Bi-Prediction: Pedestrian Trajectory Prediction Based on Bidirectional LSTM Classification

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

***Abstract*—Pedestrian trajectory prediction is important in various applications such as driverless vehicles, social robots, intelligent tracking systems and space planning. Existing methods focus on analysing the influence of neighbours but ignore the effect of the intended destinations of pedestrians which also plays a key role in route planning. In this paper, we propose a novel two-stage trajectory prediction method to yield multiple prediction trajectories with different probabilities towards different destination regions in the scene. Our method, which we refer to as *Bi-Prediction*, uses a bidirectional LSTM architecture to automatically classify trajectories into a small number of route classes before trajectory prediction. We have evaluated our method against two baseline methods and three state-of-art methods on two benchmark datasets. Our experimental results show that the extra classification stage improves the accuracy of the predicted trajectories.**

I. Introduction

Although artificial intelligence research has made significant advances in the last decade, there are still challenges remaining in mining a large volume of video data of crowded scenes to automatically analyze and predict the trajectories of people. These challenges are mainly due to the complex movement patterns of pedestrians and the clutters in the scene (e.g., people have to detour slightly to avoid obstacles). An example scenario is the New York Ground Central station shown in Figure 1. As humans, we can predict the location of any particular individual in the next few frames using information such as the direction that their head is facing and walking speed. However, to automatically do so for many pedestrians is not an easy task. Training computers to give them the ability to predict trajectories through the mining of this large volume of data would be useful for space planning, scheduling of public transport, social robots, and intelligent tracking systems.

Fig. 1. Example of a crowded scene.

Existing research work on pedestrian trajectory prediction can be divided into model based and Long Short Term Memory (LSTM) based methods. The traditional model based methods use manually designed energy functions and specific settings of pedestrian properties rather than learning from trajectory data, so they often fail to predict trajectories in complicated crowded scenes [1, 2]. Recently, motivated by the successful LSTM applications in sequence data processing, trajectory prediction based on the LSTM architecture has attracted much attention [3–5]. For instance, some LSTM based pedestrian trajectory prediction methods [6, 7] use one LSTM for each pedestrian and incorporate factors due to the influence of neighbouring pedestrians. While this one-LSTM-one-pedestrian strategy is intuitive and simple, it is very computationally expensive. Besides, the layout of the scene, which plays an important part in trajectory prediction, has mostly been ignored. Ideally, with multiple entry and exit points present in most scenes, a trajectory prediction method should predict multiple possible trajectories heading toward different destinations and each predicted trajectory should have a probability measure indicating how likely the trajectory would be taken.

In this paper, we propose a novel two-stage trajectory prediction method to overcome the above shortcomings. After partitioning the scene into several regions, the first stage of our method is using bidirectional LSTM classification to predict a pedestrian's possible destination regions, with a short observed trajectory as input. The second stage is choosing differently trained LSTMs to predict the trajectory for each destination region. One advantage of our approach is that we explicitly address the problem of how to output multiple possible prediction results and the associated probabilities. Our proposed method is named *Bi-Prediction*, which has two meanings: using bidirectional LSTM in trajectory prediction and having two stages in the whole process.

The contributions of this paper are: 1) introducing a bidirectional LSTM architecture for predicting trajectories for the first time; 2) proposing a trajectory prediction system that can yield multiple prediction results for different destination regions. The rest of paper is organized as follows. Related work is presented in Section 2. Section 3 describes our proposed two-

978-1-5386-2839-3/17/$31.00 ©2017 IEEE

stage trajectory prediction method in detail. Section 4 analyzes our experiment results, and Section 5 concludes the paper and outlines our future work.

II. Related Work

Research work on pedestrian trajectory prediction in the literature can be broadly divided into two categories: model based and LSTM based methods. Helbing and Molnar [1] presented a social force model to describe the motion of pedestrians. There are three key forces in their social force model: acceleration towards the desired velocity of motion, repulsive effect, and attractive effect. Yamaguchi et al. [2] improved the original social force model by exploiting behaviours, such as damping and collision, in social interactions and incorporating these properties to the pedestrian behavioural prediction problem. Similar to the social force model, agent-based modelling [8] has been used to simulate behavioural patterns of individuals also. Antonini et al. [9] utilized strong priors in a discrete choice framework to model human interaction behaviour. In general, model based pedestrian trajectory prediction methods rely on some hand-crafted factors like the preferred walking speeds of pedestrians.

Given that traditional neural networks and deep learning architecture such as multilayer perceptron cannot fully handle time sequence information in the data, Recurrent Neural Networks (RNN) has been introduced to deal with time sequence data. However, simple RNNs have difficulties in remembering long-term input information because of the gradient vanishing or gradient exploding issues [10]. Long Short Term Memory (LSTM) [11] and Gated Recurrent Units (GRU) [12] have therefore been designed to deal with data that have long-term dependencies. Recently, RNN and its variants including LSTM and GRU have demonstrated their successes in time sequence modelling tasks such as speech recognition [13], language translation [14, 15], image captioning [16] and so on. A convolutional LSTM has also been used in the precipitation nowscasting problem [17].

The trajectories of pedestrians can be considered as time sequence data, so RNN models can be used for predicting trajectories of pedestrians. Alahi et al. [6] suggested a social LSTM model which combines the behaviour of other people within a large neighbourhood. They used one LSTM model for every person in crowded scenes and they introduced a social pooling strategy to connect different LSTM models to describe the interaction of people in the same neighbourhood. However, this social-LSTM trajectory prediction approach considers only neighbours' influence while ignoring some other factors like the layout of the scene. Also, one LSTM for each pedestrian strategy is complicated in crowded scenarios. Recently, based on social-LSTM, Fernando et al. [7] proposed an attention based LSTM model for prediction and abnormal detection. Although they took into account the scene context by learning separate models for different entry/exit zones, they clustered their trajectories based on the entire trajectory length in the training phase. This is different from our approach proposed in the paper in that our algorithm can deal with short observed trajectories in the prediction phase and, based on these short trajectories, can learn separate trajectory prediction models.

Compared with the traditional one direction LSTM architecture which considers only the past information, the bidirectional LSTM architecture can take full use of both the past and the future context in a data sequence. Bidirectional LSTMs have recently attracted special attention in solving sequence labelling problems in the area of speech recognition [18, 19], translation [20], and handwriting recognition [21]. However, to the best of our knowledge, bidirectional LSTMs have not been used for trajectory prediction.

III. Proposed Model

A. System Overview

At time instant t, the i^{th} pedestrian in the scene is represented by the image coordinates $\left(x_t^i, y_t^i\right)$. We observe the positions of all the pedestrians from time $t = T_1$ to $t = T_{\text{obs}}$ and our aim is to predict their positions from $t = T_{\text{obs}} + 1$ to $t = T_{\text{pred}}$. The problem of trajectory prediction can therefore be defined as a sequence generation problem:

Given: Observed trajectories $\mathbf{X}_i^{\text{obs}} = \left[\left(x_1^i, y_1^i\right), \cdots, \left(x_{\text{obs}}^i, y_{\text{obs}}^i\right)\right], \forall i$

Objective: Predict future trajectories $\mathbf{X}_i^{\text{pred}} = \left[\left(x_{\text{obs}+1}^i, y_{\text{obs}+1}^i\right), \cdots, \left(x_{\text{pred}}^i, y_{\text{pred}}^i\right)\right], \forall i$

In real applications, after walking into a shopping square or a train terminal, pedestrians would choose their routes mostly based on their intended destinations. So trajectory prediction will have much higher accuracy if the intended destinations can be learned from the trajectory data. Furthermore, to model the complex movement patterns of pedestrians, rather than a single predicted trajectory for each pedestrian, being able to generate multiple possible trajectories for each pedestrian would be more useful for further analysis such as anomaly detection.

Unlike existing trajectory prediction methods that generate a single predicted trajectory, our proposed two-stage Bi-Prediction method can learn the potential destinations in a scene and can generate multiple trajectory predictions. The problem of trajectory prediction can be reformulated to one where, based on the observed trajectory $\mathbf{X}_i^{\text{obs}}$, different sequences $\mathbf{X}_{i,j}^{\text{pred}}$ corresponding to a small number of potential destination candidates D_j's need to be generated. In our model, the aim of the first stage is to predict the destination candidates D_j and the probability of choosing D_j. After acquiring these values, candidates D_j, the second stage is to generate different sequences $\mathbf{X}_{i,j}^{\text{pred}}$ based on these candidate destinations D_j. Details of these two stages will be described in the subsections after a brief review on LSTM.

B. LSTM - A Brief Review

Consider a typical LSTM network architecture with the input sequence represented by $(x_1, \cdots, x_T)$ and the hidden state denoted by h. The output sequence y_t can be obtained

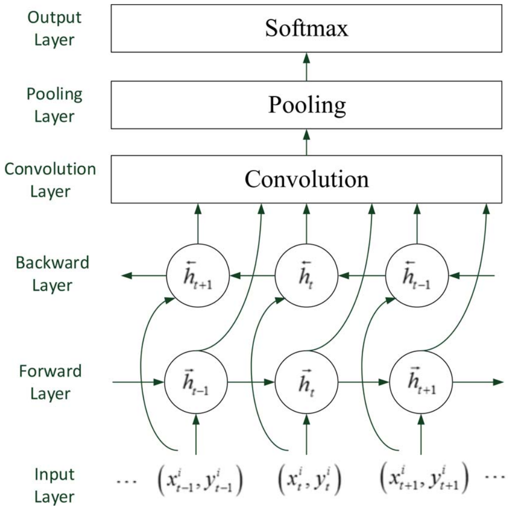


Fig. 2. Stage 1: Bidirectional LSTM network for trajectory classification.

by iteratively computing Eqs. (1) and (2) for $t = 1, \cdots, T$:

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

$$y_t = W_{hy} h_t + b_y, \tag{2}$$

where the W terms denote the different weight matrices and b denotes the bias vector. In each LSTM cell, the hidden state is determined by the input gate i, forget gate f, output gate o and the cell state c via the equations given below:

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

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

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

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

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

The $\sigma(.)$ function in Eqs. (3)-(7) is the sigmoid activation function.

C. Stage 1: Bidirectional LSTM Classification

An overview of the architecture of our system for stage 1 is shown in Figure 2. It is a 6-layer network consisting of an input layer, which takes in the trajectory sequences, and a softmax output layer. The 4 hidden layers are: a forward and a backward LSTM layer, convolution, and pooling.

To deal with the complex movement patterns of pedestrians, the scene is first partitioned into regions. Taking the observed trajectory in Figure 3 as an example, if the person wants to go to region D_3 to buy tickets, his or her trajectory would be in the top half of the scene in most cases, which differs greatly from trajectories going into region D_1 or the region D_2. This indicates that destination regions should be determined first.

After scene partitioning, our Bi-Prediction method attempts to classify the pedestrians' trajectories into different *route classes*. We use $R_{i,j}$ to denote the route from source region D_i to destination region D_j. Specifically, because the network is bidirectional, $R_{i,j}$ and $R_{j,i}$ belong to the same route class. For a scene that is partitioned into N regions, the total number of route classes is then $\binom{N}{2} + N = N(N+1)/2$. In the two datasets that we analyze in this paper, the routes $R_{i,i}, \forall i$ (where the source and destination regions are the same) are ignored due to the insufficiency of data for classification. This reduces the total number of route classes to $N(N-1)/2$. We adopt the standard one-hot encoding dictionary to label the route classes. For the 4 regions in Figure 3, the 6 route classes (ignoring $R_{i,i}$'s) are shown in Table I.

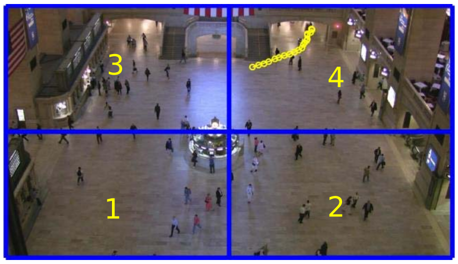


Fig. 3. A scene with 4 partitioned regions and a pedestrian's observed trajectory superimposed.

TABLE I
ONE-HOT ENCODING DICTIONARY FOR A SCENE THAT IS PARTITIONED INTO 4 REGIONS (FOR EXAMPLE, AS IN FIGURE 2).

Class No.	Route	Label
1	$R_{1,2}$ and $R_{2,1}$	[0 0 0 0 0 1]
2	$R_{1,4}$ and $R_{4,1}$	[0 0 0 0 1 0]
3	$R_{1,3}$ and $R_{3,1}$	[0 0 0 1 0 0]
4	$R_{2,4}$ and $R_{4,2}$	[0 0 1 0 0 0]
5	$R_{2,3}$ and $R_{3,2}$	[0 1 0 0 0 0]
6	$R_{3,4}$ and $R_{4,3}$	[1 0 0 0 0 0]

It should be noted that the classification problem in this stage is different from the traditional classification problems as our overall objective is to achieve "foreseeing" power for the trajectory prediction stage later on. Given an observed trajectory in the training process, our proposed method will group trajectories into different classes depending on the routes taken by the pedestrians. It is important to note that the route is not well-defined if the trajectory is very short. For example, only based on the yellow trajectory in Figure 3, we cannot tell whether the destination would be in region D_1 or region D_3, or maybe this person suddenly changes direction and goes to region D_2. One way to overcome this is to design a special network (as in Figure 2) that combines a recurrent neural network with a convolution neural network whereby the former develops the "foreseeing" power and the latter takes charge of classification task to output multiple probabilities of different destination regions.

Unlike the traditional one direction LSTM networks which only learn from previous information, the bidirectional LSTM in Figure 2 can make use of previous information and future information by processing the data bidirectionally through the forward layer and backward layer. For the bidirectional LSTM network architecture, Eqs. (1)-(2) become:

$$\overrightarrow{h}_t = \mathrm{LSTM}\left(x_t, \overrightarrow{h}_{t-1}; \overrightarrow{W}\right) \tag{8}$$

$$\overleftarrow{h}_t = \mathrm{LSTM}\left(x_t, \overleftarrow{h}_{t-1}; \overleftarrow{W}\right) \tag{9}$$

$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y, \tag{10}$$

where, as before, the W terms denote different weight matrices. Specifically, $\overrightarrow{h}_t$, $\overrightarrow{W}$, $\overleftarrow{h}_t$, and $\overleftarrow{W}$ are the hidden states and weight matrices of the forward and backward layers respectively. In our observed trajectory classification case, at $t \leq T_{\text{obs}}$, the Bi-Prediction method can acquire and process both the previous information $\mathbf{X}_{i,t}^{\text{previous}} = \left[\left(x_1^i, y_1^i\right), \cdots, \left(x_t^i, y_t^i\right)\right]$ and the future information $\mathbf{X}_{i,t}^{\text{future}} = \left[\left(x_{t+1}^i, y_{t+1}^i\right), \cdots, \left(x_{T_{\text{obs}}}^i, y_{T_{\text{obs}}}^i\right)\right]$ from the observed trajectory $\mathbf{X}_i^{\text{obs}}$ at the same time. Note that $\mathbf{X}_{i,t}^{\text{previous}}$ is different from $\mathbf{X}_{i,t}^{\text{pred}}$ at the prediction stage. The former contains information up to time t but the latter contains unknown information (to be predicted) from the time T_{obs} onward. In fact, we have $\mathbf{X}_{i,t}^{\text{previous}} \oplus \mathbf{X}_{i,t}^{\text{future}} = \mathbf{X}_i^{T_{\text{obs}}}$, where $\oplus$ represents the concatenation operator.

Inspired by many successes on using convolution layers in Convolution Neural Networks (CNN) for image/object classification, the output from the forward and backward layers are passed to the convolution layer in our proposed network architecture. The convolution layer and pooling layer handle the learning of trajectory coordinates for classification. The softmax layer is used as the output layer so that the outputs of the network are the probabilities of the route classes. For example, for the input observed trajectory given in Figure 3, the output should be the probabilities of $R_{4,1}$, $R_{4,2}$, and $R_{4,3}$. Thus, by using our bidirectional LSTM classification network, we can predict potential destination region candidates based on a short observed trajectory.

D. Stage 2: Prediction with Classification

The second stage of our method is to predict future trajectories corresponding to different route classes. When it comes to future trajectory prediction, existing LSTM based methods which use the one-LSTM-one-pedestrian policy would suffice. However, this would make both the training process and the predicting process very computationally expensive. On the other hand, if we use only one LSTM for predicting the trajectories of all pedestrians, the network would not be able to learn all kinds of pedestrian movement patterns. As a tradeoff, in stage 2, our method uses one sub-LSTM for each route class. Taking Figure 3 again as an example, our method trains 6 sub-LSTMs for the 6 route classes shown in Table I for trajectory prediction.

Figure 4 shows the steps involved in stage 2 of our method. After stage 1, the system has obtained the probabilities for the route classes for each input observed trajectory. If the probability of a route class is larger than a pre-defined threshold τ, the system automatically selects the corresponding trained sub-LSTMs to output the predicted trajectories for each pedestrian.

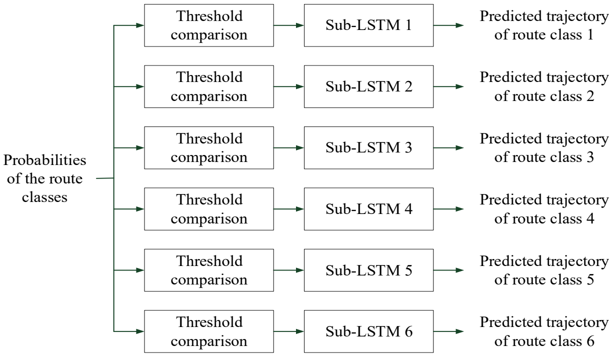

Probabilities of the route classes
Threshold comparison
Sub-LSTM 1
Predicted trajectory of route class 1
Threshold comparison
Sub-LSTM 2
Predicted trajectory of route class 2
Threshold comparison
Sub-LSTM 3
Predicted trajectory of route class 3
Threshold comparison
Sub-LSTM 4
Predicted trajectory of route class 4
Threshold comparison
Sub-LSTM 5
Predicted trajectory of route class 5
Threshold comparison
Sub-LSTM 6
Predicted trajectory of route class 6

Fig. 4. Stage 2: Future trajectory prediction based on the classification from stage 1.

As for each sub-LSTM, we use a general encoder-decoder network where the LSTM encoder layer receives an observed trajectory as input and generates a hidden state sequence. From the hidden state sequence, the LSTM decoder layer can output a predicted trajectory. The input size and output size of the sub-LSTMs are determined by the length of the observed trajectories and predicting trajectories respectively. Each sub-LSTM is trained separately with the training trajectory data in each route class. Moreover, the ReLU (rectified liner units) non-linearity activation function is used in the LSTM layers of each sub-LSTM. Through this novel two-stage trajectory prediction network, the Bi-prediction method can predict pedestrian trajectories heading towards different destination regions and predict multiple possible trajectories for each given observed trajectory. This is equivalent to yielding the top *k* results in image/object classification problems.

E. Implementation Details

In the current implementation of the Bi-prediction method, the region partitioning in stage 1 is manually performed. For the bidirectional LSTM network in the first stage, the forward and backward LSTM layers have a fixed hidden state dimension of 128. We use a one dimension convolution layer followed by a one dimension max pooling layer. The size of the max pooling windows is 4, which means that the pooling layer will extract the max value in every 4 output values from the convolution layer. As for each sub-LSTM in stage 2, the hidden state is also 128-dimensional.

Our two-stage trajectory prediction model is built using Python on Keras with a Tensorflow backend. The parameters of our proposed network are trained with a *RMSprop* optimizer (see https://keras.io/optimizers) and the learning rate is set to 0.003. In all of our experiments, we follow the same setting as [7] and use 20-frame observed trajectories to train each sub-LSTM and predict trajectories of 20 frames long, i.e., $T_{\text{obs}} = 20$ and $T_{\text{pred}} = 40$. The pre-defined threshold τ is set to 0.01.

IV. Experiments

A. Datasets and Metrics

We present our experiments on two publicly available crowded scenes datasets with labelled pedestrian trajectories: the New York Grand Central [22] and the Edinburgh Informatics Forum [23] datasets. The New York Grand Central dataset contains over 12,000 accurate pedestrian walking routes annotated as ground truth from an one-hour crowd video. The Edinburgh Informatics Forum dataset consists of a set of detected people walking through the Informatics Forum in the main building of the School of Informatics at the University of Edinburgh. The data covers several months of observations which result in about 1,000 observed trajectories on each working day and more than 92,000 observed trajectories in total. The frame rates of the original surveillance video of the NYGC and the Edinburgh datasets are 23 frames per second and 9 frames per second respectively. The coordinates of each trajectory in the NYGC dataset are annotated every 20 frames from the original video while trajectories in Edinburgh dataset are tracked at every frame. In the experiments, our bi-prediction method automatically filters out some short and fragmented trajectories. For both datasets, we take the same region partitioning scheme as shown in Figure 3.

Let n be the number of predicted trajectories, $\mathbf{X}_{i,t}^{\text{pred}}$ be the predicted position of the trajectory of i^{th} pedestrian at time instant t, and $X_{i,t}^{\text{gt}}$ be the corresponding ground truth position. Two widely used evaluation metrics for evaluating the performance of pedestrian trajectory prediction are the average displacement error (ADE) and the final displacement error (FDE).

The ADE is the mean squared error over all the estimated points and the ground truth points of all trajectories. This error can be calculated based on the Euclidean distance between every predicted location and actual location in each pedestrian trajectory:

$$\text{ADE} = \frac{\sum_{i=1}^{n} \sum_{t=T_{\text{obs}}+1}^{T_{\text{pred}}} \left\| \mathbf{X}_{i,t}^{\text{pred}} - \mathbf{X}_{i,t}^{\text{obs}} \right\|}{n\left(T_{\text{pred}} - (T_{\text{obs}}+1)\right)}. \tag{11}$$

Similar to the ADE, the final displacement error (FDE) is the distance between the final destinations of the predicted trajectories and those of the ground truth trajectories. Compared to the ADE, the FDE focuses on the prediction accuracy of pedestrian destinations. The FDE is defined as:

$$\text{FDE} = \frac{\sum_{i=1}^{n} \left\| \mathbf{X}_{i,T_{\text{pred}}}^{\text{pred}} - \mathbf{X}_{i,T_{\text{pred}}}^{\text{obs}} \right\|}{n}. \tag{12}$$

We use these two metrics to evaluate our method and other state-of-art techniques quantitatively.

B. Quantitative Results

The two baseline methods and 3 state-of-art pedestrian trajectory prediction methods with which we compare our method are:

TABLE II
Quantitative Results: Comparison of Different Trajectory Prediction Methods. The lowest and second lowest errors are highlighted in red and blue respectively.

Methods	Edinburgh		NYGC	
	ADE	FDE	ADE	FDE
Linear	0.803	1.526	3.218	6.989
LSTM	2.132	3.005	3.696	4.723
SF [2]	3.124*	3.909*	3.364*	5.808*
Social-LSTM [6]	1.524*	2.510*	1.990*	4.519*
Attention-LSTM [7]	0.986*	1.311*	1.096*	3.011*
Bi-Prediction-1	0.648	1.027	1.686	3.037
Bi-Prediction-3	0.599	0.931	1.205	2.026

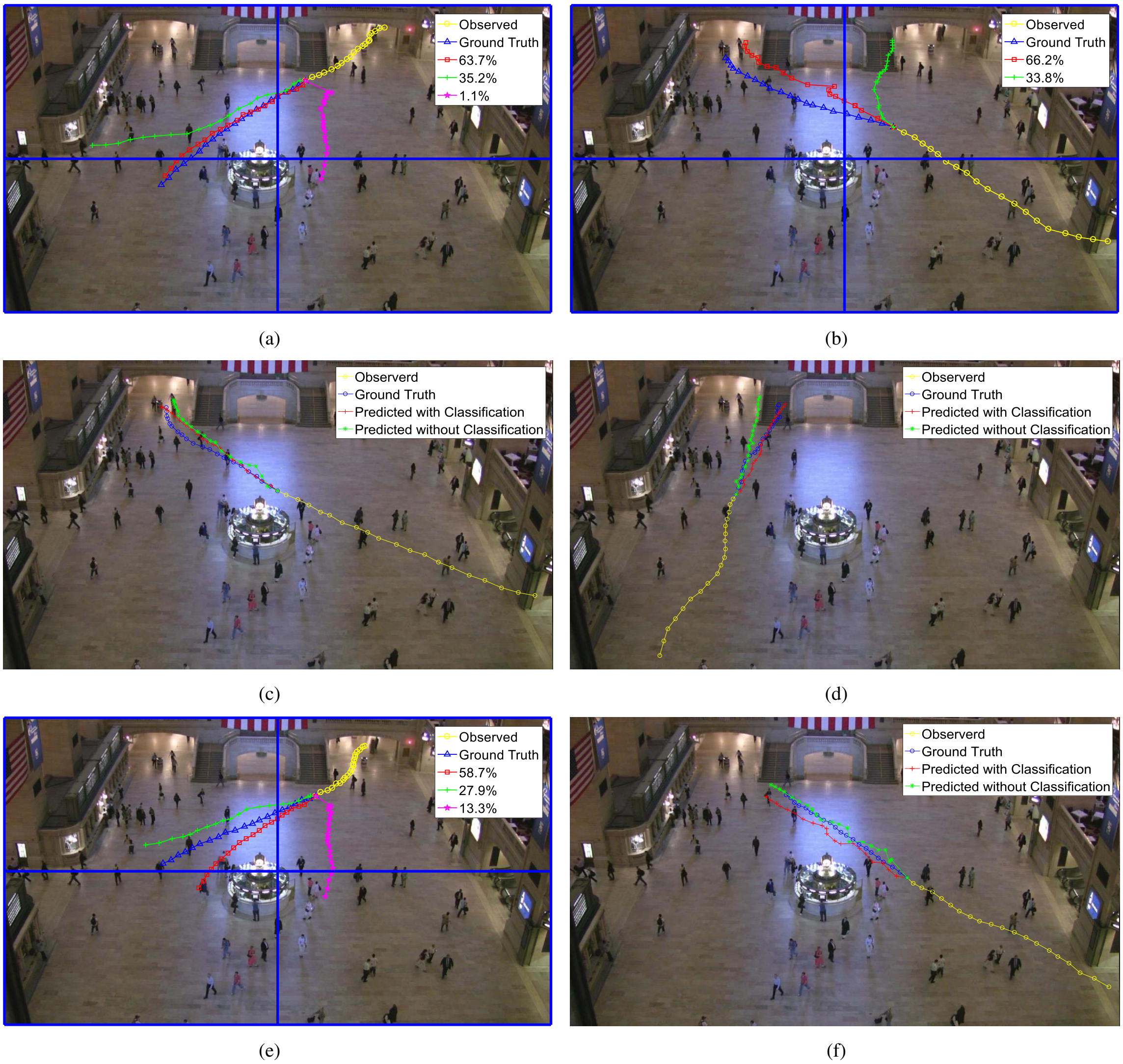
* results extracted from [7]

- *Linear Prediction*. We use a preliminary linear prediction method to predict plausible trajectories with the assumption that people move in the same direction and speed as in the observed trajectories. This method can be considered to be the baseline method for trajectory prediction.
- *LSTM*. This is the basic vanilla LSTM based trajectory prediction method which does not consider destination region classification. This is the baseline method for LSTM based trajectory prediction.
- *Social Force (SF)*. The Social Force model [2] uses factors such as preferred walking speed, destination and group affinity to model pedestrians' moving behaviours for trajectory prediction.
- *Social-LSTM*. The Social LSTM model proposed by Alahi et al. [6] combines a social pooling layer with the traditional LSTM network to model pedestrian-pedestrian interactions in the trajectory predicting process.
- *Attention-LSTM*. An attention-based mechanism is introduced to modify social-LSTM model by Fernando et al. [7], which considers the entire sequence of hidden states for both the predicted trajectory of each pedestrian and his/her neighbours.

For our Bi-Prediction method, we investigate two cases where $k = 1$ and $k = 3$:

- *Our Bi-Prediction-1*. This is our proposed Bi-prediction method which gives the best predicted trajectories based on the highest classification probability.
- *Our Bi-Prediction-3*. This is like the top k results (where $k = 3$) used in image classification task. Our two-stage prediction model would predict k trajectories based on the top k probabilities given by the bidirectional LSTM classification network in stage 1.

In Table II, we report the quantitative performance in terms of ADE and FDE using metres as the unit. The best results are shown in red and the second best results are in blue. Results of social force, social-LSTM and attention-LSTM are extracted from Fernando et al.'s paper [7]. In general, all prediction

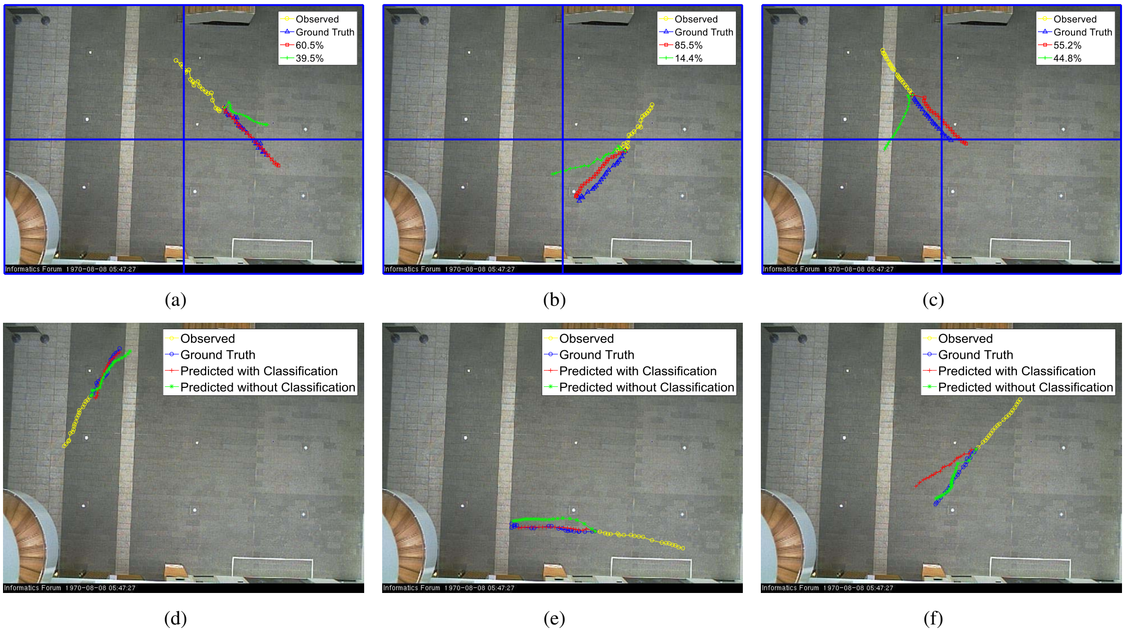

Observed
Ground Truth
63.7%
35.2%
1.1%
Observed
Ground Truth
66.2%
33.8%
Observerd
Ground Truth
Predicted with Classification
Predicted without Classification
Observerd
Ground Truth
Predicted with Classification
Predicted without Classification
Observed
Ground Truth
58.7%
27.9%
13.3%
Observerd
Ground Truth
Predicted with Classification
Predicted without Classification

(a) (b) (c) (d) (e) (f)

Fig. 5. Illustration of a few predicted trajectories from our Bi-prediction method on the NYGC dataset. The observed trajectories and ground truth are given in yellow and blue respectively. The first row shows the multiple predicted trajectories output by our method in red (highest probability), green (second probability), and magenta (third probability) colours. The second row shows a comparison between the predicted trajectories with and without classification in red and green. The last row presents some slightly worse predicted trajectories.

methods perform better on the Edinburgh dataset. The main reason for this is that the predicted trajectories for the NYGC dataset are much longer than those in the Edinburgh dataset. Considering the frame rates (23 fps for NYGC versus 9 fps for Edinburgh) and the annotation rate of trajectory coordinates in the two datasets, the 20-frame observed trajectories from the NYGC dataset are about 10 times longer than those in the Edinburgh dataset. Consequently, all the methods also need to predict longer pedestrian trajectories in the NYGC dataset.

While other methods have similar performance in both datasets, the basic linear prediction method performs much better on the Edinburgh dataset than the NYGC dataset. This can be explained by the different complexities of the scenes as well. As shown in Figures 5 and 6, the background scene of NYGC is much more complicated than that of Edinburgh. For example, people have to bypass the information centre in the middle of the NYGC scene while people can move in a straight line without any obstacles in the Edinburgh scene. Because of the relative simple layout of the Edinburgh scene, even the vanilla LSTM method performs better than the Social Force model based method. For the Edinburgh dataset, both of our Bi-Prediction-1 and Bi-Prediction-3 methods give better prediction accuracy than the state-of-art methods. For the ADE measure for the NYGC dataset, our Bi-Prediction-1 method comes in the second place, performing slightly worse than Attention-LSTM but outperforming the other methods by a large margin. For the FDE measure for NYGC, our Bi-Prediction-3 method outperforms all other methods.



(a) (b) (c)

(d) (e) (f)

Fig. 6. Illustration of a few predicted trajectories from our Bi-prediction method on the Edinburgh dataset. The observed trajectories and ground truth are given in yellow and blue respectively. The first row shows the multiple predicted trajectories output by our method in red (highest probability) and green (second probability) colours. The second row shows a comparison between the predicted trajectories with and without classification in red and green.

The experiments on these two datasets show that using the potential destination regions to guide the prediction of trajectories can improve the prediction accuracy on both the ADE and FDE measures. While the Attention-LSTM method focuses on the influence of neighbours to help to predict trajectories of people walking in groups, in crowded scenes like train terminals, people walk closely together even though they do not belong to the same group; furthermore, their trajectories are governed more by their intended destinations rather than by other pedestrians nearby.

C. Qualitative Results

We illustrate our experimental results on the NYGC dataset in Figure 5. Parts (a) and (b) of the figure show the multiple predicted trajectories with different probabilities towards the different destination regions. As we train different sub-LSTM models for each route class, if the probability of one route class is larger than the threshold τ (set to 0.01 in all experiments), a predicted trajectory will be generated by the corresponding trained sub-LSTM network. We also compare our proposed model with the traditional (vanilla) LSTM trajectory prediction method which do not include classification. The results in Figure 5(c) and Figure 5(d) show that the introduced bidirectional LSTM classification improves the accuracy of prediction. For example, in Figure 5(d), the prediction method might give a wrong predicted destination without classification because it can be inferred that this pedestrian is going to the second floor through the left stairs in the scene. However, without classification, based on the observed trajectory one may assume that that this pedestrian would still be walking on the ground floor.

The slightly worse classification result from stage 1 can occasionally compromise the trajectory prediction process in stage 2. The last row of Figure 5 shows some trajectories that are not so accurately predicted. In part (e) of the figure, compared to the predicted trajectory with the second highest probability, the one with the highest probability is further away from the ground truth trajectory. This also explains Bi-Prediction-3 performs better than Bi-Prediction-1. Although the predicted trajectories with the highest probabilities may sometimes be slightly further away from the ground truth trajectories, our proposed method still yields good trajectories in the top-k prediction. Figure 5(f) shows a case where the traditional vanilla LSTM method (without classification) outperforms our Bi-Prediction method (with classification). However, on average, as shown in the results in Table II, our method outperforms vanilla LSTM by a significant margin on both the ADE and FDE measures in both datasets.

Figure 6 depicts some prediction results on the Edinburgh dataset. Similar to our experiments on the NYGC dataset, The first row shows the multiple predicted trajectories of our method. Although the predicted trajectories in Figure 6(c) with the highest probability is quite close to the ground truth, because of the small difference between the two probability values, this particular case might be considered to be equivocal. Two examples comparing the predictions from our method (with classification) and vanilla LSTM (without classification) are given in Figure 6(d) and Figure 6(e). Notice that we choose the predicted trajectory with the highest probability in this comparison. Figure 6(f) shows a slightly worse prediction result from our method (with classification). In this case, although we have predicted the right destination region, there are more than one exit in this region (two exits in the left bottom of the figure) which affects some of our prediction trajectory results in this region of the scene. Thus, a more refined region partitioning scheme (as part of our future work) is expected to further improve the prediction. These slightly worse cases are just a few odd cases. On average, our Bi-prediction method gives better performance (Table II).

As shown in the first row of Figure 6, there are only two predicted trajectories which have higher probabilities than the threshold τ in each case. However, in the NYGC dataset (Figure 5), there are three predicted trajectories in most cases. This is a sign of complex movement patterns of pedestrians when the scene is cluttered and/or has many entries/exits. Rather than having one or two dominating route probabilities, the total probability splits into several small probabilities, showing a diversity of possible destinations.

V. Conclusions and Future Work

We have presented a novel two-stage pedestrian trajectory prediction method for crowded scenes. We take advantage of the bidirectional LSTM architecture to classify potential destination regions, which allows the method to improve the accuracy of prediction. We have compared our method with two baseline methods and three state-of-art methods and demonstrated that our method outperforms these methods. In addition, our proposed prediction method can yield multiple prediction trajectories with different probabilities, which can be used for anomaly analysis and space planning. For instance, trajectories that have probabilities lower than a given threshold value can be automatically flagged as possible abnormal trajectories for further analysis. In the current implementation, the region partitioning in stage 1 is manually identified. Part of the future work of our trajectory prediction project is to automate region partitioning and to use training trajectory data to detect entry and exit points in the scene. We also intend to further improve the prediction accuracy by considering factors such as stationary talking groups and pedestrians stopping to buy tickets or ask for information.

Acknowledgment

Author Hao Xue is supported by an International Postgraduate Research Scholarship.

References

[1] D. Helbing and P. Molnar, "Social Force Model for Pedestrian Dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[2] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who Are You With and Where Are You Going?" in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1345–1352.

[3] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts," in *AAAI*, 2016, pp. 194–200.

[4] F. Wu, K. Fu, Y. Wang, Z. Xiao, and X. Fu, "A Spatial-Temporal-Semantic Neural Network Algorithm for Location Prediction on Moving Objects," *Algorithms*, vol. 10, no. 2, p. 37, 2017.

[5] X. Song, H. Kanasugi, and R. Shibasaki, "Deeptransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level," in *IJCAI*, 2016.

[6] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[7] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft+ Hardwired Attention: An LSTM Framework for Human Trajectory Prediction and Abnormal Event Detection," *arXiv preprint arXiv:1702.05552*, 2017.

[8] E. Bonabeau, "Agent-based Modeling: Methods and Techniques for Simulating Human Systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[9] G. Antonini, S. V. Martinez, M. Bierlaire, and J. P. Thiran, "Behavioral Priors for Detection and Tracking of Pedestrians in Video Sequences," *International Journal of Computer Vision*, vol. 69, no. 2, pp. 159–180, 2006.

[10] R. Pascanu, T. Mikolov, and Y. Bengio, "On the Difficulty of Training Recurrent Neural Networks," *ICML (3)*, vol. 28, pp. 1310–1318, 2013.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[13] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *ICML*, vol. 14, 2014, pp. 1764–1772.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv preprint arXiv:1409.0473*, 2014.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[16] A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.

[18] A. Graves, A. r. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6645–6649.

[19] A. Graves, N. Jaitly, and A. r. Mohamed, "Hybrid Speech Recognition with Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec 2013, pp. 273–278.

[20] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation Modeling with Bidirectional Recurrent Neural Networks," in *EMNLP*, 2014, pp. 14–25.

[21] P. Doetsch, A. Zeyer, and H. Ney, "Bidirectional Decoder Networks for Attention-Based End-to-End Offline Handwriting Recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 361–366.

[22] S. Yi, H. Li, and X. Wang, "Understanding Pedestrian Behaviors from Stationary Crowd Groups," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3488–3496.

[23] B. Majecka, "Statistical Models of Pedestrian Behaviour in the Forum," *Master's thesis, School of Informatics, University of Edinburgh*, 2009.