Car Damage Classification Using a Convolutional Neural Network

Neil Tellez[1], Maria Moreno[1], Liam San Pedro[2], Dr. Bhrigu Celly[3]

[1]Kean University, [2]University of Texas RGV, [3]California State University Dominguez Hills

Abstract

Image classification to review car insurance claim images for car damage could accelerate processing insurance claims significantly. In this research project, we developed a Convolutional Neural Network (CNN) model using Keras, a high-level neural network API for Python that is built on top of TensorFlow, to classify car damage photos which could reduce the need for manual review during often lengthy insurance claim processes. The CNN model was trained using only car images displaying entire car bodies either damaged or undamaged. Images exhibiting damaged or undamaged cars were sorted in their respective folders, then split into two sets with one to train the model and another to validate the model accuracy which was used to prevent overfitting, or training data bias. To expand the variety of training data, we used data augmentation techniques which helped reduce the chance of overfitting the model. The results of the best CNN model we found was validated to have an accuracy of 84.4% and a training data accuracy of 93.75%. For future improvements on the model, more data can be collected and the hyperparameters, along with data augmentation techniques we used, could be tweaked to obtain a better accuracy.

Introduction

Car accidents happen frequently every day while insurance companies take time to provide coverage or compensation. Classifying a car as either damaged or undamaged is a classic binary problem that a CNN could be used for. In theory, CNN models can be continuously improved with new data inputs because each of the layers in the network adjust the weight and biases of the neurons highlighting different features of a damaged and undamaged car. The model then should make the best prediction of how likely a car in an image is either damaged or undamaged by generalizing what features of a damaged or undamaged car would have.

Background

Car classification with deep learning is a textbook example of the use of CNNs. 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 or an undamaged car. CNNs can be used to automate and accelerate long manual reviewing processes.

Problem Statement

Reviewing car damage claims in a timely manner is a major problem that car insurance companies still deal with today -- leading to processes that could last months, consuming time and money along the way. Insurance claims typically have a timetable of 30 days, rarely meeting the deadline.

Hypothesis

By providing a reliable use of car damage classification, a CNN can be used to automate the review of car insurance claim images which results in reduced stress, time and money expenditure during the insurance claim process. Overtime as more data is gathered, predictions and insurance claims processes should improve significantly.

Dataset

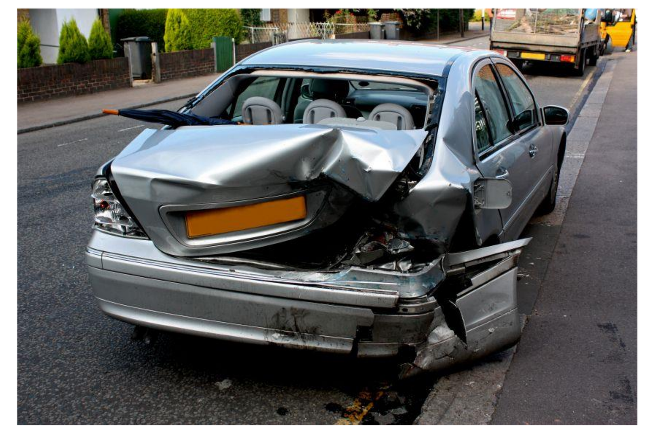


Figure 1. Example of a car damage image

The data gathered for the CNN is made up of whole car images with either damage or no damage on the vehicles. These images have been collected from several areas on the internet such as Google, Yandex, Kaggle and we used keywords such as 'car', 'accident', 'damage' and 'vandalism'. The entire dataset consists of 1,638 car images which is then split into two sets. One set is used to train the model, while the other set is used to validate the model. The validation set is used to check the model for overfitting, or has high bias for, the training data. This ensures the model is generally accurate for data it has not seen or trained with.

Methods

Before developing the layers for the CNN model, we augmented the training dataset to provide more training data and to reduce the chance of overfitting the model. The data augmentation techniques used include horizontally flipping, rotating, shearing and cropping images.

The methods in which we developed a car damage classification CNN model is by using three pairs of convolution and max pooling layers using the Keras sequential API. The resulting output from those three pairs of layers are then flattened into a single array, which becomes fully connected into a dense layer which, depending on their neuron activations, will either activate or deactivate the output layer made up of 1 neuron -- indicating that the car in the image is either damaged or not.

The convolution layer takes the features of the car from the input image pixels and uses the rectified linear activation function (ReLu) for the model to have better performance. Then, the max pooling layers extracts the most prominent features of the image reducing the number of pixels, 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 an accurate prediction.

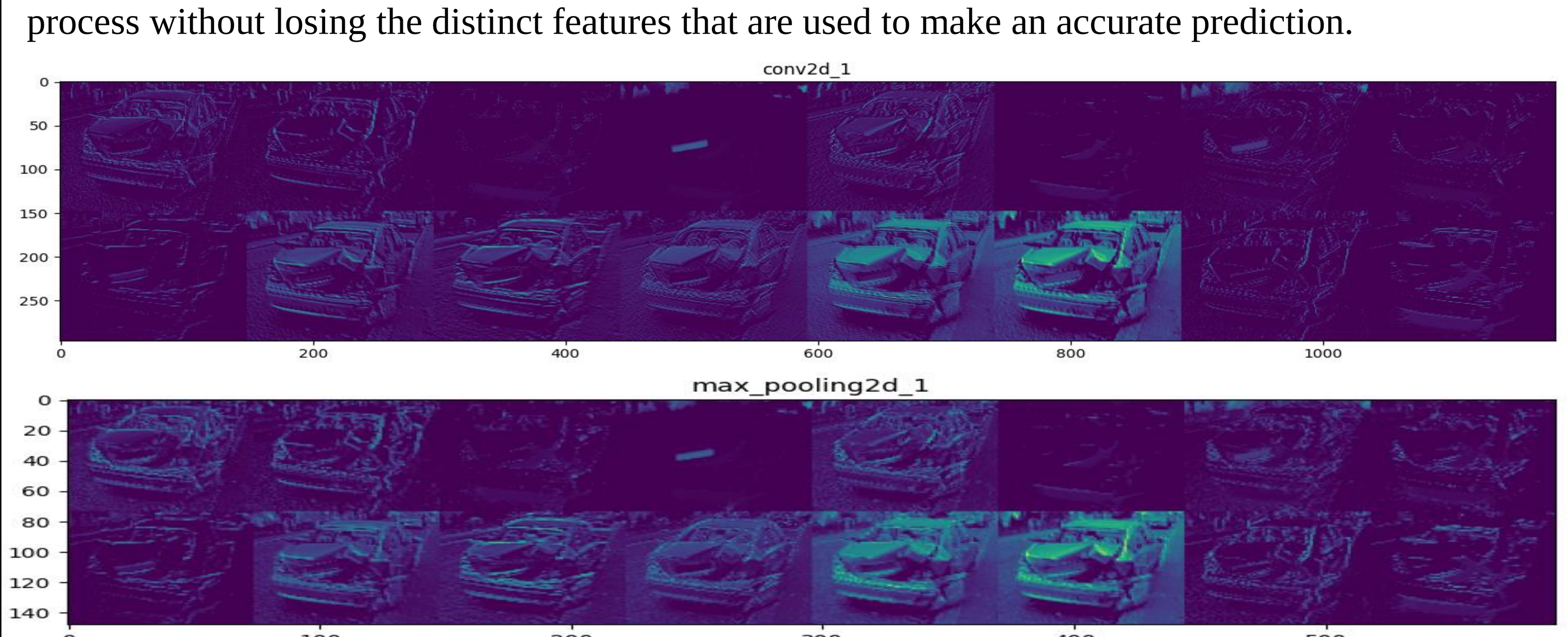


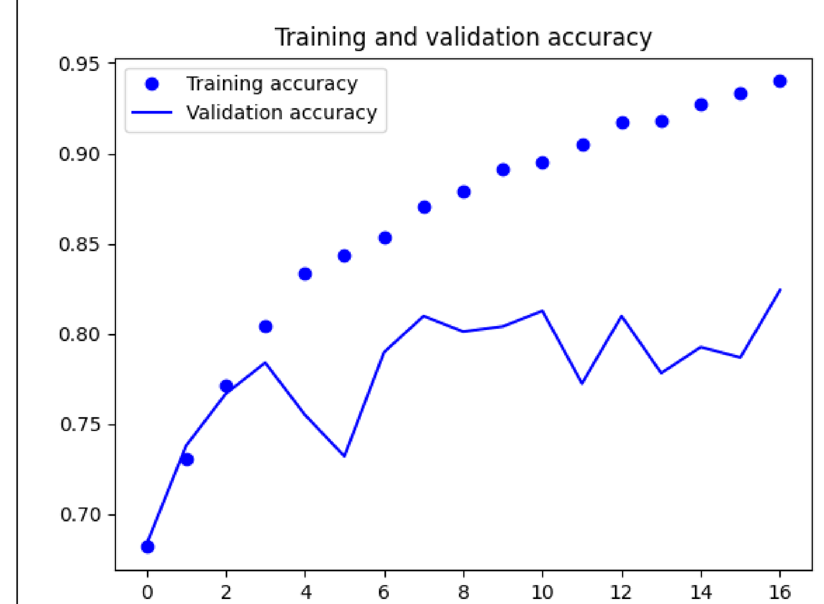
Figure 2.
The second layer of the convolution and max pooling layer visualized

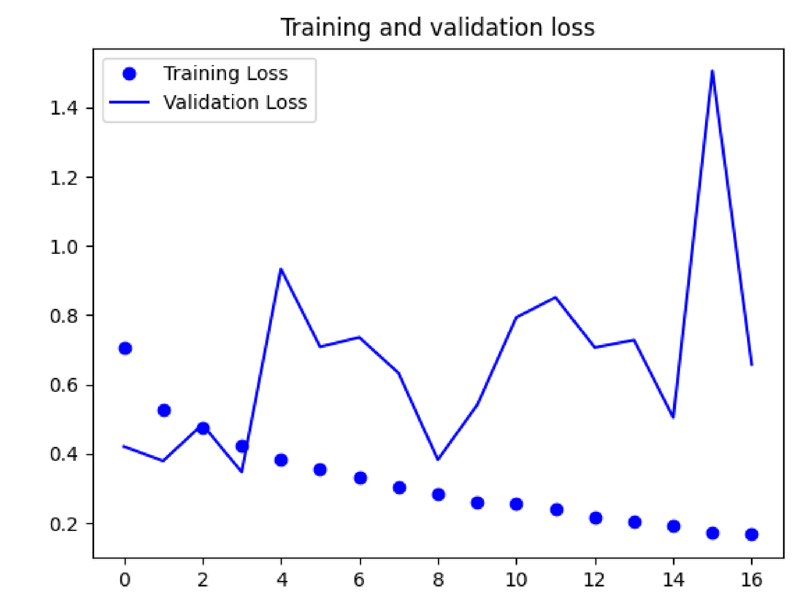
Results

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

To obtain these results we tuned the batch size, epoch, epoch step and learning rate to be 40, 17, 200 and 0.001, respectively. These are the hyperparameter configurations we found to have provided the best training and validation accuracy without overfitting. The batch size is the number of images that will be processed before the model is updated. The epoch is the number of times a model processes a dataset. The epoch step is the number of batches that is considered one epoch. The learning rate is a value between 0.0 and 0.1 that determines how large of a change the CNN model makes for each update.

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.





Conclusions

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. Insurance companies could benefit from implementing a model like this in their business because it can reduce stress, time and money expenditure during the insurance claim process.

References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

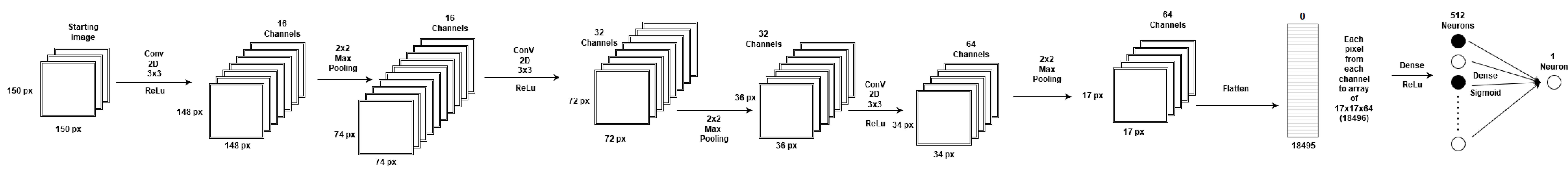


Figure 3.
An illustration of the CNN architecture. The network's input is a 150x150 image with 3 channels representing the RGB color system.