Path Predictions using Object Attributes and Semantic Environment

Hiroaki Minoura, Tsubasa Hirakawa, Takayoshi Yamashita and Hironobu Fujiyoshi
Chubu Univeysity, Kasugai, Aichi, Japan

Keywords: Convolutional Neural Network, Long Short-Term Memory, Path Prediction.

Abstract: Path prediction methods with deep learning architectures take into account the interaction of pedestrians and the features of the physical environment in the surrounding area. These methods, however, process all prediction targets as a unified category and it becomes difficult to predict a path suitable for each category. In real scenes, it is necessary to consider not only pedestrians but also automobiles and bicycles. It is considered possible to predict the path corresponding to the type of target by considering the types of multiple targets. Therefore, aiming to achieve path prediction in accordance with individual categories, we propose a path prediction method that represents the target type as an attribute and simultaneously considers the physical environment information. The proposed method inputs feature vectors in a long short-term memory that represents i) past object trajectory, ii) the attribute, and iii) the semantics of the surrounding area. This makes it possible to predict a path that is proper for each target. Experimental results show that our approach can predict a path with higher precision. Also, changes in accuracy were analyzed by introducing the attribute of the prediction target and the physical environment information.

1 INTRODUCTION

Path prediction, one of the challenging tasks in the field of computer vision, estimates how a target object like a pedestrian or an automobile will move and on what path. Path prediction is expected to have a wide range of applications, such as preventing car accidents (SchneiderNicolas and M., 2013)(Keller and Gavrila, 2014)(Kooij et al., 2014) or autonomously controlling robots (Ziebart et al., 2009)(Karasev et al., 2016)(Vemula et al., 2017)(A. Vemula and OhSocial, 2017). Therefore, it has received much attention and various prediction methods have already been proposed (Rehder and Kloeden, 2015)(Huang et al., 2016)(Xie et al., 2013)(Walker et al., 2014)(Park et al., 2016)(Su et al., 2017). In recent years, because of advancements in deep leaning, prediction methods utilizing a convolutional neural network (CNN) (Lecun et al., 1989) or a long short-term memory (LSTM) (S.Hochreiter, 1997) have also been developed (A. Vemula and OhSocial, 2017)(Yi et al., 2016)(Alahi et al., 2016)(Lee et al., 2017)(Fernando et al., 2017b)(Fernando et al., 2017a)(Gupta et al., 2018). To predict paths accurately, several factors are introduced. For instance, the interactions between pedestrians (Alahi et al., 2016)(Lee et al., 2017)(Helbing and Molnar, 1995)(Yamaguchi et al., 2011)(Robicquet et al., 2016)(Ma et al., 2017) are modeled to predict and avoid collisions. Scene semantics are also introduced for reliable prediction (Lee et al., 2017)(Kitani et al., 2012)(Ballan et al., 2016). However, these approaches have a problem that all target objects are considered to be in the same class. In practical scenes, it is necessary to predict the path of a target object in an environment where there are a variety of prediction targets, not only pedestrians but also cars and bicycles. This means that the speed, traveling distance, and area may differ depending on the type of target object. If we simultaneously predict the paths of multiple target objects, it would be difficult to predict them in accordance with the type of target. Although a naive solution for this problem is creating models for each object type and making predictions accordingly, it would be impractical.

In this paper, we propose a method to simultaneously predict paths of different types of target objects such as pedestrians and bicycles (see Figure 1). Specifically, our method leverages three pieces of information: the type of target object, the physical environment surrounding the target, and a past object trajectory. We define the target object type (i.e., pedestrian, bicycle) as an attribute and represent it as a one-hot vector. For the physical environment, a feature vector is extracted from semantic scene labels (e.g., pavement, grass, and building) via convolutional layers. The past object trajectories correspond to coordinates

Minoura, H., Hirakawa, T., Yamashita, T. and Fujiyoshi, H.
Path Predictions using Object Attributes and Semantic Environment.
DOI: 10.5220/0007297500190026
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 19-26
ISBN: 978-989-758-354-4

at each time step. We obtain a coordinate of the next time step from the output of the network by inputting these vectors of current time step into an LSTM. At the time of prediction, we can make a prediction that takes the past object trajectory into account by sequentially inputting the network output to the input of the next time step. Simultaneously introducing the target attribute and semantic label enables us to predict a path considering the difference in the speed of each target and the area where the target tends to move preferably. Also, we use a relative coordinate, that is, direction and magnitude obtained from the difference between two successive coordinates. Introducing relative coordinates prevents the prediction results from depending on the trained scene and enables us to predict paths over multiple different scenes.

We have two contributions. i) To the best of our knowledge, this is the first attempt to predict paths of different kinds of prediction targets with a unified framework. ii) We contribute a scene label dataset that is annotated for the path prediction dataset published by Robicquet et al. (Robicquet et al., 2016).

2 RELATED WORK

Over the last decade, several approaches have been proposed to solve the path prediction problem. One classical approach is a method based on Bayesian models (SchneiderNicolas and M., 2013)(Kooij et al., 2014)(Ballan et al., 2016). Schneider *et al.* (SchneiderNicolas and M., 2013) proposed a path prediction method based on an extended Kalman filter to predict the walking path of a pedestrian captured by an onboard camera. Kooij *et al.* (Kooij et al., 2014) predicted the movement of pedestrians crossing a pavement using a Dynamic Bayesian Network (DBN)(Robinson and Hartemink, 2009). They use the pedestrian's head direction, the distance between the pedestrian and a car, and the distance the pedestrian to the curb as observations of the DBN and estimate a mode showing whether the pedestrian stops or crosses the street. These Bayesian prediction methods focus on pedestrians while our approach handles multiple kinds of target objects simultaneously.

In recent years, a path prediction method has been proposed that uses deep learning architectures, particularly LSTMs (Alahi et al., 2016)(Lee et al., 2017)(Fernando et al., 2017b)(Fernando et al., 2017a). Alahi *et al.* (Alahi et al., 2016) proposed a method to predict paths of multiple pedestrians in a scene. They aimed to predict collision avoidance behaviors between pedestrians and proposed a pooling layer called Social Pooling (S-Pooling). S-Pooling encodes hidden states of other pedestrians along with the spatial relationships. Lee *et al.* (Lee et al., 2017) proposed a path prediction method using a RNN encoder-decoder (Cho et al., 2014) and a conditional variational auto-encoder (Kingma et al., 2014). This method achieved high prediction performance by considering the semantic scene context of the surrounding area in addition to the interaction between the targets as with S-Pooling. However, they focused on predicting pedestrian targets or targets considered to be the same types of objects. In contrast, our approach inputs the attribute of a prediction target itself in addition to the surrounding physical environment.

Attempting to develop a method that takes into account the attribute of a target object, Ma *et al.* (Ma et al., 2017) proposed a method to predict pedestrian paths from a single image on the basis of an inverse reinforcement learning framework. Assuming that the walking speed of the pedestrian differs depending on age and gender, they first estimate the pedestrian attributes and then predict the paths of multiple pedestrians. This method makes predictions for environments where there are only pedestrians and does not use environmental data. Our method, however, predicts paths by simultaneously considering the attribute of the target object and the environmental data of the surrounding area.

3 PROPOSED METHOD

As mentioned in the previous sections, we focus on predicting paths of multiple kinds of target objects. We use the attribute of a target object and the surrounding physical environment information as inputs in addition to the past object trajectories.

Figure 1 illustrates the overview of our proposed network. First, to represent the object type, the attribute is embedded as a one-hot vector. Then, we extract a feature map via a convolutional neural network (CNN) to describe the environment around the target. A static scene label is used as an input for the CNN, focusing on the target object in the scene. The one-hot and feature vector are concatenated with the past object trajectory and input in an LSTM. We obtain the coordinates of the target object for the next time step as an output of the LSTM.

Our prediction method is relatively simple compared with other recent LSTM-based prediction methods (A. Vemula and OhSocial, 2017)(Alahi et al., 2016)(Lee et al., 2017)(Fernando et al., 2017b)(Fernando et al., 2017a). Instead of modeling complex architectures, we focused on reconsidering

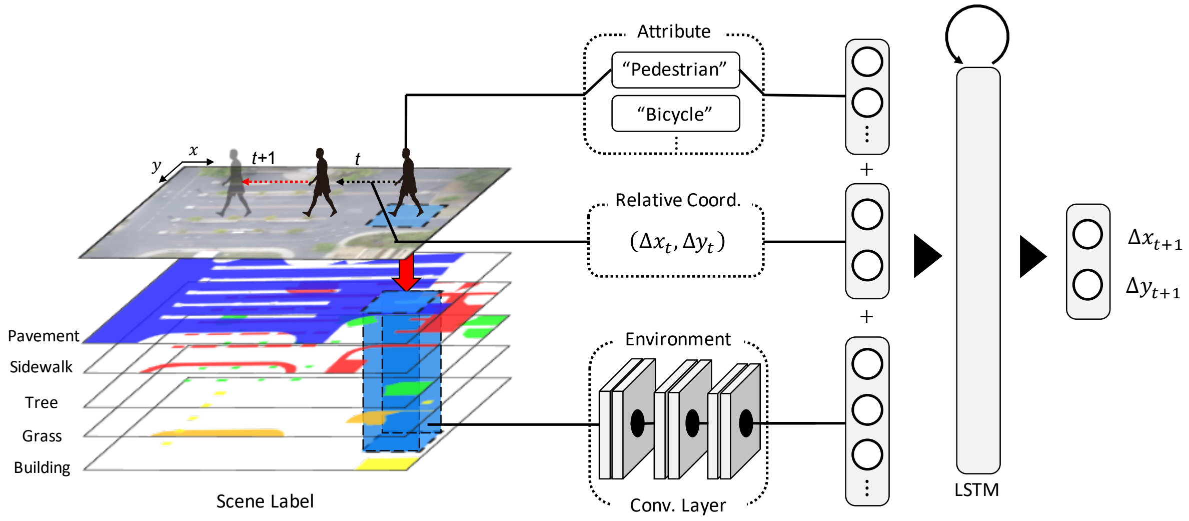


Figure 1: The overview of the proposed method. Our method uses the attribute of a prediction target, a relative coordinate, and the surrounding physical environment of the target as input for the network. The target attribute is embedded as a one-hot vector and a feature vector is extracted from semantic scene labels via convolutional layers. These vectors and the relative coordinate of the current time step are input to an LSTM and the relative coordinate of the next time step is output.

the information that can be useful for prediction. In the following subsections we describe the method used to represent the input data and how the data is input in the network.

3.1 Attribute

To predict paths of multiple kinds of target objects, we need to introduce some additional information representing object type as an input. We assume the object type as an inherent attribute included in the target and represent the attribute as a one-hot vector (see Figure 2). Specifically, given target attributes (e.g., pedestrian or car), these attributes are embedded into N_{attr}-dimensional vectors, where N_{attr} is the number of attributes being considered. The element corresponding to the input attribute is set to 1 and the others are set to 0. Inputting this vector enables us to predict a unique path with respect to speed and turn. Moreover, the area where the target tends to move is also considered by combining the one-hot vector with the feature vector representing physical environmental information.

3.2 Object Trajectory

We use relative coordinates as has been mentioned. Specifically, we calculate the travel distance $(\Delta x_t, \Delta y_t)$ from the past location data and the current location data, that is, the difference in the absolute coordinates. By using the relative coordinates as input to the LSTM, we obtain the relative coordinates of the next time step. Using relative coordinates enables us to always set the current location of the target object as the base point, i.e., $(x_t, y_t) = (0,0)$, and to make a prediction without depending on implicit scene information derived from coordinates of training data. Therefore, we can predict paths in multiple scenes.

3.3 Environment

The environmental information is also essential to improve prediction performance. Accordingly, we extract a feature map that represents the surrounding environment by using semantic scene labels added to a scene from a sidewalk, building, etc. Figure 3 shows the procedure for extracting input data for the proposed network from a whole semantic scene label. First, we extract a label map by trimming the label of the area (100×100 [pixels]) - focused on the target object - from the scene label. Then, we convert the extracted label map to a binary map whose channels correspond to each semantic object (e.g., building and sidewalk). The feature map for the surrounding environment is extracted from this binary map via a CNN. Inputting the environmental data enables us to make path predictions in which any existing obstacles or areas are taken into account in accordance with the attributes of the target objects.

3.4 Method to Input Data in the Network

By inputting the attributes, environmental feature map, and relative coordinate in the LSTM, we obtain

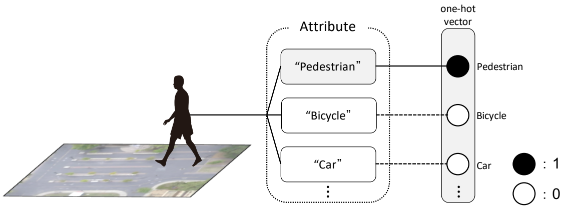


Figure 2: The representation of the attribute of a target object. This shows that the attribute is a pedestrian.

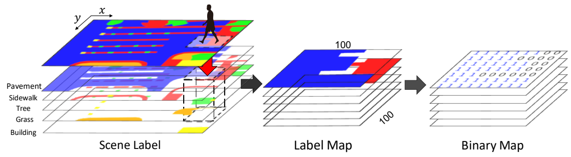


Figure 3: The representation of the physical environment surrounding a prediction target. We first extract a label map by trimming the area centering around the target object from the scene label. The trimmed label map is converted into a binary map. A feature map is extracted from this binary map via convolutional layers.

the location of the target object in the next time sequence. Specifically, we use the data of the target object as the observation data and make a prediction. We use the true value that the target object actually moves as the observation data. We input the observation data sequentially in the frames until we start predicting. When we make a prediction, we sequentially input the prediction value (i.e., an output of the LSTM) to the next time sequence. We carry out the process until the prediction ends so we can make a prediction.

4 EXPERIMENT

This section demonstrates the effectiveness of the proposed path prediction method.

4.1 Dataset

For the evaluation, we used the Stanford Drone Dataset (SDD) (Robicquet et al., 2016). The SDD consists of eight different prediction scenes and each scene contains several video clips filmed on different days and/or times, consisting of a total of 60 video clips. In the SDD, six classes of target objects (i.e., bicycle, pedestrian, cart, car, bus, and skateboarder) are given and these are added to annotated paths. In our experiments, we used the six object classes as attributes. We observed the coordinates of the path used in our experiments every 20 frames. Because the SDD clips are filmed at 30 fps, each time step corresponds to about 0.66 [s]. During the test time, we observed a path for the first five frames (i.e., 3.3 [s]) and then predicted the following eight frames (i.e., about 5.3 [s]).

Table 1: Training and test data details.

		train	test
	No. of scenes	52	8
attribute	bicycle	2,369	545
	pedestrian	2,696	500
	cart	71	15
	car	75	5
	bus	17	2
	skateboarder	137	15

The proposed method leverages semantic scene labels to extract the feature map of the physical environment. However, the SDD does not include such scene semantics. We therefore annotated semantic scene labels for every 60 prediction scenes with respect to the following three movable region classes and four obstacle classes: sidewalk, pavement, grass, bicycle storage, tree, building, and roundabout. Figure 4 shows examples of annotated scene labels. These scene labels do not reflect only the visual appearance from bird's eye view images but also the ground where prediction targets move. It should be noted that the SDD contains a lot of incorrect and/or inaccurate annotated paths; examples are shown in Figure 5. In these examples, lost, occlusion, and interpolation flags are annotated in addition to the coordinates. However, as far as we were able to confirm, target objects corresponding to incorrect paths do not exist in the original video clips even if we take the flags into account (see Figure 5 (a, b, c). Figure 5(d) provides an example of an inaccurately annotated path. Using such paths for training and evaluation decreases the prediction performance and makes fair comparisons difficult. Hence, we carefully selected only the accurate and correct annotations. As a result, the number of target objects selected was 5,365 for learning and 1,082 for evaluation. Table 1 shows the details of the data being used. This our annotated dataset will be publicly available after acceptance.

4.2 Evaluation Metrics and Baselines

In these experiments we used two metrics for quantitative evaluation. The first is *final displacement error*, which is a Euclidean distance for the ground truth trajectory and the predicted trajectory in the last prediction time steps. The second is *average displacement error*, which is the average of Euclidean distances between the ground truth trajectory and the predicted trajectory in every prediction time step.

We compare our method with Kalman filter (KF)(Kalman, 1960) and Social LSTM(S-LSTM) as a baseline prediction approach.

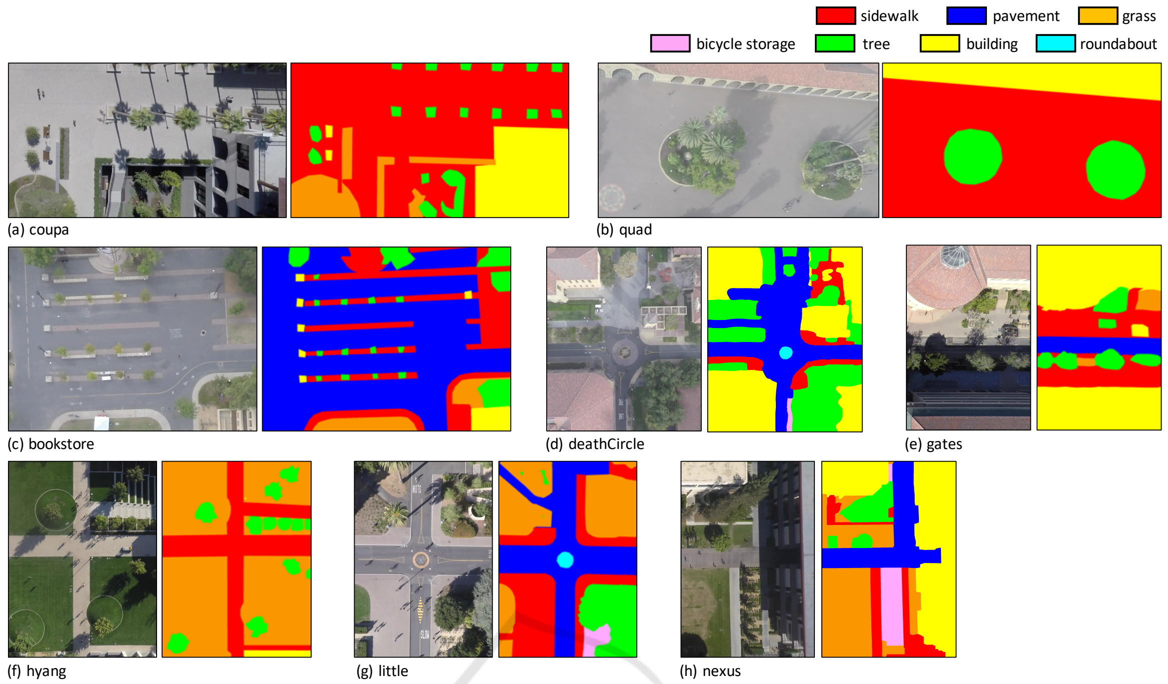


Figure 4: Examples of annotated scene labels in the SDD. For each sub-figure, the left shows an original scene image from a bird's eye view and the right shows the corresponding semantic scene labels. We annotated scenes into seven classes in accordance with the ground rather than with the visual appearance of the scene images. These labels will be made publicly available after acceptance.

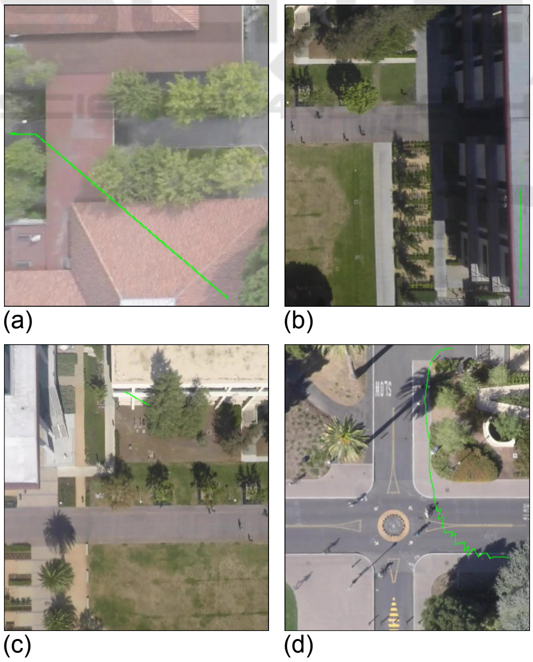


Figure 5: Examples of incorrect annotations in the SDD. The green lines show the annotated paths.

4.3 Learning Details

Table 2 shows the details of the network architecture. We trained our model with RMSprop optimizer (Tieleman and Hinton, 2012) with the initial learning rate of 0.01, $\alpha = 0.99$, and $\varepsilon = 10^{-8}$. All prediction models were trained for 100 epochs with a batch size of 10. During the training, we input ground truth coordinates as past object trajectories through every time step, i.e., from the beginning of observation to the end of prediction. All the LSTM-based prediction models were implemented using the Chainer framework and trained with the Nvidia Titan Xp graphics card in an end-to-end manner.

4.4 Results

Table 3 shows the quantitative results of prediction methods and Figure 6 shows examples of prediction results. Because past trajectories are only considered as observations with KF, with this method the prediction results follow the same direction as the observations and thus linear predictions without obstacle regions are provided. The LSTM-based method provided similar prediction results when a trajectory is used (Figure 6 (d)). However, in other cases its prediction results were poorer than those of KF (Figure 6

Table 2: The detailed network architecture of the proposed method. Convolutional layers are applied for the input with respect to the environment. The feature map via the convolutional layers and other inputs (i.e., attributes and coordinates) are input to an LSTM.

layer	kernel size	output size	remarks
input (attribute)		6	
input (coordinate)		2	
input (environment)		(100, 100, 7)	
conv1	(5, 5)	(48, 48, 16)	ReLU, stride=2
norm1		(48, 48, 16)	batch norm.
pool1	(2, 2)	(24, 24, 16)	max pool.
conv2	(5, 5)	(20, 20, 32)	ReLU, stride=1
norm2		(20, 20, 32)	batch norm.
pool2	(2, 2)	(10, 10, 32)	max pool.
conv3	(5, 5)	(6, 6, 32)	ReLU, stride=1
pool3	(2, 2)	(3, 3, 32)	max pool.
concat		296	
LSTM		128	
output		2	

(a, c, g)). S-LSTM does not outperform our method and even KF. Although we have carefully selected parameters to reproduce the result, we could not obtain reasonable results. The obtained prediction results of S-LSTM were catastrophic. Therefore, for the sake of visibility, we do not show prediction results for S-LSTM in Figure 6. The same problem is reported in (Gupta et al., 2018).

As can be seen in Table 3, introducing other information into the LSTM improves the prediction accuracy. In particular, introducing physical environment information makes it possible to predict paths accurately while avoiding obstacles (Figure 6 (h)). However, the improvement is relatively small from the viewpoint of quantitative evaluation and the errors differ from the KF errors. Meanwhile, our proposed method, trajectory + attribute + environment, outperforms the other methods. The proposed method was able to predict paths close to the ground truth in Figure 6 (a, b, c, g).

Figure 6 (b, c) shows the trajectory of the bicycle; the ground truth is moving while avoiding obstacles. However, it has been confirmed that when only KF, object trajectory, and attribute information are introduced as input, a target will go straight ahead without avoiding obstacles. In addition, when introducing environmental information, it predicts the trajectory to take to avoid obstacles, but this confirms that predictions different from the ground truth can be made. However, when both attribute and environmental information are introduced, a trajectory similar to the ground truth is predicted. Figure 6(d, e, f) shows the trajectory of the pedestrian. The results obtained in this case showed that all the path prediction methods traced a path close to the ground truth. This is probably because the path of the pedestrian can be predicted easily because the movement intervals are narrower than those for the bicycle. Figure 6(g) shows the trajectory of the car, where the object to be predicted along the roadway. However, when only KF, object trajectory, and environmental information are introduced as input, the prediction result is that it will go straight ahead. When attributes are introduced in the environment, it can be seen that a trajectory similar to the ground truth is predicted. However, as shown in Figure 6(h, i), when environmental information is introduced the prediction results show a trajectory different from the ground truth.

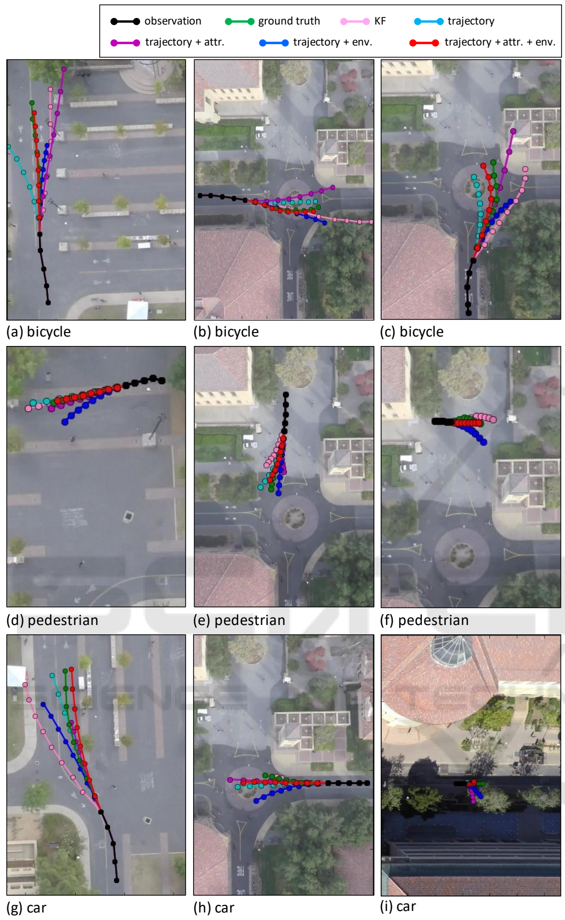
The above results confirmed the proposed method has the highest accuracy among the path prediction methods compared. Although the conventional method KF predicts linear trajectories well, it is difficult for it to predict nonlinear trajectories such as those made in obstacle avoidance cases. To predict paths more accurately, it is necessary to introduce attributes and environmental information into object trajectories.

4.5 Failure Cases

Figure 7 shows examples of failed prediction results with relative coordinates. Figure 7(a) shows a case in which the speed of the bicycle suddenly changes from slow to fast. In such cases, prediction methods provide a slowly moving path by following the observations although the ground truth moves faster. In Figure 7(b), although the ground truth path turned left, the prediction results are almost straight lines. In cases where there may be several prediction candidates, our method follows the direction of the past movement. In Figure 7(c), the proposed method provides paths that move towards the pavement so as to avoid collisions with obstacles, while the ground truth takes a different path. The reason is that a car moves in accordance with specific traffic rules, making it necessary to consider common social practice. Figure 7(d, e, f) are prediction results for a cart and skateboarders. As shown in Table 1, there was insufficient training data (and also test data) for these attributes. As a result, the training was insufficient. Consequently, all the prediction methods predicted incorrect paths and could not even avoid obstacles (i.e., building and roundabout). Hence, achieving efficient training for cases involving rare attribute targets is a subject for our future work.

Table 3: Quantitative results for prediction methods (unit: pixels). Introducing attribute and environment information improves the prediction performance. Our method, trajectory + attribute + environment, achieves the best performance with respect to both final displacement error and average displacement error.

Metric	KF	S-LSTM	trajectory	trajectory + attribute	trajectory + environment	**trajectory + attribute + environment**
Final disp. error	174.42	206.22	196.13	173.04	172.12	**109.44**
Avg. disp. error	116.02	125.41	86.42	76.32	76.01	**53.20**

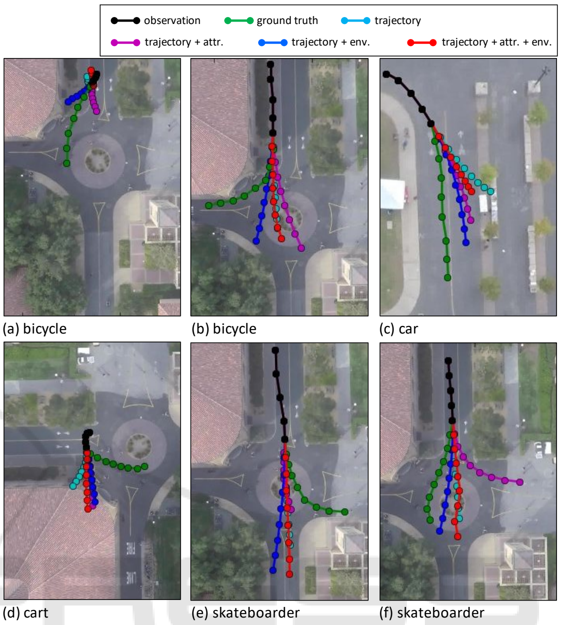


(a) bicycle (b) bicycle (c) bicycle
(d) pedestrian (e) pedestrian (f) pedestrian
(g) car (h) car (i) car

Figure 6: Examples of prediction results with relative coordinates on SDD. From top to bottom row: prediction results for a bicycle, pedestrian, and car.

5 CONCLUSIONS

In this paper, we proposed a path prediction method that takes target object attributes and physical environment information into account. The method represents the attributes as one-hot vectors and encodes the physical attributes via convolutional layers. Furthermore, we used relative coordinates as the past motion history of prediction targets. Sequentially inputting these data items in a long short-term memory enables the method to make predictions. Experimental results obtained using the Stanford Drone Dataset show that our approach to introducing those factors improves the prediction performance. Our future work will include taking the interaction between the target objects and dynamic environmental changes into consideration.



(a) bicycle (b) bicycle (c) car
(d) cart (e) skateboarder (f) skateboarder

Figure 7: Selected failed prediction results. Our proposed method cannot predict paths (a) that change their moving speed suddenly, (b) that may have multiple candidates, and (c) that follow common social practice. The bottom row (d, e, f) shows the results obtained for rare attribute targets. Trained models with fewer training samples predict incorrect paths.

ACKNOWLEDGMENTS

This work was supported in part by JSPS KAKENHI grant number JP16H06540. And, we gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

A. Vemula, K. M. and OhSocial, J. (2017). Attention: Modeling attention in human crowds. *International Conference on Robotics and Automation*.

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. (2016). Social Lstm: Human Trajectory Prediction in Crowded Spaces. In *Computer Vision and Pattern Recognition*, pages 961–971.

Ballan, L., Castaldo, F., Alahi, A., Palmieri, F., and Savarese, S. (2016). Knowledge transfer for scene-specific motion prediction. In *European Conference on Computer Vision*, pages 697–713.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Fernando, T., Denman, S., McFadyen, A., Sridharan, S., and Fookes, C. (2017a). Tree memory networks for modelling long-term temporal dependencies. *arXiv preprint arXiv:1703.04706*.

Fernando, T., Denman, S., Sridharan, S., and Fookes, C. (2017b). Soft + hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *arXiv preprint arXiv:1702.05552*.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. (2018). Social gan: Socially acceptable trajectories with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264.

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

Huang, S., Li, X., Zhang, Z., He, Z., Wu, F., Liu, W., Tang, J., and Zhuang, Y. (2016). Deep learning driven visual path prediction from a single image. *IEEE Transactions on Image Processing*, 25(12):5892–5904.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

Karasev, V., Ayvaci, A., Heisele, B., and Soatto, S. (2016). Intent-aware long-term prediction of pedestrian motion. In *International Conference on Robotics and Automation*, pages 2543–2549.

Keller, C. G. and Gavrila, D. M. (2014). Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):494–506.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589.

Kitani, K. M., Ziebart, B. D., Bagnell, J. A., and Hebert, M. (2012). Activity forecasting. In *European Conference on Computer Vision*, pages 201–214.

Kooij, J. F. P., Schneider, N., Flohr, F., and Gavrila, D. M. (2014). Context-based pedestrian path prediction. In *European Conference on Computer Vision*, pages 618–633.

Lecun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Backpropagation Applied to handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H. S., and Chandraker, M. (2017). Desire: Distant future prediction in dynamic scenes with interacting agents. In *Computer Vision and Pattern Recognition*, pages 336–345.

Ma, W., Huang, D., Lee, N., and Kitani, K. M. (2017). Forecasting interactive dynamics of pedestrians with fictitious play. In *Computer Vision and Pattern Recognition*, pages 774–782.

Park, H. S., Hwang, J. J., Niu, Y., and Shi, J. (2016). Egocentric future localization. In *Computer Vision and Pattern Recognition*, pages 4697–4705.

Rehder, E. and Kloeden, H. (2015). Goal-directed pedestrian prediction. In *Workshop on International Conference on Computer Vision*, pages 139–147.

Robicquet, A., Sadeghian, A., Alahi, A., and Savarese, S. (2016). Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, pages 549–565.

Robinson, J. W. and Hartemink, A. J. (2009). Non-stationary dynamic bayesian networks. In *Advances in Neural Information Processing Systems*, pages 1369–1376.

SchneiderNicolas and M., G. (2013). Pedestrian path prediction with recursive bayesian filters: A comparative study. In *German Conference on Pattern Recognition*, pages 174–183.

S.Hochreiter (1997). LONG SHORT-TERM MEMORY. *Neural Computation*, 9(8):1735–1780.

Su, S., Hong, J. P., Shi, J., and Park, H. S. (2017). Predicting behaviors of basketball players from first person videos. In *Computer Vision and Pattern Recognitionr*, pages 1502–1510.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*.

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

Walker, J., Gupta, A., and Hebert, M. (2014). Patch to the future: Unsupervised visual prediction. In *Computer Vision and Pattern Recognition*, pages 3302–3309.

Xie, D., Todorovic, S., and Zhu, S. C. (2013). Inferring 'Dark Matter' and 'Dark Energy' from videos. In *International Conference on Computer Vision*, pages 2224–2231.

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

Yi, S., Li, H., and Wang, X. (2016). Pedestrian behavior understanding and prediction with deep neural networks. In *European Conference on Computer Vision*, pages 263–279.

Ziebart, B., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J., Hebert, M., Dey, A., and Srinivasa, S. (2009). Planning-based prediction for pedestrians. In *International Conference on Intelligent Robots and Systems*, pages 3931–3936.