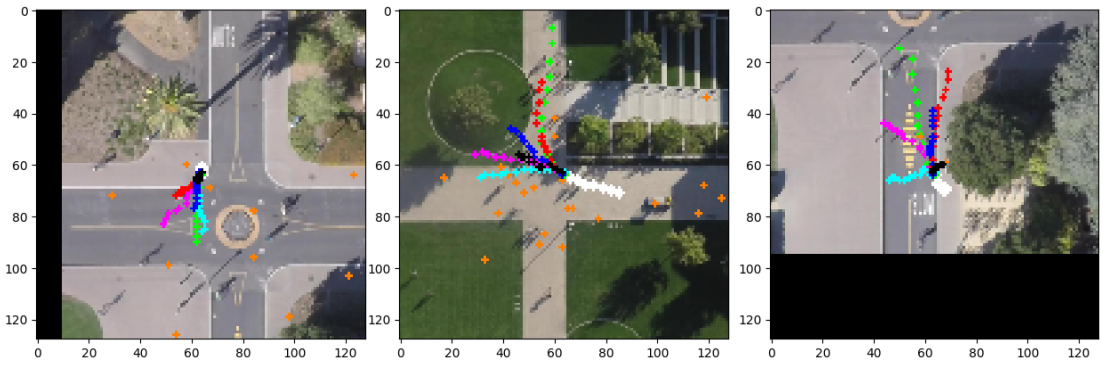


(a) (b) (c)

(d) (e) (f)

Source: Elaborated by the author

Figure 26 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. There are some examples where the speed in the future trajectory is faster than the motion observed in the past trajectory, in these scenes the model fails in predicting trajectories with lengths that match the GT trajectory.



(a) (b) (c)

Source: Elaborated by the author

Figure 27 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. In some examples the predicted trajectories have different lengths, this might be helpful in scenarios where the agent starts stopping or changes his motion speed.

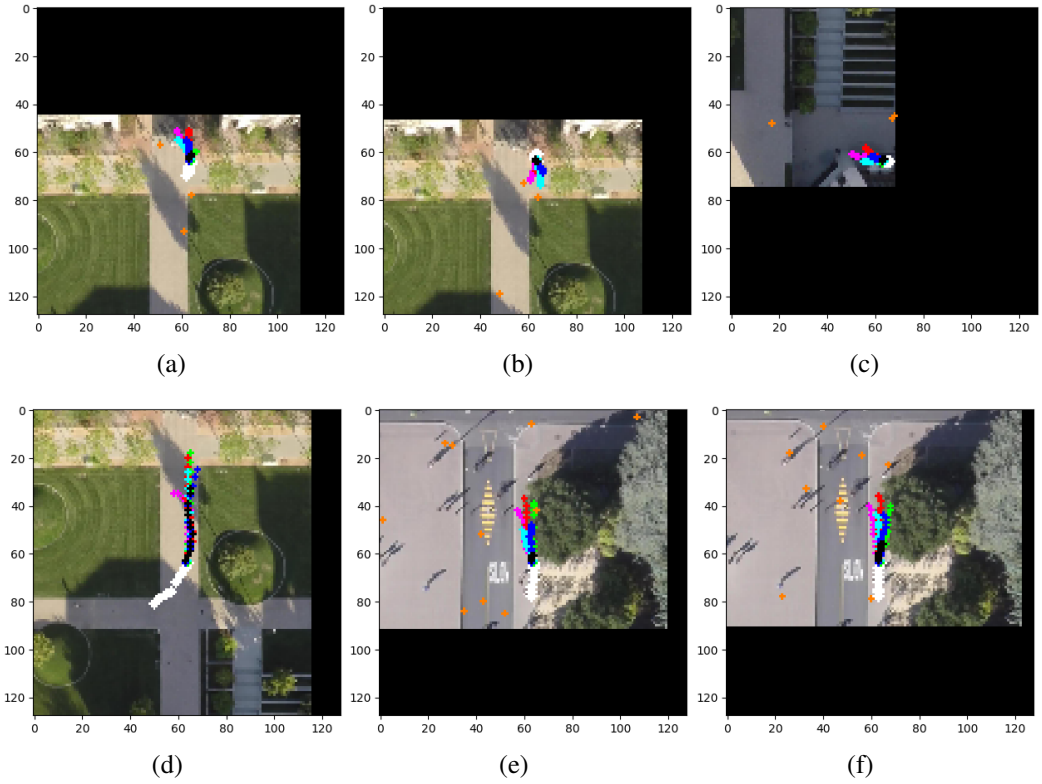
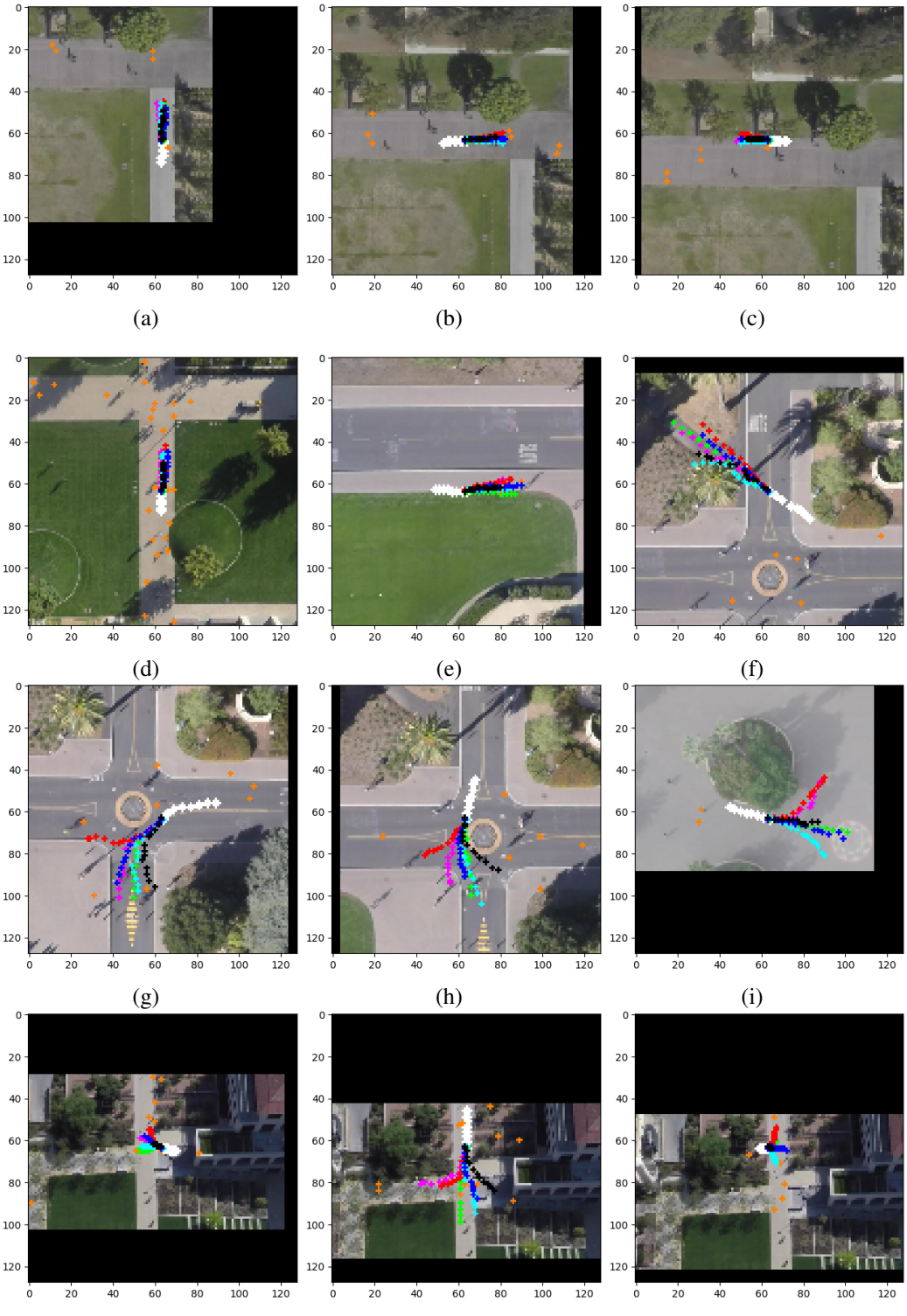


Figure 28 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. In the first and second rows of images, the trajectories are more squeezed together, we believe this behavior arises due to the visible path in the images. In the third and fourth rows, the predicted trajectories are more spread in open scenarios.



(a) (b) (c)

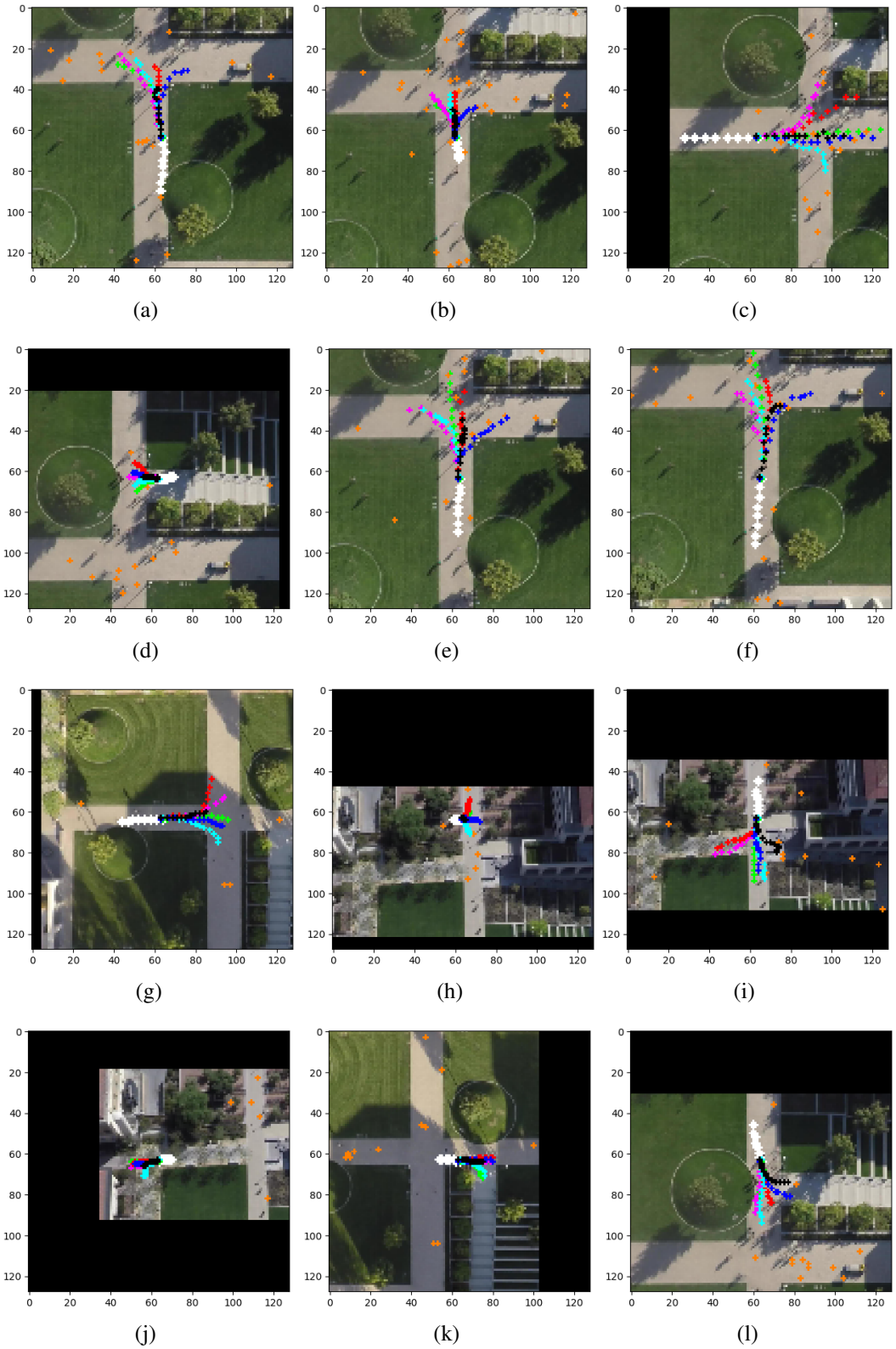
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

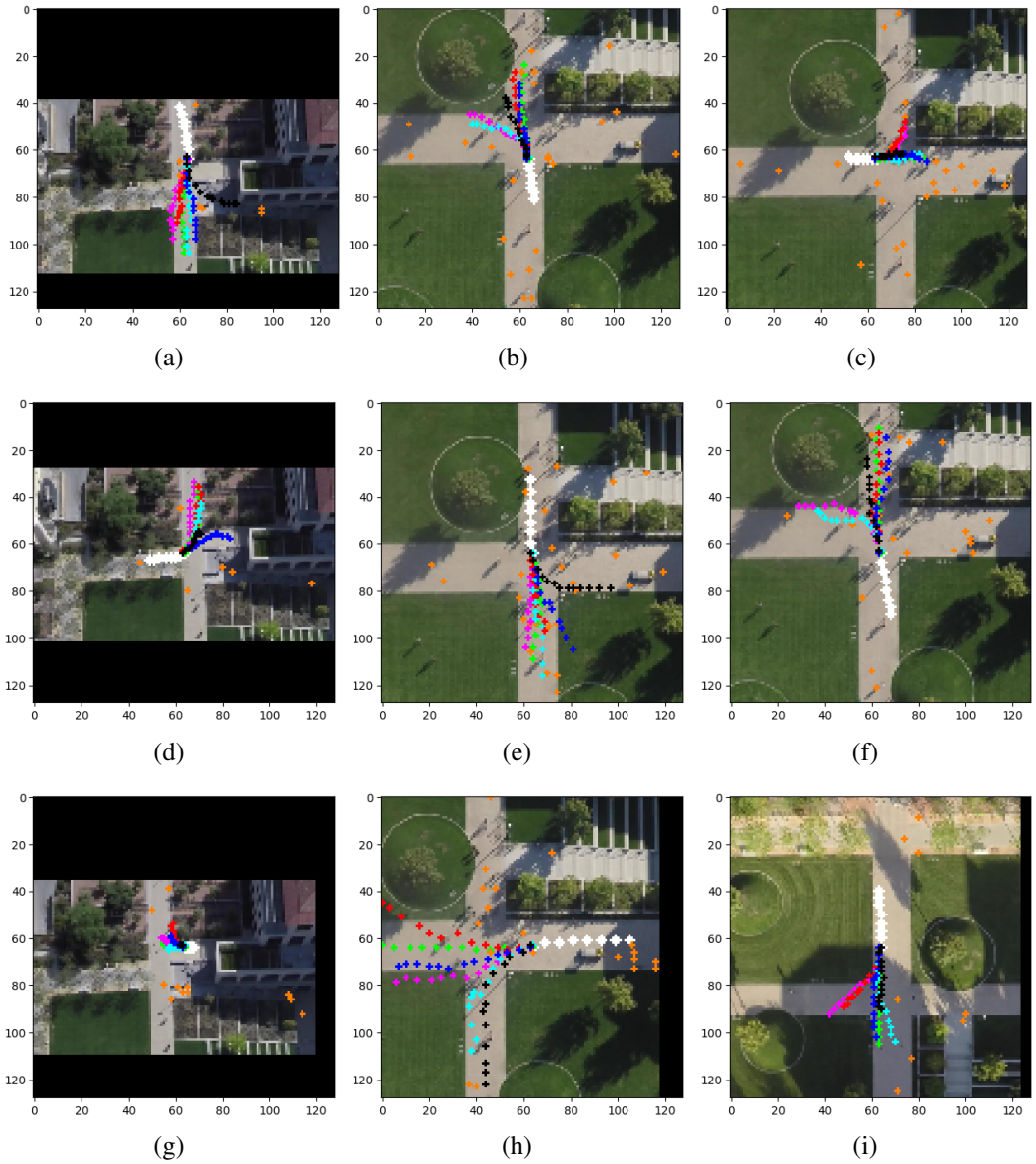
Source: Elaborated by the author

Figure 29 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. Examples where the proposed model generated trajectories that comply with the scene possible paths.



Source: Elaborated by the author

Figure 30 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. Examples where the proposed model generated trajectories that comply with the scene but not all possible scene paths, we believe the orientation of the past trajectory restricted the space of possible paths generated.



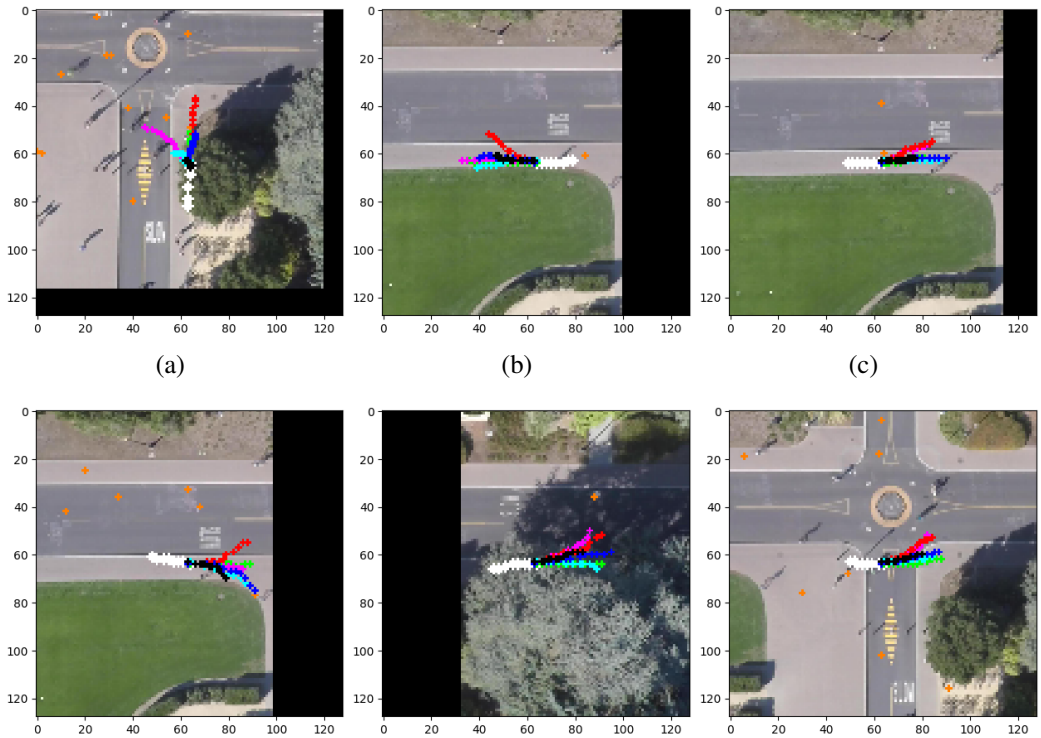
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Source: Elaborated by the author

Figure 31 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. Some predicted trajectories presented this pattern of crossing the street in a diagonal.

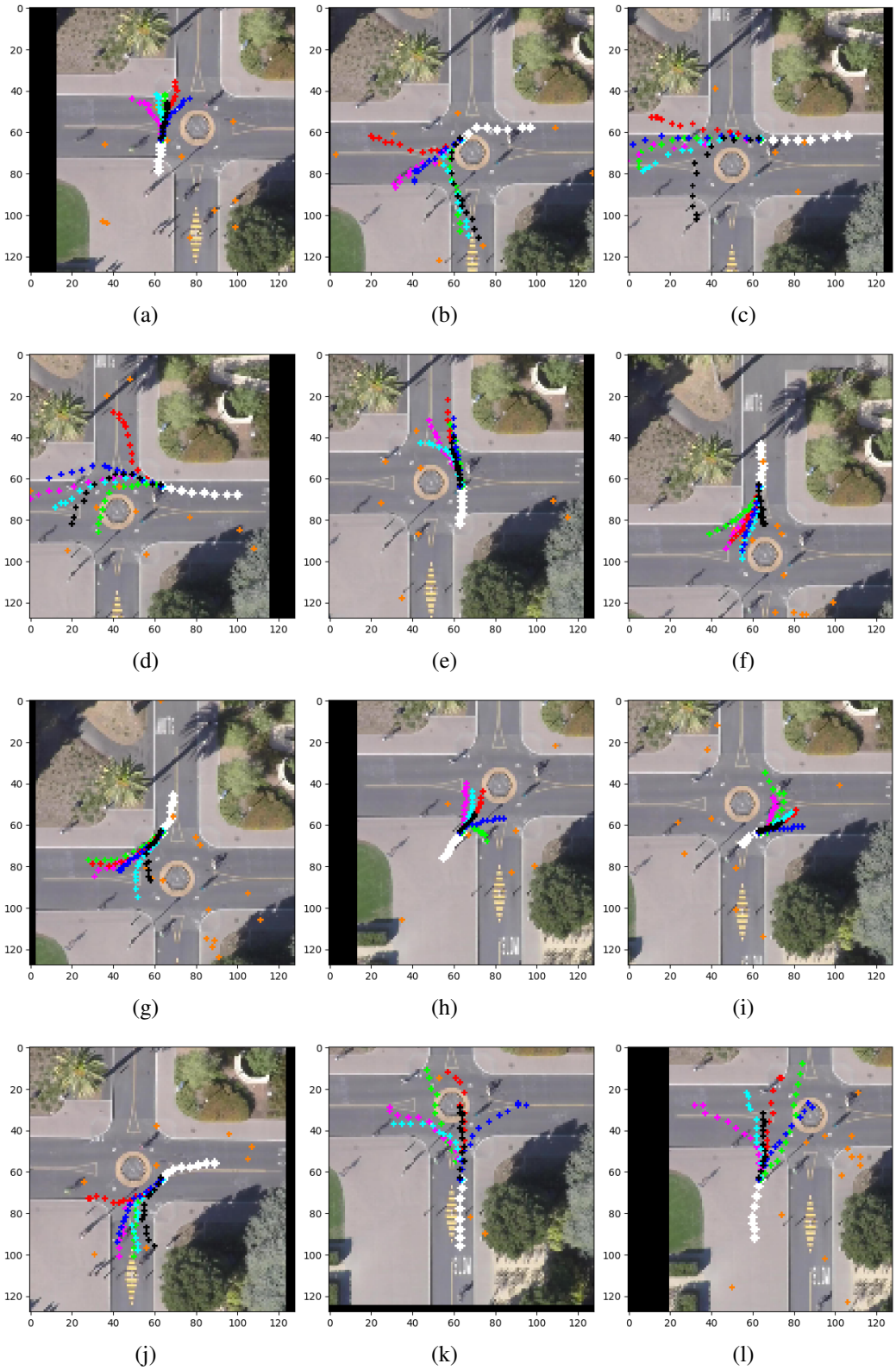


(a) (b) (c)

(d) (e) (f)

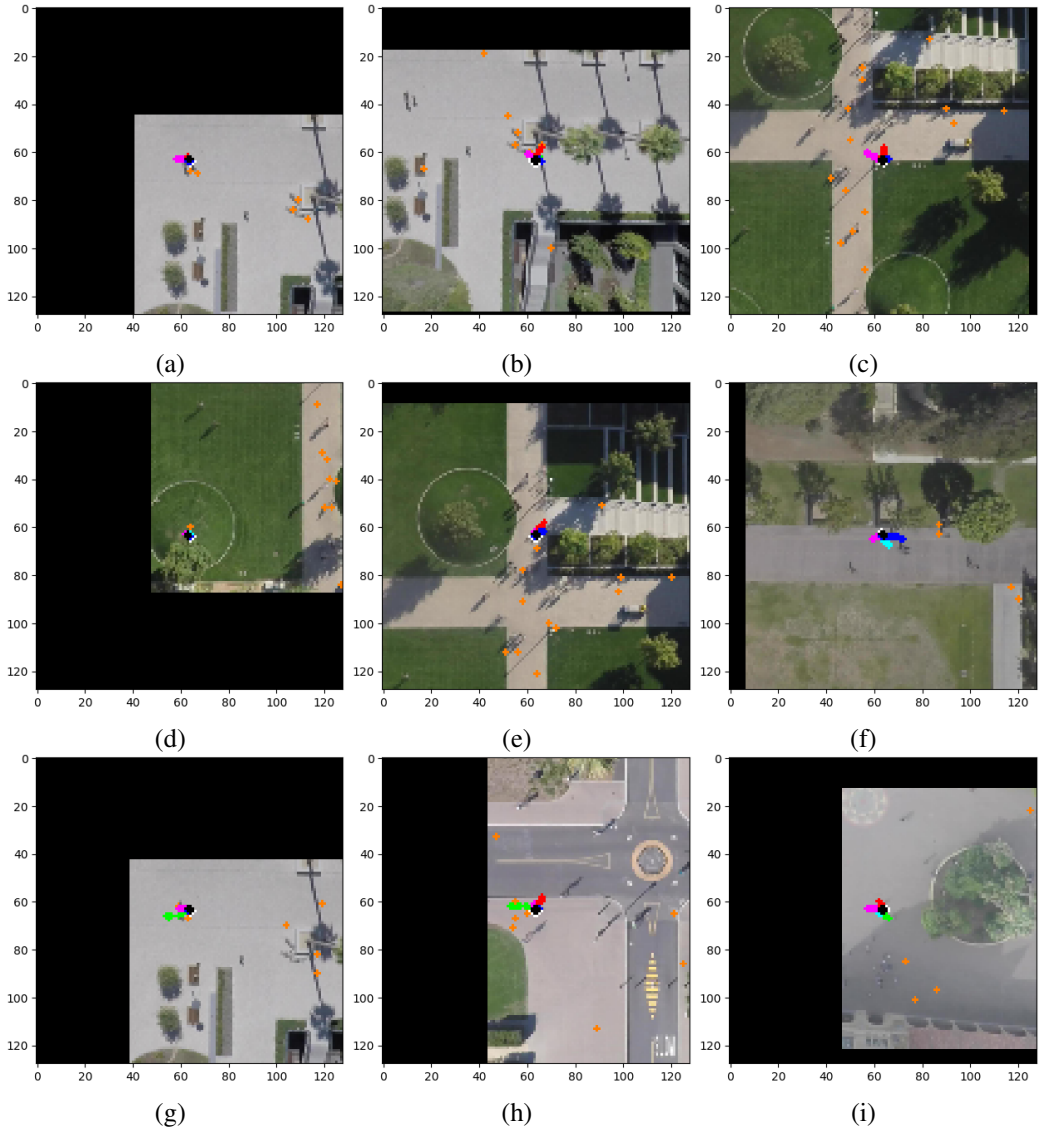
Source: Elaborated by the author

Figure 32 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. Most of the predicted trajectories in roundabouts exhibit a contouring behavior, however as shown in Fig. (j, k, and l) the dataset had examples of jaywalker behavior.



Source: Elaborated by the author

Figure 33 – **Qualitative results of the proposed method**. White represents past trajectory, green represents future trajectory (GT), orange surrounding agents, and other colors (light blue, dark blue, black, red, and magenta) represent the five predicted trajectories. When an agent is stopped in the past trajectory (τ^p) the model is not able to retrieve the orientation, therefore, the agent can decide to start moving in any possible direction. Predicting future trajectories without a grasp of the possible orientation seems to be a challenge. In some examples, all trajectories estimated that the agent would keep stopped (d), and in the other examples, the model tried to make some guesses of possible future motion.



Source: Elaborated by the author

5.4 Final Considerations and Limitations

We believe the model SCPTSA-CC-PG helped in bridging the gap between the models SCPT and SCPT-PG, therefore, reaching results that improve prior work. The effect of using surrounding agents' information could not be qualitatively seen in the generate probability maps, however, we cannot discard that the model was able to learn something from it. Challenges arise when the agent is stopped (in τ^p) and decide to move (in $\mathbf{Y}^f$) because of the lack of information on the direction of motion, which makes the number of possible orientations high, therefore five guesses are not enough to cover all possible orientations. The model also lacks in finding the correct length of future trajectory when the change in velocity is only performed in GT. Patterns for moving around roundabouts could be seen in some examples and disregarded in others as the data contained jaywalker behavior. In general, the proposed approach was able to generate diverse trajectories that comply with past trajectories and scenes. Dealing with trajectories represented in image space is a non-trivial task as the size of the grid directly implies the maximum trajectory size that can be represented in such a structure.

CHAPTER

6

CONCLUSIONS AND FUTURE WORK

Several researchers around the globe share the effort of improving the performance of algorithms for autonomous driving. Along with autonomous cars reaching the core of cities, a task of special interest is the trajectory prediction of surrounding traffic participants as pedestrians and cyclists. Trajectory prediction means estimating the exact locations a pedestrian (or other agent types) will walk through in the future. Predicting future trajectories is especially important because if an agent's predicted future trajectory shares the same spatial location as the ego-vehicle planned path, the ego-vehicle has to plan a collision-avoidance maneuver, thus guaranteeing the safety of surrounding agents.

In this thesis, we have proposed a scene-compliant spatio-temporal multi-modal multi-agent long-term trajectory forecasting. The task of predicting trajectories is usually labeled as short or long-term, concerning the length of the future predicted trajectory. The prediction of long-term trajectories is a challenging task because such trajectories tend to be non-linear when compared with short-term, where the non-linear paths are segmented in smaller linear sections. Multi-modal or uni-modal defines the number of estimated trajectories for each given sample. Most current approaches take into consideration the multi-modal nature of the problem, as the decision for one path, among many possible, can depend on a piece of information that is not observed by an external agent. Multi-agent refers to the usage of different agent types. We also use surrounding agents' static locations. Such information has been exploited because there are common patterns agents exhibit when navigating in a shared environment as following or avoiding behavior.

While prior methods rely on high definition maps, focus solely on an agent type, neglect semantic information, or fail to establish a correspondence between trajectories and scene. We have proposed the prediction of multi-modal long-term trajectories by using spatio-time agent-centric grids for the scene, past trajectory, and surrounding agents. As an intermediate step, we generate probability grids that store the most probable paths according to the input information. Such probability grids enforce the generated trajectories to better comply with the scene. Rather

than using high-definition maps, that are costly to build and scale, our model automatically learns relevant scene features through the incorporation of a pre-trained semantic segmentation module. We do not focus on exclusive agent's type, neither we fix the number of surrounding agents. Current metrics face a problem for evaluating multi-modal trajectories due to the unimodality of the ground truth data. That means despite the model generates several trajectories, only one future trajectory is stored in the dataset, consequently arising a problem regarding the evaluation of the predicted multi-modal trajectories. Hence, we propose a new metric that estimates scene compliance for multi-modal trajectories. With such a metric we can evaluate how reasonable all the predicted trajectories are. Such metric can be used as a complement to the widely known ADE and FDE metrics.

By transforming the problem from the time-series space to the image space we can take advantage of using CNNs. We used U-Net and a ResNet-based network to, respectively, encode past trajectory and other agents, and the scene. ConvLSTMs were used to generate probability grids and trajectories, with the CoordConv layer to help the mapping from grid locations to trajectories. Our quantitative results on the SDD real-world dataset achieved positive competitive results when compared to the state-of-the-art. Qualitative results show that the predicted trajectories have interrelated consistency among consecutive points, are diverse, and conform with the past trajectory and the scene. We include a set of different ablations that demonstrate the contribution of different network setups. We provided a deep discussion regarding the network's configuration and the qualitative results, exploring different experiments and failure scenarios. We also provide a grasp into the interpretability in a global manner through the visualization of the time-expanded probability grids, and in a local manner through the visualization of the generated trajectories. We present a discussion concerning the qualitative results through the visualization of the generated trajectories. The proposed approach was able to identify common motion patterns along with different agents and scenes. We believe other agents' data can provide additional information, however, the improvement in the generated probability maps could not be qualitatively fully identified, i .e. through analyzing the generated probability maps we could not understand if the network was able to encode any common patterns of avoidance. Generated trajectories struggle to match the GT when the target agent is stopped in the past trajectory and only starts to move in the future trajectory.

Future work can extend this model by exploring different sampling techniques, predicting multiple agents' trajectories in parallel, using a forward-backward technique to refine the trajectories, and also predicting a K value that adapts to each different context. The exploitation of different sampling methods can be done by using the time-expanded probability maps generated by the first module, therefore replacing the second module of the network. We only used the static locations of surrounding agents, the usage of surrounding agents' past trajectories can provide more information regarding the direction they are walking towards, and therefore improve the results. Predicting several agent trajectories in parallel can be helpful because at each time step the actions of each agent are affecting the behavior of others. Also, predicting surrounding

agents' future trajectories might be helpful to predict the future trajectory of the target agent. The forward-backward error technique can be used for estimating if the forward and backward paths generate the same start/end point, if they do not, the distance between such points can be used in the training loss. Other training losses could also be tried to improve the performance. Most of the approaches use a fixed K value (1, 5, 10, or 20). A future approach could compute a K value that varies according to the number of plausible visible observed paths.

BIBLIOGRAPHY

ALAHI, A.; GOEL, K.; RAMANATHAN, V.; ROBICQUET, A.; FEI-FEI, L.; SAVARESE, S. Social LSTM: Human trajectory prediction in crowded spaces. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 961–971. Citations on pages 39, 40, 41, and 67.

AMINI, A. **MIT Introduction to Deep Learning Lecture**. Acessed on July 2020, 2020. Available: <https://www.youtube.com/watch?v=njKP3FqW3Sk>. Citation on page 44.

AMIRIAN, J.; HAYET, J.; PETTRÉ, J. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. **CoRR**, abs/1904.09507, 2019. Available: <http://arxiv.org/abs/1904.09507>. Citations on pages 28, 39, and 40.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). **Computer Vision – ECCV 2006**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 404–417. ISBN 978-3-540-33833-8. Citation on page 55.

BENGIO, Y. **Perspectives on AI: Next Steps for Deep Learning - Keynote at Khipu**. Accessed on Nov 2019, 2019. Available: <https://tv.vera.com.uy/video/55313>. Citation on page 46.

BENGIO, Y.; DELALLEAU, O.; ROUX, N. L. **The Curse of Dimensionality for Local Kernel Machines**. [S.l.], 2005. Available: <https://www.microsoft.com/en-us/research/publication/the-curse-of-dimensionality-for-local-kernel-machines/>. Citation on page 49.

BERTOZZI, M.; BROGGI, A.; FASCIOLI, A.; TIBALDI, A.; CHAPUIS, R.; CHAUSSE, F. Pedestrian localization and tracking system with Kalman filtering. In: **IEEE Intelligent Vehicles Symposium, 2004**. [S.l.: s.n.], 2004. p. 584–589. Citations on pages 34 and 37.

BERTOZZI, M.; CASTANGIA, L.; CATTANI, S.; PRIOLETTI, A.; VERSARI, P. 360 degree; detection and tracking algorithm of both pedestrian and vehicle using fisheye images. In: **2015 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2015. p. 132–137. ISSN 1931-0587. Citation on page 33.

BOCK, J.; KOTTE, J.; BEEMELMANNS, T.; KLÖSGES, M. Self-learning trajectory prediction with recurrent neural networks at intelligent intersections. In: **Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems**. [S.l.: s.n.], 2017. p. 346–351. ISBN 978-989-758-242-4. Citation on page 36.

BONNIN, S.; WEISSWANGE, T. H.; KUMMERT, F.; SCHMUEDDERICH, J. Pedestrian crossing prediction using multiple context-based models. In: **17th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2014. p. 378–385. ISSN 2153-0009. Citations on pages 27, 28, 35, and 38.

CALONDER, M.; LEPETIT, V.; STRECHA, C.; FUA, P. Brief: Binary robust independent elementary features. In: DANIILIDIS, K.; MARAGOS, P.; PARAGIOS, N. (Ed.). **Computer**

Vision – ECCV 2010. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 778–792. ISBN 978-3-642-15561-1. Citation on page 55.

CASAS, S.; LUO, W.; URTASUN, R. Intentnet: Learning to predict intention from raw sensor data. In: **Conference on Robot Learning**. [S.l.: s.n.], 2018. p. 947–956. Citations on pages 40 and 41.

CHEN, X.; GIRSHICK, R.; HE, K.; DOLLAR, P. Tensormask: A foundation for dense object segmentation. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2019. Citation on page 55.

CHO, K.; MERRIENBOER, B. van; BAHDANAU, D. On the properties of neural machine translation: Encoder–decoder approaches. 06 2014. Available: <https://arxiv.org/pdf/1409.1259.pdf>. Citations on pages 49 and 50.

CHOROMANSKA, A.; HENAFF, M.; MATHIEU, M.; AROUS, G. B.; LECUN, Y. The loss surfaces of multilayer networks. In: **Artificial Intelligence and Statistics**. [S.l.: s.n.], 2015. p. 192–204. Citation on page 48.

CHOU, F.-C.; LIN, T.-H.; CUI, H.; RADOSAVLJEVIC, V.; NGUYEN, T.; HUANG, T.-K.; NIEDOBA, M.; SCHNEIDER, J.; DJURIC, N. Predicting motion of vulnerable road users using high-definition maps and efficient convnets. **CoRR**, abs/1906.08469, 2019. Available: <https://arxiv.org/abs/1906.08469>. Citations on pages 39, 40, and 41.

CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 12 2014. Available: <https://www.researchgate.net/publication/269416998_Empirical_Evaluation_of_Gated_Recurrent_Neural_Networks_on_Sequence_Modeling>. Citations on pages 49 and 50.

CHURCHLAND, P. S.; SEJNOWSKI, T. J. Book. **The computational brain / Patricia S. Churchland and Terrence J. Sejnowski**. [S.l.]: MIT Press Cambridge, Mass, 1992. xi, 544 p. : p. ISBN 0262031884. Citation on page 45.

CLOUTIER, M.-S.; LACHAPELLE, U.; OUELLET, A.-A. d'Amours; BERGERON, J.; LORD, S.; TORRES, J. "outta my way!" individual and environmental correlates of interactions between pedestrians and vehicles during street crossings. In: . [s.n.], 2017. v. 104, n. Supplement C, p. 36 – 45. ISSN 0001-4575. Available: <http://www.sciencedirect.com/science/article/pii/S0001457517301446>. Citation on page 38.

CUI, H.; RADOSAVLJEVIC, V.; CHOU, F.-C.; LIN, T.-H.; NGUYEN, T.; HUANG, T.-K.; SCHNEIDER, J.; DJURIC, N. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 2090–2096. Citations on pages 27, 28, 39, 40, and 41.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01**. USA: IEEE Computer Society, 2005. (CVPR '05), p. 886–893. ISBN 0769523722. Available: <https://doi.org/10.1109/CVPR.2005.177>. Citation on page 55.

DEO, N.; TRIVEDI, M. Convolutional social pooling for vehicle trajectory prediction. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2018. p. 1468–1476. Citations on pages 39 and 40.

DEO, N.; TRIVEDI, M. M. Learning and predicting on-road pedestrian behavior around vehicles. In: **2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2017. p. 1–6. Citation on page 28.

Deo, N.; Trivedi, M. M. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In: **2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018**. [s.n.], 2018. p. 1179–1184. Available: <https://doi.org/10.1109/IVS.2018.8500493>. Citation on page 40.

DEO, N.; TRIVEDI, M. M. Scene induced multi-modal trajectory forecasting via planning. **CoRR**, abs/1905.09949, 2019. Available: <http://arxiv.org/abs/1905.09949>. Citations on pages 28 and 40.

DJURIC, N.; CUI, H.; SU, Z.; WU, S.; WANG, H.; CHOU, F.-C.; MARTIN, L. S.; FENG, S.; HU, R.; XU, Y. *et al.* Multinet: Multiclass multistage multimodal motion prediction. **arXiv preprint arXiv:2006.02000**, 2020. Citations on pages 40 and 41.

DJURIC, N.; RADOSAVLJEVIC, V.; CUI, H.; NGUYEN, T.; CHOU, F.-C.; LIN, T.-H.; SINGH, N.; SCHNEIDER, J. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In: **The IEEE Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2020. p. 2095–2104. Citations on pages 40 and 41.

DOMINGUEZ-SANCHEZ, A.; CAZORLA, M.; ORTS-ESCOLANO, S. Pedestrian movement direction recognition using convolutional neural networks. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 18, n. 12, p. 3540–3548, 2017. Citation on page 36.

ENZWEILER, M.; GAVRILA, D. M. Monocular pedestrian detection: Survey and experiments. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 31, n. 12, p. 2179–2195, 2009. Citation on page 33.

FÄRBER, B. **Communication and Communication Problems Between Autonomous Vehicles and Human Drivers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. 125–144 p. ISBN 978-3-662-48847-8. Available: <https://doi.org/10.1007/978-3-662-48847-8_7>. Citation on page 28.

FERGUSON, S.; LUDERS, B.; GRANDE, R. C.; HOW, J. P. Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions. In: AKIN, H. L.; AMATO, N. M.; ISLER, V.; STAPPEN, A. F. van der (Ed.). **Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics**. Cham: Springer International Publishing, 2015. p. 161–177. ISBN 978-3-319-16595-0. Citations on pages 27, 28, 33, and 36.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, p. 193–202, 1980. Citation on page 50.

FURUHASHI, R.; YAMADA, K. Estimation of street crossing intention from a pedestrian's posture on a sidewalk using multiple image frames. In: **The First Asian Conference on Pattern Recognition**. [S.l.: s.n.], 2011. p. 17–21. ISSN 0730-6512. Citation on page 34.

GANDHI, T.; TRIVEDI, M. M. Computer vision and machine learning for enhancing pedestrian safety. In: PROKHOROV, D. (Ed.). **Computational Intelligence in Automotive Applications**.

Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 59–77. ISBN 978-3-540-79257-4. Available: <https://doi.org/10.1007/978-3-540-79257-4_4>. Citations on pages 28 and 33.

GAVRILA, D. M. The visual analysis of human movement: A survey. **Computer Vision and Image Understanding**, v. 73, p. 82–98, 1999. Citation on page 33.

GAVRILA, D. M.; MUNDER, S. Multi-cue pedestrian detection and tracking from a moving vehicle. **International Journal of Computer Vision**, v. 73, n. 1, p. 41–59, Jun 2007. ISSN 1573-1405. Available: <https://doi.org/10.1007/s11263-006-9038-7>. Citation on page 33.

GERS, F.; SCHMIDHUBER, J. Recurrent nets that time and count. In: **Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium**. [S.l.: s.n.], 2000. v. 3, p. 189–194 vol.3. Citation on page 50.

GIRSHICK, R. Fast r-cnn. In: **2015 IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015. p. 1440–1448. Citation on page 55.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Region-based convolutional networks for accurate object detection and segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 38, n. 1, p. 142–158, Jan 2016. ISSN 0162-8828. Citation on page 55.

GOLDHAMMER, M.; GERHARD, M.; ZERNETSCH, S.; DOLL, K.; BRUNSMANN, U. Early prediction of a pedestrian's trajectory at intersections. In: **16th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2013. p. 237–242. ISSN 2153-0009. Citation on page 37.

GOLDHAMMER, M.; KÖHLER, S.; DOLL, K.; SICK, B. Camera based pedestrian path prediction by means of polynomial least-squares approximation and multilayer perceptron neural networks. In: **2015 SAI Intelligent Systems Conference (IntelliSys)**. [S.l.: s.n.], 2015. p. 390–399. Citations on pages 27, 28, 36, and 37.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. Book in preparation for MIT Press. Available: <http://www.deeplearningbook.org>. Citations on pages 43, 45, 46, 47, 48, 49, 50, 51, and 53.

GRAVES, A. Generating sequences with recurrent neural networks. **CoRR**, abs/1308.0850, 2013. Available: <http://arxiv.org/abs/1308.0850>. Citation on page 50.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. Lstm: A search space odyssey. **IEEE Transactions on Neural Networks and Learning Systems**, v. 28, n. 10, p. 2222–2232, Oct 2017. ISSN 2162-237X. Citations on pages 49 and 50.

GU, Y.; HASHIMOTO, Y.; HSU, L. T.; KAMIJO, S. Motion planning based on learning models of pedestrian and driver behaviors. In: **2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2016. p. 808–813. Citation on page 38.

GUPTA, A.; JOHNSON, J.; FEI-FEI, L.; SAVARESE, S.; ALAHI, A. Social gan: Socially acceptable trajectories with generative adversarial networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 2255–2264. Citations on pages 28, 39, 40, 41, 63, and 67.

HAN, D.; KIM, J.; KIM, J. Deep pyramidal residual networks. **CoRR**, abs/1610.02915, 2016. Available: <http://arxiv.org/abs/1610.02915>. Citation on page 47.

HARIYONO, J.; JO, K. Detection of pedestrian crossing road: A study on pedestrian pose recognition. **Neurocomputing**, v. 234, n. Supplement C, p. 144 – 153, 2017. ISSN 0925-2312. Available: <http://www.sciencedirect.com/science/article/pii/S0925231216315788>. Citations on pages 34, 37, and 38.

HARIYONO, J.; JO, K. H. Detection of pedestrian crossing road using action classification model. In: **2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)**. [S.l.: s.n.], 2015. p. 21–24. ISSN 2159-6247. Citations on pages 34 and 38.

HARIYONO, J.; JO, K.-H. Pedestrian action recognition using motion type classification. In: **2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)**. [S.l.: s.n.], 2015. p. 129–132. Citations on pages 34 and 38.

HARIYONO, J.; SHAHBAZ, A.; JO, K. H. Estimation of walking direction for pedestrian path prediction from moving vehicle. In: **2015 IEEE/SICE International Symposium on System Integration (SII)**. [S.l.: s.n.], 2015. p. 750–753. Citation on page 37.

HARRIS, C. G.; STEPHENS, M. *et al.* A combined corner and edge detector. In: CITESEER. **Alvey vision conference**. [S.l.], 1988. v. 15, n. 50, p. 10–5244. Citation on page 55.

HASAN, I.; SETTI, F.; TSESMELIS, T.; Del Bue, A.; GALASSO, F.; CRISTANI, M. MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses. **CoRR**, abs/1805.00652, 2018. Available: <http://arxiv.org/abs/1805.00652>. Citation on page 39.

HASHIMOTO, Y.; GU, Y.; HSU, L. T.; KAMIJO, S. Probability estimation for pedestrian crossing intention at signalized crosswalks. In: **2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES)**. [S.l.: s.n.], 2015. p. 114–119. Citations on pages 37 and 38.

HASHIMOTO, Y.; YANLEI, G.; HSU, L. T.; SHUNSUKE, K. A probabilistic model for the estimation of pedestrian crossing behavior at signalized intersections. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2015. p. 1520–1526. ISSN 2153-0009. Citations on pages 27, 35, 37, and 38.

HASTIE, R. T. T.; FRIEDMAN, J. The elements of statistical learning: Data mining, inference, and prediction. **Journal of the Royal Statistical Society: Series A (Statistics in Society)**, v. 173, n. 3, p. 693–694, 2010. Available: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-985X.2010.00646_6.x>. Citations on pages 48, 49, and 65.

HAYKIN, S. S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009. Citations on pages 44 and 45.

HE, K.; GKIOXARI, G.; DOLLáR, P.; GIRSHICK, R. Mask r-cnn. In: **2017 IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2017. p. 2980–2988. Citation on page 55.

He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778. ISSN 1063-6919. Citations on pages 15, 55, and 61.

HEBB, D. O. **The organization of behavior: A neuropsychological theory**. New York: Wiley, 1949. Hardcover. ISBN 0-8058-4300-0. Citation on page 44.

HELBING, D.; MOLNÁR, P. Social force model for pedestrian dynamics. **Phys. Rev. E**, American Physical Society, v. 51, p. 4282–4286, May 1995. Available: <https://link.aps.org/doi/10.1103/PhysRevE.51.4282>. Citation on page 38.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>. Citation on page 49.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017. Citation on page 55.

IACOBONI, M.; MOLNAR-SZAKACS, I.; GALLESE, V.; BUCCINO, G.; MAZZIOTTA, J. C.; RIZZOLATTI, G. Grasping the intentions of others with one's own mirror neuron system. **PLOS Biology**, Public Library of Science, v. 3, n. 3, 02 2005. Available: <https://doi.org/10.1371/journal.pbio.0030079>. Citation on page 33.

Isola, P.; Zhu, J.; Zhou, T.; Efros, A. A. Image-to-image translation with conditional adversarial networks. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017. p. 5967–5976. ISSN 1063-6919. Citation on page 64.

JIN, L.; NIU, Q.; HOU, H.; SHUNXI, H.; FANGRONG, W. Study on vehicle front pedestrian detection based on 3d laser scanner. In: **Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)**. [S.l.: s.n.], 2011. p. 735–738. Citation on page 33.

JOHNSON, J. **Stanford Winter Quarter 2016 class: CS231n: Convolutional Neural Networks for Visual Recognition, Lecture 11: ConvNets in practice**. Accessed on April 2020, 2016. Available: <https://www.youtube.com/watch?v=pA4BsUK3oP4&list=PLkt2uSq6rBVctENoVBg1TpCC7OQi31AlC&index=11>. Citation on page 54.

KARASEV, V.; AYVACI, A.; HEISELE, B.; SOATTO, S. Intent-aware long-term prediction of pedestrian motion. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2016. p. 2543–2549. Citations on pages 28 and 37.

KARPATHY, A. **Stanford Winter Quarter 2016 class: CS231n: Convolutional Neural Networks for Visual Recognition**. Accessed on April 2020, 2016. Available: <https://www.youtube.com/watch?v=NfnWJUyUJYU&list=PLkt2uSq6rBVctENoVBg1TpCC7OQi31AlC>. Citation on page 49.

KARPATHY, A.; JOHNSON, J.; LI, F. Visualizing and understanding recurrent networks. **CoRR**, abs/1506.02078, 2015. Available: <http://arxiv.org/abs/1506.02078>. Citation on page 49.

KELLER, C. G.; DANG, T.; FRITZ, H.; JOOS, A.; RABE, C.; GAVRILA, D. M. Active pedestrian safety by automatic braking and evasive steering. **IEEE Transactions on Intelligent Transportation Systems**, v. 12, n. 4, p. 1292–1304, Dec 2011. ISSN 1524-9050. Citation on page 33.

KELLER, C. G.; GAVRILA, D. M. Will the pedestrian cross? a study on pedestrian path prediction. **IEEE Transactions on Intelligent Transportation Systems**, v. 15, n. 2, p. 494–506, April 2014. ISSN 1524-9050. Citations on pages 34 and 37.

KELLER, C. G.; HERMES, C.; GAVRILA, D. M. Will the pedestrian cross? probabilistic path prediction based on learned motion features. In: MESTER, R.; FELSBERG, M. (Ed.). **Pattern Recognition: 33rd DAGM Symposium, Frankfurt/Main, Germany, August 31 – September 2, 2011. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 386–395. ISBN 978-3-642-23123-0. Available: <https://doi.org/10.1007/978-3-642-23123-0_39>. Citation on page 34.

KIM, K.; OWECHKO, Y.; MEDASANI, S. Active safety and collision alerts using contextual visual dataspace. In: **2010 IEEE Intelligent Vehicles Symposium**. [S.l.: s.n.], 2010. p. 424–430. ISSN 1931-0587. Citation on page 38.

KIM, K.-S.; LEE, J.-B.; ROH, M.-I.; HAN, K.-M.; LEE, G.-H. Prediction of ocean weather based on denoising autoencoder and convolutional lstm. **Journal of Marine Science and Engineering**, v. 8, n. 10, 2020. ISSN 2077-1312. Available: <https://www.mdpi.com/2077-1312/8/10/805>. Citation on page 50.

KINGMA, D.; BA, J. Adam: A method for stochastic optimization. **International Conference on Learning Representations**, 12 2014. Citations on pages 48 and 64.

KIRILLOV, A.; GIRSHICK, R.; HE, K.; DOLLAR, P. Panoptic feature pyramid networks. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019. Citation on page 55.

KITANI, K. M.; ZIEBART, B. D.; BAGNELL, J. A.; HEBERT, M. Activity forecasting. In: FITZGIBBON, A.; LAZEBNIK, S.; PERONA, P.; SATO, Y.; SCHMID, C. (Ed.). **Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 201–214. ISBN 978-3-642-33765-9. Available: <http://dx.doi.org/10.1007/978-3-642-33765-9_15>. Citations on pages 28 and 37.

KOEHLER, S.; GOLDHAMMER, M.; BAUER, S.; ZECHA, S.; DOLL, K.; BRUNSMANN, U.; DIETMAYER, K. Stationary detection of the pedestrian's intention at intersections. **IEEE Intelligent Transportation Systems Magazine**, v. 5, n. 4, p. 87–99, winter 2013. ISSN 1939-1390. Citations on pages 27 and 35.

KÖHLER, S.; GOLDHAMMER, M.; BAUER, S.; DOLL, K.; BRUNSMANN, U.; DIETMAYER, K. Early detection of the pedestrian's intention to cross the street. In: **2012 15th International IEEE Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2012. p. 1759–1764. ISSN 2153-0009. Citation on page 34.

KÖHLER, S.; GOLDHAMMER, M.; ZINDLER, K.; DOLL, K.; DIETMEYER, K. Stereo-vision-based pedestrian's intention detection in a moving vehicle. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2015. p. 2317–2322. ISSN 2153-0009. Citation on page 34.

KOOIJ, J. F. P.; SCHNEIDER, N.; FLOHR, F.; GAVRILA, D. M. Context-based pedestrian path prediction. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). **Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI**. Cham: Springer International Publishing, 2014. p. 618–633. ISBN 978-3-319-10599-4. Available: <https://doi.org/10.1007/978-3-319-10599-4_40>. Citations on pages 27, 28, 36, and 38.

KOOIJ, J. F. P.; SCHNEIDER, N.; GAVRILA, D. M. Analysis of pedestrian dynamics from a vehicle perspective. In: **2014 IEEE Intelligent Vehicles Symposium Proceedings**. [S.l.: s.n.], 2014. p. 1445–1450. ISSN 1931-0587. Citation on page 37.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGES, C. J. C.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems 25**. Curran Associates, Inc., 2012. p. 1097–1105. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>. Citations on pages 46, 50, and 55.

KWAK, J.-Y.; KO, B. C.; NAM, J.-Y. Pedestrian intention prediction based on dynamic fuzzy automata for vehicle driving at nighttime. **Infrared Physics and Technology**, v. 81, n. Supplement C, p. 41 – 51, 2017. ISSN 1350-4495. Available: <http://www.sciencedirect.com/science/article/pii/S1350449516304935>. Citations on pages 27 and 35.

LECUN, Y. **Self-Supervised Learning World Models - Plenary talk on ICRA 2020**. Accessed on May 2021, 2020. Available: <https://www.youtube.com/watch?v=eZo1zEepWc0>. Citation on page 29.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Citations on pages 46 and 50.

LEE, N.; CHOI, W.; VERNAZA, P.; CHOY, C. B.; TORR, P. H.; CHANDRAKER, M. Desire: Distant future prediction in dynamic scenes with interacting agents. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 336–345. Citations on pages 28, 30, 39, 40, 41, 63, and 67.

LI, H.; XU, Z.; TAYLOR, G.; STUDER, C.; GOLDSTEIN, T. Visualizing the loss landscape of neural nets. In: BENGIO, S.; WALLACH, H.; LAROCHELLE, H.; GRAUMAN, K.; CESA-BIANCHI, N.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems 31**. Curran Associates, Inc., 2018. p. 6389–6399. Available: <http://papers.nips.cc/paper/7875-visualizing-the-loss-landscape-of-neural-nets.pdf>. Citation on page 61.

LI, J.; LI, Q.; CHEN, N.; WANG, Y. Indoor pedestrian trajectory detection with lstm network. In: **2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)**. [S.l.: s.n.], 2017. v. 1, p. 651–654. Citations on pages 36 and 49.

LI, X.; LI, L.; FLOHR, F.; WANG, J.; XIONG, H.; BERNHARD, M.; PAN, S.; GAVRILA, D. M.; LI, K. A unified framework for concurrent pedestrian and cyclist detection. **IEEE Transactions on Intelligent Transportation Systems**, v. 18, n. 2, p. 269–281, Feb 2017. ISSN 1524-9050. Citation on page 33.

LI, Y. Pedestrian path forecasting in crowd: A deep spatio-temporal perspective. In: **Proceedings of the 25th ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2017. (MM '17), p. 235–243. ISBN 978-1-4503-4906-2. Available: <http://doi.acm.org/10.1145/3123266.3123287>. Citation on page 40.

Li, Y. Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019. Citation on page 40.

LIANG, J.; JIANG, L.; NIEBLES, J. C.; HAUPTMANN, A. G.; FEI-FEI, L. Peeking into the future: Predicting future person activities and locations in videos. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019. Citation on page 39.

LIN, S. F.; LEE, C. H. Pedestrians and vehicles recognition based on image recognition and laser distance detection. In: **2016 16th International Conference on Control, Automation and Systems (ICCAS)**. [S.l.: s.n.], 2016. p. 1232–1237. Citation on page 33.

LIU, R.; LEHMAN, J.; MOLINO, P.; SUCH, F. P.; FRANK, E.; SERGEEV, A.; YOSINSKI, J. An intriguing failing of convolutional neural networks and the coordconv solution. **CoRR**, abs/1807.03247, 2018. Available: <http://arxiv.org/abs/1807.03247>. Citation on page 63.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **Int. J. Comput. Vision**, Kluwer Academic Publishers, USA, v. 60, n. 2, p. 91–110, Nov. 2004. ISSN 0920-5691. Available: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>. Citation on page 55.

LUO, W.; YANG, B.; URTASUN, R. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: **Proceedings of the IEEE conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 3569–3577. Citations on pages 40 and 41.

MARSLAND, S. **Machine Learning: An Algorithmic Perspective, Second Edition**. 2nd. ed. [S.l.]: Chapman & Hall/CRC, 2014. ISBN 1466583282, 9781466583283. Citation on page 43.

MCCULLOCH, W.; PITTS, W. A logical calculus of ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 127–147, 1943. Citation on page 44.

MEISSNER, S. R. D.; DIETMAYER, K. Real-time detection and tracking of pedestrians at intersections using a network of laserscanners. In: **2012 IEEE Intelligent Vehicles Symposium**. [S.l.: s.n.], 2012. p. 630–635. ISSN 1931-0587. Citation on page 33.

MELLO, R. F. de; FERREIRA, M. D.; PONTI, M. A. Providing theoretical learning guarantees to deep learning networks. **arXiv preprint arXiv:1711.10292**, 2017. Citation on page 51.

MITCHELL, T. M. **Machine Learning**. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citations on pages 44 and 45.

MøGELMOSE, A.; TRIVEDI, M. M.; MOESLUND, T. B. Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations. In: **2015 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2015. p. 330–335. ISSN 1931-0587. Citations on pages 34 and 37.

NIEDOBA, M.; CUI, H.; LUO, K.; HEGDE, D.; CHOU, F.-C.; DJURIC, N. Improving movement prediction of traffic actors using off-road loss and bias mitigation. In: **Workshop on'Machine Learning for Autonomous Driving'at Conference on Neural Information Processing Systems**. [S.l.: s.n.], 2019. Citations on pages 40 and 41.

NIKHIL, N.; MORRIS, B. T. Convolutional neural network for trajectory prediction. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 0–0. Citation on page 39.

OHN-BAR, E.; TRIVEDI, M. M. Looking at humans in the age of self-driving and highly automated vehicles. **IEEE Transactions on Intelligent Vehicles**, v. 1, n. 1, p. 90–104, March 2016. ISSN 2379-8858. Citation on page 34.

Ohn-Bar, E.; Trivedi, M. M. Looking at humans in the age of self-driving and highly automated vehicles. **IEEE Transactions on Intelligent Vehicles**, v. 1, n. 1, p. 90–104, March 2016. ISSN 2379-8904. Citation on page 34.

PAPADIMITRIOU, E.; LASSARRE, S.; YANNIS, G. Pedestrian risk taking while road crossing: A comparison of observed and declared behaviour. **Transportation Research Procedia**, v. 14, p. 4354–4363, 2016. ISSN 2352-1465. Transport Research Arena TRA2016. Available: <https://www.sciencedirect.com/science/article/pii/S2352146516303635>. Citation on page 73.

PELLEGRINI, S.; ESS, A.; SCHINDLER, K.; GOOL, L. van. You'll never walk alone: Modeling social behavior for multi-target tracking. In: **2009 IEEE 12th International Conference on Computer Vision**. [S.l.: s.n.], 2009. p. 261–268. ISSN 1550-5499. Citation on page 38.

POMERLEAU, D. A. Alvinn: An autonomous land vehicle in a neural network. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1989. p. 305–313. Citation on page 45.

PONTI, M. A.; COSTA, G. B. P. d. Como funciona o deep learning. In: ____. **Brazilian Symposium on Databases - SBBD**. [S.l.]: SBC, 2017. Citation on page 51.

QUINTERO, R.; PARRA, I.; LLORCA, D.; SOTELO, M. Pedestrian intention and pose prediction through dynamical models and behaviour classification. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2015. p. 83–88. ISSN 2153-0009. Citations on pages 27, 28, 33, 34, and 35.

QUINTERO, R.; PARRA, I.; LLORCA, D. F.; SOTELO, M. A. Pedestrian path prediction based on body language and action classification. In: **17th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2014. p. 679–684. ISSN 2153-0009. Citation on page 34.

R. OMRAN M., H. J. B.; SCHIELE, B. Ten years of pedestrian detection, what have we learned? In: L., B. M. M. A.; C., R. (Ed.). **Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II**. Cham: Springer International Publishing, 2015. p. 613–627. ISBN 978-3-319-16181-5. Available: <http://dx.doi.org/10.1007/978-3-319-16181-5_47>. Citation on page 33.

RAHMAN, M. M.; SIDDIQUI, F. H. An optimized abstractive text summarization model using peephole convolutional lstm. **Symmetry**, v. 11, n. 10, 2019. ISSN 2073-8994. Available: <https://www.mdpi.com/2073-8994/11/10/1290>. Citation on page 50.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. Citation on page 55.

REHDER, E.; KLOEDEN, H. Goal-directed pedestrian prediction. In: **2015 IEEE International Conference on Computer Vision Workshop (ICCVW)**. [S.l.: s.n.], 2015. p. 139–147. Citation on page 37.

REHDER, E.; WIRTH, F.; LAUER, M.; STILLER, C. Pedestrian prediction by planning using deep neural networks. In: **IEEE Int. Conf. International Conference on Robotics and Automation**. [S.l.: s.n.], 2018. Citation on page 36.

RHINEHART, N.; KITANI, K. M.; VERNAZA, P. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 772–788. Citations on pages 39 and 40.

RIDEL, D.; REHDER, E.; LAUER, M.; STILLER, C.; WOLF, D. A literature review on the prediction of pedestrian behavior in urban scenarios. In: **2018 21st International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2018. p. 3105–3112. ISSN 2153-0017. Citation on page 34.

RIDEL, D. A.; DEO, N.; WOLF, D. F.; TRIVEDI, M. M. Understanding pedestrian-vehicle interactions with vehicle mounted vision: An LSTM model and empirical analysis. In: **IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2019. Citation on page 39.

ROBICQUET, A.; SADEGHIAN, A.; ALAHI, A.; SAVARESE, S. Learning social etiquette: Human trajectory understanding in crowded scenes. In: **ECCV**. [S.l.: s.n.], 2016. Citations on pages 61, 65, and 67.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: . [S.l.: s.n.], 2015. v. 9351, p. 234–241. ISBN 978-3-319-24573-7. Citations on pages 15, 55, 56, 57, and 62.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citation on page 44.

ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). **Computer Vision – ECCV 2006**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 430–443. ISBN 978-3-540-33833-8. Citation on page 55.

ROTTENSTEINER, F.; SOHN, G.; JUNG, J.; GERKE, M.; BAILLARD, C.; BENITEZ, S.; BREITKOPF, U. The isprs benchmark on urban object classification and 3d building reconstruction. **ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1**, Göttingen: Copernicus GmbH, v. 1, n. 1, p. 293–298, 2012. Citations on pages 15, 61, and 62.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: An efficient alternative to sift or surf. In: **2011 International Conference on Computer Vision**. [S.l.: s.n.], 2011. p. 2564–2571. Citation on page 55.

RUDENKO, A.; PALMIERI, L.; HERMAN, M.; KITANI, K. M.; GAVRILA, D. M.; ARRAS, K. O. Human motion trajectory prediction: A survey. **CoRR**, abs/1905.06113, 2019. Available: <http://arxiv.org/abs/1905.06113>. Citation on page 34.

SADEGHIAN, A.; KOSARAJU, V.; SADEGHIAN, A.; HIROSE, N.; REZATOFIGHI, H.; SAVARESE, S. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019. Citations on pages 28, 39, 40, 41, 63, 64, and 67.

SADEGHIAN, A.; LEGROS, F.; VOISIN, M.; VESEL, R.; ALAHI, A.; SAVARESE, S. Car-net: Clairvoyant attentive recurrent network. In: FERRARI, V.; HEBERT, M.; SMINCHISESCU, C.; WEISS, Y. (Ed.). **Computer Vision – ECCV 2018**. Cham: Springer International Publishing, 2018. p. 162–180. ISBN 978-3-030-01252-6. Citations on pages 39, 40, 41, and 67.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2018. Citation on page 55.

SCHLOSSER, J.; CHOW, C. K.; KIRA, Z. Fusing lidar and images for pedestrian detection using convolutional neural networks. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2016. p. 2198–2205. Citation on page 33.

SCHMIDT, S.; FÄRBER, B. Pedestrians at the kerb – recognising the action intentions of humans. **Transportation Research Part F: Traffic Psychology and Behaviour**, v. 12, n. 4, p. 300 – 310, 2009. ISSN 1369-8478. Available: <http://www.sciencedirect.com/science/article/pii/S1369847809000102>. Citations on pages 28 and 37.

SCHNEIDER, N.; GAVRILA, D. M. Pedestrian path prediction with recursive bayesian filters: A comparative study. In: WEICKERT, J.; HEIN, M.; SCHIELE, B. (Ed.). **Pattern Recognition: 35th German Conference, GCPR 2013, Saarbrücken, Germany, September 3-6, 2013. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 174–183. ISBN 978-3-642-40602-7. Available: <https://doi.org/10.1007/978-3-642-40602-7_18>. Citations on pages 27, 33, 34, 35, and 37.

SCHULZ; STIEFELHAGEN, R. Pedestrian intention recognition using latent-dynamic conditional random fields. In: **2015 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2015. p. 622–627. ISSN 1931-0587. Citations on pages 27, 28, and 37.

SCHULZ, A. T.; STIEFELHAGEN, R. A controlled interactive multiple model filter for combined pedestrian intention recognition and path prediction. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2015. p. 173–178. ISSN 2153-0009. Citations on pages 36, 37, and 38.

Schöller, C.; Aravantinos, V.; Lay, F.; Knoll, A. What the constant velocity model can teach us about pedestrian motion prediction. **IEEE Robotics and Automation Letters**, v. 5, n. 2, p. 1696–1703, 2020. Citation on page 30.

SHI, J.; TOMASI. Good features to track. In: **1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 1994. p. 593–600. Citation on page 55.

SHI, X.; CHEN, Z.; WANG, H.; YEUNG, D.-Y.; WONG, W.-K.; WOO, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. **Advances in neural information processing systems**, v. 28, 2015. Citation on page 50.

SHIRAZI, M. S.; MORRIS, B. Observing behaviors at intersections: A review of recent studies amp; developments. In: **2015 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2015. p. 1258–1263. ISSN 1931-0587. Citation on page 34.

SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. 2015. Citation on page 55.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. *et al.* Going deeper with convolutions. In: CVPR. [S.l.], 2015. Citations on pages 47 and 55.

TAMURA, Y.; LE, P. D.; HITOMI, K.; CHANDRASIRI, N. P.; BANDO, T.; YAMASHITA, A.; ASAMA, H. Development of pedestrian behavior model taking account of intention. In: **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.: s.n.], 2012. p. 382–387. ISSN 2153-0858. Citation on page 38.

TAN, M.; LE, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In: CHAUDHURI, K.; SALAKHUTDINOV, R. (Ed.). **Proceedings of the 36th International Conference on Machine Learning**. PMLR, 2019. (Proceedings of Machine Learning Research, v. 97), p. 6105–6114. Available: <http://proceedings.mlr.press/v97/tan19a.html>. Citation on page 55.

VASISHTA, P.; VAUFREYDAZ, D.; SPALANZANI, A. Natural vision based method for predicting pedestrian behaviour in urban environments. In: **2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2017. p. 1–6. Citation on page 34.

VEMULA, A.; MUELLING, K.; OH, J. Social attention: Modeling attention in human crowds. In: IEEE. **2018 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2018. p. 1–7. Citation on page 39.

VÖLZ, B.; MIELENZ, H.; AGAMENNONI, G.; SIEGWART, R. Feature relevance estimation for learning pedestrian behavior at crosswalks. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2015. p. 854–860. ISSN 2153-0009. Citations on pages 27 and 35.

VÖLZ, B.; MIELENZ, H.; SIEGWART, R.; NIETO, J. Predicting pedestrian crossing using quantile regression forests. In: **2016 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2016. p. 426–432. Citation on page 38.

WANG, E.; CUI, H.; YALAMANCHI, S.; MOORTHY, M.; CHOU, F.-C.; DJURIC, N. Improving movement predictions of traffic actors in bird's-eye view models using gans and differentiable trajectory rasterization. **arXiv preprint arXiv:2004.06247**, 2020. Citations on pages 40 and 41.

WANG, L.; ZHANG, L.; YI, Z. Trajectory predictor by using recurrent neural networks in visual tracking. **IEEE Trans. Cybernetics**, v. 47, n. 10, p. 3172–3183, 2017. Available: <https://doi.org/10.1109/TCYB.2017.2705345>. Citation on page 40.

WENG, X.; WANG, J.; LEVINE, S.; KITANI, K.; NICK, R. 4D Forecasting: Sequantial Forecasting of 100,000 Points. **ECCVW**, 2020. Citation on page 33.

World Health Organization. **Global status report on road safety 2018**. Accessed on May 2021, 2018. 424 p. Available: <https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/>. Citation on page 27.

Yamaguchi, K.; Berg, A. C.; Ortiz, L. E.; Berg, T. L. Who are you with and where are you going? In: **CVPR 2011**. [S.l.: s.n.], 2011. p. 1345–1352. ISSN 1063-6919. Citation on page 67.

YE, Q.; LIANG, J.; JIAO, J. Pedestrian detection in video images via error correcting output code classification of manifold subclasses. **IEEE Transactions on Intelligent Transportation Systems**, v. 13, n. 1, p. 193–202, March 2012. ISSN 1524-9050. Citation on page 33.

ZENG, W.; CHEN, P.; NAKAMURA, H.; IRYO-ASANO, M. Application of social force model to pedestrian behavior analysis at signalized crosswalk. **Transportation Research Part C: Emerging Technologies**, v. 40, p. 143 – 159, 2014. ISSN 0968-090X. Available: <http://www.sciencedirect.com/science/article/pii/S0968090X14000114>. Citation on page 38.

ZENG, W.; LUO, W.; SUO, S.; SADAT, A.; YANG, B.; CASAS, S.; URTASUN, R. End-to-end interpretable neural motion planner. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019. Citation on page 39.

ZHANG R. BENENSON, M. O. J. H. S.; SCHIELE, B. How far are we from solving pedestrian detection? In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 1259–1267. Citation on page 33.

Zyner, A.; Worrall, S.; Nebot, E. Naturalistic driver intention and path prediction using recurrent neural networks. **IEEE Transactions on Intelligent Transportation Systems**, p. 1–11, 2019. ISSN 1524-9050. Citations on pages 28, 39, and 40.

GLOSSARY

ADE Average Displacement Error.

AGV Autonomous Ground Vehicle.

B-GPDM Balanced Gaussian Process Dynamical Models.

BEV Birds Eye View.

CNN Convolutional Neural Network.

ConvLSTM Convolutional LSTM.

CP Constant Position.

CS Correspondence to Scene.

CV Constant Velocity.

DBN Dynamic Bayesian Network.

DL Deep Learning.

EKF Extended Kalman Filter.

FDE Final Displacement Error.

FFA Fuzzy Finite Automata.

GAN Generative Adversarial Network.

GP Gaussian Process.

GPU Graphical Processing Units.

GRU Gated Recurrent Unit.

GT Ground Truth.

IMM Interacting Multiple Model.

IMM-EKF Interacting Multiple Model Extended Kalman Filter.

ISPRS International Society for Photogrammetry and Remote Sensing.

KF Kalman Filter.

LDCRF Latent Dynamic Conditional Random Fields.

LDS Linear Dynamical System.

LSTM Long Short-Term Memory.

MCHOG Motion Contour Histogram of Oriented Gradients.

ML Machine Learning.

MLP Multilayer Perceptron.

PF Particle Filter.

PHTM Probabilistic Hierarchical Trajectory Matching.

ReLU Rectified Linear Units.

ResNet Residual Networks.

RGB Red Green Blue.

RNN Recurrent Neural Network.

ROC Receiver Operating Characteristic Curve.

SDD Stanford Drone Dataset.

SLDS Switching Linear Dynamics System.

SVM Support Vector Machine.

UAV Unmanned Aerial Vehicle.

APPENDIX

A

PUBLICATIONS

Published journal articles:

1. RIDEL, DANIELA A..; DEO, NACHIKET.; WOLF, DENIS.; and TRIVEDI, MOHAN. Scene Compliant Trajectory Forecast with Agent-Centric Spatio-Temporal Grids, In: Robotics and Automation Letters (RA-L) and International Conference on Robotics and Automation (ICRA), Paris/France, 2020. (A1)

Published conference papers:

1. RIDEL, DANIELA A..; DEO, NACHIKET.; WOLF, DENIS.; and TRIVEDI, MOHAN. Understanding Pedestrian-Vehicle Interactions with Vehicle Mounted Vision: An LSTM Model and Empirical Analysis, In: Intelligent Vehicles Symposium (IV), Paris/France, 2019. (A1)

2. RIDEL, DANIELA A..; REHDER, EIKE.; LAUER, MARTIN.; STILLER, CHRISTOPH.; and WOLF, DENIS. A Literature Review on the Prediction of Pedestrian Behavior in Urban Scenarios, In: The 21st IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui/Hawaii, 2018. (A2)

3. RIDEL, DANIELA A..; SHINZATO, PATRICK Y.; PEREIRA, ANA R.; GRASSI, VALDIR.; and WOLF, DENIS F. . Obstacle avoidance using stereo-based generic obstacle tracking, In: 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba/Brazil, 2017, pp. 1-6. (B1)

4. SHINZATO, PATRICK Y. ; DOS SANTOS, TIAGO C. ; ROSERO, LUIS ALBERTO; RIDEL, DANIELA A. ; MASSERA, CARLOS M. ; ALENCAR, FRANCISCO ; BATISTA, MARCOS PAULO ; HATA, ALBERTO Y. ; OSORIO, FERNANDO S. ; WOLF, DENIS F. . CaRINA dataset: An emerging-country urban scenario benchmark for road detection

systems. In: IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, Rio de Janeiro/Brazil. p. 41. (A2)


ICMC
USP
SÃO CARLOS