Learning Spatio-temporal Features via 3D CNNs to Forecast Time-To-Accident

Taif Anjum[1], Louis Chirade[2], Beiyu Lin[2], and Apurva Narayan[1]
[1] *Department of Computer Science, University of British Columbia, Kelowna, Canada*
[2]*Department of Computer Science, University of Nevada, Las Vegas , USA*
Taif.Anjum@ubc.ca, Louis.Chirade@edu.esiee.fr, Beiyu.Lin@unlv.edu, Apurva.Narayan@ubc.ca

Keywords: Collision Avoidance Systems, Deep Learning, Spatio-temporal feature learning, 3D Convolutional Neural Networks

Abstract: Globally, traffic accidents are one of the leading causes of death. Collision avoidance systems can play a critical role in preventing accidents or minimizing their severity. Time-to-accident (TTA) is considered the principal parameter for collision avoidance systems allowing for decision-making in traffic, dynamic path planning, and accident mitigation. Despite the importance of TTA, the literature has insufficient research on TTA estimation for traffic scenarios. The majority of recent work focuses on accident anticipation by providing a probabilistic measure of an immediate or future collision. We propose a novel approach of time-to-accident forecasting by predicting the exact time of the accident with a prediction horizon of 3-6 seconds. Leveraging the Spatio-temporal features from traffic accident videos, we can recognize accident and non-accident scenes while forecasting the TTA. Our method is solely image-based, using video data from inexpensive dashboard cameras allowing for an accessible collision avoidance tool that can be integrated with any vehicle. Additionally, we present a regression-based 3D Convolutional Neural Network (CNN) architecture that requires significantly less parameters compared to its counterparts making it feasible for real-time usage. Our best models can estimate TTA with an average prediction error of 0.30s on the Car Crash Dataset (CCD) and 0.79s on the Detection of Traffic Anomalies (DoTA) dataset elucidated by the longer prediction horizon. Our comprehensive experiments suggest that spatio-temporal features from sequential frames perform significantly better than only spatial features extracted from static images.

1 INTRODUCTION

According to a global report on road safety, traffic accidents account for over 3,700 daily deaths which add up to 1.35 million deaths annually (World Health Organization, 2018). To combat this, automakers are including collision avoidance features as part of their Advanced Driver Assistance Systems (ADAS). Studies show that Collision avoidance features reduced front-to-rear crashes of cars by 50%, trucks by 41% and crashes with injuries by 56%, (The Insurance Institute, 2022). Time-to-accident (TTA) is considered the principal parameter for collision avoidance systems allowing for better decision-making in traffic, dynamic path planning, and accident mitigation (Saffarzadeh et al., 2013; Manglik et al., 2019). We define TTA as the the time duration before collision between two (or more) road users is inevitable. Despite the importance of TTA, recent studies focus on the early anticipation of accidents but fail to estimate or predict the TTA. One study (Suzuki et al., 2018) proposes an adaptive loss function for early risk anticipation and a Quasi-Recurrent Neural Network (QRNN) to learn the Spatio-temporal features. Their model generates the probability of a possible accident with a prediction horizon of 3 seconds, however, it does not predict the time of the accident. Similarly, (Bao et al., 2020) proposes a Graph Convolutional Network (GCN) with RNN cell to learn Spatio-temporal features followed by Bayesian Neural Network (BNNs) to generate accident probability. (Chan et al., 2016) proposed a Dynamic-Spatial Attention Recurrent Neural Network (DSA-RNN) for anticipating accidents from dashboard camera videos. Such accident anticipation technologies intend to previse an accident before it takes place, however, only being able to anticipate or detect a possible accident is not enough. For effective decision-making, path planning, and collision avoidance, we need a temporal estimation for the accident.

To bridge this gap in existing research, we pro-

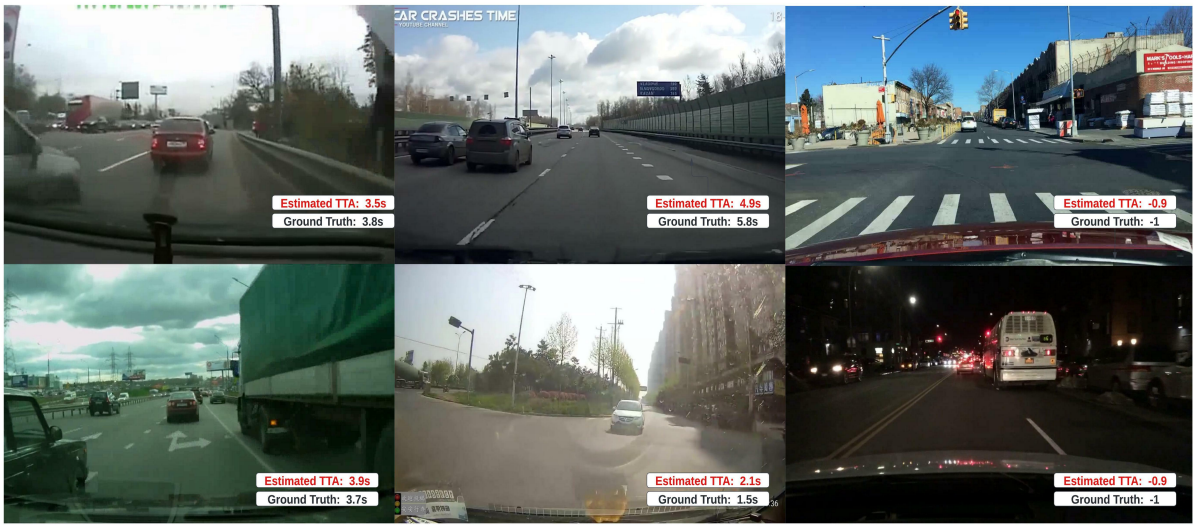

Figure 1: Prediction samples from our test data. First two columns represent accident scenes and the third column represents non-accident scenes.

pose to estimate the exact time of the accident with a prediction horizon of 3-6 seconds. Our approach utilizes inexpensive dashboard cameras and does not require any additional sensors. This can allow any vehicle to have a collision avoidance tool for as little as 50$ whereas existing collision avoidance systems can cost over $2,500 and an extra $3,000 on average for repair in-case of an accident (Wardlaw, 2020). Figure 2 shows samples from our test data annotated with the estimated and ground truth value. We select two publicly available datasets, namely, the Car Crash Dataset (CCD) and Detection of Traffic Anomaly (DoTA) to test our proposed method. Both these datasets have frame-wise annotations indicating the exact frame where the accident began. We use these annotations to calculate the TTA value for each video which is our ground truth. The annotations are in the form of binary labels associated with each frame indicating if the frame is an accident (positive) or non-accident (negative) frame. The first positive label indicates the beginning of the accident. As there are 10 frames per second (fps) in each video, each frame represents 0.1 seconds. Therefore, the first positive label represents the exact time step where the accident began. For example, if a video is 5 seconds long then there are 50 frames (10 fps) in total. If the first positive label is on the 31st frame, then the TTA for the video will be 3.1 seconds. Using this methodology, we label each accident video with its measured TTA value. For non-accident videos, we label them as -1, indicating an infinite TTA. This allows our model to recognize both accident and non-accident scenes while estimating the TTA as shown in figure 1. Our approach is to predict the TTA of each video using only the first N-frames. Given the fact that we require a model that is both efficient and highly accurate for real-world implementation, we present a 3D CNN regression architecture that is efficient, lightweight and high performing. We test our architecture with varying spatial resolution and temporal depth to identify the role they play in the model's performance. Our best model achieves a mean absolute error (MAE) of 0.30 seconds with only 8 frames from the CCD dataset with an average prediction horizon of 3 seconds. Our model obtains a MAE of 0.79 seconds on the DoTA dataset with the first 16 frames which has a prediction horizon of 6 seconds. Furthermore, our model can recognize accident and non-accident scenes with 100% accuracy across both datasets. Our comparative analysis showed our model outperforms an extensive list of state-of-the-art CNN architectures. The contribution of this paper is as follows - 1) Novel approach of forecasting time-to-accident and accident classification. 2) Presents a 3D CNN architecture that demonstrates state-of-the-art performance with fewer parameters. 3) Demonstrates the superiority of spatio-temporal features over only spatial features for the proposed task. 4) Analyzes the role of spatial resolution and temporal depth for our specific application.

2 Related Work

Time-to-Accident (TTA) refers to the period before two (or more) objects will collide as defined by (Hayward, 1972). They proposed time-measure-to-collision (TMTC) as a measure of danger to an accident which was estimated using the velocity and distance between vehicles. The study by (Jiménez et al., 2013) builds on that work and proposes a

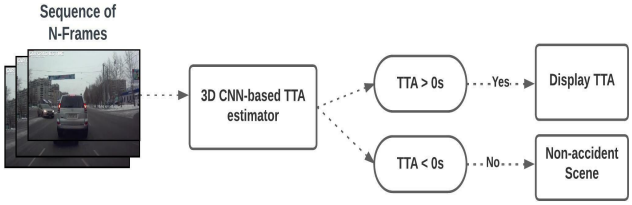

Figure 2: Time-To-Accident (TTA) prediction pipeline. If the estimated value is less than zero the scene will be considered a non-accident scene otherwise there is a risk of accident.

more computationally efficient and accurate calculation of TTA between two vehicles colliding at constant speed along a straight path. These studies provide mathematical equations for TTA estimation, however, they do not discuss the application side of it. With the emergence of deep learning and advancements in computer vision, TTA calculation and estimation techniques have evolved. TTA estimations can now be done using object detection, tracking, instance segmentation, and trajectory prediction (Tøttrup et al., 2022). TTA is not only limited to automotive vehicles but critical component for navigation in robotics, vessels, and Unmanned Aerial Vehicles. One study (Tøttrup et al., 2022) introduced a framework that utilizes object detection to detect objects around a vessel and generate bounding boxes which are used to track the objects and produce velocity vectors. In robot navigation TTA is estimated by tracking the trajectory and measuring the velocity of surrounding objects or pedestrians (Bewley et al., 2016; Sharma et al., 2018). A study by (Manglik et al., 2019) proposes time-to-near collision prediction between a suitcase-shaped robot and nearby pedestrians using a monocular camera and lidar sensors. Combined with the video and lidar data they predict when pedestrians will be within one meter of the robot from a sequence of frames. The aforementioned approaches rely on high-quality sensors and depth imaging devices to detect and track objects. However, sensor noise and error in object detection can easily cause such approaches to fail. Inaccuracies in-depth estimation or 2D bounding box detection can result in significant changes in velocity resulting in inaccurate trajectory estimates (Manglik et al., 2019).

Spatio-temporal feature learning is necessary for video tasks including action recognition, scene recognition, accident anticipation, pose estimation, and more. Spatio-temporal features provide the motion information from a sequence of images which can be used to recognize activities in sequential data such as videos. Before the success of CNN and its variants, Spatio-temporal features were handcrafted using algorithms such as SIFT-3D, HOG-3D and Motion Boundary Histogram. However, in recent years automated Spatio-temporal feature learning has gained tremendous success due to the emergence of Deep Learning (DL) algorithms. Spatio-temporal feature learning in the DL domain can be split into two categories, (i) two-stream method where spatial and temporal features are extracted separately and then fused (ii) Spatio-temporal kernels are applied directly to the videos. Combinations of CNNs and RNNs belong to the first category. Like the work by (Yue-Hei Ng et al., 2015) that uses GoogleLet to extract the spatial features, an LSTM for temporal features and fuses them before feeding to the fully connected layers. 3D CNNs fall into the second category where 3D convolutional kernels are directly applied sequences of images or videos to capture both the spatial and temporal features.

Accident anticipation methods seek to predict an accident before it takes place. For vision-based approaches, we require a first-person or ego-centric view such as the view from dashboard cameras. Several works proposed Spatio-temporal learning frameworks along with car accident datasets comprised of videos from dashboard cameras. One study (Suzuki et al., 2018) proposed a Near-miss Incident DataBase (NIDB) for near-miss traffic accident anticipation. To evaluate their dataset they present an Adaptive Loss for Early Anticipation (AdaLEA) and Quasi-Recurrent Neural Network (QRNN). AdaLEA is a loss function that aims to learn earlier anticipation as training progresses. The QRNN is an efficient alternative to LSTM for temporal feature learning. Their system outputs a probability of a possible accident in the future and can anticipate a near-miss incident or accident about 3 seconds in advance. Similarly, (Bao et al., 2020) proposes a Graph Convolutional Network (GCN) with RNN cell to learn Spatio-temporal features followed by Bayesian neural network (BNNs) to generate accident probability. (Chan et al., 2016) proposed a Dynamic-Spatial Attention Recurrent Neural Network (DSA-RNN) for anticipating accidents from dashcam videos. They use object detection to identify candidate objects and incorporate spatial and temporal features from sequential images using their model. The aforementioned works can anticipate a possible accident, however, they fail to predict the exact time of the accident.

3 Datasets

Our objective is to forecast the time-to-accident for automotive vehicles based on only visual data. For this, we require video data from the driver's field of view such as videos from dashboard cameras. However, such data is scarce in the literature. To the best

of our knowledge, there are four publicly available datasets, namely, Dashcam Accident Dataset (DAD) (Chan et al., 2016), AnAn Accident Detection (A3D) (Yao et al., 2019), Detection of Traffic Anomaly (DoTA) (Yao et al., 2022) and the Car Crash Dataset (CCD) (Bao et al., 2020). We exclude DAD from our experiments to the lack of annoatations. DoTA is an extension of A3D where the authors increased the number of accident clips to 4,677 and added more categories of accidents.

DoTA, A3D and the Car Crash Dataset (CCD) is constructed from accident videos collected from YouTube. The videos are structured such that the accident occurs within the last two seconds of the video. For our particular application of forecasting time-to-accident, we require annotations indicating the start time of accidents. As CCD and DoTA provide such annotations, they are appropriate datasets for our experiments. In addition to accident clips, we also require non-accident scenes to create a model robust to false alarms. CCD includes 3,000 randomly sampled normal driving video clips from the Berkley Driving Dataset (BDD100K) (Yu et al., 2020) which are also recorded via dashboard cameras. These clips are used as negative or non-accident samples for our experiments. Each non-accident clip is 5 seconds long with 10 frames per second.

Name	Positive Samples	Length (in s)	Fps	Annotations
DAD	1,130	5	20	No
A3D	1,500	2.3-20.8	10	Yes
DoTA	4,677	2.3-20.8	10	Yes
CCD	1,500	5	10	Yes

Table 1: Original size and characteristics of traffic accident datasets with egocentric view.

3.1 Pre-processing

The videos in both datasets have annotations indicating if the ego vehicle was involved in the crash or not. Ego vehicle is defined as the subject whose behavior is of primary interest. In our case, the vehicle on which the camera is mounted will be referred to as the ego vehicle. The videos where the ego vehicle was not involved in the crash included accidents between other road users which were captured by the ego vehicle's dashboard camera. The ego vehicle being involved in the crash means there was a direct collision between the ego vehicle and other road user(s). These two scenarios are illustrated in figure 3. Our goal is to develop a system that will warn the ego vehicle of a potential danger to itself. Considering that, we remove the videos where the ego vehicle is not involved in the crash.

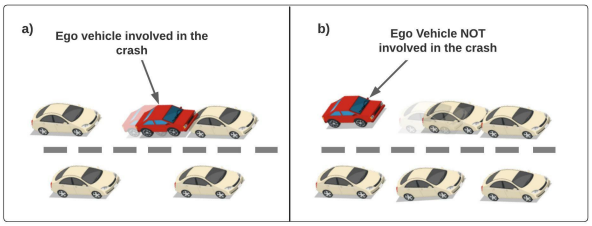

Figure 3: The vehicle of interest (i.e., ego vehicle) is the red car. a) Scenario where the ego vehicle is involved in the crash. b) Scenario where the ego vehicle witnesses an accident.

The final data processing step was data augmentation. To perform the augmentation, we deconstructed the videos and applied augmentation to each frame before re-compiling them as a video. Various combinations of augmentation techniques were applied to the data such as rotation, horizontal flip, gaussian blur, gaussian noise, scaling, and random crop. Out of which, applying only horizontal flip provided the best results. Table 2 shows how the size of the datasets evolved as data was processed.

Name	Original	Post-processing	Post-augmentation
CCD	4,500	3,801	6,841
DoTA	7,677	5,353	9,634

Table 2: Size of the datasets at each stage of processing.

4 Methodology

4.1 Forecasting Time-To-Accident

The aforementioned datasets, CCD and DoTA have frame-wise annotations indicating the exact time step at which the accident begins. We utilize these annotations to generate labels to train our time-to-accident prediction model. For a given accident clip, we have M binary labels $\{label_1, label_2, label_3 ... label_M\}$, where M is the total number of frames in the video. The binary labels are associated with each frame of a video and each frame represents 0.1 seconds as there are 10 fps. The first positive label in this sequence is the time step where the accident has started or is inevitable according to the annotations. We denote the first accident label in this sequence of labels as T, then our ground truth time-to-accident (TTA) is t = T/10 seconds. Figure 4 depicts a clip from the CCD dataset and shows how we utilize the annotations to calculate our TTA value. As a pre-processing step, we label each accident clip with the ground truth TTA value and non-accident clips with the value -1. Given a sequence of N consecutive frames $\{F_1, F_2, F_3 ... F_N\}$, our goal is to use this sequence as history to estimate the time-to-accident. If the estimation is less than 0, we can say that there will be no accidents in the near

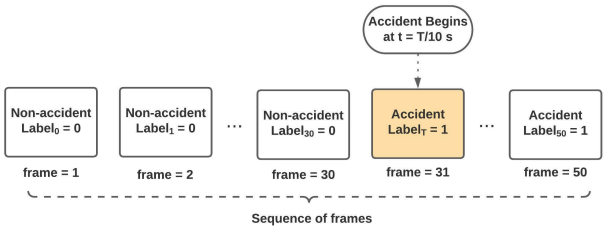

Figure 4: Sample clip from the Car Crash Dataset

future. As we are predicting a continuous variable, we formulate the problem of TTA as a regression problem.

4.2 Proposed 3D CNN Architecture

To predict time-to-accident from a sequence of frames, we require both the spatial and temporal features of those frames. A standard 2D-CNN can only extract the spatial features without considering the temporal features. Several architectures exist in the literature for Spatio-temporal feature learning, e.g., N-stream VGG (Manglik et al., 2019), CNN-RNN, CNN-LSTM, 3D-CNN (Tran et al., 2015) etc. Out of which, only 3D-CNN provides an end-to-end solution for learning from videos. 3D-CNN has also been shown to outperform its counterparts for various video-based tasks. However, 3D CNN architectures are notorious for their large training times, high inference latency, and require significant computational resources. As our approach requires both high performance and efficiency for real-time usage, we perform rigorous experiments to come up with the parameters for a 3D CNN architecture for our application that is efficient and does not compromise on performance. For example, our proposed architecture has about 1.8 million parameters for image sizes of 72x128 and temporal depth of 4 whereas C3D has over 128 million parameters. Our experiments show that even with significantly fewer parameters our architecture outperforms C3D. As shown in figure 5, our 3D CNN architecture has 6 3D convolution layers, 3 max-pooling layers, and 1 fully connected layer. We conducted experiments with varying kernel dimensions and found the convolution kernel size of 3x3x3 to perform best, this aligns with the findings of the systemic study (Tran et al., 2015) on 3D CNN architectures. For the pooling kernels, a dimension of 3x3x3 performed best for our application. We use dropouts at regular intervals to prevent over-fitting and batch normalization before feeding to the output layer. Relu is used as the activation function for all layers except the output layer where a linear activation function is used. Our network is trained with the following loss function.

$$Loss_{mse} = \frac{1}{2}||t_{gt} - f(l_1, l_2, ..., l_N)||^2 \qquad (1)$$

The $Loss_{mse}$ is the mean squared loss between our ground truth time t_{gt} and predicted time $f(l_1, l_2, ..., l_N)$. The loss is optimized using the Adam optimizer, with a batch size of 64 and an initial learning rate of 0.001. We run our experiments for 200 epochs and use two callbacks, early stopping, and a learning rate scheduler. Early stopping prevents overfitting by halting training when validation loss stops improving for 40 epochs. The learning rate scheduler reduces the learning rate by a factor of 0.20 if the validation loss does not improve for 10 epochs. As discussed in the previous section, we increase our training data by two folds through horizontal flip transformation.

5 Experiments & Results

Our objective is to determine if the Spatio-temporal information from the first N frames can be used as a history to forecast the TTA and recognize an accident scene. We use a maximum of 10 frames to make our approach suitable for real-world scenarios. Given the prediction horizon is between 30 to 60 frames, using more than 10 frames would not allow the driver enough time to take action to prevent the accident. In addition to varying the temporal depth, we resize the frames to two different resolutions, 36x64 and 72x128. This was motivated by studies such as (Gaurav et al., 2021) which showed both temporal depth and spatial resolution can impact the performance of 3D CNNs for video-based tasks such as scene recognition. All experiments in this section were conducted on a system with 11th Generation Intel Core i7-11800H, 32GB RAM, and Nvidia GeForce RTX 3080 GPU with 16GB memory. The images and labels were normalized before being fed into the networks.

We compare our proposed architecture directly to C3D due to their similar characteristics. Our implementation of the C3D architecture is identical to the original paper, no fine-tuning was performed as the authors claim their architecture can perform well without fine-tuning regardless of the application. For C3D, the experiments were trained for 250 epochs with a batch size of 64, an initial learning rate of 0.001, MSE as the loss function, and a Stochastic gradient descent optimizer. Additionally, two callbacks were used, namely, Early stopping and Learning rate scheduler.

Table 3 and figure and 4 show our experimental results on the datasets CCD and DoTA respectively. The reported results show the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) over an average of 5 runs. Both the models were trained from scratch and no pre-trained weights were used.

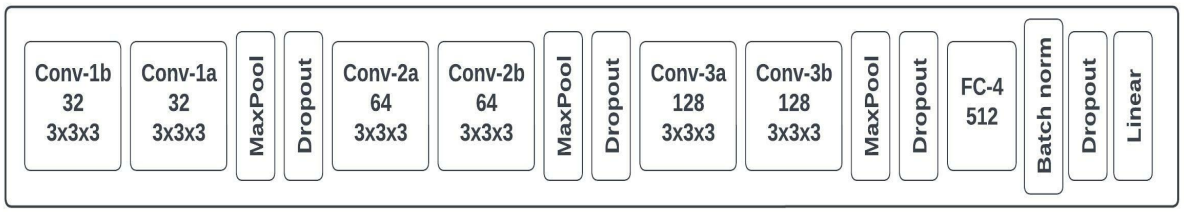

Figure 5: Proposed 3D CNN Architecture, "32" refers to the number of filters while "3x3x3" refers to the kernel dimensions

For C3D, only using 2-6 frames produce poor results compared to our architecture across both datasets. Along with temporal depth, the spatial resolution also influenced performance. In table 3, we can notice an improvement in RMSE and MAE for both architectures as the resolution is increased from 36x64 to 72x128. A similar trend can be noticed for DoTA in table 4, however, the improvement is less significant. This may be because the videos in CCD are of low quality compared to DoTA. Hence, reducing the resolution further causes more loss of spatial information for CCD compared to DoTA. We also notice that both the RMSE and MAE are significantly higher for DoTA which is since DoTA has significantly longer videos with the longest video beings 18.4 seconds long. DoTA also has a higher average prediction horizon of 6 seconds compared to 3.4 seconds for CCD. For some of the experiments, our proposed model performs very similarly to C3D, however, it is important to note that our model is significantly faster and requires much less computational resources compared to C3D. We also note that there isn't a piecewise monotonic relation between temporal depth and mean absolute error. Our findings align with the studies (Manglik et al., 2019; Kayukawa et al., 2019) where they conclude that length of temporal history does not necessarily increase or decrease error in prediction for applications such as trajectory prediction or robot-pedestrian collision. An interesting observation was that our best model predicted positive TTA value for all accident videos and negative values all for non-accident videos. This means the model was able to recognize accident and non-accident scenes with 100% accuracy. To compare the results, we performed separate experiments for binary accident classification using the C3D architecture and categorical cross-entropy loss function. For the same temporal depth and spatial resolution, the classification accuracy was 91% for DoTA and 93% for CCD. This suggests that our approach is not only an effective method for TTA estimation but can also be used as an effective accident classification strategy.

		OURS	OURS	C3D	C3D
Resolution	Frames	RMSE	MAE	RMSE	MAE
36x64	2	0.906	0.386	1.229	0.802
36x64	**4**	**0.872**	**0.319**	1.215	0.769
36x64	6	0.885	0.325	1.060	0.614
36x64	8	0.900	0.334	0.902	0.333
36x64	10	0.915	0.337	0.941	0.353
72x128	2	0.858	0.369	1.130	0.703
72x128	4	0.842	0.312	1.127	0.676
72x128	6	0.871	0.340	0.952	0.347
72x128	8	0.822	0.315	0.926	0.338
72x128	**10**	**0.819**	**0.300**	0.990	0.365

Table 3: Results in seconds on CCD with varying Spatial Resolution and Temporal Depth

		OURS	OURS	C3D	C3D
Resolution	Frames	RMSE	MAE	RMSE	MAE
36x64	2	1.545	0.969	2.081	1.610
36x64	4	1.521	0.886	2.047	1.560
36x64	6	1.505	0.831	2.193	1.515
36x64	8	1.496	0.809	1.479	0.811
36x64	**10**	1.498	0.821	**1.467**	**0.801**
72x128	2	1.535	0.879	2.138	1.660
72x128	4	1.510	0.802	2.003	1.413
72x128	6	1.517	0.819	1.903	1.262
72x128	**8**	**1.457**	**0.786**	1.473	0.800
72x128	10	1.490	0.822	1.514	0.834

Table 4: Results in seconds on DoTA with varying Spatial Resolution and Temporal Depth

5.1 Comparative Analysis

We perform comprehensive experiments against state-of-the-art CNN architectures proposed for similar applications to examine the robustness of our proposed method. For a fair comparison, we train and fine-tune the CNN architectures on CCD and DoTA.

1) 2D CNN Architectures: We compare our work against two types of 2D CNN architectures, namely, VGG and ResNet. The work by (Manglik et al., 2019) proposes a multi-stream VGG-16 for robot-pedestrian near collision scenarios. Their model extracts spatial features from N-frames, concatenates them to learn the motion information, and feeds them to a fully connected layer before being fed to the output layer. As

the authors found 6 frames to perform the best, we compare our model against 6-Stream VGG-16. The model was initialized with pre-trained weights from ImageNet similar to the original work. The authors fine-tuned the model on PASCAL VOC as ImageNet does not have a person class. However, as our application is based on traffic accidents and ImageNet contains a vehicle class we skipped the fine-tuning step. Similar to the original work, We used 224x224 RGB images, SGD as the optimizer, and MSE as the loss function. A learning rate of 0.001 and the model was trained for 50 epochs as these parameters performed the best. Additionally, we also implement a single frame VGG-16 as a baseline model to gain some insight into the influence of only spatial features vs Spatio-temporal features. The results in tables 5 and 6 show that 6 frames perform better than 1 frame. This indicates leveraging Spatio-temporal features can provide better performance compared to only using spatial features. However, the standard deviation of residuals (i.e., RMSE) was lower for the single image variant, this may be due to the lower complexity of the model. Our proposed architecture outperforms both VGG variants by a substantial margin.

To diversify our list of 2D architectures we implement a ResNet-8 model proposed for collision avoidance in drones (Loquercio et al., 2018). In the original work, the model is fed a single image and generates a steering angle and a collision probability to recognize and avoid collisions. In our implementation, we replaced the two output layers with a single regression layer that produces the time-to-accident estimation. Similar to the original work, 224x224 greyscale images were used with an initial learning rate of 0.0001, with MSE as the loss function and Adam as the optimizer. Our proposed model outperformed the ResNet-8 model as shown in tables 7 and 8 however, additionally, it performed better than both VGG variants.

2) Video Architectures: A combination of CNN with an RNN variant is typically used for video classification apart from 3D-CNNs due to their computational efficiency. RNN models have different variations such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Initialized RNNs and Convolutional LSTM. For our purpose, we use a GRU-based RNN, GRU was considered over LSTM because it controls the flow of the information, does not require a memory unit, and is better in terms of computational efficiency and performance (Bahmei et al., 2022). For the spatial features, we used an Inception V3 model pre-trained on ImageNet to extract the features from our traffic accident datasets. The features are then fed to a sequence model with two Gated Recurrent Unit layers (GRU) with 16 and 8 neurons respectively followed by a dropout layer and a fully connected layer with 8 neurons, finally all the features are fed to the output layer with a linear activation function. RNN-CNN model performed best with 16 frames and gave the best results after the 3D CNN architectures.

Method	RMSE (s)	MAE (s)
VGG-16 (1 frame)	1.279	0.882
VGG-16 (6 frames)	1.456	0.750
ResNet-8 (1 frame)	1.029	0.637
CNN-RNN (16 frames)	1.078	0.388
C3D (8 frames)	0.902	0.333
OURS (10 frames)	**0.819**	**0.300**

Table 5: TTA estimation on **CCD**: Comparison of our work with single frame VGG-16, multi-stream stream VGG-16 (Manglik et al., 2019), ResNet-8 (Loquercio et al., 2018), CNN-RNN and C3D (Tran et al., 2015)

Method	RMSE (s)	MAE (s)
VGG-16 (1 frame)	1.62	1.46
VGG-16 (6 frames)	1.33	1.35
ResNet-8 (1 frame)	1.64	1.01
CNN-RNN (16 frames)	1.51	0.95
C3D (8 frames)	1.47	0.80
OURS (8 frames)	**1.46**	**0.79**

Table 6: TTA estimation on **DoTA**: Comparison with single frame VGG-16, N-stream VGG-16 (Manglik et al., 2019), ResNet-8 (Loquercio et al., 2018), CNN-RNN and C3D (Tran et al., 2015)

6 Conclusion & Futurework

We propose a novel approach to forecast time-to-accident (TTA) by leveraging Spatio-temporal features extracted from traffic accident videos. Our approach uses inexpensive and easy-to-install dashboard cameras as opposed to expensive depth imaging devices or sensors that require experts for installation. This can allow for easy integration with any vehicle and can be used as a collision avoidance tool. Our approach only uses the first N-frames where N is at most 10 (1 second), this allows the driver enough time to take action to mitigate the risks given the prediction horizon is between 3-6 seconds. Additionally, we present an efficient 3D CNN architecture with significantly fewer parameters compared to state-of-the-art 3D CNN architectures (e.g., C3D) without compromising performance. This can enable our approach to be implemented in real-time scenarios where minimum inference latency, low computational cost, and high accuracy are necessary. Comparing the results of our multi-frame experiments against the single-

frame experiments there is clear evidence that Spatio-temporal features perform better as opposed to using only spatial features. Apart from estimating TTA, our model can also recognize accident and non-accident scenes with 100% accuracy. This can be beneficial for avoiding false alarms in real-time applications. We also notice that there is no clear monotonic relationship between temporal depth and prediction error, our findings align with other studies in the literature as mentioned in the previous section. Apart from the temporal depth our experiments suggest that spatial resolution impacts the predicted outcome. As a part of future work, we will work on the interpretability of our model to analyze the features that impact the prediction error. We also plan to integrate our model with an accident localization framework to detect various road users that pose a collision threat. Furthermore, we will implement our approach in real-world scenarios and assess the feasibility of our solution in real-time.

REFERENCES

Bahmei, B., Birmingham, E., and Arzanpour, S. (2022). Cnn-rnn and data augmentation using deep convolutional generative adversarial network for environmental sound classification. *IEEE Signal Processing Letters*, 29:682–686.

Bao, W., Yu, Q., and Kong, Y. (2020). Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2682–2690.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. *CoRR*, abs/1602.00763.

Chan, F.-H., Chen, Y.-T., Xiang, Y., and Sun, M. (2016). Anticipating accidents in dashcam videos. In *Asian Conference on Computer Vision*, pages 136–153. Springer.

Gaurav, R., Tripp, B., and Narayan, A. (2021). Driving scene understanding: How much temporal context and spatial resolution is necessary? In *Canadian Conference on AI*.

Hayward, J. C. (1972). Near miss determination through use of a scale of danger.

Jiménez, F., Naranjo, J. E., and García, F. (2013). An improved method to calculate the time-to-collision of two vehicles. *International Journal of Intelligent Transportation Systems Research*, 11(1):34–42.

Kayukawa, S., Higuchi, K., Guerreiro, J., Morishima, S., Sato, Y., Kitani, K., and Asakawa, C. (2019). Bbeep: A sonic collision avoidance system for blind travellers and nearby pedestrians. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Loquercio, A., Maqueda, A. I., Del-Blanco, C. R., and Scaramuzza, D. (2018). Dronet: Learning to fly by driving. *IEEE Robotics and Automation Letters*, 3(2):1088–1095.

Manglik, A., Weng, X., Ohn-Bar, E., and Kitanil, K. M. (2019). Forecasting time-to-collision from monocular video: Feasibility, dataset, and challenges. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8081–8088. IEEE.

Saffarzadeh, M., Nadimi, N., Naseralavi, S., and Mamdoohi, A. R. (2013). A general formulation for time-to-collision safety indicator. In *Proceedings of the Institution of Civil Engineers-Transport*, volume 166, pages 294–304. Thomas Telford Ltd.

Sharma, S., Ansari, J. A., Murthy, J. K., and Krishna, K. M. (2018). Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. *CoRR*, abs/1802.09298.

Suzuki, T., Kataoka, H., Aoki, Y., and Satoh, Y. (2018). Anticipating traffic accidents with adaptive loss and large-scale incident db. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3521–3529.

The Insurance Institute, H. S. (2022). Real-world benefits of crash avoidance technologies.

Tøttrup, D., Skovgaard, S. L., Sejersen, J. l. F., and Pimentel de Figueiredo, R. (2022). A real-time method for time-to-collision estimation from aerial images. *Journal of Imaging*, 8(3):62.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.

Wardlaw, C. (2020). Driver assists: What are the costs to buy, insure and repair?

World Health Organization, W. (2018). Global status report on road safety.

Yao, Y., Wang, X., Xu, M., Pu, Z., Wang, Y., Atkins, E., and Crandall, D. (2022). Dota: unsupervised detection of traffic anomaly in driving videos. *IEEE transactions on pattern analysis and machine intelligence*.

Yao, Y., Xu, M., Wang, Y., Crandall, D. J., and Atkins, E. M. (2019). Unsupervised traffic accident detection in first-person videos. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 273–280. IEEE.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702.