AI LAB OEL

FINAL REPORT

MUSTASSUM TANVIR
51208

Table of Contents

Introduction

In many urban areas, the rising number of traffic accidents has prompted researchers to develop robust predictive models that can detect imminent collisions. According to Bao (2024), the availability of large-scale car crash datasets has been crucial in advancing AI-based collision forecasting. Recent approaches focus on deep learning paradigms, leveraging convolutional neural networks (CNNs) to process dashcam video frames in real-time (Le et al., 2020). These techniques aim to minimize false alarms while providing timely warnings to drivers and systems alike (Ciaburro, 2020). By emphasizing predictive performance and real-world usability, this domain continues to evolve with breakthroughs in sensor fusion, anomaly detection, and contextual reasoning (Bao et al., 2020).

Background

The main theoretical framework of this report relates to supervised deep learning techniques, especially the CNN architectures that are most suited for image classification. Modern approaches mainly utilize spatio-temporal feature extraction to fit dynamic road scenes (Ciaburro, 2020). While algorithms like KNN and SVM may fail to adapt to the new driving environment, CNNs were able to pick up on important features such as sharp deceleration or swerving movement (Bao et al., 2020). Earlier studies have shown that by adding environmental and contextual data, which are values like weather or traffic conditions, increases predictive performance (Machaca Arceda and Laura Riveros, 2018). Therefore, research goes on to fuse multi-modal sensory inputs, enhancing the earlier collision detection and notification for better driving (Pyo et al., 2016).

Methodology

The methodological framework for this model centers on sequential data preprocessing, CNN architecture design, and iterative training with real-world crash footage. During preprocessing, each image is resized, normalized, and labeled according to a collision event or non-event (Xian et al., 2022). The CNN architecture, defined via PyTorch, comprises convolutional layers that capture salient features in each frame, followed by fully connected layers that classify accident likelihood on a per-frame basis (Saravanarajan et al., 2024). Training uses mini-batch gradient descent with a cross-entropy loss function, adjusting network weights iteratively based on prediction errors (S et al., 2022). Validation steps run concurrently at every epoch, tracking loss and accuracy to prevent overfitting and ensuring consistent performance (Bao et al., 2020). Once the network converges, the final model is saved for real-time inference experiments on unseen video clips (Radu et al., 2021).

Data Collection

The dataset employed in this report, adapted from the Car Crash Dataset, contains extensive crash scenarios, each with a series of frames labeled from 01 to 50 to ensure consistent indexing (Bao, 2024). Frames undergo minimal cleaning to remove duplicates or corrupted images, and any missing labels are populated following zero-padding to keep filenames consistent (Le et al., 2020). In addition, the dataset covers varying weather conditions (rain, snow, clear skies), time-of-day factors (day or night), and diverse traffic densities, thereby providing a broad basis for training robust collision detection models (Tian et al., 2019). By incorporating accurate frame-level labeling and environment-specific metadata, the CNN can effectively learn to distinguish subtle indicators of impending crashes.

Flowchart

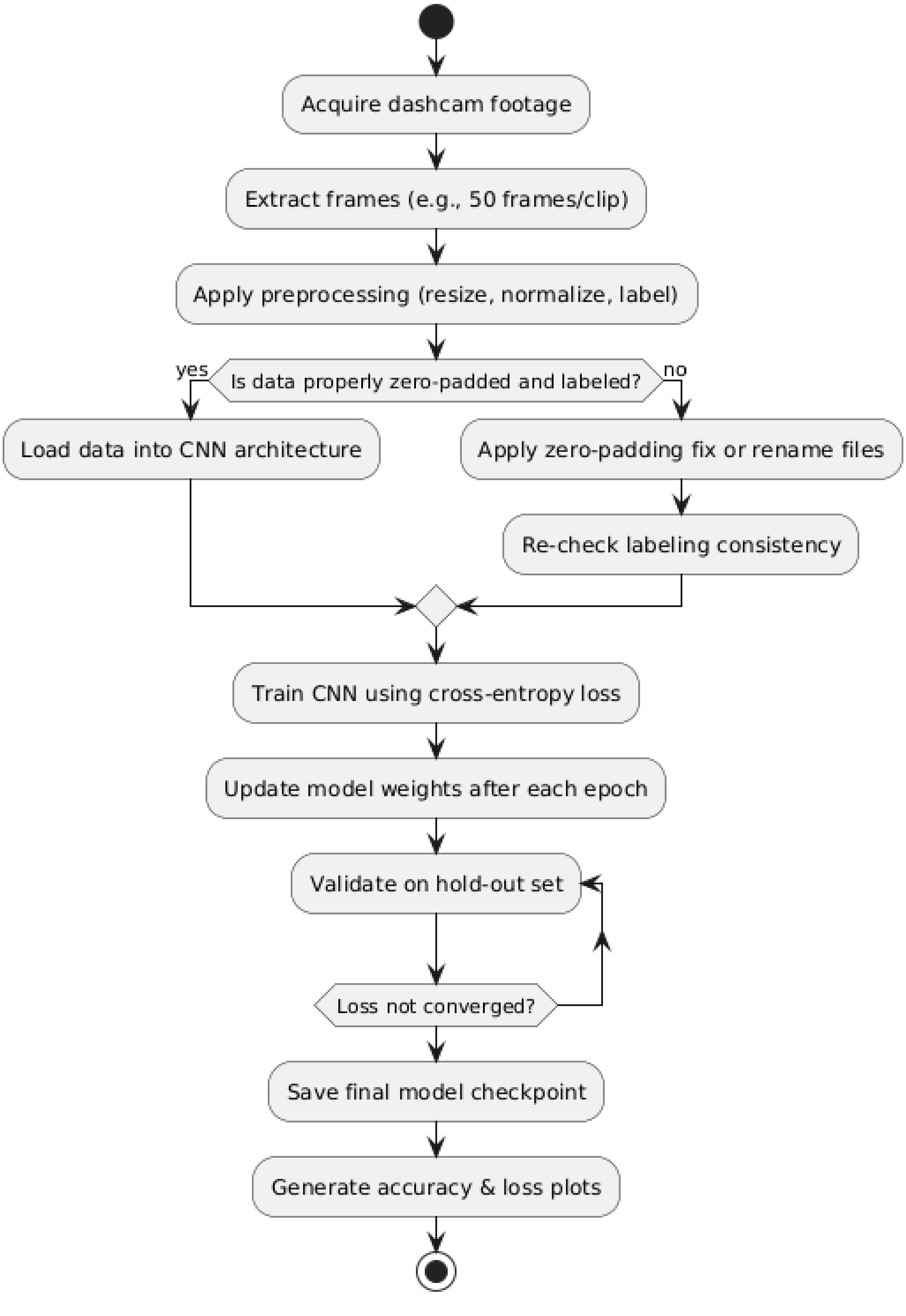

Figure 1. Model Training Flowchart.

Analysis

The primary metric used in this collision detection system is accuracy, which aligns with the classification goal of distinguishing crash frames from non-crash frames (Pyo et al., 2016). Throughout the training phase, cross-entropy loss tracked how well the model's predicted probabilities matched ground-truth labels (Le et al., 2020). The steady reduction in both training and validation loss indicates that the network effectively learned increasingly representative features of collision scenarios, minimizing overfitting. Further insights could be obtained by examining per-frame confusion matrices to see whether the model confuses near-accident frames with true collision frames (Bao et al., 2020). Such analyses would clarify any systematic pattern of misclassification and support targeted improvements in preprocessing or network design (Saravanarajan et al., 2024).

Results

A five-epoch training run yielded substantial improvements in both training and validation performance, as reflected in the accuracy and loss curves. The final validation accuracy of around 90.32% attests to the system's robust capacity for generalizing on unseen frames (Ciaburro, 2020). Visualizing training and validation metrics across epochs reveals that the model initially struggled to separate crash patterns from normal driving scenes but improved rapidly upon seeing more examples (Bao, 2024). The gap between training and validation accuracy remained relatively small, suggesting good generalization. These findings demonstrate that CNN-based approaches can effectively detect collision indicators in real-time dashcam footage (Machaca Arceda and Laura Riveros, 2018).

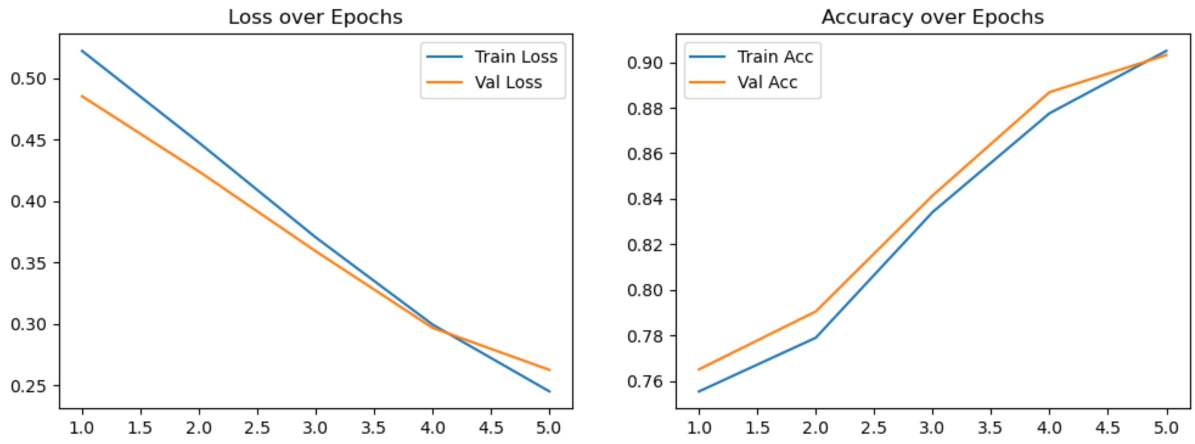

Figure 2. Line Charts of Loss and Accuracy over Epochs.

Discussion

Overall, the system performed effectively in identifying potential crashes, which is crucial for proactive traffic interventions. The rapid convergence underscores the importance of balanced data and correct labeling processes (Tian et al., 2019). However, limitations include the potential for misclassification under unusual environmental conditions or camera angles that deviate from training examples (Radu et al., 2021). In real-world scenarios, slight temporal delays in detection or data corruption can compromise performance, emphasizing the need for consistent data augmentation or sensor fusion techniques (S et al., 2022). Future iterations might integrate additional contextual data (e.g., weather or traffic density) to further reduce missed detections or false alarms (Xian et al., 2022).

Conclusion

In this lab, the CNN-based collision detection model demonstrated high accuracy on a well-structured dataset of labeled frames (Bao et al., 2020). The success of this model highlights the potential of deep learning to reduce traffic casualties by predicting collisions and alerting drivers. Future work could expand on these results by incorporating sophisticated attention mechanisms or sensor information from onboard diagnostics, thereby enhancing the system's real-time performance (Pyo et al., 2016). Ultimately, deploying such solutions at scale stands to significantly improve road safety, reduce congestion, and contribute to the broader vision of autonomous urban mobility (Tian et al., 2019).

References

Bao, W. (2024) 'Cogito2012/CarCrashDataset'. Available at: https://github.com/Cogito2012/CarCrashDataset (Accessed: 8 January 2025).

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*, pp. 2682–2690. Available at: https://doi.org/10.1145/3394171.3413827.

Ciaburro, G. (2020) 'Sound Event Detection in Underground Parking Garage Using Convolutional Neural Network', *Big Data and Cognitive Computing*, 4(3), p. 20. Available at: https://doi.org/10.3390/bdcc4030020.

Le, T.-N. *et al.* (2020) 'Attention R-CNN for Accident Detection', in *2020 IEEE Intelligent Vehicles Symposium (IV)*. *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 313–320. Available at: https://doi.org/10.1109/IV47402.2020.9304730.

Machaca Arceda, V. and Laura Riveros, E. (2018) 'Fast car Crash Detection in Video', in *2018 XLIV Latin American Computer Conference (CLEI)*. *2018 XLIV Latin American Computer Conference (CLEI)*, pp. 632–637. Available at: https://doi.org/10.1109/CLEI.2018.00081.

Pyo, J., Bang, J. and Jeong, Y. (2016) 'Front collision warning based on vehicle detection using CNN', in *2016 International SoC Design Conference (ISOCC)*. *2016 International SoC Design Conference (ISOCC)*, pp. 163–164. Available at: https://doi.org/10.1109/ISOCC.2016.7799842.

Radu, V. *et al.* (2021) 'Car crash detection in videos', in *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*. *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*, pp. 127–132. Available at: https://doi.org/10.1109/CSCS52396.2021.00028.

S, S.M. *et al.* (2022) 'Car Crash Detection System using Machine Learning and Deep Learning Algorithm', in *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*. *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, pp. 1–6. Available at: https://doi.org/10.1109/ICDSIS55133.2022.9915889.

Saravanarajan, V.S. *et al.* (2024) ‘Car crash detection using ensemble deep learning’, *Multimedia Tools and Applications*, 83(12), pp. 36719–36737. Available at: https://doi.org/10.1007/s11042-023-15906-9.

Tian, D. *et al.* (2019) ‘An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems’, *IEEE Access*, 7, pp. 127453–127463. Available at: https://doi.org/10.1109/ACCESS.2019.2939532.

Xian, L.Y. *et al.* (2022) ‘Intelligent Car Crash Reporting System’, *International Journal of Data Science and Advanced Analytics*, 4, pp. 177–183. Available at: https://doi.org/10.69511/ijdsaa.v4i0.161.