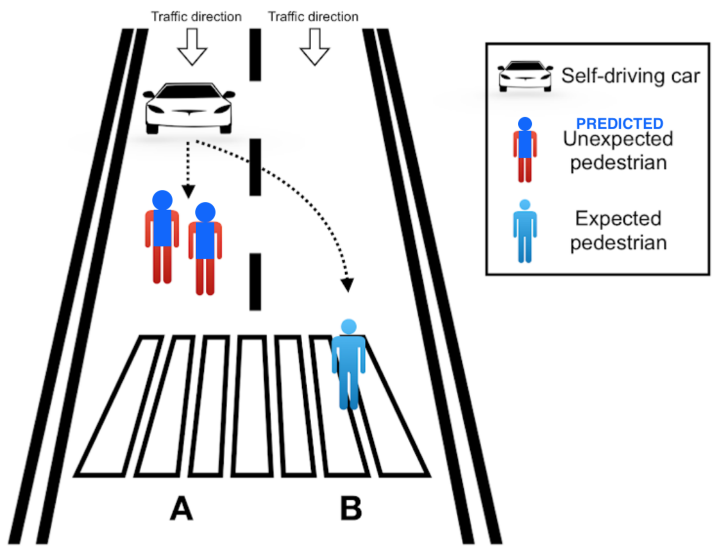
Pedestrian trajectory prediction

Rodolphe Farrando & Romain Gratier
EPFL – ENAC Faculty
{rodolphe.farrando, romain.gratierdesaint-louis}@epfl.ch

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Introduction



- Trajectory prediction is crucial for improving autonomous vehicles behaviour
- Could avoid situations seen in the ethical lectures

Previous Work

In their project, they used different components to make the structure:

- One LSTM per pedestrian
- Social Pooling
- Prediction per frame

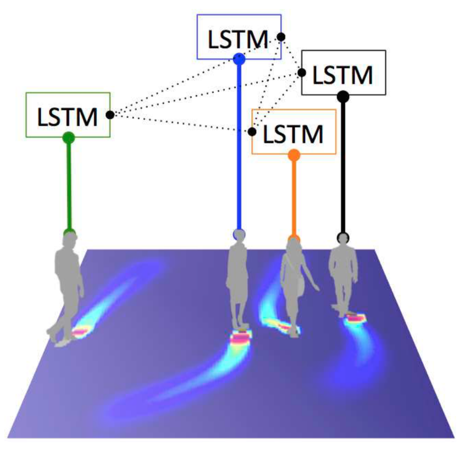


Figure 1: Application

Idea for our Models

In our project, we use used the following structures independently :

- One CNN, or one LSTM
- Prediction per pedestrian

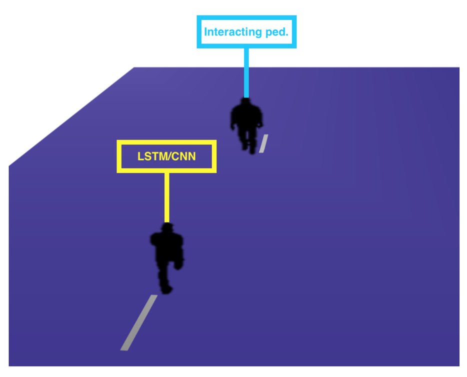


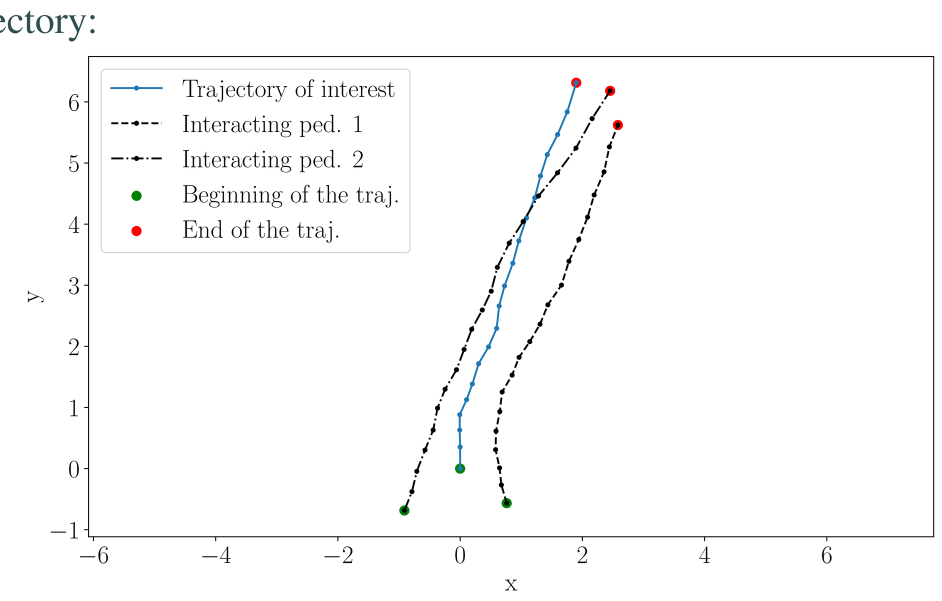
Figure 2: Application

Data

The preprocessing is divided in 5 steps:

1. Isolate each trajectory along with its interaction
2. Normalize the trajectories: the first point is at $(0, 0)$; the second is at $(0, y_1)$
3. Calculate axis velocities V_x and V_y
4. For each frame, if there is an interacting pedestrian we add his/her coordinates and speed otherwise we add zeros
5. Data augmentation: flip and add noise to trajectories

Representation of a trajectory:



Final shape of the data:

- Pedestrians ID
- Frame number
- Twenty sets of x and y coordinates per pedestrian

Frame Number	ID	x	y	V_x	V_y
0	i	0	0	0	0
10	i	0	y_1	0	V_{y_1}
⋮	⋮	⋮	⋮	⋮	⋮

Objectives

- Train on the 10 first coordinates and speed and their interaction
- Predict the next 10
- Inputs have the following shape: $[10, N, 4 * N_{inter}]$

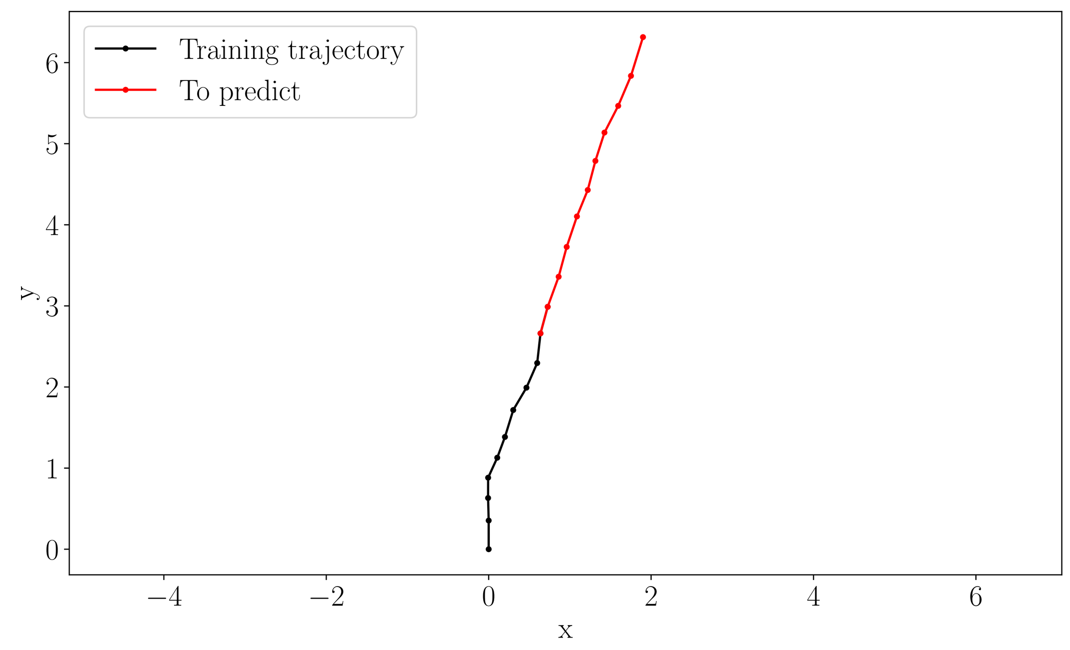
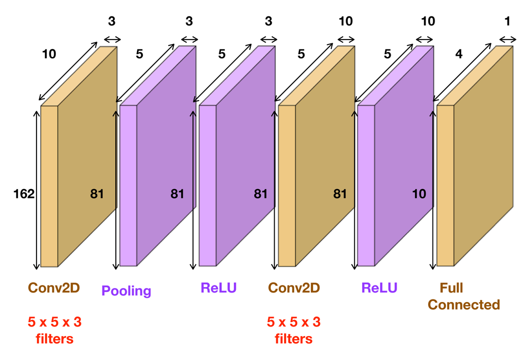


Figure 3

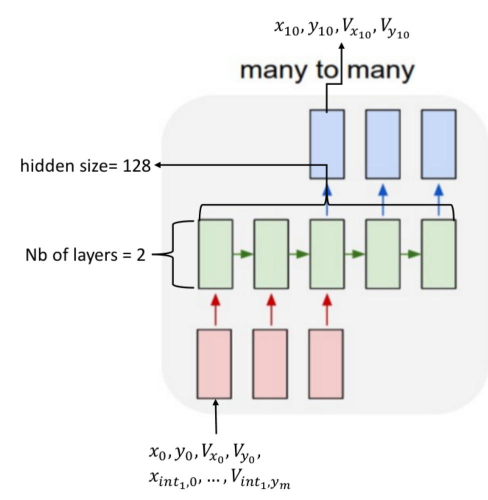
Models

Inputs: sequence of coordinates and velocities of the trajectory of interest and of the interacting trajectories
Outputs: sequence of predicted coordinates and velocities for the trajectory of interest.

CNN



LSTM



1 Results

1.1 Introduction

To calculate the correctness of the prediction two indicators are used:

1. The final displacement error: $e_{fin} = \sqrt{(X_{gt,n} - X_{pred,n})^2}$
2. The mean displacement error: $e_{mean} = \sqrt{\frac{\sum_{i=0}^{n}(X_{gt,i} - X_{pred,i})^2}{(n)}}$

Depending on the inputs two ways are possible to find the predicted coordinates:

1. If the coordinates are predicted: directly use them
2. If the velocities are predicted: $X_t = X_{t-1} + V_t \cdot 0.4$, with 0.4 the time between two frames in seconds

Four different cases, that corresponds to four losses, are tested for each model:

1. Predict coordinates with loss defines as $L_1 = (X - X_{pred})^2$ with $X = [x, y]$
2. Predict speeds with loss defines as $L_2 = (V - V_{pred})^2$ with $V = [V_x, V_y]$
3. Predict both coordinates and speeds with loss defines as $L = L_1 + L_2$
4. Predict both coordinates and speeds with loss defines as $L = L_1 + L_2 + L_3$, with $L_3 = (X - X_{t-1} + V_t * 0.4)^2$

Test set separate in three trajectory types:

1. Static
2. Linear trajectories
3. Non-linear trajectories

1.2 Table

Results with linear prediction:

- Type 1: *Mean* = 0.141, *Final* = 0.322
- Type 2: *Mean* = 0.541, *Final* = 0.93
- Type 3: *Mean* = 0.651, *Final* = 1.457
- Total: *Mean* = 0.512, *Final* = 0.982

		CNN				LSTM			
		Type 1	**Type 2**	**Type 3**	**Total**	**Type 1**	**Type 2**	**Type 3**	**Total**
Coord.	**Mean**	4.696	5.144	4.674	4.176	1.309	0.777	0.862	0.877
	Final	10.246	7.009	10.501	5.602	1.385	0.92	1.108	1.037
Speed	**Mean**	0.567	5.133	1.911	4.17	0.726	0.573	0.651	0.616
	Final	0.77	6.971	3.882	5.587	1.412	1.045	1.231	1.148
2 Losses	**Mean**	1.269	5.134	1.762	**4.163**	0.695	0.532	0.627	**0.581**
	Final	2.727	6.978	3.546	**5.57**	1.302	0.963	1.2	**1.076**
3 Losses	**Mean**	0.549	5.135	3.829	4.163	0.748	0.607	0.681	0.647
	Final	0.758	6.983	4.962	5.573	1.364	1.072	1.308	1.177

Forthcoming Research

Vivamus molestie, risus tempor vehicula mattis, libero arcu volutpat purus, sed blandit sem nibh eget turpis. Maecenas rutrum dui blandit lorem vulputate gravida. Praesent venenatis mi vel lorem tempor at varius diam sagittis. Nam eu leo id turpis interdum luctus a sed augue. Nam tellus.