LSTM networks for human trajectory prediction in simulated crowded scenes with embedded group and obstacle information

Roy Alia Asiku Md Azgar Hossain Shuvo
Department of Information Engineering and Computer Science)
University of Trento
Trento, Italy

***Abstract*—The analysis of crowded scenes is one of the most interesting research areas in visual surveillance due to the wide variety of factors that affect such an analysis: environment structure and details, occlusions, and obstacles for example. More traditional methods such as Kalman filters relied on one-step forecasting but solutions are emerging that use recurrent neural networks to learn spatial-temporal dependencies of moving agents in crowded scenes. Recent work has used Recurrent Neural Networks (RNNs) and group sociology to improve the performance of the trajectory prediction task by segmenting users based on the measure of the coherency of their motion. Pedestrians are clustered into a single unit if they show similar or coherent motion patterns. This clustering better models social dependencies between and among agents in a visual scene, thus improving the performance of the prediction task. We build on this work, extending it to simulated crowds and embeding additional information about the environment and social psychology to further improve the results of the Long Short Term Memory (LSTM) networks on the prediction task.**

***Index Terms*—Trajectory prediction, Group, Obstacle, LSTM-based, Simulated Crowd**

I. Introduction

Crowd analysis is an important topic in computer vision, with challenges ranging from crowd dynamics modelling, crowd segmentation, crowd activity classification, abnormal behaviour detection, to density estimation and crowd behaviour prediction tasks. [1] In prediction, we observe the motion histories of the subject in the scene and exploit that information. Traditional methods use one-step forecasting, an example is the Kalman filter, but this did not exploit all the available information in the input scenes. Recurrent Neural Networks have emerged as a viable solution to the prediction problem. Previously, researchers focused on anticipating the subject's future trajectory based on a precise motion history. This easily fails in dense environments due to occlusion. We can pay more attention to the whole scene on a macro scale instead of the micro interpretation, extracting coarse salient features. This is possible because people usually follow implicit or explicit social rules such as preservation of personal space, avoidance of collision, moving in groups and sticking to the group trajectory. Thus, people moving in a group can follow coherent motion patterns that can be exploited. [1] This work proposes an extension of the work done in [2]:

- Test the performance of [2] when the input video is a simulated crowd
- Choose the best epoch to produce the results and work on optimizing the LSTM architecture that's used.
- Add another tensor to encode additional information related to the scene or the subjects
- Improve UCY performance by conditioning grouping based on whole dataset metrics of social information
- Consider multiple neighbourhoods, not just one
- Consider that in each scene, there is a small number of actual intents overall of where a subject wishes to go, notwithstanding the obstacles, for example a bus stop, to go to sit down at a table, to take a walk down the road etc.

II. Related Work

A. Group analysis of crowds

Earlier approaches used a coherent motion clustering [3] of similar motion trends to group pedestrians. Clustering algorithms included the K-means clustering [4] and the Support Vector Clustering methods [5]. In the representation of collective activities, there are probabilistic representations [6] and quantitative descriptions of scene-independent descriptors such as collectiveness, stability, uniformity, and conflict [7].

B. Obstacle Avoidance

The literature considers the Social Force Model to bound obstacles to moving agents or the Fluid dynamics model in which they're considered as the boundaries to a fluid. Data-driven approaches have also come onto the stage recently to model and capture interactions among people and obstacles in crowd scenarios. The main objective consists in extracting local and global features for the crowd behaviour. Multiple works use vector fields to learn the velocity and navigation features of real scenes. [2]

C. Human Behaviour Prediction

The forecasting of social activities of crowds or groups of people has also gained some attention recently. This research domain involves trajectory prediction, interaction modelling, and contextual modelling. Helbing et al. [8] introduced the Social Force Model, which models interactions among agents

in a scene using Newtonian forces. The Continuum crowds model reproduces human interactions using priors [9], the Crowd is modelled as a Fluid, and agents are influenced by the position, goal, preferred speed, and a discomfort factor. Other models include the Social Affinity Maps that use Multi-view cameras [10], the Reciprocal Velocity Obstacle used in Robotics [11], and more recently, neural networks have been employed in the trajectory prediction task. Emerging deep generative models such as RNN, LSTM, and VAE (Variational Auto-encoders) solve the long-term prediction task directly [12] [13][14][15].

III. Our extension work

A. Background of the implementation

LSTM networks have shown good capabilities in predicting the behavior of pedestrians. The Social LSTM for example can capture the status of the neighborhood of each agent to refine trajectory prediction. This model employs the clustering of agents moving coherently and assigning agents to clusters. This social pooling allows agents to share their hidden state, enabling the network to make predictions based not only on the hidden state of an agent but also on those of other agents in the neighbourhood. This intuition derives from and obeys to the Social Force Model mentioned earlier. The neighborhood of the agent is described by a social pooling layer (Fig 1) defined as a tensor with dimensions related to the hidden-state dimension D and the neighbourhood size N. The prediction of the next position of the agent depends on the hidden state at the previous time-step. The predicted parameters are characterized by a bivariate Gaussian distribution (x and y) spatial parameters, and a correlation coefficient. The LSTM networks can be trained by employing the negative log-likelihood loss function for each agent or pedestrian. [2] proposed to exploit social relationships between pedestrians and embed this information in the model, and called it Group LSTM. Also proposed is Obstacle LSTM which extends the description of the current state of the agent with information relating to the fixed obstacles and semi-obstacles in the agent environment. Semi-obstacles are those obstacles preferably avoided by the pedestrian. Examples may include a flower bed around a tree, a meadow or a snowy part of a street. Obstacles on the other hand, cannot be avoided, examples may include trees, light poles or walls. This differentiation of obstacles from semi-obstacles is the driver of two experiments. These obstacles are annotated in the image plane and their coordinates are then projected to the ground plane in meters using available homography. Two versions of obstacles are defined: the presence of the obstacle, and the distance to the obstacle.

These considerations drive five different experiments:

- The Social LSTM
- Group LSTM distance to semi-obstacles.
- Group LSTM distance to obstacles.
- Group LSTM presence of semi-obstacles.
- Group LSTM presence of obstacles.

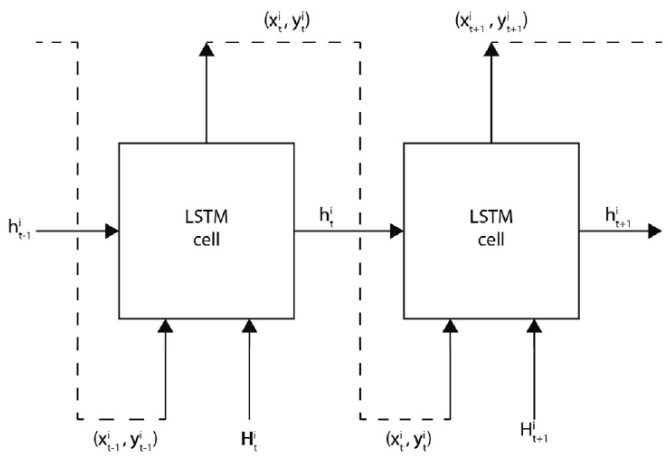

Fig. 1. The figure represents the chain structure of the LSTM network between two consecutive time steps, t and $t+1$. At each time step, the inputs of the LSTM cell are (x^i_{t-1}, y^i_{t-1}) the previous position and the Social pooling tensor H^i_t. The output of the LSTM is the current position (x^i_t, y^i_t)

B. Simulated data-sets

There has been an undeniable availability of data-sets for researching the pedestrian trajectory prediction problem. The most important setback with these data-sets is the annotation which is sometimes done manually, and thus time-consuming and expensive. There has been an emergence of tools and software for crowd simulations using artificial software agents to represent pedestrians. These tools model actual pedestrians with a stunning accuracy and thus can be used in the study of pedestrian tracking. In this work, we use such a simulated crowd movement dataset and demonstrate that the performance is in the same region as that of a real video.

Table 1 below reports the characteristics of the simulated data-set we tested.

Simulated Data-set characteristics	
Parameter	Value
Video Length	27 Seconds
Frame Width	2560
Frame Height	1440
Frame Rate	24 frames per second
Total number of frames	648

TABLE I
Simulated Data-set Characteristics.

We note that the video length is rather short compared with the other available open datasets. A short video results in little data input to the model and thus a poor performance. For a good experiment, it's vital to have video lengths comparable with the other datasets.

C. Data Pre-processing

The data required for the pedestrian tracking task includes: Frame ID, Pedestrian ID, x coordinate, y coordinate, Group ID. To get this, it was necessary to extract the frames from the video, do a detection of the objects in the video, filter out the persons in the frames, and track the detected person

across the frames. For the detection, we used YOLO-v3, and the Kalman filter-based Simple Online Real-time Tracking (SORT) tracker to track the detected pedestrians across frames. The image plane tracks are then projected to the floor plane in the following way. We calculate a Homography matrix by mapping points on the target floor plane to points in the image. We used this information to calculate the inverse perspective projection from image to ground plane. The Homography, together with an image Cartesian x and y coordinates are used to calculate the real-world floor coordinates.

IV. Results and Discussion

The simulated video data-set is short compared with the rest of the available datasets. This implies we cannot use the same default parameters as the rest of the datasets. To this end, we used the following parameters as a bare minimum to enable training.

Training parameters- Simulated Video	
Parameter	Value
Batch size	4
Sequence length	3
Prediction length	3

TABLE II
PARAMETERS FOR TRAINING THE SIMULATED CROWD DATASET.

For comparison, the default training parameters for larger datasets were as follows:

Training parameters- ETH UCY datasets	
Parameter	Value
Batch size	8
Sequence length	20
Prediction length	12

TABLE III
PARAMETERS FOR TRAINING ETH AND UCY DATASETS .

1) Rank Correlation of Results: We achieved a Spearman's Rank Correlation Coefficient of over 0.8 and a p-value 0.06 between the results in [2] and our results. These results are reported in Table IV for Average Displacement Error (ADE) and and Table V for Final Displacement Error (FDE).

2) Simulated Video Model Results: In table VI (ADE) and VII (FDE), we report the results for the simulated video dataset in comparison with the other datasets. The Figures 2 and 3 below help visualize the results better. It can be noticed that the simulated data performs overall worse than the other datasets, but interestingly the range of values is comparable to the other datasets. This implies that the simulated video is sufficient for use in our analysis of tracking problems. We can also note that all the models struggle with the UCY dataset. This could imply that the dataset shares some characteristics with the UCY dataset.

The Final Displacement Error shows are pattern rather different than what the ADE shows. There is no immediately discernible difference in performance between the results with ETH, UCY, and the Simulated datasets. This has profound implications, primarily that the detection and tracking worked for the simulated video as well as it did for the real datasets.

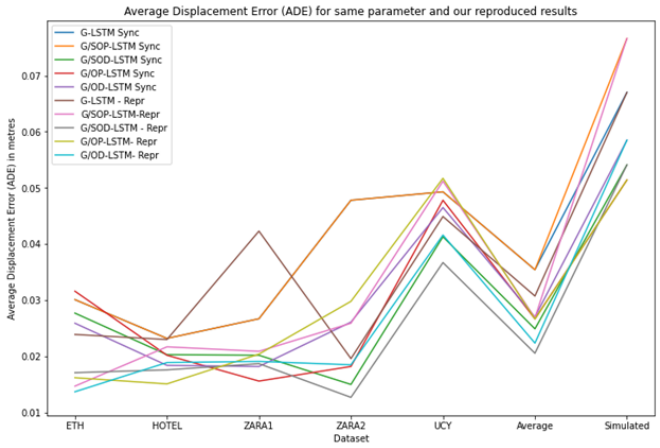

Fig. 2. Average Displacement Error for simulated dataset, ETH, and UCY. Sync denotes the use of same parameters for all models, Repr denotes our reproduced results.

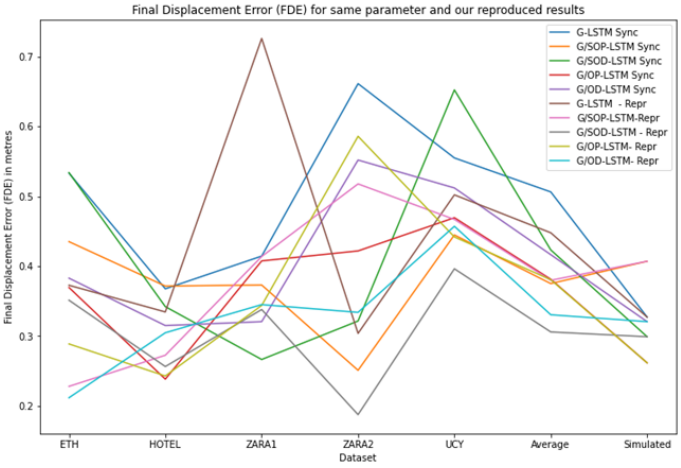

Fig. 3. Final Displacement Error for simulated dataset, ETH, and UCY.

We report below the performance of the model trained on the simulated video, superposing the true trajectory, predicted trajectory, and the actual scene from a frame (61) extracted from the video.

In Figure 4, the blue dots represent the true trajectories time step points, and the green points represent the predicted trajectories. In the absence of an overlap between the predicted and true trajectories, we can see the spatial separation between the two dots, representing the degree of error on that particular prediction. The model does fairly well on average since it correctly predicted the correct general trajectory of the crowd over a certain temporal horizon.

V. Conclusion

In this work, we reproduced the results in the reference papers, demonstrating high positive correlation of prediction results. We also did analysis and pre-processing of a simulated video dataset. This dataset was used as input for the training of a group LSTM trajectory prediction model and validated on

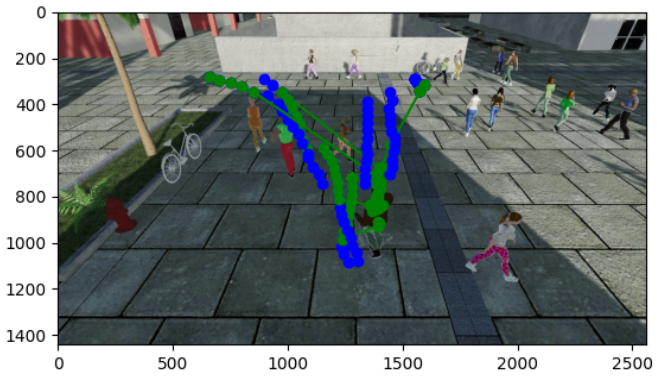

Fig. 4. True (blue) and Predicted (green) Trajectories of crowd.

the UCY dataset. As expected, the simulated dataset performs worse than the real dataset, most probably because they come from different domains. However, given that the range of results is comparable, this demonstrates that in the absence of large annotated real-world datasets, we can use simulated crowd generators to produce a usable large annotated dataset for the study of crowd trajectory prediction.

REFERENCES

[1] Bisango, N., Zhang, B., Conci, N.: Group LSTM: Group Trajectory Prediction in Crowded Scenarios (2018).

[2] Bisagno, N., Saltori, C., Zhang, B., De Natale, F., Conci, N.: Embedding group and obstacle information in LSTM networks for human trajectory prediction in crowded scenes. In: Computer Vision and Image Understanding 203 (2021).

[3] Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L.: Who are you with and where are you going? In: Proceedings of the International Conference on Computer Vision and Computer Vision. pp. 1345–1352. IEEE (2011).

[4] Zhong, J., Cai, W., Luo, L., Yin, H.: Learning behavior patterns from video: a data-driven framework for agent-based crowd modeling. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems. pp. 801–809 (2015).

[5] Lee, N., Choi, W., Vernaza, P., Choy, C., Torr, P., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. pp. 2165–2174. IEEE (2017).

[6] Ryoo, M., Aggarwal, J.: Stochastic representation and recognition of high-level group activities. International Journal of Computer Vision.

[7] Shao, J., Loy, C.C., Wang, X.: Learning scene-independent group descriptors for crowd understanding. IEEE Transactions on Circuits and Systems for Video Technology 27(6), 1290–1303 (2017).

[8] Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E 51(5), 4282 (1995).

[9] Treuille, A., Cooper, S., Popovi´c, Z.: Continuum crowds. vol. 25, pp. 1160–1168. ACM (2006).

[10] Alahi, A., Ramanathan, V., Li, F.F.: Socially-aware large-scale crowd forecasting. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. pp. 2203–2210. IEEE (2014).

[11] Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Robotics research, pp. 3–19. Springer (2011).

[12] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. pp. 961–971. IEEE (2016).

[13] Lee, N., Choi, W., Vernaza, P., Choy, C., Torr, P., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. pp. 2165–2174. IEEE (2017).

[14] Ballan, L., Castaldo, F., Alahi, A., Palmieri, F., Savarese, S.: Knowledge transfer for scene-specific motion prediction. In: Proceedings of the European Conference on Computer Vision. pp. 697–713. Springer (2016).

[15] Jain, A., Zamir, A., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatiotemporal graphs. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. pp. 5308–5317. IEEE (2015).

Metric	ADE						
Dataset	ETH	HOTEL	ZARA1	ZARA2	UCY	Simulated	Average
Lin. (Alahi et al,. 2016)	1.33	0.39	0.62	0.77	0.82		0.79
S-LSTM (Alahi et al,. 2016)	1.09	0.67	0.41	0.52	0.61		0.7
S-GAN (Gupta et al., 2018)	0.81	0.72	0.34	0.42	0.6		0.58
G-LSTM (Bisagno et al., 2018)	0.48	0.47	0.23	0.34	0.56		0.42
OP-LSTM	0.4	0.44	0.42	0.4	0.39		0.41
OD-LSTM	0.4	0.43	0.38	0.4	0.4		0.4
SOP-LSTM	0.46	0.46	0.37	0.34	0.38		0.4
SOD-LSTM	0.44	0.4	0.36	0.37	0.37		0.39
G/SOP-LSTM	0.36	0.34	0.18	0.26	0.73		0.37
G/SOD-LSTM	0.4	0.31	0.17	0.2	0.81		0.38
G/OP-LSTM	0.38	0.33	0.2	0.26	0.75		0.38
G/OD-LSTM	0.36	0.33	0.21	0.22	0.64		0.35
G-LSTM (Bisagno et al., 2018) - Reproduced	0.0239	0.023	0.0423	0.0196	0.0449	0.067	0.03074
G/SOP-LSTM-Reproduced	0.0147	0.0217	0.0209	0.0259	0.0512	0.0766	0.035167
G/SOD-LSTM - Reproduced	0.0171	0.0176	0.0187	0.0127	0.0367	0.0541	0.02615
G/OP-LSTM- Reproduced	0.0162	0.0151	0.0204	0.0298	0.0517	0.0514	0.030767
G/OD-LSTM- Reproduced	0.0137	0.0189	0.0191	0.0185	0.0416	0.0585	0.028383

TABLE IV
REPRODUCED AVERAGE DISPLACEMENT ERRORS

Metric	FDE						
Dataset	ETH	HOTEL	ZARA1	ZARA2	UCY	Simulated	Average
Lin. (Alahi et al,. 2016)	2.94	0.72	1.21	1.48	1.59		1.59
S-LSTM (Alahi et al,. 2016)	2.41	1.91	1.11	1.31	0.88		1.52
S-GAN (Gupta et al., 2018)	1.52	1.61	0.84	1.26	0.69		1.18
G-LSTM (Bisagno et al., 2018)	1.12	0.89	0.91	1.49	1.48		1.18
OP-LSTM	0.87	1.1	1.01	1.68	1.89		1.31
OD-LSTM	0.85	0.83	0.81	1.57	1.67		1.14
SOP-LSTM	1.19	0.86	0.86	1.35	1.74		1.2
SOD-LSTM	1.2	0.88	0.81	1.57	1.67		1.23
G/SOP-LSTM	0.78	1.08	0.8	1.15	1.75		1.11
G/SOD-LSTM	0.92	0.83	0.69	0.91	2.04		1.08
G/OP-LSTM	0.79	1.06	0.9	1.12	1.95		1.16
G/OD-LSTM	0.75	0.93	0.92	0.95	1.96		1.1
G-LSTM (Bisagno et al., 2018) - Reproduced	0.3726	0.3344	0.7261	0.3037	0.5021	0.327	0.42765
G/SOP-LSTM-Reproduced	0.2278	0.2724	0.4138	0.5178	0.4669	0.407	0.384283
G/SOD-LSTM - Reproduced	0.3512	0.2562	0.3379	0.1873	0.3962	0.2989	0.304617
G/OP-LSTM- Reproduced	0.2886	0.2427	0.3437	0.586	0.4418	0.2615	0.360717
G/OD-LSTM- Reproduced	0.2115	0.3046	0.3449	0.3338	0.4571	0.3205	0.328733

TABLE V
REPRODUCED FINAL DISPLACEMENT ERRORS

Metric	ADE						
Dataset	ETH	HOTEL	ZARA1	ZARA2	UCY	Simulated	Average
G-LSTM (Bisagno et al., 2018) - Sync	0.0301	0.0232	0.0267	0.0478	0.0493	0.067	0.03542
G/SOP-LSTM-Sync	0.0277	0.0203	0.0202	0.015	0.0413	0.0766	0.033517
G/SOD-LSTM - Sync	0.0316	0.0202	0.0156	0.0182	0.0478	0.0541	0.03125
G/OP-LSTM- Sync	0.5933	0.0165	0.0226	0.0252	0.0446	0.0514	0.1256
G/OD-LSTM- Sync	0.0259	0.0184	0.0182	0.0261	0.0465	0.0585	0.032267

TABLE VI
SIMULATED VIDEO MODEL AVERAGE DISPLACEMENT ERRORS

Metric	FDE						
Dataset	ETH	HOTEL	ZARA1	ZARA2	UCY	Simulated	Average
G-LSTM (Bisagno et al., 2018) - Sync	0.5333	0.3675	0.4144	0.6614	0.555	0.327	0.476433
G/SOP-LSTM-Sync	0.4351	0.3715	0.373	0.2508	0.4447	0.407	0.38035
G/SOD-LSTM - Sync	0.5339	0.342	0.2663	0.3214	0.6524	0.2989	0.402483
G/OP-LSTM- Sync	0.3694	0.2381	0.4075	0.4218	0.4694	0.2615	0.361283
G/OD-LSTM- Sync	0.3828	0.3149	0.3205	0.552	0.512	0.3205	0.40045

TABLE VII
SIMULATED VIDEO MODEL FINAL DISPLACEMENT ERRORS