# DAIT Project

## Trajectory Preduiction for Human-Human Interaction

Rodolphe Farrando, Romain Gratier

23.05.2018

EPFL

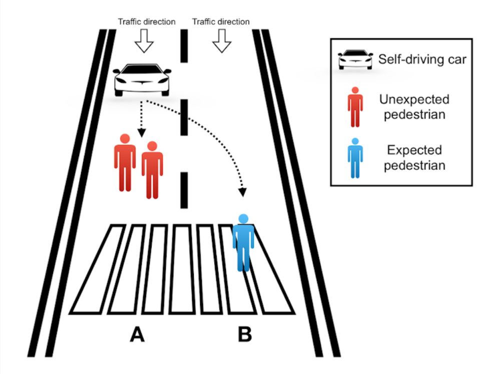
# Table of contents

# Introduction

## Introduction

Motivation:

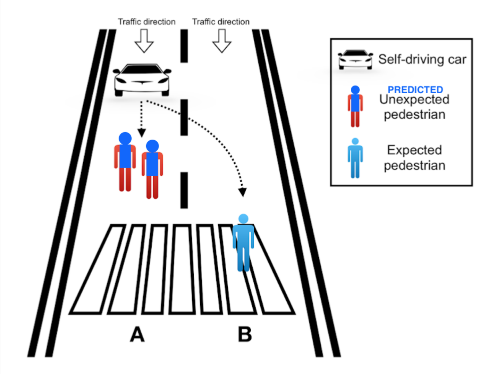
- Trajectory prediction is crucial for improving autonomous vehicles behaviour



# Introduction

Motivation:

- 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
- A Social Pooling
- per frame

In our project we only use :

- One CNN, or one LSTM
- per pedestrian

# Preprocessing and Postprocessing

# Preprocessing

We have :

- files with pedestrians id
- frame number
- coordinates

We want :

- future coordinates

Each trajectories are divided in two (two sets of 10 x and y coordinates):

- Training coordinates
- Ground truth
- We want to predict a sequence of 10 x and y coordinates such that they are close to the ground truth

# Preprocessing

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

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.

## Outputs structure

- The models can predict either coordinate or speed or both
- We test our two models for 4 different cases

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

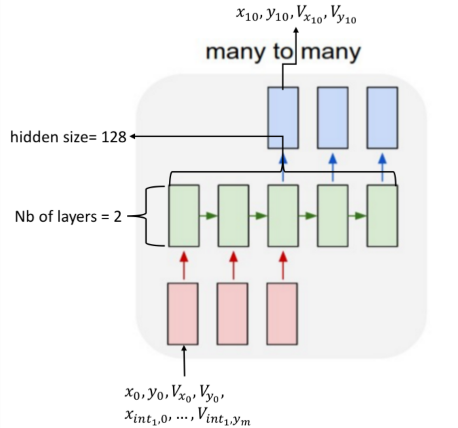
The fourth case ensure that coordinates and speeds are not predicted independently.

# Models

# CNN

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: LSTM

| | Model 3 | | | Model 4 | | |
|---|---|---|---|---|---|---|
| | Traj type | | | Traj type | | |
| | 1 | 2 | 3 | 1 | 2 | 3 |
| Mean disp. L2 | 0.519 | 0.484 | 0.568 | 0.537 | 0.473 | 0.576 |
| Final disp. L2 | 0.979 | 0.871 | 1.093 | 0.992 | 0.86 | 1.125 |

# Representation