Anticipating Accidents in Dashcam Videos

[1]Fu-Hsiang Chan, [1]Yu-Ting Chen, [2]Yu Xiang, [1]Min Sun

[1]Dept. of Electrical Engineering, National Tsing Hua University, Taiwan.
[2]Dept. of Computer Science & Engineering, University of Washington, USA.
corgi1205@gmail.com, s728039@gmail.com, yuxiang@cs.washington.edu,
sunmin@ee.nthu.edu.tw

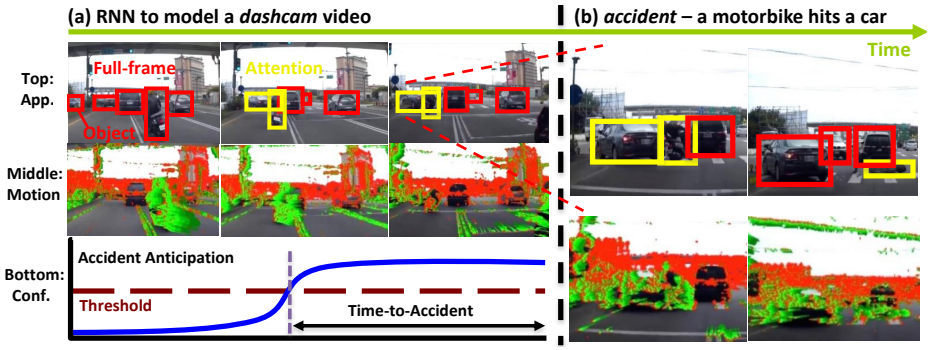


Fig. 1. Illustration of Accident Anticipation. Given a dashcam video (playing from left to right in panel (a)), we extract both appearance (top-row: App.) [1] and motion (middle-row) [2] features. For appearance feature, we consider both full-frame and object-level (red-boxes) features. For motion feature, keypoints with large motion are shown in green dots. Our proposed dynamic-spatial-attention Recurrent Neural Network (RNN) distributes attention to objects (yellow-boxes indicate strong attention) and predicts the confidence of accident anticipation at each frame (bottom-row: Conf.). Once the confidence reaches a threshold (brown-dash-line), our system will trigger an accident alert "time-to-accident" seconds before the true accident (Panel (b)).

Abstract. We propose a Dynamic-Spatial-Attention (DSA) Recurrent Neural Network (RNN) for anticipating accidents in dashcam videos (Fig. 1). Our DSA-RNN learns to (1) distribute soft-attention to candidate objects dynamically to gather subtle cues and (2) model the temporal dependencies of all cues to robustly anticipate an accident. Anticipating accidents is much less addressed than anticipating events such as changing a lane, making a turn, etc., since accidents are rare to be observed and can happen in many different ways mostly in a sudden. To overcome these challenges, we (1) utilize state-of-the-art object detector [3] to detect candidate objects, and (2) incorporate full-frame and object-based appearance and motion features in our model. We also harvest a diverse dataset of 678 dashcam accident videos on the web (Fig. 3). The dataset is unique, since various accidents (e.g., a motorbike hits a car, a car hits another car, etc.) occur in all videos. We manually mark the time-location of accidents and use them as supervision to train and evaluate our method. We show that our method anticipates accidents about 2 seconds before they occur with 80% recall and 56.14% precision. Most importantly, it achieves the highest mean average precision (74.35%) outperforming other baselines without attention or RNN.

1 Introduction

Driving a car by an Artificial Intelligent (AI) agent has been one of the greatest dream in AI for decades. In the past 10 years, significant advances have been achieved. Since 2009, Google's self-driving car has accumulated 1,011,338 autonomous driving miles on highways and busy urban streets [4]. Recently, Tesla's Autopilot can drive on highway-like environment primarily relying on cheap vision sensors. Despite these great advances, there are two major challenges. The first challenge is how to drive safely with other "human drivers". Google's self-driving car is involved in 12 minor accidents [4] mostly caused by other human drivers. This suggests that a self-driving car should learn to anticipate others' behaviors in order to avoid these accidents. The other important challenge is how to scale-up the learning process. In particular, how to learn from as many corner cases as possible? We propose to take advantage of the cheap and widely available dashboard cameras (later referred to as dashcam) to observe corner cases at scale.

Dashcam is very popular in places such as Russia, Taiwan and Korean. For instance, dashcams are equipped on almost all new cars in the last three years in Taiwan. Its most common use case is to record how accidents occur in order to clarify responsibilities. As a result, many dashcam videos involving accidents have been recorded. Moreover, according to statistics, ~ 90 people died per day due to road accidents in the US [5]. In order to avoid these casualty, we propose a method to learn from dashcam videos for anticipating various accidents and providing warnings a few seconds before the accidents occur (see Fig. 1).

Learning to anticipate accidents is an extremely challenging task, since accidents are very diverse and they typically happen in a sudden. Human drivers learn from experiences to pay attention on subtle cues including scene semantic, object appearance and motion. We propose a Dynamic-Spatial-Attention (DSA) Recurrent Neural Network (RNN) to anticipate accidents before they occur. Our method consists of three important model designs:

- Dynamic-spatial-attention: The DSA mechanism learns to distribute soft-attention to candidate objects in each frame dynamically for anticipating accidents.
- RNN for sequence modeling: We use RNN with Long Short-Term Memory (LSTM) [6] cells to model the long-term dependencies of all cues to anticipate an accident.
- Exponential-loss: Inspired by [7] on anticipating drivers' maneuvers, we adopt the exponential-loss function as the loss for positive examples.

To effectively extract cues, we rely on state-of-the-art deep learning approaches to reliably detect moving objects [3] and represent them using learned deep features [1] as the observations of our DSA-RNN model. All these components together enable accident anticipation using a cheap vision-based sensor.

In order to evaluate our proposed method, we download 678 dashcam videos with high video quality (720p in resolution) from the web. The dataset is unique, since various accidents (e.g., a motorbike hits a car, a car hits another car, etc.) occur in all videos. Moreover, most videos are captured across six cities

in Taiwan. Due to the crowded road with many moving objects and complicated street signs/billboards, it is a challenging dataset for vision-based method (Fig. 3-Right). For each video, we manually annotated the bounding boxes of car, motorbike, bicycle, human, and the time when the accident occurs. 58 out of 678 videos are used only for training the object detector. Among the remaining 620 videos, we manually select 620 positive clips and 1130 negative clips, where each clip consists of 100 frames. A positive clip contains the moment of accident at the last 10 frames, and a negative clip contains no accident. We split the dataset into training and testing, where the number of training clips is roughly three times the number of testing clips. We show in experiments that all our model designs help improve the anticipation accuracy. Our method can anticipate accident 1.8559 seconds ahead with high recall (80%) and reasonable precision (56.14%).

In summary, our method has the following contributions:

- We show that, using deep learning, a vision-based sensor (dashcam) can provide subtle cues to anticipate accidents in complex urban driving scenarios.
- We propose a dynamic-spatial-attention RNN to achieve accident anticipation 1.8559 seconds ahead with 80% recall and 56.14% precision.
- We show that potentially a vast amount of dashcam videos can be used to improve self-driving car ability.
- We introduce the first crowd-sourced dashcam video dataset for accident anticipation available online at http://aliensunmin.github.io/project/dashcam/.

In the following sections, we will first describe the related work. Then, our method is introduced in Sec. 3. Finally, experiments are discussed in Sec. 4.

2 Related Work

We first discuss related work of anticipation in computer vision, robotics and intelligent vehicle. Then, we mention recent works incorporating RNN with attention mechanism in computer vision. Finally, we compare our dashcam accident anticipation dataset with two large-scale dashcam video datasets.

2.1 Anticipation

A few works have been proposed to anticipate events — classify "future" event given "current" observation. Ryoo [8] proposes a feature matching techniques for early recognition of unfinished activities. Hoai and Torre [9] propose a max-margin-based classifier to predict subtle facial expressions before they occur. Lan et al. [10] propose a new representation to predict the future actions of people in unconstrained in-the-wild footages. Our method is related to event anticipation, since accident can be consider as a special event. Moreover, our dashcam videos are very challenging, since these videos are captured by different moving cameras observing static stuff (e.g., building, road signs/billboards, etc.) and moving objects (e.g., motorbikes, cars, etc.) on the road. Therefore, we propose a dynamic-spatial-attention mechanism to discovery subtle cues for anticipating accidents.

Anticipation has been applied in tasks other than event prediction. Kitani et al. [11] propose to forecast human trajectory by surrounding physical environment (e.g., road, pavement, etc.). The paper also shows that the forecasted trajectory can be used to improve object tracking accuracy. Yuen and Torralba [12] propose to predict motion from still images. Julian et al. [13] propose a novel visual appearance prediction method based on mid-level visual elements with temporal modeling methods.

Event anticipation is also popular in the robotic community [14–17]. Wang et al. [14] propose a latent variable model for inferring unknown human intentions. Koppula and Saxena [15] address the problem of anticipating future activities from RGB-D data. A real robotic system has executed the proposed method to assist humans in daily tasks. [16, 17] also propose to anticipate human activities for improving human-robot collaboration.

There are also many works for predicting drivers' intention in the intelligent vehicle community. [18–21] have used vehicle trajectories to predict the intent for lane change or turn maneuver. A few works [22–24] address maneuver anticipation through sensory-fusion from multiple cameras, GPS, and vehicle dynamics. However, most methods assume that informative cues always appear at a fixed time before the maneuver. [25, 7] are two exceptions which use an input-output HMM and a RNN, respectively, to model the temporal order of cues. On one hand, our proposed method is very relevant to [7], since they also use RNN to model temporal order of cues. On the other hand, anticipating accidents is mush less addressed than anticipating specific-maneuvers such as lane change or turn, since accidents are rare to be observed and can happen in many different ways mostly in a sudden. In order to address the challenges in accident anticipation, our method incorporates a RNN with spatial attention mechanism to focus on object-specific cues at each frame dynamically. In summary, all these previous methods focus on anticipating specific-maneuvers such as lane change or turn. In contrast, we aim at anticipating various accidents observed in naturally captured dashcam videos.

2.2 RNN with Attention

Recently, RNN with attention has been applied on a few core computer vision tasks: video/image captioning and object recognition. On one hand, RNN with soft-attention has been used to jointly model a visual observation and a sentence for video/image caption generation [26, 27]. Yao et al. [26] incorporate a "temporal" soft-attention mechanism to select critical frames to generate a video caption. Xu et al. [27] demonstrate the power of spatial-attention mechanism for generating an image caption. Compared to [27], our proposed dynamic-spatial-attention RNN has two main differences: (1) their spatial-attention is for a single frame, whereas our spatial-attention changes dynamically at each frame in a sequence; (2) rather than applying spatial-attention on a regular grid, we apply a state-of-the-art object detector [3] to extract object candidates for assigning the spatial-attention. On the other hand, a few RNN with hard-attention models have been proposed. Mnih et al. [28] propose to train a RNN with hard-attention model with a reinforcement learning method for image classification tasks. Sim-

ilarly, Ba et al. [29] propose a RNN with hard-attention model to both localize and recognize multiple objects in an image. They apply their method on transcribing house number sequences from Google Street View images.

2.3 Dashcam Video Dataset

Dashcam videos or videos captured by cameras on vehicles have been collected for studying many recognition and localization tasks. For instance, CamVid [30], Leuven [31], and Daimler Urban Segmentation [32] have been introduced to study semantic understanding of urban scenes. There are also two recently collected large-scale datasets [33, 34]. KITTI [33] is a well-known vision benchmark dataset to study vision-based self-driving tasks including object detection, multiple-objects tracking, road/lane detection, semantic segmentation, and visual odometry. KITTI consists of videos captured around the mid-size city of Karlsruhe, in rural areas and on highways. Moreover, all videos are captured by vehicles with the same equipment under normal driving situation (i.e., no accident), whereas our dataset consists of accident videos harvested from many online users across six cities. Recently a large-scale dashcam dataset [34] is released for evaluating semantic segmentation task. It consists of frames captured in 50 cities. Among them, 5k frames and 30k frames are labeled with detail and coarse semantic labels, respectively. Despite the diverse observation in this new dataset, most frames are still captured under normal driving situation. We believe our dashcam accident anticipation dataset is one of the first crowd-sourced datasets for anticipating accidents.

3 Our System

We formally define accident anticipation and then present our proposed Dynamic-spatial-attention (DSA) Recurrent Neural Network (RNN). The goal of accident anticipation is to use observations in a dashcam video to predict an accident before it occurs. We define our observations and accident label for the j^{th} video as $((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)_j, y_j)$, where $\mathbf{x}_t$ is the observation at frame t, T is the total number of frames in the video, and y_j is the accident label to specify at which frame the accident started (defined below). For instance, if $y = \hat{t}$, any $t < \hat{t}$ is a frame before the accident. With a bit abuse of notation, we use $y = \infty$ to specify the video as free from accident. During training, all the observations and accident labels are given to train a model for anticipation. While in testing, our system are given an observation of $\mathbf{x}_t$ one at a time following the order of the frames. The goal is to predict the future accident as early as possible given the observations $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t)|t < y$ before accident occurs at frame y.

Our proposed dynamic-spatial-attention RNN is built upon standard RNN based on Long Short-Term Memory (LSTM) cells. We first give preliminaries of the standard RNN and LSTM before we describe the dynamic-spatial-attention mechanism (Sec. 3.2) and training procedure for anticipation (Sec. 3.3).

3.1 Preliminaries

RNN. Standard RNN is a special type of network which takes a sequence of observations $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ as input and outputs a sequence of learned hidden representations $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T)$, where $\mathbf{h}_t$ encodes the sequence observations

$(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t)$ up to frame t. The hidden representation is generated by a recursive equation below,

$$\mathbf{h}_t = g(\mathbf{W}\mathbf{x}_t + \mathbf{H}\mathbf{h}_{t-1} + \mathbf{b}) \ , \tag{1}$$

where $g(\cdot)$ is a non-linear function applied element-wise (e.g., sigmoid), $\mathbf{W}, \mathbf{H}, \mathbf{b}$ are the model parameters to be learned. The hidden representation $\mathbf{h}_t$ is used to predict a target output. In our case, the target output is the probability of discrete event a_t,

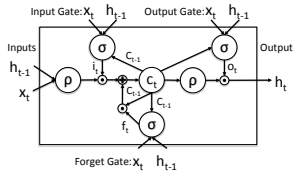
$$\mathbf{a}_t = \text{softmax}(\mathbf{W}_a\mathbf{h}_t + \mathbf{b}_a) \ , \tag{2}$$

where $\mathbf{a}_t = [\ldots, a_t^i, \ldots]$. The softmax function computes the probability of events (i.e., $\sum_i a_t^i = 1$), and $\mathbf{W}_a, \mathbf{b}_a$ are the model parameters to be learned. For accident anticipation, accident and non-accident are the discrete events and their probabilities are denoted by a_t^0 and a_t^1, respectively. In this work, we denote matrices with bold, capital letters, and vectors with bold, lower-case letters. The recursive design of RNN is clear and easy to understand. However, it suffers from a well-known problem of vanishing gradients [35] such that it is hard to train a RNN to capture long-term dependencies. A common way to address this issue is to replace function $g(\cdot)$ with a complicated Long Short-Term (LSTM) Memory cell [6]. We now give an overview of the LSTM cell and then define our dynamic-spatial-attention RNN based on LSTM cells.

Long-Short Term Memory Cells. LSTM introduces a memory cell $\mathbf{c}$ to maintain information over time. It can be considered as the state of the recurrent system. LSTM extends the standard RNN by replacing the recursive equation in Eq. 1 with

$$(\mathbf{h}_t, \mathbf{c}_t) = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \ , \tag{3}$$

where the memory cell $\mathbf{c}$ allows RNN to model long-term contextual dependencies. To control the interaction among the input, memory cell, and output, three gates are designed: input gate $\mathbf{i}$, forget gate $\mathbf{f}$, and output gate $\mathbf{o}$ (see Fig. 3.1). Each gate is designed to either block or non-block information. At each frame t, LSTM first computes gate activations: $\mathbf{i}_t, \mathbf{f}_t$ (Eq. 4,5) and updates its memory cell from $\mathbf{c}_{t-1}$ to $\mathbf{c}_t$ (Eq. 6). Then it computes the output gate activation $\mathbf{o}_t$ (Eq. 7), and outputs a hidden representation $\mathbf{h}_t$ (Eq. 8). We now define the



Workflow of LSTM [6].

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{c}_{t-1} + \mathbf{b}_i) \tag{4}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{V}_f\mathbf{c}_{t-1} + \mathbf{b}_f) \tag{5}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \rho(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{6}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{c}_t + \mathbf{b}_o) \tag{7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \rho(\mathbf{c}_t) \tag{8}$$

common notations in Eq. 4–8. $\odot$ is an element-wise product, and the logistic function σ and the hyperbolic tangent function ρ are both applied element-wise. $\mathbf{W}_*, \mathbf{V}_*, \mathbf{U}_*$, $\mathbf{b}_*$, and $\mathbf{V}_*$[1] are the parameters. Note that the input and forget

[1] The subscript $*$ denotes any symbol.

gates participate in updating the memory cell (Eq. 6). More specifically, forget gate controls the part of memory to forget, and the input gate allows newly computed values (based on the current observation and previous hidden representation) to add to the memory cell. The output gate together with the memory cell computes the hidden representation (Eq. 8). Since the current memory cell only goes through a binary operation (i.e., forget gate) and a summation (Eq. 6), the gradient with respect to the memory cell does not vanish as fast as standard RNN.We now describe our dynamic-spatial-attention RNN architecture based on LSTMs for anticipation.

3.2 Dynamic Spatial Attention

For accident anticipation, we would like our RNN to focus on spatial-specific observations corresponding to vehicles, pedestrian, or other objects in the scene. We propose to learn a dynamic spatial attention model to focus on candidate objects on specific spatial locations at each frame (Fig. 2). We assume that there are J spatial-specific object observations $\mathbf{X}_t = \{\hat{\mathbf{x}}_t^j\}_{j\in\{1,\ldots,J\}}$ and their corresponding locations $\mathcal{L}_t = \{\mathbf{l}_t^j\}_{j\in\{1,\ldots,J\}}$. We propose to adapt the recently proposed soft-attention mechanism [27] to take dynamic weighted-sum of spatially-specific object observations $\mathbf{X}_t$ as below,

$$\phi(\mathbf{X}_t, \boldsymbol{\alpha}_t) = \sum_{j=1}^{J} \alpha_t^j \hat{\mathbf{x}}_t^j, \tag{9}$$

where $\phi(\mathbf{X}_t, \boldsymbol{\alpha}_t)$ is the dynamic weighted-sum function[2], $\sum_{j=1}^{J} \alpha_t^j = 1$ and α_t^j is computed at each frame t along with the LSTM. We refer $\boldsymbol{\alpha}_t = \{\alpha_t^j\}_j$ as the attention weights. They are computed from unnormalized attention weights $\mathbf{e}_t = \{e_t^j\}_j$ as below,

$$\alpha_t^j = \frac{\exp(e_t^j)}{\sum_j \exp(e_t^j)} \ . \tag{10}$$

We design the unnormalized attention weights to measure the relevance between the previous hidden representation $\mathbf{h}_{t-1}$ and each spatial-specific observation $\hat{\mathbf{x}}_t^j$ as below,

$$e_t^j = \mathbf{w}^T \rho(\mathbf{W}_e \mathbf{h}_{t-1} + \mathbf{U}_e \hat{\mathbf{x}}_t^j + \mathbf{b}_e) \ , \tag{11}$$

where $\mathbf{w}, \mathbf{W}_e, \mathbf{U}_e$, and $\mathbf{b}_e$ are model parameters. Then, we replace all $\mathbf{x}_t$ in Eq. 4,5,6,7,8 with $\phi(\mathbf{X}_t)$, which is a shorthand notation of $\phi(\mathbf{X}_t, \boldsymbol{\alpha}_t)$ in Eq. 9. Note that Xu et al. [27] apply spatial-attention on a regular grid in a single frame, whereas our method applies spatial-attention on candidate object regions detected by state-of-the-art deep-learning-based detector [3]. Moreover, rather than applying spatial-attention on a single frame for caption generation, we apply spatial-attention on a "sequence" of frames dynamically which are jointly modeled using RNN.

[2] $\boldsymbol{\alpha}_t$ is often omitted for conciseness.

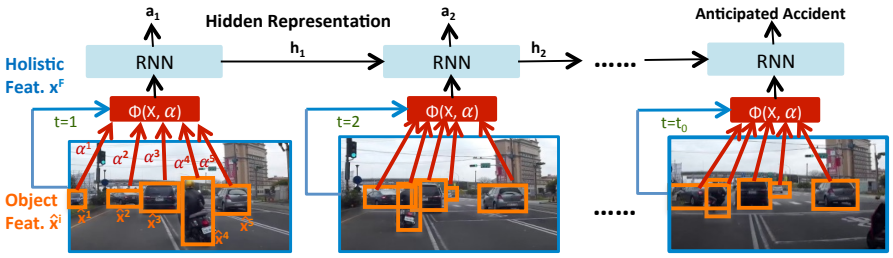


Fig. 2. The model visualization of our dynamic-spatial-attention RNN which takes weighted sum of the full-frame feature $\mathbf{x}^F$ and object features $\mathbf{X} = \{\hat{\mathbf{x}}^i\}_i$ as observation (one variant in Sec. 3.2). This example shows that the accident is anticipated at time t_0, which is $y - t_0$ seconds before true accident. $\hat{\mathbf{x}}^i$ denotes observation of the i^{th} object, The function $\phi(\mathbf{X}, \boldsymbol{\alpha})$ in Eq. 9 computes the weighted-sum of all features. $\mathbf{a}_t$ is the probability of a future accident defined in Eq. 2. $\mathbf{h}_t$ is the learned hidden representation which propagates to the next RNN (see Eq. 7,8). Feat. stands for feature. Note that the subscript t is omitted when it is clear from the context that a variable is time-specific.

Combining with full-frame feature. Spatial-specific object observations incorporate detail cues of moving objects which might involve in the accident. However, full-frame feature can capture important cues related to the scene or motion of the camera, etc. We propose two ways to combine the full-frame feature with spatial-specific object feature.

- Concatenation. We can simply concatenate the full-frame feature $\mathbf{x}^F$ with the weighted-summed object feature $\phi(\mathbf{X})$ as $\mathbf{x} = [\mathbf{x}^F; \phi(\mathbf{X})]$.
- Weighted-sum. We can treat the full-frame as an object as large as the whole frame. Then, the attention model will assign a soft-weight for the full-frame feature using the mechanism described above. Note that this way of combining reduces the combined feature dimension by two.

3.3 Training Procedure

Accident probability $\mathbf{a}_t$ is the targeted output of our DSA-RNN. We describe its corresponding loss function.

Anticipation loss. Intuitively, the penalty of failing to anticipate an accident at a frame very close to the accident should be higher than the penalty at a frame far away from the accident. Hence, we use the following exponential loss [7] for positive accident training videos,

$$\mathrm{L}_p(\{\mathbf{a}_t\}) = \sum_t -e^{-\max(0, y-t)} \log(a_t^0) \ , \tag{12}$$

where the accident happens at frame y, and a_t^0 is the anticipated probability of accident at frame t. For negative training videos, we use the standard cross-entropy loss,

$$\mathrm{L}_n(\{\mathbf{a}_t\}) = \sum_t -\log(a_t^1) \ , \tag{13}$$

where a_t^1 is the anticipated probability of non-accident at frame t.

The final loss is the sum of all these losses across all training videos,

$$\sum_{j \in P} \mathrm{L}_p(\{\mathbf{a}_t^j\}) + \sum_{j \in N} \mathrm{L}_n(\{\mathbf{a}_t^j\}) \ , \tag{14}$$

where j is the video index, $P = \{j; y_j \neq \infty\}$ is the set of positive videos, and $N = \{j; y_j = \infty\}$ is the set of negative videos. Since all loss functions are differentiable, we use stochastic gradient with the standard back-propagation through time (BPTT) algorithm [36] to train our model. Detail training parameters are described in Sec. 4.2.

4 Experiments

In this section, we first introduce our novel dashcam accident dataset and describe the implementation details. Finally, we describe all the baseline methods for comparison and report the experimental results.

4.1 Dashcam Accident Dataset

A dashcam is a cheap aftermarket camera, which can be mounted inside a vehicle to record street-level visual observation from the driver's point-of-view (see Fig. 3-Top-Right-Corner). In certain places such as Russia and Taiwan, dashcams are equipped on almost all new cars in the last three years. Hence, a large number of dashcam videos have been shared on video sharing websites such as YouTube[3]. Instead of recording dashcam videos ourselves similar to other datasets [33, 34], we harvest dashcam videos shared online from many users. In particular, we target at accident videos with human annotated address information or GPS locations. In this way, we have collected various accident videos with high video quality (720p in resolution). The dataset consists of 678 videos captured in six major cities in Taiwan (Fig. 3-Right). Our diverse accidents include: 42.6% motorbike hits car, 19.7% car hits car, 15.6% motorbike hits motorbike, and 20% other types. Figure. 3 shows a few sample videos and their corresponding locations on Google map. We can see that almost all big cities on the west coast of Taiwan are covered. Our videos are more challenging than videos in the KITTI [33] dataset due to the following reasons,

- Complicated road scene: The street signs and billboards in Taiwan are significantly more complex than those in Europe.
- Crowded streets: The number of moving cars, motorbikes, and pedestrians per frame are typically larger than other datasets [33, 34].
- Diverse accidents: Accidents involving cars, motorbikes, etc. are all included in our dataset.

We manually annotate the temporal locations of accidents and the moving objects in each video. 58 videos are used only for training the object detector. Among the remaining 620 videos, we sample 1750 clips, where each clip consists of 100 frames (5 seconds). These clips contain 620 positive clips containing the moment of accident at the last 10 frames[4], and 1130 negative clips containing no accidents. We randomly split the dataset into training and testing, where

[3] https://www.youtube.com/watch?v=YHFvSCAg4DE

[4] Hence, we use the first 90 frames to anticipate accidents.

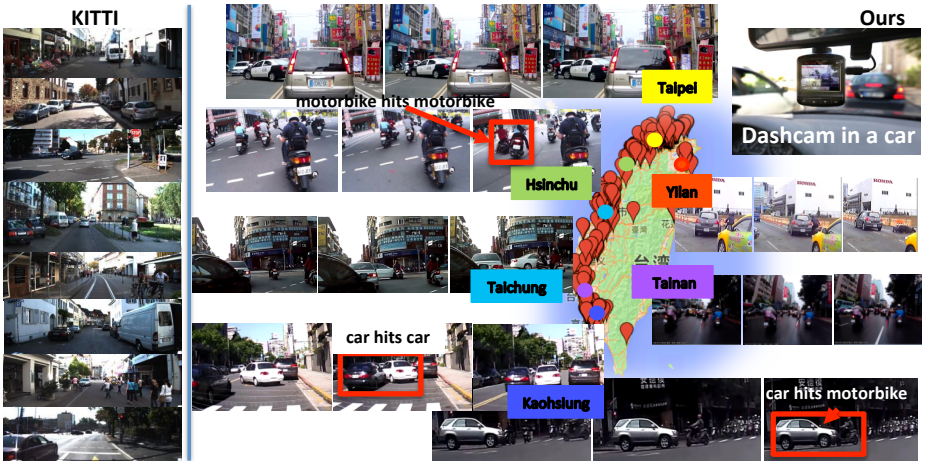


Fig. 3. Our dashcam accident dataset consists of a large number of diverse accident dashcam videos (Right-panel). It typically contains more moving objects and complicated street signs/billboards than the KITTI [33] dataset (Left-panel).

the number of training clips is about three times the number of testing clips: 1284 training clips 455 positive and 829 negative clips) and 466 testing clips (165 positive and 301 negative clips). We will make all the original videos, their annotated accident locations, and our sampled clips publicly available.

4.2 Implementation Details

Features. Both appearance and motion cues are intuitively important for accident anticipation. We extract both single-frame-based appearance and clip-based local motion features. For capturing appearance, we use pre-trained VGG [1] network to extract a fixed 4096 dimension feature for each frame at 20fps. For motion feature, we extract improved dense trajectory (IDT) feature [2][5] for a clip consisting of 5 consecutive frames. Then, we first use PCA to reduce the trajectory feature dimension to 100, and train a Gaussian-Mixture-Model (GMM) with 64 clusters. Finally, we use the 1st order statistic of fisher vector encoding to compute a fixed 6400 dimension feature. For VGG, we extract features both on a full-frame and on each candidate object, and we combine them following the methods described in Sec. 3.2. For IDT, we only extract features on a full-frame, since many candidate object regions do not contain enough trajectories to compute a robust IDT feature. In addition, we design a Relativity-Motion (RM) features using relative 2D motion among nearby objects (5x5 median motion encoding).

Candidate objects. As mentioned in Sec. 3.2, we assume our model observes J spatial-specific object regions. Our proposed dynamic-spatial-attention model will learn to distribute its soft-attention to these regions. Given an image, there are a huge number of possible object regions, when considering all locations and

[5] IDT also includes Histogram of Oriented Gradient (HOG) [37] (an appearance feature) on the motion boundary.

scales. To limit the number of object regions, we use a state-of-the-art object detector [3] to generate less than 20 candidate object regions for each frame. Since the object detector pre-trained on MSCOCO dataset [38] is not trained to detect objects in street scenes, we finetune the last three fully connected layers on street scenes data including KITTI dataset [33], our collected 58 videos, and randomly sampled 10 frames in 455 positive training clips. Our finetuned detector achieves 52.3% mean Average Precision (mAP) across five categories[6], which significantly outperforms the pre-trained detector (41.53%) (see more detail in supplementary material).

Model learning. All experiments use 0.0001 learning rate, 40 maximum epoch, 10 batch size. We implement our method on TensorFlow [39].

4.3 Evaluation Metric.

We evaluate every method based on the correctness of anticipating a future accident. Given a video, a method needs to generate the confidence/probability of future accident a_t^0 at each frame. At frame t when the confidence is higher than or equal to a threshold q, the method claims that there will be an accident in the future. If the video is an accident video, this is a True Positive (TP) anticipation. The accident is correctly anticipated at frame t, which is $y - t$ frames before it occurs at frame y. We define $y - t$ as time-to-accident. If the video is a non-accident video, this is a False Positive (FP) anticipation. On the other hand, if all the confidence $\{a_t^0\}_{t<y}$ are smaller than the threshold q, the method claims that there will not be an accident in the future. If the video is an accident video, this is a False Negative (FN) prediction. If the video is a non-accident video, this is a True Negative (TN) prediction. For each threshold q, we can compute the precision $= \frac{\text{TP}}{\text{TP+FP}}$ and recall $= \frac{\text{TP}}{\text{TP+FN}}$. By changing the threshold q, we can compute many pairs of precision and recall and plot the precision v.s. recall curve (see Fig. 4-Left). Given a sequence of precision and recall pairs, we can compute the average precision, which is used to show the system's overall accuracy. For each threshold q, we can also collect all the Time-to-accident (ToA) of the true positive anticipation, and compute the average ToA as the expected anticipation time.

4.4 Baseline Methods

We compare different variants of our method using RNN and a few baseline methods without modeling the temporal relation between frames. Here we present these variants and baselines as a series of simplifications on our proposed method.

- Dynamic-Spatial-Attention RNN. This is our proposed method. Our method has three variants (see Sec. 3.2): (1) no full-frame features, only attention on object candidates (D); (2) weighted-summing full-frame feature with object-specific features (F+D-sum); (3) concatenating full-frame features with object features (F+D-con.).
- Average-Attention RNN. We replace the inferred spatial-attention with a average attention (no dynamic attention), where all candidate object observations are average-pooled to a fixed dimension feature. Then, we either use only the average attention feature (avg.-D), or concatenate the full-frame feature with the average

[6] human, bicycle, motorbike, car and bus.

attention feature (F+avg.-D-con.). These baselines highlight the effect of using dynamic spatial-attention.
- Frame-based RNN. We remove all candidate object observations and use only full frame observation (F). This baseline highlights the effect of using candidate object observations.
- Average-Attention Single-frame Classifier (SFC). We start from Average-Attention RNN (avg.-D and F+avg.-D-con.) and replace RNN with a Single-frame Classifier (SFC). Then, the same loss function in our method is used to train the single-frame classifier using standard back-propagation. These baselines highlight the importance of RNN.
- Maximum-Probability Single-frame Classifier (SFC). We replace the average-attention with the maximum accident anticipation probability over all objects as the accident anticipation probability at each frame. We either use only the object feature (max.-D). These baselines highlight the effect of using RM vs. VGG.
- Frame-based Single-frame Classifier (SFC). We start from Frame-based RNN (F) and replace RNN with a single-frame classifier. Then, the same loss functions in our method are used to train the single-frame classifier using standard back-propagation. This baseline also highlights the importance of RNN.

We first evaluate all methods using VGG appearance feature and IDT motion feature separately to compare the effectiveness of both features. Next, we combine the best VGG variant with the best IDT variant using late-fusion to take advantage of both appearance and motion features.

4.5 Results

We report the Average Precision (AP) of all methods in Table. 1, and discuss our results below.

- For VGG feature,
  - RNN consistently outperforms SFC. Without using dynamic attention, VGG+RNN (the first row in Table. 1) consistently outperform VGG+SFC (the second row in Table. 1) by at most 23.80% in AP (see avg.-D).
  - Object observation improves over full-frame observation. Both VGG+RNN+ avg.-D and VGG+RNN+F+avg,-D-con. outperform VGG+RNN+F.
  - Dynamic Spatial-attention further improves over RNN. Both dynamic attention F+D-sum and F+D-con. outperform average attention (VGG+RNN+ F+avg.-D-con.) by at most 21.02% in AP. Object only dynamic attention (VGG+RNN+D) also outperforms object only average attention (VGG+RNN+ avg.-D) by 3.28% in AP.
- For IDT feature,
  - IDT is a powerful full-frame feature. IDT's frame-based SFC outperforms VGG+SFC+F and VGG+RNN+F by at least 2.26% in AP.
  - RNN is worse than SFC. This is different from our finding using VGG feature. We believe that when the long IDT feature (6400 dimensions) is forced to embedded into 512 dimensions for RNN encoding, some discriminative information might be lost.
- For RM feature,
  - RM+SFC+max.-D (49.36%) is worse than VGG+SFC+max.-D.(66.05%) It shows just detecting objects and estimating their motion direction can not compare with VGG.

Table 1. Accident anticipation accuracy in Average Precision (AP). avg. stands for average. con. stands for concatenate. All methods are efined in Sec. 4.4

	No Dynamic Attention			Dynamic Attention		
Type	F	avg.-D	F+avg.-D-con	D	F+D-sum	F+D-con
VGG+RNN	51.89%	**64.88%**	**52.51%**	68.16%	68.21%	**73.53%**
VGG+SFC	46.61%	41.08%	49.01%	—	—	—
IDT+RNN	49.73%	—	—	—	—	—
IDT+SFC	**54.15%**	—	—	—	—	—

– We combine the best IDT method (IDT+SFC+F) with the best VGG method (VGG+RNN+F+D-con.) into Fused-F+D-con. In particular, we fuse the anticipation probability outputs of both methods using equal-weight-summation. This fused method achieves the best 74.35% AP.

We plot the precision v.s. recall curves of the combined method (Fused-F+D-con.), the best VGG method (VGG+RNN+F+D-con.), and many full-frame baselines in Fig. 4-Left.

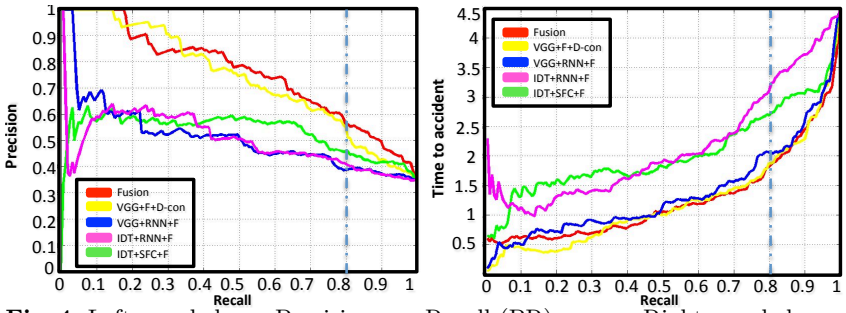


Fig. 4. Left panel shows Precision v.s. Recall (PR) curves. Right panel shows average Time-to-Accident v.s. Recall (ToAR) curves. As indicated by the dash-vertical-lines in both panels, our fused method on average anticipates the accident 1.8559 seconds before it occurs with 56.14% precision and 80% recall. Note that, compared to other methods, IDT+RNN+F has longer ToA but much worse precision. This implies IDT+RNN+F has a much higher false alarm rate.

Average time-to-accident (ToA). We report the average time-to-accident v.s. recall curves of the combined method (Fused-F+D-con.), the best VGG method (VGG+RNN+ F+D-con.), and many full-frame baselines in Fig. 4-Right. Our fused method on average anticipate the accident 1.8559 seconds before it occurs with 56.14% precision and 80% recall. We report the performance at 80% recall, since our system aims at detecting most true accidents. Note that, compared to other methods, IDT+RNN+F and IDT+SFC+F has longer ToA but much worse precision. This implies that they have much higher false alarm rates.

5 Conclusion

We propose a Dynamic-Spatial-Attention RNN model to anticipate accidents in dashcam videos. A large number of dashcam videos containing accidents have

been harvested from the web. In this challenging dataset, our proposed method consistently outperforms other baselines without attention or RNN. Finally, our method fusing VGG appearance and IDT motion features can achieve accident anticipation about 2 seconds before it occurs with 80% recall and 56.14% precision. We believe the accuracy can be further improved if other sensory information such as GPS or map information can be utilized.

Acknowledgements. We thank Industrial Technology Research Institute for their support.

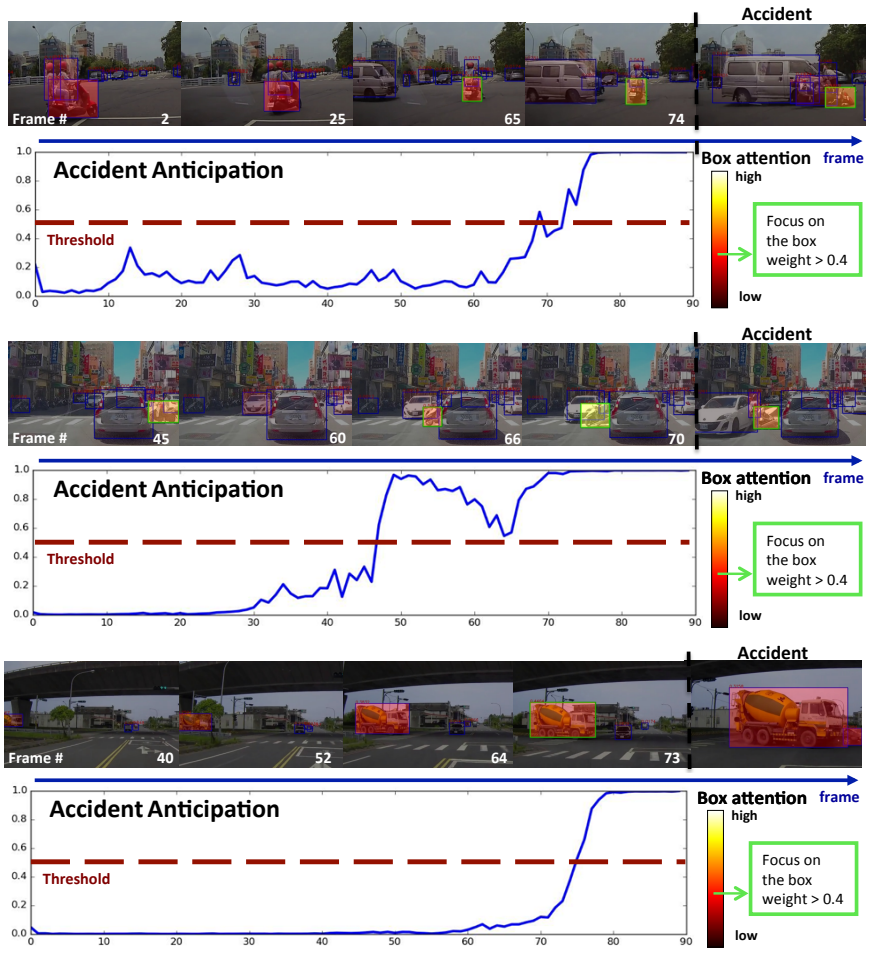


Fig. 5. Typical accident anticipation examples. In each example, we show the sampled frames overlaid with the attention weights (i.e., a value between zero and one) on the object bounding boxes, where yellow, red, and dark indicate high, medium, and low attention, respectively. When the outline of a box turns green, this indicates that its attention is higher than 0.4. On the bottom row, we visualize the predicted confidence of anticipated accident. The threshold is set to 0.5 for visualization.

References

1. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
2. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV. (2013)
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
4. Google Inc.: Google self-driving car project monthly report (2015)
5. N. Highway Traffic Safety Administration: 2012 motor vehicle crashes: overview (2013)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation (1997)
7. Jain, A., Singh, A., Koppula, H.S., Soh, S., Saxena, A.: Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In: ICRA. (2016)
8. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV. (2011)
9. Hoai, M., De la Torre, F.: Max-margin early event detectors. In: CVPR. (2012)
10. Lan, T., Chen, T.C., Savarese, S.: A hierarchical representation for future action prediction. In: ECCV. (2014)
11. Kitani, K.M., Ziebart, B.D., Bagnell, J.A.D., Hebert , M.: Activity forecasting. In: ECCV. (2012)
12. Yuen, J., Torralba, A.: A data-driven approach for event prediction. In: ECCV. (2010)
13. Walker, J., Gupta, A., Hebert, M.: Patch to the future: Unsupervised visual prediction. In: CVPR. (2014)
14. Wang, Z., Deisenroth, M., Ben Amor, H., Vogt, D., Schölkopf, B., Peters, J.: Probabilistic modeling of human movements for intention inference. In: RSS. (2012)
15. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. PAMI **38** (2016) 14–29
16. Koppula, H.S., Jain, A., Saxena, A.: Anticipatory planning for human-robot teams. In: ISER. (2014)
17. Mainprice, J., Berenson, D.: Human-robot collaborative manipulation planning using early prediction of human motion. In: IROS. (2013)
18. Berndt, H., Emmert, J., Dietmayer, K.: Continuous driver intention recognition with hidden markov models. In: Intelligent Transportation Systems. (2008)
19. Frohlich, B., Enzweiler, M., Franke, U.: Will this car change the lane? - turn signal recognition in the frequency domain. In: Intelligent Vehicles Symposium (IV). (2014)
20. Kumar, P., Perrollaz, M., Lefvre, S., Laugier, C.: Learning-based approach for online lane change intention prediction. In: Intelligent Vehicles Symposium (IV). (2013)
21. Liebner, M., Baumann, M., Klanner, F., Stiller, C.: Driver intent inference at urban intersections using the intelligent driver model. In: Intelligent Vehicles Symposium (IV). (2012)
22. Morris, B., Doshi, A., Trivedi, M.: Lane change intent prediction for driver assistance: On-road design and evaluation. In: Intelligent Vehicles Symposium (IV). (2011)
23. Doshi, A., Morris, B., Trivedi, M.: On-road prediction of driver's intent with multimodal sensory cues. IEEE Pervasive Computing **10** (2011) 22–34

24. Trivedi, M.M., Gandhi, T., McCall, J.: Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. IEEE Transactions on Intelligent Transportation Systems **8** (2007) 108–120
25. Jain, A., Koppula, H.S., Raghavan, B., Soh, S., Saxena, A.: Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In: ICCV. (2015)
26. Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville., A.: Describing videos by exploiting temporal structure. In: ICCV. (2015)
27. Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. arXiv preprint arXiv:1502.03044 (2015)
28. Mnih, V., Heess, N., Graves, A., kavukcuoglu, k.: Recurrent models of visual attention. In: NIPS. (2014)
29. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. In: ICLR. (2015)
30. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: ECCV. (2008)
31. Leibe, B., Cornelis, N., Cornelis, K., Gool, L.V.: Dynamic 3d scene analysis from a moving vehicle. In: CVPR. (2007)
32. Scharwchter, T., Enzweiler, M., Roth, S., Franke, U.: Efficient multi-cue scene segmentation. In: GCPR. (2013)
33. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR. (2012)
34. Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset. In: CVPR Workshop on The Future of Datasets in Vision. (2015)
35. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063 (2012)
36. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proceedings of the IEEE **78** (1990) 1550–1560
37. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPr. (2005)
38. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. (2014)
39. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from tensorflow.org.