Predicting Car Collisions from Dashcam Footage

Riya Raj
Columbia University
rr3630@columbia.edu

Niranjan Sundararajan
Columbia University
ns3888@columbia.edu

Kaushal Damania
Columbia University
kd2990@columbia.edu

Thilina Balasooriya
Columbia University
tnb2119@columbia.edu

Abstract

We address the task of early vehicle collision and near-miss prediction from dashcam videos using the Nexar dataset, aiming to enhance advanced driver assistance systems (ADAS). Our approach integrates hybrid architectures (ResNet-LSTM, ViT-LSTM, ViT-Transformer) with fine-tuned vision transformers (XCLIP, Timesformer) to model spatiotemporal dynamics. To improve early detection, we introduce a soft Gaussian labeling scheme with asymmetric temporal weighting and an alpha-balanced loss. The training pipeline employs data augmentation, gradient clipping, and AdamW regularization for robust generalization. Our models achieve strong average precision and high recall, demonstrating effective early accident anticipation under real-world conditions and contributing to safer ADAS and autonomous driving systems. We have open sourced the code at: GitHub

1 Introduction

Early prediction of vehicle collisions and near-miss events from dashcam footage is vital for enhancing road safety in Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. This task requires modeling complex spatiotemporal dynamics and delivering timely risk assessments for preventive interventions.

The Nexar Dashcam Crash Prediction Challenge (Kaggle) offers a large-scale annotated dataset with 1,500 training and 1,344 test videos across three categories: collision, near-miss, and normal driving. The goal is to assign a binary risk score to each video, emphasizing early and accurate warnings to maximize practical utility.

In this project, we analyze and extend existing methods to develop an effective solution for this challenge. We propose a multi-model framework combining custom architectures (ResNet-LSTM, ViT-LSTM, ViT-Transformer) with fine-tuned vision transformers (Microsoft XCLIP, Facebook Timesformer). To prioritize early detection, we introduce a soft Gaussian labeling scheme with asymmetric temporal weighting and use an alpha-balanced loss to address class imbalance and emphasize recall. Our training pipeline incorporates exploratory data analysis, augmentation, gradient clipping, and AdamW optimization.

This work presents a hybrid, transfer learning-based approach to vision-based collision prediction, offering robust early risk anticipation in complex, real-world driving scenarios.

2 Related Work

Vision-based collision prediction has advanced with dashcam datasets like DAD and CCD, focusing on accident anticipation. Early methods, such as Chan et al. [2], used CNN-LSTM models to capture

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

spatial and temporal features, achieving moderate crash detection success. Recent approaches employ Transformer architectures for video understanding, with Timesformer [1] introducing space-time attention to outperform LSTMs in action recognition, and XCLIP [7] leveraging vision-language pre-training for video classification. In crash prediction, two-stream models combining RGB and optical flow, such as Simonyan and Zisserman [8], show promise but struggle with early prediction due to binary labeling limitations.

Our work extends these methods by integrating custom and fine-tuned Transformer models for dashcam-based crash prediction. We address binary labeling constraints with a soft Gaussian labeling approach and asymmetric temporal weighting, enhancing early detection. Our EDA-driven preprocessing and multi-model strategy, combining CNN-LSTM, ViT-LSTM, and Transformer architectures, provide a tailored solution for the Nexar dataset's challenges.

3 Technical Prerequisites

This section outlines the technical foundations of the components used in our approach.

3.1 Soft Gaussian Labeling

Soft Gaussian labeling assigns continuous-valued labels to video frames based on a Gaussian function centered at a key timestamp, typically the `alert_time`—the earliest moment a collision or near-miss can be predicted. Unlike binary labeling schemes, which assign 1 to event frames and 0 otherwise, soft labeling produces a temporally smooth signal. The label at frame t is computed as:

$$\text{Label}(t) = \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right)$$

where μ is the center (`alert_time`) and σ controls the spread. To bias the model toward early detection, we use an *asymmetric* temporal weighting: $\sigma_{\text{before}} = 2.0$ for frames preceding `alert_time`, and $\sigma_{\text{after}} = 0.5$ for subsequent frames. This configuration peaks at 1.0 at `alert_time` and decays smoothly over time, assigning non-zero values (e.g., 0.607 at $t = \mu - 2$) to earlier frames. This strategy emphasizes the detection of precursors to risky events, addressing the tight average 1.6s window between `alert_time` and `event_time` in the dataset.

3.2 Alpha-Balanced Loss Function

We employ a composite alpha-balanced loss to jointly optimize two objectives: per-frame likelihood regression and video-level binary classification. Both tasks utilize binary cross-entropy (BCE) loss:

$$\text{BCE} = -\sum [y \log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

The total loss is defined as:

$$L = \alpha L_{\text{frame}} + (1-\alpha) L_{\text{binary}}$$

where L_{frame} compares predicted per-frame scores against soft Gaussian labels, and L_{binary} compares the max-aggregated video score to the ground truth class (positive if max frame `label > 0.5`, else negative). We use $\alpha = 0.5$ to ensure a balanced trade-off between early frame-level signal learning and accurate sequence-level risk assessment. This formulation is applied in ResNet-LSTM and ViT-LSTM models and is well-suited for the class-balanced dataset.

3.3 CNN-LSTM for Spatio-Temporal Modeling

CNN-LSTM models combine Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks [5] to capture spatio-temporal dynamics in video data. CNNs like ResNet-18 [4] extract spatial features from frames $I \in \mathbb{R}^{H \times W \times C}$, producing a feature map:

$$F[i,j] = \sum_{m,n} W[m,n] \cdot I[i+m, j+n] + b$$

where W is a filter (e.g., 3×3) and b is a bias. ResNet-18 outputs 512-dimensional features f_t, forming a sequence $\{f_t\}_{t=1}^{T}$ (e.g., $T = 20$), which a bidirectional LSTM processes to model temporal dynamics. In ResNetLSTM, this captures static (e.g., road layout) and dynamic (e.g., vehicle motion) patterns, enabling early crash prediction within the Nexar dataset's 1.6-second `alert_time` to `event_time` window.

3.4 Vision Transformers and Timesformer for Spatio-Temporal Modeling

Vision Transformers (ViTs) and Timesformer model spatio-temporal dynamics in dashcam videos by processing spatial and temporal dimensions complementarily. ViTs [3] divide frames into patches (e.g., 16×16), projecting them into tokens and applying self-attention to capture spatial dependencies, computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V$$

where $Q, K, V \in \mathbb{R}^{N \times D}$ are query, key, and value matrices for N patches and embedding dimension D. In models like ViT + LSTM, these spatial features are sequenced for temporal modeling, while Timesformer [1] uses divided space-time attention, applying spatial then temporal attention across frames, aggregating features via a `[CLS]` token. Together, ViTs excel at local spatial relationships, while Timesformer captures both short- and long-range temporal dependencies. We use ViTs and Timesformer to model inter-object relationships and temporal evolution in the Nexar dataset, enabling early crash prediction within the 1.6-second window between `alert_time` and `event_time`.

3.5 Regularization Techniques

Regularization mitigates overfitting by imposing constraints during training, balancing model complexity and generalization. Dropout randomly deactivates neurons with a probability p (e.g., 0.3) at each step, modifying their output as:

$$h_i' = \begin{cases} 0 & \text{with probability } p, \\ \frac{h_i}{1-p} & \text{otherwise,} \end{cases}$$

forcing the network to learn more robust features by preventing over-reliance on specific neurons. Weight decay adds an L2 penalty to the loss, $L_{\text{total}} = L_{\text{data}} + \lambda \sum_i w_i^2$, with $\lambda = 1 \times 10^{-4}$, discouraging large weights to improve generalization. AdamW [6] enhances optimization by decoupling weight decay from the learning rate, ensuring better convergence across diverse training scenarios. We apply dropout in ResNetLSTM and ViT + LSTM, and AdamW in Transformer models, addressing the Nexar dataset's variability (e.g., weather, occlusions) to ensure reliable generalization for ADAS applications.

4 Dataset and Exploratory Analysis

4.1 Dataset Description

The Nexar Dashcam Crash Prediction Challenge dataset includes 1500 training and 1344 test videos of real-world driving scenarios with varying conditions like weather and occlusions. The training set is balanced, with 750 normal (negative, target=0) and 750 collision/near-miss (positive, target=1) videos, each typically 720×1280 pixels at 30 FPS and 40 seconds long. Positive training videos are annotated with `event_time` and `alert_time`, while negative ones lack these. Test videos, around 10 seconds long, end at time-to-accident intervals (500 ms, 1000 ms, 1500 ms) without annotations, evaluated against private leaderboard values.

4.2 Exploratory Data Analysis

To guide the design of our modeling pipeline, we began with a thorough exploratory analysis of the metadata. A key observation was the dataset's perfectly balanced class distribution, consisting of 750 positive and 750 negative samples. This balance eliminates the risk of class bias during training and provides a stable foundation for applying loss functions that treat both classes symmetrically. Temporal patterns within the positive samples revealed a notably tight predictive window: on average, the `alert_time` occurs at 17.50 seconds, preceding the `event_time` at 19.10 seconds by approximately 1.6 seconds. Moreover, the majority of events cluster near the end of each video (median: 19.80 seconds), highlighting the importance of temporally sensitive labeling to account for uncertainty around the moment of impact. We also noted

considerable variability in video durations, ranging from as short as 3.03 seconds to as long as 56.8 seconds. This heterogeneity necessitated temporal normalization techniques, including uniform sampling and sequence padding, to ensure consistent input lengths across the dataset. In contrast, spatial resolution remained constant, with all videos rendered at 1280×720, enabling the use of standardized resizing and cropping procedures without risk of distortion. Collectively, these insights shaped critical methodological choices across data preprocessing, temporal alignment, and model training.

5 Methods

Our approach for the Nexar Dashcam Crash Prediction Challenge uses a streamlined pipeline of data preprocessing, model architecture design, and training procedures to model spatio-temporal dynamics for early collision prediction, ensuring timely and accurate risk assessment in real-world driving scenarios while addressing the dataset's unique challenges.

5.1 Data Preprocessing

We extract 10-second segments of 20 or 30 frames at 2 or 3 FPS to balance temporal resolution and computational efficiency, capturing critical dynamics within the prediction window. Negative samples use the initial 10 seconds, reflecting typical safe driving periods, while positive samples are drawn around `alert_time` from $\max(0, \text{alert\_time} + \text{tta} - 10)$ to $\min(\text{alert\_time} + \text{tta}, \text{duration})$, with $\text{tta} \in [0.5, 1.5]$ seconds introducing variation to improve model robustness across diverse scenarios. Missing frames are zero-filled for consistency, with labels set to 0.0 for negatives and soft Gaussian values in $[0.0, 1.0]$ for positives to emphasize early detection. Feature-based models resize frames to 224×224, normalize to stabilize training, and use ResNet-18 (512D) or ViT-B/16 (768D) encoders for spatial feature extraction. Transformer models train end-to-end with augmentations (random cropping, flipping, color jittering) to handle diverse driving conditions like varying lighting and occlusions, and validation uses resized, center-cropped inputs to ensure consistent evaluation across models.

5.2 Model Architectures

We evaluated seven model configurations to capture spatial and temporal patterns in dashcam videos, iteratively refining our approach based on Mean Average Precision (mAP) across time-to-accident intervals (500 ms, 1000 ms, 1500 ms) to prioritize early prediction accuracy. ResNetLSTM combines a pre-trained ResNet-18 (512D features) with a bidirectional LSTM to model spatio-temporal dynamics, achieving an mAP of 0.728 by effectively capturing local spatial and temporal patterns, establishing a strong baseline. LSTM + ViT replaces ResNet-18 with ViT-B/16 (768D features) paired with an LSTM (hidden dimension 256, dropout 0.3), scoring 0.694, impacted by noise in cluttered frames due to ViT's global attention mechanism. ViT + Transformer (non-regularized) processes ViT-B/16 features through a Transformer encoder (2 layers, 768 hidden dimensions, 12 attention heads), aggregating via a `[CLS]` token, but reached only 0.676, limited by overfitting to complex patterns; regularization improved this to 0.731, enhancing generalization. Microsoft XCLIP, a vision-language Transformer, underperformed at 0.659, likely due to its vision-language pretraining mismatch with dashcam data requirements. Timesformer (non-regularized), a space-time Transformer, achieved an mAP of 0.720, benefiting from its space-time attention mechanism to model temporal evolution and spatial interactions. Finally, Timesformer (regularized) reached the highest mAP of 0.741, leveraging robust spatio-temporal attention and improved generalization to excel in early crash prediction across diverse scenarios.

5.3 Training Procedures

ResNetLSTM and ViT + LSTM train on pre-extracted features with a 75/25 train-validation split, using alpha-balanced BCE loss to balance frame- and sequence-level learning, optimized with the Adam optimizer (learning rate 10^{-3}, dropout 0.3) to prevent overfitting, and early stopping based on validation metrics to avoid overtraining and ensure optimal performance. Transformer models train end-to-end on image sequences with BCE loss on `[CLS]` token outputs, optimized via AdamW [6] (weight decay 1×10^{-4}) to improve convergence stability across the dataset's variability. We incorporate gradient clipping to manage exploding gradients, learning rate warmup to ensure smooth initial training, and cosine annealing to adaptively adjust the learning rate, collectively enabling stable convergence, robust performance, and effective handling of diverse driving conditions in the Nexar dataset.

5.4 Algorithm

Algorithm 1 Collision Prediction Pipeline

Input: Video V, `alert_time`, `target`, encoder (ResNet-18/ViT-B/16), temporal model (LSTM/Transformer)
Output: Collision probability p_{seq}
// Frame Sampling
if `target` = 0 **then**
    Sample initial 10s at 2/3 FPS, label $y_t = 0.0$
else
    Sample 10s around `alert_time` ▷ With tta $\in [0.5, 1.5]$
    Label $y_t = \exp\left(-\frac{(t-\text{alert\_time})^2}{2\sigma^2}\right)$ ▷ Asymmetric σ
end if
Form sequence $\{I_t\}_{t=1}^{T}$ ($T = 20/30$), zero-fill missing frames
// Feature Extraction
Extract features $\{f_t\}_{t=1}^{T}$ from $\{I_t\}_{t=1}^{T}$ using encoder
// Temporal Modeling
if model is LSTM-based **then**
    Process $\{f_t\}$ with LSTM, predict p_{seq} via heads
else
    Process $\{f_t\}$ with Transformer, predict p_{seq} via `[CLS]`
end if
// Optimization
Optimize with alpha-balanced loss $L = \alpha L_{\text{frame}} + (1-\alpha) L_{\text{binary}}$
Return: p_{seq}

6 Experiments

6.1 Evaluation Metrics

Submissions are evaluated using Mean Average Precision (mAP) across multiple time-to-accident intervals, defined by the gap between the `event_time` and `alert_time`. Specifically, three Precision-Recall (PR) curves are computed, corresponding to time-to-accident thresholds of 500 ms, 1000 ms, and 1500 ms. The Average Precision (AP) is calculated for each PR curve, and the final evaluation score is the mean of these three AP values. This metric captures the model's ability to balance precision and recall while anticipating incidents within varying lead times.

Each submission consists of one line per test video, containing the video ID and a predicted probability score in the range $[0, 1]$, where higher scores indicate a greater likelihood of a collision or near-miss event.

6.2 Results

The performance of our models on the Nexar test set was evaluated using Mean Average Precision (mAP) across time-to-accident intervals of 500 ms, 1000 ms, and 1500 ms. Table 1 and Figure 1 summarize and visualize the results, respectively.

Model	mAP
Timesformer (reg)	0.741
ViT + Trans. (reg)	0.731
ResNetLSTM	0.728
Timesformer (non-reg)	0.720
LSTM + ViT	0.694
ViT + Trans. (non-reg)	0.676
XCLIP	0.659

Table 1: mAP scores of models on the Nexar test set.

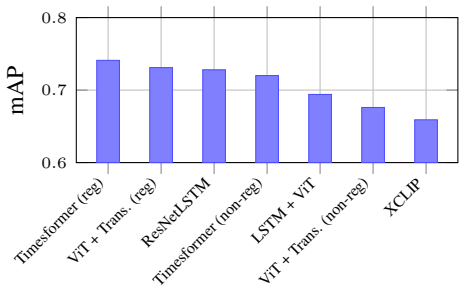

Figure 1: mAP comparison across models.

7 Discussion

7.1 Novelty

Our work introduces several innovative elements to advance vision-based crash prediction on the Nexar dataset. We propose a novel soft Gaussian labeling scheme with asymmetric temporal weighting to prioritize early prediction within the tight 1.6-second window between `alert_time` and `event_time`, encouraging models to detect subtle precursors of collisions. Unlike standard binary labeling, which assigns discrete labels (1 for event frames, 0 otherwise) and often overlooks temporal uncertainty, our approach generates a smooth probability distribution that emphasizes early frames, enabling the model to learn nuanced patterns preceding an event. This is complemented by an alpha-balanced loss function that ensures unbiased learning across both frame-level regression and sequence-level classification on a balanced dataset, enhancing early detection and overall risk assessment. In contrast to focal loss, commonly used in imbalanced datasets to focus on hard examples, our alpha-balanced loss explicitly balances frame-level and sequence-level objectives with equal weighting ($\alpha = 0.5$), better suited for the balanced Nexar dataset and the dual-task nature of early prediction and classification. Additionally, we pioneer the application of advanced regularization techniques, including dropout and AdamW weight decay, to improve generalization across variable driving conditions, such as occlusions and adverse weather. Our multi-model strategy integrates custom architectures—ResNetLSTM, ViT-LSTM, and ViT-Transformer—with fine-tuned pre-trained Transformers such as XCLIP and Timesformer. This ensemble approach leverages the complementary strengths of various architectures to address the complex spatio-temporal dynamics present in dashcam video data. Finally, we incorporate advanced training techniques—including data augmentation, gradient clipping, and learning rate scheduling—to stabilize and optimize Transformer performance. These enhancements ensure robust and efficient learning, supporting practical deployment in real-world ADAS applications.

7.2 Contribution

All team members contributed equally throughout the project. We collaboratively designed the multi-model architecture with custom and fine-tuned Transformers, and shared tasks such as dataset preprocessing, frame extraction, and implementing soft Gaussian labeling. Model development, hyperparameter tuning, and evaluation were evenly divided, with key decisions—like training strategies and regularization—made jointly. This report was co-authored to reflect our collective effort in advancing vision-based crash prediction for the Nexar Challenge.

8 Conclusion

We proposed a multi-model framework for early collision and near-miss prediction using dashcam videos, developed in the context of the Nexar Dashcam Crash Prediction Challenge. Our approach integrates custom-designed architectures (ResNetLSTM, ViT + LSTM, ViT + Transformer) with fine-tuned pre-trained Transformers (XCLIP, Timesformer), coupled with a novel soft Gaussian labeling scheme and an alpha-balanced loss function to enhance temporal sensitivity and classification robustness.

Exploratory data analysis informed our preprocessing and training pipeline, aligning model design with the dataset's characteristics, including class balance, variable video durations, and narrow time-to-accident windows. Through this work, we demonstrate the potential of hybrid spatial-temporal modeling and transfer learning to advance vision-based accident prediction for real-world ADAS applications.

Quantitative results from the Kaggle leaderboard will serve as a benchmark for performance evaluation and future extensions toward real-time deployment scenarios.

References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning (ICML)*, 2021.

[2] Fu-Hsiang Chan, Yu-Ting Chen, Yu Xiang, and Min Sun. Anticipating accidents in dashcam videos. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 136–153, 2016.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Matthias Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit,

and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

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

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[7] Xin Ma, Li Wang, Hongsheng Fan, Yu Li, and Yang Wang. Xclip: A large-scale video-text model. *arXiv preprint arXiv:2206.11040*, 2022.

[8] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.