Pedestrian Path Forecasting in Crowd: A Deep Spatio-Temporal Perspective

Yuke Li
State Key Lab of LIESMARS, Wuhan University
sunfreshing@whu.edu.cn

ABSTRACT

Predicting the walking path of a pedestrian in crowds is a pivotal step towards understanding his/her behavior. This is one of the recently emerging tasks in computer vision scarcely addressed to date. In this paper, we put forth a deep spatio-temporal learning-forecasting approach, which is composed of two modules. First, displacement information from pedestrians' walking history is extracted and fed into a convolutional layer in order to learn the undergoing motion patterns and produce high-level representations. Second, unlike the mainstream literature which learns the temporal or the spatial dynamics among the pedestrians separately, we propose to embed both components into a single framework via a Long-Short Term Memory based architecture that takes as input the previously extracted high-level motion cues and outputs the potential future walking routes of all pedestrians in one shot. We evaluate our approach on three large benchmark datasets, and show that it introduces large margin improvements with respect to recent works in the literature, both in short and long-term forecasting scenarios.

KEYWORDS

Path forecasting; deep learning; crowd motion dynamics; computer vision;

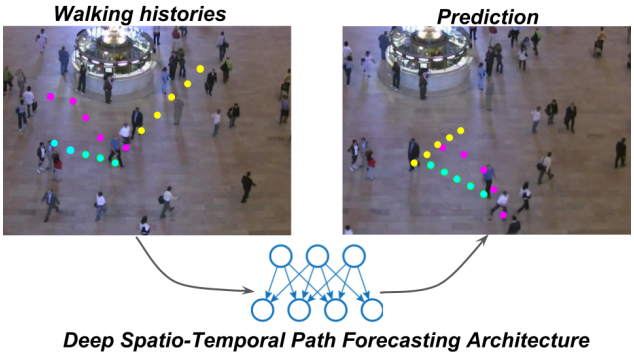

Figure 1: We build a deep spatio-temporal architecture to jointly learn the spatio-temporal dependencies from all the pedestrians in the scene, which enables an accurate estimation of the potential the forthcoming paths. The cyan, magenta and yellow dots denote the walking histories of three independent pedestrians (left), which are opportunely harnessed to foresee their respective future walking trajectories (right).

1 INTRODUCTION

Understanding the motion dynamics of people in crowds, especially in dense public spaces, is an essential step to face many tasks such as crowd traffic management, or smoothly carrying out preventive security measures. Owing to the rise of powerful processing facilities,it is now established that modeling human behavior tendency is not out of reach. In this respect, several topics have been addressed recently, such as activity recognition [44], anomaly detection [19], crowd counting and profiling [24, 39, 41, 42].

Pedestrian path prediction is a relatively new emerging computer vision task, which refers to forecasting the potential future walking course of an individual based on his/her prior walking history. With respect to other computer vision based crowd modeling/analysis topics, path forecasting has received a quite scarce attention lately.

Forecasting the future paths of pedestrians in crowds is regarded as a multi-facet and complicated task, as it entails handling multiple factors simultaneously. It essentially requires to understand the complex, and often subtle motion dynamics that take place in crowded areas, and usually involves two key factors:

- Spatial dependencies: the path of a person of interest is usually influenced by the people around him/her, which is also known as "social interaction"[21]. Apart from unusual situations, it is socially evident that humans generally adopt commonsense rules (e.g., a common way to get through a path that is obstructed by an object or a pedestrian is to detour to avoid collision).
- Temporal dependencies: in crowded scenarios, people often tend to move at fluctuated paces and in apparently casual directions. For this reason, the motion dependencies of pedestrians exhibit high variability over time.

The literature reports several efforts to solve the aforementioned challenges by two steps settings: modeling the spatial

MM'17, October 23–27, 2017, Mountain View, CA, USA.

DOI: https://doi.org/10.1145/3123266.3123287

and temporal dependencies independently. For instance, in [1], an Long-Short Term Memory (LSTM) model is assigned per pedestrian. Moreover, a pooling layer is adopted as to capture the spatial context of each LSTM within a certain radius. Another architecture, was suggested in [29], where an LSTM model is assigned per tracklet, and a coherent regularization [41] is applied in order to model the interaction between the tracklets.

A common denominator in both above methods is that they require additional terms (i.e., pooling layer, coherent regularization) in order to model the interaction among the already large number of LSTM models due to the crowd. Such issues were partially addressed in [37], where a Convolutional Neural Network (CNN) is designed to learn the motion behaviors of all the pedestrians in a scene. Nevertheless, with respect to LSTM architectures which can inherently deal with sequential data, the proposed CNN does not explicitly learn the motion dependencies, especially the temporal one. In this context, a better model could be the one in which spatial and temporal dependencies are jointly considered, which would reflect in higher accuracies. To this end, we propose a LSTM based deep spatio-temporal approach for pedestrian path prediction in crowds (see Figure 1). In our approach, a single model for all pedestrians is learnt, rather than attributing one model per each pedestrian, and this enables a simultaneous prediction of walking routes for all pedestrians. Moreover, our work differs significantly from the existing studies on path prediction in other features we will highlight later on in the paper. The main contributions of our work are:

(i) We introduce a deep spatio-temporal method considering spatio-temporal dependencies in a one step seamless way for pedestrian path prediction.
(ii) This method takes advantage of convolutional (*conv*) layer to better consider spatial information, and LSTM based model to capture motion dynamics information.
(iii) We thoroughly evaluate our framework on three widely used benchmark datasets under two distinct pedestrian path forecasting scenarios and show that significant gains can be attained with respect to recent works.

To the best of our knowledge, this is the first study towards explicitly incorporating the spatio-temporal cues in a unified framework in the path prediction context.

The remainder of this paper is organized as follows: Firstly, relevant works are introduced in 2. Section 3 details the framework in terms of motion cues extraction, and the proposed learning-forecasting architecture. Section 4 conducts experimental findings and discussion. Finally, the conclusion will be presented in Section 5.

2 RELATED WORK

The relevant literature has accumulated several attempts to tackle the challenges of path prediction. For instance, a pioneering work presented in [14], suggests exploiting the semantics of the scene in order to build a trajectory forecasting model, which is based on Markov Decision Processes [8] jointly with inverse reinforcement learning [20]. Nonetheless, the proposed framework was assessed in a one person scenario, which raises concerns about its suitability to dense public spaces. Another work [32] suggests a non-parametric approach for visual prediction. But the definition of agents in this work appears to be subjective given that, in heavily crowded scenes, normally large-scale occlusions take place. The work in [16] integrates Gaussian Process Regression with hand-crafted motion features applied specifically for predicting the trajectories of American football players. This leaves open the question of its applicability to different scenarios. The contribution in [38] puts forth a Bayesian cascade model that couples Topic Mixture Model and Gaussian mixtures, fed with an ensemble of words encoding motion tracklets over a grid in a given scene. However, given that tracklets are discontinuous, it is hard to assign the predicted outcomes to individuals, which does not meet the aim of human path prediction. The framework of [2] considers a Dynamic Bayesian Network for modeling motion dependencies and transferring them to a different scene. Nevertheless, the loss that may occur at knowledge transfer level is prone to compromise the prediction results.

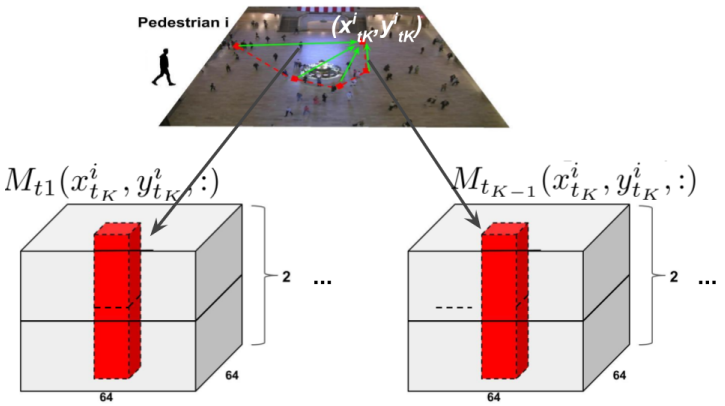

Figure 2: The example of motion feature extraction.

Recently, Deep Neural Networks (DNNs) [9] have shown to achieve cutting-edge performance in several vision tasks related to visual understanding [5, 12, 13, 23, 26], action recognition [25, 27, 43] and scene segmentation [3, 22, 40]. One remarkable instance is the LSTM model, which was successfully tailored to sequential data learning tasks [10, 11, 28]. This inspired an LSTM based path prediction in [1]. The underlying idea is to assign one LSTM per walking route, and then top them with a pooling layer to ensure the modeling of the spatial dependencies. Another LSTM driven method shares a similar idea [29]. Temporal dependencies are modeled via LSTM while spatial dependencies are handled by a hand-crafted based coherent regularization . Though the LSTM can well handle the temporal changes, these two approaches suffer from two main drawbacks. On the one hand, they associate an LSTM model to each pedestrian/tracklet

in the scene, which inflates the computation cost. On the other hand, adding extra pooling/regularization layers is necessary, provided that LSTM inherently lacks the capacity of handling spatial information. "Behavior CNN"[37] presents a designated CNN to learn the motion of all the pedestrians in a scene. Nevertheless, to explicitly learn the motion dependencies, especially the temporal one remains a issue.

In this work, we propose a LSTM based deep spatio-temporal path forecasting architecture. We would like to stress the fact that, (i) by contrast to [1] and [29], our approach does not require any pooling/regularization terms as we learn a single architecture for all the pedestrians in a scene rather than designing one model per person/tracklet, which implies a much less complexity, and good capability of spatial context modeling; (ii) unlike [37], our end-to-end approach can better handle the temporal dependencies. Such property is inherited from the latent capacity of LSTM in treating sequential data as detailed further.

3 METHODOLOGY

As abovementioned, the walking tendencies of pedestrians in crowded settings are characterized by spatio-temporal dependencies. Spatial because pedestrians influence each other's motion patterns, and temporal given that the potential future walking locations of an individual have a strong tie with his/her previous walking history. To this end, we first express the walking records of all pedestrians by means of displacement vectors. Second, we feed the displacement information into the LSTM based model, which in turn comprises two submodules. The first one assimilates the previously extracted displacement tensors as input and learns the undergoing motion dynamics of the scene, whilst the second one serves as a prediction stage that infers the forthcoming walking routes for all the pedestrians at one shot. In what follows, we provide details outlining each part.

3.1 Motion dynamics description

We proceed by characterizing the motion patterns of pedestrians through a set of displacement vectors. Let $\{x_t^i, y_t^i\}(t \in \{t_1, t_2, ...t_K\})$ be the coordinates of pedestrian i within a uniformly sampled time interval from t_1 to t_K. Hence, the displacement for pedestrian i at time t is given by $\{x_{t_K}^i - x_t^i, y_{t_K}^i - y_t^i\}(t \in \{t_1, t_2, ...t_K\})$.

Therefore, at time instant t, a three-dimensional tensor $M_t \in \mathbb{R}^{H \times W \times 2}$ is built, where H, W are the height and width of the scene, respectively. The entry of the first layer in M_t at position $\{x_t^i, y_t^i\}$ represents the displacement value along the x axis at time t, whilst the entry of the second layer at the same position represents the displacement value along the y axis, *i.e.* $\{M_t(x_{t_K}^i, y_{t_K}^i, 1) = x_{t_K}^i - x_t^i + W; M_t(x_{t_K}^i, y_{t_K}^i, 2) = y_{t_K}^i - y_t^i + H\}(t \in \{t_1, t_2, ...t_K\})$. W and H are added to ensure the corresponding entries in M_t are positive. Thus, each nonzero index belongs to a particular pedestrian in the scene at time t. Yet, the motion dynamics (expressed by the displacement entries) of the entire crowds in the scene are accommodated by a single tensor M_t. In this work, we are interested in predicting the future positions of all pedestrians denoted as $M_{t'}$, which includes $\{x_{t'}^i, y_{t'}^i\}(t' \in \{t_{K+1}, t_{K+2}, ...t_{K+K'}\})$, by capitalizing on the prior walking history stored in M_t. Figure 2 further highlights how the tensor M_t is constructed for a given pedestrian i.

3.2 Proposed architecture

To fully capture the spatio-temporal motion dependencies of all the pedestrians in the scene, we carry out the path forecasting concern in a sequence-to-sequence learning fashion via an LSTM based learning-forecasting framework. Figure 3 depicts the different blocks outlining our proposed architecture, which encloses learning and forecasting modules.

We draw the attention to the fact that the tensor M_t, which encompasses the displacement patterns of all pedestrians at time instance t, may manifest some sparsity, which hurdles the training of the network. In this respect, some prior works in the recognition field suggest harnessing high-level visual percepts extracted from the top-layers of a CNN architecture [26, 27]. The output representations are able to preserve the spatial topology of the input. However, directly applying a regular LSTM on such visual percepts would drastically magnify the number of the parameters of LSTM in characterizing the input-to-state transformations due to the size of the visual percepts. In order to address this issue, we replace the fully-connected LSTM linear product operation by a convolution operation. Moreover, we leverage high-level motion representations $I_t(t \in \{t_1, t_2, ...t_K\})$ from a single *conv* layer fed by M_t, instead of going through a long-chain CNN structure (such as AlexNet[15] and GoogLeNet [30]).

3.2.1 Convolutional LSTM. LSTM has been demonstrating an outstanding capacity in performing sequential data learning tasks [10, 34] with respect to other deep network structures. Consequently, LSTM has been successfully finding their way into several computer vision tasks such as video understanding [26, 28].

We invoke the point that an indispensable factor in designing a path prediction pipeline is to jointly handle both the temporal and spatial motion dependencies in the scene. A regular LSTM architecture, though can handle the temporal aspect to some extent, does not suit our purpose as it cannot cope with the spatial part. This respect lead us to convolutional LSTM (ConvLSTM) [33], which offers the advantage of tackling both earlier components in a single framework and for all the pedestrians in the scene. We experimentally demonstrate that a ConvLSTM outperforms the LSTM, often by a large margin, in the path forecasting context.

Traditional LSTM operate recurrently over sequences of one-dimensional vectors through biased linear transformations, topped by non-linearities to calculate gate and cell activations. We replace the fully connected transformations with convolutional operations to comprise the spatial context

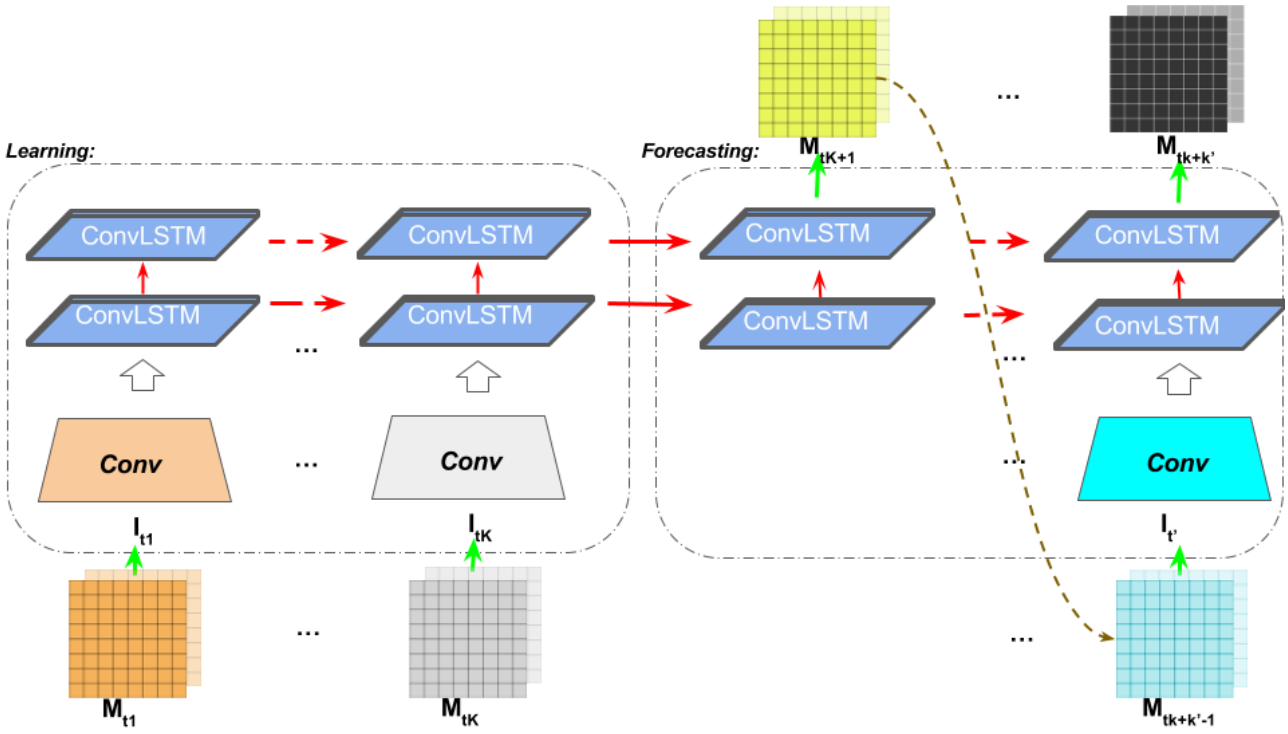

Figure 3: The pipeline of our proposed architecture. The arrows highlight the information flow. Both the learning and the forecasting modules involve two stacked layers of ConvLSTM units. The high-level motion representations $I_t(t \in \{t_1, t_2, ...t_K\})$ extracted from M_t are fed to learning module. The hidden state and cell state pass to the next ConvLSTM units. In the forecasting module, ConvLSTM units take the motion maps $I_{t'}$ extracted from the previous predicted motion dynamics' descriptions $M_{t'}(t' \in \{t_{K+1}, t_{K+2}, ...t_{K+K'-1}\})$ as input. Note that the prediction output $M_{t_{K+1}}$ from the first ConvLSTM unit of the forecasting module is inferred from the hidden state and cell state of the last ConvLSTM unit of the learning module.

among the pedestrians. Therefore, ConvLSTM is given by:

$$\begin{cases} i_t = \sigma(\omega_{Ii} * I_t + \omega_{hi} * h_{t-1} + b_i) \\ f_t = \sigma(\omega_{If} * I_t + \omega_{hf} * h_{t-1} + b_f) \\ \tilde{c}_t = tanh(\omega_{\tilde{c}I} * I_t + \omega_{\tilde{c}h} * h_{t-1} + b_{\tilde{c}}) \\ c_t = \tilde{c} \odot i_t + c_{t-1} \odot f_t \\ o_t = \sigma(\omega_{Io} * I_t + \omega_{ho} * h_{t-1} + b_o) \\ h_t = o_t \odot tanh(c_t) \end{cases} \quad (1)$$

where I_t are the high-level motion maps produced by the convolutional layer at time step t. $i_t, f_t, \tilde{c}_t, o_t$ denote input, forget, cell and output gate at time step t, respectively. $\{h_{t-1}, h_t\}$ and $\{c_{t-1}, c_t\}$ denote the hidden state and memory state at time step $t-1$ and t. $\sigma, tanh$ are sigmoid and hyperbolic tangent nonlinear activation functions inside each ConvLSTM cell. ω_*, b_* are the weights and bias. $*$ denotes convolution operation and $\odot$ pertains to Hadamard product.

The advantage of ConvLSTM with respect to the traditional LSTM is traced back to the prevalence of conv transformations over linear transformations. Since they can meet the temporal information with the spatial context, ConvLSTM is more suitable for vision based crowd analysis/modeling tasks, notwithstanding the fact that they require tuning up fewer parameters thus less memory.

3.2.2 Learning - forecasting network. In this subsection, we describe the proposed architecture, which combines a ConvLSTM along with a *conv* layer. We employ encoder - decoder framework, inspired by [6, 28], to build our "learning - forecasting"architecture.

The proposed architecture framework shares similar idea with AutoEncoder [7, 31]. At the learning stage, the aim is to make use of the high-level representations produced by the appended *conv* layer in order to learn the inherent characteristics of the motion dynamics taking place in the scene (see Figure 3). The *conv* aspect of the ConvLSTM network serves for capturing the spatial interactions among the pedestrians. The temporal part is handled by the ConvLSTM's recurrent nature. Subsequently, the learnt dynamics are encoded in the form of hidden state as detailed above, which are expressed at time instance t as:

$$h_t^{learn} = ConvLSTM_{learn}(I_t, h_{t-1}^{learn}) \quad (2)$$

where h_t^{learn} is the hidden state of learning at time instance $t(t \in \{t_1, t_2, ...t_K\})$. I_t is the high-level motion representation, which is taken as input of the network. h_{t-1}^{learn} is the output hidden state of the learning module at time $t-1$.

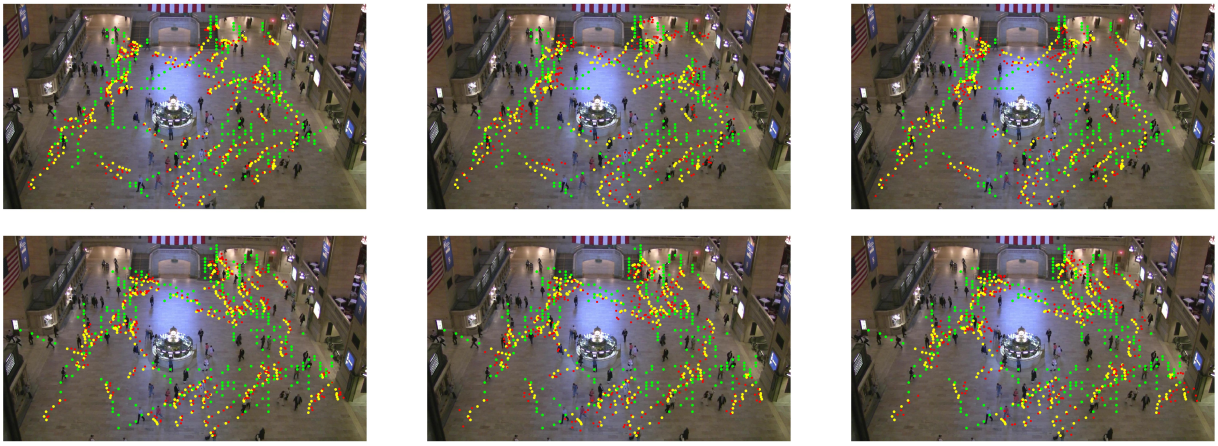

Figure 4: The example of walking paths experimental results on PWPD dataset. From left to right, proposed approach, *Baseline 1* and *Baseline 2*. The green, yellow and red dots are the history coordinates, groundtruth and our predicting outputs, respectively.

Instead of reconstructing the input of the learning module, the produced hidden state then serve as the input to the forecasting part to foresee the potential future walking routes. The output of the forecasting module at time t' is defined as follows:

$$h_{t'}^{fore} = ConvLSTM_{fore}(I_{t'}, h_{t'-1}^{fore}) \quad (3)$$

where $h_t'^{fore}$ is the output hidden state at time instance $t'(t' \in \{t_{K+1}, t_{K+2}, ...t_{K+K'}\})$ in the forecasting module, $h_{t'-1}^{fore}$ is the hidden state of the forecasting module at time $t'-1$. $I_{t'}$ is the high-level motion representation produced from $M_{t'-1}$ by the *conv* layer. It is received as input to our forecasting module at time t'. In our case, each ConvLSTM block contains two stacked ConvLSTM layers. In order to obtain the predicted future motion dynamics M_t' in the same size as M_t, we append a *conv* layer which takes over the hidden state $h_{t'}^{fore}$ of the forecasting module as inputs.

3.3 Cost function

The cost function is expressed in terms of the $L2$ distance between the predicted motion dynamics $M_{t'}$ and the groundtruth $GT_{t'}$:

$$COST = \frac{1}{\#GT_{t'}^{nonzero}} \parallel (M_{t'} - GT_{t'}) \odot 1_{M_{t'}, GT_{t'}} \parallel_2^2 \quad (4)$$

where $(t' \in \{t_{K+1}, t_{K+2}, ...t_{K+K'}\})$. $1_{M_{t'}, GT_{t'}}$ is an indicator function, which equals to one if $M_{t'}$ have the same non-zero indexes with $GT_{t'}$, otherwise it holds a zero. $\odot$ represents the Hadamard product. The error between the prediction and the groundtruth is averaged by the number of nonzero entries in $GT_{t'}$. We follow back-propagation through time (BPTT) [9] to train the network.

4 EXPERIMENTS

Three large-scale benchmark datasets are exploited to validate the effectiveness of the proposed framework, namely

Method	NMSE
Ours	**1.97%**
Behavior CNN [37]	2.41%
Baseline 1 [28]	3.50%
Baseline 2	2.78%

Table 1: The quantitative results of walking path forecasting on PWPD dataset with NMSE criterion.

Pedestrian Walking Path Dataset (PWPD) [36], ETH [21] and UCY [18] are selected. PWPD comprises over 10000 pedestrians from a one-hour video, where the complete trajectories from the time they enter the scene to the exit time is uniformly annotated at a 20-frames rate. PWPD makes a good ground to realistically evaluate our framework as it manifests a high pedestrian density over large portions of the video. The ETH dataset contains two scenes each with 750 different pedestrians and is split into two sets (ETH and Hotel). The UCY dataset contains three scenes with 786 people: UCY, ZARA-01 and ZARA-02. All these datasets involve very challenging crowd scenarios as suggested in [21]. For instance couples walking together, groups crossing each other and groups forming and dispersing.

As per comparison, we assess the performance of our framework versus recent leading works. The first one is Social LSTM [1], which assigns an LSTM model per person and then appends a pooling layer to treat the spatial context. The second one is Behavior CNN [37], which learns a CNN network for the entire scene. In order to underline the merit over ConvLSTM of the traditional LSTM, we also compare our work with [28], which we refer to as *Baseline 1*. Finally, we also judge the performance of our architecture without appending the *conv* layer, which we name *Baseline 2*.

4.1 Implementation details

The spatial coordinates from the annotations provided by the datasets are embedded to a dimension of 64×64. The

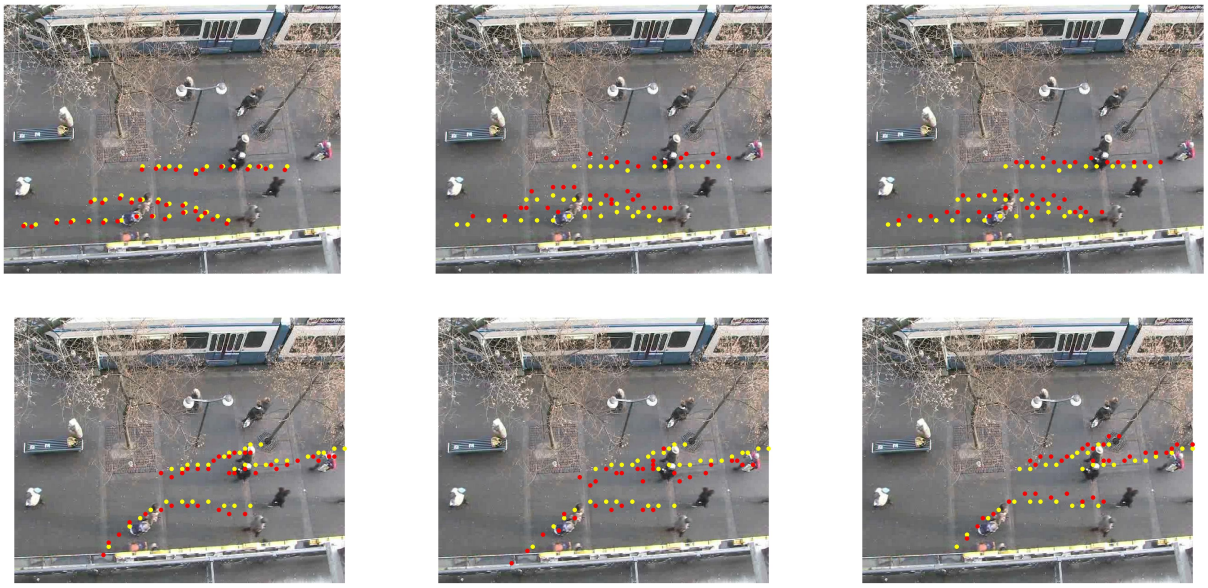

Figure 5: Path forecasting instances from the ETH dataset. From left to right, proposed approach, *Baseline 1* and *Baseline 2*. The yellow and red dots pertain to the groundtruth and the estimated walking locations, respectively.

motion dynamics description M_t within the time lapse $t \in \{t_1, t_2, ...t_K\}$ and groundtruth $GT_{t'}$ within $t' \in \{t_{K+1}, t_{K+2}...t_{K+K'}\}$, which denote the motion history and the future walking records, respectively, are constructed in the same manner as described in subsection 3.1. $GT_{t'}$ is the same size as M_t, and shares the same nonzero indexes and entries with M_t. It is to note that our model can predict the motion dynamics within different time intervals by setting K and K'.

The motion representations I_t It are extracted via eight 5×5 kernel with a stride of 1 and zero-padding. In our experiments, each ConvLSTM module has two stacked ConvLSTM layers with 128 hidden states. The input-to-state and state-to-state kernel of ConvLSTM size is 3×3, with a stride of 1 and zero-padding. We adopt a single *conv* layer with a 1×1 kernel size which transforms $h_{t'}^{fore}$ into final prediction output $M_{t'}$, whilst ensuring $M_{t'}$ is the same size as M_t. The parameters of the ConvLSTM applied for our architecture and *Baseline 2*, except the biases, were initialized from a uniform distribution $\mu(-0.08, 0.08)$. The biases of the forget gates are initialised to 1; for the other gates we set the biases to 0. The hidden state dimension of 128 is also fixed with both *Baselines 1* and 2. The parameters of *Baseline 1* are initiated by the same settings as in [28].

The training is done by minimizing the cost defined in equation 4 using BPTT and *rmsprop* with a learning rate initialized at 5×10^{-5} and a decay rate of 0.85.

Our implementation is based on Torch 7 [4] and extended *rnn* library [17]. The experiments were carried out on a Nvidia GeForce GTX 1080, supplied with a 8G memory.

Method	ETH	HOTEL	ZARA1	ZARA2	UCY
Ours	**0.31**	0.13	**0.18**	**0.20**	**0.20**
Behavior CNN [37]	0.35	0.18	0.20	0.23	0.23
Social LSTM [1]	0.50	**0.11**	0.22	0.25	0.27
Baseline 1 [28]	0.58	0.15	0.40	0.52	0.50
Baseline 2	0.35	0.16	0.22	0.22	0.23

Table 2: The quantitative results on average MSE criterion of walking path forecasting on ETH and UCY datasets.

4.2 Pedestrian path forecasting

We first assess the effectiveness of our framework as compared to the earlier two baselines as well as the work presented in [37] on PWPD dataset. In order to allow a fair comparison, we follow the experimental setting opted for in [37], by setting both K and K' to 5. Thus, the entire PWPD dataset is uniformly divided into 4990 video clips, with a sampling rate of 20 frames (0.8 seconds). In other words, the proposed network predicts the forthcoming 4 seconds, based on a walking history of the prior 4 seconds. We set 80% of the dataset for training, 10% for validation and the rest 10% for test. The normalized mean square error (NMSE) [37] is adopted as metric.

Table 1 reports the quantitative path prediction results on PWPD. On the one hand, our framework outperforms, by far, *Baseline 2*, thanks to the appended *conv* layer that plays a solid contribution in capturing the undergoing motion patterns across the scene. On the other hand, Behavior CNN [37]

Figure 6: The examples of our destination estimation results. From left to right, proposed approach, *Baseline 1* and *Baseline 2*. The green, yellow and red dots are the history coordinates, groundtruth and our predicting outputs, respectively.

Method	*Top 1*	*Top 2*	*Top 3*
Ours	**61%**	**77%**	**89%**
Behavior CNN [37]	53%	72%	84%
Baseline 1 [28]	49%	66%	75%
Baseline 2	55%	74%	84%

Table 3: The quantitative results of Destination Estimation of Pedestrians on PWPD datset with *Top-N* accuracy.

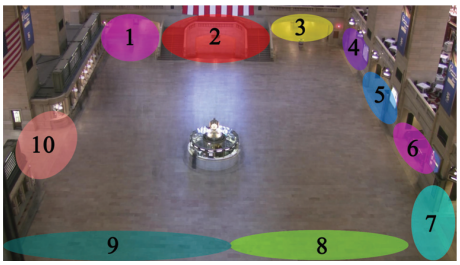

Figure 7: Labeled entries/exits in the PWPD dataset.

which represents the so far best score in the state-of-the-art on this dataset, is exceeded by our work. This is a significant gain provided the density of pedestrians characterizing this dataset, which to the best of our knowledge, manifests the most dense crowd setting among all the available benchmarks. As per *Baseline 1* [28], the worst NMSE of 3.5% was obtained. It is traced back to the fact that, regular LSTM does not take spatial dependencies into account. However, the results favor our approach in which spatio-temporal cues are tied into a single framework. In fact, even our *Baseline 2* still outdoes *Baseline 1*. Figure 4 displays path forecasting examples. It is evident that our approach exhibits better prediction outcomes in both linear and non-linear trajectories.

In the second part of the experiments, we extend the values of K and K' to 8 and 12, respectively, on ETH and UCY datasets as in [1] and [37]. The evaluation criterion follows a leave-one-out cross-validation strategy, and average MSE as in [1, 21] is used as evaluation metric. Table 2 summarizes the results on these 2 datasets (entailing 5 sub-datasets). The proposed architecture achieves the best scores on almost all datasets, closely followed by Behavior CNN [37]. except on the Hotel dataset where Social LSTM [1] yields the best result, which is slightly ahead of our method. Our framework remains superior to *Baselines 1* and *2* owing to the reasons mentioned earlier. Figure 5 depicts several examples on the ETH dataset.

4.3 Destination estimation

Estimating the potential destinations of pedestrians is another task that is critical to comprehending the behavioral tendencies of the crowds. Unlike path prediction, destination estimation implies observing the entire trail of a pedestrian all the way up to the pouring point, which determines his/her likely destination. Thus, it can be regarded as a long-term prediction. Our framework enables long-term path prediction by manipulating the values of K and K', which are set to 5 and 15 as to comply with the evaluation scenario in [37]. *Top-N* accuracy (correct destination estimation belongs to the *Top-N* predictions) is used to quantify the accuracy, in line with [37].

In order to determine the final destinations, we adopt the same procedure in [37] on the PWPD dataset, by calculating the Euclidean distance between the last predicted coordinate and the center of the labeled entries/exits.

The quantitative results are reported in Table 3. It can be seen that our approach incurs large margin improvements versus Behavior CNN [37] and both baselines. For instance,

Top 1 rate amounts to 61% , which drastically advances the Behavior CNN [37] by 8%. The obtained score from our approach also confirms the superiority of the proposed architecture in the long-term prediction scenario, which meets our expectations thanks to the integration of temporal and spatial dependencies in a single pipeline, which is further boosted by the *conv* layer. We provide prediction examples in Figure 6. The labeled groundtruth of the entries/exits provided by [35] is displayed in Figure 7.

5 CONCLUSION

This paper proposed a deep spatio-temporal architecture for pedestrians' path prediction in crowds provided their walking records across a given scene. The convolutional as well as recurrent nature of the framework enables capturing the spatial and temporal motion dynamics in one unified framework. The proposed architecture is thoroughly assessed on three widely used benchmark datasets in the contexts of path forecasting and destination estimation. We have shown that it can outperform trending works.

We believe that the presented approach can benefit from further improvements by opting for instance for a deeper architecture. This would expectedly learn richer spatio-temporal motion dependencies yet leads to better prediction scores. Another interesting direction is to investigate our framework on a different crowd related task, such as crowd profiling.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 961–971.

[2] Lamberto Ballan, Francesco Castaldo, Alexandre Alahi, Francesco Palmieri, and Silvio Savarese. 2016. Knowledge Transfer for Scene-specific Motion Prediction. In *European Conference on Computer Vision*. Springer, 697–713.

[3] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. 2016. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision*. Springer, 549–565.

[4] R. Collobert, K. Kavukcuoglu, and C. Farabet. 2011. Torch7: A Matlab-like Environment for Machine Learning. In *BigLearn, NIPS Workshop*.

[5] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. 2016. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4772–4781.

[6] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2625–2634.

[7] Bo Du, Wei Xiong, Jia Wu, Lefei Zhang, Liangpei Zhang, and Dacheng Tao. 2017. Stacked convolutional denoising auto-encoders for feature representation. *IEEE transactions on cybernetics* 47, 4 (2017), 1017–1027.

[8] Bob Givan and Ron Parr. 2001. An introduction to Markov decision processes. (2001).

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

[10] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[11] Alex Graves and Navdeep Jaitly. 2014. Towards End-To-End Speech Recognition with Recurrent Neural Networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1764–1772.

[12] Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing* 7, 11 (2015), 14680–14707.

[13] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. 2016. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5308–5317.

[14] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. 2012. Activity forecasting. In *European Conference on Computer Vision*. Springer, 201–214.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[16] Namhoon Lee and Kris M Kitani. 2016. Predicting wide receiver trajectories in American football. In *WACV*. IEEE, 1–9.

[17] Nicholas Léonard, Sagar Waghmare, Yang Wang, and Jin-Hwa Kim. 2015. rnn: Recurrent library for torch. *arXiv preprint arXiv:1511.07889* (2015).

[18] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by example. In *Computer Graphics Forum*, Vol. 26. Wiley Online Library, 655–664.

[19] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[20] Andrew Y Ng and Stuart Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *in Proc. 17th International Conf. on Machine Learning*.

[21] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*. IEEE, 261–268.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[23] Jing Shao, Kai Kang, Chen Change Loy, and Xiaogang Wang. 2015. Deeply learned attributes for crowded scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4657–4666.

[24] Jing Shao, Chen Change Loy, and Xiaogang Wang. 2014. Scene-independent group profiling in crowd. In *CVPR*.

[25] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. 2015. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119* (2015).

[26] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4168–4176.

[27] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*. 568–576.

[28] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised Learning of Video Representations using LSTMs. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 843–852.

[29] Hang Su, Yinpend Dong, Jun Zhu, Haibin Lin, and Bo Zhang. 2016. Crowd Scene Understanding with Coherent Recurrent Neural Networks. In *Proceedings of the IJCAI 2016*. 3469–3476. http://www.ijcai.org/Abstract/16/490; http://dblp.uni-trier.de/rec/bib/conf/ijcai/SuDZLZ16

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.

[31] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, Dec (2010), 3371–3408.

[32] Jacob Walker, Abhinav Gupta, and Martial Hebert. 2014. Patch to the future: Unsupervised visual prediction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3302–3309.

[33] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*. 802–810.

[34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 2048–2057.

[35] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2015. Pedestrian Travel Time Estimation in Crowded Scenes. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE.

[36] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2015. Understanding Pedestrian Behaviors from Stationary Crowd Groups. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE.

[37] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2016. Pedestrian Behavior Understanding and Prediction with Deep Neural Networks. In *European Conference on Computer Vision*. Springer, 263–279.

[38] YoungJoon Yoo, Kimin Yun, Sangdoo Yun, JongHee Hong, Hawook Jeong, and Jin Young Choi. 2016. Visual Path Prediction in Complex Scenes With Crowded Moving Objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[39] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 833–841.

[40] Jianming Zhang, Zhe Lin, Jonathan Shen Xiaohui Brandt, and Stan Sclaroff. 2016. Top-down Neural Attention by Excitation Backprop. In *European Conference on Computer Vision(ECCV)*.

[41] Bolei Zhou, Xiaoou Tang, and Xiaogang Wang. 2015. Learning collective crowd behaviors with dynamic pedestrian-agents. *International Journal of Computer Vision* 111, 1 (2015), 50–68.

[42] Bolei Zhou, Xiaoou Tang, Hepeng Zhang, and Xiaogang Wang. 2014. Measuring Crowd Collectiveness. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (2014), 1586–1599.

[43] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao. 2016. A key volume mining deep framework for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1991–1999.

[44] Maryam Ziaeefard and Robert Bergevin. 2015. Semantic human activity recognition: a literature review. *Pattern Recognition* 48, 8 (2015), 2329–2345.