# Car Damage Classification using a Convolutional Neural Network

N.Tellez[1], M.Moreno[1] , L. San Pedro[2], B. Celly[3]
Kean University, Union, NJ 07083[1]
University of Texas Rio Grande Valley, Edinburg, TX 78539[2]
California State University Dominguez Hill, Carson, CA 90747[3]

Keywords: CNN, Car Damage, Vandalism, Image Classification, Insurance Claim.

Reviewing car damage claims in a timely manner is a major problem that car insurance companies still deal with today that leads to lengthy processes, consuming time and money along the way. Car accidents happen frequently every day as well, while insurance companies take time to provide coverage or compensation. Insurance claims typically have a timetable of 30 days, rarely meeting the deadline. Image classification to assess car accident pictures for car damage could accelerate processing these insurance claims significantly. The use of car classification in deep learning is a textbook example of the use of convolutional neural networks (CNNs) and in the case of car damage, it is the process of manually classifying a dataset of car images and labeling them binarily, either as a damaged car or an undamaged car.

The intention of this research project is to develop a CNN for the possibility of automating the review of car accident pictures for car damage to speed up lengthy car insurance claims process. CNNs are made up of multiple layers. There are a couple of layers that make it unique -- the convolutional layer and the pooling layers. The convolutional networks have the ability to learn these characteristics or filters on the images.

The data gathered for the CNN is made up of whole car images with either damage or no damage on the vehicles. These images are taken from several areas on the internet such as Google, Yandex, and from other car datasets on Kaggle. After selecting images that fit the experiment, they are preprocessed, using data augmentation techniques. The entire dataset of 1,638 car images is then split into two sets. One set is used to train the model, while the other set is used to validate the model. This is done to prevent overfitting during training to prevent bias and check for accuracy.

It should be noted that the more data CNNs have access to, the more effective they can be. Because there is a limited amount of images the model has access to, data augmentation is used to provide more training data and reduce overfitting on models. The data augmentation techniques used include horizontally flipping, rotating, shearing, etc.

With CNNs, the pixels are grouped in such a way that each node in the CNN will contain a distinct feature. For example, a node could contain something as simple as the shape of a regular car, or eventually something as specific as a type of scratch that would appear on a car. With each sequential layer, the layers design themselves to detect features that are more and more distinct. But because of how distinct these features could get, and how small a dataset could be, this could lead to fit a certain type of problem or a bias towards a sample of the population.

The methods in which we achieve a respectable car damage classification CNN is by using three pairs of convolution and max pooling layers using the Keras sequential API. The convolution layer takes the features of the input image pixels. Then, the max pooling layers extracts the most prominent features which helps decrease computational processing. These layers reduce the images into a form that is easier to process without losing the distinct features that are used to make a prediction. The resulting output from those layers are then flattened into a single array, which becomes fully connected to a dense layer which, depending on their activations, will activate or deactivate the output layer made up of 1 neuron -- indicating that the car in the image is either damaged or not. Before training the CNN, hyperparameters are used to set variables that determine how the training data is processed. Hyperparameter values are important because they provide training and validation accuracy without overfitting. For future improvements on the model, it is expected to tweak these values to gain a better accuracy and lower loss function, indicating a lower chance for error in predicting car damage.

After training, the CNN model was validated to have an 84.44% accuracy and the training accuracy was shown to be 93.75%. The CNN with this validation accuracy was obtained from the best model out of 17 epochs, or the number of times the training algorithm has gone through the dataset.

In conclusion, there are always improvements to be made in neural networks. There are sizable amounts of things to change, from adding more images to the dataset, to changing the hyperparameters used. But with this newfound knowledge and an eventual respectable accuracy, there are many possibilities. Eventually, a model could be trained to figure out how to decide a case between multiple wrecks of cars. Additionally, insurance companies could benefit from implementing a model like this in reviewing insurance claim images because it saves time and money.