Learning Spatio-temporal Features via 3D CNN to Forecast Time-To-Accident for Traffic Collisions

Abstract

Globally, traffic accidents account for over 3,700 daily deaths, equating to 1.35 million deaths annually. Studies show that collision avoidance systems can significantly reduce the probability and intensity of accidents. 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. TTA refers to the period before two (or more) objects will collide. 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 temporal measure for 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. Additionally, we present a novel regression-based 3D Convolutional Neural Network (CNN) architecture for TTA forecasting. Our model achieves a mean absolute error of 0.30s on the Car Crash Dataset (CCD) and 0.80s on the Detection of Traffic Anomalies (DoTA) dataset elucidated by the longer prediction horizon. Furthermore, our model can recognize both accident and non-accident scenes with 100% accuracy. Our comparative analysis showed our proposed architecture outperforms an extensive list of state-of-the-art CNN architectures.

Introduction

According 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; Van Der Horst and Hogema 1993). TTA or time-to-collision (TTC) was first defined by (Hayward 1972) as the the time duration before two (or more) objects collide.

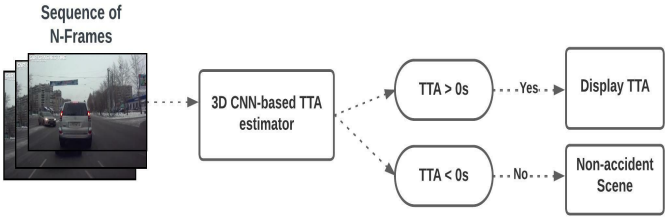

Figure 1: Time-To-Accident (TTA) prediction pipeline. If the estimated value is less than zero the scene will be considered a non-accident scene otherwise there is a risk of accident.

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, Yu, and Kong 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 propose to estimate the exact time of the accident with a prediction horizon of 3-6 seconds. 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

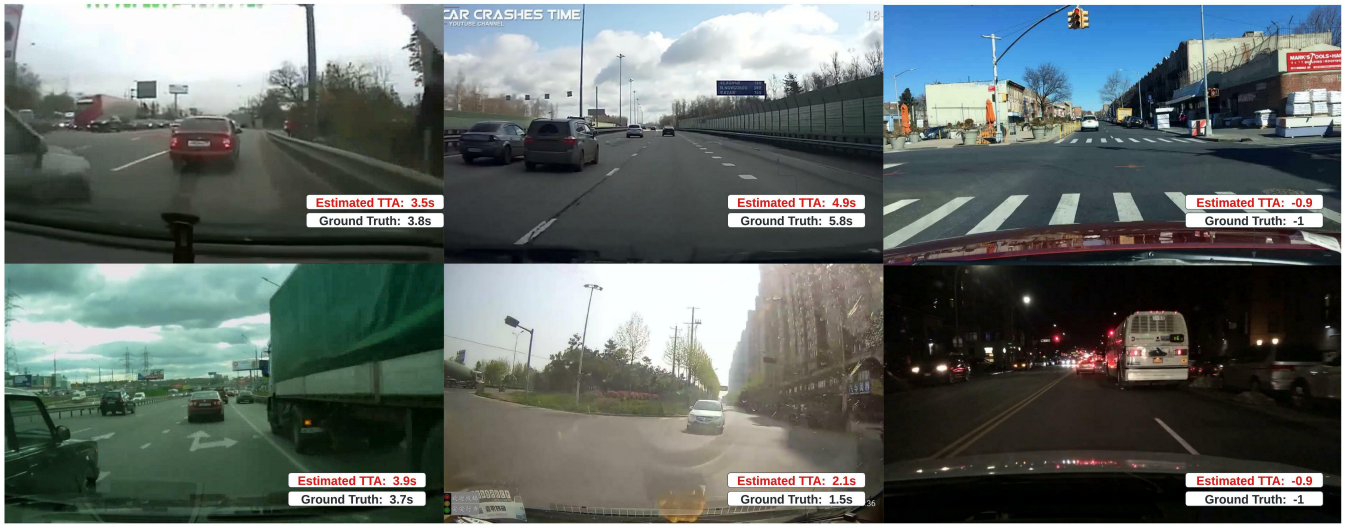

Figure 2: Prediction samples from our test data. First column is from CCD, second column is from DoTA and the third column is the non-accident scenes from BDD100k (Yu et al. 2020)

Dataset (CCD) and Detection of Traffic Anomaly (DoTA) to test our proposed method. Both these datasets have frame-wise annotations indicating the exact time step at which the accident began. 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. To learn the Spatio-temporal features and forecast TTA, we propose a novel 3D CNN regression architecture. 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 0.31 seconds with only 8 frames from the CCD dataset with an average prediction horizon of 3 seconds. Our model obtains a MAE of 0.82 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.

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. Another study (Miller and Huang 2002) proposes a simplified calculation by using the initial position of the vehicles, their speed, and direction. The study by (Jiménez, Naranjo, and García 2013) builds on that work and proposes a 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. The velocity vectors are then used to calculate TTA using the equation proposed by (Jiménez, Naranjo, and García 2013). 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; Phillips and Likhachev 2011). 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). To the best of our knowledge, the most relevant to our work is a study by (Manglik et al. 2019). They propose time-to-near collision prediction between a suitcase-shaped robot and nearby pedestrians using a monocular camera. They construct their dataset and measure the distance between the robot and pedestrians using 3D point clouds. When the distance between the robot and pedestrian is less than 1 meter they annotate that time-step as the time-to-near-collision. They propose a multi-stream VGG-16 that concatenates the spatial features from sequential frames to learn the temporal information and predict the time-to-near-collision. Apart from the differences in application and proposed architecture, their work only looks at near-collision scenarios whereas we include both accident and non-accident scenarios making our model more robust to false alarms.

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 (Scovanner, Ali, and Shah 2007), HOG-3D (Klaser, Marszałek, and Schmid 2008) and Motion Boundary Histogram (Dalal, Triggs, and Schmid 2006). 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. Combination 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 (Hochreiter and Schmidhuber 1997) for temporal features and fuses them before feeding to the fully connected layers. Several other studies also proposed similar two-stream approaches and best practices (Wang et al. 2015; Feichtenhofer, Pinz, and Zisserman 2016; Feichtenhofer, Pinz, and Wildes 2017)

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. 3D CNN was first introduced by (Ji et al. 2012) for human action recognition from video data. Their work was the first deep learning based approach for learning spatio-temporal features from multiple sequential frames. Following their work, (Tran et al. 2015) conducted an empirical study to find the best kernel dimensions and introduced the C3D architecture. Their study showed that the C3D architecture can excel at various videos tasks without the need for fine tuning. C3D demonstrates superior performance but it comes at a high computational cost, as an efficient alternative (Tran et al. 2017) proposed a ResNet style architecture named Res3D which is faster and more compact compared to C3D. Both C3D and Res3D are limited to short clips that are at most 16 frames long. This limitation was addressed by (Varol, Laptev, and Schmid 2017) who developed long-term temporal convolution (LTC) architecture that can apply 3D convolution on videos of upto 100 frames in length.

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, Yu, and Kong 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.

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, Yu, and Kong 2020).

DAD consists of high resolution videos with complex road scenes, crowded streets, and diverse accidents. However, the dataset has not included any form of annotations that provide information such as weather conditions, start or end time of the accident, type of collisions, etc. As we specifically require annotations indicating the starting time of the accident, this dataset is not suitable for us.

In A3D, the accidents are categorized according to their type eg., collisions with pedestrians or with vehicles that are either stopped, moving, or stationary. Each video has 10 fps and the lengths of videos from 2.3 seconds to 20.8 seconds as shown in table 1. DoTA is an extension of A3D where the authors increased the size of the dataset to 4,677 video clips and added more categories of accidents. The authors suggest using DoTA instead of A3D as DoTA is an improved version of A3D.

Similar to DoTA and A3D, 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. Both CCD and DoTA have annotations indicating the start time of accidents. This was done by annotating each frame with a binary label indicating an accident (positive) or non-accident (negative) frame. Each frame corresponds to 0.1 seconds as there are 10 fps in the video clips. All videos begin with a normal period of driving before the accident occurs. Hence the first positive frame corresponds to the time step at which the accident began. The annotations were performed with the consensus of multiple human annotators. For our particular application of forecasting time-to-accident, we require annotations indicating the approximate 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. Howver, DoTA and CCD only include accident scenarios. Hence, we collect 3,000 normal driving video clips from the Berkley Driving Dataset (BDD100K) (Yu et al. 2020) as our non-accident data. 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.

Pre-processing

The videos in both the datasets have annotations indicating if the ego vehicle was involved in the crash or not. Ego vehicle is defined as the subject whose behaviour 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.

The videos in the DoTA dataset are categorized into nine types of accidents. For example, a collision with a pedestrian or another vehicle that turns into or crosses a road. Among the nine categories, two categories did not apply to our purpose and were discarded. The first category is where the vehicle loses control and leaves the roadway, such situations can occur due to mechanical failures or poor road conditions. If the vehicle loses control, a time to accident estimation would not help the driver or an autonomous system perform path planning. The other category was categorized as unknown. These videos included scenarios such as a rock or an object suddenly hitting the windshield. Such scenarios occur with very little warning and cannot be anticipated from the initial spatial and temporal features.

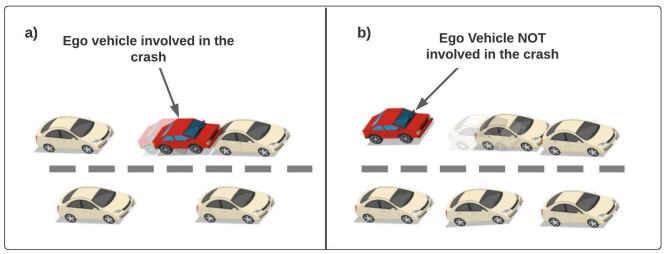

Figure 3: The vehicle of interest (i.e., ego vehicle) is the red car. a) Scenario where the ego vehicle is involved in the crash. b) Scenario where the ego vehicle witnesses an accident.

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. After the addition of 3,000 non-accident videos from the BDK100 (Yu et al. 2020) dataset, CCD and DoTA have 4,500 and 7,677 videos respectively. After the removal of videos that were irrelevant to our application sizes were reduced to 3,801 and 5,353. Data augmentation was only applied to the training data which resulted in 6,080 videos in the training set and 761 in the test set. For DoTA, the training set has 8,564 videos and 1,070 videos in the test set.

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. Original refers to the combined count of accident and non-accident videos. Post-processing count refers to the size after irrelevant videos were discarded. And post-augmentation is the total dataset size after augmentation was applied to the training data.

Methodology

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

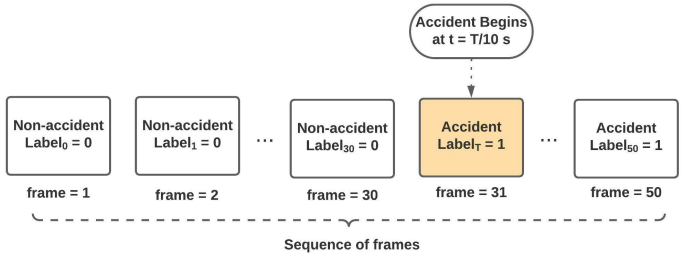

Figure 4: Sample clip from the Car Crash Dataset

$\{F_1, F_2, F_3...F_N\}$, 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 future.

Classification or Regression?

Time-to-accident prediction can be formulated as either a multi-class classification or regression problem. The drawback of multi-class classification approach is that all mispredictions are penalized equally. For example, for the ground-truth of 3.0 seconds, predictions of 0.5 seconds and 2.9 seconds will be penalized equally with the standard categorical cross-entropy loss function. Whereas the former should be penalized more than the latter. One way to approach the problem would be to create a differentiable loss function that penalizes accordingly (Manglik et al. 2019). Another solution would be to formulate it as a regression problem with mean-squared error as the loss function. We decided go with the second solution as it met our requirements without the need for creating a loss function.

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 (Bahmei, Birmingham, and Arzanpour 2022), CNN-LSTM (Shi et al. 2015), 3D-CNN (Tran et al. 2015; 2017) etc. Out of which, only 3D-CNN provides end-to-end solution for learning from videos. 3D-CNN has also been shown to outperform its counterparts for various video-based task (Tran et al. 2017).

In this paper, we propose a novel 3D-CNN architecture for regression to estimate time-to-accident for automotive vehicles. Our network is trained with the following loss function.

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

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 over-fitting 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.

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.

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. 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, Tripp, and Narayan 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.

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, initial learning rate of 0.001, MSE as the loss function and Stochastic gradient descent optimizer. Additionally, two callbacks were used, namely, Early stopping and Learning rate scheduler.

Tables 3 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. As the convolution kernels of both the architectures have a temporal depth of 3 we decided to begin at a

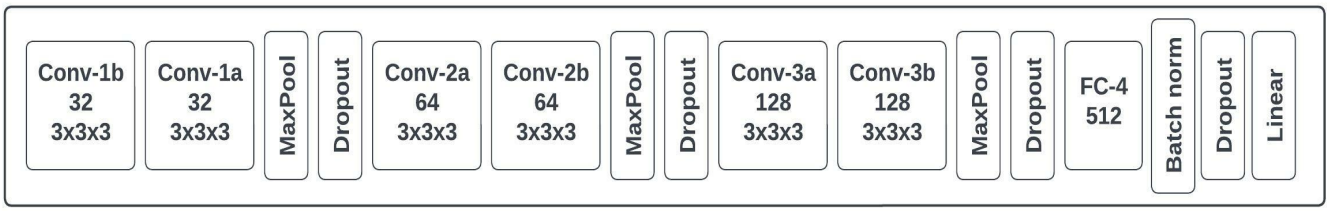

Figure 5: Proposed 3D CNN Architecture, "32" refers to the number of filters while "3x3x3 refers to the kernel dimensions

temporal depth of 4 frames. For C3D, only using 4-6 frames produce poor results compared to our architecture across both datasets. This can be because C3D has a large number of parameters which causes the model to overfit with a lower number of frames. Along with temporal depth, 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. 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 due to the fact that DoTA has significantly longer videos with the longest video beings 18.4 seconds long. DoTA also has a higher average prediction horizon of 5.7 seconds compared to the 3.2 seconds for CCD.

An interesting observation here is 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.

		OURS	OURS	C3D	C3D
Resolution	Frames	RMSE (s)	MAE (s)	RMSE (s)	MAE (s)
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	4	0.842	0.312	1.127	0.676
72x128	6	0.871	0.330	0.952	0.347
72x128	8	0.822	0.305	0.926	0.338
72x128	**10**	**0.819**	**0.300**	0.990	0.365

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

Tables 5 and 6 show our best-performing models' (refer to tables 3 and 4) accuracy when it comes to recognizing an accident or non-accident scene. As we include non-accident videos with the label -1 and accident videos with their respective TTA value, we can conclude that TTA predictions of less than 0 can be considered as non-accident scenes and a TTA of more than 0 will be an accident scene. Our best models were able recognize both scenes with 100% accuracy. This can be very beneficial for avoiding false alarms which is a critical concern for collision avoidance systems.

		OURS	OURS	C3D	C3D
Resolution	Frames	RMSE(s)	MAE(s)	RMSE(s)	MAE(s)
36x64	4	1.512	0.826	2.047	1.560
36x64	6	1.505	0.831	2.193	1.515
36x64	8	1.498	0.821	1.479	0.811
36x64	**10**	1.496	0.809	**1.467**	**0.801**
72x128	4	1.510	0.802	2.049	1.560
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: OURS vs C3D: Results on DoTA with varying Spatial Resolution and Temporal Depth

Dataset	Positive Samples	# Predicted TTA > 0	Accuracy
CCD	161	161	100%
DoTA	470	470	100%

Table 5: Accident recognition accuracy on test data. Positve samples refer to accident videos. If the predicted TTA value is over 0 then it is an accident scene.

Dataset	Negative Samples	# Predicted TTA < 0	Accuracy
CCD	600	600	100%
DoTA	600	600	100%

Table 6: Non-accident recognition accuracy on test data. Negative samples refer to non-accident videos. If the predicted TTA value is less than 0 then it is a non-accident scene.

Comparative Analysis

We perform comprehensive experiments against state-of-the-art CNN architectures to examine the robustness of our proposed method. For a fair comparison, we train 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) is most relevant

to our work in the literature hence we compare against their proposed architecture. They propose an N-stream VGG-16 that extracts features from N-frames, concatenates them, and feeds them to a fully connected layer before being fed to the regression-based output layer. As the authors found 6 frames to perform the best, we compare our model against 6-Stream VGG-16. Additionally, we also implement the same model with 1 frame to gain some insight into how the temporal information is affecting the model. The model was initialized with pre-trained weights from ImageNet similar to the original work. The authors fine-tuned the model on PASCAL VOC (Everingham et al. 2010) 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. The results in tables 7 and 8 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 grey scale images were used with an initial learning rate of 0.0001, 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) (Hochreiter and Schmidhuber 1997), Gated Recurrent Unit (GRU) (Cho et al. 2014), Initialized RNNs (Le, Jaitly, and Hinton 2015) and Convolutional LSTM (Shi et al. 2015). 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, Birmingham, and Arzanpour 2022; Chung et al. 2014). 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. The implementation was adapted from (Pual 2021). 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 7: TTA estimation on **CCD**: Comparison of our work with N-stream VGG (Manglik et al. 2019), ResNet-8 (Loquercio et al. 2018), CNN-RNN (Pual 2021) 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 8: TTA estimation on **DoTA**: Comparison with N-stream VGG (Manglik et al. 2019), ResNet-8 (Loquercio et al. 2018), CNN-RNN (Pual 2021) and C3D (Tran et al. 2015)

Conclusion & Futurework

We propose to forecast time-to-accident (TTA) leveraging spatio-temporal features extracted from traffic accident videos. Comparing the results of our multi-frame experiments (i.e., OURS, C3D, CNN-RNN, 6-Stream VGG-16) against the single-frame experiments (i.e., 1-frame VGG-16 and ResNet-8) there is clear evidence that Spatio-temporal features perform better as opposed to using only spatial features. Additionally, we propose a novel regression-based 3D CNN architecture that outperformed an extensive list of CNN architectures for the task of forecasting TTA.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 suggests that spatial resolution impacts the predicted outcome. As a part of future work, we plan to integrate our model with an accident detection framework and use the Spatio-temporal features from the detected bounding boxes to estimate TTA.

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*, 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*, 136–153. Springer.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Dalal, N.; Triggs, B.; and Schmid, C. 2006. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, 428–441. Springer.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2):303–338.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4768–4777.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1933–1941.

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.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35(1):221–231.

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*, 1–12.

Klaser, A.; Marszałek, M.; and Schmid, C. 2008. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, 275–1. British Machine Vision Association.

Le, Q. V.; Jaitly, N.; and Hinton, G. E. 2015. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*.

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)*, 8081–8088. IEEE.

Miller, R., and Huang, Q. 2002. An adaptive peer-to-peer collision warning system. In *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No. 02CH37367)*, volume 1, 317–321. IEEE.

Phillips, M., and Likhachev, M. 2011. Sipp: Safe interval path planning for dynamic environments. In *2011 IEEE International Conference on Robotics and Automation*, 5628–5635. IEEE.

Pual, S. 2021. Video classification with a cnn-rnn architecture.

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, 294–304. Thomas Telford Ltd.

Scovanner, P.; Ali, S.; and Shah, M. 2007. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, 357–360.

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.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28.

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*, 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*, 4489–4497.

Tran, D.; Ray, J.; Shou, Z.; Chang, S.-F.; and Paluri, M. 2017. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*.

Van Der Horst, R., and Hogema, J. 1993. Time-to-collision and collision avoidance systems.

Varol, G.; Laptev, I.; and Schmid, C. 2017. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence* 40(6):1510–1517.

Wang, L.; Xiong, Y.; Wang, Z.; and Qiao, Y. 2015. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*.

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

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)*, 273–280. IEEE.

Yao, Y.; Wang, X.; Xu, M.; Pu, Z.; Wang, Y.; Atkins, E.; and Crandall, D. 2022. Dota: unsupervised detection of traf-

fic anomaly in driving videos. *IEEE transactions on pattern analysis and machine intelligence*.

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*, 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*, 4694–4702.