Estimating Automobile Crash Characteristics from Images using Deep Learning

Hardik Manek1, Daniel L. Silver2, Matthew Kay1 and Paul Travis1
1Talem Health Analytics, 319 Charlotte Street, Suite 302, Sydney, Nova Scotia, Canada B1P 1E1
2Jodrey School of Computer Science, Acadia University, Wolfville, Nova Scotia, Canada B4P 2R6
hardik@talemhealthanalytics.com, danny.silver@acadiau.ca

Abstract
Crash characteristics such as crash velocity (Delta-V) and location of collision (LOC) are important determinants of the severity of the injury sustained by an occupant of an accident vehicle. Based on the predicted severity levels of injury, insurance companies can estimate the claim’s cost and better plan their financial reserves. We present a promising approach for accurately predicting Delta-V and LOC using deep learning methods, without the need for a forensic crash reconstruction. We constrain the study to small passenger vehicles and to front and rear collisions with crash velocities under 96 km/h. We first develop and refine our image processing and deep CNN architectures using images created by using vehicle crash simulation software. Using a k-fold cross-validation approach, our methods are able to predict the crash velocity of simulated collisions (108 images) with a MAE of 3.41 km/h (MAPE of 8.2%). Similarly, a multiple task learning CNN is able to predict Delta-V of real-world collisions (310 images) with a MAE of 4.19 km/h (MAPE of 16.2%) and classify the LOC with 92% accuracy.

Introduction
Early estimates of the severity of injury to the occupant of an accident vehicle are of considerable value to automobile insurers. Such estimates are the basis of forecasting a claim’s cost and the planning of the company’s financial reserves. Unfortunately, insurance providers rarely have access to medical data until several weeks following a claim. Crash characteristics such as crash velocity (Delta-V), location of collision (LOC) and principal direction of force (PDOF), can be used to predict the severity of the injury sustained by an occupant. Currently, the insurance industry estimates these characteristics by using either inexpensive manual methods that have wide margins for error or expensive simulated or physical accident reconstruction methods. For this reason, artificial intelligence is of interest to the InsurTech industry (Cortis et al. 2019). This paper presents an initial effort to estimate automobile crash characteristics, such as Delta-V and LOC, using post-collision photographs of vehicles and deep learning methods (Manek 2021).

This research has been completed in collaboration with Talem Health Analytics, Canada. Talem provided the problem definition, system requirements and forensic insights.

The objective is to prove the technical feasibility of predicting physical crash characteristics from images using computer vision and machine learning within an acceptable range of accuracy. To the best of our knowledge, there has been no prior work in this area.

We constrain the problem to front-end and rear-end collisions involving mid-size passenger vehicles such as sedans and minivans. The range of crash velocity considered for the simulated images is from 10 to 96 km/h, and for real-world collision images, the range of Delta-V considered is from 10 to 40 km/h. Our success criteria is to classify LOC with an accuracy of 95% and to predict Delta-V with a mean absolute error (MAE) of no more than 10 km/h. Because real-world PDOF ground truth values are difficult to obtain we will be focusing on its prediction in future work.

Background
Crash characteristics, such as Delta-V, LOC, and PDOF, detail the physics of a vehicle collision and can be used to estimate the severity of injuries sustained by the occupants. Delta-V for a vehicle is simply the difference between the post-collision velocity \( v' \) and pre-collision velocity \( v \), given by \( \Delta v = v' - v \) (Shelby 2011). The higher the value of Delta-V, the higher the severity of injury to the occupants of the vehicle. Unfortunately, Delta-V has been of limited use because estimating it requires a detailed forensic investigation to systematically measure the velocities and the vehicle’s deformation at the time of impact.

The Abbreviated Injury Scale (AIS) severity score is an ordinal score from 1 (minor injury) to 6 (maximum injury) based on prior crash history. A car crash that sustains an injury with a score of 3 or higher on the AIS is classified as a serious injury. For example, the risk of a serious injury at Delta-V of 48 km/h is 38.9% for front-end collisions, 83.8% for near-side collisions (same side as occupant), 47.8% for far-side collisions (opposite side to occupant) and 19.9% for rear-end collisions (Bahouth et al. 2004). Therefore, injury risk is a function of Delta-V and LOC.

Event Data Recorders (EDRs) installed in modern automobiles are able to record impact velocities and from this PDOF values can be calculated and LOC inferred. However, due to privacy laws, EDRs can be accessed only by the Original Equipment Manufacturers (OEMs) and government organizations who may be investigating accidents. They cannot be accessed by insurance companies.
The traditional methods for recording crash characteristics fall into four categories. (1) Manual estimation using details of the crash and photographs captured at the accident scene are prone to wide margins of error. (2) Mathematical models of the crash based on accident scene details require special knowledge to set up and analyse and so does not scale well. Lumped parameter modeling and the finite element method are the most popular analytical tools for car crash modeling (Pawlus, Robbersmyr, and Karimi 2011). (3) Physical reconstruction of a crash is expensive and is normally only undertaken when there is a questionable high claim. (4) Simulated reconstruction of a crash is becoming easier to do with software such as WinSmash developed by the NHTSA (Nance et al. 2006), however it still requires special knowledge and has known problems with its predictions (Hampton and Gabler 2009).

**Theory and Approach**

The problem of predicting crash characteristics, such as Delta-V, LOC, and PDOF from images is unexplored. Humans use their vision system to process images of an accident vehicle to estimate the associated crash characteristics. To gauge the depth of a car’s deformation from a single image, humans use the occipital and temporal lobes of their cerebral cortex, including common knowledge from past experiences of the world (Freeman 1999). Unfortunately, such common knowledge is not available to current AI systems.

**Hypothesis**

As crash velocity increases, the energy absorbed by a vehicle also increases, which in-turn increases the depth and width of the deformation of that vehicle. Therefore, crash velocity is systematically related to the nature of deformation of a vehicle. We propose that there is a collection of deformation patterns for a particular crash velocity range for each type of vehicle and for each location of impact (front, rear, or side). Images of the crashed vehicle capture those deformation patterns in the form of pixel values. We propose that, given sufficient images and matching target values, a deep learning algorithm can train a neural network to extract the features from the pixels that relate the nature of the deformation to the crash velocity. Similarly, the problem of classifying the LOC involves learning the pixel features that characterize the location of the deformation on the vehicle.

**Approach**

To test our hypothesis of predicting crash characteristics from images, we first develop and refine our image processing and deep convolutional neural network architectures using images created by vehicle crash simulation software. Using this knowledge, we select real-world images of accident vehicles, extracted from a United States government database and use them to further develop and test our approach to predict Delta-V and LOC values.

**Simulated collision images.** Rigs of Rods is an open source vehicle simulator that uses soft-body physics to simulate the motion and deformation of vehicles (Öhldal 2020). The deformations generated by the software are close approximations to that of real crashes. Simulation provides the flexibility to control factors such as Delta-V, LOC and PDOF on the vehicle, as well as the type of vehicle, its colour, brightness and contrast levels of light, and the distance and angle the camera is from the vehicle. Figure 2 shows the post-collision images of the two vehicles used in the simulations; a red Honda Accord and a grey Ford Windstar.

**Real-world collision images.** The National Highway Traffic Safety Administration (NHTSA) is a government organization responsible for researching injuries during a car crash. NHTSA examines real-world collisions, records the crash images and associated data in their National Automotive Sampling System (NASS) database (USA Dept of Transportation). Figure 8 shows examples of real-world collision images extracted from the NASS’s website using a web-scraper that we also developed.

**Image Processing.** We implemented an image processing pipeline that transforms the crash images into a standard format for the machine learning software. The pipeline consists of the follow steps: (1) crop the images using Adobe Photoshop to reduce background noise; (2) create single-channel greyscale images from the 3-channel RGB colour images using OpenCV; and (3) resize all images to 250 x 250 pixels and normalize pixel values ready for input to the neural network. The simulated images are already n x n square, so the OpenCV resize function can be used directly to do this. However, the real-world collision images are rarely square, so they are first padded with white background pixels to make them n x n square.

**Deep Learning Approaches Considered.** To gauge the feasibility of predicting the crash characteristics within an acceptable range of accuracy we train Convolutional Neural Networks (CNN) from scratch. We will experiment with various CNN architectures, including VGG type networks where the layers are arranged in the form of blocks consisting of convolutional, max-pooling layers, and, potentially, dropout layers (Simonyan and Zisserman 2015). At the top of the network are fully connected dense layers. Early stopping is used as the regularization technique to promote model generalization. The CNN models are built using Python and the Keras and Tensorflow libraries that take advantage of a NVIDIA GeForce GTX 1660 Ti GPU.

**Empirical Studies**

This section describes experiments that test our hypothesis. The first two subsections report on experiments that develop models to predict the crash velocities of simulated collision images. The simulation images provide a controlled input space to develop a proof-of-concept for Delta-V prediction and to determine the key factors for success. The last two subsections present experiments performed to predict Delta-V and classify the LOC from real-world collision images.

**Baseline Studies using Simulation Images**

**Objective:** The first step is to test the ability to develop deep learning models that can accurately predict the velocity of a crash using simulation images. Separate deep CNN models are developed and tested using images of the two vehicles and their performances are compared.
Data and Methods: The dataset consists of 54 simulation images of a red Honda Accord and 54 simulation images of a grey Ford Windstar LX colliding with a brick wall at speeds ranging from 10 to 96 kph (uniform random distribution). Figure 2 depicts two images of the simulated crash vehicles used in the experiments.

The best network architecture for all the simulation image experiments is a deep VGG CNN consisting of 3 convolutional blocks and 1 fully connected block (see Figure 1). The Rectified Linear Unit (ReLU) is used as the activation function in the convolutional blocks. A linear activation function in the last layer outputs the crash velocity. We use the Adam optimizer with a learning rate of 0.0001. Each network is trained for 1000 epochs, with a batch size of 12 images.

Figure 1: Basic VGG CNN architecture.

To thoroughly test our method, a 9-fold cross-validation technique is used; the complete dataset of 54 images is split into 9 folds, where for each model, 7 folds of data (42 images) are used for training and 1 fold (6 images) for validation, and the remaining fold (6 images) is used as the test set. The validation loss is monitored to select the best weights. The test set is used to measure each model’s performance.

Results and Discussion: The models developed using the Honda Accord images achieved an MAE of 3.77 kph with a 95% confidence interval of ±0.92 kph (MAPE of 8.70%) and the Ford Windstar models had an MAE of 7.80 ± 1.81 kph (MAPE of 24.70%). The results show that one can develop accurate models for estimating crash velocity from images. The scatter plot and correlation between the predicted values and actual values for all 54 images for the Honda and the Ford models are shown in Figure 3. There is a significant difference in model accuracy (t-test p-value = 0.001782) in favour of the models developed for the red Honda. This variation in accuracy is due to either the difference in the vehicle geometry or colour of the vehicles.

Determining Factors Affecting Model Performance using Simulation Images

Objectives: To determine the key factors affecting the performance of the models we created test images from the red Honda Accord and grey Ford Windstar images, with variations in vehicle geometry and color. Each of these variations were tested using one of the best deep CNN models developed using the red Honda Accord simulation images.

Data and Methods: From Figure 2 one can see that a significant difference in the geometry of the two vehicles is the headlights. The Ford Windstar’s headlights are large compared to the Honda Accord’s. We used Photoshop to remove the headlights from the Ford making it look more like the Honda. We also created a second Ford test set with no headlights and a red body (Figure 4). Finally, we used Photoshop to modify the red Honda Accord images to create 54 new blue Honda images and 54 new green Honda images.

Results and Discussion: The top half of Table 1 summarizes the results of the model developed using red Honda images and tested on the original grey Ford Windstar images, modified grey Ford without headlights, and modified red Ford without headlights. The Honda Accord model’s MAE improves when tested on images of the modified Ford; most significantly as a result of the change in the headlights. We conclude that a vehicle’s geometry has the most significant effect on the deep learning model’s performance.

The bottom half of Table 1 shows a comparison of the same model tested on the red, blue, and green Honda im-
ages. The t-test p-values indicate a significant reduction in the model’s performance when tested on different car colors (0.0336 blue, 0.0192 green). This suggests that there may be benefit in eliminating the model’s color dependency.

Table 1: Performance of Honda Accord model on various Honda Accord and Ford Windstar images.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Original Ford</th>
<th>Grey No lights</th>
<th>Red no lights.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>21.73 ± 3.04</td>
<td>16.91 ± 3.90</td>
<td>16.72 ± 3.81</td>
</tr>
<tr>
<td>MAPE</td>
<td>72.90%</td>
<td>38.5%</td>
<td>38.8%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Red Honda</th>
<th>Blue Honda</th>
<th>Green Honda</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>3.77 ± 0.92</td>
<td>4.92 ± 1.32</td>
<td>5.31 ± 1.32</td>
</tr>
<tr>
<td>MAPE</td>
<td>8.70%</td>
<td>12.90%</td>
<td>12.70%</td>
</tr>
</tbody>
</table>

Combined Dataset of Honda Accord and Ford Windstar Images

Objective: To develop and test models using the combined images of the Honda Accord and Ford Windstar. This experiment also tests the generalization accuracy of CNN models developed from greyscale versus colour images.

Data and Methods: The 54 simulation images of Honda Accord and Ford Windstar are combined to create a dataset of 108 examples. Separate models are trained using (1) the color images and (2) the greyscale images. A 9-fold cross-validation technique is used; 84 images for the training set and 12 images for each of the validation and test sets. The network configuration and learning parameters are the same as that of the last experiment.

Results and Discussion: The models developed using the coloured images achieved an MAE of 3.94 ± 0.77 (MAPE of 10.70%) and the greyscale image models had an MAE of 3.63 ± 0.75 (MAPE of 11.06%). Figure 5 compares the scatter plots of the models built and tested using the color images and those built and tested using the greyscale images. The p-value of the t-test between the MAE of the two models is 0.5289; indicating no significant difference between the performance of the model developed using the color versus greyscale images. However, the model trained using the greyscale images has a slightly lower MAE. Furthermore, greyscale images eliminates the color dependency problem, and requires less memory and computational resources than color images for network training.

Studies Using Real-world Images

In the prior section, we showed that crash velocities can be predicted from simulation images of accident vehicles using deep learning. Crash velocity is the instantaneous velocity at the time of a crash. Delta-V is the difference between post-collision and pre-collision velocity reflecting the relative crash velocity when one or more vehicles are involved in a collision (Sharma et al. 2007). Therefore, we feel confident that Delta-V can be predicted from images using the same image preparation and deep learning methods.

Models for Front and Rear Collisions Using Real-world Images

Objective: To train a CNN model to develop a mapping function from vehicle deformation as seen in real-world crash images to corresponding Delta-V values.

Data and Methods: The real-world crash data comes from the NASS databases curated by NHTSA. From this database we extract a collection of front and rear car crash images and associated crash characteristics. Figure 8 shows two example images after being prepared and converted to greyscale. Each front and rear collection, consists of 155 images, where there are five images for each Delta-V value ranging from 10 to 40 kph, in the steps of 1 kph. A 5-fold cross-validation technique is implemented to develop the models; 108 images are used for the training set, 16 images for the validation set, and 31 images for the test set.

The simulated images experiments employed a VGG CNN architecture. When we tried this same architecture with real-world collision images, we found that the performance was significantly less than with the simulation images. The real-world collision data has greater variability; for each value of Delta-V the associated images can vary more in terms background and deformation patterns. For example, the damage to the two vehicles in Figure 8 looks quite different but they have the same Delta-V of 10 kph. Accordingly, for this experiment, instead of blocks of layers, we have simplified the architecture to contain individual layers which promotes greater generalization. Figure 6 shows the architecture of the CNN. In total, 9 layers are used; 3 pairs of convolutional and max-pooling layers, 1 flattening layer, and 2 fully connected layers.

Results and Discussion: The MAE of the front-end image models is 3.83 kph, with a 95% confidence interval of ±0.91 kph. The MAPE is 16.90% with an $R^2$ value of 0.4688. The MAE of the rear-end image models is 3.49 kph, with a 95% confidence interval of ±0.83 kph. The MAPE is 18.18% with an $R^2$ value of 0.5272.

From the scatter-plot of predicted versus actuals shown in Figure 7, we observe that a few predictions by the front-end collision model are particularly inaccurate (marked with a red circle). The left-hand image of Figure 8 shows an outlier vehicle predicted to have a Delta-V of 28 kph, which
Figure 6: CNN architecture used for the real-world data experiments.

the NASS dataset claims to be 10 kph. By comparing this image with several others such as the right-hand image of Figure 8, the outlier’s deformation pattern looks far greater than 10 kph. We conclude that there may be some errors in the ground-truth Delta-V values selected from the NASS dataset.

Figure 7: Scatter plot of actual vs. predicted Delta-V (kph) for real-world front-end collisions.

Figure 8: Real-world actual and predictions - outlier (left), more typical (right).

MTL Models to Predict Delta-V and LOC

Objective: To test the learning of two tasks using a Multi-Task Learning (MTL) CNN (Caruana 1997); the prediction of Delta-V and the classification of LOC for a given image. We propose that a deep MTL CNN can be trained to predict both values, simultaneously, better than a single-task learning (STL) CNN, which predicts only Delta-V. By being forced to predict both crash characteristics for each training example, a deep MTL CNN should develop shared representation from a greater number of examples that can better extract features useful for both tasks.

Data and Methods: The real-world front and rear-end collision images from the prior experiment are combined to form a dataset of 310 images. The image processing pipeline described in the Approach section is applied to create the greyscale images used to train and test the MTL networks. A 5-fold cross-validation approach is used for model development and testing; where 217 images are used for training, 31 images for validation, and 62 images for testing.

Figure 9 shows the architecture of the MTL network used in this final experiment. To learn two tasks simultaneously, the MTL network must be a little larger than its STL counterpart to create the shared representation needed by both tasks. The MTL network has two output branches with different output nodes; a Linear node for Delta-V and two Softmax nodes for LOC classification (front and rear). The learning hyper-parameters are the same as the last experiment, but with the learning rate reduced to 0.000001.

Figure 9: MTL architecture to predict Delta-V and classify the LOC from real-world images.

Results and Discussion: The MTL models correctly classify the LOC on 92% of the test images. However, the MAE for predicting Delta-V is 4.19 kph, with a 95% CI of ±0.702 kph (MAPE of 16.20%). This is not as good as the performance of the counterpart STL CNN models.

Figure 10 shows the scatter plot and correlation between the actual and predicted Delta-V values by the MTL models. The models frequently error by under-predicting high Delta-V values. Figure 11 shows two front-end and two rear-end collision images and their actual and predicted Delta-V and LOC. The images suggest that the deformation patterns for front and rear collisions are quite different at higher velocities. For example, consider the bottom two images of Figure 11 with the same actual Delta-V value. For the front-end collision, the bonnet and bumper are compressed inwards, whereas the rear-end collision has the trunk lid driven up and outward. These differences in deformation patterns suggest that an MTL approach for this problem may not be best.
MTL CNN models, developed using real-world images of crash vehicles, can predict Delta-V with an MAE of 4.19 kph and classify the LOC with an accuracy of 92%, simultaneously. We have also shown that soft-body simulation is an effective method of investigating key vehicle and environmental crash factors for predictive model development.

There are several directions for future research. Software could be developed to automate the image cropping task to reduce background noise. A semi-supervised machine learning approach could be used to develop a MTL model that acts as an autoencoder to reconstruct the input image while at the same time predicting crash characteristics. And 3D image representations can now be capture using inexpensive cameras that employ LIDAR technology. This would provide more accurate deformation data to improve the accuracy of the predictive models.

References


