

# DAIT Project

Trajectory Prediction for Human-Human Interaction

---

Rodolphe Farrando, Romain Gratier

23.05.2018

EPFL

# Table of contents

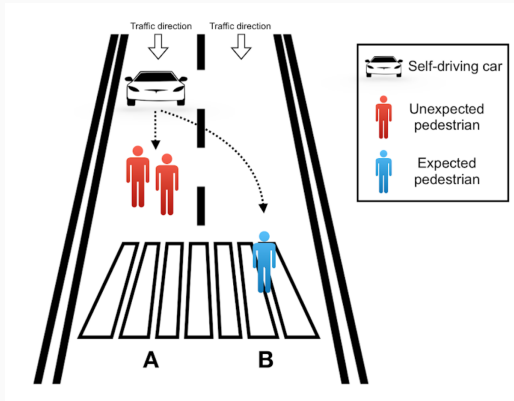
1. Introduction
2. Data
3. Models
4. Results
5. Representation

# Introduction

---

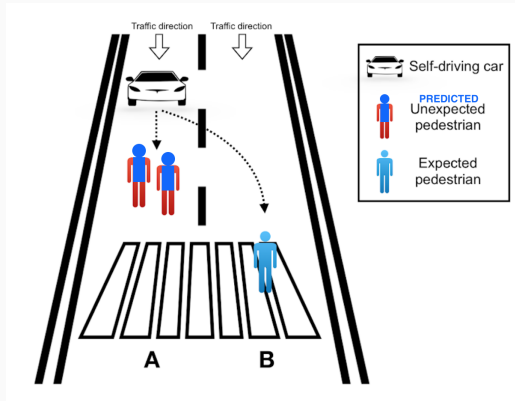
# Introduction

- Trajectory prediction is crucial for improving autonomous vehicles behaviour



# Introduction

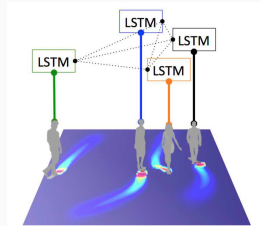
- Trajectory prediction is crucial for improving autonomous vehicles behaviour
- Could avoid situations seen in the ethical lectures



# Previous work Social LSTM : Human Trajectory Prediction in Crowded Spaces

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

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

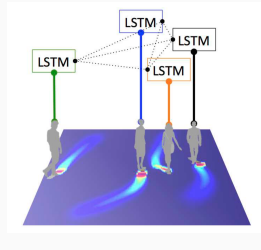


*Images from Social LSTM: Human Trajectory Prediction in Crowded Spaces and courses lecture*

## Previous work Social LSTM : Human Trajectory Prediction in Crowded Spaces

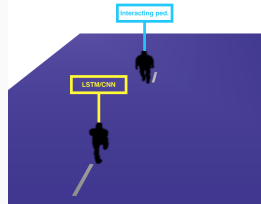
In their project, they used different components to make the structure:

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



In our project we only use:

- One CNN, or one LSTM
- Prediction per pedestrian



Images from Social LSTM: Human Trajectory Prediction in Crowded Spaces and courses lecture

# Data

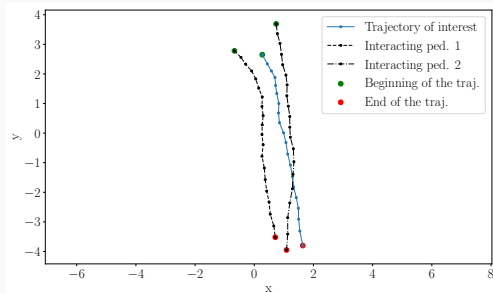
---



# Pre-processing

The preprocessing is divided in 4 steps:

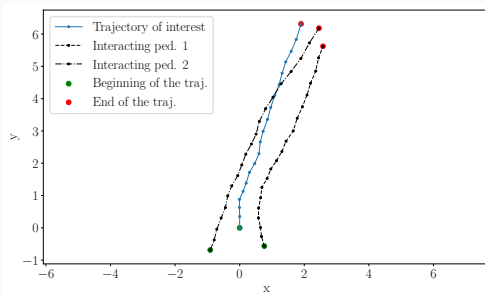
1. We isolate each trajectory along with his interaction, that is the other trajectories that are around within the same frames



# Pre-processing

The preprocessing is divided in 4 steps:

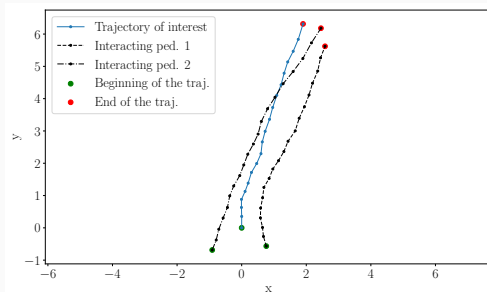
1. We isolate each trajectory along with his interaction, that is the other trajectories that are around within the same frames
2. We normalize the trajectories such that the first point is at  $(0,0)$  and the second is at  $(0, y_1)$



# Pre-processing

The preprocessing is divided in 4 steps:

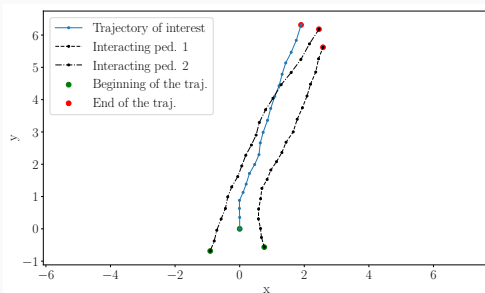
1. We isolate each trajectory along with his interaction, that is the other trajectories that are around within the same frames
2. We normalize the trajectories such that the first point is at  $(0,0)$  and the second is at  $(0, y_1)$
3. We calculate axis velocities  $V_x$  and  $V_y$



# Pre-processing

The preprocessing is divided in 4 steps:

1. We isolate each trajectory along with his interaction, that is the other trajectories that are around within the same frames
2. We normalize the trajectories such that the first point is at  $(0,0)$  and the second is at  $(0, y_1)$
3. We calculate axis velocities  $V_x$  and  $V_y$
4. For each frame, if there is a interacting pedestrian we add its coordinates and speed otherwise zeros are added



# Data

We have a file with:

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

Frame Number	ID	$x$	$y$	$V_x$	$V_y$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

# Data

We have a file with:

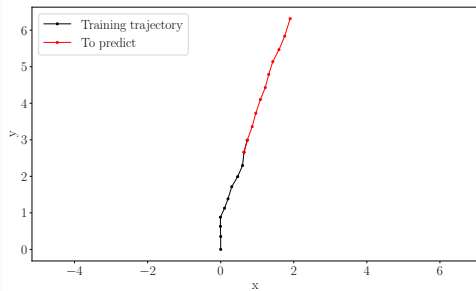
- Pedestrians ID
- Frame number
- Twenty sets of  $x$  and  $y$  coordinates per pedestrian

Frame Number	ID	$x$	$y$	$V_x$	$V_y$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

We want :

- Train on the 10 first coordinates
- Predict the next 10

Example of trajectory to predict



# Outputs structure

Finally our inputs have the following shape:  $[10, N, 4 * N_{inter}]$ , with

- 10: sequence length
- $N$ : The number of data
- $4 * N_{inter}$ : 4 (being the  $x$  and  $y$  coordinates and  $V_x$  and  $V_y$  velocities) times the number of pedestrians interacting with the one of interest.

The four different cases are:

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  $L_3 = (X - X_{t-1} + V_t * 0.4)^2$

The fourth case ensure that coordinates and speeds are not predicted independently.

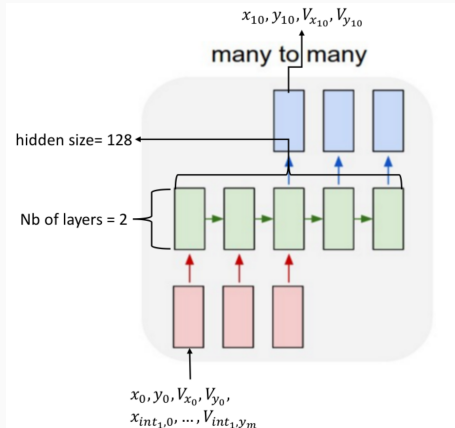
# Models

---



Define CNN

# LSTM



**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

# Results

---

## Results: Introduction

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

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

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

# Results

Results for both models:

- 3 Types of trajectory defined: Static, linear and non-linear trajectories
- Mean and Final displacements

		CNN				LSTM			
		Type 1	Type 2	Type 3	Total	Type 1	Type 2	Type 3	Total
Coord.	Mean					1.16	0.768	0.817	0.837
	Final					1.269	0.911	1.087	1.011
Speed	Mean					0.742	0.397	0.448	0.461
	Final					1.433	0.716	0.874	0.863
2 Loss	Mean					0.519	0.484	0.568	0.511
	Final					0.979	0.871	1.093	0.946
3 Loss	Mean					0.537	0.473	0.576	0.51
	Final					0.992	0.86	1.125	0.95

# Representation

---