SS-LSTM: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction

Hao Xue Du Q. Huynh Mark Reynolds
Department of Computer Science and Software Engineering
The University of Western Australia, Perth, Australia
hao.xue@research.uwa.edu.au, {du.huynh, mark.reynolds}@uwa.edu.au

Abstract

Pedestrian trajectory prediction is an extremely challenging problem because of the crowdedness and clutter of the scenes. Previous deep learning LSTM-based approaches focus on the neighbourhood influence of pedestrians but ignore the scene layouts in pedestrian trajectory prediction. In this paper, a novel hierarchical LSTM-based network is proposed to consider both the influence of social neighbourhood and scene layouts. Our SS-LSTM, which stands for Social-Scene-LSTM, uses three different LSTMs to capture person, social and scene scale information. We also use a circular shape neighbourhood setting instead of the traditional rectangular shape neighbourhood in the social scale. We evaluate our proposed method against two baseline methods and a state-of-art technique on three public datasets. The results show that our method outperforms other methods and that using circular shape neighbourhood improves the prediction accuracy.

1. Introduction

The problem of pedestrian future trajectory prediction based on deep learning method has renewed interest in recent years [1, 22, 14, 23, 26] in computer vision and artificial intelligence communities. This prediction is about generating pedestrians' future locations in terms of trajectories based on their previously observed trajectories. The prediction of pedestrians' trajectories in crowded scenes is highly valuable for social robot human-awareness navigation and intelligent tracking. However, to automatically predict pedestrians' trajectories is not an easy task for artificial intelligent systems because of the complex movement behaviours and clutter in crowded scenes.

Existing pedestrian trajectory prediction algorithms can be grouped into two categories: model-based methods and deep learning methods based on the Long Short Term Memory (LSTM) architecture. Model-based methods [8, 19, 28] depend on manually designed behavioural model functions and hand-crafted settings of pedestrian properties instead of learning pedestrian movement behaviours from the training data. These methods cannot reliably predict trajectories in more crowded and complicated scenes because it is difficult to combine all movement patterns into one model. For the LSTM-based methods [1, 6, 14, 23], incorporating the information from the pedestrians' neighbourhood in the training process and using scene context to refine trajectories have both been attempted. However, the layouts of the scenes, which have a more global effect on pedestrians' route planning during navigation, have not been studied widely.

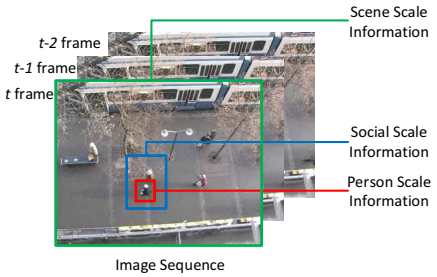


Figure 1. Our SS-LSTM network incorporates information from 3 different scales for pedestrian trajectory prediction: *person scale*, which captures individual pedestrian's past trajectory information; *social scale*, which captures the neighbourhood information of each pedestrian; and *scene scale*, which captures the scene layout features.

In this paper, we propose a hierarchical LSTM network consisting of three scales to overcome the above limitations. We name this network *Social-Scene-LSTM* (SS-LSTM). When pedestrians walk in a crowded place, other neighbouring pedestrians and scene layouts would affect their moving trajectories. For example, pedestrians usually keep a comforable distance from strangers but walk closer together with friends or family members; pedestrians would detour slightly to avoid obstacles or walk towards a particular exit. As shown in Figure 1, we tackle the pedestrian trajectory prediction problem using three different

Hao Xue would like to thank the support of an International Postgraduate Research Scholarship at UWA.

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00135

scales: the *person scale*, which captures each individual's past trajectory information; the *social scale*, which captures information in the neighbourhood around each pedestrian; the *scene scale*, which captures information about the scene layouts. Our work on using the social scale is inspired by the state-of-the-art Social-LSTM technique [1]. We use occupancy maps to extract neighbourhood features. We also use a more robust circular neighbourhood region instead of the widely used rectangle occupancy grids [1, 26]. This novel SS-LSTM network can automatically learn the social neighbour and the scene influences from data.

The contributions of this paper are: (1) proposing a novel hierarchical LSTM model for human trajectory prediction which has three hierarchical scales incorporating all the possible factors that influence pedestrians' navigation; (2) implementing three different categories of occupancy maps (*grid maps*, *circle maps* and *log maps*) to fully model social scale human-human interactions and comparing the prediction performance of these occupancy maps.

2. Related Work

2.1. Model-Based Trajectory Prediction

The original social force model [8] is proposed to model the movement behaviours of pedestrians. Three forces are included in this social force model: acceleration towards the desired velocity of motion, repulsive force and attractive force. Based on the original social force model, Yamaguchi *et al*. [28] improved the model for trajectory prediction by exploiting more behaviour factors such as damping, collision and social interactions. Another behaviour factor that has been used in the trajectory research literature is the *time-to-collision* [11] factor which describes the time duration before two pedestrians collide if they continue at their current walking velocities.

Agent-based modelling [3, 13] has been used to model behavioural patterns of pedestrians as well. Yi *et al*. [29] incorporated the stationary crowd group factor, moving pedestrians factor, and scene layout factor together in a novel agent-based model to improve forecasting performance in dense crowds. Pellegrini *et al*. [19] proposed a Linear Trajectory Avoidance (LTA) model for short term pedestrian trajectory prediction. Recently, Vemula *et al*. [26] proposed an interaction model to describe cooperative human behaviour in crowded scenes based on the Interacting Gaussian Processes (IGP) model [25].

A drawback of model-based methods is they largely depend on hand-crafted factors like the preferred walking speeds of pedestrians. Moreover, it is not straightforward to combine all trajectory influence factors into one single model. This limits the application of model-based methods for trajectory prediction in crowded scenes.

2.2. LSTM-based Trajectory Prediction

Recurrent Neural Networks (RNNs) have been designed to deal with time sequence data based on the recurrent architecture in the network. However, when the distance between the unit holding the relevant information and the unit where the information is needed becomes wider, RNNs have difficulties in learning to connect the information because of the gradient vanishing or exploding issue [18]. Thus, by introducing a three-gate architecture (input gate, forget gate and output gate), Long Short Term Memory (LSTM) networks [9] have been designed to improve the original RNNs. Recently, both RNN and LSTM have proved their successes in time sequence data processing areas such as speech recognition [7, 20], language translation [24], action recognition [30, 16] and image captioning [12, 4].

Intuitively, the trajectories of pedestrians can be considered as time sequence data, so LSTMs can be used for predicting trajectories of pedestrians. Compared with model-based methods, using LSTM for trajectory prediction is a more generic data-driven approach. Alahi *et al*. [1] suggested a Social-LSTM model which combines the behaviour of other people within a large neighbourhood. However, this Social-LSTM does not include important scene context information in prediction. More recently, the DESIRE (Deep Stochastic Inverse optimal control RNN Encoder-decoder) framework proposed by Lee *et al*. [14] uses scene context fusion to rank and refine the generated trajectories instead of incorporating scene information into the trajectory prediction process. Compared to Social-LSTM and DESIRE, our proposed hierarchical SS-LSTM takes into consideration both human-human interaction influences and scene scale features in the prediction process. Recently, LSTM-based methods have been used in citywide level applications like public transportation prediction [21] and location prediction problem [17]. Hierarchical LSTM architectures have also been used in contextual event prediction [10] and activity recognition [27]. However, to the best of our knowledge, a hierarchical LSTM architecture has not been designed for pedestrian trajectory prediction.

3. Our Method

In real applications, the route taken by a pedestrian is affected by the locations of other pedestrians in the neighbourhood. In complicated scenes, pedestrians would go around or detour slightly to avoid obstacles such as trees, garbage bins and benches. In our method, each pedestrian's past trajectory (i.e., the trajectory before the prediction time step) is referred to as the information at the *person scale*. We use two additional LSTMs to tackle trajectories that are affected by neighbouring pedestrians and by the scene layouts. We refer to them as the LSTMs at the *social scale* and the *scene scale*. Our proposed hierarchical

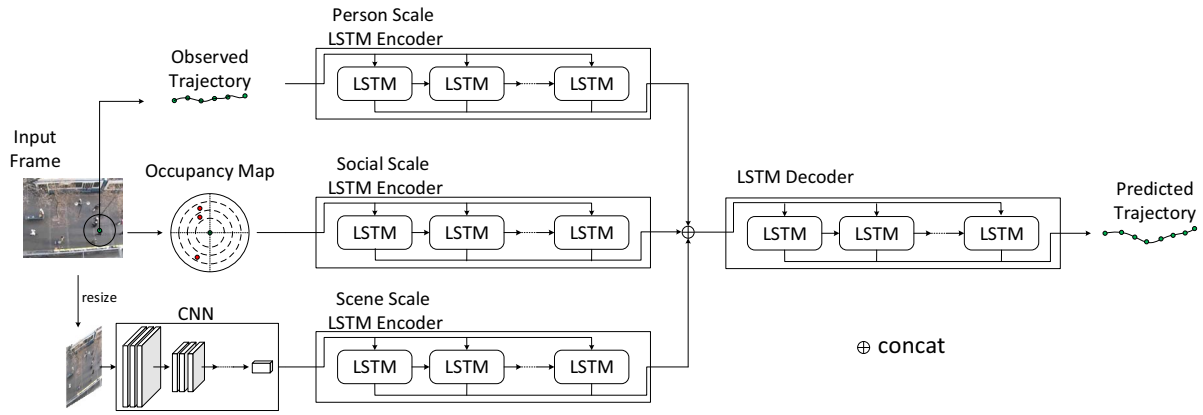


Figure 2. Pipeline of our proposed SS-LSTM network. There are three LSTM encoders for the three scales. The scene scale also includes a CNN. The encoded vectors are concatenated to form the input for the LSTM decoder to yield the predicted trajectory for each input observed trajectory.

framework which we coin *Social-Scene-LSTM* (SS-LSTM) is thus composed of three LSTM encoders for these three scales and one LSTM decoder for predicting the trajectory coordinates (Figure 2). Details of our proposed method will be described in the subsections following a brief review on LSTM.

3.1. Brief Review on LSTM

In a basic LSTM network architecture, given an input sequence represented by $(x_1, \cdots, x_T)$, the output sequence y_t can be obtained by iteratively computing Eqs. (1) and (2) for $t = 1, \cdots, T$:

$$h_t = \text{LSTM}\left(h_{t-1}, x_t; \mathbf{W}\right) \tag{1}$$

$$y_t = \mathbf{W}_{hy} h_t + b_y, \tag{2}$$

where the $\mathbf{W}$ terms denote the different weight matrices, b_y denotes the bias vector for the output y_t, and h denotes the hidden state. In the cell function $\text{LSTM}(\cdot)$, the hidden state is determined by the input gate i, forget gate f, output gate o, and the cell state c via the equations below:

$$i_t = \sigma\left(\mathbf{W}_{xi} x_t + \mathbf{W}_{hi} h_{t-1} + \mathbf{W}_{ci} c_{t-1} + b_i\right) \tag{3}$$

$$f_t = \sigma\left(\mathbf{W}_{xf} x_t + \mathbf{W}_{hf} h_{t-1} + \mathbf{W}_{cf} c_{t-1} + b_f\right) \tag{4}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(\mathbf{W}_{xc} x_t + \mathbf{W}_{hc} h_{t-1} + b_c\right) \tag{5}$$

$$o_t = \sigma\left(\mathbf{W}_{xo} x_t + \mathbf{W}_{ho} h_{t-1} + \mathbf{W}_{co} c_t + b_o\right) \tag{6}$$

$$h_t = o_t \tanh\left(c_t\right), \tag{7}$$

where $\mathbf{W}_{ab}$ is the weight matrix from layers a to b; $\sigma(\cdot)$ denotes the sigmoid activation function; each b term with a subscript is the bias vector for the appropriate layer.

3.2. Person Scale

The person scale LSTM encodes the observed trajectories which contain the basic information for trajectory prediction. At time step t, the i^{th} pedestrian's trajectory is represented by the image coordinates $\mathbf{X}_t^i = \left(x_t^i, y_t^i\right)$.

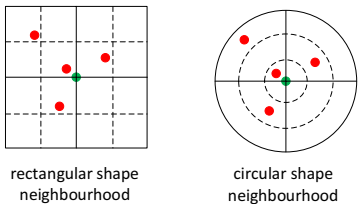


Figure 3. The traditional rectangular and our proposed circular occupancy maps. Occupancy maps are used to model the influence of other pedestrians (red) surrounding the pedestrian under study (green).

We observe the positions of all the pedestrians from time $t = 1$ to $t = \text{obs}$ and our aim is to predict their positions from $t = \text{obs} + 1$ to $t = \text{pred}$. The problem of trajectory prediction can therefore be defined as a sequence generation problem for the future trajectories $\mathbf{X}_{\text{pred}}^i = [(x_{\text{obs}+1}^i, y_{\text{obs}+1}^i), \cdots, (x_{\text{pred}}^i, y_{\text{pred}}^i)]$ from the input observed trajectories $\mathbf{X}_{\text{obs}}^i = [(x_1^i, y_1^i), \cdots, (x_{\text{obs}}^i, y_{\text{obs}}^i)], \forall i$.

For the person scale LSTM encoder, the observed trajectory coordinates $\mathbf{X}_{\text{obs}}^i$ of the i^{th} pedestrian is read in as input. The hidden state p_t^i for this pedestrian at time t is updated via:

$$p_t^i = \text{LSTM}_1^{\text{enc}}\left(p_{t-1}^i, x_t^i, y_t^i; \mathbf{W}_1\right), \tag{8}$$

where $\text{LSTM}_1^{\text{enc}}$ denotes the person scale LSTM encoder and $\mathbf{W}_1$ is the unknown weight matrix that would be estimated in the training process.

3.3. Social Scale

To capture the influence from other pedestrians in the neighbourhood, we build an occupancy map for each pedestrian to incorporate the social scale factor into our model. An occupancy map is built by partitioning the pedestrian's neighbourhood into non-overlapping cells. The spatial relationship between the i^{th} pedestrian and the surrounding neighbours at time step t is modelled in the occupancy map

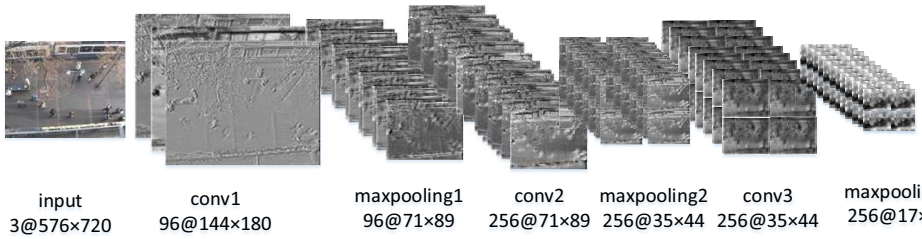


Figure 4. The convolutional architecture used in the scene scale CNN for extracting global features, comprising 3 convolutional layers, 3 maxpooling layers and 2 batch normalization layers (one after `maxpooling1`; one after `maxpooling2`). The feature maps at the different layers of the network capture both the static scene structure and the dynamic scene features induced by pedestrians' movements.

matrix O_i^t.

In the social scale LSTM, we adopt two neighbourhood shapes, as shown in Figure 3, to form three different categories of occupancy maps: *grid maps*, *circle maps* and *log maps*. For the *grid map* category, a rectangular neighbourhood is used; for the *circle map* and *log map* categories, a circular neighbourhood is used. The difference between a *circle map* and a *log map* is on how the radius of the map is defined. In a circle map, the linear scale for the radius is used; in a log map, the log scale is used. Compared to the traditional rectangular shape neighbourhood setting [1, 26], the circular neighbourhood is a more appropriate shape because the significance of social influence is mainly governed by the distance between the i^{th} pedestrian and the neighbouring pedestrians.

The size of an occupancy map is determined by the number of cells that partition the map. For example, in Figure 3, the size of the grid map is 4×4 and the size of the circle map is 3×4. Based on the number of neighbouring pedestrians occupying each cell, the occupancy map matrix O_i^t is computed as follows:

$$O_t^i\left(a,b\right) = \sum_{j \in \mathcal{N}^i} \alpha_{ab}\left(x_t^j, y_t^j\right), \tag{9}$$

where $\alpha_{ab}\left(\cdot,\cdot\right)$ is a discrimination function which classifies whether the coordinates of the j^{th} pedestrian are in the neighbourhood set $\mathcal{N}^i$ of the i^{th} pedestrian for the (a,b) cell of the occupancy map.

Using the occupancy map O_t^i as input, the social scale LSTM encoder computes the hidden state $s_{\text{o}}^{i,t}$ as follows:

$$s_{\text{o}}^{i,t} = \text{LSTM}_2^{\text{enc}}\left(s_{\text{o}}^{i,t-1}, O_t^i; \mathbf{W}_2\right), \tag{10}$$

where $\mathbf{W}_2$ is the corresponding weight matrix.

3.4. Scene Scale

Compared to the social scale, the scene scale has not received much attention in the literature of pedestrian trajectory prediction. Scene layouts such as entries, exits, stationary obstacles, etc, can be manually specified. However, a better alternative is to use a more generic data-driven method. Similar to the social scale which captures the local interactions between one pedestrian and his or her neighbours, we introduce a scene scale LSTM to capture the scene features in the prediction framework (see the Scene Scale LSTM Encoder in Figure 2). Another motivation of using the scene scale is that, while social scale focuses more on the pedestrian's local neighbourhood, scene features capture the global information of the scene for trajectory prediction. This global information is valuable when long trajectories need to be predicted.

For the scene scale, we train a CNN to extract scene features F_t of each frame at time step t. Unlike traditional CNNs which are trained for classification tasks, our CNN is specifically trained with other LSTMs in our framework for trajectory prediction. It contains three convolutional layers with maxpooling layers (Figure 4). Two fully connected layers are used after the convolutional part of CNN. Batch normalization layers are also used to avoid overfitting. The output F_t produced by the last fully connected layer is a 256 dimensional feature vector. Each input video frame that is passed to the CNN contains the moving pedestrians. As the camera capturing the scene is fixed, the changes in the CNN features between video frames are mainly induced by the pedestrians' movements.

When the CNN features extracted from the video frames are fine-tuned through an LSTM network targeting at time series prediction in our case, both static and dynamic scene features would be provided to the prediction network. Furthermore, this feature variation is in sync with the pedestrian information captured at the social and the person scales. This scene feature matrix is fed to our scene scale $\text{LSTM}_3^{\text{enc}}$ to compute the hidden state vector $s_{\text{c}}^{i,t}$ for the i^{th} pedestrian at time t:

$$s_{\text{c}}^{i,t} = \text{LSTM}_3^{\text{enc}}\left(s_{\text{c}}^{i,t-1}, F_t; \mathbf{W}_3\right), \tag{11}$$

where $\mathbf{W}_3$ is the associated weight matrix.

The next part of our proposed network is a merge layer $\varphi\left(.\right)$ which concatenates all the vectors p_t^i, $s_{\text{o}}^{i,t}$ and $s_{\text{c}}^{i,t}$ computed above, giving the hidden state h_t^i of the three scale LSTM encoders as follows:

$$h_t^i = \varphi\left(p_t^i, s_{\text{o}}^{i,t}, s_{\text{c}}^{i,t}\right) = p_t^i \oplus s_{\text{o}}^{i,t} \oplus s_{\text{c}}^{i,t}, \tag{12}$$

where $\oplus$ represents the concatenation operator. The concatenated state h_t^i is composed of all the past trajectory

Table 1. Quantitative results on the ETH and UCY datasets (in normalized pixels). All the methods predict trajectories for 12 frames using 8 frames' observed trajectories. The S-LSTM-g is identical to the Social-LSTM network of Alahi *et al*. [1].

Methods	ETHhotel		ETHuniv		UCYuniv		zara01		zara02		average	
	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Linear	0.137	0.261	0.143	0.298	0.099	0.197	0.141	0.264	0.144	0.268	0.133	0.257
LSTM	0.128	0.216	0.160	0.263	0.151	0.267	0.163	0.302	0.134	0.243	0.147	0.258
S-LSTM-g	0.076	0.125	0.195	0.366	0.196	0.235	0.079	0.109	0.072	0.120	0.124	0.169
S-LSTM-c	0.065	0.085	0.165	0.313	0.122	0.185	0.051	0.079	0.074	0.117	0.095	0.156
S-LSTM-l	**0.053**	**0.079**	0.149	0.295	0.099	0.159	0.052	**0.078**	0.069	0.104	0.084	0.143
SS-LSTM-g	0.081	0.129	0.170	0.291	0.108	0.157	0.057	0.089	0.072	0.109	0.097	0.155
SS-LSTM-c	0.066	0.101	0.182	0.328	0.098	0.144	0.051	0.081	0.065	0.107	0.092	0.152
SS-LSTM-l	0.070	0.123	**0.095**	**0.235**	**0.081**	**0.131**	**0.050**	0.084	**0.054**	**0.091**	**0.070**	**0.133**

information of pedestrian i, all the pedestrian neighbourhood information, and the scene layout information at time step t.

3.5. Trajectory prediction

To predict the trajectory coordinates of a pedestrian at a later time, the LSTM decoder $\mathrm{LSTM}^{\mathrm{dec}}$ in Figure 2 takes the encoded h_t^i as input and predicts the position $\left(\hat{x}_t^i, \hat{y}_t^i\right)$ of pedestrian i at time t via:

$$\hat{h}_t^i = \mathrm{LSTM}^{\mathrm{dec}}\left(\hat{h}_{t-1}^i, h_t^i; \mathbf{W}_d\right), \tag{13}$$

$$\left(\hat{x}_t^i, \hat{y}_t^i\right) = \mathbf{W}_o \hat{h}_t^i + b_o. \tag{14}$$

Similar to the basic LSTM in Eqs. (1) and (2), $\mathbf{W}_d$ and $\mathbf{W}_o$ are the weight matrices of the LSTM decoder and the output layer, and b_o is the bias term of the output layer.

3.6. Implementation Details

In our SS-LSTM models, all the LSTM layers have 128 dimensions and the hidden states have non-linear ReLU (Rectified Linear Units) activations. The input dimensions of the person, social and scene scale LSTMs are based on the length of the observed trajectories, the size of occupancy maps, and the size of CNN features matrix, respectively. In order to avoid overfitting, the dropout value is set to 0.2. The parameters of our proposed network are trained with a *RMSprop* optimizer [5] and the learning rate is set to 0.003. All the models are trained for 1000 epochs. Our SS-LSTM models are built using Python on Keras with a Tensorflow backend and trained with a NVIDIA GTX-1080 GPU (codes available at https://github.com/xuehaouwa/SS-LSTM).

4. Experiments

4.1. Datasets

ETH and UCY: ETH [19] and UCY [15] are two publicly available datasets of pedestrian trajectories that cover challenging movement patterns like walking together and collision avoidance. There are totally 5 subsets with hundreds of annotated pedestrian trajectories from these two datasets. These subsets are *ETHhotel*, *ETHuniv*, *UCYuniv*, *zara01* and *zara02*.

Similar to the setting adopted in Alahi *et al*. [1], we implement a leave-one-out splitting policy on the training/testing sets and use normalized pixel unit. During the experiments on these datasets, we train our network using 4 subsets and test it on the remaining 1 subset. The lengths of the observed trajectories and the predicted trajectories are 3.2 seconds and 4.8 seconds, which means we observe 8 frames and predict the next 12 frames. The neighbourhood and grid sizes are set to 32 pixels and 4 pixels, giving 8×8 grid occupancy maps. We use the same size of occupancy maps for *circle maps* and *log maps*.

Town Centre Dataset: The Town Centre dataset [2] contains hundreds of human trajectories in a real world crowded scene. The annotation file of the Town Centre dataset provides bounding boxes of the head and body of each pedestrian. In our experiments, the centres of body bounding boxes are considered as the trajectory coordinates. The video has 25 frames per second and the resolution is 1920×1080 pixels. We pre-process the annotated trajectories by down-sampling to every 5^{th} frame.

During the experiments, we also observe the first 8 frames (1.6 seconds, at 5 fps) of each trajectory. Because of the high resolution of the video and relatively larger pedestrians in the image (camera is close to people), we choose a 160×160 pixels neighbourhood and 20 pixels as the grid size. In this setting, the sizes of occupancy maps are the same as the occupancy maps used in the ETH and UCY experiments.

4.2. Baselines and Evaluation Metrics

We compare the performance of the following methods and different occupancy map settings:

- *Linear*: A baseline method based on linear regression. This method assumes that each pedestrian walks in straight

paths.

• *LSTM*: This is the basic vanilla LSTM-based trajectory prediction method which does not consider any social or scene scale information. This is the baseline method for LSTM-based trajectory prediction.

• *S-LSTM*: An LSTM-based prediction method that incorporates a social pooling layer identical to Social-LSTM proposed by Alahi *et al*. [1]. There are 3 variants: S-LSTM-g, S-LSTM-c, and S-LSTM-l, denoting respectively the use of *grid maps*, *circle maps* and *log maps*.

• *SS-LSTM*: This is our proposed Social-Scene-LSTM method. Similarly, there are 3 variants, denoted by SS-LSTM-g, SS-LSTM-c, SS-LSTM-l, depending on the occupancy maps used.

Because we adopt the setting in Alahi *et al*.'s public codes, which use a normalized pixel unit, we cannot compare our method with other pedestrian trajectory prediction algorithms that use other units (such as *metre*, which is used by Yamaguchi *et al*. [28]). We also do not compare with the DESIRE method [14] as different datasets were used there and the authors' computer code is not publicly available.

We use the average displacement error (ADE) [19] and the final displacement error (FDE) [1] as measures for evaluating the performance of pedestrian trajectory prediction. The ADE is defined as the average distance between the positions in the predicted and ground truth trajectories of all pedestrians. If the predicted destination is far away from the actual destination, the prediction is considered as a failure. The final displacement error (FDE) is the distance between the final destination of a predicted trajectory and the actual destination of the pedestrian. Compared to the average displacement error, the FDE focuses on the accuracy of predicted destinations of the pedestrians.

4.3. Quantitative Results

4.3.1 ETH and UCY

The quantitative results of our experiments are given in Table 1. Generally, our proposed SS-LSTM-l method outperforms other methods in most scenarios. Moreover, for the S-LSTM based methods, the *log maps* have better performance as well. From the ADE and FDE values, it is clear that *circle maps* and *log maps* outperform the traditional *grid maps*. The advantage of using the circular shape neighbourhood setting is evident. It is notable that even basic linear prediction method is comparable in some scenarios (like *ETHuniv* and *UCYuniv*). This is because the pedestrians' movement patterns are simple in these scenes. For example, if there are no obstacles between a pedestrian and his/her target destination, he/she would go straight towards the destination point.

From the average results, SS-LSTM-l has the best result with respect to both ADE and FDE. This means our method can generate better predicted trajectories and give more accurate destination prediction. Besides, all those methods which incorporate human-human interactions (both S-LSTMs and SS-LSTMs) perform better than the baseline methods (the linear and the vanilla LSTM methods). This shows that the neighbourhood surrounding a pedestrian has an extremely important impact on the trajectory prediction of that pedestrian. Our proposed hierarchical SS-LSTMs outperform the S-LSTMs with the same category of occupancy maps. These experiments show that scene scale information helps in trajectory prediction.

Table 2. Prediction errors of SS-LSTMs on the Town Centre dataset (in pixels). The length of the observed trajectories is 8 frames. The T_p values correspond to 4, 8, 12 and 16 frames, respectively.

T_p (secs)	Grid Maps ADE/FDE	Circle Maps ADE/FDE	Log Maps ADE/FDE
0.8	29.21/39.04	30.42/38.52	**29.01**/**36.88**
1.6	48.36/76.43	55.05/76.28	**43.31**/**69.49**
2.4	69.65/104.16	60.67/104.82	**52.90**/**91.58**
3.2	89.65/126.96	74.31/116.13	**71.94**/**107.66**

4.3.2 Town Centre

For the Town Centre experiments, we focus on the performance of SS-LSTM with different occupancy maps for different T_p values. In the trajectory prediction terminology, the term *prediction horizon* (introduced by Venula *et al*. [26]), denoted by T_p from hereon, is commonly used to describe the length of the predicted trajectories. In our experiments, given the same 1.6 seconds' length of observed trajectories, we predict the trajectories for a range of different prediction horizons, namely $T_p = 0.8, 1.6, 2.4, 3.2$ seconds. In the training and testing stages, trajectories shorter than 4.8 seconds (24 frames at the frame rate of 5) are filtered out.

The prediction results of SS-LSTMs (in pixel unit) are shown in Table 2. The occupancy map category giving the best performance for each T_p value is highlighted in bold. Although some FDE values are over 100 pixels, given that the image resolution is 1920×1080, a 100-pixel displacement error is considered to be acceptable. When the *prediction horizon* T_p is longer than the input observed trajectories, the use of *circle maps* and *log maps* can decrease ADE and FDE significantly. When the length of observed trajectories is larger than T_p, the performances from rectangular and circular neighbourhoods are about the same. This is because the differences between a rectangular shape neighbourhood and a circular shape neighbourhood are in regions that are far away from the pedestrian. For example, if the sizes of *grid map* matrix and *circle map* matrix are the same (both 8×8 in the experiment), people in the outermost part of the neighbourhood have the same distance from the predicted pedestrian in the circular shape. For the rectangu-

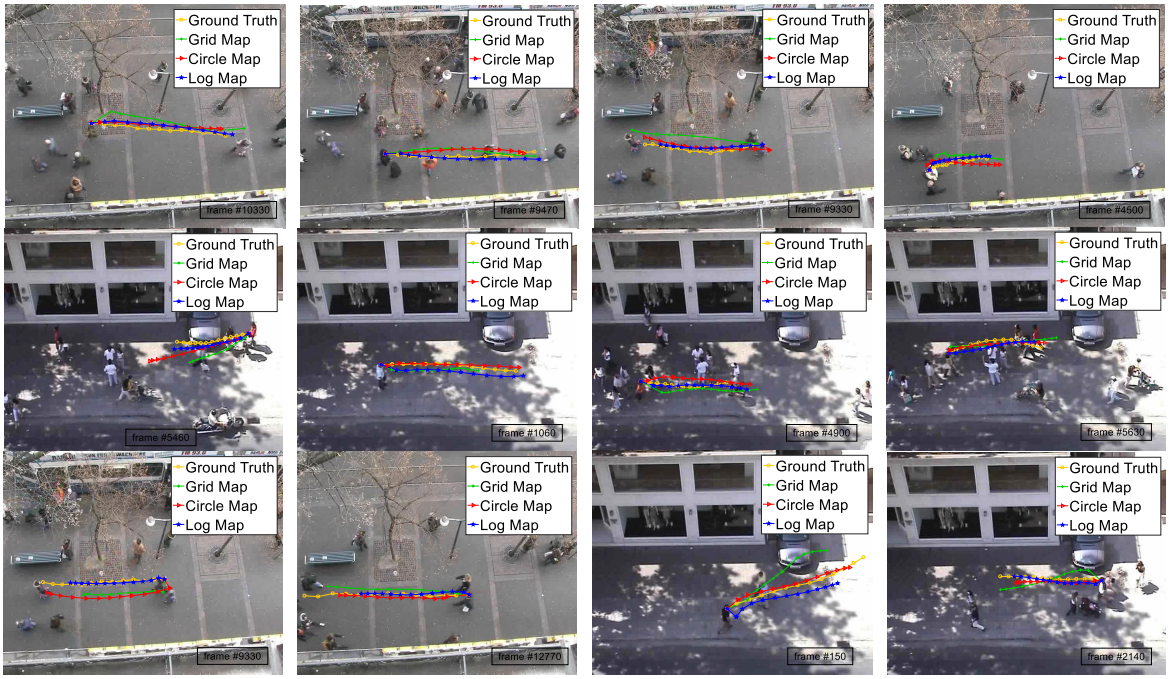


Figure 5. Illustration of a few predicted trajectories from our SS-LSTM method. The first two rows show the results on the ETH and UCY datasets. The last row shows four examples of slightly worse predicting results. The yellow trajectory in each subfigure represents the ground truth trajectory. Predicted trajectories using *grid map*, *circle map* and *log map* are shown in green, red and blue respectively (better viewed in color).

Table 3. Prediction errors on the Town Centre dataset for small and large prediction horizons.

T_p (secs)	LSTM ADE/FDE	S-LSTM-l ADE/FDE	SS-LSTM-l ADE/FDE
0.8	46.99/52.68	33.62/**36.62**	**29.01**/36.88
3.2	105.41/127.44	93.53/115.22	**71.94**/**107.66**

lar shape, people in the corners of the neighbourhood have a larger distance than pedestrians near the sides. Therefore, when T_p is small, all three categories of occupancy maps have nearly the same performance. When T_p is large, circular shape neighbourhood gives better performance than the conventional rectangular shape neighbourhood setting. Furthermore, as the amount of social influence depends on the distance between two pedestrians, *log maps* demonstrate to be better at depicting this kind of distance relationship. Thus, in general, *log maps* perform better than *circle maps* and *grid maps* in terms of the final displacement error for all the T_p values.

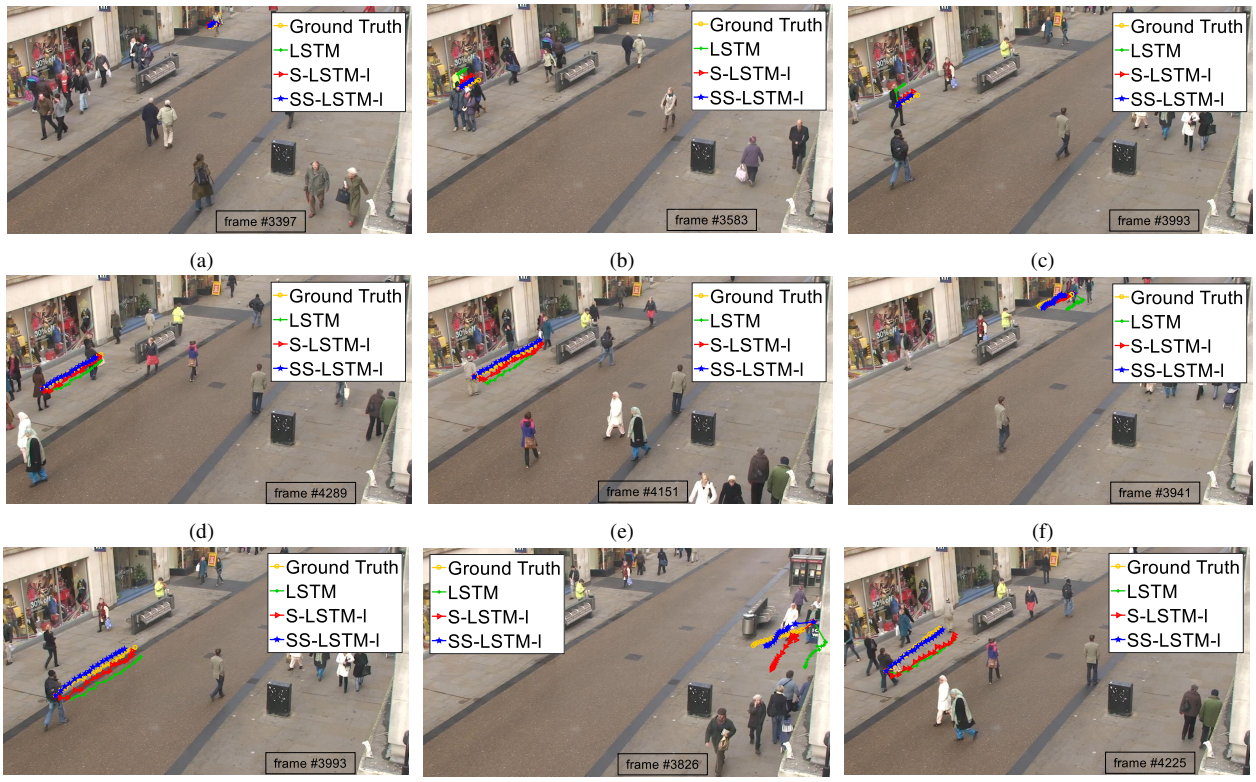
To demonstrate the advantage of using the scene scale, we compare the vanilla LSTM, S-LSTM-l and SS-LSTM-l for a small and a large prediction horizon T_p. The result is listed in Table 3. S-LSTM-l and SS-LSTM-l perform better than the vanilla LSTM method in both cases. When T_p is small, both S-LSTM-l and SS-LSTM-l perform well. It shows that the influence of the scene scale information is limited when T_p is small. However, SS-LSTM-l outperforms S-LSTM-l evidently when T_p is large. The results above confirm that the scene level features learned in our SS-LSTM capture more global information about the scene and they help improve the prediction accuracy of long trajectories.

4.4. Qualitative Results

In Figure 5, we present some predicted trajectories from our SS-LSTMs with different occupancy maps on the ETH (the first row) and the UCY (the second row) datasets. With the help of the social scale and the scene scale, SS-LSTMs are able to predict trajectories in the scenarios where people walk together with others or obstacles such as trees and parked cars in the vicinity. Not only can our SS-LSTMs make successful predictions when a pedestrian walks linearly but they also robustly handle the situation when the pedestrian turns around a corner into another street.

The third row of Figure 5 shows some examples where the prediction results are not so ideal. In these cases, our predicted trajectories are shorter than the ground truth trajectories. The velocities of these pedestrians have been wrongly predicted. However, the predicted trajectories are still acceptable in terms of the general moving directions of the pedestrians. A possible cause is that we consider neighbours at different distances (from the target person) to be the same (given the same weight in the occupancy map).

A comparison among the vanilla LSTM, S-LSTM and SS-LSTM is shown in Figure 6. We choose a small ($T_p =$



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 6. Illustration of a few predicted trajectories from different methods. $T_p = 0.8$ seconds in cases (a) to (c); $T_p = 3.2$ seconds in cases (d)-(i). In all cases, the observed trajectories are 1.6 seconds (i.e., 8 frames) long.

0.8 seconds) and a large ($T_p = 3.2$ seconds) prediction horizon. For the S-LSTM and SS-LSTM methods, *log maps* are used in the social scale. For the small T_p value (Figure 6, row 1), the performance of these three prediction methods are about the same. Both predicted trajectories from S-LSTM and SS-LSTM are very close to the ground truth. In this case, the influence of the scene scale is minimal; using just the pedestrian's neighbourhood is sufficient to get a good prediction. In Figure 6(f), where the pedestrian walks closely with nearby neighhbours, the vanilla LSTM gives the worse predicted trajectory compared to S-LSTM and SS-LSTM. This example shows the significance of the social scale information.

In Figure 6(h), because only the woman in white gown is in the neighbourhood of the woman in black, using S-LSTM, the red trajectory produced is toward the group of people walking in the opposite direction. However, the global scene information enables SS-LSTM to overcome this problem. For the vanilla LSTM method, the situation is even worse as the first predicted point is already a long way off. In Figure 6(i), there is an obstacle on the pedestrian's potential path. Without the scene scale information, trajectories generated by LSTM and S-LSTM are heading towards the obstacle which is clearly invalid. Therefore, considering the global scene scale is essential in the prediction network.

5. Conclusions and Future Work

We have presented a novel hierarchical LSTM-based method for pedestrian trajectory prediction in crowded scenes. We have shown that our proposed SS-LSTM outperforms other methods on three benchmark datasets. In addition, we have demonstrated that using the circular shape neighbourhood gives better trajectory prediction results than the traditional rectangular shape neighbourhood occupancy maps. From our experiments on the Town Centre dataset for predicting trajectories of different lengths, we have demonstrated that our SS-LSTM and *log maps* in the social scale are better when T_p is large.

Although our SS-LSTM outperforms other methods when the prediction horizon is large, the prediction results are still not as accurate as those for short trajectories. In the future, we will focus on further improving the prediction performance by assigning influence weights to neighbours based on their inter-pedestrian distances. We also intend to incorporate a spatial-temporal attention mechanism and an additional network to learn other movement factors such as comfortable distances from other pedestrians into our SS-LSTM prediction model.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F.-F. Li, and S. Savarese. Social LSTM: human trajectory prediction in crowded spaces. In *CVPR*, pages 961–971, June 2016.

[2] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, pages 3457–3464, June 2011.

[3] E. Bonabeau. Agent-based modeling: methods and techniques for simulating human systems. In *Proceedings of the National Academy of Sciences*, pages 7280–7287, 2002.

[4] M. Chen, G. Ding, S. Zhao, H. Chen, Q. Liu, and J. Han. Reference based LSTM for image captioning. In *AAAI*, pages 3981–3987, 2017.

[5] Y. Dauphin, H. de Vries, and Y. Bengio. Equilibrated adaptive learning rates for non-convex optimization. In *Advances in neural information processing systems*, pages 1504–1512, 2015.

[6] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Soft+ hardwired attention: an LSTM framework for human trajectory prediction and abnormal event detection. *arXiv preprint arXiv:1702.05552*, 2017.

[7] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, volume 14, pages 1764–1772, 2014.

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

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

[10] L. Hu, J. Li, L. Nie, X.-L. Li, and C. Shao. What happens next? future subevent prediction using contextual hierarchical LSTM. In *AAAI*, pages 3450–3456, 2017.

[11] I. Karamouzas, B. Skinner, and S. J. Guy. Universal power law governing pedestrian interactions. *Physical review letters*, 113(23):238701, 2014.

[12] A. Karpathy and F.-F. Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015.

[13] S. Kim, S. J. Guy, W. Liu, D. Wilkie, R. W. Lau, M. C. Lin, and D. Manocha. BRVO: Predicting pedestrian trajectories using velocity-space reasoning. *The International Journal of Robotics Research*, 34(2):201–217, 2015.

[14] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017.

[15] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[16] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. Global context-aware attention LSTM networks for 3D action recognition. In *CVPR*, pages 1647–1656, 2017.

[17] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[18] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[19] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268, 2009.

[20] J. S. Ren, Y. Hu, Y.-W. Tai, C. Wang, L. Xu, W. Sun, and Q. Yan. Look, listen and learn - a multimodal LSTM for speaker identification. In *AAAI*, pages 3581–3587, 2016.

[21] X. Song, H. Kanasugi, and R. Shibasaki. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *IJCAI*, pages 2618–2624, 2016.

[22] H. Su, Y. Dong, J. Zhu, H. Ling, and B. Zhang. Crowd scene understanding with coherent recurrent neural networks. In *IJCAI*, pages 3469–3476, 2016.

[23] H. Su, J. Zhu, Y. Dong, and B. Zhang. Forecast the plausible paths in crowd scenes. In *IJCAI*, pages 2772–2778, 2017.

[24] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[25] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 797–803. IEEE, 2010.

[26] A. Vemula, K. Muelling, and J. Oh. Modeling cooperative navigation in dense human crowds. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1685–1692, May 2017.

[27] M. Wang, B. Ni, and X. Yang. Recurrent modeling of interaction context for collective activity recognition. In *CVPR*.

[28] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *CVPR*, pages 1345–1352. IEEE, 2011.

[29] S. Yi, H. Li, and X. Wang. Understanding pedestrian behaviors from stationary crowd groups. In *CVPR*, pages 3488–3496, 2015.

[30] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, et al. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *AAAI*, pages 3697–3703, 2016.