The Journal of Engineering

The 2nd 2018 Asian Conference on Artificial Intelligence Technology (ACAIT 2018)

IET Journals
The Institution of Engineering and Technology

Pedestrian trajectory prediction via the Social-Grid LSTM model

eISSN 2051-3305
Received on 18th July 2018
Accepted on 26th July 2018
E-First on 8th November 2018
doi: 10.1049/joe.2018.8316
www.ietdl.org

Bang Cheng[1], Xin Xu[1] ✉, Yujun Zeng[1], Junkai Ren[1], Seul Jung[2]

[1]College of Intelligence Science and Technology, National University of Defense Technology, Changsha, People's Republic of China
[2]Department of Mechatronics Engineering, Chungnam National University, Daejeon, Republic of Korea
✉ *E-mail: xuxin_mail@263.net*

Abstract: In the design of intelligent driving systems, reliable and accurate trajectory prediction of pedestrians is necessary. With the prediction of pedestrians' trajectory, the possible collisions can be avoided or warned as early as possible by changing the behaviour of intelligent vehicles. The trajectory prediction problem can be considered as a sequence learning problem, in which one of the recurrent neural network (RNN) models called long short term memory (LSTM) has been regarded as a promising method. The authors present a new method for predicting the pedestrian's trajectory, which is called Social-Grid LSTM based on RNN architecture. The proposed method combines the human–human interaction model called social pooling and the Grid LSTM network model. The performance of the proposed method is demonstrated on two available public datasets, and compared with two baseline methods (LSTM and Social LSTM). The experimental results indicate that the authors' proposed method outperforms previous prediction approaches.

1 Introduction

Recently, the research in intelligent vehicles has made significant advances. As is well known, the pedestrian–vehicle interaction plays an important role on the research in making intelligent vehicles drive safely. Meanwhile, there are still some challenges in predicting future pedestrian trajectories by analysing their past trajectories. These challenges mostly come from the complex movement patterns of pedestrians, such as pedestrians have to stop or turn right immediately to avoid collisions with other pedestrians or vehicles. As shown in Fig. 1, there is a crowd scene of pedestrians. For humans, any individual position in the future few seconds can be predicted by considering their heading direction and walking speed information. However, it is not an easy task to predict trajectories for many pedestrians synchronously. Therefore, it is necessary to make intelligent driving systems learn to get the ability of predicting pedestrian trajectories based on labelled observation data.

In the existing pedestrian trajectory prediction research, the existing methods can be classified into two types: model-based prediction methods and recurrent neural network (RNN)-based methods. In model-based trajectory prediction methods, the mathematical functions were usually designed and specific pedestrian properties were defined [1, 2]. In general, model-based trajectory methods can only predict future pedestrian trajectories in a short time period and the trajectories were usually predicted inaccurately in complex scenes.

With the developments of deep learning, the long short term memory (LSTM) models [3] have been used to solve sequence learning and prediction tasks. In recent years, it has attracted much attention to apply LSTM-based deep learning methods to solve trajectory prediction tasks. In recent years, some LSTM-based approaches have been developed to predict pedestrian trajectories. The simplest method is to use one LSTM model for each pedestrian to predict trajectories. Another approach based on the LSTM is to incorporate the influence factors among neighbouring pedestrians, which is called Social LSTM [4]. However, the prediction errors in existing methods are usually large for complex scenes. To better avoid collisions between pedestrians with intelligent vehicles, it is important to develop new methods for reducing the trajectories prediction error.

In this paper, a new LSTM-based trajectory prediction method is proposed to reduce the prediction error of pedestrians, which is called Social-Grid LSTM. The proposed Social-Grid LSTM makes use of the LSTM cell structure by adding a social pooling operation to establish an influence relationship among neighbouring pedestrians. In the basic LSTM cell structure, we adopt the two-dimensional Grid LSTM architecture [5] which is different with the general LSTM structure in layer-to-layer parameter transfer mechanism. One innovation of the proposed Social-Grid LSTM method is that it integrates the human–human interaction model called social pooling and the two-dimensional Grid LSTM model. The performance of the proposed method was tested on two benchmark datasets, and the performance was compared with two popular trajectory prediction methods which include LSTM and Social LSTM. The experimental results show the advantages of the proposed method.

The remainder of the paper is arranged as follows. The second section introduces the research background including problem formulation and related works in pedestrian trajectory prediction. The third section will describe the details of our proposed Social-Grid LSTM model. The experiments and results analysis will be presented in the fourth section. In the end, in the fifth section, we will get the conclusion of the research and discuss some future works.

2 Research background

2.1 Problem formulation

At first, we will describe the pedestrian trajectory prediction problem as below. The inputs are the observed position coordinates and the outputs are the next future position coordinates. We assume that the spatial coordinates of all pedestrians in each scene are obtained at every different time instants. At time instant t, the ith pedestrian's position in the scene can be represented as (x_t^i, y_t^i). In general, firstly, we observe all pedestrians' history positions from $t = 1$ to $t = T_o$. At the next step, we predict the future trajectories of pedestrians from time $t = T_{o+1}$ to $t = T_p$. Therefore, the problem of pedestrian trajectory prediction can be defined as follows:

Inputs: Observed history trajectories.

$$X_i^o = [(x_1^i, y_1^i), (x_2^i, y_2^i), \ldots, (x_o^i, y_o^i)] \quad (1)$$

Outputs: Predicted next future trajectories.

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

$$X_i^{\mathrm{p}} = [(x_{\mathrm{o}+1}^i, y_{\mathrm{o}+1}^i), (x_{\mathrm{o}+2}^i, x_{\mathrm{obs}+2}^i), \ldots, (x_{\mathrm{p}}^i, y_{\mathrm{p}}^i)] \quad (2)$$

Pedestrians moving in the scenes take actions under the influence of their neighbours' behaviours. Therefore, we predict the future trajectories with a social pooling operation, and with a simple LSTM model, it will make some prediction errors. This motivates us to adopt a two-dimensional Grid LSTM model for reducing prediction errors.

2.2 Related works

As is known, in the research of intelligent vehicles, the interaction between pedestrian and vehicles is increasingly becoming an important and indispensable research problem. For safety considerations, the intelligent vehicle needs to select good driving strategies to avoid collisions with pedestrians and other static or dynamic obstacles. Therefore, the research on pedestrian trajectory prediction is critical for avoiding collisions between pedestrians with vehicles.

At present, the research on pedestrian trajectory prediction can be roughly categorised into two classes. The first class are traditional model-based methods. The second class are RNN-based methods. Considering human–human interactions, a social force model was proposed by Helbing and Molnar [1] to describe a pedestrian motion model. For modelling human–human interactions, Antonini *et al.* [6] proposed a discrete choice framework. Bonabeau [7] also presented an agent-based method to model behavioural patterns of individuals. Tay and Laugier [8] has proposed a Gaussian processes model to predict pedestrian smooth paths. Kooij *et al.* [9] presented a dynamic Bayesian network for pedestrian trajectory prediction, where spatial layout of the environment, situation and the pedestrian awareness were united as latent states on the switching linear dynamical system (SLDS) to predict pedestrian dynamics changes. Normally, the traditional model-based approaches rely on manually designed energy functions and hand-crafted factors. Those methods can only predict trajectories in a short term.

Recently, deep learning has received much attention for classification and prediction applications [10]. Although the RNN model has been applied to solve sequence learning tasks [11], it resulted in the problem of gradient vanishing or the problem of gradient exploding [12] when training with simple RNNs. Therefore, some RNNs' variants including LSTM [13] and Gated Recurrent Units [14] were proposed to sequence learning tasks and obtained better performance. The performance of LSTM has been demonstrated in machine translation, image captioning and so on. Park *et al.* [15] proposed a method based on LSTM to predict vehicle trajectory.

For the pedestrian trajectory prediction problem, it can be defined as a sequence learning task so that the RNN model can be taken into account for solving the task. Alahi *et al.* [4] proposed a LSTM-based networks model which is called Social LSTM. Each pedestrian in a scene is modelled with one LSTM for predicting trajectory and through the social pooling processing to share the information between each other. Besides, they proposed a trajectory prediction method which is called Social Attention [16]; this method captures the relative importance of each pedestrian. Moreover, for the traditional single-direction LSTM architecture, only the past information in a data sequence was considered. Another bidirectional LSTM has been proposed to predict trajectories, which takes both the past and future context into account [17]. Lee *et al.* [18] presented an RNN encoder–decoder framework which applies variational auto-encoder for predicting trajectories.

As we all know, the special gating mechanism is used in a LSTM network [13]. It means that the specific parts of input data are selected by the reading gate, writing gate and erasing gate from memory cell in the sequential direction. However, deep networks including RNN suffer from the problem that the inputs cannot be dynamically selected between layers in the depth direction. The Grid LSTM was proposed in [5], where a network settled in a grid of more than one dimension; and its architecture has shown good performance in character prediction, machine translation and image classification.

However, as far as we know, the Grid LSTM method has not been applied in trajectory prediction tasks. Furthermore, the Grid LSTM model needs to have the ability of considering the relationship among neighbouring pedestrians. Therefore, in this paper, a novel method called Social-Grid LSTM is proposed, which combines two-dimensional Grid LSTM with social pooling operation which can effectively estimate the neighbouring influences among pedestrians.

3 Social-Grid LSTM model

In the following, in order to improve the prediction accuracy of pedestrian trajectories in complex scenes, we will present a novel Social-Grid LSTM model which combines the social pooling operation and the two-dimensional Grid LSTM network. The details of the method are introduced below.

3.1 Overall structure of the Social-Grid LSTM model

LSTM network models have been shown to address the sequence learning tasks successfully. Therefore, we integrate a two-dimensional Grid LSTM model with the social pooling operation for the trajectory prediction tasks. Particularly, we use one Grid LSTM for each pedestrian in a scene. The Grid LSTM models

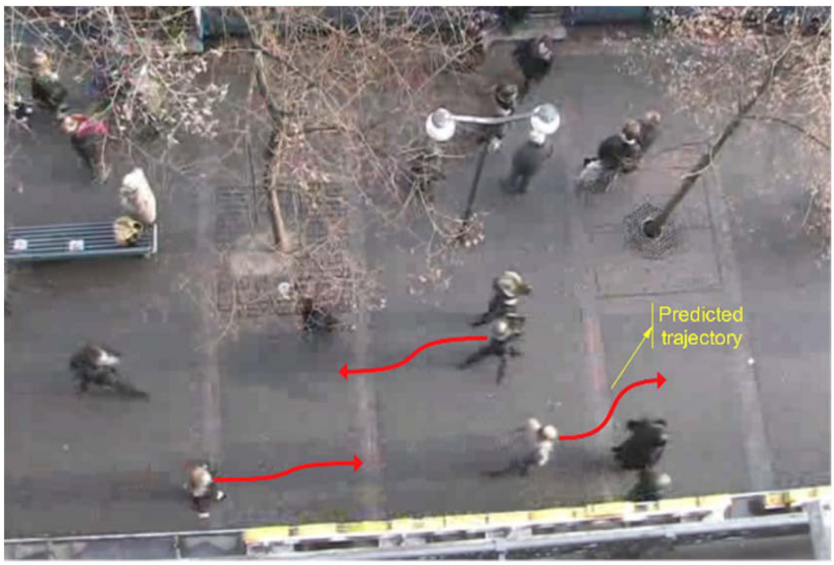

Fig. 1 *Example of a crowd scene with many pedestrians*

This is an open access article published by the IET under the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/3.0/)

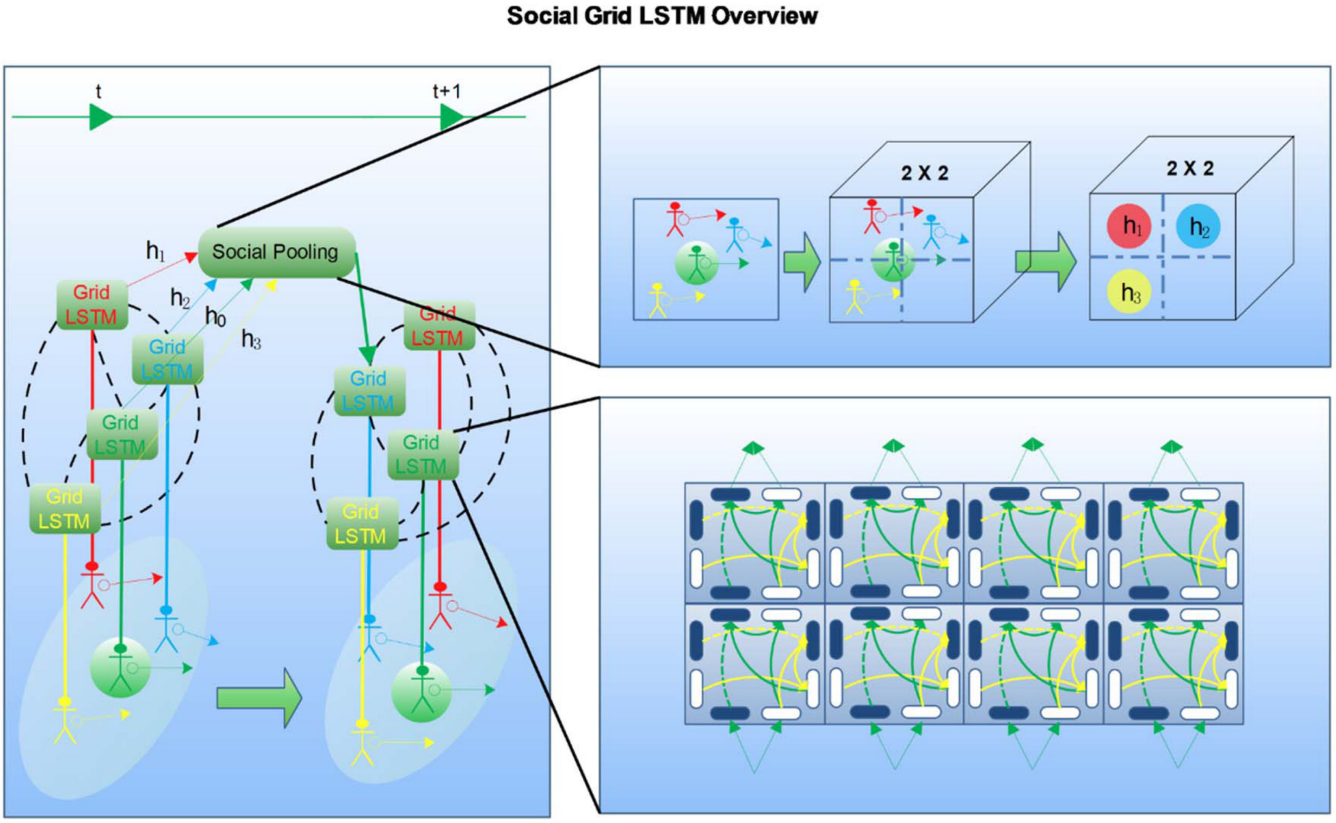

Fig. 2 *Overview of our proposed Social-Grid LSTM model. A separate two-dimensional Grid LSTM network is used for each pedestrian trajectory in a scene. The left shows the total structure of the model, pedestrians corresponding to Grid LSTMs are connected by the social pooling layer which makes the information shared with each other. The right top shows the details of social pooling operation with a 2 × 2 grid. The details of the structure of the two-dimensional Grid LSTM network are shown in the right bottom*

learn the state of the pedestrian by training on history data. Taking the interaction of pedestrians in a neighbourhood into account, we adopt the social pooling strategy by connecting the neighbouring Grid LSTM models. The overview of our model is shown in Fig. 2.

In general, the pedestrian individuals adjust their motions by taking the movements of neighbouring pedestrians into consideration. They are influenced by others in their current surroundings and would change the motion over time. Therefore, we share the states between the neighbouring two-dimensional Grid LSTM models. However, the problem is that the number of neighbours of each pedestrian is different in a crowd scene. Hence, we solve the problem by adding a social pooling layer, it is described in right top of Fig. 2. It means that the pooled hidden state information is received from the neighbours' Grid LSTM cells by Grid LSTM cell at every time step. In this paper, we adopt a 2×2 pooling grid to pooling the information of neighbours.

The hidden state of the Grid LSTM at time instant t of the ith pedestrian is defined as h_t^i. We represent the social pooled hidden state between neighbours with a tensor H_t^i. The hidden-state dimension is D, and neighbourhood grid size $N_0 = 2$. Therefore, the H_t^i can be defined as follows:

$$H_t^i(m,n,:) = \sum_{j \in N_i} 1_{mn}[x_t^j - x_t^i,\ y_t^j - y_t^i]h_{t-1}^j \tag{3}$$

where h_{t-1}^j is the hidden state of the Grid LSTM for the jth pedestrian at time instant $t-1$. To verify if (x, y) is in the (m, n) cell of the $N_0 \times N_0$ grid or not, we apply the function of $1_{mn}[x, y]$, and N_i is the set of pedestrian neighbours for the ith pedestrian.

The entire implementation process of our proposed method called Social-Grid LSTM is shown in the above Algorithm 1 (see Fig. 3). It mainly include two steps, the first is the process of model creation and the second is the training phase. Next, we will introduce some details of the two-dimensional Grid LSTM network model used in the proposed method.

3.2 Two-dimensional Grid LSTM for trajectory prediction

In this subsection, as a major step in the proposed Social-Grid LSTM method, we will apply the two-dimensional Grid LSTM method developed in [5] for training the model for pedestrian trajectory prediction. For the LSTM network model solving sequential problems, a series of input and output pairs are processed by the network, which can be represented as $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$. Pedestrian trajectory position data can be represented in this form. Therefore, for each pair (x_i, y_i), the output y_i is produced by both the last time output hidden value h_{t-1} and the current input x_i. The hidden value h_{t-1} is determined by all the previous inputs $x_1, x_2, \ldots, x_{i-1}$.

Meanwhile, as is shown in Fig. 4, the state of the cell in the network as memory vector $\boldsymbol{m}_t$ is determined by the previous inputs $x_1, x_2, \ldots, x_{i-1}$ and current input x_i. For each LSTM cell, the hidden state vector $\boldsymbol{h}_t$ is identified by the forgetting gate G_f, input gate G_i, output gate G_o. The brief structure is shown above, and the computation mechanism at every step in the cell was defined in [13] as follows:

$$G_\mathrm{f}^t = \sigma(\boldsymbol{U}_\mathrm{gf}[h_{t-1}, x_t] + \boldsymbol{v}_\mathrm{gf}) \tag{4}$$

$$G_\mathrm{i}^t = \sigma(\boldsymbol{U}_\mathrm{gi}[h_{t-1}, x_t] + \boldsymbol{v}_\mathrm{gi}) \tag{5}$$

$$G_\mathrm{c}^t = G_\mathrm{f}^t G_\mathrm{c}^{t-1} + G_\mathrm{i}^t \tanh(\boldsymbol{U}_\mathrm{gc}[h_{t-1}, x_t] + \boldsymbol{v}_\mathrm{gc}) \tag{6}$$

$$G_\mathrm{o}^t = \sigma(\boldsymbol{U}_\mathrm{go}[h_{t-1}, x_t] + \boldsymbol{v}_\mathrm{go}) \tag{7}$$

$$\boldsymbol{h}_t = G_\mathrm{o}^t \tanh(G_\mathrm{c}^t) \tag{8}$$

where σ is the logistic sigmoid activation function, $\boldsymbol{U}_\mathrm{gf}$, $\boldsymbol{U}_\mathrm{gi}$, $\boldsymbol{U}_\mathrm{gc}$, $\boldsymbol{U}_\mathrm{go}$ are the different weight matrices of the network, $\boldsymbol{v}_\mathrm{gf}$, $\boldsymbol{v}_\mathrm{gi}$, $\boldsymbol{v}_\mathrm{gc}$, $\boldsymbol{v}_\mathrm{go}$ are bias vectors. G_c^t is an intermediate variable. The H_{t-1} is

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

```
Initialize parameters: num_layers, lstm_size,batch_size, seq_length,num_epochs, learning_rate, grid_size.
Data = DataLoader()
##########################################
Create Model
##########################################
Construct the cell: cell = Grid2LSTMCell()
Define LSTM states and hidden output states
Compute gradients: gradients = tf.gradients()
Clip the gradients: grads, _ = clip_by_global_norm()
Define optimizer: optimizer = RMSPropOptimizer()
train_op = optimizer.apply_gradients
##################################
##########Training##############
############################
forepoch = 1 to num_epochs :
   Data.reset_batch_pointer()
forbatch = 1 to num_batches:
for b_size = 1 to batch_size:
        Get the source and target data: x_batch, y_batch
        grid_batch =getSequenceGridMask(), #Get the grid masks for all the frames in the sequence
        feed={model.input:x_batch, model.target: y_batch, model.grid_data: grid_batch}
train_loss=sess.run([model.cost,model.train_op],feed)
    end for
    Validate the current training batch
end for
Ouput the trained model
end for
```

Fig. 3 *Social-Grid LSTM*

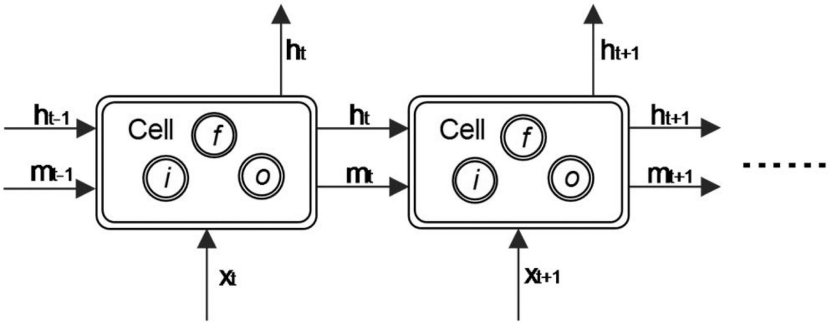

Fig. 4 *Brief architecture of long short term memory*

assumed to describe the concatenation of the previous hidden state h_{t-1} and the current input x_i:

$$H_{t-1} = [h_{t-1}, x_t] \tag{9}$$

The computation of each cell outputs a new hidden $\boldsymbol{h}_t$ and a memory $\boldsymbol{m}_t$. The output is computed by considering the hidden vector $\boldsymbol{h}_t$. It can be viewed as a functional LSTM():

$$(\boldsymbol{h}_t, \boldsymbol{m}_t) = \text{LSTM}(H_{t-1}, m_{t-1}, U) \tag{10}$$

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

where U is the concatenation of the weight matrices $\boldsymbol{U}_{\mathrm{gf}}$, $\boldsymbol{U}_{\mathrm{gi}}$, $\boldsymbol{U}_{\mathrm{gc}}$, $\boldsymbol{U}_{\mathrm{go}}$.

However, the Grid LSTM deploys cells along more than one dimension including the depth. In the proposed Social-Grid LSTM method, we use the two-dimensional Grid LSTM in which the cells are deployed along two dimensions, where the vertical one is along depth and the temporal one is for timing. The two-dimensional Grid LSTM can be viewed as a parameter transferring mechanism where the values cannot grow combinatorially in the cells.

The two-dimensional blocks in a two-dimensional Grid LSTM receives two memory vectors $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and two hidden vectors $\boldsymbol{h}_1$, $\boldsymbol{h}_2$ as inputs. The computation is shown below. The input two hidden vectors from each dimension are concatenated firstly as vector $\boldsymbol{H}$:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{bmatrix} \tag{11}$$

Then the cell computes with the two transform functions LSTM(), each function for each dimension, getting the expected outputs:

$$(h_t^1, m_t^1) = \mathrm{LSTM}(\boldsymbol{H}, m^1, U^1) \tag{12}$$

$$(h_t^2, m_t^2) = \mathrm{LSTM}(\boldsymbol{H}, m^2, U^2) \tag{13}$$

where the U^j of the each transform has distinct weight matrices $\boldsymbol{U}_{\mathrm{gf}}^j$, $\boldsymbol{U}_{\mathrm{gi}}^j$, $\boldsymbol{U}_{\mathrm{gc}}^j$, $\boldsymbol{U}_{\mathrm{go}}^j$.

Each transform function LSTM() applies the basic LSTM mechanism as (10) across the two dimensions. For a block, the grid of the network model is processed with the input of memory and hidden vectors two sides, and it outputs the memory and hidden vectors two sides at next time. However, considering that the input data are not separated to be sent to a block, the data along one of the sides of the grid will be processed by a pair of input memory and hidden vectors.

4 Experiments

4.1 Datasets and metrics

In this section, we mainly do experiments on two public pedestrian-trajectory datasets: ETH [19] and UCY [20]. The first includes two scenes and has two components: UNIV dataset and HOTEL dataset. The second also has two scenes; however, it is split into three small datasets: ZARA-01 dataset, ZARA-02 dataset and UNIV dataset. We do the experiment to evaluate our proposed model on the above five datasets. As is shown in [19, 20], these five datasets exhibit many complex interactions between pedestrians such as crossing each other, walking together, collision avoidance and groups dispersing and forming in the scenes. The datasets were provided in the form of series of combination of four elements which contains frame number, pedestrian ID, x-coordinate and y-coordinate. These datasets are recorded at 0.4 s per frame. In [4], the methods based on LSTM perform better than any other traditional approaches such as Linear Model [1], Social Force [2] and Iterative Gaussian Process [21]. Therefore, to compare the performance of the proposed method, we chose the LSTM and Social LSTM methods for performance comparisons.

To evaluate the performance of the different methods, we adopt two metrics which are given as follows:

- *Average displacement error:* It means that the mean Euclidean distance error over all estimated points at each time instant of the predicted trajectory and the true trajectory.
- *Final displacement error:* It means that computing the mean Euclidean distance error between the final points of the predicted trajectory and the true trajectory after T_{p} time steps.

4.2 Implementation details

In our proposed method, we adopt the Grid LSTM of two dimensions as shown in Fig. 2. The hidden state dimension is set to be 128 for the Grid LSTM which has two layers in the depth. Additionally, the embedding layers in the network model embed the input data into a 64-dimensional vector with rectified linear units non-linearity.

In the training, the batch size is set to be 16 and the network model was trained with the learning rate of 0.003. The optimisation function is RMS-prop function [22]. Besides, before training, the coordinates of two adjacent positions on the different datasets are preprocessed to the same interval number of frames by interpolating.

In order to take advantage of all five datasets while training the models, we adopt the approach of leaving one out. It means that we train the model on four of these datasets and test on the last dataset. The approach is repeated on all five datasets. At the same time, the two other methods used for comparison were trained and tested by the same procedure.

The frame rate is 0.4 s per frame. Therefore, during the testing stage, we observed eight frames and predict for the next 12 frames, it corresponds to observing a trajectory of 3.2 s and predicting for the future trajectory of the next 4.8 s. To compare the performance of different methods, the two above metrics were adopted.

4.3 Result analysis

The prediction errors in two metrics of all the methods on the five datasets are presented in Table 1. The best results are shown in bold. As we can see, the independent LSTM network model has large prediction errors which are due to not considering the pedestrian–pedestrian interactions. However, in the evaluation of the UCY-Univ dataset, the independent LSTM method outperforms slightly the other two methods. For our proposed Social-Grid LSTM method, it out performs both the Social LSTM method and the independent LSTM network model on all the five datasets in the two metrics.

In our comparison of other methods, by improving the LSTM network structure in depth with two-dimensional Grid LSTM, the trajectory prediction error has been reduced.

Table 1 Quantitative results of the three methods on all five datasets. The prediction error is in metre

Metric	Datasets	LSTM	Social LSTM	Social-Grid LSTM
average displacement error	ETH-UNIV	0.13	0.06	**0.06**
	ETH-HOTEL	0.16	0.11	**0.06**
	UCY-ZARA01	0.28	0.24	**0.19**
	UCY-ZARA02	0.37	0.34	**0.27**
	UCY-UNIV	**0.15**	0.20	0.16
	average	0.25	0.19	**0.15**
final displacement error	ETH-UNIV	0.22	0.13	**0.11**
	ETH-HOTEL	0.28	0.24	**0.09**
	UCY-ZARA01	0.51	0.37	**0.33**
	UCY-ZARA02	0.65	0.55	**0.52**
	UCY-UNIV	**0.23**	0.34	0.29
	average	0.38	0.33	**0.27**

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

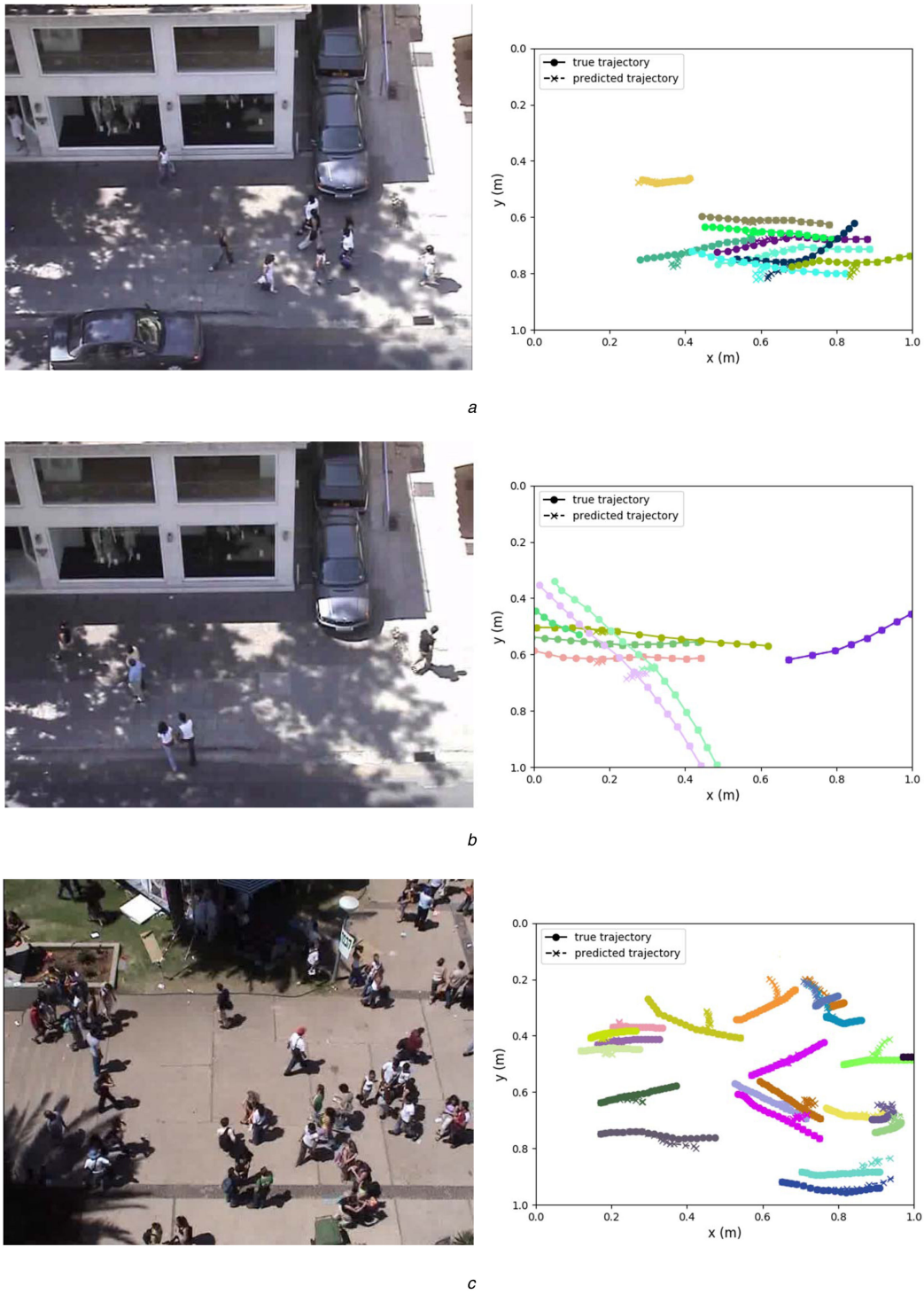

Fig. 5 *True and predicted trajectories on three crowd scene*
(a) UCY-ZARA01 scene, ***(b)*** UCY-ZARA02 scene and ***(c)*** UCY-UNIV scene. The right of the three subfigures are the predicted results. The lines with node '*o*' are the true trajectories. The predicted trajectories are presented with the lines of node '*x*'. The different colours of the trajectories represent different pedestrian in the scene

As shown in Fig. 5, we plot the true trajectories and predicted trajectories for all pedestrians in three of the scenes. The lines with node '*o*' are the true trajectories. However, the predicted trajectories are presented with the lines of node '*x*'. The different colours of the trajectories represent different pedestrian in the scene. From the result of the figure, it can be seen that the displacement errors between true trajectories and predicted trajectories are very small, and the errors are mainly generated at the end of the prediction time instant.

5 Conclusion

In this paper, a novel pedestrian trajectory prediction method called Social-Grid LSTM is proposed, which integrates the social pooling layer and the two-dimensional Grid LSTM network model. The social pooling layer learns the relative influence of each pedestrian in the crowded scenes by sharing the information between each network model corresponding to one pedestrian. We also analysed how the information is transformed between two layers in the depth of the Grid LSTM. It was demonstrated that the proposed method outperformed the Social LSTM method and the independent LSTM network model on two public datasets.

For the future work, we may take the next step of combining attention mechanisms [23, 24] to handle the trajectory prediction task. In addition, we will apply the proposed method into the advanced driver assistance system [25] of the intelligent vehicle by predicting the pedestrians' trajectories.

This is an open access article published by the IET under the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/3.0/)

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants U1564214, 61751311 and 61611540348.

7 References

[1] Helbing, D., Molnar, P.: 'Social force model for pedestrian dynamics', *Phys. Rev. E*, 1995, **51**, (5), p. 4282

[2] Yamaguchi, K., Berg, A.C., Ortiz, L.E., *et al.*: 'Who are you with and where are you going?'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, USA, June 2011, pp. 1345–1352

[3] Greff, K., Srivastava, R.K., Koutník, J., *et al.*: 'LSTM: a search space odyssey', *IEEE Trans. Neural Netw. Learn. Syst.*, 2015, **28**, (10), pp. 2222–2232

[4] Alahi, A., Goel, K., Ramanathan, V., *et al.*: 'Social LSTM: human trajectory prediction in crowded spaces'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, June 2016, pp. 961–971

[5] Kalchbrenner, N., Danihelka, I., Graves, A.: 'Grid long short-term memory', arXiv preprint arXiv: 1507.01526, 2015

[6] Antonini, G., Martinez, S.V., Bierlaire, M., *et al.*: 'Behavioral priors for detection and tracking of pedestrians in video sequences', *Int. J. Comput. Vis.*, 2006, **69**, (2), pp. 159–180

[7] Bonabeau, E.: 'Agent-based modeling: methods and techniques for simulating human systems', *Proc. Natl. Acad. Sci.*, 2002, **99**, (3), pp. 7280–7287

[8] Tay, M.K.C., Laugier, C.: 'Modelling smooth paths using Gaussian processes', in 'Field and service robotics' (Springer, Berlin, 2008), pp. 381–390

[9] Kooij, J.F.P., Schneider, N., Flohr, F., *et al.*: 'Context based pedestrian path prediction'. European Conf. on Computer Vision, Cham, 2014, vol. 8694, pp. 618–633

[10] Hao, X., Du, Q.H., Reynolds, M.: 'SS-LSTM: a hierarchical LSTM model for pedestrian trajectory prediction'. IEEE Winter Conf. on Applications of Computer Vision (WACV), Lake Tahoe, USA, March 2018, pp. 1186–1194

[11] Bock, J., Beemelmanns, T., Klösges, M., *et al.*: 'Self-learning trajectory prediction with recurrent neural networks at intelligent intersections'. Int. Conf. on Vehicle Technology and Intelligent Transport Systems, Porto, Portugal, April 2017, pp. 346–351

[12] Pascanu, R., Mikolov, T., Bengio, Y.: 'On the difficulty of training recurrent neural networks'. Int. Conf. on Machine Learning (ICML), Atlanta, USA, June 2013, vol. 28, pp. 1310–1318

[13] Hochreiter, S., Schmidhuber, J.: 'Long short-term memory', *Neural Comput.*, 1997, **9**, (8), pp. 1735–1780

[14] Chung, J., Gulcehre, C., Cho, K., *et al.*: 'Empirical evaluation of gated recurrent neural networks on sequence modeling', arXiv preprint arXiv:1412.3555, 2014

[15] Park, S.H., Kim, B.D., Kang, C.M., *et al.*: 'Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture'. IEEE Intelligent Vehicles Symp. (IV), Chang Shu, China, June 2018, pp. 1672–1678

[16] Vemula, A., Muelling, K., Oh, J.: 'Social attention: modeling attention in human crowds', arXiv preprint arXiv: 1710.04689v1, 2017

[17] Xue, H., Huynh, D.Q., Reynolds, M.: 'Bi-prediction: pedestrian trajectory prediction based on bidirectional LSTM classification'. Int. Conf. on Digital Image Computing: Techniques and Applications, Sydney, Australia, December 2017, pp. 1–8

[18] Lee, N., Choi, W., Vernaza, P., *et al.*: 'Desire: distant future prediction in dynamic scenes with interacting agents', arXiv preprint arXiv: 1704.04394, 2017

[19] Pellegrini, S., Ess, A., Schindler, K., *et al.*: 'You'll never walk alone: modeling social behavior for multi-target tracking'. IEEE 12th Int. Conf. on Computer Vision, 2009, pp. 261–268

[20] Lerner, A., Chrysanthou, Y., Lischinski, D.: 'Crowds by example', in '*Computer graphics forum*' (Wiley Online Library, 2007), vol. **26**, pp. 655–664

[21] Trautman, P., Ma, J., Murray, R.M., *et al.*: 'Robot navigation in dense human crowds: the case for cooperation'. IEEE Int. Conf. on Robotics and Automation (ICRA), 2013, pp. 2153–2160

[22] Dauphin, Y.N., Vries, H.D., Chung, J., *et al.*: 'RMSProp and equilibrated adaptive learning rates for non-convex optimization', CoRR, abs/1502.04390, 2015

[23] Vaswani, A., Shazeer, N., Parmar, N., *et al.*: 'Attention is all you need'. Int. Conf. on Neural Information Processing Systems (NIPS), 2017, pp. 1–11

[24] Xu, K., Ba, J., Kiros, R., *et al.*: 'Show, attend and tell: neural image caption generation with visual attention'. 32nd Int. Conf. on Machine Learning (ICML 2015), Lile, France, July 2015, pp. 2048–2057

[25] Geronimo, D., Lopez, A.M., Sappa, A.D., *et al.*: 'Survey of pedestrian detection for advanced driver assistance systems', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, **32**, (7), pp. 1239–1258

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)