Using Graph Convolutional Networks Skeleton-Based Pedestrian Intention Estimation Models for Trajectory Prediction

Dong Cao* and Yunbin Fu

DeepBlue Technology (Shanghai) Co., Ltd. No.369, Weining Road, Shanghai, China
Email: doocao@gmail.com

Abstract. In autonomous driving scenarios, pedestrian trajectory prediction is an important research direction. Based on the spatio-temporal graph convolutional neural network, we propose a new pedestrian trajectory prediction algorithm. The new algorithm constructs a series of new models around pedestrian intention estimation. The construction of the estimation algorithm considers the following aspects: the contextual information of pedestrians and the surrounding environment, the “pedestrian ego-vehicle” interaction combined with the vehicle speed estimation, the pedestrian’s own skeletal structure information and body language estimation, which includes head joints and the relative structural relationship of the torso joints, including whether it is out of the same plane, is rotated, and so on. Skeleton information feature extraction and construction adopts the method of graph convolutional neural network to structure pedestrians into joints in the form of graphs in non-Euclidean space, and further adopts spatial temporal graph convolutional network for feature extraction and learning. The new method is named a “head-torso”-based spatial temporal graph convolutional network (HT-STGCN). On the dataset PID, the novel method achieves substantial improvements over mainstream methods. Experimental results show that combining HT-STGCN with observed action can improve trajectory prediction.

1. Introduction

In autonomous driving scenarios and the field of Advanced Driver Assistance Systems (ADAS) for urban environments, it is important to predict pedestrian intent. Based on the pedestrian intent prediction, the pedestrian trajectory can be further estimated to understand its next actions, which can greatly reduce the risk of accidents. Ref. [1] proposed an image-based 2D pose estimation algorithm. The algorithm is used for pedestrian intention estimation. Relevant behaviours include crossing the road in front of the ego-vehicle, stopped suddenly before entering the road, and so on. They evaluate the effectiveness of the 2D pose estimation algorithm on the JADD dataset [2], the proposed pipeline provides satisfactory results.

Based on two-dimensional pose estimation, Fang et al. [3] proposed an algorithm for pedestrians and cyclists intention recognition. Ref. [4] proposed Composite Fields method, this method can be effectively used for human pose estimation.

We focus on two types of areas, one is at intersections and nearby areas, and the other is bus stops and nearby areas. These areas are high incidence areas for pedestrians to cross the road, so pedestrian behaviour and intentions in these areas are the focus of our attention. Pedestrian behaviour in these areas can consist of many types: Pedestrians who simply reside in the area and do not intend to cross the road. Pedestrians who wander in the area and attempt to cross the road but have not yet crossed the road...
immediately. Pedestrians intend to cross the road and implement crossing the road. Pedestrians look around on the road near the bus platform and look forward to the arrival of the bus have no intention to cross the road, but they need to be highly alert for driving vehicles. The specific details are shown in figure 1.

![Multiple types of pedestrian behaviour in urban traffic environments.](image)

**Figure 1.** Multiple types of pedestrian behaviour in urban traffic environments.

Action recognition methods include RGB-based methods and optical flow-based methods. In these research methods, many scholars have proposed a variety of optimization algorithms [5-9].

Based on the feature information of the skeleton and joints, action recognition has obvious advantages. For example, when the lighting conditions change significantly, for the same action, the RGB information has changed significantly, but the information of the skeleton and joints still has distinctive features, which can ensure the model has good generalization performance. In a variety of application scenarios, action recognition algorithms based on joint trajectories and skeleton have excellent robustness [10].

Under the existing technical conditions, it is no longer difficult to obtain the skeleton and joint track feature information [10]. Joint features can be obtained by means of professional depth image capture equipment, or related pose estimation algorithm and other means.

2. Related Work

2.1. Skeleton-Based Action Recognition
Shotton et al. proposed a method for real-time human pose recognition based on the features of body parts in a single depth image [11], which combined depth information without using temporal features to predict the human body from a single image 3D position of the joint. The pose estimation problem is transformed into a pixel classification problem. Good generalization ability is achieved on accurate full skeleton nearest neighbor matching. Du et al. [12] proposed a hierarchical recurrent neural network method based on skeletal features to achieve better performance of action recognition.

Using skeleton feature information, Yan et al. proposed an action recognition algorithm based on spatial temporal graph convolutional networks [13]. This algorithm can automatically learn spatio-temporal features from data, surpassing previous methods in this regard. This method has stronger generalization ability. In the Kinetics dataset and the NTU-RGBD dataset [14-16], Novel algorithm have achieved better performance than other advanced methods. Spatial temporal graph convolutional networks [17, 18] have been extensively and intensively studied recently and have been applied in many fields with satisfactory results. Using actional-structural graph convolutional networks, Li et al. proposed a method of skeleton-based action recognition [15]. In their proposed method, an encoder-decoder structure is introduced. This method can capture richer dependencies. This module can capture action-specific latent dependencies, include actional links.

2.2. Graph Representation of Joint Nodes
Ref. [13] introduced a method of constructing a Spatial temporal graph convolutional network, using human skeleton joints as modeling objects, combining the concepts of joints and body parts to achieve more efficient algorithm performance [14, 19, 20]. The construction of the skeleton-based graph convolutional neural network model in Ref. [13] can be divided into two steps: first, the design of the spatial graph convolution network; second, based on the spatial graph network, taking into account the
time series factors, focusing on the trajectories of the same joints between consecutive frames, and their contribution to model feature information extraction.

In the aspect of the design of spatial graph convolution network, each frame in the video is taken as the research object. The purpose is to extract the spatial feature information of every frame, especially the features of pedestrian object. The concept of “body parts” is used to realize the local limitation of joint trajectories, which can effectively realize the hierarchical representation of skeletal sequences, and further can realize the automatic extraction of object features through convolutional networks.

Skeleton information can be obtained using the OpenPose toolbox [10] for 2D pose estimation. For each pedestrian in each frame of image, we usually set to extract 18 joint points to form a set of joint coordinate sets. Each joint acts as a node. The set of nodes can be expressed as:

$$V = \{v_{ti}| t = 1, \ldots, T, i = 1, \ldots, N\}$$  \hspace{1cm} (1)

where N is the number of nodes. The natural connection between the various joints of the body part is regarded as edges. The set of edges includes two parts, the subset of edges in the frame and the subset of edges between consecutive frames. The intra-frame edge set refers to a set of skeletal edges composed of N joint points on a frame at time $\tau$, which can be expressed as:

$$E_S(\tau) = \{v_{ti}v_{tj}| t = \tau, (i, j) \in H\}$$  \hspace{1cm} (2)

where $H$ is the set of natural connections of human joints. The subset of edges between consecutive frames refers to the same joint-connected edges between frames, expressed as:

$$E_F = \{v_{ti}v_{(t+1)i}\}$$  \hspace{1cm} (3)

2.3. Neighbor Set and Sampling Function
The neighbour set of the node $v_{ti}$ is represented as:

$$B(v_{ti}) = \{v_{tj}| d(v_{tj}, v_{ti}) \leq D\}$$  \hspace{1cm} (4)

where $d$ represents the shortest distance between two nodes. Generally, if $D = 1$ is set, then the sampling function has the following form:

$$p: B(v_{ti}) \rightarrow V$$  \hspace{1cm} (5)

which can be denoted as:

$$p(v_{ti}, v_{tj}) = v_{tj}$$  \hspace{1cm} (6)

For the spatial temporal graph with the above structure, the definition of its weight function is based on the methods introduced in Refs [21, 22]. The neighbor set $B(v_{ti})$ of a joint node $v_{ti}$ is divided into $K$ subsets, and the $K$ subsets are numbered separately. The obtained mapping is expressed as:

$$l_{ti}: B(v_{ti}) \rightarrow \{0, \ldots, K - 1\}$$  \hspace{1cm} (7)

Then, by indexing a $(c, K)$ -dimensional tensor, the weight function can be implemented as $w(v_{ti}v_{tj}): B(v_{ti}) \rightarrow R^c$.

3. Method
The new method we constructed is skeleton-based spatio-temporal graph convolutional network. We propose a “head-torso” partitioning strategy to achieve efficient pedestrian intent estimation and further improve the performance of pedestrian trajectory prediction models.
3.1. Pose Estimation Based on Skeleton
The input to the model is a 2D skeleton of a pedestrian, which is represented as a 2D coordinate. This coordinate is obtained by detecting the pedestrian object in the image and then using a deep neural network algorithm to detect it. The algorithm uses the method of Ref. [10]. Use the OpenPose toolbox provided by the algorithm to get 2D pose estimation results. Based on the pose estimation results, we can get 18 joint coordinates of each pedestrian. Based on the detected human joint point coordinates, nodes and edges can be further defined to construct a spatial temporal graph.

3.2. Head-Torso Partitioning Strategy
On the basis of the skeleton-based spatial temporal graph completed above, further research found that [13], in the representation of the spatial temporal graph convolutional network, it is very important to study and optimize the partition strategy, the related partition strategy directly affects the implementation of the label map. Ref. [13] introduced three common partitioning strategies, namely Uni-labeling, Distance partitioning, and Spatial configuration partitioning. For the first partitioning strategy, Uni-labeling, the number of subsets k = 1, and the mapping \( l_{ti}: \mathcal{B}(v_{ti}) \rightarrow \{0, \ldots, K - 1\} \) is simplified as \( l_{ti}(v_{tj}) = 0, \forall t, j \in V \). The second partitioning strategy, Distance partitioning, the number of subsets k = 2, \( l_{ti}(v_{tj}) = d(v_{tj}, v_{ti}) \), the mapping is simplified as \( l_{ti}: \mathcal{B}(v_{ti}) \rightarrow \{0, 1\} \). The specific details are shown in figure 2.

**Figure 2.** Distance partitioning, subset of root node (blue, distance 0), the other subset of neighbor joints (green, distance 1).

The third partitioning strategy, Spatial configuration partitioning. The number of subsets k = 3, the mapping is simplified to \( l_{ti} \rightarrow \{0, 1, 2\} \).

Among the above three partitioning strategies, Spatial configuration partitioning considers centripetal group and centrifugal group, and achieves the best performance of the three known methods. Based on a comparative analysis of the above three partitioning strategies, we propose the use of a ‘Head-Torso’ spatial configuration partitioning model and a Head-Torso spatial temporal graph convolutional network called HT-STGCN as shown in figure 3 below.

**Figure 3.** Head-torso spatial configuration partitioning, this partitioning includes two subsets. One of the subsets is composed of head nodes (green) and the other is composed of torso nodes (orange).
Constructed “head joint point group” (for the head) and “torso joint point group” (outside the head). The connections between head joint nodes are shown as adjacency matrix $A_H$, and the connections between torso joint nodes are shown as adjacency matrix $A_T$. This “head-torso” partitioning strategy specifically examines the positional relationship of face orientation relative to body posture and can provide steering and action information when pedestrians try to cross the road. For example, the head node group plane and the body joint point group are not in the same plane, indicating that the pedestrian is looking at the left or right, then there is a reason to judge that the possibility of crossing the road has increased significantly.

Implementation of ST-GCN based on “head-torso” partitioning strategy, combined with the construction methods of Refs. [13, 23]. The connection between the joints of each pedestrian in a single frame can be expressed as an adjacency matrix $A$ and an identity matrix $I$ represents self-connection. But for partitioning based on “head-torso”, there are partitioned adjacency matrices $A_H$ and $A_T$.

$$f_{out} = \sum_j \Lambda_j^{-\frac{1}{2}} (A_j \otimes M_j) \Lambda_j^{-\frac{1}{2}} f_{in} W_j$$

$$= \Lambda_0^{-\frac{1}{2}} (A_0 \otimes M_0) \Lambda_0^{-\frac{1}{2}} f_{in} W_0 + \Lambda_H^{-\frac{1}{2}} (A_H \otimes M_H) \Lambda_H^{-\frac{1}{2}} f_{in} W_H + \Lambda_T^{-\frac{1}{2}} (A_T \otimes M_T) \Lambda_T^{-\frac{1}{2}} f_{in} W_T$$

(8)

where $A + I = \sum_j A_j$, the adjacency matrix $A$ is split into multiple matrices $A_0$, $A_H$ and $A_T$. $M_j$ represents a learnable weight matrix. $f_{in}$ represents an input feature map, and $f_{out}$ represents an output feature map.

The importance of joints in different parts of the body is different. The learnable weight matrix is used to differentiate the importance expressions of joints in various parts of the body to improve the performance of the model. This “head-torso” partitioning design method assigns different importance weights to edges, such as the “head and neck” joints describing whether a pedestrian intends to turn around the road. It can effectively improve the performance of pedestrian intent estimation. Based on a good pedestrian intent estimation algorithm, further, HT-STGCN performs skeleton-based action recognition and expects to improve pedestrian trajectory prediction.

3.3. Pre-trained Model of HT-STGCN for Pedestrian Trajectory Prediction

According to the 6 types of actions that have been marked in the PIE dataset [24] (walking, standing, looking, not looking, crossing, not crossing). We train HT-STGCN on the PIE dataset [24]. In addition, based on the PIE dataset, we added a dataset of Chinese street scenes and annotated the classification of pedestrian action. The expanded dataset enhanced the model's generalization ability for Chinese pedestrian crossing styles.

The training result is used as a pre-trained model to input the Decoding module of the “Intention Estimation”. The model architecture is shown figure 4. The HT-STGCN module structure draws on the construction method of “Semi-Supervised GCN” [23].

![Figure 4. Vehicle speed prediction, intention estimation and trajectory prediction.](image-url)
The model is divided into three functional modules, the first is the vehicle speed prediction module, the second is the intention estimation module, and the third is the trajectory prediction module. Among them, the first and third modules adopt the same structure as the algorithm in Ref. [24]. The first module is Vehicle speed prediction, and its input is the current speed sequence of the ego-vehicle, expressed as \(\{s^t, \ldots, s^{t-m+1}, s^{t-m}\}\). The output is the predicted ego-vehicle speed sequence \(\{s^{t+1}, s^{t+2}, \ldots, s^{t+r}\}\). Its prediction speed is input into the third Trajectory prediction module. The second module is Intention Estimation, where the Encoder part is improved by combining two modules, Encoder-1 and Encoder-2. The input of the module Encoder-1 is based on the “head-torso” partition of the pedestrian joint point coordinate tensor. The input of the module Encoder-2 is a sequence of square cropped images around the pedestrians. After the results of Encoder-1 and Encoder-2 are combined, the pedestrian box coordinates information is combined and used as the Encoder result. The result of the Encoder is input to the Decoder part of the Intention Estimation, and the result of the encoder is input to the third Trajectory prediction module. The third module is Trajectory prediction, and the other input is the pedestrian box coordinates of pedestrians. Based on the prediction speed sequence of the ego-vehicle, the output is the predicted sequence of pedestrian trajectories \(\{l^{t+1}, l^{t+2}, \ldots, l^{t+r}\}\).

4. Experiments
The structure of the model and the format of input and output data have been described in detail in the previous section, which will not be repeated here.

The part that needs to be introduced in detail in the model is intention estimation module. The intention estimation module consists of three parts: encoder-1, encoder-2 and decoder. The encoder-1 implements the function of HT-STGCN. The encoder-2 implements the function of CNN+Conv-LSTM. The decoder implements the function of LSTM, and outputs the result of intent estimation.

The implementation of the encoder-1 is similar to the structure described in literature [13]. The input of encoder-1 of intention estimation module is the pedestrian joint point coordinate. The input data is preprocessed as follows. Adjust the resolution of the input video to \(340 \times 256\). Frame rate adjusted to 30 FPS. The coordinates of the 18 joints of each pedestrian were estimated using the OpenPose toolbox [10]. The HT-STGCN model consists of 7 layers of spatial temporal graph convolution operators. In these 7 layers, the number of channels used for output in each layer is 64, 64, 128, 128, 128, 256, 256, respectively. Temporal kernel size is set to 9 for these layers, and dropout is 0.6. Based on the results of Ref. [13], we first pre-train the HT-STGCN model in the aspect of action classification on the PIE dataset [24]. The implementation of the encoder-2 is similar to the structure described in Ref. [24]. The encoder-2 contains 64 filters of Convolutional LSTM. Its kernel size is \(2 \times 2\), and the stride is 1. The input of encoder-2 of intention estimation module is pedestrian centric and contains a series of objects in the surrounding area. Therefore, the input image is cropped to twice the size of the pedestrian’s bounding box \((Obj_{z_{bbox}})\), squared and adjusted to \(224 \times 224\). Pedestrian intention results are shown in table 1. Benefit from skeleton sequence information, the new method HT-STGCN achieved better performance.

In table 1, \(Obj_{z_{bbox}}\) stands for twice the size of the pedestrian’s bounding box. \(Bbox_{loc}\) represents bounding box coordinates of pedestrian. \(Skel\) represents skeleton coordinates of pedestrian. \(PIE_{int}\) stands for pedestrian intention estimation of PIE model [24]. \(HT-STGCN\) model represents our novel method. Our model HT-STGCN can well implement pedestrian intention estimation and trajectory prediction, as is shown in figure 5.

| Model   | Input data                                       | acc  | F1  |
|---------|-------------------------------------------------|------|-----|
| PIE_{int} | \(Obj_{z_{bbox}} + Bbox_{loc}\)               | 0.79 | 0.87|
| \(HT-STGCN\) | \(Obj_{z_{bbox}} + Bbox_{loc} + Skel\)        | 0.81 | 0.89|

Table 1. Pedestrian intention estimation results.
Figure 5. Pedestrian intent estimation and real-time display.

The horizontal bar following the top of the pedestrian’s head indicates the intention of the pedestrian. Red bars indicate that pedestrians are about to cross or are crossing the road. Green bars indicate that pedestrians have no intention of crossing the road. The white translucent ball around the pedestrian’s waist indicates the direction of the pedestrian’s trajectory. If the white ball is to the right of the pedestrian, then it indicates that the pedestrian is moving to the right, and vice versa.

Figure 6 shows the pedestrian intention estimation and trajectory prediction in four scenarios.

Figure 6. Pedestrian intention estimation and trajectory prediction in four scenarios.

Table 2 shows the performance comparison between our method $HT$-$STGCN$ and PIE method [24] in trajectory prediction. In table 2, location, intention and vehicle speed are provided by the PIE dataset. $PIE_{intention}$ stands for pedestrian intention estimation of PIE model. $PIE_{speed}$ stands for speed of ego-vehicle predicted by the PIE model. $HT_{intention}$ stands for pedestrian intention estimation predicted by our HT-STGCN model. Obviously, the performance of pedestrian intent estimation has been significantly improved based on the graph network and skeleton method. Furthermore, the effect of pedestrian trajectory prediction is improved.

Table 2. Trajectory prediction performance comparison.

| Model       | Input data                 | MSE  |
|-------------|----------------------------|------|
| $PIE_{pred}$| $Location + PIE_{intention} + PIE_{speed}$ | 559  |
| $HT$-$STGCN$| $Location + HT_{intention} + PIE_{speed}$ | 551  |

5. Conclusion

On the PIE dataset, we studied pedestrian intent estimation, performed trajectory prediction based on pedestrian intent, and improved the latest pedestrian trajectory prediction algorithms. We propose a graph convolutional network algorithm based on the head-torso partitioning strategy. The pedestrian body structure is refined into joint points in the form of graphs in non-Euclidean space, which overcomes the limitation of Euclidean spatial image as input. Our new model is superior in performance to the latest
state-of-the-art technology, and when combined with ego-vehicle speed estimation, can effectively improve pedestrian intent estimation and trajectory prediction.

Acknowledgments
Many thanks to the anonymous IEEE IV reviewers who provided tremendously helpful comments. The research leading to these results has received funding from the Equipment pre-research project (No. 31511060502), and the basic and frontier technology research project of Henan Province (152300410103). This work was supported by the National Natural Science Foundation of China (11804081).

References
[1] Fang Z and López A M 2018 Is the Pedestrian going to Cross? Answering by 2D Pose Estimation in IEEE Intelligent Vehicles Symposium (IEEE IV 2018) pp 1271-1276.
[2] Rasouli A, Kotseruba I and Tsotsos J K 2017 Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior ICCV pp 206-213.
[3] Fang Z and López A M 2019 Intention recognition of pedestrians and cyclists by 2D pose estimation arXiv: 1910.03858 pp 1-10.
[4] Kreiss S, Bertoni L and Alahi A 2019 Pifpaf-composite fields for human pose estimation CVPR pp 11977-11986.
[5] Christoph F, Axel P and Andrew Z 2016 Convolutional two-stream network fusion for video action recognition CVPR pp 1933-1941.
[6] Ali D, Ali P and Gool L V 2016 Efficient two-stream motion and appearance 3D CNNs for video classification ECCV pp 1-4.
[7] Zhu Y, Lan Z, Newsam S and Hauptmann A 2018 Hidden two-stream convolutional networks for action recognition ACCV pp 363-378.
[8] Sun D, Yang X, Liu M-Y and Kautz J 2018 PWC-Net: CNNs for optical flow using pyramid warping and cost volume CVPR pp 8934-8943.
[9] Hara K, Kataoka H and Satoh Y 2018 Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet CVPR pp 6546-6555.
[10] Cao Z, Simon T, Wei S-E and Sheikh Y 2017 Realtime multi-person 2D pose estimation using part affinity fields CVPR pp 7291-7299.
[11] Shotton J, Sharp T, Kipman A, Fitzgibbon A, Finocchio M, Blake A, Cook M and Moore R 2011 Real-time human pose recognition in parts from single depth images CVPR pp 1297-1304.
[12] Du Y, Wang W and Wang L 2015 Hierarchical recurrent neural network for skeleton based action recognition CVPR pp 1110-1118.
[13] Yan S, Xiong Y and Lin D 2018 Spatial temporal graph convolutional networks for skeleton-based action recognition AAAI Page(s): 7444-7452.
[14] Shahroudy A, Liu J, Ng T-T and Wang G 2016 Ntu rgb+ d-A large scale dataset for 3d human activity analysis CVPR Page(s): 1010 - 1019
[15] Li M, Chen S, Chen X, Zhang Y, Wang Y and Tian Q 2019 Actional-Structural graph convolutional networks for skeleton-based action recognition CVPR pp 3595-3603.
[16] Kay W, Carreira J and Simonyan K 2017 The kinetics human action video dataset arXiv: 1705.06950 pp 1-22.
[17] Shi L, Zhang Y, Cheng J and Lu H 2019 Two-stream adaptive graph convolutional networks for skeleton-based action recognition CVPR pp 12026-12035.
[18] Papadopoulos K, Ghorbel E, Aouada D and Ottersten B 2019 Vertex feature encoding and hierarchical temporal modeling in a spatial-temporal graph convolutional network for action recognition arXiv: 1912.09745 pp 1-6.
[19] Liu J, Shahroudy A, Xu D and Wang G 2016 Spatio-temporal LSTM with trust gates for 3d human action recognition ECCV pp 1-18.
[20] Zhang S, Liu X and Xiao J 2017 On geometric features for skeleton-based action recognition using multilayer LSTM networks WACV pp 148-157.

[21] Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H and Wei Y 2017 Deformable convolutional networks arXiv:1703.06211 pp 1-12.

[22] Niepert M, Ahmed M and Kutzkov K 2016 Learning convolutional neural networks for graphs ICML pp 2014-2023.

[23] Kipf T N and Welling M 2017 Semi-supervised classification with graph convolutional networks ICLR pp 1-14.

[24] Rasouli A, Kotseruba I, Kunic T and Tsotsos J K 2019 PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction ICCV pp 6262-6271.