Deep Learning Driven Visual Path Prediction From a Single Image

Siyu Huang, Xi Li, Zhongfei Zhang, Zhouzhou He, Fei Wu, Wei Liu, Jinhui Tang, and Yueting Zhuang

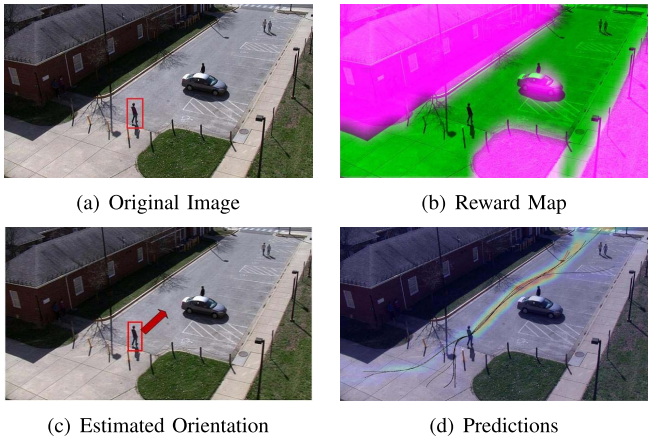
***Abstract*— Capabilities of inference and prediction are the significant components of visual systems. Visual path prediction is an important and challenging task among them, with the goal to infer the future path of a visual object in a static scene. This task is complicated as it needs high-level semantic understandings of both the scenes and underlying motion patterns in video sequences. In practice, cluttered situations have also raised higher demands on the effectiveness and robustness of models. Motivated by these observations, we propose a deep learning framework, which simultaneously performs deep feature learning for visual representation in conjunction with spatiotemporal context modeling. After that, a unified path-planning scheme is proposed to make accurate path prediction based on the analytic results returned by the deep context models. The highly effective visual representation and deep context models ensure that our framework makes a deep semantic understanding of the scenes and motion patterns, consequently improving the performance on visual path prediction task. In experiments, we extensively evaluate the model's performance by constructing two large benchmark datasets from the adaptation of video tracking datasets. The qualitative and quantitative experimental results show that our approach outperforms the state-of-the-art approaches and owns a better generalization capability.**

***Index Terms*— Visual path prediction, visual context model, convolutional neural networks, deep learning.**

I. Introduction

INFERENCE and prediction are significant capabilities of intelligent visual systems [1] such that they have been popular topics in computer vision community during recent years. As part of visual inference and prediction, we address the visual path prediction problem, with the goal of inferring the most possible future path for an object in a static scene image. For instance, given a single static image like Fig. 1(a), we humans can easily recognize the objects inside it, and tell others which ones are active — persons and car will move, but grass and house will remain still. Furthermore, for the active objects, we could naturally infer their intentions and future motions. Taking the man at bottom left with red bounding box for an example, he is most likely to walk straight, meanwhile, bypassing the car which appears to be an obstacle for him. The aforementioned visual inference process is illustrated as the red path in Fig. 1(d). As a matter of fact, these predictions are naturally driven by a human visual system and supported by the prior knowledge stored in it.

In this work, we aim to automatically learn this prior knowledge from a diverse set of videos, and further infer the possible future motions of objects. The prior knowledge here includes both the scene structure and motion patterns underlying the video sequences. More specifically, they can be respectively associated with the contextual properties of the scene structure from the spatio-temporal perspectives. Therefore, the key way of solving the visual path prediction task is modeling the spatial and temporal context, followed by a certain inference algorithm to predict the future path.



(a) Original Image (b) Reward Map (c) Estimated Orientation (d) Predictions

Fig. 1. Illustration of our approach. Image (a) shows a man in the parking lots. The goal of visual path prediction is to infer the possible paths for him in the future. In this paper, we first generate a reward map (b) representing regions the man can reach in the future (green). Then, we estimate his facing orientation (c). Finally, we incorporate the results of (b) and (c) to plan the most likely paths as shown in (d), where the red line and the black lines respectively show our top-1 and top-10 predictions.

Manuscript received January 13, 2016; revised June 24, 2016; accepted August 29, 2016. Date of publication September 26, 2016; date of current version October 25, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61672456, Grant 61472353, and Grant U1509206 and in part by the Zhejiang Provincial Engineering Center on Media Data Cloud Processing and Analysis. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dacheng Tao. *(Corresponding author: Xi Li.)*

S. Huang and Z. He are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: siyuhuang@zju.edu.cn; zhouzhouhe@zju.edu.cn).

X. Li, F. Wu, and Y. Zhuang are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: xilizju@zju.edu.cn; wufei@cs.zju.edu.cn; yzhuang@cs.zju.edu.cn).

Z. Zhang is with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, and also with the Computer Science Department, Watson School, The State University of New York at Binghamton University, Binghamton, NY 13902 USA (e-mail: zhongfei@zju.edu.cn).

W. Liu is with the Tencent AI Laboratory, Shenzhen 518057, China (e-mail: wliu@ee.columbia.edu).

J. Tang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: jinhuitang@njust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2613686

Such a task is very challenging because it not only needs deep semantic understanding of videos, but also is often confronted with very complicated situations. For instance, just a single scene in this task may contain various kinds of appearance which are easy to confuse with each other. To address this dilemma, the visual representation is typically required to be of rich semantics and high discrimination. On the other hand, the scenes and objects are usually diverse. It is required for the context model and visual representation to possess strong generalization ability for the adaptation to complex scenarios.

More recently, topics of visual inference and prediction are widely studied by computer vision reseachers, and there has been some work referring to the visual path prediction task. Earlier work [2], [3] focuses on the matching-based approaches. For instance, Yuen et al. [3] explore scene modeling by searching straightforwardly in image space with keypoint matching techniques using descriptors like GIST and dense SIFT. In general, these matching-based methods rely on large amount of data and do not really analyze the semantics of the scene. In the testing phase, they have to compare with all the alternative samples, leading to a high computation cost. In contrast, more recent work has paid attention into the learning-based approaches [4], [5]. The key concept is learning context model to capture the structure relationships between the scene and specific objects, followed by learning robust temporal models like IOC [4] for inference. The learning-based approaches seek to establish inductive models to understand the scene in depth, which results in the state-of-the-art performance in the visual path prediction task.

While in practice, the complex and cluttered situations in this task (e.g., a crowd of cars and people moving at the crossroads) have raised higher demands on the effectiveness and robustness of our models. In general, the conventional visual representations are based on handcrafted features, which are often much restrictive in complex visual scenes and thus cannot provide abundant semantic information about the visual content. Besides, the context model built in the aforementioned approaches is relatively simple and shallow, which leads to the incapability of modeling the intrinsic contextual interactions among objects as well as their associated scene structures. For instance, Walker et al. [5] build their context model by straightforwardly counting the votes from training data. Such a practice is difficult to effectively model the contextual information.

Motivated by these observations, in this paper we propose a unified deep learning framework for visual path prediction, which simultaneously performs deep feature learning for visual representation and spatio-temporal context modeling. Then, we propose a unified path planning scheme to predict accurate future path based on the analytic results of the context models. Compared with the conventional approaches to visual path prediction, the visual representation employed in our framework is highly effective because it has a better discrimination and generalization capability. Meanwhile, our deep context models can be better adapted to the complex scenarios. These improvements ensure that our framework can deeply analyze the scenes and motion patterns, consequently improving the performance in the visual path prediction.

The key contributions of our paper are summarized as follows:

1) We present a novel deep learning framework based on the CNNs for visual path prediction task. To the best of our knowledge, it is the first work to leverage a deep learning approach in this task. Our framework models both of the scene structure information and motion patterns. The abstraction of visual representation and the learning of context models are accomplished in a unified framework. It largely improves the scene understanding capability compared with the previous approaches.
2) We propose a unified path planning scheme to infer the future paths on the basis of the analytic results returned by our context models. In this scheme, the problem of future path inference is equivalently converted into an optimization problem, which can be solved in an efficient way.
3) We construct two benchmark datasets for visual path prediction from the adaptation of two large video tracking datasets [6], [7]. The adapted datasets are much larger than those used in the previous work and cover a diverse set of scenes and objects. They can be used for comprehensively evaluating the performance of a visual path prediction model.

II. Related Work

In general, the methods for visual path prediction contain two components: (1) Understanding the scene and motion pattern of the video sequences. (2) Inferring the future path based on information obtained by step (1). This section will review the representative methods of these two steps respectively.

A. Understanding the Scene and Motion Pattern

Scene understanding is a prerequisite to many high level tasks for intelligent systems operating in real world environments. In the past ten years, researchers have made great efforts in understanding the static image scene at a deeper level, including several typical topics like scene classification [8]–[12], semantic segmentation and labeling [13]–[17], depth estimation [18]–[20]. What these approaches have in common is that they learn and model the scene structure prior to recover different aspects of a scene. Accordingly, Yao et al. [9] propose a holistic structure prediction model based on CRF to jointly solve several scene understanding problems.

Except for modeling scenes in static images and inferring knowledge at the current state [21]–[23], an intelligent visual system is supposed to be able to infer what will happen in the near future. In more recent years, many researchers have paid attention to modeling the motion pattern in video sequences for temporal aspect of recognition and prediction. For instance, recognition and forecasting of human action [4], [24]–[30], event [3], [31]–[33] and scene transition [5], [34] have caught lots of interest. For dynamic scene understanding, the key is to model the structure relationships among different frames. As well, techniques of static scene understanding play a significant role in it.

B. Path Inference

Methods for path inference can be generally classified into two categories: the matching-based methods [2], [3], [35] and the learning-based methods [4], [5]. The matching-based methods simply retrieve the information from databases to the queries without building an inference model. For instance, Liu et al. [2] propose a method by matching a query image to a large amount of video data and warping the ground truth paths from the nearest neighbour videos to the static query image with SIFT Flow. Instead of the warping process, Yuen et al. [3] build localized motion maps as probability distributions after merging votes from several nearest neighbors. These matching-based approaches rely on the richness of the databases.

On the other hand, the learning-based methods learn temporal inference models to capture the spatio-temporal variation of scenes and objects. Temporal models such as Markov Logic Networks [36], IOC [4], CRF [34], ATCRF [24] and EDD [37] are often employed. These models help infer the future of individual objects. Further work has taken into consideration the relationships between objects and scenes. Kitani et al. [4] detect the physical scene features based on semantic scene labeling techniques [38], [39], and then, fuse them into the reward function of IOC. Walker et al. [5] build a temporal model based on the effective mid-level patches [40]. They learn patch-to-patch transition matrix, which serves as the temporal model, and learn a context model for the interaction between mid-level patch and the scene. These approaches draw on the strength of scene semantic understanding in depth, and successfully advance the overall performance for visual path prediction task.

However, the cluttered situations in real world have raised higher demands on the effectiveness and robustness of models. In aforementioned path inference approaches, the handcrafted features and shallow context models are often restrictive in complex visual scenes. Motivated by these issues, we employ deep visual representation in our framework because it has a better discrimination and generalization capability. Meanwhile, the deep context models built in our framework can be better adapted to complex scenarios. The unified path planning scheme is able to make reasonable path predictions based on the analytic results returned by the context models.

C. Convolutional Neural Networks

The proposed framework in this paper is built upon the convolutional neural networks (CNNs). The CNNs are a popular and leading visual representation technique, for they are able to learn powerful and interpretable visual representations [41]. The CNNs have given the state-of-the-art performance on various computer vision tasks [12], [14], [15], [42], [43]. The key enabling factors behind these results are techniques for scaling up the networks to tens of millions of parameters and massive labeled datasets that can support the learning process [12]. In recent years, the CNNs have been applied to scene parsing [14], [15], [44] and road detection [45]. On the other aspect, some work has combined CNNs with temporal modeling. Donahue et al. [46] use CNNs and LSTM [47] for visual recognition and description tasks. From the perspective of temporal prediction, Walker et al. [48] employ the same architecture of [46] to predict long term motion of pixels in terms of optical flow.

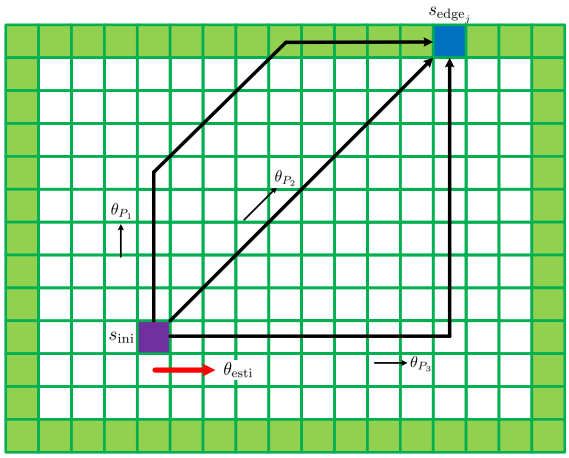


Fig. 2. A simple illustration of the visual path prediction problem. Between the object location s_{ini} and the j-th edge point s_{edge_j}, we desire to plan a path P which has the lower spatial matching costs on the cost map, meanwhile, the smaller angular difference between its initial moving direction θ_P and the estimated direction θ_{esti}.

III. Our Approach

A. Problem Formulation

In this work, we aim to build a framework to automatically solve the visual path prediction problem. Given a static scene image I and the bounding box $B = (b_1, b_2, w, h)$ of an object in I, the goal is to infer the most possible path $P = (s_1, s_2, s_3, \ldots, s_n)$ of the object in the future. Here, b_1, b_2, w, h are respectively the top left coordinate, the width, and the height of B. And $s = (x, y)$ represents the coordinate of a position, such that P consists of a sequence of adjacent positions. Fig. 2 gives an illustration of the problem. We formalize the original scene into a grid graph such that each grid corresponds to a specific position s_i of the scene. Between the object location s_{ini} (the center of B) and a certain edge location s_{edge_j}, there are a large amount of alternative paths. The question is how to select such an appropriate path P from the very large path space $\mathbb{P}$? We convert the original problem into an optimization problem of planning a path P with the lowest cost $\mathcal{C}$:

$$\begin{aligned} &\min_{P} \ \mathcal{C}(P), \\ &\text{s.t. } P \in \mathbb{P}. \end{aligned} \tag{1}$$

Then, the issue is how to formulate the cost $\mathcal{C}$ of a path P. Intuitively, if there are more obstacles on a path, the associated cost of it ought to be higher:

$$\mathcal{C}_{\text{S}}(P) = \sum_{s_i \in P} \boldsymbol{R}_{\text{cost}}(s_i). \tag{2}$$

$\boldsymbol{R}_{\text{cost}}$ is a cost map of the scene representing the cost of each coordinate position s_i. Therefore, we need to discover which regions of the scene the object can reach. Such a

TABLE I
THE DETAILED DESCRIPTION OF THE VARIABLES

Variables	Description
I	the image
B	the object bounding box
P	the path of the object
$\mathbb{P}$	the path space
G	the directed graph
W	the weights of graph edges
s	the coordinate of the position
θ	the angle
ψ	the parameters of neural network
$\boldsymbol{R}_{\text{cost}}$	the cost map of the scene
$\boldsymbol{R}_{\text{reward}}$	the reward map of the scene
$\boldsymbol{q}$	the local environment patches
$\mathcal{C}$	the cost of the path
$\mathcal{D}$	the difference between two angles
$\mathcal{F}$	the forward propagation process

structure relationship between the object and the scene is referred to as "spatial context matching"; thus $\mathcal{C}_{\text{S}}$ is referred to as the "spatial matching cost" in this paper. We build a deep context model called **Spatial Matching Network** to learn the spatial contextual information from the video sequences in the training phase. In the testing phase, Spatial Matching Network generates a cost map $\boldsymbol{R}_{\text{cost}}$ according to a testing scene image.

On the other hand, the object's current moving direction also crucially influences the path selection. Hence, paths which are consistent with the object's current moving direction θ_{GT} should have lower costs:

$$\mathcal{C}_{\text{O}}(P) = \mathcal{D}(\theta_P, \theta_{\text{GT}}). \tag{3}$$

Here, $\mathcal{C}_{\text{O}}$ is called as "orientation cost" in this paper. θ_P is the initial moving direction of P, and $\mathcal{D}(\theta_1, \theta_2)$ represents the angular difference between two angles θ_1 and θ_2. For the sake of motion orientation modeling, we build another deep context model called **Orientation Network** to learn the temporal contextual information underlied in video sequences. In the testing phase, Orientation Network estimates an object's facing orientation θ_{esti} as θ_{GT} from the single object image.

The above two types of costs — the spatial matching cost $\mathcal{C}_{\text{S}}$ and the orientation cost $\mathcal{C}_{\text{O}}$ adequately help us semantically understand the scene and make a decision about the future path. As shown in Fig. 2, suppose that the three paths P_1, P_2, P_3 have the same average accumulated costs on cost map $\boldsymbol{R}_{\text{cost}}$. Which one is optimal? P_3 wins out because its initial direction θ_{P_3} is closer to θ_{esti}. Therefore, the path cost $\mathcal{C}$ is written as

$$\mathcal{C}(P) = \mathcal{C}_{\text{S}}(P) + \varepsilon \mathcal{C}_{\text{O}}(P), \tag{4}$$

where ε is a trade-off coefficient between the two types of costs. Finally, substituting $\mathcal{C}(P)$ into the optimization problem (1), we propose a unified path planning scheme to solve it in an easy and efficient way. For reading convenience, we summarize a collection of important notations used in this paper as Table I.

Fig. 3 shows our general framework in the testing phase. The far left of the figure is the input of visual path prediction problem, containing a scene image of parking lots and a bounding box of a car. We employ two CNNs to semantically analyze different aspects of the scene and the object. The first CNN, which we call Spatial Matching Network, generates a reward map $\boldsymbol{R}_{\text{reward}}$ representing the reward of every pixel on the scene image. The larger reward means the higher probability the car will reach that pixel position in the future. The reward map $\boldsymbol{R}_{\text{reward}}$ is then converted into a cost map $\boldsymbol{R}_{\text{cost}}$ for the subsequent path planning. The second CNN, which we call Orientation Network, outputs an estimated facing orientation θ_{esti} of the car. And then, based on the analytic results $\boldsymbol{R}_{\text{cost}}$ and θ_{esti}, we infer the most possible future paths of the car by solving the optimization problem (1).

In such a framework, there are still some important problems to solve in what follows: For the two networks, how do we learn the contextual information from video sequences, and, what are the appropriate architectures of them? How do we efficiently solve the optimization problem (1)? We will discuss these issues in the following subsections.

B. Spatial Matching Network

We build Spatial Matching Network to model the interaction relationships between various objects and regions in scenes, namely the spatial context. More intuitively, for example, a pedestrian is more likely to walk on the pavement than climbing over the fence beside it. Here we call the pedestrian and the pavement as spatial context matching, while the pedestrian and the fence are not spatial context matching. As another example, if there is a house in front of a car, the car is supposed to detour the house but not to crash into it. Obviously the car is not spatial context matching with the house in this case. In our framework, such relationships are modeled by Spatial Matching Network.

Fig. 4 illustrates the architecture of Spatial Matching Network. We expect the network to model the relationship between two instances, so it takes two image patches as its input at the same time. One represents the given object and the other is a certain local environment patch obtained by a sliding window on the entire scene image, denoted as the blue boxes with dotted lines shown in Fig 4. The two inputs respectively propagate through two CNNs from conv1 to fc7 and then concatenated into a new fully connected layer fc8. The layers from conv1 to fc7 are similar to the AlexNet [42]. Note that the parameters of the two CNNs are different, as their inputs come from two different semantic spaces. We use a softmax layer at the output end of Spatial Matching Network. In the training phase, the label $L_{\text{S}} \in \{0, 1\}$ of the network is set as 1 if the two input patches are spatial context matching. Otherwise it is set as 0. In the testing phase, the network outputs the likelihood r of spatial context matching between the object patch I_{object} and the local environment patch q:

$$r = \mathcal{F}_{\text{S}}(I_{\text{object}}, q; \psi_{\text{S}}). \tag{5}$$

I_{object} is obtained according to the object bounding box B on the scene image I. $\mathcal{F}_{\text{S}}$ represents the forward propagation in

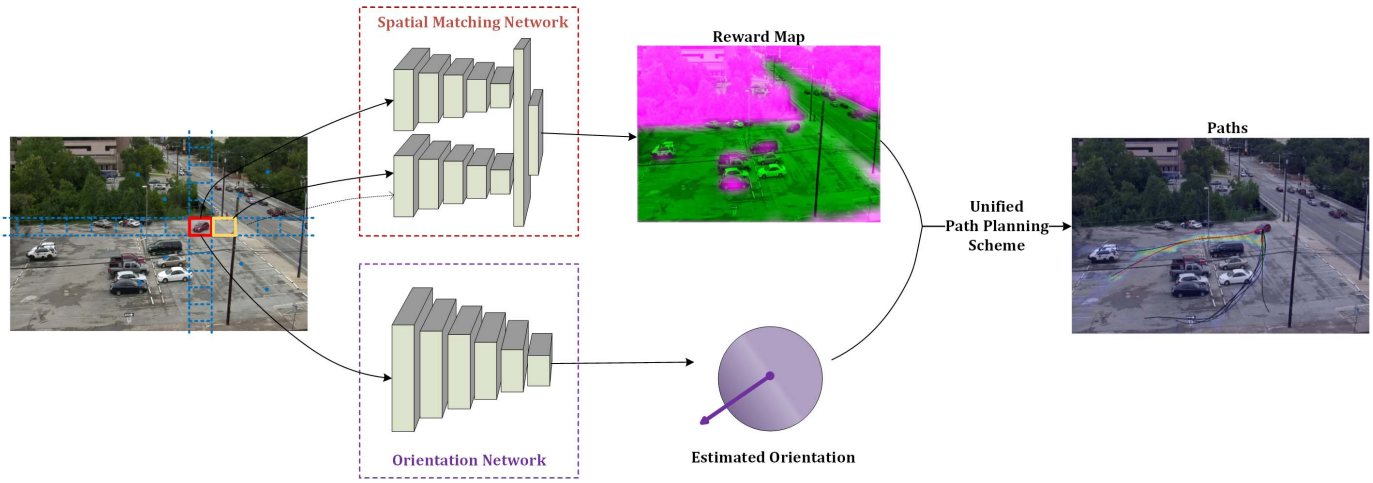


Fig. 3. The overview of our framework. Spatial Matching Network and Orientation Network are two CNNs, which respectively model the spatial and temporal contexts. We repeatedly input images of the object and local environment patches into Spatial Matching Network to generate the reward map of the scene. Intuitively, it helps us decide whether the object could reach certain areas of the scene. Orientation Network estimates the object's facing orientation, which indicates the object's preferred moving direction in the future. Then we incorporate this analysis and infer the most likely future paths with a unified path planning scheme.

Spatial Matching Network, and ψ_{S} are its learned parameters. For a scene image I, we can crop out the local environment patches $\boldsymbol{q} = \{q_{s_1}, q_{s_2}, \ldots, q_{s_t}\}$ with an overlapped sliding window on I, where $s_i = (x_i, y_i)$ is the central position of patch q_{s_i}. In this way, we can generate a reward map $\boldsymbol{R}_{\mathrm{reward}}$ for an object I_{object} and a scene image I by repeatedly inputting all the local environment patches $\boldsymbol{q}$ with the same object patch I_{object} into Spatial Matching Network:

$$\boldsymbol{R}_{\mathrm{reward}}(s_i) = \mathcal{F}_{\mathrm{S}}(I_{\mathrm{object}}, q_{s_i}; \psi_{\mathrm{S}}). \tag{6}$$

$\boldsymbol{R}_{\mathrm{reward}}(s_i) \in [0, 1]$ is the reward r for each position s_i. The larger value means the higher reward for that position, namely the higher probability the object will reach that position in the future. Visualizations of our reward maps generated on different scenes are shown in the middle column of Fig. 7.

It is noted that the reward function in the previous work [4], [5] only models the scene itself. However, different objects may have different relationships with the same region of the scene. So the reward map in our method is built with respect to both of the specific object and the scene appearance for the purpose of generalization across a diverse set of scenes and objects.

The reward map $\boldsymbol{R}_{\mathrm{reward}}$ can be converted to the cost map $\boldsymbol{R}_{\mathrm{cost}}$, such that:

$$\boldsymbol{R}_{\mathrm{cost}}(s) = \frac{1}{1 + e^{-\alpha(\boldsymbol{R}_{\mathrm{reward}}(s) - \gamma)}}, \tag{7}$$

where α is the tolerance to obstacles. γ is fixed to 0.5, as the scale of $\boldsymbol{R}_{\mathrm{reward}}(s)$ is [0, 1]. Based on this formulation, we can compute the spatial matching cost $\mathcal{C}_{\mathrm{S}}$ of a path P according to Eq. (2).

C. Orientation Network

In this subsection, we discuss how to build Orientation Network to learn the temporal context from video sequences in the training phase, and to estimate an object's facing orientation θ_{esti} in the testing phase. Since the scene is assumed to be static in the visual path prediction task, we only focus on modeling the temporal context of the object itself. In other words, we are to model the time-dependent variation of the object's own state. The state here includes the physical appearance and the spatial position. As the information about physical appearance has been integrated in Spatial Matching Network, we only model the position variation of the object itself, namely the relative position of the object at a different time. When in the testing phase, it is represented as the object's facing orientation with the input of a single image. The temporal context also plays an important role in selecting the future path. For instance, imagine a man walking on the street; he is most likely to walk along his facing orientation. Similarly, any kind of active object follows this rule if there are no other external factors disturbing it.

Therefore, we build Orientation Network to estimate an object's facing orientation θ_{esti}. The architecture of Orientation Network is shown in Fig. 5. We first extract image features using the standard seven-layer architecture similar to AlexNet [42] and then embed the features to low-dimensional space with linear mapping. At the output end of Orientation Network, the low-dimensional features are finally regressed into a single value $\theta_{\mathrm{esti}} \in (-\pi, +\pi]$, which represents the estimated angle of the object's facing orientation. Now we decide an appropriate loss function. Intuitively, the orientation estimation can be posed as either classification or regression. Walker et al. [5] treat it as classification because the state space of their temporal model is discrete. However, the orientation angle has reasonably high spatial self-correlation. The labels in classification task are often not sufficiently related to each other or even mutually exclusive. Therefore, we use regression as the output of Orientation Network, in view of its correlation between labels. We use Euclidean distance as the regression loss $loss_{\mathrm{o}}$ of Orientation Network:

$$loss_{\mathrm{o}} = \mathcal{D}^2(\theta_{\mathrm{GT}}, \theta_{\mathrm{esti}}), \tag{8}$$

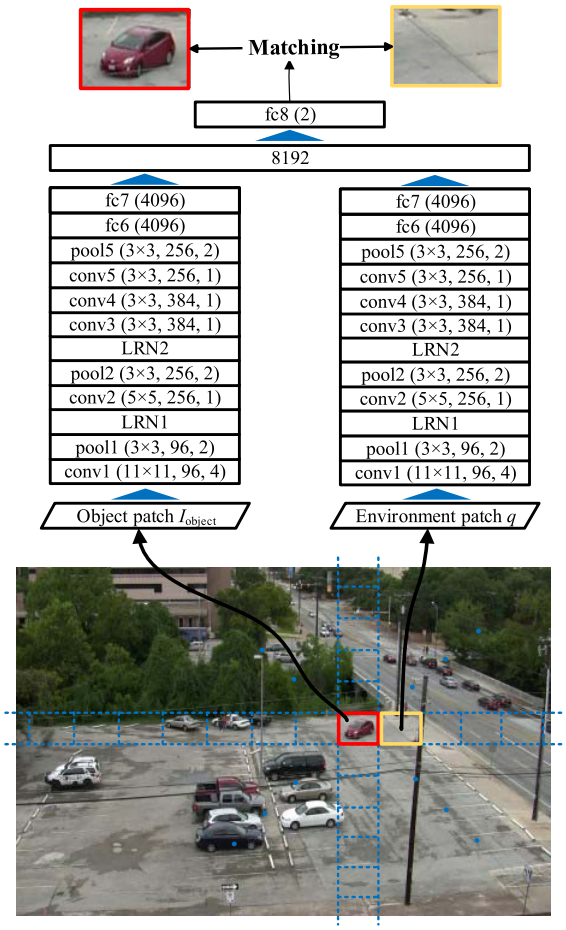


Fig. 4. Illustration of Spatial Matching Network. The bottom is the scene image. We crop out the object image patch I_{object} with the bounding box shown in red. We use a sliding window on the entire scene image to crop out the local environment patches, shown as the blue boxes with dotted lines. Each time we input the object patch I_{object} and an environment patch q into the network. The network outputs the likelihood of spatial context matching between the two patches. In this figure, two inputs are the car and the ground. They are spatial context matching, so label L_{S} for this sample is set as 1 during training.

where $\theta_{\text{GT}} \in (-\pi, +\pi]$ is the ground truth angle set as the relative position of the same object between two neighbouring frames, and θ_{esti} is the output of Orientation Network. $\mathcal{D}(\theta_1, \theta_2)$ is the angular difference between two angles θ_1 and θ_2:

$$\mathcal{D}(\theta_1, \theta_2) = \begin{cases} |\theta_1 - \theta_2| & |\theta_1 - \theta_2| \leqslant \pi, \\ 2\pi - |\theta_1 - \theta_2| & |\theta_1 - \theta_2| > \pi. \end{cases} \tag{9}$$

In the testing phase, we can estimate the facing orientation θ_{esti} of the input object image I_{object} by doing forward propagation $\mathcal{F}_{\text{O}}$ in Orientation Network:

$$\theta_{\text{esti}} = \mathcal{F}_{\text{O}}(I_{\text{object}}; \psi_{\text{O}}), \tag{10}$$

where ψ_{O} are the learned parameters of Orientation Network.

D. Path Planning

Up to now, the contextual properties of the scene structure are respectively formalized to be a cost map $\boldsymbol{R}_{\text{cost}}$ corresponding to the scene and an estimated facing orientation θ_{esti}

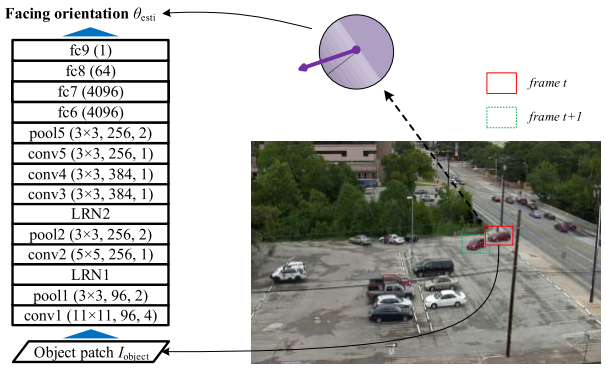


Fig. 5. Illustration of Orientation Network. What is the facing orientation of the given object? We train Orientation Network to estimate it accurately. The network takes an object image I_{object} as input. It outputs the estimated facing orientation angle θ_{esti} of the object. The architecture from conv1 to fc7 is similar to AlexNet [42]. We add fc8 and fc9 to reduce features' dimensionality, followed by a regression layer. The relative position of the same object between neighbouring frames serves as the ground truth label.

corresponding to the object. How do we plan the most probable future path for the given object? We propose a unified path planning scheme by efficiently solving the primitive optimization problem (1). The right part of Fig. 3 illustrates the function of this scheme.

The optimization problem (1) aims to find the optimal path $P = (s_1, s_2, \ldots, s_n)$ from the path space $\mathbb{P}$, which has the lowest path cost $\mathcal{C}(P)$. By combining Eqs. (2), (3), (4) and a few constraints to $\mathbb{P}$, we rewrite problem (1) as:

$$\begin{aligned} &\min_{P} \sum_{s_i \in P} \boldsymbol{R}_{\text{cost}}(s_i) + \varepsilon \mathcal{D}(\theta_P, \theta_{\text{esti}}), \\ &\text{s.t. } s_i \text{ and } s_{i+1} \text{ are spatially adjacent,} \\ &\quad s_1 = s_{\text{ini}}, \\ &\quad s_n = s_{\text{edge}_j}, \quad j = 1, \ldots, m, \end{aligned} \tag{11}$$

where the first constraint means that the object can only move to one of its adjacent positions in every step. In our experiments we use eight directions (top, left, bottom, right, top-left, top-right, bottom-left, bottom-right). The second and the third constraints specify the starting and ending positions of paths, where m is the number of edge points. The initial moving direction θ_P of P is obtained by computing the relative position between the initial position s_{ini} and a certain position s_i on P. In our experiments, the distance d between s_{ini} and s_i is fixed to the diagonal length of the object bounding box B: $d = \lfloor \sqrt{w^2 + h^2} \rfloor$, where $\lfloor \cdot \rfloor$ is the rounding floor. ε is set to 5 as a matter of experience.

In order to solve problem (11) more efficiently and easily, we employ a graph shortest path algorithm. We build a directed graph $G = (V, E)$ whose nodes v_i respectively correspond to the positions s_i of map $\boldsymbol{R}_{\text{cost}}$. The weights W of edges $e(v_i, v_j)$ are:

$$W(v_i, v_j) = \begin{cases} \boldsymbol{R}_{\text{cost}}(s_j) + \varepsilon \mathcal{D}(\theta_{s_j}, \theta_{\text{esti}}) & \text{for (I)}, \\ \boldsymbol{R}_{\text{cost}}(s_j) & \text{for (II)}, \\ +\infty & \text{others}, \end{cases} \tag{12}$$

Algorithm 1 Visual Path Planning Framework

Input: Scene image I, object bounding box B, network parameters $\psi_{\mathrm{S}}, \psi_{\mathrm{O}}$
Output: Predicted paths $\boldsymbol{P} = (P_1, P_2, \ldots, P_m)$
Scene Analysis
1. Generate the reward map $\boldsymbol{R}_{\mathrm{reward}}$
- Crop out the object image I_{object} according to B, and the scene patches $\boldsymbol{q} = \{q_{s_1}, q_{s_2}, \ldots, q_{s_t}\}$ with an overlapped sliding window on I;
- **for** $i = 1$ *to* t **do**
  - $\boldsymbol{R}_{\mathrm{reward}}(s_i) = \mathcal{F}_{\mathrm{S}}(I_{\mathrm{object}}, q_{s_i}; \psi_{\mathrm{S}})$;
2. Estimate the object's facing orientation θ_{esti}
- $\theta_{\mathrm{esti}} = \mathcal{F}_{\mathrm{O}}(I_{\mathrm{object}}; \psi_{\mathrm{O}})$;
Path Planning
1. Find the optimal paths $\boldsymbol{P}$
- Obtain the cost map $\boldsymbol{R}_{\mathrm{cost}}$ according to Eq. (7);
- Build a directed graph G, whose edge weights W are set according to Eq. (12);
- Compute the shortest paths between v_{ini} and $\boldsymbol{v}_{\mathrm{edge}}$ on graph G, and sort them as $\boldsymbol{P} = (P_1, P_2, \ldots, P_m)$ based on their lengths $\boldsymbol{l} = (l_1, l_2, \ldots, l_m)$ in an ascending order.

where

$$
\begin{aligned}
(I): \quad & \|s_j - s_{\mathrm{ini}}\|_1 = d \text{ or } d+1, \\
& s_i \text{ and } s_j \text{ are spatially adjacent,} \\
(II): \quad & \|s_j - s_{\mathrm{ini}}\|_1 \neq d \text{ and } d+1, \\
& s_i \text{ and } s_j \text{ are spatially adjacent.}
\end{aligned}
$$

where θ_{s_j} is the relative position between s_{ini} and s_j. On graph G we can compute the shortest paths between the node of the initial position v_{ini} and the nodes of all the edge points $\boldsymbol{v}_{\mathrm{edge}} = (v_{\mathrm{edge}_1}, v_{\mathrm{edge}_2}, \ldots, v_{\mathrm{edge}_m})$ using DijsktraâŁ™s algorithm. These paths are sorted according to their lengths $\boldsymbol{l} = (l_1, l_2, \ldots, l_m)$ in an ascending order, represented as $\boldsymbol{P} = (P_1, P_2, \ldots, P_m)$. $\boldsymbol{P}$ are the top predicted paths for the visual path prediction task. The shortest one P_1 is exactly the solution of problem (11) on large scales. The whole path planning procedure is summarized in Algorithm 1.

IV. Experiments

A. Experimental Setup

We give the details on the networks, datasets, the comparison algorithms, and the evaluation metric in the following.

1) Networks: We build the CNNs based on the popular Caffe toolbox [49]. Fig. 4 and Fig. 5 respectively illustrate the network architectures of Spatial Matching Network and Orientation Network. Specifically, in the two figures 'conv' represents a convolution layer, 'fc' represents a fully connected layer, 'pool' represents a max-pooling layer, and 'LRN' represents a local response normalization layer. Numbers in the parentheses are respectively kernel size, number of outputs, and stride. All convolutional layers and fully connected layers are followed by ReLU activation function.

Fig. 6. The nine scenes of the first evaluation set, which is used in the evaluation of visual path prediction performance. The scenes include different parking lots, streets, and campuses. They are in a semi-birdseye view, and the videos are shot by cameras at different heights and locations to the grounds. The blue box denotes the scene used in the previous work [5]. In this paper, we make quantitative experiments on every scene.

Spatial Matching Network is trained for 2K iterations with a batch size of 256 and learning rate of 10^{-3}. The input images are uniformly resized to 256×256 and cropped to patches with the size of 227×227. Orientation Network is trained for 10K iterations with a batch size of 256 and learning rate of 10^{-5}. The input images of Orientation Network are directly resized to 227×227 without any cropping operation, because a part of an object image often cannot represent its exact facing orientation. The weights of both networks are initialized randomly for a fair comparison with the other algorithms.

2) Datasets: For the evaluation of visual path prediction task, we adapt a new large evaluation set. Raw data of this set come from VIRAT Video Dataset Release 2.0 [6]. VIRAT[1] is a public video dataset collected in multiple natural scenes, with people or vehicles performing actions with cluttered backgrounds. It contains a total of 8.5 hours HD videos from 11 different outdoor scenes, with a varieaty of camera viewpoints, and diverse types of activities which involve both human and vehicles. The ground truth object bounding boxes are manually annotated. Previous work [4], [5] on path prediction has also built their evaluation set with VIRAT, but with only a single or very few scenes. We do it in a different manner. We select 9 applicable scenes from the total 11 scenes to form our evaluation set. The 9 scenes are shared across the training and testing phase for all the methods. Fig. 6 shows the chosen scenes clearly, where the coloured box denotes the scene adopted by previous work. Among the total 195 videos, we use 152 videos for training, and 43 videos for testing. From the testing set, we automatically extract objects with at least 200 pixels in path length to form a total of 386 testing samples.

For evaluating the model's generalization capability, we also adapt a novel evaluation set. Raw data of this set come from KIT AIS Dataset,[2] which comprises aerial image sequences

[1] http://www.viratdata.org/
[2] http://www.ipf.kit.edu/english/code.php

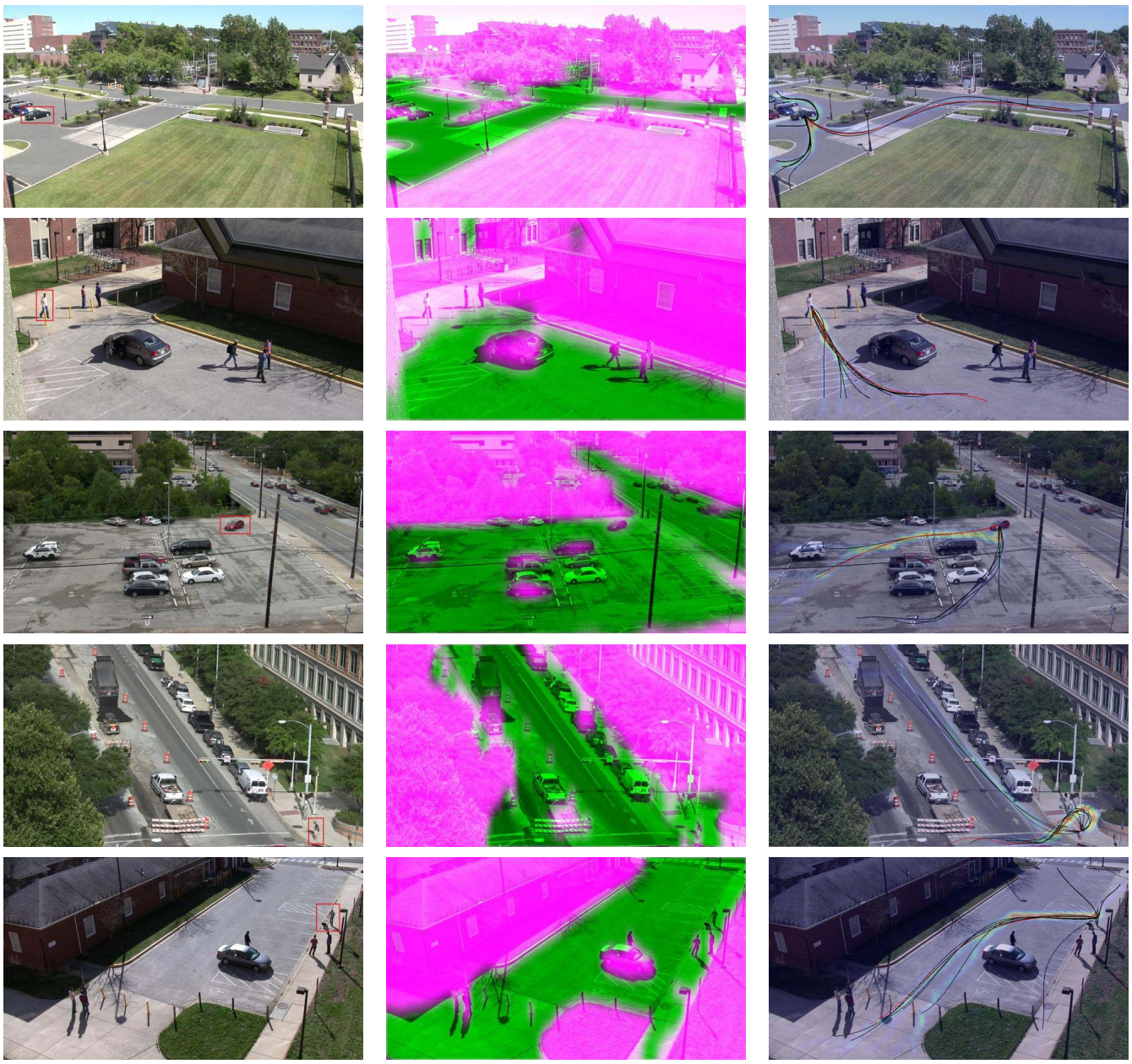


Fig. 7. Qualitative results generated by our algorithm. Each row represents a sample. The left column shows the input images. Red boxes on them denote the given objects. The middle column shows the generated reward maps. The right column shows predicted top-10 paths. Our framework can output discriminative reward maps and make accurate predictions on a diverse set of scenes. (a) Original Image. (b) Reward Map. (c) Predicted Paths.

with manually labeled trajectories of the visible vehicles. This dataset is entirely novel to visual path prediction task to our knowledge. It is relatively smaller than VIRAT, so we only use it for testing without training. From the total 9 scenes, we select 8 appropriate scenes and automatically extract 136 samples from the labeled trajectories to construct our evaluation set. The selected trajectories have larger distance between their starting and ending points.

3) Comparison Methods: There has been only a little work in the field of visual path prediction, so in this paper we compare our model with two methods:

1) Nearest neighbour searching with SIFT Flow warping [2], [3]. Identical to the implementation in Walker et al. [5], we use a Gist-matching approach [50] similar to Yuen et al. [3], and warp the labeled path of the nearest neighbour scene into the test scene using SIFT Flow [2].
2) The mid-level elements based temporal modeling [5]. It is the current state-of-the-art approach for visual path prediction task. We use their publicly available implementation code and train a model according to their hyper-parameters on the VIRAT dataset.

In our experiments, all the methods share the same training and testing sets. Because of the large size of the evaluation set and the high resolution of the scene images, images are uniformly downsampled into 640×360 for method (1) and (2).

(a) Original Image (b) Nearest Neighbour (c) Mid-level Elements (d) Ours

Fig. 8. Some qualitative comparison results. Each row represents a sample. Column (a) shows the input images with red boxes denoting the objects. Column (b), (c) and (d) respectively show the predicted paths generated by different approaches: NN [3], MLE [5] and ours. Our approach has better performance in most of the scenes. The predictions generated by our approach are closer to the common sense. (a) Original Image. (b) Nearest Neighbour. (c) Mid-level Elements . (d) Ours.

Evaluation metric: We employ the commonly used [4], [5] metric: modified Hausdorff distance (MHD) [51] as the metric for the distance between two paths. The MHD allows for finding the best local point correspondence and it is robust to outlier points. Since there may be several reasonable future paths for a given object in the path prediction task, we employ three indicators for comprehensive comparison: (1) top-1, (2) top-5 average and (3) top-10 average. The top-N average means that for a method on a certain testing sample, we first compute the MHDs between the ground truth path and the top-N paths predicted by this method, and then take an average of these distances as the method's performance on this sample.

B. Path Prediction

1) Qualitative: Fig. 7 shows some qualitative results generated by our method on different scenes of the evaluation set. Each row represents a sample. The left column is the input images, in which we mark the given objects with red boxes. The middle column shows the reward maps generated by our algorithm, in which those green areas are accessible (high reward) while pink areas are obstacles (low reward). We can see that the grass, tree and house in the maps are detected as low reward, while the road and parking lot are of high reward. Notice the fourth and fifth maps, where the sidewalks is recognized as high reward area for the corresponding pedestrians. The right column shows the predicted paths for corresponding input images, where the red lines represent the top-1 predictions and the black lines represent the other top-10 predictions. Visually, the predicted paths are close to our human's inference. Notice how the predicted paths avoid other objects (cars, pedestrians) or obstacles (grass, trees, buildings) and go along the road. Furthermore, we can see

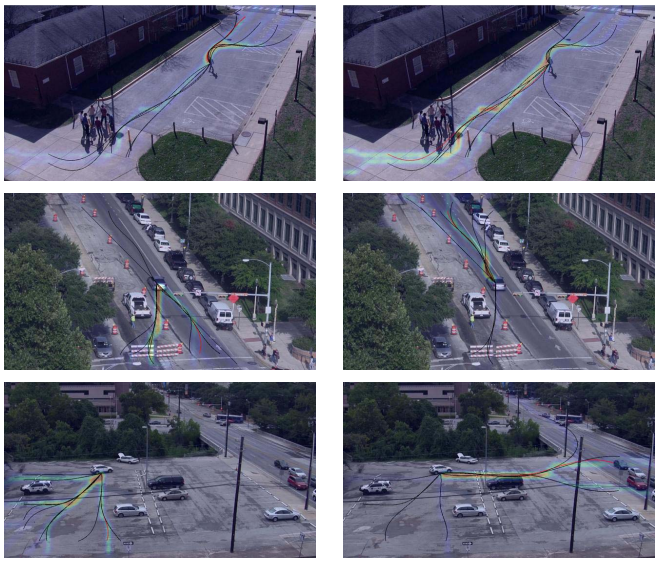
TABLE II
QUANTITATIVE RESULTS FOR VISUAL PATH PREDICTION TASK

#Scene	A	B	C	D	E	F	G	H	I	Total
Samples	53	26	21	36	26	41	44	46	93	386
Top-1										
NN [3]	19.57	25.47	16.15	18.19	24.78	29.16	23.16	14.84	12.31	19.12
MLE [5]	17.63	17.55	24.12	15.06	13.72	19.27	20.47	16.57	18.13	17.97
Rewards (ours)	20.83	20.43	**13.72**	19.01	21.67	17.13	16.06	16.03	13.30	16.98
Ours	**13.37**	**13.81**	16.34	**13.29**	**12.95**	**10.99**	**10.41**	**12.24**	**10.42**	**12.09**
Top-5 Average										
NN [3]	22.34	25.75	16.35	17.10	28.89	31.09	22.86	14.65	12.72	19.95
MLE [5]	17.41	17.49	22.77	15.03	13.75	19.00	20.22	16.51	18.09	17.78
Rewards (ours)	18.87	18.80	**13.68**	17.94	20.55	17.13	17.06	13.50	11.73	15.86
Ours	**13.21**	**13.43**	15.71	**13.17**	**12.78**	**11.71**	**10.22**	**11.60**	**10.57**	**12.00**
Top-10 Average										
NN [3]	22.81	26.31	15.86	16.19	28.31	31.38	23.37	15.88	14.42	20.55
MLE [5]	17.04	16.85	20.44	15.92	13.49	18.19	20.16	15.59	16.68	17.09
Rewards (ours)	17.62	17.43	**14.97**	17.57	19.65	16.16	17.27	12.24	11.16	15.20
Ours	**13.44**	**12.89**	15.59	**12.96**	**12.55**	**12.11**	**11.79**	**11.14**	**10.69**	**12.15**

that our framework is able to make correct prediction of the destination. In the third image, the red car will be parked in the square. In the fourth image, the person probably wants to walk across the street. A correct destination estimation will largely improve the performance of path planning.

Besides, we make qualitative comparison among different methods as shown in Fig. 8. We select various scenes and objects for testing. Each row represents a testing sample. Column (a) is the input images. Column (b) and (c) show the predicted paths generated by the comparison methods NN [3] and MLE [5]. Our predictions are shown in column (d). We can see that the NN approach does not give effective performance. It is nearly betting that there have been appropriate paths stored in database. The last image of column (b) shows this clearly where most trajectories of the nearest samples in database are distributed along the road. It is not effective in practical use. MLE approach produces comparatively better performance. However, limited to its visual representation ability on diverse scenes, the predicted paths do not appear reasonable. In the fifth image of column (c), the man would attempt to climb over the fence in front of him. In the sixth image of column (c), the car would attempt to drive across the trees. On most scenes shown in Fig. 8, our approach makes reasonable predictions that is consistent with the common sense. Furthermore, our method infers a variety of appropriate optional paths as shown in the first, third and sixth image of column (d).

Fig. 9 shows a qualitative comparison between our complete framework and our rewards only method, in which we only use our reward map for prediction without the help of Orientation Network. Column (a) and (b) are respectively the results of the rewards only method and the complete framework. As a result of lack of facing orientation estimation, most of the predicted paths in column (a) are not reasonable as they do not follow the current facing orientation of the objects. For example in the third image of column (a), the white car is facing right,



(a) Rewards Only (b) Rewards + Orientation

Fig. 9. Comparison between (a) our path planning scheme using only rewards and (b) the complete framework. Orientation Network estimates the facing orientation of the object. With its help, the path planning scheme is able to rectify the paths according to the object's current moving direction, consequently improving the final performance.

but its predicted paths tend to go left and down. Instead, the third image of column (b) shows more proper predictions with the help of orientation estimation. This indicates that the facing orientation estimated by our Orientation Network offers much help to the complete framework.

2) Quantitative: For quantitative evaluation, we compare our method with the competing methods on all scenes in the evaluation set. Table II shows the results on each scene with a total of 386 testing samples. Our method outperforms the comparison methods by large margins on all of the scenes. Compared with the state-of-the-art methods on the

TABLE III
QUANTITATIVE RESULTS FOR GENERALIZATION CAPABILITY EVALUATION

#Scene	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Total
Samples	5	36	5	6	45	22	6	11	136
					Top-1				
NN [3]	**14.54**	18.75	52.77	40.67	51.58	43.38	17.67	15.54	35.35
MLE [5]	24.50	20.12	**32.20**	25.66	75.45	42.70	16.19	14.05	42.26
Rewards (ours)	22.42	**8.57**	56.96	**23.71**	23.97	37.52	**15.42**	10.73	21.78
Ours	18.23	8.99	49.24	**23.71**	**21.53**	**34.19**	19.77	**10.14**	**20.25**
					Top-5 Average				
NN [3]	17.28	21.91	50.80	34.78	64.64	44.38	16.04	16.71	40.46
MLE [5]	24.28	18.76	**32.20**	25.73	75.33	42.61	16.21	14.32	41.87
Rewards (ours)	18.90	6.70	55.31	14.90	**19.46**	34.79	**12.16**	**8.82**	18.48
Ours	**16.29**	**6.63**	48.76	**12.82**	19.70	**32.12**	14.97	9.76	**17.88**
					Top-10 Average				
NN [3]	17.36	20.31	53.27	34.82	61.38	46.13	19.29	15.75	39.41
MLE [5]	22.92	16.22	**43.71**	25.66	75.39	42.59	16.14	12.92	41.47
Rewards (ours)	17.68	6.56	47.84	**13.93**	**21.24**	30.02	**11.78**	**9.65**	17.94
Ours	**15.78**	**6.37**	43.82	14.80	21.47	**27.55**	12.45	10.01	**17.45**

entire evaluation set, our method makes 33%, 33%, 29% improvement respectively under the top-1, top-5 average and top-10 average metric. For each scene, the improvement varies from 6% to 49% under the top-1 metrics. In addition, the results of our method show a relatively smaller inter-scene variance than those of the other methods. The smaller inter-scene variance means that our method can better fit different complex scenes and is more robust for scene change.

The third row in every sheet shows the results of our rewards only method. It shows 6%, 11%, 11% improvement over the other comparison methods under the three metrics, respectively, demonstrating the value of our Spatial Matching Network. However, the error of the rewards only method is larger than that of our complete framework on most of the scenes. This indicates the value of the temporal context which is modeled by our Orientation Network.

C. Generalization Capability

We have evaluated the visual path prediction performance, where the training set and testing set own the same scenes. However, a robust path prediction framework ought to perform well on novel scenes and objects. In this experiment, we evaluate the generalization capability of the methods on the second evaluation set described in subsection IV-A. We simply test the models on this evaluation set without retraining the models. Parameters of the models remain the same as those in the path prediction experiment of subsection IV-B.

Table III documents the quantitative results of the generalization capability evaluation. Our method respectively makes 43%, 56%, 56% improvement over the comparison methods under the top-1, top-5 average and top-10 average metrics on the entire evaluation set. These improvements are larger than those in the primary experiments as Table II, showing that our method has a better generalization ability than the existing work. Most of the absolute MHD values in Table III have increased, while meantime the inter-scene variance has also increased. This is in line with our intuition that the models have never seen the testing samples in this experiment. In addition, different from the other two methods, the top-5 average metric and top-10 average metric of our method in Table III show some improvement over the top-1 metric on most of the scenes. To some extent this indicates that our method can explore more proper underlying paths on unknown scenes than the other methods. Compared with the rewards only method, the complete framework performs better on half of the eight scenes and a little worse on the entire dataset. This indicates that in this experiment the Orientation Network does not help much. It is possibly due to the inadequate training samples.

V. CONCLUSION

In this paper we have proposed a deep learning framework to address the visual path prediction problem. The proposed deep learning framework simultaneously performs deep feature learning for visual representation in conjunction with spatio-temporal context modeling, which largely enhances the scene understanding capability. In addition, we have presented a unified path planning scheme to infer the future paths on the basis of the analytic results returned by our context models. For comprehensively evaluating the model's performance on the visual path prediction task, we have constructed two large benchmark datasets from the adaptation of the existing video tracking datasets. The experimental results demonstrate the effectiveness and robustness of our approach in comparison with the state-of-the-art literature.

REFERENCES

[1] J. Hawkins and S. Blakeslee, *On Intellligence*. New York, NY, USA: Macmillan, 2007.
[2] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "Sift flow: Dense correspondence across different scenes," in *Proc. ECCV*, pp. 28–42, Oct. 2008.

[3] J. Yuen and A. Torralba, "A data-driven approach for event prediction," in *Proc. ECCV*, 2010, pp. 707–720.

[4] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. ECCV*, 2012, pp. 201–214.

[5] J. Walker, A. Gupta, and M. Hebert, "Patch to the future: Unsupervised visual prediction," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 3302–3309.

[6] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3153–3160.

[7] B. Jutzi. (2016). *Kit-ipf-Software and Datasets*. [Online]. Available: http://www.ipf.kit.edu/english/code.php

[8] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification," in *Proc. ECCV*, Sep. 2010, pp. 57–69.

[9] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 702–709.

[10] J. Yu, D. Tao, Y. Rui, and J. Cheng, "Pairwise constraints based multiview features fusion for scene classification," *Pattern Recognit.*, vol. 46, no. 2, pp. 483–496, 2013.

[11] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 923–930.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 1725–1732.

[13] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Proc. IEEE ICCV Workshops*, Nov. 2011, pp. 601–608.

[14] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 580–587.

[16] L. Zhang, Y. Yang, Y. Gao, Y. Yu, C. Wang, and X. Li, "A probabilistic associative model for segmenting weakly supervised images," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4150–4159, Sep. 2014.

[17] Q. Li, X. Chen, Y. Song, Y. Zhang, X. Jin, and Q. Zhao, "Geodesic propagation for semantic labeling," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4812–4825, Nov. 2014.

[18] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D depth reconstruction from a single still image," *Int. J. Comput. Vis.*, vol. 76, no. 1, pp. 53–69, 2008.

[19] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 1253–1260.

[20] J. Lin, X. Ji, W. Xu, and Q. Dai, "Absolute depth estimation from a single defocused image," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4545–4550, Nov. 2013.

[21] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.

[22] C. Xu, T. Liu, D. Tao, and C. Xu, "Local rademacher complexity for multi-label learning," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1495–1507, Mar. 2016.

[23] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 980–993, Mar. 2015.

[24] H. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, Jan. 2016.

[25] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, "Activity recognition using a mixture of vector fields," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1712–1725, May 2013.

[26] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in *Proc. ECCV*, Sep. 2014, pp. 689–704.

[27] H. Wang, C. Yuan, W. Hu, H. Ling, W. Yang, and C. Sun, "Action recognition using nonnegative action component representation and sparse basis selection," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 570–581, Feb. 2014.

[28] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 810–822, Feb. 2014.

[29] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[30] X. Li, T. Liu, J. Deng, and D. Tao, "Video face editing using temporal-spatial-smooth warping," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 3, p. 32, 2016.

[31] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from Web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, Sep. 2012.

[32] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev, "Semantic model vectors for complex video event recognition," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 88–101, Jan. 2012.

[33] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," *Int. J. Multimedia Inf. Retr.*, vol. 2, no. 2, pp. 73–101, Jun. 2013.

[34] D. F. Fouhey and C. L. Zitnick, "Predicting object dynamics in scenes," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 2027–2034.

[35] C. G. Keller, C. Hermes, and D. M. Gavrila, "Will the pedestrian cross? probabilistic path prediction based on learned motion features," in *Proc. Pattern Recognit.*, 2011, pp. 386–395.

[36] S. D. Tran and L. S. Davis, "Event modeling and recognition using Markov logic networks," in *Proc. ECCV*, 2008, pp. 610–623.

[37] C. H. Lampert, "Predicting the future behavior of a time-varying probability distribution," in *Proc. IEEE Conf. CVPR*, 2015, pp. 942–950.

[38] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *Proc. ECCV*, 2010, pp. 57–70.

[39] D. Munoz, J. A. Bagnell, and M. Hebert, "Co-inference for multi-modal scene analysis," in *Proc. ECCV*, 2012, pp. 668–681.

[40] S. Singh, A. Gupta, and A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. ECCV*, 2012, pp. 73–86.

[41] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. NIPS*, 2012, pp. 1097–1105.

[43] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 3642–3649.

[44] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with lstm recurrent neural networks," in *Proc. IEEE Conf. CVPR*, Jun. 2015, pp. 3547–3555.

[45] J. Li, X. Mei, and D. Prokhorov, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. Learn.Syst.*, 2006. [Online.] Available: http://ieeexplore.ieee.org/document/7407673/.

[46] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. CVPR*, 2015, pp. 2625–2634.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] J. Walker, A. Gupta, and M. Hebert, "Dense optical flow prediction from a static image," in *Proc. IEEE Conf. CVPR*, Dec. 2015, pp. 2443–2451.

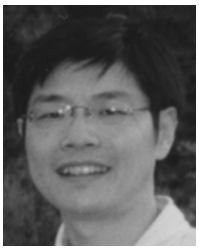
[49] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Multimedia*, 2014, pp. 675–678.

[50] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[51] M.-P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proc. 12th IAPR Int. Conf. Pattern Recognit. (ICPR)*, vol. 1. Oct. 1994, pp. 566–568.

Siyu Huang received the bachelor's degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2014. He is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University. His advisors are Prof. Z. Zhang and Prof. X. li. His current research interests are primarily in computer vision, machine learning, and deep learning.

Xi Li received the Ph.D. degree from the National Laboratory of Pattern Recognition, Chinsese Academy of Sciences, Beijing, China, in 2009. From 2009 to 2010, he was a Post-Doctoral Researcher with CNRS Telecomd ParisTech, France. He was a Senior Researcher with the University of Adelaide, Australia. He is currently a Full Professor with Zhejiang University, China. His research interests include visual tracking, motion analysis, face recognition, web data mining, and image and video retrieval.



Zhongfei Zhang received the B.S. degree (Hons.) in electronics engineering and the M.S. degree in information sciences from Zhejiang University, China, and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, MA, USA. He is a QiuShi Chaired Professor with Zhejiang University, where he directs the Data Science and Engineering Research Center. He is on leave from the State University of New York, Binghamton, USA, where he is a Professor with the Computer Science Department and directs the Multimedia Research Laboratory, the Computer Science Department. His research interests include knowledge discovery from multimedia data and relational data, multimedia information indexing and retrieval, and computer vision and pattern recognition.

Zhouzhou He received the bachelor's degree in electrical and information engineering from Zhejiang University, in 2009. He is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His advisors are Prof. Z. Zhang and Prof. X. Li. His current research interests are primarily in data mining and computer vision, especially e-commerce business model mining, face age estimation, and deep learning.

Fei Wu received the B.S. degree from Lanzhou University, Lanzhou, Gansu, China, the M.S. degree from Macao University, Taipa, Macau, and the Ph.D. degree from Zhejiang University, Hangzhou, China. He was a Visiting Scholar with the Prof. B. Yu's Group, University of California at Berkeley, Berkeley, CA, USA, from 2009 to 2010. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. His current research interests include multimedia retrieval, sparse representation, and machine learning.

Wei Liu received the M.Phil. and Ph.D. degrees in EECS from Columbia University, New York, NY, USA, in 2012. He has been a Research Scientist with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, since 2012. He is currently a Computer Vision Director with the Tencent AI Laboratory, Shenzhen, China. He has authored over 100 peer-reviewed journal and conference papers, including the Proceedings of the IEEE, the IEEE TPAMI, the IEEE TIP, the IEEE TKDE, NIPS, ICML, KDD, CVPR, ICCV, ECCV, IJCAI, AAAI, UAI, SIGIR, and SIGCHI. He has been involved in research and development in the fields of computer vision, machine learning, data mining, and information retrieval. He is a recipient of the 2011–2012 Facebook Fellowship, the 2013 Jury Award for the best thesis of Columbia University, the 2014 CVPR Young Researcher Support Award, and the 2016 SIGIR Best Paper Award Honorable Mention.

Jinhui Tang received the B.E. and Ph.D. degrees from the University of Science and Technology of China, in 2003 and 2008, respectively. From 2008 to 2010, he was a Research Fellow with the School of Computing, National University of Singapore. He is a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He has authored over 100 journal and conference papers in these areas. His current research interests include large-scale multimedia search. He is a recipient of the ACM China Rising Star Award, and a co-recipient of the Best Student Paper Award in MMM 2016 and best paper awards in ACM MM 2007, PCM 2011, and ICIMCS 2011.

Yueting Zhuang received the B.S., M.S., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1986, 1989, and 1998, respectively. From 1997 to 1998, he was a Visitor with the Department of Computer Science and Beckman Institute, University of Illinois at Urbana–Champaign, Champaign, IL, USA. He is currently a Professor and the Dean of the College of Computer Science, Zhejiang University. His current research interests include multimedia databases, artificial intelligence, and video-based animation.