Pedestrian Behavior Understanding and Prediction with Deep Neural Networks

Shuai Yi[1,2], Hongsheng Li[1(✉)], and Xiaogang Wang[1(✉)]

[1] Department of Electronic Engineering,
Chinese University of Hong Kong, Hong Kong, China
{syi,hsli,xgwang}@ee.cuhk.edu.hk
[2] Sensetime Group Limited, Hong Kong, China
yishuai@sensetime.com

Abstract. In this paper, a deep neural network (Behavior-CNN) is proposed to model pedestrian behaviors in crowded scenes, which has many applications in surveillance. A pedestrian behavior encoding scheme is designed to provide a general representation of walking paths, which can be used as the input and output of CNN. The proposed Behavior-CNN is trained with real-scene crowd data and then thoroughly investigated from multiple aspects, including the location map and location awareness property, semantic meanings of learned filters, and the influence of receptive fields on behavior modeling. Multiple applications, including walking path prediction, destination prediction, and tracking, demonstrate the effectiveness of Behavior-CNN on pedestrian behavior modeling.

1 Introduction

Pedestrian behavior modeling is gaining increasing attention and can be used for various applications including behavior prediction [1–4], pedestrian detection and tracking [5–7], crowd motion analysis [8–11], and abnormal detection [12–14].

Modeling pedestrian behaviors is challenging. Pedestrian decision making is complex and can be influenced by various factors. The decision making process of individuals [15], the interactions among moving and stationary pedestrians [4,16], and historical motion statistics of a scene provide information for predicting future behaviors of pedestrians. While existing works focused some of these aspects with simplified rules or energy functions [15,17], our proposed model takes all these factors into account through a complex deep convolution neural network (Behavior-CNN) and makes more reliable predictions.

When using deep neural networks to model pedestrian behaviors, the main difficulty is how to make good use of pedestrian walking information as the input of networks. A straightforward way was to use dense optical flow maps to describe motions of a whole frame. However, it introduces ambiguities when merging and splitting events happen frequently in crowded scenes. As shown in Fig. 1(c), two separate pedestrians A and B at time $t-1$ move to occlude each other at location C at time t. The two flow vectors $(A \rightarrow C)$ and $(B \rightarrow C)$ describe the associations between $t-1$ and t. If the two pedestrians move to locations D and E

B. Leibe et al. (Eds.): ECCV 2016, Part I, LNCS 9905, pp. 263–279, 2016.
DOI: 10.1007/978-3-319-46448-0_16

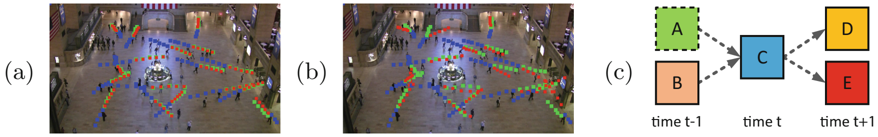


Fig. 1. Prediction results by the proposed Behavior-CNN (a) and the Social Force Model [15] (b). The input, predicted and ground-truth walking paths are shown as blue, red, and green dots, respectively. Only some pedestrians' prediction results are shown in the figure. (c) Illustration of association ambiguity in dense flow maps. (Color figure online)

at $t+1$ with flow vectors $(C \rightarrow D)$ and $(C \rightarrow E)$, it is obvious that the association ambiguities between (A, B) and (D, E) cannot be clarified by the flow vectors. It implies important information loss by using flow maps as the representation of input. A motion encoding scheme is proposed. The displacement volumes are used as the input/output of Behavior-CNN to address association ambiguity across multiple frames and avoid cumulative errors during prediction. As shown in Fig. 1(a), the input to our system is encoded from previous walking paths of all the pedestrians in the scene (blue dots) while the output of Behavior-CNN can recover future walking paths of all these pedestrians (red dots).

The contribution of this paper can be summarized into three-folds. (1) Long-term pedestrian behaviors is modeled with deep CNN. In-depth investigations on the proposed Behavior-CNN is conducted on the learned location map and the location awareness property, semantic meaning of learned filters, and the influence of receptive fields on behavior modeling. (2) A pedestrian behavior encoding scheme is proposed to encode pedestrian walking paths into sparse displacement volumes, which can be directly used as input/output for deep networks without association ambiguities. (3) The effectiveness of Behavior-CNN is demonstrated through applications on path prediction, destination prediction, and tracking.

2 Related Work

2.1 Pedestrian Walking Behavior Modeling

There have been a large number of works on modeling motion patterns. Topic models [18–21] were widely used for modeling crowd flows based on spatio-temporal dependency. Trajectory clustering was another way of learning motion patterns [22,23]. These methods only learned general historical motion statistics of a scene, without modeling the decision making process of each individual.

Katani's work [24] focus on path planning of a single target based on static scene structures. It does not model person-to-person interactions and cannot quickly adapt to varying scene dynamics.

Agent-based models [12,15,17,25,26] could model the decision making process of individuals and their interactions, and were used for simulation, prediction, and abnormal detection. However, historical motion statistics of scenes were not well utilized. Moreover, most agent-based methods used predefined rules. How to design the rules and whether the rules were proper to describe the complex pedestrian behaviors in a particular scene could not be guaranteed.

2.2 Deep Learning

Deep CNNs have shown impressive performance on various vision tasks [27], such as image classification [28], object detection [21,29,30], object tracking [31], and image segmentation [32,33]. However, no deep model has been specially designed for pedestrian behavior modeling. The main difficulty arises from how to design the network input and output, which properly encode pedestrian behavior information and are also suitable for the CNN.

The motion patterns of a whole frame were represented by dense optical flow maps for tasks such as motion segmentation [34], action recognition [35], and crowd scene understanding [36]. As discussed in Sect. 1, ambiguity exists when associating dense optical flows across multiple frames. Someone tried to learn motions directly from video input for human action recognition [37] and video classification [38]. It is not an efficient way of describing pedestrian walking behaviors from raw videos. Some methods used dynamic texture to model video motion [39,40]. They could only capture incremental motion information cross frames, but not long-term motion of pedestrian behaviors. Trajectories were most widely used for pedestrian behavior understanding in non-deep-learning approaches. However, it is not clear how to make them suitable as the input and output of CNN, as they are of variable lengths and observed in different periods.

3 Pedestrian Behavior Modeling and Prediction

The overall framework is shown in Fig. 2. The input to our system is pedestrian walking paths in previous frames (colored curves in Fig. 2(a)). They could be obtained by simple trackers such as KLT [41]. They are then encoded into a displacement volume (Fig. 2(b)) with the proposed walking behavior encoding scheme. Behavior-CNN in Fig. 2(c) takes the encoded displacement volume as

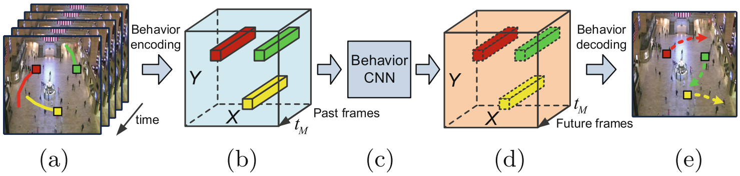


Fig. 2. System flowchart. (a) Pedestrian walking paths in previous frames. Three examples are shown in different colors. Rectangles indicate current locations of pedestrians. (b) The displacement volume encoded from pedestrians' past walking paths in (a). (c) Behavior-CNN. (d) The predicted displacement volume by Behavior-CNN. (e) Predicted future pedestrian walking paths decoded from (d).

input and predict an output displacement volume (Fig. 2(d)) for all the pedestrians simultaneously. A behavior decoding scheme then translates the output displacement volume to future walking paths of all individuals (Fig. 2(e)).

The pedestrian walking behavior encoding scheme is introduced in Sect. 3.1, and Behavior-CNN is discussed in Sect. 3.2. The walking behavior decoding is the inverse process of the encoding. The loss function and training schemes are introduced in Sect. 3.3.

3.1 Pedestrian Walking Behavior Encoding

The walking paths are encoded as displacement volumes and used as input/output for Behavior-CNN. The gap between walking path information and feature representations can be bridged without ambiguity by the proposed encoding scheme.

The encoding process is illustrated in Fig. 3. Let $p_1, ..., p_N$ be N pedestrians in a scene, $t_1, ..., t_M$ be M uniformly sampled time points to be used as input for behavior encoding, and t_M be the current time point. The normalized spatial location of p_i ($i \in [1, N]$) at time point t_m ($m \in [1, M]$) is denoted as $\mathbf{l}_i^m = [x_i^m/X, y_i^m/Y]$, where $x_i^m \in [1, X]$, $y_i^m \in [1, Y]$ are the spatial coordinates of p_i at time t_m, and $[X, Y]$ is the spatial size of the input frames. The locations are grid based and thus discrete. A $2M$-dimensional displacement vector $\mathbf{d}_i = [\mathbf{l}_i^M - \mathbf{l}_i^1, \mathbf{l}_i^M - \mathbf{l}_i^2, ..., \mathbf{l}_i^M - \mathbf{l}_i^{M-1}, \mathbf{l}_i^M - \mathbf{l}_i^M]^T \in \mathbb{R}^{2M}$ is used to describe pedestrian p_i's walking path in the past M frames with respect to t_M (Fig. 3(b)).

The input of CNN is constructed as a 3D displacement volume $\mathcal{D} \in \mathbb{R}^{X \times Y \times 2M}$ based on $\mathbf{d}_i$. For each pedestrian p_i, all the $2M$ channels of $\mathcal{D}$ at p_i's current location (x_i^M, y_i^M) are assigned with the displacement vector $\mathbf{d}_i$. $\mathcal{D}(x_i^M, y_i^M, :) = \mathbf{d}_i + \mathbf{1}^T$, where $\mathbf{1}^T$ represents an all-one vector. All the remaining entries of $\mathcal{D}$ are set as zeros. The elements in $\mathbf{d}_i$ is within the range of $(-1, 1)$. By adding 1, $\mathbf{d}_i$ is transformed to be in the range of $(0, 2)$ before being assigned

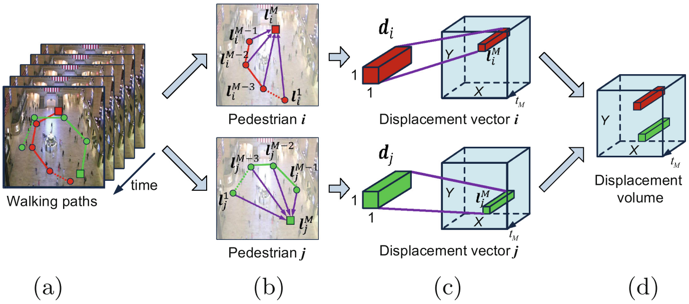


Fig. 3. Illustration of the pedestrian walking behavior encoding scheme. (a) Pedestrian walking paths in the previous M time points, $t_1, ..., t_M$. Two pedestrians, i (red) and j (green) are shown as examples. (b) Spatial locations of each person at these time points, $\mathbf{l}_i^m$ and $\mathbf{l}_j^m$ for $m \in [1, M]$. (c) Computed $2M$-dimensional displacement vector $\mathbf{d}_i$ and $\mathbf{d}_j$ for pedestrians i and j. (d) Encoded displacement volume $\mathcal{D}$ combined from displacement vectors of all pedestrians in the scene. (Color figure online)

to $\mathcal{D}$ so that pedestrians with no movements (1 displacement value in $\mathcal{D}$) can now be distinguished from background locations (0 displacement value in $\mathcal{D}$).

With the proposed encoding process, pedestrian walking path information are well aligned to the current location of this pedestrian ($\mathbf{l}_i^M$ in Fig. 3(c)). All the pedestrians in the scene and their spatial relationships are preserved in $\mathcal{D}$. Importantly, such encoding and its inverse decoding schemes avoid association ambiguity when describing pedestrian walking paths.

3.2 Behavior-CNN

Behavior-CNN takes the displacement volume $\mathcal{D} \in \mathbb{R}^{X\times Y\times 2M}$ as input, and predict future displacement volume ($\mathcal{D}^* \in \mathbb{R}^{X\times Y\times 2M^*}$) as output. $t_1, ..., t_M$ are M previous time points, and $t_{M+1}, ..., t_{M+M^*}$ are M^* future time points to predict. As shown in Fig. 4, Behavior-CNN contains three bottom convolution layers (Fig. 4(b)), one max-pooling layer and an element-wise addition layer (Fig. 4(c)), three top convolution layers (Fig. 4(d)), and one deconvolution layer (Fig. 4(e)). `conv1-5` are followed by `ReLU` nonlinearity layers.

Three bottom convolution layers, `conv1`, `conv2`, and `conv3`, are to be convolved with input data of size $X \times Y \times 2M$. `conv1` contains 64 filters of size $3 \times 3 \times 2M$, while both `conv2` and `conv3` contain 64 filters of size $3 \times 3 \times 64$. Zeros are padded to each convolution input in order to guarantee feature maps of these layers be of the same spatial size with the input. The three bottom convolution layers are followed by max pooling layers `max-pool` with stride 2. The output size of `max-pool` is $X/2 \times Y/2 \times 64$. In this way, the receptive field of the network can be doubled. Large receptive field is necessary for the task of pedestrian walking behavior modeling because each individual's behavior are significantly influenced by his/her neighbors. A learnable location bias map of size $X/2 \times Y/2$ is channel-wisely added to each of the pooled feature maps. Every spatial location has one independent bias value shared across channels. With the location bias map, location information of the scene can be automatically learned by the proposed Behavior-CNN. As for the three top convolution layers, `conv4` and `conv5` contain 64 filters of size $3 \times 3 \times 64$, while `conv6` contains $2M^*$ filters of size $3 \times 3 \times 64$ to output the predicted displacement volume. Zeros are also

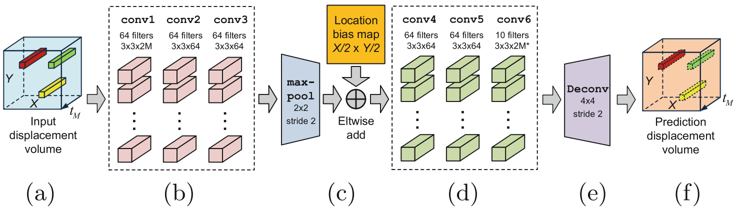


Fig. 4. Behavior-CNN architecture. (a) An input displacement volume $\mathcal{D}$. (b) Three bottom convolution layers. (c) A max-pooling layer and an element-wise addition layer that adds a learnable bias to each location of the feature maps. (d) Three top convolution layers. (e) A deconvolution layer. (f) An output displacement volume $\mathcal{D}^*$.

padded to each convolution input to keep the output spatial size unchanged. Some high-level walking path information and complex walking behaviors of pedestrians are expected to be encoded in the output volume of `conv6`. Finally, a deconvolution layer is used to upsample the output prediction of `conv6` to the same spatial size as the input displacement volume, *i.e.* $\mathcal{D}^* \in \mathbb{R}^{X \times Y \times 2M^*}$.

3.3 Loss Function and Training Schemes

During the training stage, the loss function of Behavior-CNN is defined as the averaged squared L_2 distance between the predicted displacement volume $\widehat{\mathcal{D}^*}$ and the ground truth output displacement volume $\mathcal{D}^*$ on all the valid (non-zero) entries of $\mathcal{D}^*$.

$$\text{Loss} = \frac{1}{\sum \mathcal{M}} ||(\widehat{\mathcal{D}^*} - \mathcal{D}^*) \circ \mathcal{M}||_2^2, \tag{1}$$

where $\circ$ is the Hadamard product operator, and $\mathcal{M}$ is a binary mask. $\mathcal{M}$ is 1 for the entries where $\mathcal{D}^*$ is non-zero, while $\mathcal{M}$ is 0 for the entries where $\mathcal{D}^*$ is zero. $\sum \mathcal{M}$ counts the total number of non-zeros entries of $\mathcal{M}$ for normalization.

The training samples of pedestrian walking paths can be obtained in multiple possible ways. Two strategies are tested in this paper. The annotated pedestrian locations are first used for both model training and evaluation to investigate the properties of the learned Behavior-CNN. Moreover, in order to handle real-world scenarios, our model is also trained with keypoint tracking results by the KLT tracker [41] while the human annotations are only used for evaluation.

Due to the high sparsity of input data, the network may converge to a bad local minimum if all the parameters are trained together from random initialization. Thus a layer-by-layer training strategy is adopted. A simpler network with three convolution layers is first randomly initialized and trained until convergence. Afterwards, the trained convolution layers are used as the bottom layers of Behavior-CNN (`conv1-3`). The following layers (`max-pool`, `eltwise-add`, `conv4-6`, `deconv`) are then appended and parameters of the newly added layers are trained from random initialization. Lastly, all the layers are jointly fine-tuned.

Stochastic gradient descent is adopted for training and the model converged at around 10k iterations. Optimal model is chosen based on a validation set which is a subset of the training samples.

4 Data and Evaluation Metric

Behavior-CNN is evaluated mainly on two datasets. Dataset I is the Pedestrian Walking Route Dataset proposed in [1]. It is 4,000 s in length and 12,684 pedestrians are annotated. Dataset II is collected and annotated by us. We follow the same annotation strategy on Dataset II as in [1]. The complete trajectories of 797 pedestrians from the time point he/she enters the scene to the time he/she leaves are annotated every 20 frames.

To prepare training and testing samples, $M + M^*$ frames at time $t_1, ..., t_M$, $t_{M+1}, ..., t_{M+M^*}$ are uniformly sampled from input videos, and resized to the size of 256×256 ($X = Y = 256$). The first M frames at time $t_1, ..., t_M$ are encoded to the input displacement volumes $\mathcal{D}$ as introduced in Sect. 3.1, which are the input of the Behavior-CNN. The following M^* frames at time $t_{M+1}, ..., t_{M+M^*}$ are encoded to the output displacement volume $\mathcal{D}^*$ as the ground truth.

The encoding of $\mathcal{D}^*$ is similar to that of $\mathcal{D}$. A $2M^*$-dimensional displacement vector $\mathbf{d}_i^* \in \mathbb{R}^{2M^*}$ is used to capture the future path of pedestrian p_i with respect to the current time point t_M, $\mathbf{d}_i^* = [\mathbf{l}_i^M - \mathbf{l}_i^{M+1}, \mathbf{l}_i^M - \mathbf{l}_i^{M+2}, ..., \mathbf{l}_i^M - \mathbf{l}_i^{M+M^*}]$, where $\mathbf{l}_i^m$ is the normalized spatial location of pedestrian p_i at time t_m ($m \in [M+1, M+M^*]$). $\mathcal{D}^* \in \mathbb{R}^{X \times Y \times 2M^*}$ are constructed by assigning $\mathbf{d}_i^*$ to $\mathcal{D}^*$, $\mathcal{D}^*(x_i^M, y_i^M, :) = \mathbf{d}_i^* + \mathbf{1}^T$. With such encoding, future walking path information of each individual is also aligned to the pedestrian current location at time t_M.

By setting different M and M^*, Behavior-CNN can make prediction at different time scales. In our current implementation, M and M^* are both set to 5, *i.e.* five time points are uniformly sampled as input and five future locations of each pedestrian are predicted. The sample interval is 20 frames (0.8 s) for both input and output. That is to say, based on the output result, our model predicts the pedestrian paths in the coming 4 s. Longer-term behaviors can be predicted by recurrently using output again as new input of Behavior-CNN (detailed in Sect. 6.2). With larger M values and more computation cost, performance should be slightly improved because more information is given.

4990 short clips are uniformly segmented from Dataset I and one sample can be obtained from each clip. For Dataset II, 550 samples are generated. The first 90 % samples are used for training while the remaining for test on both datasets.

Mean squared error (MSE) is adopted as the evaluation metric for the task of pedestrian walking path prediction. The average L_2 distance between normalized predicted pedestrian locations and normalized ground-truth pedestrian locations of all the N pedestrians at all the M^* predicted time points are computed.

$$\mathrm{MSE} = \frac{1}{NM^*} \sum_{i=1}^{N} \sum_{m=1}^{M^*} ||\mathbf{l}_i^{M+m} - \widehat{\mathbf{l}}_i^{M+m}||_2 \times 100\,\%, \tag{2}$$

where $\widehat{\mathbf{l}}_i^{M+m} = [x_i^{M+m}/X, y_i^{M+m}/Y]$ is the normalized location of p_i at time t_{M+m} with respect to the size of the scene.

5 Investigations on Behavior-CNN

In-depth investigations are conducted on Behavior-CNN. It reveals underlying properties of the proposed deep behavior model. Human annotated pedestrian walking paths are used to train the models in this section.

5.1 Bias Map and Location Awareness Property of Behavior-CNN

For a specific scene, different locations generally have different traffic patterns because of scene structures. The proposed bias map helps capture such information. Experiments are conducted to investigate the effect of the location

Table 1. (a) Prediction results with/without the location bias map. (b) Prediction results of different flipping strategies.

Investigations on	MSE	Dataset I	Dataset II
(a) Location bias map	With	2.421 %	2.348 %
	Without	2.703 %	2.628 %
(b) Flipping strategies	No flipping	2.421 %	2.348 %
	Horizontal flipping	2.470 %	2.592 %
	Vertical flipping	2.468 %	2.585 %
	Horizontal and vertical flipping	2.502 %	2.668 %

bias map. The errors of the proposed method with/without the bias map are listed in Table 1(a). Without the bias map, prediction errors increase for both datasets.

One more experiment is conducted to validate the location awareness of Behavior-CNN. Given the trained model (with location bias map) fixed, testing samples are flipped horizontally and/or vertically, and the results of different flipping strategies are reported in Table 1(b). If the prediction of our model has location invariance, flipping all the pedestrian paths at all the locations in the same way will not make difference on prediction errors. However, Table 1(b) shows that prediction error increases if testing samples are flipped, which indicates different locations have different dependence on moving directions.

With the learned location bias map, our Behavior-CNN can distinguish different locations of the scene based on the motion patterns of small regions (receptive field size of the Behavior-CNN). In Fig. 5(b), the scene is segmented into 8 by 8 grids. For each grid, the distributions of the walking directions of all the training samples, together with the distributions of the walking directions of all the predicted paths by Behavior-CNN are computed. Two example grids are shown in Fig. 5(a) and (c). Three types of walking patterns, moving up, down, and left, are observed frequently for the "crossing" grid (Fig. 5(a)). Two types of walking patterns, moving up and moving down, are common patterns in the "corridor" grid (Fig. 5(c)). Strong correlations are observed between the predicted walking pattern and the training walking pattern. The correlation for the crossing grid (Fig. 5(a)) is 0.88 while the correlation for the corridor grid (Fig. 5(c)) is 0.91. With the location bias map, our learned model is able to capture the location information and scene layout from the input pedestrian walking paths, such as the patterns shown in Fig. 5(a) and (c).

Based on location awareness, our model can successfully infer scene structures from local motion patterns in input and the learned location bias map. The pedestrian spatial distributions of training samples, which reflect scene layout, and our model's predictions are shown in Fig. 5(d). Strong correlations are observed between them, which demonstrates that our model can capture the scene layout information. From the prediction distribution, some impossible locations such as scene obstacles can be automatically distinguished by Behavior-CNN.

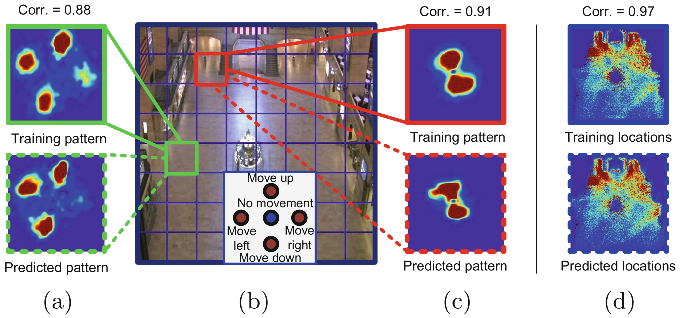


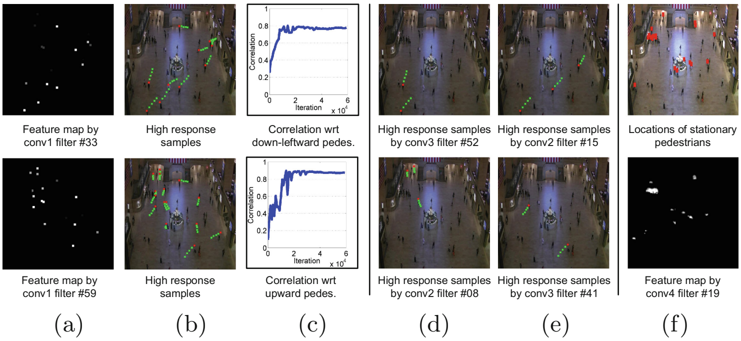
Fig. 5. Investigation on location awareness of Behavior-CNN. (a–c) Behavior-CNN can capture different motion patterns for different regions of the scene. The scene is segmented into 8 by 8 grids in (b). The motion patterns of training samples and prediction results of the "crossing" grid (green) is shown in (a) and those of the "corridor" grid (red) are shown in (c). Warmer color indicates higher frequency of corresponding motion as indicated in (b). (d) Strong correlation can also be observed between the spatial distributions of the training samples, which reflect scene structures and the existence of obstacles, and the spatial distributions of predictions. (Color figure online)

5.2 Learned Feature Filters of Behavior-CNN

From feature maps generated by filters in different layers, strong correlations between specific walking patterns and filter response maps can be well observed. Generally speaking, the three bottom convolution layers (`conv1-3`) take all the pedestrian behaviors as input and gradually classify them into finer and finer categories according to various criteria. In top layers, the influences of all different categories are combined together to generate the prediction.

For bottom convolution layers, different pedestrians are roughly classified by filters based on their walking behaviors. Examples are shown in Fig. 6(a–c). Two feature maps generated from filter #33 and filter #59 of `conv1` are shown in Fig. 6(a). The high-response pedestrians in the two feature maps are visualized in Fig. 6(b). It is observed that most pedestrians with high response to filter #33 move down-leftwards, while pedestrians with high response to filter #59 move upwards. In this way, the input pedestrian paths can be classified into some rough categories by the filters in `conv1`. We computed the correlations between the feature maps by the two filters (Fig. 6(a)) and the locations of all moving down-leftwards/upwards pedestrians at different training iterations. As shown by the correlation curves in Fig. 6(c), the two filters gradually learned to capture these specific motion patterns during training.

Some high-response pedestrians by filters of `conv2` and `conv3` are shown in Fig. 6(d–e). These filters generally classify pedestrians into finer and more specific categories compared with those of `conv1`. In Fig. 6(d), down-leftward/upward pedestrians in Fig. 6(a) are further classified based on spatial locations, such as the left-bottom corner and the left-up corner. In Fig. 6(e), pedestrians are more meticulously classified based on precise moving directions.


Feature map by conv1 filter #33

High response samples

Correlation wrt down-leftward pedes.

High response samples by conv3 filter #52

High response samples by conv2 filter #15

Locations of stationary pedestrians

Feature map by conv1 filter #59

High response samples

Correlation wrt upward pedes.

High response samples by conv2 filter #08

High response samples by conv3 filter #41

Feature map by conv4 filter #19

Fig. 6. Investigations on learned filters. (a) Two feature maps generated by filter #33 and filter #59 of `conv1`. (b) Input pedestrian walking paths with high responses on the feature maps in (a). Red dots indicate current locations. Filter #33 corresponds to moving down-leftwards while filter #59 corresponds to moving upwards. (c) Correlation values between the feature maps in (a) and the location maps of all down-leftwards/upward pedestrians in the scene at different training iterations. (d–e) Some high response pedestrians by filters of `conv2-3`. (f) Stationary pedestrians captured by the feature map of filter #19 in `conv4`. (Color figure online)

For filters in higher-level layers, they generally encode more complex behaviors. As shown by one example in Fig. 6(f), stationary pedestrians are assigned with high-responses by the filter #19 of `conv4`, which demonstrates that stationary crowds could influence other pedestrians' walking patterns.

5.3 Receptive Fields

We observe that pedestrian walking behaviors are significantly influenced by nearby pedestrians. By increasing the size of the receptive field, the sensing range of the network can be increased and the predictions are more reliable. The current receptive field size is around 10 % of the scene, which is large enough to capture the pedestrians and activities within their nearby regions.

Two alternative net structures are designed to decrease the receptive field size. (a) The filter size of all layers is changed from 3 × 3 to 1 × 1. In order to keep the same parameter size, the numbers of filters are all increased by 9 times in the meanwhile. (b) The proposed net structure (`3conv+pool+3conv+deconv`) is simplified to `3conv+pool+3conv` and `3conv+3conv` by removing some layers. The alternatives are used to demonstrate the power of large receptive field size when predicting future pedestrian walking behaviors.

The results of different net structures are shown in Table 2. With the same model complexity, the prediction error increases for the 1 × 1 filters compared with the 3×3 filters. Moreover, the better performance of the `3conv+pool+3conv` structure compared with the `3conv+3conv` structure also demonstrates the effectiveness of large receptive field introduced by the pooling layer.

Table 2. Prediction results (MSE) of different net structures on Dataset I.

	3×3 (ours)	1×1
`3conv+pool+3conv+deconv` (ours)	2.421 %	2.555 %
`3conv+pool+3conv`	2.431 %	2.571 %
`3conv+3conv`	2.468 %	2.858 %

6 Experiments

6.1 Pedestrian Walking Path Prediction

The prediction results of the proposed Behavior-CNN are evaluated quantitatively and qualitatively for both Dataset I and Dataset II. For each of the dataset, two trained models were evaluated. One was trained with the human annotated pedestrian locations and the other one was trained with KLT trajectories. The trajectories are not verified and may contain mistakes. All the models are evaluated using the annotated ground truth pedestrian walking paths. Due to the insufficient training samples of Dataset II, the models trained on Dataset I were used as the initial points to train the models for Dataset II.[1]

Three baselines and three state-of-the-art methods [2,15,17] on pedestrian behavior prediction were compared. The constant velocity and constant acceleration regressors were used as the first two baselines to predict future walking path of each pedestrian. Moreover, the same displacement vectors were used as features and a second-order SVM regressor was used for prediction. Existing computer vision methods in comparison include the Social Force Model (SFM) [15] where pedestrian walking paths were predicted as its simulation results, the Linear Trajectory Avoidance model (LTA) [17] where pedestrian walking paths were predicted based on energy minimization, and Temporal Information Model (TIM) [2] where pedestrian walking paths were predicted as the minimal paths.

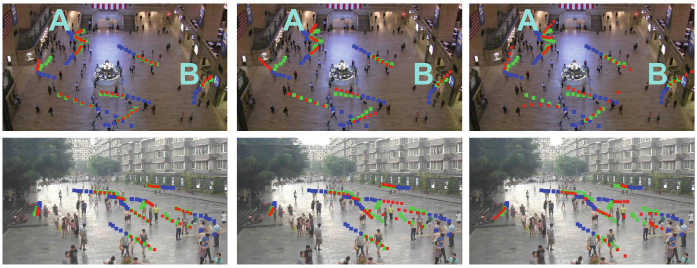
MSE introduced in Sect. 4 was evaluated and the results are reported in Table 3. Behavior-CNN achieves the best performance among all the comparisons. This is because the learned feature representations of Behavior-CNN are much more powerful and can capture complex pedestrian behaviors. The model trained with annotations (2.421 %) performs only slightly better than that trained with KLT (2.517 %) on Dataset I, which also demonstrates the robustness of the proposed method to KLT errors.

Several examples of prediction results are visualized in Fig. 7. Behavior-CNN can successfully predict some complex walking patterns, such as change of walking directions, slowing down, speeding up (Pedestrian A in Fig. 7(a)). It also learns the scene layout, which cannot be learned by the other two methods from

[1] The model trained solely with annotations on Dataset I generates a 4.18 % error if directly testing on Dataset II, which is still better than the comparisons. However, with bias map removed, the error decreases to 3.42 %. It indicates that the bias map hinders model transfer ability to a certain degree.

Table 3. Prediction results (MSE) of different methods trained on the annotated pedestrian walking paths or the KLT trajectories on Dataset I and Dataset II.

	Dataset I	Dataset II	Dataset I	Dataset II
	(Annotation)	(Annotation)	(KLT)	(KLT)
Behavior-CNN	2.421 %	2.348 %	2.517 %	3.816 %
Constant velocity	6.091 %	6.468 %	5.864 %	5.635 %
Constant acceleration	9.899 %	9.428 %	6.619 %	7.656 %
SVM regression	4.639 %	4.276 %	5.053 %	5.327 %
SFM [15]	4.280 %	5.921 %	4.447 %	5.044 %
LTA [17]	4.723 %	4.571 %	4.346 %	4.639 %
TIM [2]	4.075 %	4.141 %	4.790 %	4.790 %



(a) Behavior-CNN (b) Constant velocity (c) LTA [17]

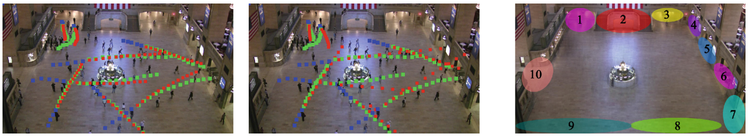
Fig. 7. Prediction results by (a) Behavior-CNN, (b) the constant-velocity model, and (c) the LTA model, with KLT trajectories as input on both datasets. The KLT trajectories are used to train the model. Input previous locations, ground truth future locations, and predicted future locations are marked by blue, green and red dots, respectively. (Color figure online)

training samples. Taking Pedestrian B in Fig. 7 as an example, our prediction avoids scene obstacles while the predictions by the other two methods indicate the pedestrian walking into a concrete wall.

In order to validate prediction robustness, the proposed method is also evaluated on five more datasets, *i.e.* ETH [17], Hotel [17], ZARA01 [42], ZARA02 [42], and UCY [42]. Following the same experimental setup and evaluation criteria as [43], leave-one-out validation is adopted and average displacement errors of our proposed method on the five datasets are 0.35, 0.18, 0.20, 0.23, and 0.25, while [43] achieves 0.50, 0.11, 0.22, 0.25, and 0.27.

6.2 Application I: Pedestrian Destination Prediction

Behavior-CNN is able to predict the walking paths of all the pedestrians in the scene for in the next 4 s ($M^* = 5$). However, by decoding the output displacement



(a) Behavior-CNN (b) SFM [15] (c) Entrance/exit regions

Fig. 8. (a–b) Long-term path prediction results in Dataset I using Behavior-CNN and Social Force Model [15]. Behavior-CNN is recurrently forward-propagated three times and locations at 15 future time points are predicted. Input previous locations, ground-truth future locations, and predicted future locations are marked by blue, green, and red dots. (c) Ten entrance/exit regions labeled in Dataset I [1]. (Color figure online)

Table 4. *Top-N* accuracies of destination prediction on Dataset I.

	Top 1	Top2	Top3
Behavior-CNN	53 %	72 %	84 %
EMM [1]	48 %	69 %	83 %
MDA [8]	43 %	-	-
UVP [44]	45 %	-	-

volume and re-encoding the prediction results, the predicted walking paths can be fed back into Behavior-CNN as input. In this way, long-term walking paths can be recurrently predicted. The prediction results of several pedestrians by Behavior-CNN and the Social Force Model [15] are shown in Fig. 8(a) and (b). Behavior-CNN can predict reasonable long-term walking paths.

The long-term prediction results can be used for destination prediction. The destination is determined as the nearest exit to the predicted future walking path. Prediction performance was evaluated on Dataset I, where ten entrance/exit regions are labeled [1] as shown in Fig. 8(c). The *top-N* accuracy (ground truth is within the top-N predictions) was adopted for evaluation.

Three existing methods were used as comparisons, *i.e.*, the energy map modeling approach (EMM) [1] where destinations were predicted by minimizing energy function, MDA [8] where predictions were made based on trajectory properties, and an unsupervised visual prediction approach (UVP) [44] where destinations were predicted as the nearest exit to the predicted trajectories. In order to make fair comparisons, all the methods use previous 5 frames (4 s in length) as input. Estimation results are reported in Table 4. Our method performs better as it can better predict long-term motion patterns.

6.3 Application II: Predictions as Tracking Prior

Based on predicted pedestrian walking paths, Behavior-CNN can provide prior information to improve tracking. The KLT tracker, whose trajectories are often fragmented or early terminated, is used as a baseline tracking algorithm to

Table 5. Results of pedestrian tracking on Dataset I

Methods	KLT+Behavior-CNN	KLT+RFT [45]	KLT
Error (L_2 distance)	83.79	228.33	411.71

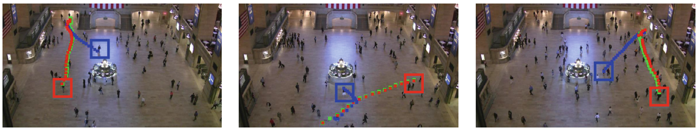


Fig. 9. Improved pedestrian tracking results by Behavior-CNN (red dots) and RFT [45] (blue dots). Ground truth trajectories are shown as green dots. Successfully tracked pedestrians of the proposed method and mis-tracked pedestrians by the RFT method are marked by the red and blue rectangles. (Color figure online)

be improved. A tracking association strategy is adopted when a key point fails to be tracked. Given successfully tracked locations (up to the failing time) as input, $M^* = 5$ future locations (4 s) can be predicted by Behavior-CNN. Then the tracklet that best matches prediction is selected to be connected with the fragmented tracklet. In this way, long-term trajectories could be formed by connecting fragmented ones and tracking performance can be improved. Another association strategy in [45] was also used for comparison (RFT). Trajectories are connected based on the local location and speed information when tracking fails.

The average L_2 distance between ground truth walking paths and tracking results of 1000 pedestrians in Dataset I were used for evaluation. The results of both strategies, together with the results of the baseline KLT tracking are listed in Table 5. The proposed association strategy significantly decreases the tracking error compared with RFT [45]. From the examples in Fig. 9, our method could successfully generate correct and complete trajectories, while the association by the RFT method made wrong associations and lost the tracking targets.

7 Conclusion

Behavior-CNN is proposed to model pedestrian behaviors. A behavior encoding scheme is adopted to encode pedestrian behavior into sparse displacement volumes which can be directly used as network input. Behavior-CNN is thoroughly investigated in terms of the learned location map and the location awareness property, semantic meanings of learned filters, and influence of receptive fields. The effectiveness is demonstrated through multiple applications, including walking path prediction, destination prediction, and improving tracking.

Acknowledgment. This work was supported in part by the Ph.D. Programs Foundation of China under Grant 20130185120039, in part by the Hong Kong Innovation and Technology Support Programme under Grant ITS/221/13FP, in part by the National Natural Science Foundation of China under Grant 61371192 and Grant 61301269, and in part by the General Research Fund through the Research Grants Council, Hong Kong, under Grant CUHK14206114, Grant CUHK14205615, Grant CUHK419412, and Grant CUHK14203015.

References

1. Yi, S., Li, H., Wang, X.: Understanding pedestrian behaviors from stationary crowd groups. In: Proceedings of CVPR (2015)
2. Cancela, B., Iglesias, A., Ortega, M., Penedo, M.: Unsupervised trajectory modelling using temporal information via minimal paths. In: Proceedings of CVPR (2014)
3. Alahi, A., Ramanathan, V., Fei-Fei, L.: Socially-aware large-scale crowd forecasting. In: Proceedings of CVPR (2014)
4. Yi, S., Li, H., Wang, X.: Pedestrian behavior modeling from stationary crowds with applications to intelligent surveillance. TIP **25**(9), 4354–4368 (2016)
5. Tang, S., Andriluka, M., Milan, A., Schindler, K., Roth, S., Schiele, B.: Learning people detectors for tracking in crowded scenes. In: Proceedings of ICCV (2013)
6. Shu, G., Dehghan, A., Oreifej, O., Hand, E., Shah, M.: Part-based multiple-person tracking with partial occlusion handling. In: Proceedings of CVPR (2012)
7. Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S.: Learning an image-based motion context for multiple people tracking. In: Proceedings of CVPR (2014)
8. Zhou, B., Wang, X., Tang, X.: Understanding collective crowd behaviors: learning a mixture model of dynamic pedestrian-agents. In: Proceedings of CVPR (2012)
9. Nascimento, J.C., Marques, J.S., Lemos, J.M.: Modeling and classifying human activities from trajectories using a class of space-varying parametric motion fields. TIP **22**(5), 2066–2080 (2013)
10. Kim, K., Lee, D., Essa, I.: Gaussian process regression flow for analysis of motion trajectories. In: Proceedings of ICCV (2011)
11. Chang, M.C., Krahnstoever, N., Ge, W.: Probabilistic group-level motion analysis and scenario recognition. In: Proceedings of ICCV (2011)
12. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: Proceedings of CVPR (2009)
13. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 FPS in matlab. In: Proceedings of ICCV (2013)
14. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: Proceedings of CVPR (2010)
15. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282 (1995)
16. Yi, S., Wang, X., Lu, C., Jia, J., Li, H.: L0 regularized stationary-time estimation for crowd analysis. TPAMI **PP**(99), 1 (2016). doi:10.1109/TPAMI.2016.2560807
17. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: modeling social behavior for multi-target tracking. In: Proceedings of ICCV (2009)
18. Kuettel, D., Breitenstein, M.D., Van Gool, L., Ferrari, V.: What's going on? Discovering spatio-temporal dependencies in dynamic scenes. In: Proceedings of CVPR (2010)
19. Wang, X., Ma, X., Grimson, W.E.L.: Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models. TPAMI **31**(3), 539–555 (2009)

20. Hospedales, T.M., Li, J., Gong, S., Xiang, T.: Identifying rare and subtle behaviors: a weakly supervised joint topic model. TPAMI **33**(12), 2451–2464 (2011)
21. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. In: Proceedings of CVPR (2008)
22. Morris, B.T., Trivedi, M.M.: Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach. TPAMI **33**(11), 2287–2301 (2011)
23. Wang, X., Ma, K.T., Ng, G.W., Grimson, W.E.L.: Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models. IJCV **95**(3), 287–312 (2011)
24. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 201–214. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33765-9_15
25. Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. PNAS **99**(Suppl 3), 7280–7287 (2002)
26. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407**(6803), 487–490 (2000)
27. Bengio, Y.: Learning deep architectures for AI. Found. Trends® Mach. Learn. **2**(1), 1–127 (2009)
28. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of NIPS (2012)
29. Girshick, R.: Fast R-CNN. In: Proceedings of ICCV (2015)
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of NIPS (2015)
31. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: Proceedings of NIPS (2013)
32. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. TPAMI **35**(8), 1915–1929 (2013)
33. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of CVPR (2015)
34. Reddy, N.D., Singhal, P., Krishna, K.M.: Semantic motion segmentation using dense CRF formulation. In: Proceedings of ICVGIP (2014)
35. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Proceedings of NIPS (2014)
36. Shao, J., Kang, K., Loy, C.C., Wang, X.: Deeply learned attributes for crowded scene understanding. In: Proceedings of CVPR (2015)
37. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. TPAMI **35**(1), 221–231 (2013)
38. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of CVPR (2014)
39. Yan, X., Chang, H., Shan, S., Chen, X.: Modeling video dynamics with deep dynencoder. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 215–230. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10593-2_15
40. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using lstms (2015). arXiv preprint arXiv:1502.04681
41. Tomasi, C., Kanade, T.: Detection and tracking of point features. School of Computer Science, Carnegie Mellon Univ. Pittsburgh (1991)

42. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum, vol. 26, pp. 655–664. Wiley Online Library (2007)
43. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: human trajectory prediction in crowded spaces. In: Proceedings of CVPR (2016)
44. Walker, J., Gupta, A., Hebert, M.: Patch to the future: unsupervised visual prediction. In: Proceedings of CVPR (2014)
45. Zhou, B., Wang, X., Tang, X.: Random field topic model for semantic region analysis in crowded scenes from tracklets. In: Proceedings of CVPR (2011)