Cycling-Net: A Deep Learning Approach to Predicting Cyclist Behaviors from Geo-Referenced Egocentric Video Data

Yichen Ding[a], Xun Zhou[a], Han Bao[a], Yanhua Li[b], Cara Hamann[a], Steven Spears[a], Zhuoning Yuan[a]
[a]University of Iowa, Iowa, [b]Worcester Polytechnic Institute, Massachusetts
{yichen-ding,xun-zhou,han-bao,cara-hamann,steven-spears,zhuoning-yuan}@uiowa.edu,yli15@wpi.edu

ABSTRACT

Cycling, as a green transportation mode, provides an environmentally friendly transportation choice for short-distance traveling. However, cyclists are also getting involved in fatal accidents more frequently in recent years. Thus, understanding and modeling their road behaviors is crucial in helping improving road safety laws and infrastructures. Traditionally, people understand road user behavior using either purely spatial trajectory data, or videos from fixed surveillance camera through tracking or predicting their paths. However, these data only cover limited areas and do not provide information from the cyclist's field of view. In this paper, we take advantage of geo-referenced egocentric video data collected from the handlebar cameras of cyclists to learn how to predict their behaviors. This approach is technically more challenging, because both the observer and objects in the scene might be moving, and there are strong temporal dependencies in both the behaviors of cyclists and the video scenes. We propose Cycling-Net, a novel deep learning model that tracks different types of objects in consecutive scenes and learns the relationship between the movement of these objects and the behavior of the cyclist. Experiment results on a naturalistic trip dataset show the Cycling-Net is effective in behavior prediction and outperforms a baseline model.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; **Data mining**.

KEYWORDS

Cyclist Behavior, Egocentric Video, Deep Learning

ACM Reference Format:
Yichen Ding[a], Xun Zhou[a], Han Bao[a], Yanhua Li[b], Cara Hamann[a], Steven Spears[a], Zhuoning Yuan[a]. 2020. Cycling-Net: A Deep Learning Approach to Predicting Cyclist Behaviors from Geo-Referenced Egocentric Video Data. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397536.3422258

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11…$15.00
https://doi.org/10.1145/3397536.3422258

1 INTRODUCTION

Cycling, as a green transportation mode, has become increasingly popular, as it is relieving environment pollution compared to motor vehicles and at the same time benefits psychological and physical health. Leading bike-sharing companies are promoting people to choose bicycle as their primary transportation tool. Meanwhile, traffic accidents involving cyclists are also increasing in recent years. Even wearing protective cycling gears, cyclists are still vulnerable on the road and may easily get injured in crashes.

To protect cyclists and promote green transportation, different approaches have been explored. For example, many cities designed biking corridors and implemented traffic regulations to protect cyclists [20]. At the same time, researchers in public health seek to understand the key drivers of bicycle crashes. In particular, a key question is to understand the behaviors of cyclists when facing various road conditions. For example, what are the driving factors for a cyclist to change the route? Why did the cyclist decide to slow down? The answers to these questions provide important scientific evidence for road infrastructure improvement and regulation adjustments.

To facilitate road safety research, rich data have been collected from volunteers, such as egocentric (i.e., first-person view) data from helmet cameras and GPS trajectories of cyclists. Traditionally, road safety researchers observe these data to code them and extract useful information, which is not scalable to larger datasets. Furthermore, traditional analysis of the risk factors are based on human judgment of the scene, therefore introducing potential biases.

In this paper, we make the first attempt to use deep learning to study the behaviors of cyclists from their geo-referenced egocentric video data. In particular, we hope to learn a predictive model that uses the past behaviors and video contents along a cyclist's trajectory to predict his/her next behavior (e.g., turning, break, accelerating). Building such a predictive model is **technically challenging**. First of all, behaviors of cyclists are influenced by different factors, including road conditions, vehicles nearby, etc. These are objects with semantic meanings. Simply using image pixels as input features will not help connecting these factors with the cyclists' behaviors. Second, the scenes in the egocentric data are changing more rapidly than those in static cameras because the orientation of the camera might change. Tracking objects in the scenes are harder. Finally, there are strong temporal dependencies in both the video contents and the behaviors of a cyclist along the path. It is non-trivial to model these dependencies in the learning process.

Despite the rapid development of deep learning techniques in recent years, state-of-the-art behavior and trajectory prediction methods still could not solve our problem. Many recent works explored deep learning solutions to the driver or pedestrian trajectory

prediction problem [1, 25, 29]. Given a partial GPS trajectory, these methods aim at predicting either the destination or the next location along the trajectory. Other works [15, 31] focused on driver behavior modeling or traffic accident prediction using deep learning methods. However, these works only use spatial trajectories but without any visual input (e.g., video, image), therefore could not solve our problem. Works in computer vision [12, 26, 30] have also focused on pedestrians or cyclist trajectory prediction. However, most of these techniques use videos from fixed cameras and predict the next "position" of the pedestrian or cyclist in the image scene. Our problem is very different from these prior works. In our data, the camera collecting data is also moving, while in the literature the majority of the methods use videos from fixed cameras. Another difference is that we aim to predict the behavior of the observer itself rather than objects in the scene.

In this paper, we propose Cycling-Net, a deep learning solution to address the problem. Cycling-Net extracts features such as bearing and distance between consecutive GPS records from the trajectories of the cyclists. It also identifies and continuously tracks the movement of the most relevant objects in the video. The moving statuses of these objects are fed to a deep neural network with parallel Long Short-Term Memory (LSTM) components, whose outputs are assembled to generate the final prediction. We study the performance of Cycling-Net on a naturalistic trip dataset collected from human volunteers and found that Cycling-Net outperforms a baseline solution.

Our main contributions of this paper are summarized as follows:

- We make the first attempt to use the deep learning method to solve the cyclist behavior prediction problem based on geo-referenced egocentric data. We formulate the problem as a prediction problem and derive important features and labels from raw data.
- We propose Cycling-Net, a deep learning solution to predict the next moves of the cyclist through continuous identification and tracking of moving objects in the scene. We give two designs of Cycling-Net with single vs. parallel LSTM structures.
- We conduct extensive experiments on a naturalistic dataset collected from real human volunteers to evaluate the proposed Cycling-Net model. Results suggest that the proposed solution outperforms the baseline in terms of accuracy, F-measure, and AUC. We also identify the impact of feature selection for Cycling-Net, with a trade-off between the two design decisions.

The rest of this paper is organized as follows. Section 2 introduces our bicycling data and presents how we model cyclist behaviors. In Section 3, we present our proposed Cycling-Net framework. Section 4 presents comprehensive evaluation results. Section 5 discusses related work. Finally, we conclude the paper in Section 6.

2 CYCLIST BEHAVIOR MODELING

This section introduces the data used in our problem and the preprocessing steps to derive necessary labels. Then we define concepts used in this paper, followed by a formal problem statement.

2.1 Data Description and Pre-Processing

We have recruited 40 adult (18+) participants to record geo-referenced egocentric videos of cycling trips. Each volunteer is equipped with a camera mounted on the handlebar of the bike, which can record both first-perspective video data and GPS trajectories at a sampling rate of 1 Hz (1 point per second). The volunteers take trips during working and leisure hours in the Iowa City area. In all, we collected 352 trips, with 81 hours of contents and totally 861 miles covered. The entire data is collected over a four-month period, from April to August 2018. The videos are at 1080p resolution with a timestamp associated with each frame.

We preprocess the data to match the videos and the corresponding GPS points. Due to different sampling rates, video data are down-sampled (i.e., only 1 frame per second is taken) to match with the GPS points based on the timestamps of each trip. Figure 1 demonstrates an example of a trip with three consecutive frames from the egocentric video. Each of these frames is matching one of the red GPS points in the trajectory on the left.

In addition, we trim the data to ensure data quality. Often we observed dense GPS points near the origin or destination of a trip but without any video being recorded. This is because the GPS recording is already/still working while the camera has been turned off by the volunteer before or after the trip. In order to avoid the impact of these invalid trajectories on cyclist behavior understanding, we only consider the actual cycling trajectories by excluding both the initial and the ending parts. For each trip, we assume it starts from the first point at least d_{start} meters away from the origin point and ends at the last point no less than d_{end} meters away from the destination point. As shown in Figure 1a, we study the trajectories out of the red and orange circles. In this paper, we set $d_{start} = 8$ and $d_{end} = 5$. The valid trajectory travels from the new origin point with a red flag to the new destination point with an orange flag. Note that, when collecting video data, each cyclist is equipped with a camera mounted on the handlebar of the bike. Therefore, the videos are mostly stable. Moreover, we remove the low-quality frames and rotate the distorted frames to guarantee the quality of the video.

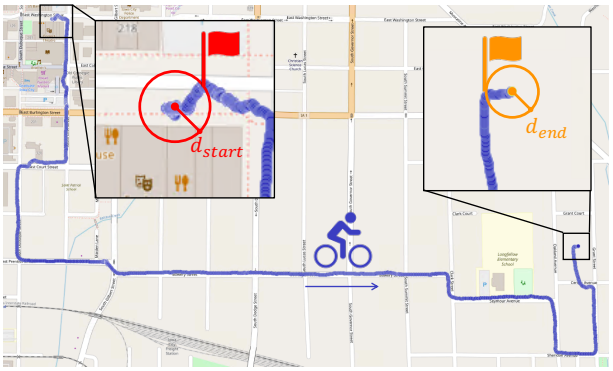
2.2 Cyclist Behavior Modeling from GPS Data

Given the above dataset, we need to define the actions of cyclists based on their movement before predicting them from the egocentric data collected. A few key concepts are introduced here.

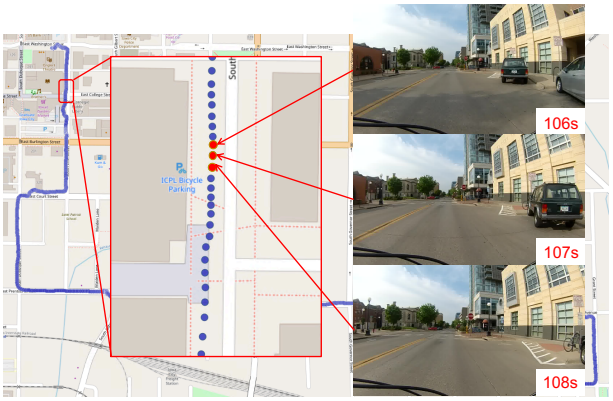
In this work, we define actions related to (i) direction of movement, and (ii) speed of movement. Specifically, actions related to the direction of movement include moving straight forward, left turn, and right turn. Actions related to the speed of movement include constant speed, decelerating, accelerating, walking (very slow but non-zero speed), and full stop.

First of all, we derive the distance and bearing between two consecutive GPS points and use them to define the above behaviors. Given two consecutive GPS points on the same trip, $p_1 = (lon_1, lat_1)$ and $p_2 = (lon_2, lat_2)$, the distance and bearing can be calculated as follows:

Distance: According to haversine formula, we use the great-circle distance between two GPS points, that is, the shortest distance over the earth's surface, and the calculation follows the equations:



(a) GPS Trajectory



(b) Geo-referenced Egocentric Images

Figure 1: Example of a trip trajectory with three first-person images from the egocentric video.

$$a = \sin^2(\frac{\Delta lat}{2}) + \cos(lat_1) * \cos(lat_2) * \sin^2(\frac{\Delta lon}{2}) \quad (1)$$

$$c = 2 * atan2\left(\sqrt{a}, \sqrt{1-a}\right) \quad (2)$$

$$d(p_1, p_2) = R * c \quad (3)$$

where lon, lat are longitude and latitude in degrees respectively, $\Delta lon = lon_2 - lon_1$, $\Delta lat = lat_2 - lat_1$, and R is earth radius (mean radius = 6,371km). Thus, $d(p_1, p_2)$ returns the distance between GPS points (lon_1, lat_1) and (lon_2, lat_2).

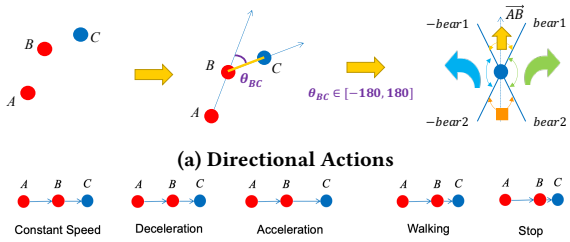
Bearing (Angle): Based on the initial bearing (also called forward azimuth), we use the compass bearing of two consecutive GPS points. The calculation follows the equations:

$$b = atan2(\sin(\Delta lon) * \cos(lat_2), \cos(lat_1) * \sin(lat_2) - \sin(lat_1) * \cos(lat_2) * \cos(\Delta lon)) \quad (4)$$

$$\theta(p_1, p_2) = (b + 360)\%360 \quad (5)$$

where $p_1 = (lon_1, lat_1)$ is the start point, $p_2 = (lon_2, lat_2)$ is the end point, and $\Delta lon = lon_2 - lon_1$.

Based on the distance and bearing, we are able to transform each trajectory into a sequence of the cyclist actions. As mentioned previously, we consider two types of cyclists' actions: directional actions and speed-related actions. As shown in Figure 2, given three consecutive GPS points A, B, and C, we label the cyclist actions at point C as follows.



(a) Directional Actions

(b) Speed-related Actions

Figure 2: Illustration of labeling the cyclist behavior at point C.

Directional Actions (a_{dir}): We define three types of directional actions, namely, turning left, turning right, and moving straight forward. According to the bearing θ_{BC} from the previous direction $\overrightarrow{AB}$, the cyclist is considered to keep moving forward if the bearing changes very little, that is, $\theta_{BC} \in [-bear_1, bear_1]$. Otherwise, the cyclist decides to change the direction with a large bearing value. If $\theta_{BC} \in [bear_1, bear_2]$ (or $\theta_{BC} \in [-bear_2, -bear_1]$), he/her turns right (or left) at point C. In addition, if $\theta_{BC} < -bear_2$ or $\theta_{BC} > bear_2$, the cyclist is considered to stop at point C. The reason is that the GPS signal might be disturbed, when the cyclist is waiting at the crossroad. Moreover, a cyclist will not move backward within a second. In this paper, we set $bear_1 = 20$ and $bear_2 = 100$.

Speed-related Actions (a_{speed}): We define four types of speed-related actions, namely, accelerating, decelerating, walking, and full stop. According to the distance $|BC|$ from the previous GPS point B, we consider the cyclist is walking (or stops) at point C with a small distance value, that is, $|BC| \in [1, 2)$ (or $|BC| < 1$ meter). Compared with distance $|AB|$, we assume the cyclist decelerates (or accelerates) with a significant distance change, that is, $|BC| < (1 - \alpha_1)|AC|$ (or $|BC| > (1 + \alpha_2)|AC|$). Otherwise, the cyclist moves with constant speed, which means $((1-\alpha_1)|AC| < |BC| < (1+\alpha_2)|AC|$. Note that, if $|BC| > 10$ meters, we assume point C is invalid, because it is hard for a normal cyclist to ride more than 10 meters within a second. In this paper, we set $\alpha_1 = 0.05$ and $\alpha_2 = 0.15$.

Finally, we use a pair of two actions (a_{dir}, a_{speed}) to describe the cyclist behavior at each GPS point. There are a number of different combinations. For example, the combination of acceleration and moving forward shows that the user speeds up and keeps moving straight compared to the locations at previous timestamps.

2.3 The Cyclist Behavior Prediction Problem

Based on the behavior modeling above, we can formulate our problem into a behavior prediction problem. First, we introduce three concepts.

Definition 1. Geo-referenced Egocentric Frame. A georeferenced Egocentric frame f is a tuple <*TripID*, t, I, lat, lon>, where *TripID* is an integer label of the trip where this frame belongs to, t is an integer timestamp counter of this image from the beginning of the trip, $t \in [1, \text{length}(TripID)]$. I is a three-dimensional image tensor (height, widths, and channels) representing the video captured by the camera along *TripID* at time t. Typically, the data has three channels (R,G,B). lat and lon are double-precision floating

point numbers representing the latitude and longitude coordinates of the location when the frame was captured.

Definition 2. Egocentric Video Trajectory. An egocentric video trajectory Tr is a sequence of geo-referenced egocentric frame with the same *TripID*, i.e., $Tr = \{f_1, f_2, ..., f_n\}$, where $t_1 < t_2 < ... < t_n, n = |Tr|$. The frames contain consecutive images observed along the route and movement information along the path of the observer. Here we assume that the frames are sampled at regular timestamps, i.e., $t_i = 1$.

Definition 3. Cyclist Behavior. Given an egocentric video trajectory Tr, a sequence of behavior labels $Y(Tr) = \{y_3, y_4, ...y_n\}$ can be derived, where $y_i = (a_{dir,i}, a_{speed,i})$ is the behavior label associated with frame i. Since we need three consecutive points to infer the behavior at one GPS point, we only have $n-3$ behavior labels for n GPS points.

In this paper, we combine different combinations of actions into two classes of behaviors: moving forward with constant speed ($y = 0$) and all other behaviors including speed or direction changes ($y = 1$), and formulate the problem as a binary classification problem. We predict if the cyclist is going to continue biking forward at a constant speed, or change the current biking status. Predicting multi-class behaviors based on more combinations of actions will be explored in our future work.

Problem Statement. Given a segment of egocentric video trip from time $t-T+1$ to t (T frames), we aim at predicting the behavior label $\hat{y}_{t+1}$ for the same trip at time $t+1$. Formally, the problem is stated as follows:

- **Input**:

  $$\{lon_{t-T+1}, lat_{t-T+1}, I_{t-T+1}, \ldots, lon_t, lat_t, I_t\}$$

  where I_i indicates the image matrix at timestamp $i \in [t-T+1, t-T+2, \ldots, t]$ and $\{(lon_{t-T+1}, lat_{t-T+1}), (lon_{t-T+2}, lat_{t-T+2}), \ldots, (lon_t, lat_t)\}$ is the GPS trajectory at the time period $[t-T+1, t-T+2, \ldots, t]$.
- **Output**:

  $$\hat{y}_{t+1} \in \{0, 1\}$$

  is the predicted behavior label of the cyclist at timestamp $t+1$.
- **Objective**:

  $$\min |y_{t+1}, \hat{y}_{t+1}|$$

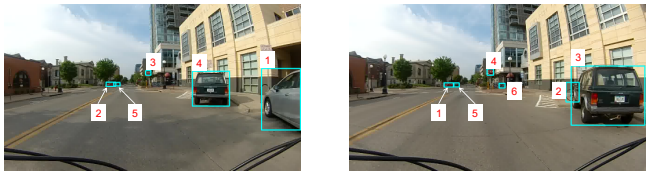
  is to minimize the difference between the ground-truth behavior label of the cyclist y_{t+1} and the predicted behavior label $\hat{y}_{t+1}$ at timestamp $t+1$.

3 CYCLIST BEHAVIOR PREDICTION

In this section, we first introduce how to extract different features from egocentric video trajectory, including mobility features and object features. Then, we present the framework of our Cycling-Net incorporating with combined features.

3.1 Feature Extraction

First, we extract different features from egocentric video trips for the prediction problem. We use two types of features, namely, mobility features (derived from GPS data) and image features (derived from video frames).



(a) Geo-referenced Egocentric Image at 106 **(b) Geo-referenced Egocentric Image at 107**

Figure 3: Example of object features from the geo-referenced egocentric images.

Table 1: Example of Object Features

ID	Category	Confidence Score (Overall Score)	Bounding Box (x1, y1, x2, y2)
1	Car	0.9975 (0.9975)	(1109, 279, 1279, 543)
2	Car	0.8928 (0.8928)	(439, 337, 482, 356)
3	Stop Sign	0.8489 (0.6791)	(611, 288, 634, 307)
4	Car	0.7742 (0.7742)	(812,291,972,440)
5	Car	0.6041 (0.6041)	(474, 338, 501, 353)

Table 2: Identified Objects

Object Category	Priority Level	% of Confidence Score
Car	1	100
Bicycle	2	90
Bus	2	90
Motorcycle	2	90
Truck	2	90
Stop Sign	3	80
Traffic Light	3	80
Person	4	70

Mobility features. We convert the raw GPS coordinates (longitude, latitude) in a trip to pairs of (bearing, distance) values following the procedures described in Section 2. A benefit of doing so is that the values of these measures are bounded and less influenced by the geographic location of the trip. In detail, trips span over a large geographic area. Using distance and bearing will mitigate spatial heterogeneity in the data. Besides, these measures are more meaningful than the raw GPS coordinates to describe the behaviors of cyclists. Note that, because bearing and distance are calculated between three consecutive points, we have $n-2$ pairs of features for a trip with length n.

Object features. To better understand the cyclist behavior, we also extract useful features from the geo-referenced egocentric images by detecting objects around cyclists. Road safety studies suggest that cyclists' behaviors are significantly affected by the road environment and other objects/vehicles sharing the road. Therefore, we first use an object detection method to identify major objects of interest in each scene and their locations. Then we use this information to help predict the behaviors of cyclists.

In detail, we adopt Mask R-CNN [7] to detect objects in each image and DAN (Deep Affinity Network) [24] to infer object affinities. In Figure 3, we identify objects in two consecutive images

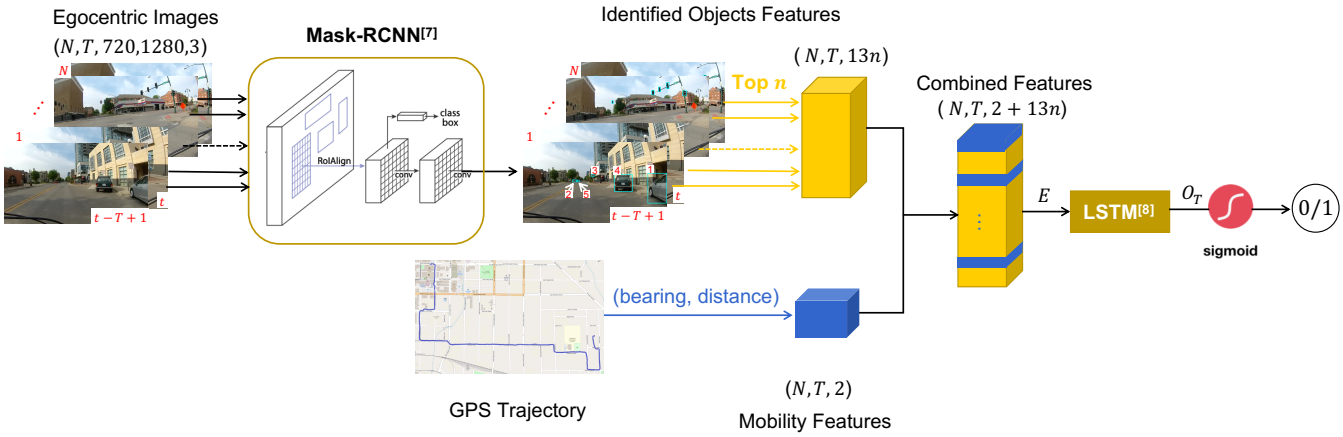


Figure 4: Framework of Baseline.

and track them with the same track ID. For each object, we use its category, confidence score, and bounding box to represent. The corresponding object features of Figure 3a are shown in Table 1. Note that, based on the domain knowledge [5], we focus on eight categories and their categories' priority level shown in Table 2.

3.2 Baseline

At first, we apply the Mask R-CNN using a ResNet-50-FPN backbone with pre-trained weights on the COCO dataset [13]. COCO is a large-scale object detection, segmentation, and captioning dataset including all eight categories in our work. For each geo-referenced egocentric image, we achieve a category label, confidence score, and bounding box of each identified object as their features.

In addition to the mobility features (i.e., bearing and distance), we incorporate Top n identified object features into the first model. At each timestamp, we order all identified objects by the overall score which is derived by multiplying their confidence scores and their categories' priority levels shown in Table 2. Then, we select Top n objects and concatenate their features with corresponding mobility features. Note that, we use one-hot encoding to convert each object category from categorical data to an integer vector with eight dimensions. In all, at each timestamp, we have a vector with $2 + 13n$ dimensions including two mobility features and Top n identified object features (13 = 8(category vector)+1(confidence score)+4(bounding box)).

In fact, understanding cyclist behavior is a sequence modeling problem. Our goal is to learn a fixed-length vector representation from sequences of cyclists' features with temporal dependencies for further usage. We commonly use the state-of-the-art sequence-to-sequence model, that is, LSTM (Long-Short Term Memory) [8] one of the families of RNN (Recurrent Neural Networks) [17]. Specifically, we apply a standard multi-layer LSTM to the multi-dimensional vector E extracted from both the GPS trajectories and the geo-referenced egocentric images. Each layer of the LSTM computes the following functions:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ \widetilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \end{aligned} \tag{6}$$

$$\begin{aligned} c_t &= i_t * \widetilde{c}_t + f_t * c_{t-1} \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ h_t &= o_t * \tanh(c_t) \end{aligned}$$

where t is the time step in terms of frames, x_t is the hidden state of the previous layer at time t, with the input $x_t = e_t$ for the first layer, h_t is the hidden state at time t, c_t is the cell state at time t, with initial values $h_0 = c_0 = 0$, $\widetilde{c}_t$ is the candidate state at time t, and i_t, f_t, o_t is the input, forget and out gates, respectively. σ is the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$, tanh is the tanh function $\tanh(x) = (e^{2x} - 1)/(e^{2x} + 1)$, and $*$ denotes Hadamard product. $W_\cdot$ and $U_\cdot$ are weight matrices and $b_\cdot$ are bias vectors.

Finally, a fully connected layer with a sigmoid function is connected to the output of the last LSTM layer O_T to predict the cyclist behavior as

$$\hat{y} = \sigma(W_T O_T + b_T) \tag{7}$$

where W_T and b_T are weight matrix and bias vector, respectively.

Overall, we feed the GPS trajectory and corresponding geo-referenced egocentric images of a valid cycling trip into the baseline shown in Figure 4.

3.3 Cycling-Net

Because objects in consecutive scenes might be in different orders according to the overall score. However, we aim to track identified objects and learn the relationship between the movement of these objects and the behavior of the cyclist. Instead of Top n identified objects, we select Top n tracking objects by averaging overall scores. Note that, we prefer to select longer tracking objects from as many different categories as possible.

Given two consecutive frames, we apply a pre-trained model to compute the affinity matrix of identified objects. In detail, if we identify n and m objects in each frame, the matrix $A \in \mathbb{R}^{n \times m}$ measures the data association of each pair of objects. The Top n Tracking algorithm is summarized in Algorithm 1. Given affinity matrices $A_\cdot$ from frame $t - T + 1$ to t and a Top number n, the algorithm starts to track each identified object according to their object ID and records its score, occurrence, and category until the predicted time t (Line 1-7). Then, it computes the average score

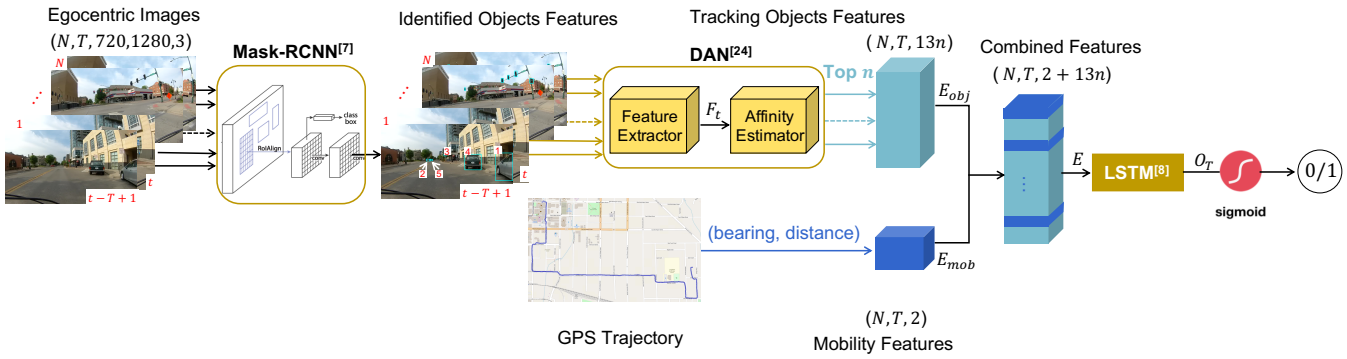


Figure 5: Framework of Cycling-Net with single LSTM.

for all identified objects and orders them by their occurrence and overall score (Line 8-9). Next, we select the Top tracking object from each identified category satisfying the number n requirement (Line 10-16). Finally, the Top n objects are reported (Line 17).

Algorithm 1 Top n Tracking algorithm

Input: Affinity matrices $A_{[t-T+1,t]}$, Top number n.
Output: Top n tracking objects Top_n.

1: Score Vector $S = \emptyset$, Length Vector $L = \emptyset$, Category Vector $C = \emptyset$.
2: **for** $i = 1$ **to** $T - 1$ **do**
3: **for** $\forall obj$ in $A_{t-T+i,.}$ **do**
4: **if** obj in S **then**
5: $S_{obj} += A_{t-T+i,obj}; L_{obj} += 1$
6: **else**
7: $S_{obj} = A_{t-T+i,obj}; L_{obj} = 1; C_{obj}$ =Category(obj)
8: $S_{obj} = S_{obj}/L_{obj}$, $\forall obj$ in S
9: Order all identified objects by L and S.
10: **while** $|Top_n| < n$ and $S \neq \emptyset$ **do**
11: **for** Each type in C_{obj} by Priority Level **do**
12: Select the Top 1 obj
13: $Top_n \leftarrow Top_n \cup \{obj\}$
14: Delete obj from S, L, C
15: **if** $|Top_n| == n$ **then**
16: **break**
17: **return** Top_n

Single LSTM. At this stage, we apply a DAN model with pre-trained weights on the MOT17 dataset [18], which is a benchmark for multi-object tracking. Then, we replace Top n identified objects features with Top n tracking objects features to combine with mobility features. Similarly, we still feed these combined features into a standard multi-layer LSTM following with a sigmoid function to predict the cyclist behavior. The new framework is shown in Figure 5.

Parallel LSTMs. To further improve our cyclist behavior prediction, we pay attention to the different underlying patterns of mobility features and Top n objects features. Particularly, mobility features reflect the movement of the cyclist while object features represent the movements of these objects. To implement this idea, we propose parallel LSTMs with joint training. Specifically, we jointly train two sub-LSTMs on the training set. One of the sub-LSTMs models the movement of the cyclist himself, while the other models the movements of Top n objects around the cyclist. We parallelize the two sub-LSTMs and merge them with a concatenate layer to jointly infer the cyclist behavior.

As shown in Figure 6, for each egocentric video trajectory, the input sequences of mobility features E_{mob} and Top n objects features E_{obj} are put separately into two sub-LSTMs in parallel to learn different moving patterns

$$\begin{aligned} O_{mob} &= \text{LSTM}(E_{mob}) \\ O_{obj} &= \text{LSTM}(E_{obj}) \end{aligned} \tag{8}$$

where LSTM denotes a standard multi-layer LSTM shown in Equation 6. Then, we merge O_{mob} and O_{obj} with a concatenate layer. At the end, the same linear projection with sigmoid function as in Equation 7 is connected to the concatenate feature matrix O_{con} to predict the cyclist behavior

$$\hat{y} = \sigma(W_T O_{con} + b_T). \tag{9}$$

Overall, the Cycling-Net learns the relationships between the movements of the cyclist and Top n objects around him/her and predicts the cyclist behavior at the end.

4 EVALUATION

In this section, we conduct extensive experiments on bicycling data with different objects features and validate the effectiveness of our proposed Cycling-Net.

4.1 Experiment Settings

We set up the experiments on Argon High Performance Computing System at the University of Iowa using a 256 GB RAM computing node with 2.6 GHz 16-Core CPU with Nvidia Tesla P100 Accelerator Cards. The primary development package is based on Tensorflow 1.14.0 in Python 3.6.

According to the data preprocessing steps in subsection 2.1, we generated 11453 trajectories in total and we use 60% (7247 samples) for training, 10% (955 samples) for validation, and rest 30% (3251 samples) for testing. For example, if we set previous timestamps $T = 5$ and number of top objects $n = 3$ in the features, the input of training data has the shape of (7247, 5, 41) corresponding to (samples, timestamps, features) while output has the shape of (7247, 1), where 1 indicates the cyclist behavior at next timestamp. We

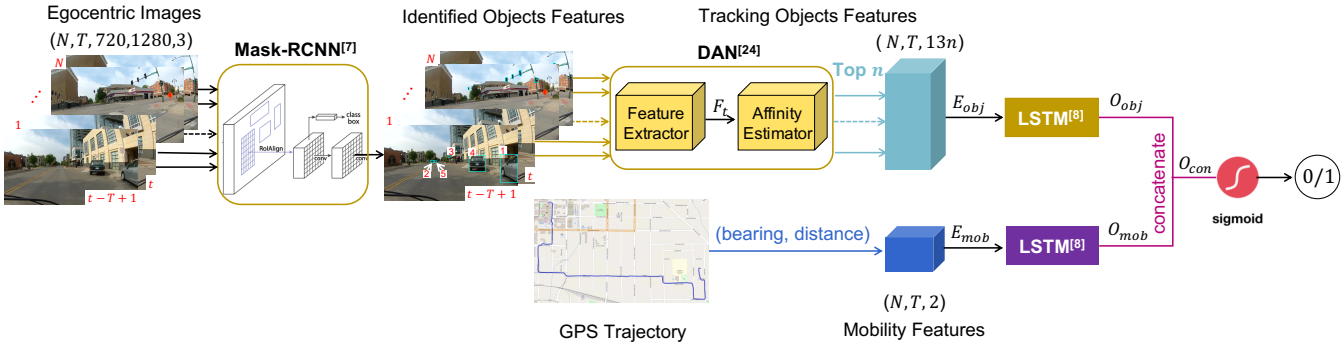


Figure 6: Framework of Cycling-Net with parallel LSTMs.

compute binary cross entropy loss for our prediction as

$$l = \sum_{n=1}^{N} \sum_{i=3}^{n} -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i), \tag{10}$$

where y_i is the binary ground-truth behavior label and $\hat{y_i}$ is predicted behavior label of the cyclist at frame i respectively. N is the total number of egocentric video trajectories and n denotes the length of each trajectory.

The metrics we compute include AUC (Area Under Curve), accuracy, recall, precision, and F-measure which are commonly used for binary classification problems.

For each proposed model, we apply different network structures (i.e., number of layers and nodes). When comparing the performance between Cycling-Net and the baseline, we use 2 hidden layers with 32 hidden nodes for each layer. In all experiments, we adopt early stopping criteria, set the batch size as 64, and select the âĂIJAdamâĂİ algorithm with default settings as our optimizer. In our experimental settings, if the validation loss decreased less than 10^{-6} after 30 epochs, the training will be terminated. In the end, we save the best weights instead of the latest weights. Each experiment is run 10 times and the average measures are reported.

4.2 Object Type Analysis

To take full advantage of object features, we conduct a brief analysis on identified objects from the egocentric video data using the R-CNN model.

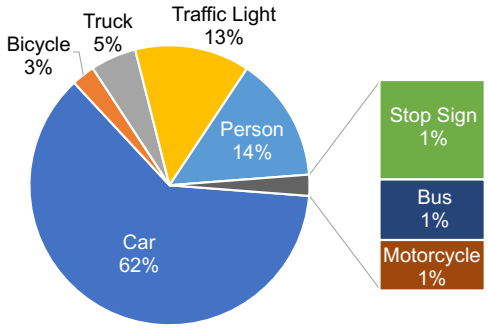


Figure 7: 8 Object Category Occupation

As shown in Figure 7, Car is the category of objects that are most frequently seen in the field of view of cyclists, around 60% of all the objects. Person and Traffic Light follow, accounting for about 15% each. The other 5 categories occupy the rest 10% in total.

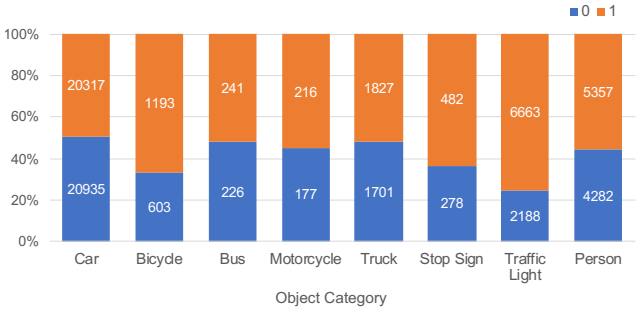


Figure 8: 8 Object Category Comparison

Figure 8 shows the proportion of each object category under different cyclist behavior types. The blue color indicates frames with behavior label 0 (moving forward with constant speed), and label 1 (with changes in direction or speed) is in orange. Besides, the exact number of objects identified is also labeled in the corresponding bar. For most object categories, the ratio is close to 1:1. However, for Traffic Light, Stop Sign, and Bicycle, the ratio is below 40%, which means when these objects are found in the scene, the label of the cyclist behavior at the same moment is more likely to be 1. Note that, for some frames, there are very few objects identified from the egocentric images. So, we ignore those frames with less than three objects identified. Besides, the total number of objects in most scenes is less than 8. Therefore, changing the number to a larger number (e.g., 10) may not make much difference and may cause the matrix to be very sparse. Finally, we test both using top-3 objects and top-5 objects as our object features in the experiments. For top-5 objects, we fill the object features with 0 if there are not enough objects found in the scene.

4.3 Performance of Cycling-Net vs. Baseline

Given the geo-referenced egocentric images, we first apply the pre-trained model Mask R-CNN mentioned in subsection 3.2 to identify object features including a category label, confidence score, and bounding box of each identified object. Then, we apply the

pre-trained model DAN mentioned in subsection 3.3 to track the movement of identified objects from each geo-referenced egocentric image and achieve tracking object features. At this stage, we select top $n = 3$ and $n = 5$ objects for each type of object. Thus, at each timestamp, we learn a vector including mobility features and object features with $41(= 2 + 13 \times 3)$ and $67(= 2 + 13 \times 5)$ dimensions, respectively. Here each object has 13 features, and there are two mobility features (bearing and distance). Next, we feed these combined features into baseline and Cycling-Net to predict the cyclist behavior with a sigmoid function at the end. We summarize comprehensive experimental results on our Cycling-Net framework compared with baseline incorporating with different Top n objects features in Table 3. We describe each group of results separately below.

4.4 Evaluation on Cycling-Net

Figure 9 compares the performance of baseline and Cycling-Net-1 (with single LSTM). The main difference between the two frameworks is that we directly take advantage of tracking identified objects by incorporating Top n tracking objects features in the Cycling-Net framework. As shown in Figure 9, Cycling-Net with Top 5 objects features performs significantly better than baseline over all metrics except precision. Particularly, Cycling-Net can balance precision-recall trade-off better than baseline with higher F-measure. In other words, recall contributes more than precision to F-measure. Likewise, we also incorporate Top 3 objects features in the two frameworks and achieve a similar trend. The detailed results are shown in Table 3. Note that, the accuracy of Top 3 objects, Cycling-Net with single LSTM is slightly lower than the baseline by 0.0074 which is caused by extreme results of 10 trials. Except for this point, for both Top 3 and Top 5 objects, Cycling-Net significantly outperforms baseline in terms of AUC, accuracy, F-measure, and recall.

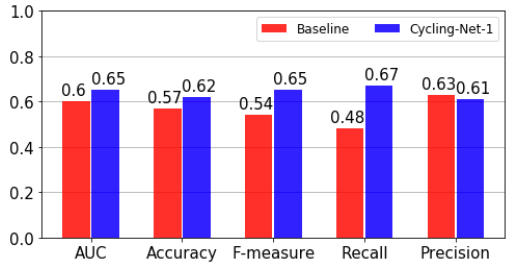


Figure 9: Performance comparison of baseline between Cycling-Net with single LSTM incorporating with Top 5 objects features.

In Figure 10, we compare the performance of Cycling-Net with single LSTM and parallel LSTMs designs, both trained with the Top 3 tracked object features. In terms of AUC and accuracy, the parallel LSTMs is significantly helpful in modeling moving patterns of both cyclists and identified objects in the field of view. Besides, both single LSTM and parallel LSTMs can balance the precision-recall trade-off well and achieve a similar F-measure. Likewise, incorporating with Top 5 tracking objects features in these two frameworks can perform a similar trend. The detailed results are shown in Table 3.

Summary. For Top n objects, because tracking different types of objects in consecutive scenes can help us better model the behavior of cyclists, incorporating tracking object features in Cycling-Net can achieve significant improvements. Besides, since the video data and GPS trajectories may exhibit different temporal dependencies, a benefit of adopting the parallel LSTMs structure is that the joint training can better capture these patterns separately. Therefore, it outperforms the single LSTM structure in Cycling-Net.

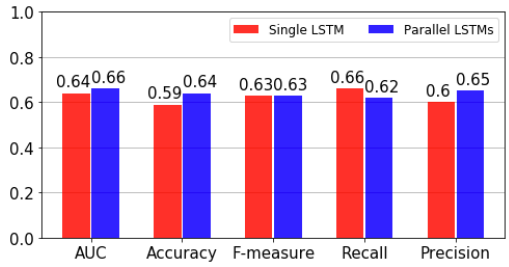


Figure 10: Performance comparison of Cycling-Net with single LSTM or parallel LSTMs incorporating with Top 3 tracking objects features.

4.5 Evaluation on Object Features

Then, we evaluate Cycling-Net by incorporating with different Top n object features and different types of objects features.

Impact of Top n Object Features. Figure 11 shows the overall performance of Cycling-Net with single LSTM incorporating with Top 3 and Top 5 tracking objects features, corresponding to the red and blue bars in the figure. In terms of all metrics, incorporating Top 5 tracking objects features always performs better than Top 3. Compared with the recall and precision of the baseline in Figure 9, we observe that the difference between recall and precision becomes small. It clearly shows that our Cycling-Net with single LSTM can balance the precision-recall trade-off, while baseline prefers improving precision at the expense of recall.

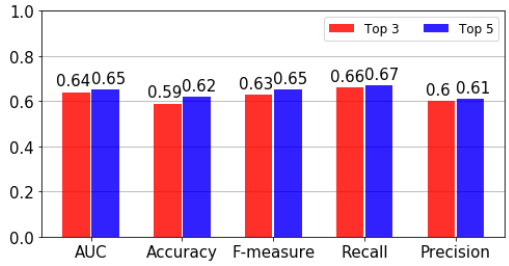


Figure 11: Overall Performance of Cycling-Net with single LSTM incorporating with Top n Tracking Objects Features

Figure 12 demonstrates the overall performance of Cycling-Net with parallel LSTMs incorporating with Top 3 and Top 5 tracking objects features, corresponding to the red and blue bars in the figure. The results show that both Top 3 and Top 5 tracking object features achieve comparable performance (all metrics value above 0.6) and can balance the trade-off between recall and precision well.

Table 3: Comprehensive Experimental Results on Cycling-Net with Baseline

Metrics	AUC		Accuracy		F-measure		Recall		Precision	
Object features	TOP 3	TOP 5	TOP 3	TOP 5	TOP 3	TOP 5	TOP 3	TOP 5	TOP 3	TOP 5
Baseline	0.6364	0.5989	0.5974	0.5663	0.5828	0.5450	0.5126	0.4774	**0.6752**	**0.6347**
Cycling-Net-1*	0.6446	0.6542	0.5900	0.6204	0.6286	**0.6523**	**0.6608**	0.6714	0.5994	0.6093
Cycling-Net-2*	**0.6648**	**0.6587**	**0.6429**	**0.6337**	**0.6345**	0.6394	0.6157	**0.6719**	0.6544	0.6098

*Cycling-Net-1 and Cycling-Net-2 denotes our Cycling-Net framework with single LSTM and parallel LSTMs, respectively.

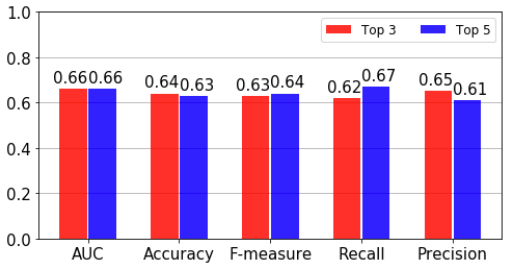


Figure 12: Overall Performance of Cycling-Net with parallel LSTMs incorporating with Top *n* Tracking Objects Features

Impact of Different Type Object Features. In addition, we try Top *n* identified objects features in our Cycling-Net framework by replacing Top *n* tracking objects features. Results in Figure 13 obviously show that with Top 5 tracking object features, Cycling-Net can significantly improve the performance on all metrics. This set of experiments confirms our intuition again that Top *n* tracking object features are more effective than Top *n* identified object features that help us better predict the behavior of cyclists.

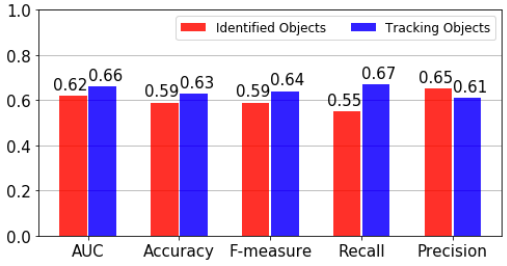


Figure 13: Performance comparison of Cycling-Net with parallel LSTMs incorporating with different Top 5 objects features.

Summary. For each type of object features, whether we select Top 3 or Top 5, they all show a similar performance trend while incorporating them in Cycling-Net. By incorporating the same Top *n* objects features, Cycling-Net with parallel LSTMs significantly outperforms other frameworks. In addition, Cycling-Net with either single LSTM or parallel LSTMs can balance the trade-off between recall and precision well and achieve higher F-measure than baseline. Overall, this set of experiments proves the effectiveness of our proposed Cycling-Net.

5 RELATED WORK

5.1 Cyclist Behavior Modeling

The problem of understanding cyclist behavior has been studied since the early 80's [2], which focused on bicycle traffic flow study and were limited to signalized intersection analysis only. Then, researchers began to model cyclist behavior with longitudinal analysis on weather and bicycle count variability data [19] and statistical analysis on the mixed traffic at signalized intersections [14]. As technology advanced, some microscopic simulation models and tools have been proposed for intelligent transport systems [3, 9, 16]. Particularly,the hardware and software implementations in [3] can provide a quantitative description of bicycle dynamics to measure cycling kinematics and bicyclist behavior. Bicycling simulation in [16] focuses on representing bicycle movements when the cyclist does not interact with others, while Huang et al. took other road users into consideration using social force model [9].

Meanwhile, there has been a trend in investigating cycling safety by examining causal factors associated with bicycle accidents [10, 11] and cycling environments [4, 6]. For example, cyclist behavioral variables are suggested to account for 70% of all single-bicycle accident injuries [21]. Hamann and Peek-Asa [5] used GPS and video data to understand bicyclist behavior variations and increase safety by selecting appropriate and targeted countermeasures.

Traditionally, social force models are commonly used for handcrafted features. Instead of measuring cycling kinematics and building statistical models to simulate cyclist behavior, we strive to generate an end-to-end framework to predict cyclist behavior directly by tracking moving objects in the cyclists' field of view automatically.

5.2 Deep Learning for Behavior Modeling

In the literature, various deep learning based methods have been proposed for driver and pedestrian trajectory prediction [1, 12, 26, 29, 30]. However, some have focused on predicting trajectories or traffic accident of a certain area only based on GPS trajectories [15, 25, 31]. Taking video data into consideration, most work focus on predicting behaviors of moving objects in video surveillance. For example, Social-LSTM [1] designs a Social pooling layer to capture the dependencies and interactions between different pedestrians. In addition to using a CNN to model the interactions between different pedestrians, Behavior-CNN [30] also adopts a 2D map to encode the history walking path. CIDNN [26] uses location-based spatial affinity to measure different pedestrians' influence on the target pedestrian. However, these works all focus on fixed third-person view camera data. A limited number of studies focused on

predicting trajectories using first-person/egocentric videos [22, 23, 27, 28]. Particularly, in [22], Park et al. associate a trajectory with the EgoRetinal map to predict a set of plausible trajectories of the camera wearer, while Yagi et al. predict future locations of people in the video [27]. Su et al. [23] require multiple first-person videos to reconstruct accurate 3D configurations of camera wearers.

Different from these works, we use both spatial trajectories and video data in our analysis. Also, we aim at predicting the behaviors of a cyclist himself/herself (the observer) whose movement pattern is likely to be different from those of pedestrians or vehicles.

6 CONCLUSION

This paper investigated the problem of cyclist behavior prediction. Given the historical trajectories and corresponding egocentric video taken along different cycling trips, we aim at learning a model to predict the next behavior of a cyclist based on his/her behaviors and egocentric camera video contents in the past few seconds. This problem is important for understanding the behaviors of cyclists in road safety research. Prior work did not address this problem as they either used static camera data or only predicted behaviors of targeted objects in the scene rather than the observer. In this paper, we proposed Cycling-Net, a deep learning framework to address the problem. Cycling-Net continuously tracks objects in the scene of the cyclist and use their movement patterns as features for prediction. Experiment results on a naturalistic cycling trip dataset showed that our proposed solution outperforms the baseline in all the measures.

7 ACKNOWLEDGEMENTS

This paper is funded partially by Safety Research using Simulation University Transportation Center (SAFER-SIM). SAFER-SIM is funded by a grant from the U.S. Department of Transportation's University Transportation Centers Program (69A3551747131). However, the U.S. Government assumes no liability for the contents or use thereof. This research was also funded in part by grant No. 1R49CE002108-01 of the National Center for Injury Prevention and Control/CDC. Yanhua Li was supported in part by NSF grants IIS-1942680 (CAREER), CNS-1952085, CMMI-1831140, and DGE-2021871.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–971.

[2] P Chao, Judson S Matthias, and Mary R Anderson. 1978. Cyclist Behavior at Signalized Intersections. *J. Transportation Research Board* 683 (1978), 34–39.

[3] Marco Dozza and Andre Fernandez. 2013. Understanding bicycle dynamics and cyclist behavior from naturalistic field data. *IEEE Trans. on Intelligent Transportation Systems* 15, 1 (2013), 376–384.

[4] Ariane Ghekiere, Benedicte Deforche, Lieze Mertens, Ilse De Bourdeaudhuij, Peter Clarys, Bas de Geus, Greet Cardon, Jack Nasar, Jo Salmon, and Jelle Van Cauwenberg. 2015. Creating cycling-friendly environments for children: which micro-scale factors are most important? An experimental study using manipulated photographs. *PloS one* 10, 12 (2015).

[5] Cara J Hamann and Corinne Peek-Asa. 2017. Beyond GPS: Improved study of bicycling exposure through added use of video data. *J. Transport & Health* 4 (2017), 363–372.

[6] M Anne Harris, Conor CO Reynolds, Meghan Winters, Mary Chipman, Peter A Cripton, Michael D Cusimano, and Kay Teschke. 2011. The Bicyclists' Injuries and the Cycling Environment study: a protocol to tackle methodological issues facing studies of bicycling safety. *Injury Prevention* 17, 5 (2011), e6–e6.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. Mask R-CNN. In *The IEEE International Conference on Computer Vision (ICCV)*.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[9] Ling Huang, Jianping Wu, Feng You, Zhihan Lv, and Houbing Song. 2016. Cyclist social force model at unsignalized intersections with heterogeneous traffic. *IEEE Trans. on Industrial Informatics* 13, 2 (2016), 782–792.

[10] Christian Juhra, Britta Wieskoetter, K Chu, L Trost, U Weiss, M Messerschmidt, A Malczyk, M Heckwolf, and M Raschke. 2012. Bicycle accidents–Do we only see the tip of the iceberg?: A prospective multi-centre study in a large German city combining medical and police data. *Injury* 43, 12 (2012), 2026–2034.

[11] Sigal Kaplan, Konstantinos Vavatsoulas, and Carlo Giacomo Prato. 2014. Aggravating and mitigating factors associated with cyclist injury severity in Denmark. *J. Safety Research* 50 (2014), 75–82.

[12] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. 2019. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5725–5734.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 740–755.

[14] Huang Ling and Jianping Wu. 2004. A study on cyclist behavior at signalized intersections. *IEEE Trans. on Intelligent Transportation Systems* 5, 4 (2004), 293–299.

[15] Jianming Lv, Qing Li, Qinghui Sun, and Xintong Wang. 2018. T-CONV: A convolutional neural network for multi-scale taxi trajectory prediction. In *Proceedings of the IEEE International Conference on Big Data and Smart Computing*. 82–89.

[16] Xiaoliang Ma and Ding Luo. 2016. Modeling cyclist acceleration process for bicycle traffic simulation using naturalistic data. *Transportation Research Part F: Traffic Psychology and Behaviour* 40 (2016), 130–144.

[17] Tomas Mikolov, Martin KarafiÃąt, Lukas Burget, Jan CernockÃ¡, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*. 1045–1048.

[18] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. (2016). arXiv:1603.00831

[19] Debbie A Niemeier. 1996. Longitudinal analysis of bicycle count variability: Results and modeling implications. *J. Transportation Engineering* 122, 3 (1996), 200–206.

[20] U.S. Department of Transportation. 2020. *Bicycle Safety Guide and Countermeasure Selection System*. Retrieved Jun 23, 2020 from http://www.pedbikesafe.org/bikesafe/countermeasures.cfm

[21] Felipe E Pedroso, Federico Angriman, Alexandra L Bellows, and Kathryn Taylor. 2016. Bicycle use and cyclist safety following BostonâĂŹs bicycle infrastructure expansion, 2009–2012. *American J. Public Health* 106, 12 (2016), 2171–2177.

[22] Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. 2016. Egocentric future localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4697–4705.

[23] Shan Su, Jung Pyo Hong, Jianbo Shi, and Hyun Soo Park. 2017. Predicting behaviors of basketball players from first person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1501–1510.

[24] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal S Mian, and Mubarak Shah. 2019. Deep affinity network for multiple object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2019).

[25] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. 2017. Modeling trajectories with recurrent neural networks. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*. 3083âĂŞ3090.

[26] Yanyu Xu, Zhixin Piao, and Shenghua Gao. 2018. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5275–5284.

[27] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. 2018. Future person localization in first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7593–7602.

[28] Yu Yao, Mingze Xu, Chiho Choi, David J Crandall, Ella M Atkins, and Behzad Dariush. 2019. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*. 9711–9717.

[29] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2015. Understanding pedestrian behaviors from stationary crowd groups. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3488–3496.

[30] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2016. Pedestrian behavior understanding and prediction with deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 263–279.

[31] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 984–992.