DashGuard: Hierarchical Attention for Dashcam Video Accident Detection

Luca Mondonico
Stanford University
lmondo@stanford.edu

Kory Yang
Stanford University
koryyang@stanford.edu

Abstract

We present DashGuard, a hierarchical attention-based deep learning framework for traffic accident prediction from dashcam video footage. Our approach combines spatial feature extraction through EfficientNet with temporal modeling via a novel hierarchical transformer architecture that processes multimodal inputs—RGB frames and optical flow fields—through temporal attention mechanisms. The hierarchical design captures both fine-grained local motion patterns and broader contextual dependencies critical for collision detection. We introduce a crash-focused sampling strategy that concentrates frame selection around critical temporal windows, improving detection of subtle pre-collision cues. Evaluated on the NEXAR Dashcam Collision Prediction Dataset containing 1,500 real-world driving scenarios, our method achieves a ROC-AUC of 0.79 and 72% accuracy, outperforming baseline approaches using standard transformers and CNN-only architectures. Ablation studies demonstrate that incorporating optical flow features and hierarchical temporal modeling both contribute meaningfully to performance, establishing the effectiveness of our multimodal spatio-temporal approach for accident prediction in real-world driving scenarios.

1. Introduction

Early collision prediction from dashcam video is a critical challenge in computer vision with profound implications for autonomous vehicle safety and Advanced Driver Assistance Systems (ADAS). Being able to predict imminent accidents seconds before they occur would timely interventions that could prevent up to 90% of crashes [1].

Traffic accidents claim approximately 1.20 million lives globally each year [9]. Having a robust early collision prediction system could greatly improve public trust in ADAS and self-driving technologies.

Development of collision prediction systems face a few fundamental challenges. The rarity of accidents creates severely imbalanced datasets, making it difficult for models to learn meaningful patterns [3, 4]. Accidents often contain subtle visual cues that are difficult to detect amidst complex scenes [9]. Finally, real-world driving environments involve numerous agents, complex geometries, and are further complicated by varying road and weather conditions, making robust scene understanding a challenging task.

This project aims to explore novel approaches to video classification applied to the NEXAR Dashcam Collision Prediction Dataset [9].

1.1. Problem Statement

The primary problem addressed in this study is the prediction of traffic incidents, specifically accidents or near-misses, from dashcam video footage. Given the inherent complexities and diverse scenarios present in real-world driving, as captured by the NEXAR dataset (e.g., varied lighting, weather conditions, and camera artifacts), the task is to develop a robust system capable of identifying precursors to such critical events.

1.1.1 Inputs

The input to our system is a dashcam video clip. Although the lengths of these clips can vary, they are typically around 40 seconds in duration, consistent with the NEXAR dataset.

1.1.2 Outputs

Video frames sampled from the input video clip are processed to extract visual and motion features (e.g. optical flow). These spatio-temporal features are then fed into a predictive model that outputs a probability score. This probability score represents the likelihood of an accident or near-miss occurring within the input video clip and is then used to classify the video clip into one of two categories:

- **Positive Label:** Indicates an accident or a near-miss.
- **Negative Label:** Indicates normal, uneventful driving.

The performance of this binary classification will be evaluated primarily using the Receiver Operating Characteristic - Area Under Curve (ROC-AUC) metric.

2. Related Work

Development of collision prediction systems faces a few fundamental challenges. The relative rarity of accidents creates imbalanced data sets, making it difficult for models to learn meaningful patterns [4, 3]. Accidents often contain subtle visual cues that are difficult to detect in complex scenarios [9]. Real-world driving environments involve numerous agents in a diverse set of road and weather conditions, which make scene understanding a challenge.

Chan et al. [4] introduced the Dynamic-Spatial-Attention Recurrent Neural Network (DSA-RNN), establishing the paradigm of combining object detection, attention mechanisms, and temporal modeling. This work demonstrated that spatial attention could focus on relevant objects while LSTM-based modeling captured long-range dependencies, achieving 74.35% mean Average Precision for accident localization on the Dashcam Accident Dataset (DAD).

Similarly, Zeng et al. (CVPRW 2017) proposed an agent-centric model: a soft-attention RNN that modeled interactions between an “ego” vehicle and other agents (or static regions). They introduced the EpicFail dataset (3000 internet videos of accidents) and approached accident prediction by jointly localizing when and where a crash might happen [16].

Other models emphasize both appearance and motion. Two-stream CNN architectures (one stream for RGB frames, one for motion data such as optical flow) allow us to encode spatio-temporal cues which may not be available from appearance alone. For instance, Kataoka et al. (ICRAW 2018) created the NIDB near-miss dataset (6200 dashcam videos) and showed that a two-stream CNN could effectively distinguish varying danger levels [7]. These models “capture the temporal feature of an image sequence” by fusing spatial and temporal streams to enhance motion representation. Another example of this came from Shi et al. (TRC 2024), where they combine CNN-based optical-flow inputs with a vision transformer to achieve high accuracy on large crash datasets [11].

More recent work replaces RNNs with temporal Transformers or attention layers to capture long-range dependencies, merging CNN backbones with self-attention. For example, AccidentBlip used a custom transformer architecture to look at how frames change over time. It achieved top performance on the DeepAccident dataset, showing it can accurately predict and detect accidents in real-world driving without extra sensors. [10].

Recent advances incorporated attention mechanisms for improved temporal modeling, methods for modeling inter-object relationships, such as Relation Networks [6], and Reinforcement Learning for interpretable decisions. The DRIVE model [3] demonstrated deep reinforcement learning potential, achieving state-of-the-art performance while providing visual explanations. Other contributions include uncertainty quantification [2], multi-modal fusion strategies, and computationally efficient architectures such as the LATTE model [17].

Despite progress, challenges remain. Achieving reliable anticipation seconds before events while relying on visual data remains an open problem. Modeling temporal dependencies across diverse conditions is a significant challenge.

This project aims to explore novel approaches to video classification applied to the NEXAR Dashcam Collision Prediction Dataset [9].

3. Methods

Our approach to video-based incident detection involves an initial baseline model followed by a series of enhancements to better capture spatial and temporal dynamics, culminating in a multimodal Transformer-based architecture. This section details the methodologies employed, from feature extraction to classification.

3.1. Baseline Model

The baseline model was designed to establish an initial performance benchmark by treating the problem as an image classification task on sampled video frames, followed by feature aggregation. Let an input video clip be denoted by V.

3.1.1 Frame Selection and Feature Extraction

From each input video clip V, a set of $N_{base} = 32$ frames, $F_{base} = \{f_1, f_2, \ldots, f_{N_{base}}\}$, is selected. We utilize a pre-trained InceptionV3 Convolutional Neural Network (IncV3), denoted as Φ_{IncV3}, as a fixed feature extractor. This leverages transfer learning from its training on the ImageNet dataset. For each frame $f_i \in F_{base}$, InceptionV3 processes the frame (resized to 299×299 pixels) to produce a high-dimensional feature vector $v_i \in \mathbb{R}^{D_{IncV3}}$, where $D_{IncV3} = 2048$ is the feature dimension from the InceptionV3's pre-logit layer.

$$v_i = \Phi_{InceptionV3}(f_i)$$

The feature vectors $\{v_1, v_2, \ldots, v_{N_{base}}\}$ extracted from the 32 frames are then aggregated into a single video-level feature representation, V_{feat}, by element-wise averaging. This results in a single tensor $V_{feat} \in \mathbb{R}^{D_{IncV3}}$ representing the entire video clip.

3.1.2 Classification

The aggregated video-level feature tensor V_{feat} is subsequently passed through a sequence of fully connected linear layers. These layers, followed by activation functions, map

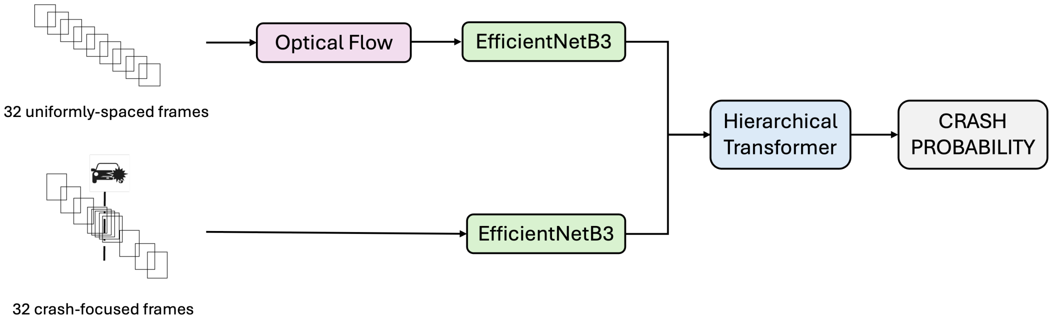

Figure 1. DashGuard multimodal spatio-temporal architecture used to predict the probability of a road accident. 32 uniformly sampled frames are used to create optical flow fields, and 32 non-uniformly-sampled frames (focused around the crash) are independently passed in to the CNN feature extractor. The two streams of output are then concatenated for each frame and then passed into a hierarchical transformer to predict the crash probability.

the video features to a probability score indicating the likelihood of an incident. If we denote the classification head as g_{cls}, the output probability p is:

$$p = \sigma(g_{cls}(V_{feat}))$$

where σ is the sigmoid activation function, ensuring the output is a probability between 0 and 1.

3.2. DashGuard Architecture

To improve upon the baseline, we introduced several modifications aimed at enhancing feature representation, incorporating motion information, and leveraging temporal relationships across frames (see Figure 1).

3.2.1 Enhanced Frame Sampling Strategy

Initial observations indicated that $N_{base} = 16$ frames might be insufficient to capture critical moments in some videos. We increased the number of sampled frames to $N_{prop} = 32$ per video. Furthermore, given that the training data included timestamps for crash events, we implemented a non-uniform "crash-focused" sampling strategy. For a video with a known crash time t_{crash}, a specified ratio of frames are densely sampled within a temporal window centered around t_{crash}. The remaining frames are sampled from the rest of the video to maintain context. This ensures that frames immediately preceding, during, and following the event are more likely to be included, providing the model with more relevant visual information for these critical moments. If a crash time is not available (e.g., for test data or negative samples), uniform sampling is used as a fallback. All frames are resized to 299 $\times$ 299 pixels.

3.2.2 Feature Extraction with EfficientNet

While our baseline model employed InceptionV3 for feature extraction, our proposed architecture adopts EfficientNet-B3 (ENB3) as the backbone CNN for superior performance and efficiency. EfficientNet offers several compelling advantages over InceptionV3 for our collision detection task [14]. First, EfficientNet-B3 achieves 81.1% top-1 accuracy on ImageNet with only 12 million parameters and 1.8 billion FLOPs, compared to InceptionV3's 78.8% accuracy with 24 million parameters and 5.7 billion FLOPs [14, 13]. This represents a significant improvement in computational efficiency—approximately 3× fewer FLOPs for higher accuracy. Second, EfficientNet's compound scaling methodology systematically balances network depth, width, and input resolution using a principled approach, optimizing performance for given computational constraints [14]. This is particularly beneficial for video analysis where processing multiple frames requires efficient feature extraction. Finally, EfficientNet's architecture leverages depthwise separable convolutions and modern normalization techniques, enabling it to capture more complex spatial patterns relevant to collision scenarios while maintaining computational efficiency suitable for real-time applications [14].

3.2.3 Optical Flow for Motion Representation

To explicitly incorporate motion information, we compute dense optical flow between consecutive frames. The intuition is that incidents like accidents or near-misses are often characterized by sudden and atypical motion patterns. For a sequence of N_{prop} RGB frames $\{f_1, f_2, \ldots, f_{N_{prop}}\}$, we calculate $N_{prop} - 1$ optical flow fields $\{o_1, o_2, \ldots, o_{N_{prop}-1}\}$, where o_i represents the flow from f_i to f_{i+1}. We use the Farneback algorithm [5] for this purpose. Each 2D optical flow field $o_i = (dx_i, dy_i)$ is then converted into a 3-channel image representation suitable for input to a CNN. The dx and dy components are normalized to the range [0, 255] and placed into two channels, with the third channel set to zero. This normalized flow image, f_i^{flow}, is then processed by the same pre-trained EfficientNet-B3 model Φ_{ENB3} (acting as a fixed feature extractor) to obtain a flow feature vector $u_i \in \mathbb{R}^{D_{ENB3}}$:

$$u_i = \Phi_{EfficientNet-B3}(f_i^{flow})$$

This results in $N_{prop} - 1$ flow feature vectors.

3.2.4 Multimodal Feature Fusion

Our model leverages both appearance (RGB) and motion (optical flow) information. For each of the N_{prop} time steps, we aim to create a combined feature vector. The RGB features, $v_j = \Phi_{ENB3}(f_j)$ for $j \in \{1, \dots, N_{prop}\}$, are extracted from the temporally sampled RGB frames. The optical flow features $\{u_1, \dots, u_{N_{prop}-1}\}$ are $N_{prop} - 1$ in number. To align these with the N_{prop} RGB features, we pad the sequence of flow features. Specifically, a zero vector of dimension D_{ENB3} is prepended to the flow feature sequence, resulting in $u'_j \in \mathbb{R}^{D_{ENB3}}$ for $j \in \{1, \dots, N_{prop}\}$, where $u'_1 = \mathbf{0}$ and $u'_j = u_{j-1}$ for $j > 1$. The RGB feature v_j and the corresponding padded flow feature u'_j are then concatenated to form a combined multimodal feature vector $x_j \in \mathbb{R}^{2 \cdot D_{ENB3}}$ for each of the N_{prop} timesteps:

$$x_j = [v_j; u'_j]$$

The dimension of this combined feature vector is $D_{combined} = D_{RGB} + D_{Flow} = 2048 + 2048 = 4096$.

3.2.5 Hierarchical Temporal Transformer

To capture multi-scale temporal patterns in collision scenarios, we employ a novel hierarchical transformer that processes the sequence of combined feature vectors $X = \{x_1, x_2, \dots, x_{N_{prop}}\}$ at two complementary temporal resolutions. Unlike standard transformers, our hierarchical approach recognizes that collision detection requires both fine-grained local motion analysis and broader temporal context understanding.

Our architecture consists of two parallel transformer branches: a *local transformer* processing all consecutive frames to capture detailed motion patterns, and a *global transformer* operating on a temporally downsampled sequence to model longer-range dependencies. This dual-scale design suits collision prediction where critical events unfold rapidly (requiring local analysis) while being preceded by gradual contextual changes (requiring global analysis).

The hierarchical processing follows:

$$F_{local} = \mathcal{T}_{local}(X) \tag{1}$$

$$F_{global} = \mathcal{T}_{global}(\text{Downsample}(X)) \tag{2}$$

$$F_{fused} = \text{Fusion}(F_{local}, \text{Align}(F_{global})) \tag{3}$$

where Downsample$(\cdot)$ reduces temporal resolution for global processing, and Align$(\cdot)$ restores the global features to match the original sequence length. The fusion operation combines the multi-scale representations to leverage both local motion details and global temporal context for collision prediction. The final prediction is then computed:

$$p_{prop} = \sigma(g_{cls}(\text{GlobalAvgPool}(F_{fused})))$$

Key parameters: input dimension $D_{combined} = 4096$, model dimension 512, 8 attention heads, 3 layers per branch, feed-forward dimension 1024, dropout 0.3.

3.3. Training Objective

Both the baseline and the proposed model are trained for a binary classification task (incident vs. non-incident). The objective is to minimize the Binary Cross-Entropy (BCE) loss between the predicted probability p and the true label $y \in \{0, 1\}$:

$$\mathcal{L}(y, p) = -[y \log(p) + (1 - y) \log(1 - p)]$$

This loss function is commonly used for binary classification tasks and penalizes confident incorrect predictions more heavily.

4. Dataset and Features

4.1. Dataset

We conduct our experiments using the Nexar Collision Prediction dataset [9], a real-world dashcam video dataset containing 1,500 training videos captured from vehicle-mounted cameras. Each video represents a driving scenario with binary collision labels, where positive samples contain actual collision events and negative samples show normal driving conditions. The videos capture diverse driving environments, weather conditions, and traffic scenarios, making this a challenging real-world computer vision task.

Each video is accompanied by temporal metadata including `time_of_event` and `time_of_alert` timestamps that precisely indicate when collisions occur within the video sequences, enabling temporal-aware modeling approaches.

For our experimental setup, we partition the original Nexar training set using a stratified split to maintain class balance:

- **Training/Validation Set**: 90% of data (1,350 samples) for cross-validation

- **Test Set**: 10% of data (150 samples) for final evaluation

Within the training/validation partition, we employ 5-fold cross-validation, where each fold maintains an 80/20

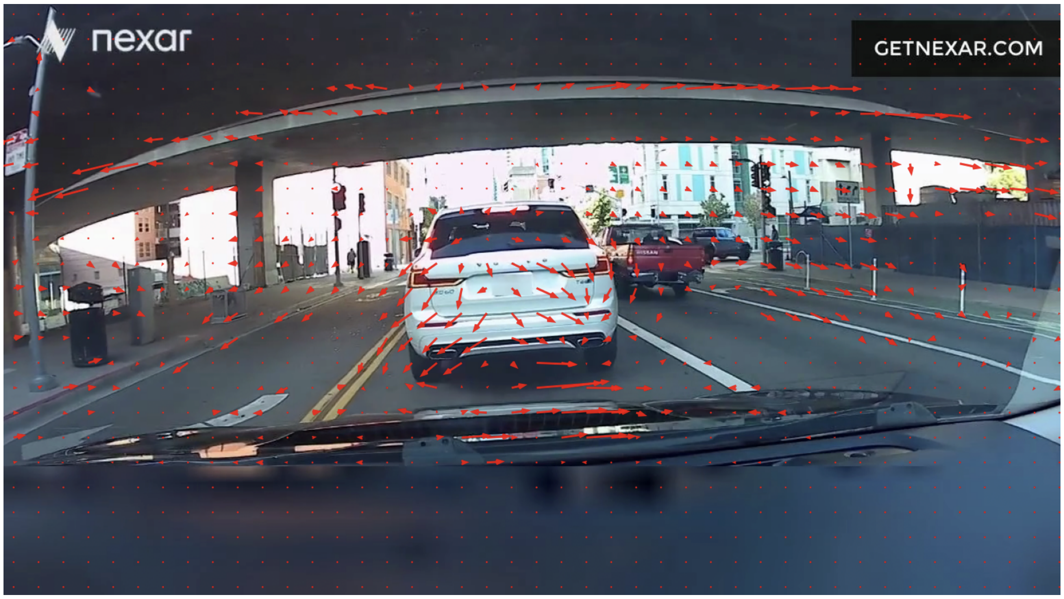

Figure 2. A visualization of the optical flow fields obtained from a pair of frames from a video in the training set. The arrow length is proportional to a pixel's estimated movement between frames.

train/validation split (960 training samples and 240 validation samples per fold). This approach ensures robust model evaluation while preserving sufficient data for final testing.

All video preprocessing extracts 32 frames per sequence at 300×300 resolution. For temporal modeling experiments, we implement crash-focused frame sampling that concentrates 70% of frames around the collision timestamp within a 5-second window, with the remaining 30% providing broader temporal context. Optical flow features are computed between consecutive frames, resulting in 31 flow vectors per video sequence.

The dataset maintains class balance across all splits through stratified sampling, ensuring consistent collision/non-collision ratios during training and evaluation phases.

4.2. Data Preprocessing and Feature Extraction

The primary input to our model consists of the sequential frames from the video clips.

1. **Raw Pixel Data:** The 1280×720 RGB frames serve as the raw input. These frames inherently contain features related to object appearance, environmental context, and road conditions.

2. **Optical Flow:** To capture motion information from the video frames, we computed **optical flow fields**. Using OpenCV, optical flow was calculated between each consecutive pair of sampled frames. This process generates a 2D vector field where each vector represents the apparent motion of image brightness patterns.

   A visualization of what the optical flow looks like for a sample in our training set is shown in Figure 2.

5. Results and Discussion

5.1. Hyperparameters and Training Setup

We employed a consistent set of hyperparameters across all experiments to ensure fair comparison. Our models used a batch size of 16, chosen to balance memory constraints with gradient stability for sequence modeling. We implemented the OneCycleLR scheduler [12] with a maximum learning rate of 8×10^{-7}, starting from an initial rate 25× lower and ending at a rate 100× lower than the peak. This aggressive learning rate schedule was selected through preliminary experiments showing superior convergence compared to standard schedulers for our video understanding task. We used the AdamW optimizer [8] with a weight decay of 1×10^{-3} for regularization.

We tracked the validation loss, validation performance, and learning rate throughout training. Figure 3 plots the validation ROC-AUC and binary cross-entropy loss across epochs, alongside the scheduled learning rate. These curves show that performance stabilizes around epoch 30, aligning with our selected training horizon of 40 epochs. The steep rise and fall of the learning rate promotes rapid convergence early on, followed by fine-tuning in later stages.

For model architecture, we set the transformer dimension to 512 with 8 attention heads across 3 layers, providing sufficient model capacity while maintaining computational efficiency. All models were trained for 40 epochs with gra-

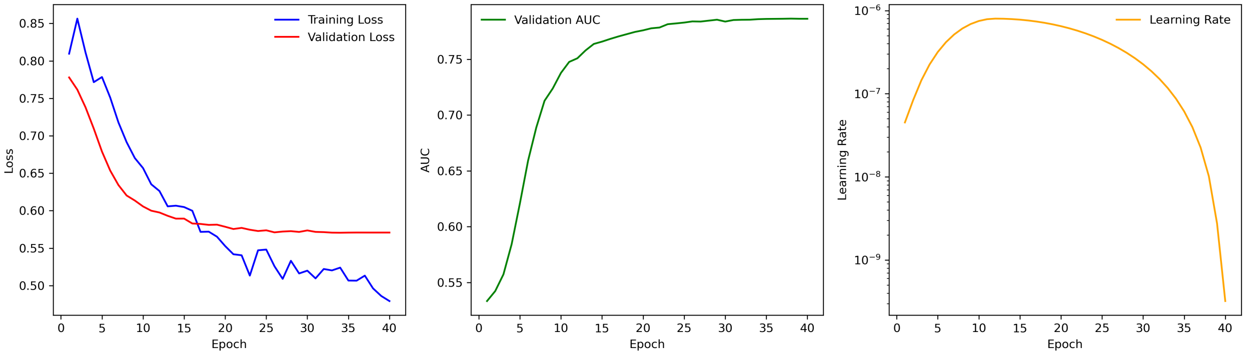

Figure 3. Training and validation loss (*left*), validation AUC (*center*) and LR schedule (*right*) curves for the DashGuard model.

dient clipping at norm 1.0 to prevent instability. We employed 5-fold cross-validation on 90% of the dataset, with each fold maintaining an 80/20 train/validation split while preserving class balance across all splits. The remaining 10% was reserved as a held-out test set for final model evaluation. This cross-validation approach ensures robust performance estimation while preventing data leakage between training and final testing phases. Hyperparameters were initially selected based on common practices for video transformers and refined through pilot experiments on a subset of data.

5.2. Evaluation and Metrics

Our primary evaluation metric is the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), which measures the model's ability to rank positive (accident/near-miss) examples higher than negative (normal driving) ones. We also report overall classification accuracy as a secondary metric.

All models were trained using binary cross-entropy (BCE) loss, which penalizes the divergence between predicted probabilities and ground-truth labels.

5.3. Performance Analysis

Table 1 shows the test set performance for ablation studies of various model configurations using ROC-AUC and accuracy. Our best-performing model combines optical flow features with an EfficientNet backbone and a hierarchical transformer head, achieving a test ROC-AUC of **0.79** and an accuracy of **0.72**. This model outperforms both baselines: one using fully connected (FC) layers instead of a transformer, and another using an InceptionV3 backbone instead of EfficientNet. Interestingly, while the InceptionV3 + OpticalFlow + Transformer variant achieves a slightly higher ROC-AUC of 0.80, its classification accuracy is lower at 0.71, suggesting a slight trade-off between AUC performance and decision accuracy.

Overall, these results demonstrate that modeling motion (via optical flow) and temporal structure (via transformers) contributes meaningfully to performance. The combination of modern CNN backbones and hierarchical temporal reasoning proves most effective for predicting near-accident events from dashcam video.

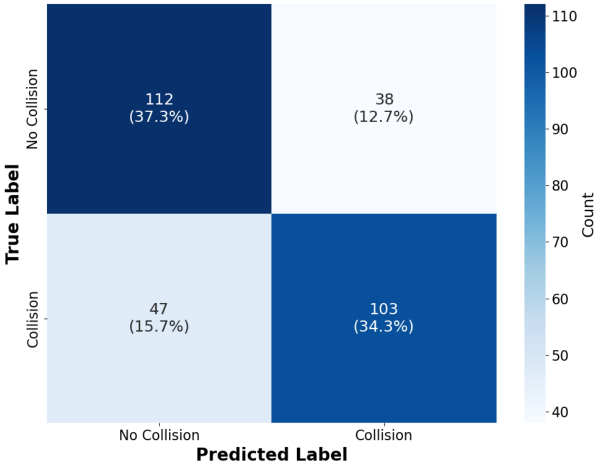

Figure 4. Confusion matrix for EfficientNet + Hierarchical Transformer model on collision prediction test set. The model achieves 71.6% accuracy with 47 false negatives (missed crashes) and 38 false positives (incorrect crash predictions).

In addition to ROC-AUC and accuracy, we include a confusion matrix (Figure 4) to visualize the distribution of true versus predicted labels on the test set. The model correctly identifies 103 of 150 collisions (true positives) and 112 of 150 non-collision cases (true negatives). It also generates 47 false negatives (missed collisions) and 38 false positives (false alarms).

5.4. Failure Mode Investigation

To better understand our model's failure modes, we inspected the false positives and false negatives from the classified examples in our test set.

Model	Test AUC	Test Accuracy
Optical Flow Features + InceptionV3 + Hierarchical Transformer	0.80	0.71
Optical Flow Features + EfficientNetB3 + FC Layers	0.70	0.65
EfficientNetB3 + Hierarchical Transformer	0.76	0.67
DashGuard (Optical Flow Features + EfficientNetB3 + Hierarchical Transformer)	**0.79**	**0.72**

Table 1. Ablation study results on collision prediction for the DashGuard model.

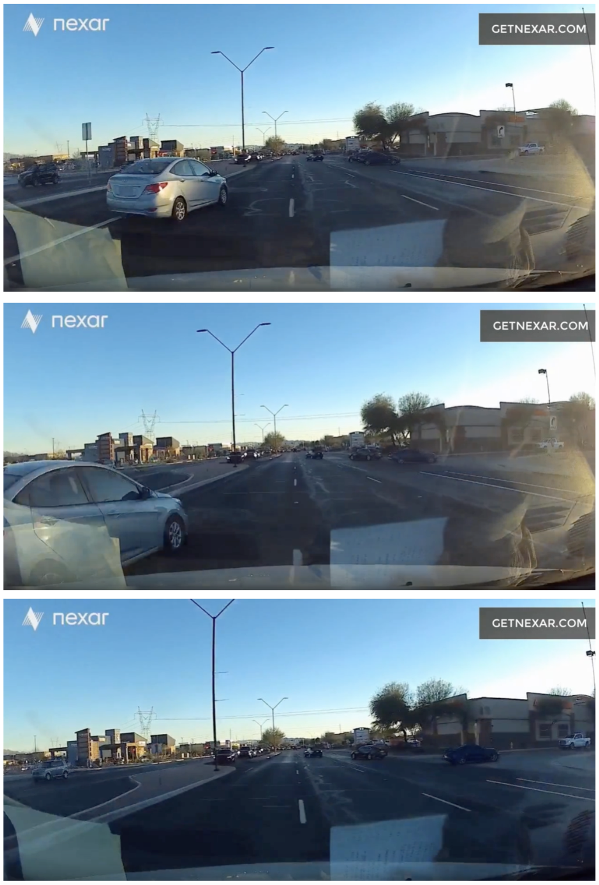

Figure 5. An example of a false negative case. The driver nearly collides with a car in a neighboring lane, but the collision is avoided and the driver continues driving at an uninterrupted speed.

5.4.1 False Negatives

A recurring pattern in false negative cases (i.e., missed collisions or near misses) is the presence of **brief but subtle events**, such as a vehicle swerving abruptly into the lane and correcting itself without any impact. These moments often last only for a few moments, making them difficult for the model to distinguish from ordinary driving. Our process of sampling only only 32 frames per video also makes it more difficult to capture brief interactions that occur between the sampled frames.

Figure 5 shows a representative example, where a vehicle in the adjacent lane briefly crosses into the driver's lane, but no collision occurs and the driver proceeds uninterrupted. Since the driver's speed is constant, the motion data provides no distinguishable features from regular driving, and the understanding of the situation being a ”near-miss” relies heavily on an understanding of where the lanes on the road are, which our model is not explicitly trained to identify.

One contributing factor to this confusion may be the labeling scheme in our dataset, both accidents (with physical collisions) and near misses are grouped under a single “positive” label. This creates a wide and heterogeneous class of positive samples ranging from high-impact collisions to minor evasive maneuvers. This diversity likely introduces noise into the training signal, making it harder for the model to learn a consistent definition of what constitutes a “positive” event.

If this dataset was labeled with a multi-class labeling scheme – separating “accidents”, “near misses”, and “normal driving” as distinct classes - this could help models learn more distinctive features.

5.4.2 False Positives

Although the number of false positives was relatively low in our results, manual inspection revealed various patterns in the false positive failure cases.

Many of these scenarios generally involved nearby vehicles entering the dashcam's field of view at high speeds, or at a close proximity. While these events can appear abruptly, they are not necessarily considered near misses if the vehicle's path is not headed towards the vehicle. These motion cues may resemble pre-collision trajectories, causing the model to incorrectly classify the event as a near miss.

5.5. Limitations and Discussion

We observed signs of slight overfitting during training. Specifically, Figure 3 shows the training loss continued to decline slightly while the validation loss plateaued after a certain point. This divergence, though not severe, suggests that the model may be overfitting to subtle patterns in the training data. To mitigate this, we incorporated several strategies, including weight decay via the AdamW optimizer [8], a OneCycle learning rate schedule [12], and dropout layers within the attention mechanism. We also tuned our hyperparameters to minimize the gap between training loss and validation loss.

One notable limitation of our approach was the reliance on optical flow for motion representation. While optical flow captured short-range motion effectively, we found that it struggled in extremely high-speed scenarios when objects moved large distances between frames. This limitation may impact the model's ability to detect motion cues preceding accidents, especially when those cues occur rapidly.

Additionally, as discussed in our qualitative analysis, a more nuanced multi-class labeling scheme could enable finer-grained modeling and improve prediction performance across the spectrum of risky driving events.

Upon inspection of the misclassified examples, particularly false positives, some appear to involve close calls that resemble near misses, but are not labeled as such. However, due to the subjective nature of interpreting motion and risk from video, it's difficult to definitively label such borderline cases. Identifying and removing outliers in video data is challenging, as context unfolds over time and depends on subtle temporal and spatial cues that are not easily separable with simple heuristics.

In general, our model demonstrates good qualitative behavior on clear-cut accident cases with distinctive visual cues such as sudden stops, collisions, or vehicles crossing boundaries aggressively. However, it struggles with ambiguous cases, especially subtle near misses. These findings suggest that while the model is sensitive to prominent visual anomalies, it may benefit from further refinement in capturing context. Future work could improve robustness by incorporating additional modalities (such as audio or speed) or through more granular labels and human-in-the-loop feedback mechanisms.

6. Conclusion

In this project, we tackled the task of predicting traffic accidents and near misses from dashcam video footage using spatio-temporal deep learning models. We explored a variety of architectures combining CNN-based visual encoders with hierarchical transformers. Among these, our highest-performing model was a hierarchical transformer operating on features from both EfficientNet RGB frames and optical flow. This achieved the best test performance, with an ROC-AUC of 0.79 and accuracy of 0.72.

We found that the addition of optical flow features generally helped all models achieve better performance across the board. Transformer-based architectures outperformed simpler baselines by better capturing long-range temporal dependencies and contextual interactions in driving scenes. However, limitations in optical flow quality and label granularity (e.g., grouping accidents and near misses together) posed challenges for fine-grained discrimination. False positives were often caused by sudden but non-threatening visual motion, and false negatives by subtle near misses that may lack obvious visual precursors.

Future work could explore additional data modalities, including audio and speed information from dashcam footage, to further enrich contextual understanding. Due to time and memory constraints, we did not extensively pursue data augmentation for video; however, with more time and compute resources, augmentation techniques could boost model performance. Finally, while our models focused on extracting features from individual video frames, leveraging video-focused architectures such as 3D CNNs or advanced models like Video MAEv2 [15] could yield promising results.

7. Contributions & Acknowledgements

L.M. and K.Y. designed the hierarchical transformer framework and evaluation methodology. L.M. and K.Y. equally contributed to model implementation, including the attention mechanisms and feature extraction components. L.M. and K.Y. jointly processed the Nexar collision prediction dataset, implemented data augmentation strategies, and conducted all training and evaluation experiments. All computational work was performed locally on an M4 MAX MacBook Pro without external cloud computing resources. L.M. and K.Y. collaboratively wrote the manuscript. The authors would like to thank Ryan Li for his mentorship and valuable feedback throughout the project development.

References

[1] Google self-driving car project monthly report. Technical report, Google Inc., May 2015. Link.

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

[3] W. Bao, Q. Yu, and Y. Kong. Drive: Deep reinforced accident anticipation with visual explanation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7599–7608, 2021.

[4] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun. Anticipating accidents in dashcam videos. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 136–153. Springer, 2017.

[5] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In J. Bigun and T. Gustavsson, editors, *Image Analysis*, pages 363–370, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[6] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3588–3597, 2018.

[7] H. Kataoka, T. Suzuki, S. Oikawa, Y. Matsui, and Y. Satoh. Drive video analysis for the detection of traffic near-miss incidents. In *Proceedings of the ICRA Workshop on Towards Human-Centered Robotics: User-Study Design & Experimental Validation (WHCERO'18)*, Brisbane, Australia, May 2018. Available as arXiv:1804.02555 [cs.CV].

[8] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[9] D. C. Moura, S. Zhu, and O. Zvitia. Nexar dashcam collision prediction dataset and challenge, 2025. arXiv preprint arXiv:2503.03848.

[10] Y. Shao, Y. Xu, X. Long, S. Chen, Z. Yan, Y. Yang, H. Liu, Y. Wang, H. Tang, and Z. Lei. Accidentblip: Agent of accident warning based on ma-former, 2025.

[11] L. Shi and F. Guo. Two-stream video-based deep learning model for crashes and near-crashes. *Transportation Research Part C: Emerging Technologies*, 166:104794, 2024.

[12] L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[14] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[15] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao. Videomae v2: Scaling video masked autoencoders with dual masking, 2023.

[16] K.-H. Zeng, S.-H. Chou, F.-H. Chan, J. C. Niebles, and M. Sun. Agent-centric risk assessment: Accident anticipation and risky region localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1000–1008. IEEE, 2017. Workshop on The Bright and Dark Sides of Computer Vision in the Real World (BDCVRW).

[17] J. Zhang, Y. Guan, C. Wang, H. Liao, G. Zhang, and Z. Li. Latte: A real-time lightweight attention-based traffic accident anticipation engine. *Information Fusion*, 122:103173, 2025.