A Hybrid Deep Learning Model for Predicting Delta-V from Real World Crash Images

Vikas Hasija, Erik G. Takhounts, Matthew J. Craig

Abstract Delta-V, which is the change in velocity of a vehicle during a collision, is an indicator used for quantifying crash severity. The National Highway Traffic Safety Administration (NHTSA) utilises WinSMASH software to estimate delta-V based on detailed information from the crash scene; however, a prior study found that it underestimates delta-V values by 13.2% on average compared to Event Data Recorder (EDR) delta-V. This study proposes a deep learning (DL) approach to improve the efficiency and accuracy of delta-V estimation. The developed hybrid DL model provides a marginal improvement over WinSMASH in predicting longitudinal delta-V for vehicle-to-vehicle frontal impacts directly from real world crash images and minimal vehicle data, reducing dependence on extensive crash scene information. The dataset was compiled from multiple crash databases and split into training, validation and test subsets. A hybrid DL model, integrating an EfficientNet-based convolutional neural network with vehicle-specific data was developed using training and validation sets. Evaluation on the unseen test dataset demonstrated that the proposed hybrid DL model underestimated delta-V by 12% on average with a Root Mean Square Error (RMSE) of 7.5 kph (4.6 mph). In comparison, WinSMASH underestimated delta-V by 14% on average with an RMSE of 8.3 kph (5.2 mph).

Keywords Deep learning***,*** machine learning, delta-V, real-world crash images.

I. INTRODUCTION

Delta-V, which represents the change in velocity of a vehicle during a collision, is a widely used indicator for assessing crash severity and predicting occupant injury risk [1]. Currently, tools like the National Highway Traffic Safety Administration's (NHTSA) WinSMASH software [2] are used to estimate delta-V of the vehicle involved in crashes. WinSMASH offers two algorithms for computing delta-V: trajectory analysis and damage analysis algorithms. Both algorithms require detailed measurements from the crash scene, vehicle damage and vehicle stiffness characteristics to estimate the delta-V. Reference [3] compared longitudinal delta-V from WinSMASH 2008 with that from Event Data Recorder (EDR). Their study found that WinSMASH 2008 reconstructions underestimated delta-V by 13.2% on average and exhibited a root mean square error (RMSE) of 9.40 kph (5.84 mph).

Computer vision, a subfield of artificial intelligence, has emerged as a transformative tool in analysing and interpreting visual data. Applications of computer vision span numerous domains, including medical diagnostics, autonomous driving, surveillance, disaster assessment, etc. The ability to extract detailed and nuanced features from images and videos makes it a potentially powerful tool for analysing vehicular collisions. Specifically, it offers an alternative for estimating important indicators like delta-V by leveraging real-world post collision crash images. This approach has the potential to deliver accurate and efficient assessments while reducing dependence on exhaustive crash scene data.

Silver et al. [4] proposed a computer vision and deep learning approach to predict crash characteristics such as delta-V and location of collision (LOC) using post-collision vehicle images, reducing the need for detailed forensic crash reconstructions. Their focus was on front-end and rear-end collisions for mid-size passenger vehicles with delta-V values ranging from 10 to 40 kph. For the front-end model, their study utilised a small dataset of 155 crash images from the National Automotive Sampling System - Crashworthiness Data System (NASS-CDS) database to demonstrate the feasibility of image-based delta-V estimation. However, their study lacked performance metrics comparing their proposed model to WinSMASH on an independent test dataset.

V. Hasija (e-mail: vikas.hasija@dot.gov) is a Mechanical Engineer at National Highway Traffic Safety Administration (NHTSA). E.G. Takhounts is a Mechanical Engineer at NHTSA. M.J. Craig is the Chief of Human Injury Research Division at NHTSA

Baek et al. [5] developed a machine learning model to predict delta-V for rear-end collisions, aiming to estimate neck injury severity in traffic accidents without relying on additional analytical tools and thus offering a streamlined approach to injury prediction. Their study utilised data from 236 crash tests conducted by the Working Group on Accident Mechanics (AGU Zurich) [6]. They did not use any post-collision crash images but instead used crash data such as speed, weight, angle and offset of the vehicles to estimate delta-V, demonstrating the potential of machine learning in predicting delta-V.

In this study, we propose a novel hybrid deep learning framework that integrates both post-collision images and minimal vehicle data to more accurately and efficiently predict longitudinal delta-V for vehicle-to-vehicle frontal collisions.

II. METHODS

This study was conducted in two phases to develop a deep learning model that more accurately predicts longitudinal delta-V for vehicle-to-vehicle frontal impacts. In the first phase, a machine learning (ML) model was developed to predict EDR-equivalent longitudinal delta-V from WinSMASH longitudinal delta-V estimates. The second phase involved developing a hybrid deep learning (DL) model, trained with EDR/EDR-equivalent longitudinal delta-V as targets, to predict EDR-equivalent longitudinal delta-V directly from post-collision crash images and minimal vehicle-related data. A flowchart outlining this process is provided in Appendix A, Fig. A1. Unless otherwise specified, delta-V in this paper refers to longitudinal delta-V.

First Phase: ML Model to Predict EDR Equivalent delta-V

Due to inaccuracies in the delta-V estimates computed by WinSMASH, the final predictive model was not trained directly on WinSMASH estimates. Instead, in phase one, a ML model was first developed to correct these delta-V estimates. The ML model predicts corrected delta-V values (EDR equivalent delta-V) based on WinSMASH estimate, and vehicle and crash-related information. The corrected delta-V estimates were subsequently used for training the final predictive model in phase two.

For phase one, crash data used to develop the ML model was sourced from Crash Investigation Sampling System (CISS), National Automotive Sampling System-Crashworthiness Data System (NASS-CDS) and Crash Injury Research Engineering Network (CIREN). The dataset was limited to vehicle-to-vehicle, non-rollover frontal impact cases (Deformation Location = Front, Principal Direction of Force (PDOF) = 10 o'clock – 2 o'clock) with non-overlapping frontal damage, where both WinSMASH and EDR longitudinal delta-V were available. No filter was used for case years. Cases with multiple EDR events associated with the same Collision Deformation Classification (CDC) code were excluded, resulting in a final dataset of 4,390 cases. The model years of the subject vehicles ranged from 1994 - 2016 for NASS-CDS, 1996 - 2023 for CISS and 1996 – 2022 for CIREN. Crash and vehicle-related information, as shown in Table I, were collected for these cases. Stiffness values were obtained from the WinSMASH categorical stiffness database [2] based on vehicle body type and impact location (front, side, or rear). All variables served as input features except EDR longitudinal delta-V, which was the target variable.

TABLE I
DATA COLLECTED FOR ML MODEL

Vehicle related	Crash related
Subject Vehicle Weight	WinSMASH longitudinal delta-V
Subject Vehicle Body Type	Max Crush
Subject Vehicle Stiffness	CDC code
Partner Vehicle Weight	EDR longitudinal delta-V
Partner Vehicle Body Type	
Partner Vehicle Struck Side	
Partner Vehicle Stiffness	

The distribution of EDR longitudinal delta-V for the 4,390 cases, categorised by data source, is presented in Fig 1. CIREN intentionally selects cases involving moderate to severe injuries, unlike CISS and NASS-CDS, which include a broader range of injury severities, resulting in different delta-V distributions between these datasets.

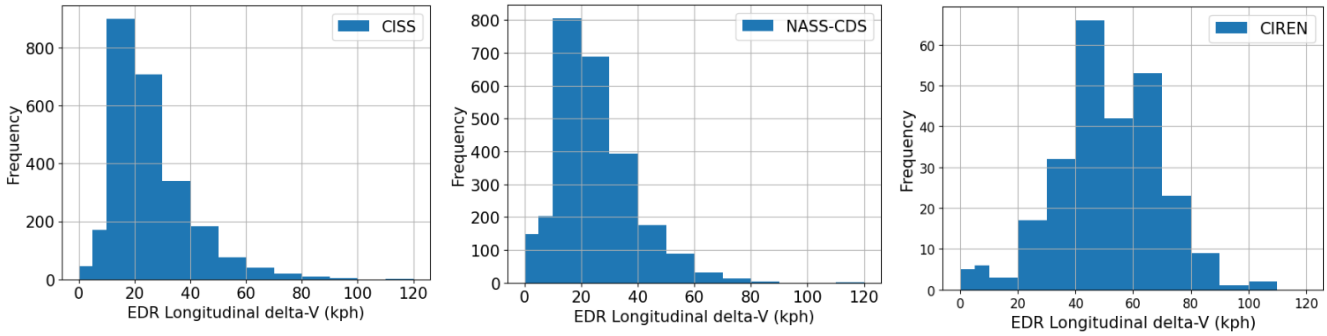


Fig. 1. EDR longitudinal delta-V distribution for CISS, NASS-CDS, and CIREN.

This dataset was split into training (73%, N=3283), validation (15%, N=580) and test sets (12%, N=527) for ML model development and evaluation. The splits were chosen to ensure there were enough samples (> 500) in both the validation and test sets. Due to limited data available for delta-V exceeding 50 kph, an additional 500 synthetic data points were generated for delta-V > 50 kph and were incorporated into the training set. These synthetic datapoints were generated using the Synthetic Data Vault (SDV) Python library [7]. The Gaussian Copula Synthesizer available in SDV was used for generating the synthetic data. Both diagnostics and quality evaluations were performed on the synthetic data using the Synthetic Data Metrics (SDMetrics) Python library [8].

Diagnostics assessed data validity and structure. Data validity ensured boundary adherence i.e., the continuous values in each column adhered to the same min/max range as the real data, and the categorical values in each column adhered to the same categories as the real data. Structure ensured that data types (e.g., integer, string) for each column were consistent between real and synthetic data. The diagnostic score was computed by combining the data validity and structure scores across all columns.

Quality evaluation focused on comparing the statistical similarity. Marginal distribution of each column and bivariate distribution for a pair of columns were compared between real and synthetic data. Marginal distributions were quantified using the Kolmogorov-Smirnov statistic (KSComplement [8]) for continuous columns and Total Variation Distance (TVComplement [8]) for categorical columns. Bivariate distribution between a pair of continuous columns was quantified by computing correlation coefficient on real and synthetic data and using them to derive similarity score (CorrelationSimilarity [8]). For all other column pairings (e.g., continuous-categorical, categorical-categorical), bivariate distribution was assessed using Contingency Tables and Total Variation Distance (ContingencySimilarity [8]). The marginal and bivariate distribution scores were combined to obtain the quality score.

On the synthetic data, diagnostic score of 100% and quality score of 87% was achieved. A comparison of real and synthetic data for WinSMASH longitudinal delta-V and EDR longitudinal delta-V is shown in Appendix B, Fig. B1.

Before model training, the input features were transformed. Numerical features, namely WinSMASH longitudinal delta-V, max crush, vehicle weights, and vehicle stiffnesses were normalised. Categorical features, namely vehicle body types and struck side were label-encoded. The CDC code was broken down into its six components [9], with each component label-encoded.

Both bagging-based (Random Forest [10]) and boosting-based (LightGBM [11]) ensemble machine learning algorithms were explored. Hyperparameter tuning was conducted using 5-fold cross-validation and bayesian optimisation [12] to find the best performing model. The best ML model was then evaluated on the unseen test dataset using the following metrics: 1) Root Mean Square Error (RMSE) between the true (EDR) and predicted

(ML) delta-V values, and 2) a regression analysis comparing the true (EDR) and the predicted (ML) delta-V values to assess the model's average underestimation/overestimation error and the coefficient of determination (R^2).

Additionally, these metrics were computed between EDR versus WinSMASH delta-V on the same test dataset, allowing for a direct comparison between EDR-WinSMASH metrics and EDR-ML metrics. Consistent with prior studies comparing WinSMASH to EDR data [2,3], this paper employed Regression Through Origin (RTO) [13] for regression analysis to maintain methodological consistency and enable meaningful comparisons.

Second Phase: Hybrid DL Model to Predict delta-V from Real World Crash Images

For this phase, the crash data and post collision images for developing the hybrid DL model were sourced from CISS, NASS-CDS and CIREN. Similar to the previous phase, only vehicle-to-vehicle, no-rollover frontal impact cases (Principal Direction of Force (PDOF) = 10 o'clock – 2 o'clock) with non-overlapping frontal damage were included. However, this phase did not require the availability of both WinSMASH and EDR delta-V. Cases with either or both values were used (Appendix A, Fig. A1). Again, no filter was used for case years.

For all selected cases, vehicle and crash related data (Table I) were collected alongside frontal crash images for the subject vehicle (Fig. 2). Cases were excluded if the images showed snow or plastic covering on the vehicle (Fig. 2d) or if the hood was opened by the crash investigator or taped shut (Fig. 2e, 2f). In addition to image-based filtering, cases with EDR delta-V that were missing any vehicle related data were dropped. For cases with only WinSMASH delta-V, those missing any vehicle related data, CDC or max crush were excluded. After applying these filtering criteria, the final dataset comprised of 9,281 cases.

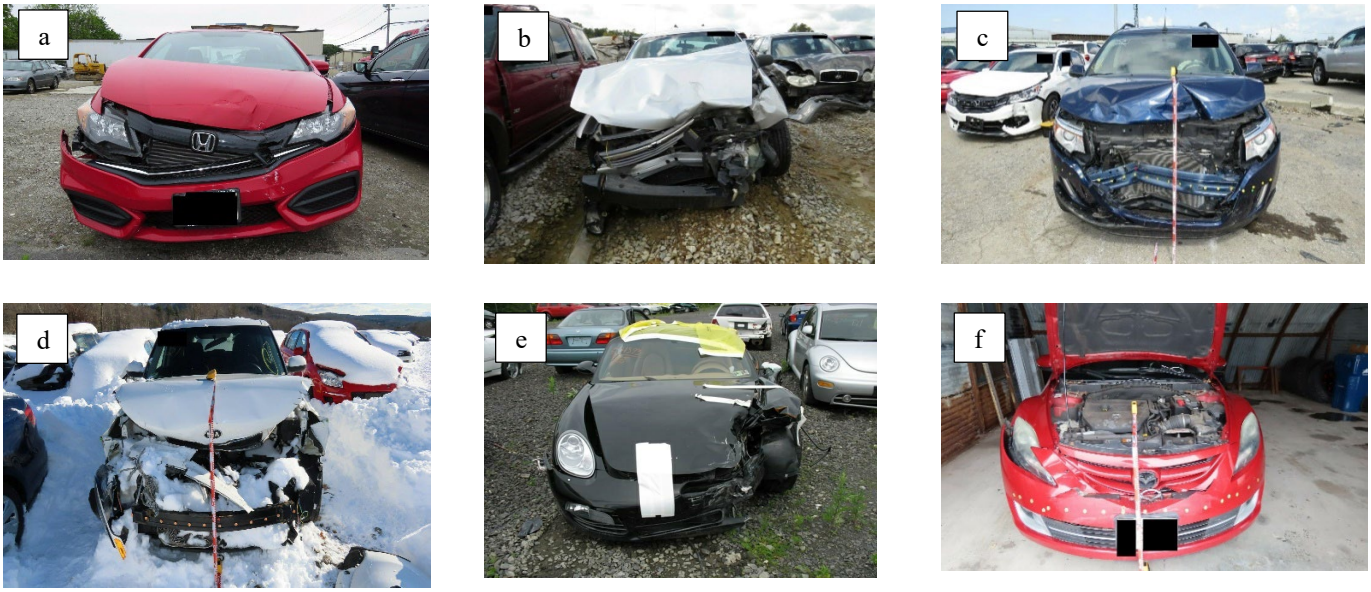


Fig. 2. Examples of frontal crash Images sourced from the crash databases.

As shown in Fig. 2, the crash images display varying backgrounds, with some showing other vehicles in the background. To minimise the impact of background noise on prediction accuracy, all images were segmented (Fig. 3). Image segmentation was performed using the pretrained U^2-Net model [14] available through the Rembg [15] Python library. Given that the crash images dataset significantly differs from the data used to train the U^2-Net model, some images were not segmented correctly. For these cases, manual segmentation was performed using the Computer Vision Annotation Tool (CVAT) [16].

Fig. 3. Examples of original and segmented Images.

The hybrid DL model was trained using both vehicle-related data and crash images as inputs (Table II). The vehicle-related data included readily available attributes namely weights, body types, and struck side, reducing the need for detailed information. Stiffness values were sourced from the WinSMASH categorical stiffness database based on vehicle body type and impact location, avoiding the need for any additional measurements. The output/target variable for training the hybrid DL model was the EDR/EDR equivalent delta-V. For cases without EDR delta-V, the ML model developed in the first phase was utilised to predict EDR equivalent delta-V using WinSMASH delta-V and other vehicle information as inputs. For cases where EDR delta-V was available, it was directly utilised. Out of the 9,281 cases in the dataset, 2,211 cases had EDR delta-V, while for the remaining cases, EDR equivalent delta-V was computed using the phase one ML model.

TABLE II
INPUTS FOR THE HYBRID DL MODEL

Subject Vehicle Weight	Partner Vehicle Weight
Subject Vehicle Body Type	Partner Vehicle Body Type
Subject Vehicle Stiffness	Partner Vehicle Struck Side
Subject Vehicle Frontal Image	Partner Vehicle Stiffness

The dataset was divided into training (79%, N=7,321), validation (16%, N=1,464), and test (5%, N=496) subsets using stratified sampling based on delta-V values to ensure equal representation across all delta-V ranges in each subset. The training and validation datasets, comprising 95% of the total data, were utilised for model development. Specifically, the training dataset was used to fit the model, while the validation dataset supported hyperparameter tuning. The test dataset, kept separate from model development, served as an unseen dataset for evaluating the final predictive model's performance.

The hybrid DL model architecture is illustrated in Fig. 4. The crash images were processed through a 2D convolutional neural network (CNN-2D) to generate the feature maps. These feature maps were then passed through a pooling layer to generate the image feature vector. The vehicle related numerical and categorical data were normalised and encoded, respectively before being concatenated into a single vector. This concatenated vector was passed through a 1D convolutional layer, followed by batch normalisation [17] and activation function [18] to generate the info feature vector. The image and info feature vector were then concatenated and passed to the regression head. The regression head consisted of two fully connected layers, each followed by a dropout layer [19]. The final output layer, with one hidden unit, produced the longitudinal delta-V value. The model was implemented in Tensorflow 2.17 [20].

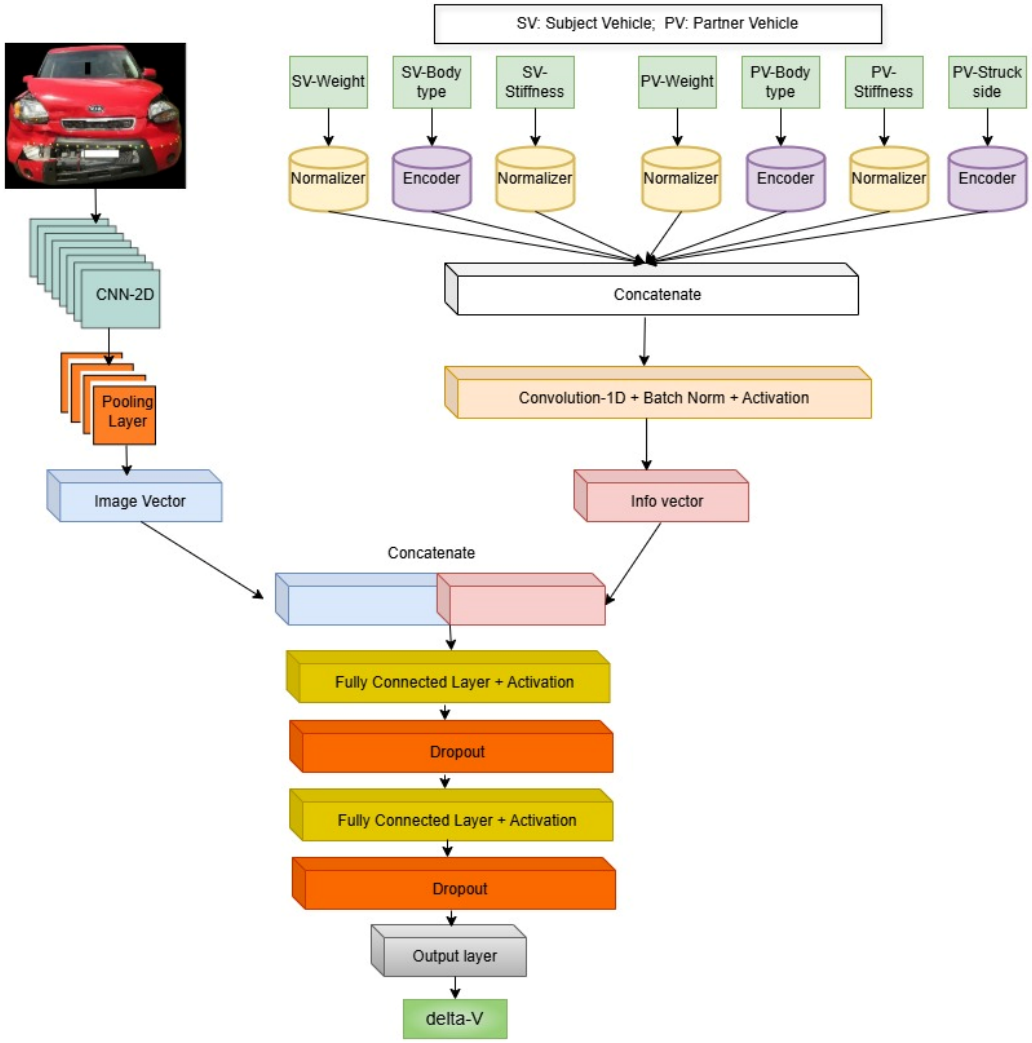


Fig. 4. Hybrid DL model architecture.

DL models involve a lot of hyperparameters and since the best hyperparameters were unknown at the start of model development, hyperparameter tuning was conducted to find the best configuration for the hybrid DL model. A total of 18 hyperparameters, listed in Appendix C, Table C1, were tuned.

Since a diverse range of convolutional neural network (CNN) architectures is available [21], seven different CNN-2D architectures were explored as part of hyperparameter tuning. Each of these CNN models were initialised with ImageNet [22] weights before training. For the remaining layers of the hybrid DL model, the initialisers listed in Appendix C, Table C1 were used. The crash images were resized to 300 x 300 using distortion free resizing, augmented, and then passed as RGB images with pixel values in the range from 0 – 255. These images were fed into the preprocessing layer of each CNN-2D before being processed by the respective CNN-2D model.

Data augmentation was used during training to help prevent overfitting and to enhance model's generalisation and robustness. The image augmentations used are shown in Appendix C, Table C2. These seven augmentations were selected based on a review of the crash images. The augmentation pipeline was configured using the Random Augmentation pipeline in Keras CV [23]. The random augmentation pipeline was set up to randomly apply 3 augmentation per image out of the seven selected augmentations. While most are self-explanatory, specific augmentations like grid mask, gaussian blur and color jitter warrant further explanation.

- **Grid Mask**: Many crash images include black boxes occluding sensitive information (e.g., Fig. 3). The grid mask augmentation randomly adds black areas to images during training. This teaches the model to treat occluded areas as "noise," preventing it from overfitting to patterns associated with occlusions.

- **Random Gaussian Blur**: The crash images vary in quality, with some being high-resolution and others low-resolution. Applying Gaussian blur ensures the model is exposed to a spectrum of image qualities during training, enhancing its robustness to such variations. Without this augmentation, the model might overly depend on fine details in high-quality images. Gaussian blur encourages the model to rely on broader, less detailed patterns.
- **Random Color Jitter**: Crash images differ in color and brightness due to variations in lighting, time of day, and camera settings. Color jitter augmentation introduces such variations during training, helping the model avoid overfitting to specific lighting or color conditions.

Model training was conducted with a batch size of 8 and a maximum of 100 epochs. Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates. For this purpose, ReduceLROnPlateau [20] callback was utilised with the patience parameter set to 5 epochs. Additionally, EarlyStopping callback [20] with a patience of 15 epochs was used. These callbacks monitor the objective and if no improvement is observed within the specified epochs, the learning rate is reduced, or training is stopped altogether. The objective for model training was to maximise R^2 score on the validation dataset.

Hyperparameter tuning was conducted using Ray Tune [24] leveraging the Asynchronous Hyperband scheduler [25] and the Tree-structured Parzen Estimator (TPE) optimisation algorithm provided through the HyperOpt [26] integration. The hyperparameter tuning process was executed on two Nvidia RTX-A6000 GPUs.

Final model evaluation

The final predictive model from phase two was evaluated on the unseen test dataset (N=496) that was not used during model development. The evaluation involved the following metrics:

1. Root Mean Square Error (RMSE)
2. RTO analysis to assess the model's average underestimation/overestimation error and the coefficient of determination (R^2).

Among the 496 test cases, 126 cases had EDR delta-V. For these cases, EDR delta-V was compared against both the WinSMASH delta-V and the DL model predicted delta-V, and the evaluation metrics were computed. Additionally, the prediction errors between EDR-WinSMASH and EDR-DL were analysed with respect to both delta-V and PDOF. The runtime performance of the hybrid DL model was also measured.

Model Interpretation

Occlusion sensitivity [27] was used for model interpretation to understand the regions of the crash image that contribute most to the model's predictions. Occlusion sensitivity doesn't rely on class-specific outputs and can be applied to regression tasks. Although computationally expensive, it is simple and model-agnostic. It works by systematically occluding (masking) parts of the crash image and analysing how the model's output changes. The results are visualised using heatmaps, highlighting the regions that have the greatest influence on the model's predictions.

III. RESULTS

First Phase

The Random Forest (RF) model achieved the best performance and was therefore selected for use in the second phase. The tuned hyperparameters and their corresponding optimised values for the RF model are shown in Appendix D, Table D1. The model was evaluated on the unseen test dataset, with results presented in Fig. 5. Fig. 5a shows the correlation between EDR delta-V and WinSMASH delta-V, while Fig. 5b shows the correlation between EDR delta-V and RF delta-V. The RF model demonstrated a significantly lower average underestimation error (6.0%) compared to WinSMASH (15.0%). Additionally, the RF delta-V showed a stronger correlation (R^2=0.93) and lower RMSE (8.46 kph) with EDR delta-V compared to WinSMASH delta-V (R^2=0.91, RMSE=9.6 kph).

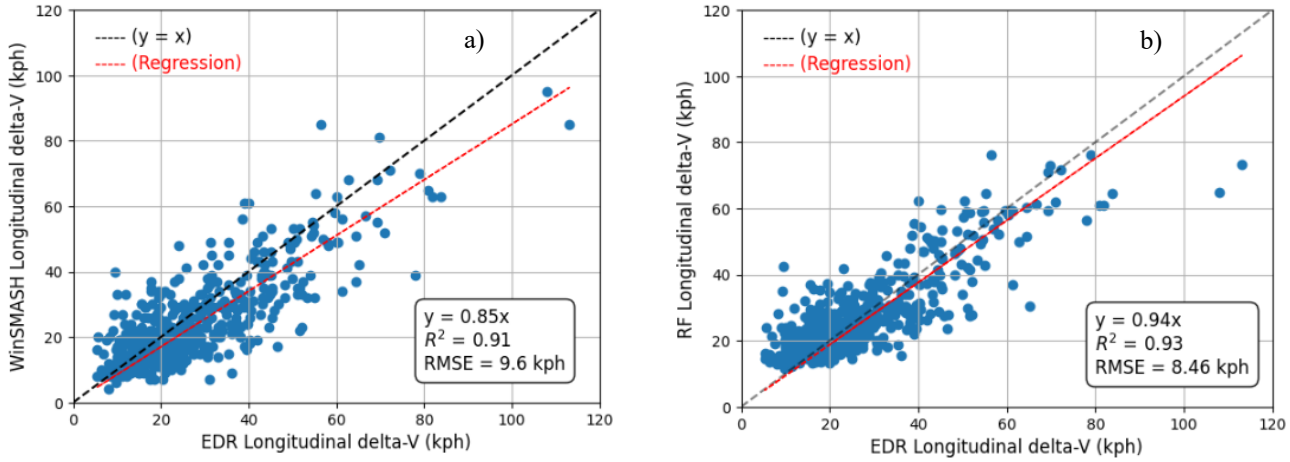


Fig. 5. Comparison between a) EDR delta-V and WinSMASH delta-V, and b) EDR delta-V and RF delta-V for the test dataset.

Prior to using the model for predictions in the second phase, the selected RF model was trained on the test set as well. This ensured that no data was wasted, and that all available knowledge was incorporated into the model.

Second Phase

The hyperparameters that achieved the best performance on the validation dataset are shown in Appendix D, Table D2. The best-performing hybrid DL model utilised the EfficientNetB5 CNN architecture, trained with Nadam optimiser [20] and a learning rate of 1.37e-4. This model had 29,129,275 trainable parameters. The best checkpoint with these hyperparameters was evaluated on the training (non-augmented) and validation datasets (Fig. 6). On the training set, R^2 was 0.99, RMSE was 2.66 kph and underestimation error was 1%, while on the validation set, R^2 was 0.95, RMSE was 6.33 kph and underestimation error was 6%.

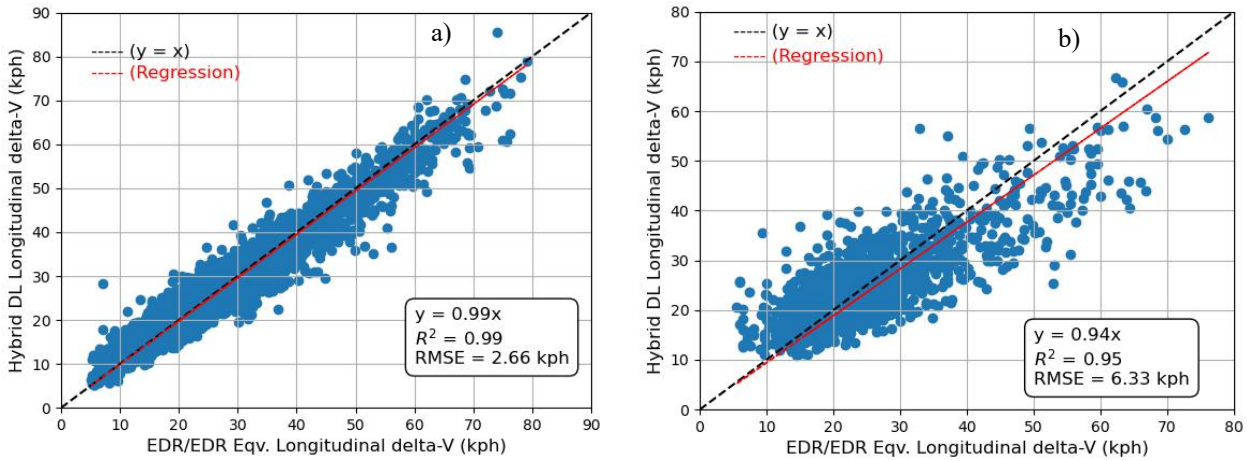


Fig. 6. Hybrid DL model performance on a) training and b) validation datasets.

Final model evaluation

The best-performing hybrid DL model was evaluated on the unseen test dataset (N=496) that was not used during model development. On the test dataset, the RMSE was 6.38 kph (3.96 mph), R^2 was 0.94 and on average the model underestimated delta-V by 7% (Fig. 7).

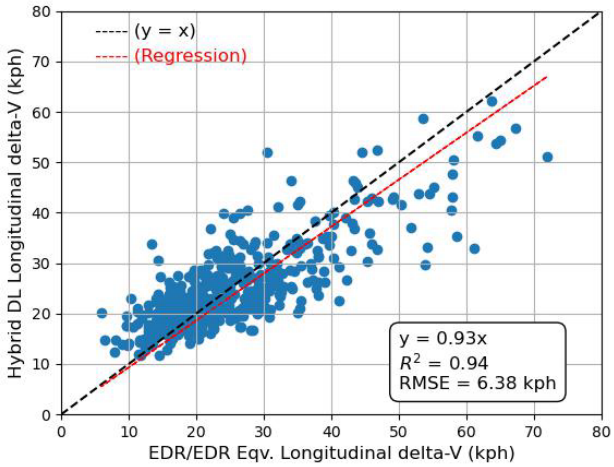


Fig. 7. Hybrid DL model performance on test dataset (N=496).

Among the 496 cases in the test dataset, 126 cases had EDR delta-V available. For these cases, EDR delta-V was compared with both the WinSMASH delta-V (Fig. 8a), and the hybrid DL model predicted delta-V (Fig. 8b). The hybrid DL model performed marginally better than WinSMASH. WinSMASH underestimated delta-V by 14% on average with a RMSE of 8.3 kph (5.2 mph), whereas the hybrid DL model underestimated delta-V by 12% on average with a RMSE of 7.5 kph (4.6 mph).

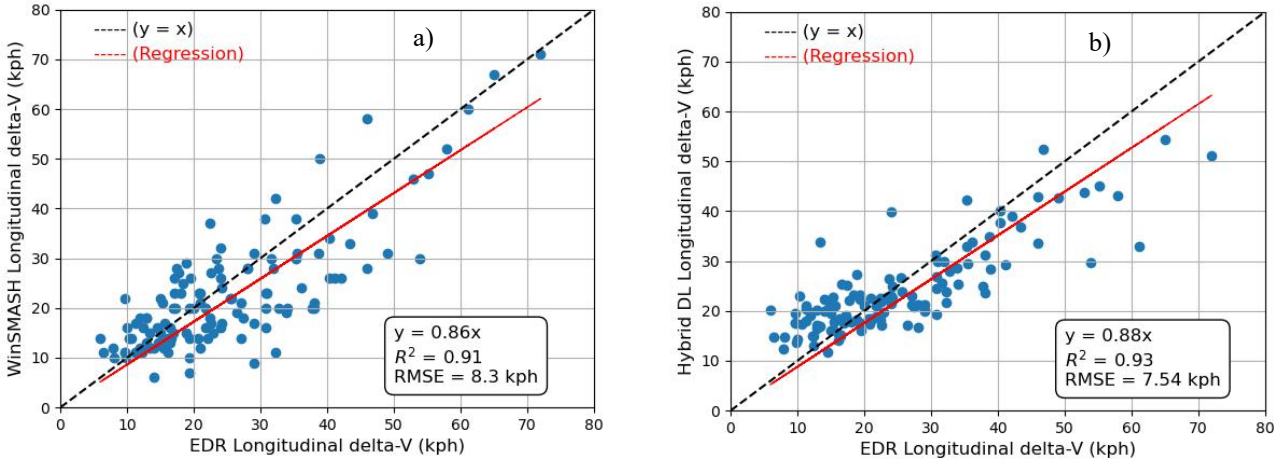


Fig. 8. Comparison between a) EDR delta-V and WinSMASH delta-V, and b) EDR delta-V and hybrid DL model predicted delta-V on the test dataset (N=126).

For the 126 test cases with EDR delta-V, both WinSMASH delta-V and hybrid DL model predicted delta-V were compared against EDR delta-V, and percentages of cases predicted within 5 mph (8.04 kph) and 10 mph (16.09 kph) of the EDR delta-V were computed (Fig. 9). The hybrid DL model predicted more cases within both 5 mph and 10 mph compared to WinSMASH.

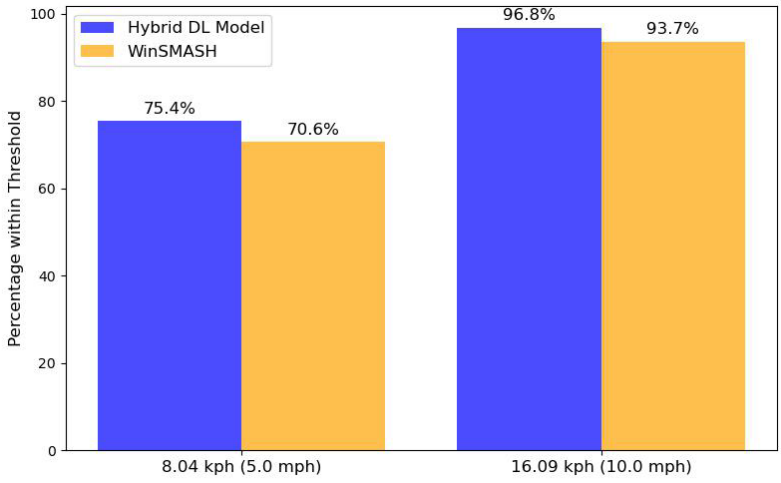


Fig. 9. Predictions within threshold for the hybrid DL model and WinSMASH.

To analyse how the prediction errors between EDR delta-V and hybrid DL model delta-V (EDR-DL) and between EDR delta-V and WinSMASH delta-V (EDR-WS) varied with EDR delta-V for the 126 test cases, two scatter plots were generated (Fig. 10). Fig 10a shows the absolute errors plotted against EDR delta-V values with the two horizontal dotted lines in red at 8.04 kph (5 mph) and 16.09 kph (10 mph). Fig 10b shows the relative errors plotted against EDR delta-V values. A second-order B-spline was fitted in both plots to observe error trends.

The hybrid DL model predicted within 5 mph for majority of the cases with delta-V below 50 kph. However, predictions errors increased for cases with higher delta-Vs exceeding 50 kph (Fig. 10a). In contrast, WinSMASH exhibited lower errors at the extremes, while the highest errors were observed in the 25 - 50 kph delta-V range (Fig. 10a). Across different delta-V magnitudes, both the hybrid DL model and WinSMASH demonstrated higher relative errors at very low delta-V values (below 10 kph) as depicted in Fig. 10b. Fig. 11 presents examples of hybrid DL model's prediction on both low and high severity crashes.

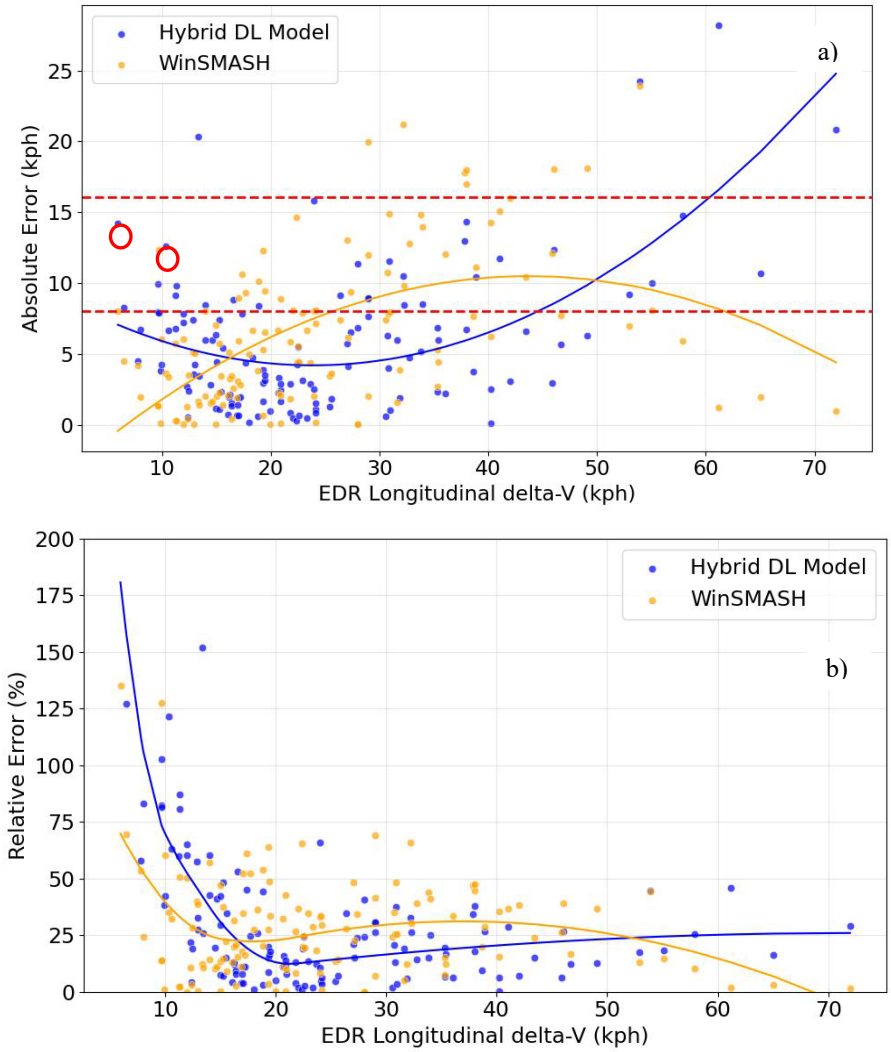
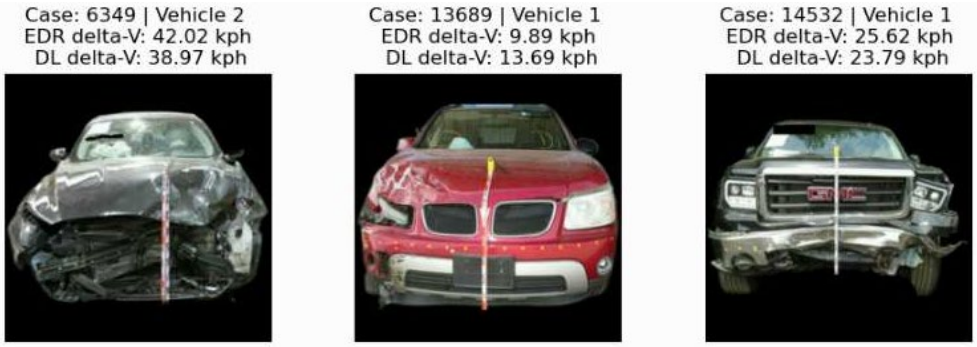


Fig. 10. a) Absolute errors, and b) Relative errors vs. EDR delta-V for Hybrid DL model and WinSMASH.

a)



b)

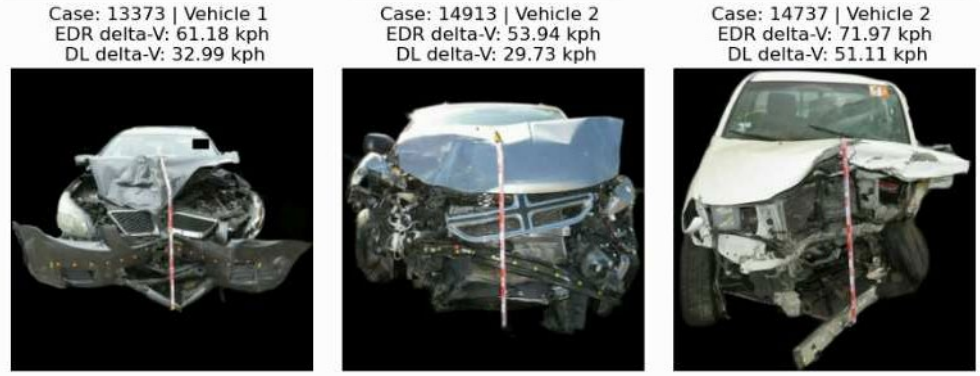


Fig. 11. Hybrid DL model's prediction on a) low severity crashes (< 50 kph), and b) high severity crashes (> 50 kph).

From Fig. 10a., a few additional cases marked with red circles were further examined. These cases had low EDR delta-V, yet the hybrid DL model predicted higher delta-V. These cases are presented in Fig. 12. Upon closer review, it was found that in these cases, the primary crash structures, such as bumper beams and longitudinal members, were never engaged. These crashes were identified as underride cases, leading to lower recorded EDR delta-V.

Fig. 12. Hybrid DL model prediction on underride cases.

To analyse how EDR-DL and EDR-WS prediction errors varied with PDOF for the 126 test cases, a bar plot was generated with PDOF on the x-axis and the mean prediction errors on the y-axis (Fig. 13). The horizontal dotted red line represents the 8.04 kph (5 mph) threshold. The hybrid DL model demonstrated lower mean prediction errors compared to WinSMASH for all PDOFs except for PDOF of 02 o'clock. The hybrid DL model showed mean prediction error under 8.04 kph (5 mph) for all PDOFs except 02 o'clock for which the mean prediction error was 10.35 kph (6.43 mph).

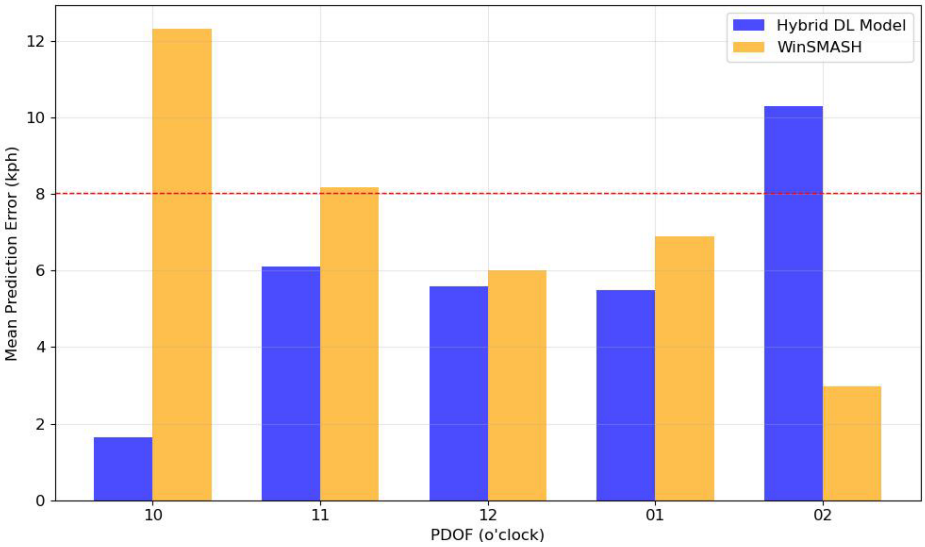


Fig. 13. Hybrid DL model and WinSMASH Prediction Errors vs. PDOF.

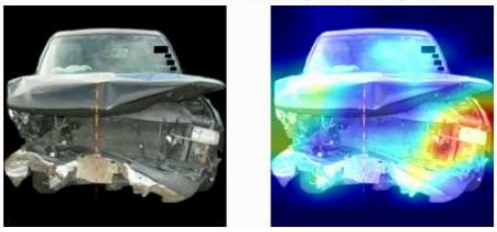
The training and inference time for the hybrid DL model was evaluated on a single Nvidia RTX-A6000 GPU. Training took approximately 3.5 hours for 46 epochs. Inference time averaged 160 ms per sample, making the model suitable for quickly predicting delta-V estimates.

Model Interpretation

The results of hybrid DL model interpretation using occlusion sensitivity are presented in Fig. 14 for different PDOFs. For each PDOF, two images are shown: the original crash image and the same image with an overlayed heatmap. The red regions represent areas of high importance that are critical for the model's prediction and cooler colors represent regions of lower importance. The results demonstrate that the model has learned to focus on the damaged areas of the vehicles, providing confidence in its decision-making process.

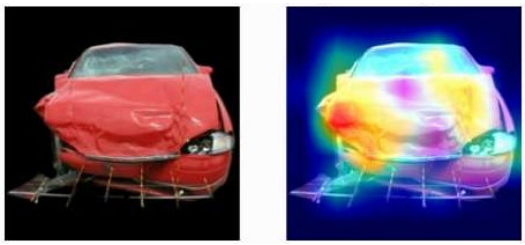
PDOF = 10 o'clock, EDR delta-V = 19.3 kph, DL delta-V = 17.6 kph

PDOF = 11 o'clock, EDR delta-V = 35.3 kph, DL delta-V = 33.0 kph



PDOF = 12 o'clock, EDR delta-V = 25.6 kph, DL delta-V = 23.7 kph

PDOF = 01 o'clock, EDR delta-V = 45.8 kph, DL delta-V = 42.9 kph



PDOF = 02 o'clock, EDR delta-V = 8.1 kph, DL delta-V = 14.7 kph

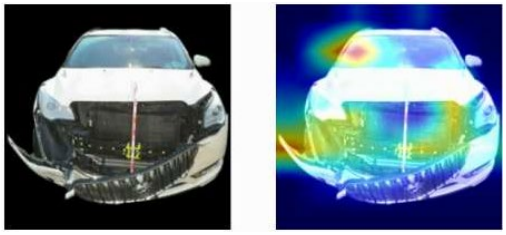


Fig. 14. Hybrid DL model Interpretation using occlusion sensitivity.

IV. DISCUSSION

In this study, a novel hybrid deep learning model that combines post-collision images and minimal vehicle data was developed to predict longitudinal delta-V for vehicle-to-vehicle impacts. Trained using EDR and EDR-equivalent delta-V, the model demonstrated marginally better performance compared to WinSMASH on the unseen test dataset of 126 EDR cases (Fig. 8). Specifically, the proposed model exhibited a lower average underestimation error (12% vs. 14%) and a reduced RMSE (7.5 kph vs. 8.3 kph) compared to WinSMASH. Importantly, the 126 EDR cases in the test dataset encompassed a diverse range of cases, including 46 unique CDC codes, PDOFs ranging from 10 to 2 o'clock, and delta-V values spanning from 5 to 72 kph, underscoring the model's performance across various frontal crash scenarios. On the overall test dataset of 496 cases, the model's average underestimation error was only 7% (Fig. 7).

For the majority of real-world crashes, delta-V typically falls within a range of 10–30 kph, and crashes occurring within ±30° frontal oblique angles (PDOF 11–1 o'clock) account for over 90% of all frontal crashes. Fig. 10a and Fig. 13 illustrate that the proposed hybrid DL model performed particularly well within this delta-V and PDOF range, highlighting its practical applicability to a substantial portion of real-world crash scenarios.

The proposed model reduces the reliance on detailed crash scene investigations, providing an efficient and practical solution for estimating delta-V using minimal vehicle data and post-collision images. This is particularly beneficial as approximately 41% of cases in the CISS/CDS databases are missing delta-V values. Within its validated bounds, the model can be applied to estimate delta-V for these cases. Furthermore, it can be used for predicting delta-V for new cases that fall within its validated range.

Previous study by Silver *et al.* [4] employed a limited dataset of only 155 crash images for their front-end model, sourced from the NASS-CDS database. Their dataset was constrained to a delta-V range of 10–40 kph and included only sedan and hatchback body types. Additionally, their study highlighted the inaccuracies in WinSMASH estimates but did not apply any corrections to the WinSMASH delta-V values, using them directly for model training. Their investigation was restricted to a Visual Geometry Group (VGG) style CNN architecture [29] with limited hyperparameter tuning, and no performance metrics were provided to compare their model against WinSMASH estimates on an independent test dataset. In contrast, this study utilised a significantly larger and more diverse dataset of 9,281 crash images, encompassing a delta-V range from 5–80 kph. The dataset included 11 distinct body types—sedan, compact utility, light pickup, station wagon, large utility, hatchback, large pickup, minivan, compact pickup, large van, and convertibles—representing the majority of vehicles on the road. To enhance model accuracy, the hybrid DL model was trained using EDR delta-V and EDR-equivalent delta-V (corrected WinSMASH estimates). Seven different CNN architectures (Appendix C, Table C1) were rigorously evaluated with exhaustive hyperparameter tuning before selecting the EfficientNetB5 architecture [30] for the final predictive model. This comprehensive approach ensured more accurate delta-V predictions across a wider range of scenarios.

The hybrid DL model developed in this study leverages EfficientNet architecture, which offers significant advantages over the VGG architecture by employing a compound scaling method that systematically and uniformly scales the depth, width, and resolution of the network, achieving superior performance and efficiency. In contrast, VGG architecture increases depth by simply stacking more layers, which can lead to inefficiencies and overfitting without delivering proportional gains in performance. EfficientNet incorporates advanced techniques such as depthwise separable convolutions [31] and squeeze-and-excitation modules [32], making it both efficient and robust. Additionally, EfficientNet has lower memory and computational requirements compared to VGG, making it well-suited for deployment on mobile or embedded systems—enabling practical applications for crash investigators. In contrast, VGG's high computational and memory demands make it less practical for resource-constrained environments.

While transformer-based architectures [33], which are the foundation of modern Large Language Models (LLMs), have shown success in computer vision tasks, they were not utilised in this study. EfficientNet's convolutional layers encode strong inductive biases, enabling effective learning from smaller datasets like the one used in this research. On the other hand, Vision Transformers [34], which rely on self-attention mechanisms

with weaker inductive biases, typically require much larger datasets to achieve good generalisation, making them less suitable for this study's data constraints.

Interpretability of machine learning models is essential as understanding the model's decision-making process ensures reliability and builds trust among practitioners. The occlusion sensitivity analysis (Fig. 14) revealed that the model correctly focused on the damaged regions of the vehicle in the crash images when predicting delta-V. This finding not only validates the model's ability to utilise relevant features in the images but also enhances confidence in its practical application. It is important to note that the deep learning models often leverage patterns within the complex feature space that statistically correlate with the target value, based on the distribution of the training data. As a result, they may prioritise regions that differ from human intuition.

While the model demonstrated better predictive performance overall, it exhibited higher prediction errors in certain scenarios:

- *Very Low (< 10 kph, Fig. 10b) and High (> 50 kph, Fig. 11b) delta-V cases:* Limited training data in this range contributed to reduced accuracy. Incorporating additional data or employing generative AI techniques to generate more images in this range for model training could help improve predictions.
- *Higher PDOF (2 o'clock, Fig. 13):* Limited training data and the use of a single frontal-plane image likely led to increased errors. Incorporating additional data and combining frontal images with side or oblique images may enhance model's performance in such cases.
- *Underride cases (Fig. 12):* Limited information provided by a single frontal image likely led to increased errors. Combining frontal images with side images may improve prediction for these cases.

While incorporating additional views, such as side and/or oblique, could potentially enhance the model's performance, frontal-plane images were selected to develop the hybrid DL model because they were consistently available for the majority of cases. Including side and oblique images would have reduced the dataset size. Moreover, relying on multiple views would limit the model's practical applicability, as it would be unable to perform inference when any of the required views are missing.

The hybrid DL model was trained using EDR delta-V and corrected WinSMASH estimates (EDR equivalent delta-V). While the WinSMASH estimates were corrected to the best extent possible given the available data, the corrected estimates still contain residual errors – albeit reduced compared to uncorrected WinSMASH estimates (Fig. 5), which can impact the performance of the hybrid DL model. Additionally, this study utilised EDR reported delta-V values as the ground truth for model training. However, prior research [35, 36] evaluated the performance of EDRs by comparing with accelerometers mounted onboard test vehicles subjected to controlled crash tests and found that longitudinal delta-V reported by EDRs may have an error of up to ±10%. No corrections were applied to the EDR reported delta-V values in this study; therefore, any inherent inaccuracies in the EDR data will carry over into the predictions of the hybrid DL model.

Future work may focus on methodologies to further improve the correction of WinSMASH estimates and gathering additional cases with EDR data for training to enhance model performance. Additionally, generative AI techniques may be explored to augment the dataset, addressing data scarcity in specific crash scenarios. Expanding the dataset with synthetic yet realistic crash images could improve model generalisation, particularly for underrepresented cases. Furthermore, the model may be extended to other crash types, including vehicle-to-object and side-impact crashes, to broaden its applicability.

V. CONCLUSIONS

The novel hybrid deep learning framework developed in this study demonstrated promising results in predicting delta-V, offering marginal improvements in accuracy and enhanced efficiency. By utilising only post-collision images and minimal vehicle data as inputs, the model reduces the dependency on detailed crash scene measurements or specific vehicle damage assessments.

VI. REFERENCES

[1] Gabauer, D.J., Gabler, H.C. (2008) Comparison of Roadside Crash Injury Metrics using Event Data Recorders, *Accident Analysis & Prevention,* **40** (2): pp. 548-558.

[2] Sharma, D., Stern, S., Brophy, J., Choi, E. (2007) An Overview of NHTSA's Crash Reconstruction Software WinSMASH, *Enhanced Safety of Vehicles.*

[3] Hampton, C.E., Gabler, H.C. (2010) Evaluation of the Accuracy of NASS/ CDS Estimates from the Enhanced WinSmash Algorithm, *Association for the Advancement of Automotive Medicine*, **54**: pp. 241-252.

[4] Silver, D., Manek, H., Kay, M., Travis, P. (2022) Estimating Automobile Crash Characteristics from Images using Deep Learning. The International FLAIRS Conference Proceedings, **35**.

[5] Baek, S., Yoon, J., Lim, J. (2022) A study on the development of delta-V prediction model for rear-end collision accidents using machine learning, *Transactions of the Korean Society of Automotive Engineers*, **30** (3): pp. 241-247.

[6] AGU Zurich, The Working Group on Accident Mechanics, http://crashdb.agu.ch/, 2021

[7] Patki, N., Wedge, R., Veeramachaneni, K. (2016) The Synthetic data vault, *IEEE International Conference on Data Science and Advanced Analytics,* **49**: pp. 399-410.

[8] DataCebo, Inc. (2023) Synthetic Data Metrics, Version 0.12.0, https://docs.sdv.dev/sdmetrics/

[9] Collision Deformation Classification (2022), *SAE International*, J224_202205.

[10] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**(1): pp.5–32.

[11] Ke, G., *et al.* (2017) LightGBM: a highly efficient gradient boosting decision tree. *International Conference on Neural Information Processing Systems,* **31**: pp. 3149-3157.

[12] Snoek, J., Larochelle, H., Adas, R.P., (2012) Practical Bayesian Optimization of Machine Learning Algorithms", *Advances in Neural Information Processing Systems*, **25**.

[13] Eisenhauer, J.G. (2003) Regression through the Origin, *Teaching Statistics*, **25**(3).

[14] Qin, X., et al. (2020) U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection, *Pattern Recognition*, **106**.

[15] Gatis, D., Rembg, https://github.com/danielgatis/rembg

[16] CVAT.ai Corporation (2023), Computer Vision Annotation Tool (CVAT), https://github.com/cvat-ai/cvat

[17] Loffe, S., Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *International Conference on Machine Learning*, **37**.

[18] Dubey, S.R., Singh, S.K., Chaudhuri, B.B., (2022) Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark, *Neurocomputing*, **503** (C): pp. 92-108.

[19] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014) Dropout: A simple way to prevent Neural Networks form Overfitting, *Journal of Machine Learning Research,* **15**: pp. 1929-1958.

[20] Abadi, M., *et al.* (2015) Tensorflow: Large-scale machine learning on heterogeneous systems, v2.17.

[21] Alzubaidi, L., *et al.* (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, *Journal of Big Data*, **53**.

[22] Deng, J., et al. (2009) ImageNet: A largescale hierarchical image database, *IEEE Conference on Computer Vision and Pattern Recognition.*

[23] Wood, L., *et al*. (2022) KerasCV, https://github.com/keras-team/keras-cv

[24] Liaw, R., et al. (2018) Tune: A Research Platform for Distributed Model Selection and Training, *ArXiv.*

[25] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., (2018) Hyperband: A novel bandit-based approach to hyperparameter optimization, *Journal of Machine Leaning Research*, **18** (185): pp. 1-52.

[26] Bergstra, J., Yamins, D., Cox, D.D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *Proceedings of the 30th International Conference on Machine Learning*, **28** (1): pp. 115 to 123.

[27] Zeiler, D., Fergus, R. (2014) Visualizing and Understanding Convolutional Networks, *European Conference on Computer Vision*.

[28] Pedregosa, F., *et al.* (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**: pp.2825–2830.

[29] Simonyan, K., Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition, *International Conference on Learning Representations.*

[30] Tan, M., Le, Q.V. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, *International Conference on Machine Learning.*

[31] Chollet, F. (2017) Xception: Deep Learning with Depthwise Separable Convolutions, *Computer Vision and Pattern Recognition*, pp. 1800-1807.
[32] Hu, J., Shen, L., Sun. G. (2018) Squeeze-and-Excitation Networks, *Computer Vision and Pattern Recognition,* pp. 7132-7141.
[33] Vaswani, A. et al. (2017) Attention Is All You Need, *Neural Information Processing Systems.*
[34] Dosovitskiy, A. et al. (2021) An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, *International Conference on Learning Representations.*
[35] Niehoff, P., Gabler, H., Brophy, J., Chidester, C., Hinch, J., Ragland, C. (2005) Evaluation of Event Data Recorders in Full Systems Crash Tests. *19th International Technical Conference on the Enhanced Safety of Vehicles (ESV).*
[36] King, C., Rich, A., Ruth, R., Sadrnia, H. (2024) Accuracy of 2016-2022 EDRs in IIHS Crash Tests, *SAE International*.

VII. DISCLAIMER

VIII. APPENDIX A: PHASE 1 AND PHASE 2 MODEL DEVELOPMENT

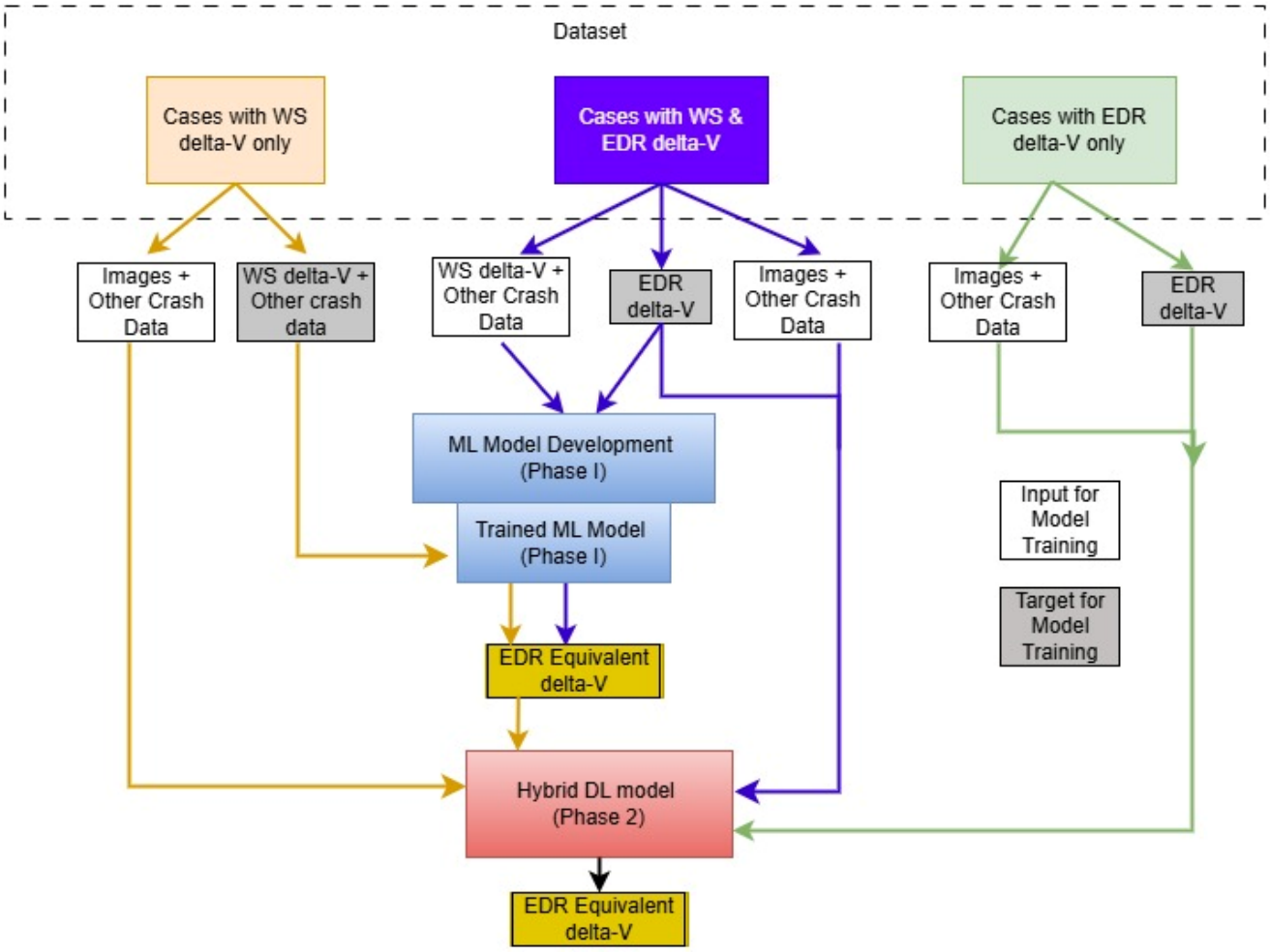


Fig. A1 Flowchart of model development.

IX. APPENDIX B: COMPARISON OF REAL AND SYNTHETIC DATA

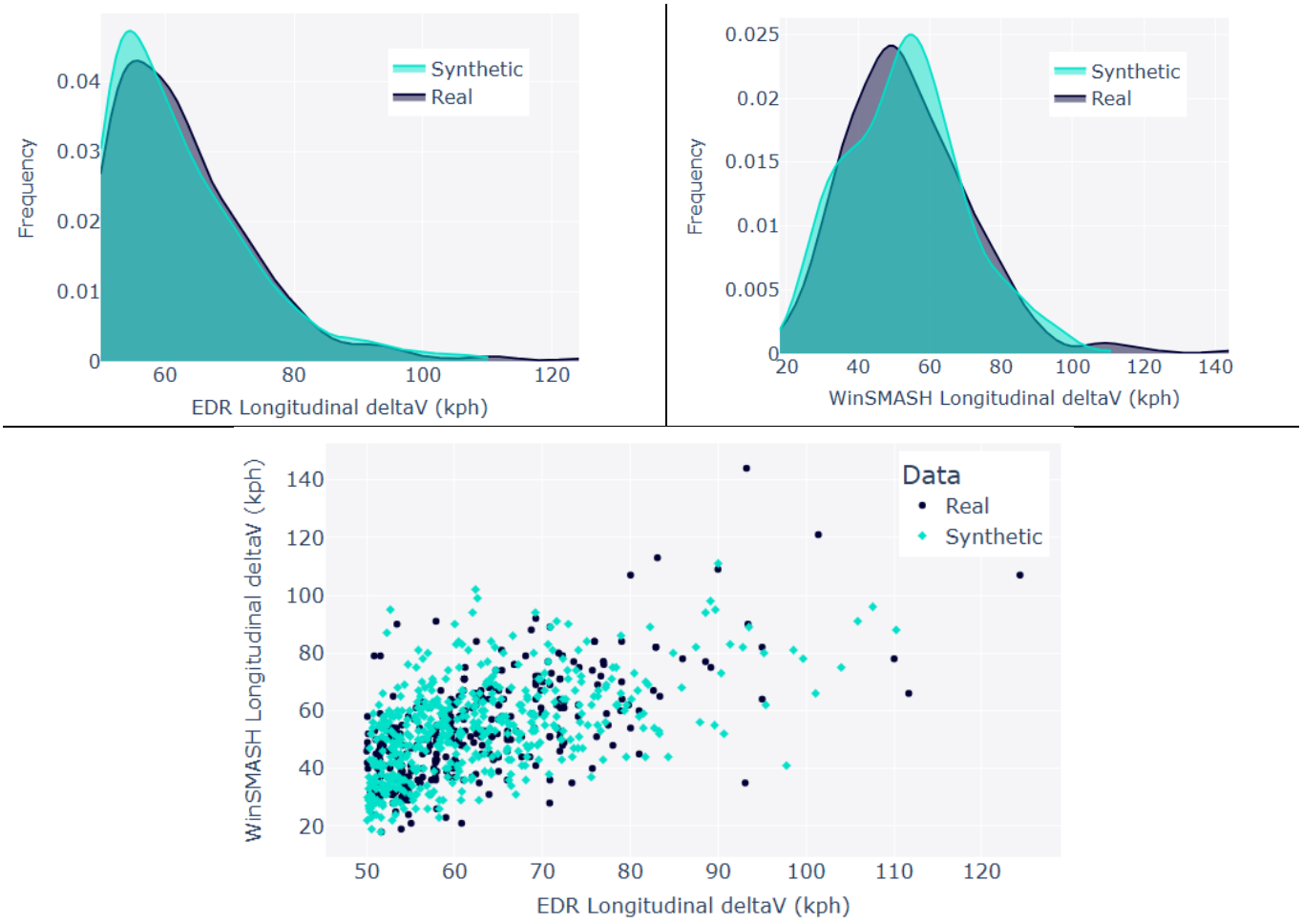


Fig. B1. Real vs synthetic data for WinSMASH and EDR longitudinal delta-V.

X. APPENDIX C: HYBRID DL MODEL HYPERPARAMETERS AND AUGMENTATIONS

TABLE C1

TUNED HYPERPARAMETERS

Hyperparameter	Values [20]
CNN-2D	EfficientNetB5, EfficientNetV2M, ConvNextBase, DenseNet201, InceptionResNetV2, Xception, ResNet152V2
Pooling Type	Global Average Pooling, Spatial Pyramid Pooling
Initialisers	He Normal, He Uniform, Glorot Normal, Glorot Uniform, Trunc Normal, Variance Scaling
Regularisers	L1, L2, L1L2
Regularisation Factor	Log uniform (1e-5, 1e-1)
Activation	ReLU, ELU, Swish, GeLU, Leaky ReLU
Leaky ReLU Slope	Uniform (0.01, 0.3)
First Dropout Layer Rate	Uniform (0.1, 0.6)
Second Dropout Layer Rate	Uniform (0.1, 0.6)
Output Layer Activation	ReLU, Linear
Learning Rate	Log uniform (1e-5, 1e-2)
Loss	Mean squared error, Mean absolute error, Huber
Optimiser	Adam, AdamW, Lion, Nadam
Adam Beta 1	Uniform (0.8, 0.99)
Adam Beta 2	Uniform (0.9, 0.99)
Adam Weight Decay	Uniform (1e-6, 1e-3)
First Fully Connected Layer Units	Random Integer (32, 256)
Second Fully Connected Layer Units	Random Integer (32, 256)

TABLE C2

IMAGE AUGMENTATIONS USED DURING MODEL TRAINING

Grid Mask	Random Zoom
Random Gaussian Blur	Random Brightness
Random Color Jitter	Random Sharpness
Random Rotation	

XI. APPENDIX D: ML MODEL AND HYBRID DL MODEL HYPERPARAMETERS

TABLE D1
RANDOM FOREST ML MODEL HYPERPARAMETERS

Hyperparameter [28]	Range	Optimised Value
n_estimators	10 - 500	300
max_depth	None, 2 - 100	None
criterion	squared_error, absolute_error, friedman_mse, poisson	squared_error
min_samples_split	2 – 10	8
min_samples_leaf	1 - 10	2
max_features	None, sqrt, log2	log2
max_samples	None, 0.1 – 1.0	None

TABLE D2
HYBRID DL MODEL OPTIMISED HYPERPARAMETERS

Hyperparameter	Optimised value [20]
CNN-2D	EfficientNetB5
Pooling Type	Global Average Pooling
Initialiser	Glorot Normal
Regulariser	L1L2
Regularisation Factor	3.01E-05
Activation	Swish
Leaky ReLU Slope	N/A
First Dropout Layer Rate	0.26
Second Dropout Layer Rate	0.55
Output Layer Activation	ReLU
Learning Rate	1.37E-04
Loss	Mean squared error
Optimiser	Nadam
Adam Beta 1	0.82
Adam Beta 2	0.94
Adam Weight Decay	3.3E-04
First Fully Connected Layer Units	304
Second Fully Connected Layer Units	373