Semester project report

Visual Intelligence For Transportation Laboratory

Human motion prediction

Author:
Ali Alami-Idrissi

Supervisor:
Pr. Alexandre Alahi

June 7, 2018

École Polytechnique Fédérale de Lausanne

Abstract

Pedestrians follow different trajectories to avoid obstacles and collisions with dynamic agents. Any autonomous robot navigating in such environments needs to learn the social rule that determines human walking behaviors. In this project, we explore different LSTM based models, first introduced by Alahi and al. [1] and which were specifically designed for forecasting human trajectories in crowded spaces. We also test their performances on the Trajnet trajectory forecasting challenge dataset[2]

1 Introduction

Humans have the ability to interact with their environment in order to walk safely in crowded spaces. For many tasks, intelligent robots need to understand these behaviors in order to make accurate predictions of their future actions and adopt a navigation style which is close to the one of humans. Joint pedestrian navigation pattern relies on multiples factors such as the relative distance to neighboring people and physical obstacles, the preferred velocity and the distance to the goal destination. Multiples research works have tried to tackle this challenge in the past by modelling the effect of the physical environment on the choice of human actions [3], or by modelling the interaction between multiple pedestrians in the same environment using models as the social force model [4] or gaussian processes [5]. However most of these approaches relies on hand-crafted functions to model human actions for specific settings and do not anticipate future action based on previously seen navigation patterns. In this project,we will explore different models that predicts human trajectories using recurrent neural networks. Our models will be based on Alahi and al. [1] paper and will be trained on the Trajnet challenge dataset [2]. We first start by presenting some related work. Then we move on to the presentation our methodology and models.We also lead an exploratory data analysis on the Trajnet dataset. And finally, we will show the results of our experiments.

2 Related work

Human trajectory prediction techniques can be grouped in two major categories:

Methods that models humans interaction with dynamic agent in their environment and other methods that models humans interaction with their physical environment.

2.1 Human-space interaction

Multiple research groups have tried to tackle the challenge of human-space interaction in the past. Pioneering work in human activity forecasting was made by Kitani and al [3]. Through their research, they showed that human trajectories can be forecasted by combining Inverse Reinforcement Learning and semantic scene understanding. Meanwhile,Walker and al .[6] used an unsupervised learning approach to predict future positions of moving agent in a scene based on information about the physical environment. Others [7] also tried to apply planning techniques to human motion forescasting.

2.2 Human-human interaction

First solutions use the social force model[8] which models pedestrian motion with attractive and repulsive forces. This model has shown conclusive results on walking pedestrians datasets. More recent social forces models use well engineered features to model different walking patterns. Alahi and al.[9] introduced a novel characterization of humans that describes the "social sensitivity" at which two humans interact. It captures both the preferred distance an individual wants to preserve with respect to his surrounding as well as the necessity to avoid collision. Meanwhile Yi and al.[10] proposed a social force model which includes stationary crowd groups as a key component. Other similar models model humans' interaction with their neighbors in a scene as gaussian processes [11]. However, these methods require hand crafted features and do not anticipate interactions that could occur in the more distant future. Hence, the models presented in this project aims to tackle this issue by using a data driven approach which relies on Long Short Term Memory Networks[12]

3 Methodology and models

Humans moving in crowded scene plan their motion based on their previous positions and on the positions of their neighbors. Hence, we will need to take into account these two factors in order to predict efficiently human trajectories. The models presented on this section are Long-short term Memory models based on Alahi and al. [1] paper and on the sequence generation technique presented by Alex Graves [13].

3.1 Problem formulation

Our aim is to construct a model which take as input (x_i^t, y_i^t) coordinates of people moving in a scene from time 1 to T_{obs} and predicts their positions for time $T_{obs} + 1$ to T_{pred}

3.2 Vanilla LSTM

Long Short-Term Memory (LSTM) networks have been shown to successfully learn the properties of temporal data. Hence, they will be useful in our problem setting to learn different motion patterns based on a limited number of previously seen trajectories.The vanilla LSTM has the following components:

- An Embedding with RELU non-linearity function and an embedding weight
- A single layer LSTM module with weights
- A linear layer of size 5. Which output the parameter of the Gaussian distribution from which we will sample our predicted velocities

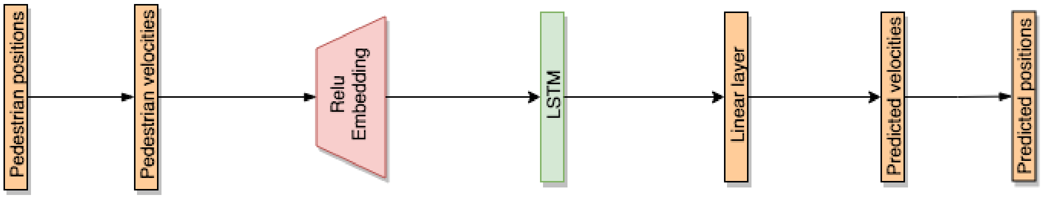


Figure 1: Illustration of the Vanilla LSTM model

3.3 Occupancy LSTM

The naive use of one LSTM model per person does not capture the interaction of pedestrians with dynamic agents around them. The occupancy LSTM solve this problem by introducing the notion of occupancy map.

Occupancy Map

The occupancy map is an additional matrix that helps the model take into account positions of neighbors in a scene and avoid collision. To compute our occupancy map,we start by discretizing our space into a grid of size $M \times N$ centered around the studied pedestrian for every frame so that every entry in the grid represents the number of neighbors located in the corresponding cell. The weights of every position in the grid are then computed using the following formula:

$$O_i^t(m,n) = \Sigma_{j \in N_i^t} 1_{mn}[pos(x_j^t - x_i^t, y_j^t - y_i^t)] \tag{1}$$

Where O_i^t represents the (m,n) position in the occupancy grid for pedestrian i at frame t, N_i^t represents the set of neighbors of pedestrian i at frame t, the function pos translates the given coordinates into the discretized space and 1_{mn} is the step function which is equal to one when $pos(x_j^t - x_i^t, y_j^t - y_i^t)$ is equal to (m,n).

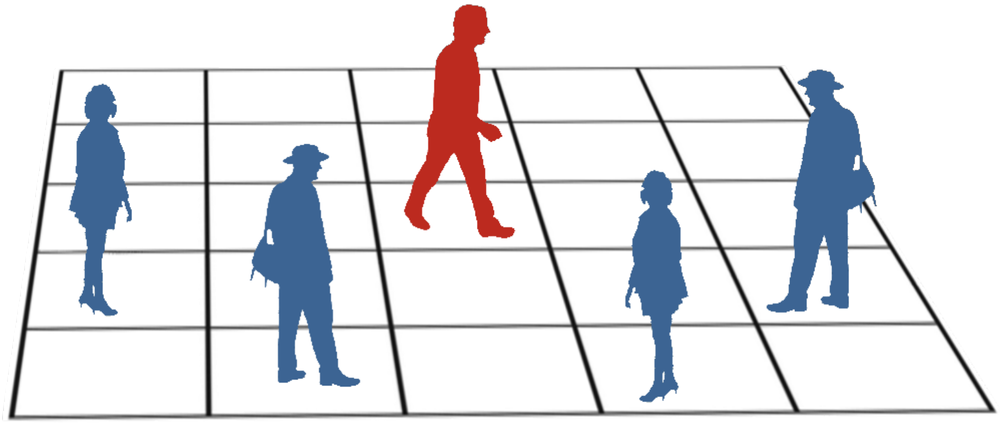


Figure 2: Illustration of the occupancy grid. The studied pedestrian is colored in red and its neighbors are colored in blue

Model

The OLSTM model is very similar to its vanilla counterpart except that we now also input the occupancy grids along with the studied pedestrian velocities for every frame. We also use the predicted velocities in order to build the occupancy grids during inference time.

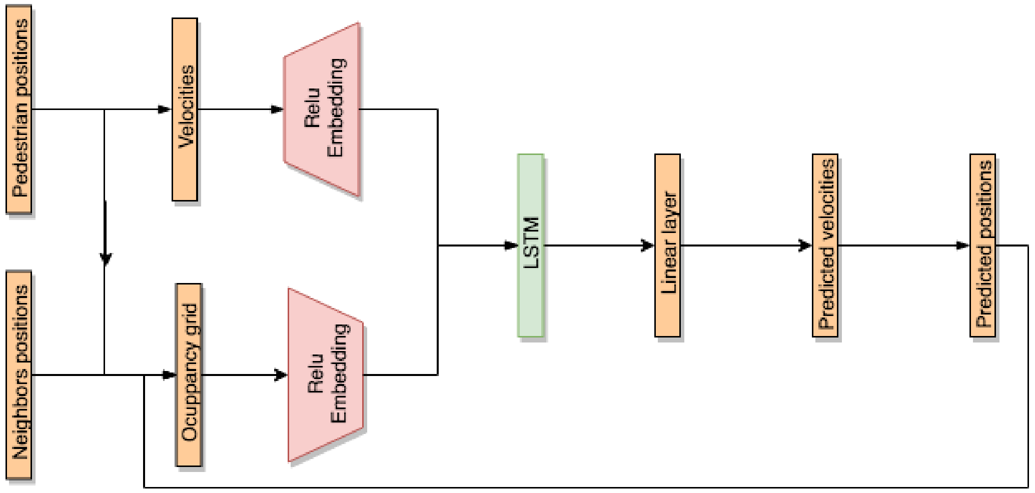


Figure 3: Block diagram of the OLSTM model

3.4 Social LSTM

The OLSTM has a more complete representation of its environment compared to the Vanilla LSTM as the occupancy grid adds information about dynamic pedestrians around the studied subject. However, it lacks information about future trajectories of neighboring pedestrians which limits its collision avoidance capabilities. The social LSTM tackles this issue by defining a new pooling strategy and introducing the notion of social pooling tensors.

Social pooling tensor

The social pooling tensor add a new dimension to the occupancy map which takes into account the hidden states of the studied pedestrian neighbors. The resulting tensor is of size $N \times M \times D$ where D is the length of the neighbors hidden state vector. We use the following formula to compute the weights of the social pooling tensor:

$$H_i^t(m,n,:) = \Sigma_{j \in N_i^t} 1_{mn}[pos(x_j^t - x_i^t, y_j^t - y_i^t)]h_j^t \tag{2}$$

Where H_i^t represents the (m,n) position in the occupancy tensor for pedestrian i at frame t, N_i^t represents the set of neighbors of pedestrian i at frame t, the function pos translates the given coordinates into the discretized space, 1_{mn} is the step function which is equal to one when $pos(x_j^t - x_i^t, y_j^t - y_i^t)$ is equal to (m,n) and h_j^t is the hidden state of neighboring pedestrian j at frame t. Hence, the vector at (m,n) is simply the sum of the hidden states of pedestrians standing in this position during the current frame.

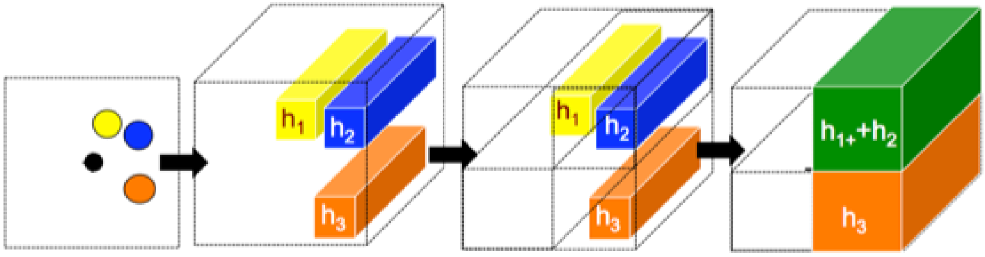


Figure 4: Illustration of the Social pooling tensor building process.Step 1 and 2 shows how the hidden states are collected, step 3 shows the space discretization process and step 4 shows how different hidden states of different pedestrians standing in the same cell are aggregated

Model

The social LSTM architecture builds upon the OLSTM by introducing new layers responsible for generating neighboring pedestrians hidden states. In Alahi and Al paper[1], the hidden states are generated by forwarding the neighbors velocities through the Social LSTM model. However, we slightly changed this approach by using a pretrained Vanilla LSTM model to generate the social pooling tensor hidden states. The block diagram in figure 5 gives a more detailed description of the complete model

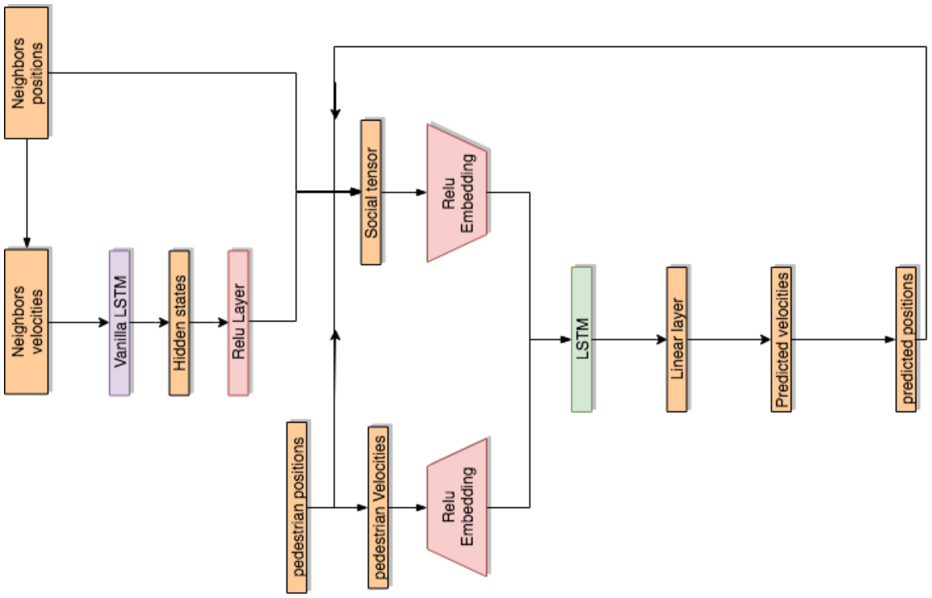


Figure 5: Illustration of the Social LSTM model

4 Dataset exploration

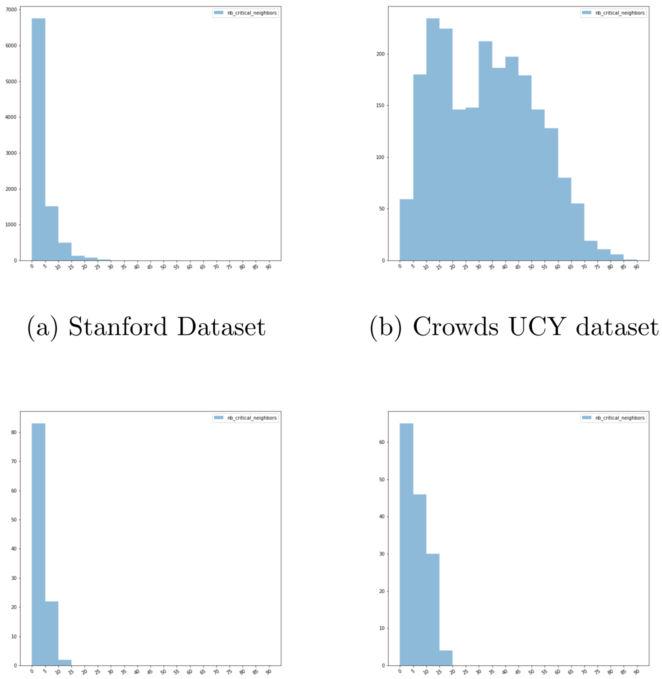
The dataset used in the scope of this project combines trajectories of walking pedestrian from 4 different public datasets segmented at a rate of 20 frames/trajectory where each frame is sampled at a framerate of 2.5fps.

Dataset	**Number of trajectories**
Stanford	8985
Crowds UCY	2211
BIWI	145
MOT	107

Table 1: Numbers of trajectories per dataset

4.1 Number of neighbors distribution

Since our models aims to learn how humans change their walking behavior with respect to their closeness to other pedestrians in their environment, we need an effective way to quantify the crowdedness in our data. An informative metric to measure this property is the number of neighboring pedestrians per trajectory. By looking at the results presented in figure 6, it seems that Crowds UCY is the only dataset with a high number of pedestrians per frame.



(a) Stanford Dataset

(b) Crowds UCY dataset

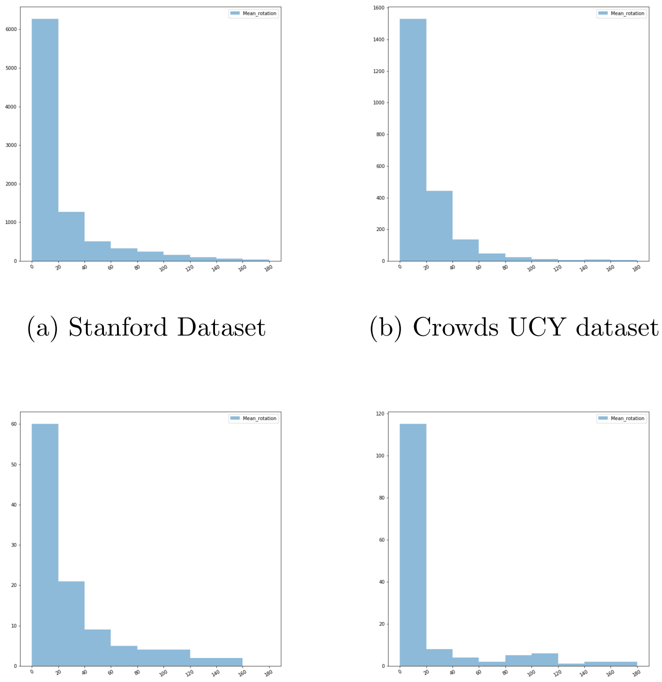
(c) MOT dataset

(d) BIWI ETH dataset

Figure 6: Distribution of the number of neighboring pedestrians within a radius of 6 meters around the studied trajectories

4.2 Non-linearity measure

The models presented in section 3 aims to learn non-linear walking behaviors. In order to accomplish this task, we need an efficient metric to quantify the non-linearity in each dataset. For this purpose, we computed the mean rotation of every pedestrian trajectory with respect to the vector between its first and second position. The histograms in figure 7 shows that the majority of trajectories in our datasets are quasi-linear. In fact the median mean deviation angle doesn't exceed 20 degrees for all the presented datasets.



(a) Stanford Dataset

(b) Crowds UCY dataset

(c) MOT dataset

(d) BIWI ETH dataset

Figure 7: Mean deviation angle with respect to the first velocity in the paths distribution

4.3 Static trajectories

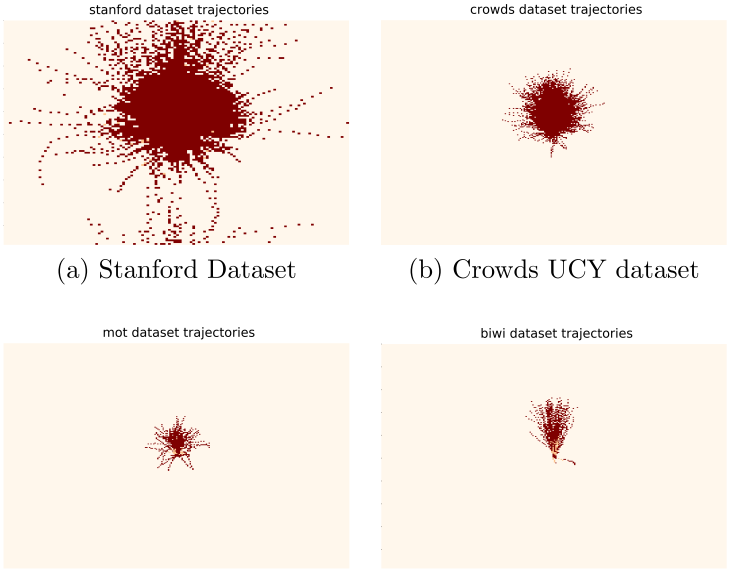
Static trajectories can bias our model since they do not allow us to effectively learn how to interact with other pedestrians in the environment. This is why we computed the number of static trajectories in every dataset.

Dataset	Number of non-static trajectories	Number of static trajectories
Stanford	7039	1946
Crowds UCY	1913	298
BIWI	115	30
MOT	104	3

Table 2: Number of static and non-static trajectories in every dataset

4.4 Trajectories spread

In order to compare the trajectories dispersion across the different datasets, we chose to plot every trajectory into a discretized map after preprocessing it according to the methodology described in section 5.1. The resulting maps plotted in figure 8 reveal a considerable difference between the Stanford dataset and the other datasets with respect to scaling and dispersion.



(a) Stanford Dataset

(b) Crowds UCY dataset

(c) MOT dataset

(d) BIWI ETH dataset

Figure 8: Mean deviation angle with respect to the first velocity in the paths distribution

5 Experiments

5.1 Data preprocessing

Before feeding the data to our models we rotate and shift every pedestrian and its neighboring trajectories so that the center track starts at the position (0,0) and its first velocity faces up. These steps aim to reduces the complexity of the prediction task and help the models to converge faster. Figure 9 gives an illustration of the data preprocessing step effect.

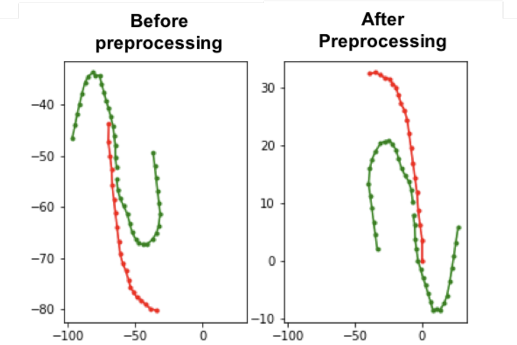


Figure 9: Illustration of the data preprocessing effect

5.2 Training and validation sets

We chose to discard the Stanford dataset from our training and validation sets since our data exploration described in section 4 suggests that the latter has a major scaling difference with other datasets available to us. Our experiments also showed that adding the Stanford dataset to our training set biases our metrics. We finally used 2163 samples for training our models and sampled 3 validations sets of 100 trajectory each from the MOT, UCY, BIWI and crowds datasets.

Our validation sets were specifically chosen so that we can track our model performance in different scenarios:

- The first validation set contains crowded scenes with multiple nonlinear trajectories
- The second validation set contains quasi linear trajectory with a low number of neighbors for every trajectory

- The third validation set contains only static trajectories

5.3 Experiments setting

The following settings were applied to all models:

- An embedding of size 64 for the spatial coordinates, the occupancy maps and the social tensor
- Hidden states of size 128
- Occupancy map and social tensor constrained to pedestrian that are located on a grid of 8.4 × 8.4 meters around our studied pedestrian
- We observe a pedestrian for 8 frames and predict its path for the following 12 frames
- Unless specified otherwise, the grid size for the occupancy map is 16×16

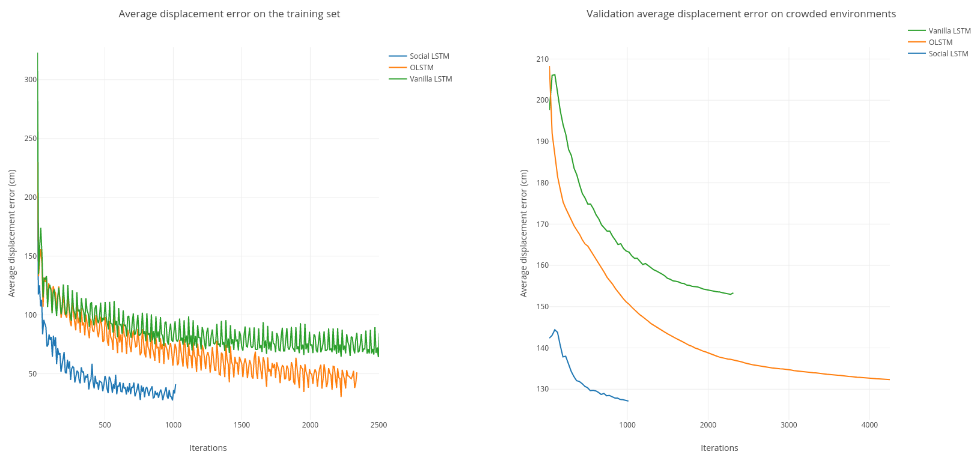
5.4 Metrics

Our models were benchmarked using two different metrics:

- **The average displacement error** which is the average L2 distance between the predicted trajectory and the true trajectory
- **The final displacement error** which is the L2 distance between the final position in the predicted trajectory and the true trajectory

5.5 Evolution of the training and validation average displacement error

In order to compare the performance of the Vanilla LSTM, OLSTM and the Social LSTM, we plotted their training and validation average displacement error in figure 10.
The resulting curves show clearly that the Social LSTM outperforms all the models presented above.



(a)

(b)

Figure 10: Average displacement error evolution on training and crowded validation set

Model	Average displacement error				Final displacement error			
	Training set	Crowded Validation Set	Non-crowded Validation set	Static Validation set	Training set	Crowded Validation Set	Non-crowded Validation set	Static Validation set
Vanilla LSTM with data preprocessing	81.1	111.2	111.4	19.1	160.1	237.1	216.1	34.9
Vanilla LSTM without data preprocessing	77.1	126.6	84.2	18.2	159.7	275.5	166.4	39.4
OLSTM with data preprocessing	47.8	126.2	93.6	12.4	88.6	258.6	184.2	21.8
OLSTM without data preprocessing	40.7	123	92	18.8	77.1	264.3	193.6	17.9
OLSTM With grid size 32*32	44.8	151.7	109.1	33.4	82.8	316.7	216.6	65.5
Social LSTM	33.3	121.7	88.7	10.4	50.5	254.3	178.6	20
Linear model	66.1	103.9	67.3	10	149.6	237.1	140.5	21.3

Table 3: Benchmarking results summary on the training and validation sets defined in section 5.2. Note that the linear model is a simple linear regression trained on the observed pedestrians' velocities.

5.6 Results

In table 3, we compare the performance of the models presented in section 3 under different experimental parameters. The resulting errors shows a slight improvement of our models performances when the data preprocessing described in section 5.1 is applying prior to training. They also show that increasing the grid size from 16 × 16 to 32 × 32 impacts negatively our OLSTM performance. Additionally, the social LSTM seems to outperform the Vanilla LSTM and OLSTM in crowded, non-crowded and static environments. However, all the presented LSTM models still under-perform the linear model on our validations sets. This may be due to our limited data size and to the fact that the validation sets doesn't contain enough non-linearity on the prediction regime.

Qualitative results

In this subsection, we will try to visually interpret the performances of our models by plotting the predicted trajectories and comparing them to the ground truth for different scenarios.

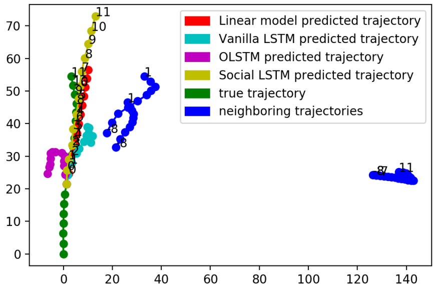
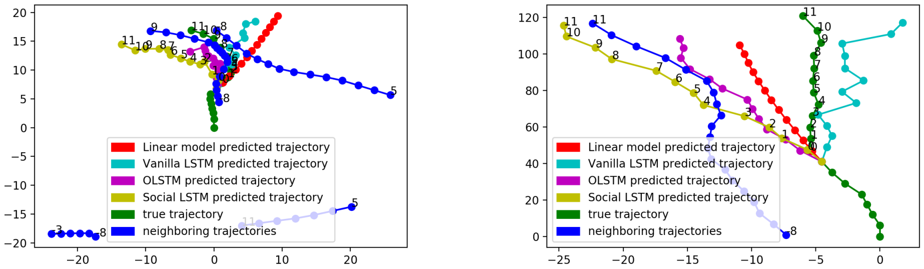


Figure 11: predicted trajectory by different models in the case of a linear walking behavior

The linear model seems to yield the best prediction results in the case where the observed pedestrian follows a linear path. However this simple baseline fails at predicting accurately complex trajectory. Figure 12 shows a concrete example where the linear model is outperformed by the OLSTM and the social LSTM. In this example, the Social LSTM is the best performing model as it predicts accurately the path curvature and keeps a reasonable

distance with the neighboring pedestrian.
Additionally, the social LSTM and OLSTM seems to be attracted by neighbors in some cases. This behavior can be observed in figure 12 where the latter models predict a shift toward a neighboring pedestrian.
Our models also lacks the ability to predict non-linear walking behaviors which are not due to interaction with dynamic agents in the scene. We can clearly see this problem in figure 13 where all models predict quasi-linear trajectories instead of a left turn. Such challenges can be overcome by feeding our models data about the surrounding physical scene features such as walls, pavements,buildings,etc...



(a)

(b)

Figure 12: predicted trajectory by different models in the case of a non-linear walking behavior(a) and in the case of a attraction effect (b)

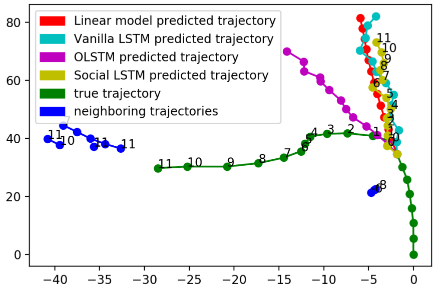


Figure 13: Predicted trajectory by different models in the case of a non predicable non-linear walking behavior

6 Conclusion

In this project, we tested the performances of the LSTM models presented in Alahi and Al. [1] on the Trajnet challenge dataset [2]. The lack of available data with occlusion and non-linear paths constrained our analysis and made it difficult to compare the performance of the presented models to a linear baseline. Future work will extend to collecting more data and building generative models that will help increase the performance of the studied LSTM models. In addition, including information about the surrounding space and the static physical obstacles into our prediction could improve our performances on non-linear trajectories. Finally, we believe that adding a penalty in function of the distance to neighboring pedestrian in our loss function can help our models improve their collision avoidance properties.

References

[1] Alahi, Alexandre, et al. ”Social lstm: Human trajectory prediction in crowded spaces.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[2] Trajnet Trajectory Forecasting Challenge.http://trajnet.epfl.edu/

[3] Kitani, Kris M., et al. ”Activity forecasting.” European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012.

[4] Pellegrini, Stefano, Andreas Ess, and Luc Van Gool. ”Improving data association by joint modeling of pedestrian trajectories and groupings.” European conference on computer vision. Springer, Berlin, Heidelberg, 2010.

[5] Trautman, Peter, et al. ”Robot navigation in dense human crowds: the case for cooperation.” Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.

[6] Walker, Jacob, Abhinav Gupta, and Martial Hebert. ”Patch to the future: Unsupervised visual prediction.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[7] Ziebart, Brian D., et al. ”Planning-based prediction for pedestrians.” Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on. IEEE, 2009.

[8] Helbing, Dirk, and Peter Molnar. ”Social force model for pedestrian dynamics.” Physical review E 51.5 (1995): 4282.

[9] Robicquet, Alexandre, et al. ”Learning social etiquette: Human trajectory understanding in crowded scenes.” European conference on computer vision. Springer, Cham, 2016.

[10] Yi, Shuai, Hongsheng Li, and Xiaogang Wang. ”Understanding pedestrian behaviors from stationary crowd groups.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[11] Tay, Meng Keat Christopher, and Christian Laugier. ”Modelling smooth paths using gaussian processes.” Field and Service Robotics. Springer, Berlin, Heidelberg, 2008.

[12] Hochreiter, Sepp, and Jürgen Schmidhuber. ”Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.

[13] Graves, Alex. ”Generating sequences with recurrent neural networks.” arXiv preprint arXiv:1308.0850 (2013).