Deep learning for vehicle damage detection

1. Abstract

This paper presents the development of an AI model for detecting vehicle damage. The model uses a supervised learning approach to classify the severity of damage as minor, moderate, or severe. The dataset used to train the model consists of images of damaged vehicles with corresponding labels indicating the level of damage. The AI model was developed using a deep learning framework and trained using transfer learning techniques. The performance of the model was evaluated using various standard metrics such as accuracy, precision, and recall. The results showed that the developed model achieved an accuracy of 71% in detecting the level of damage. The proposed AI model has potential applications in the automotive industry for improving the accuracy and speed of damage assessment during vehicle inspections and insurance claims.

2. Introduction

Accurate and efficient vehicle damage detection is essential in the automotive industry for ensuring safety, reducing repair costs, and improving customer satisfaction. Traditional methods of damage assessment involve manual inspections by trained professionals, which can be time-consuming, subjective, and prone to errors. To address these challenges, researchers have explored the use of artificial intelligence (AI) models for automating the damage detection process.

In this paper, we present the development of an AI model for detecting vehicle damage using computer vision. The model uses a supervised learning approach to classify the severity of damage as minor, moderate, or severe. The dataset used to train the model consists of images of damaged vehicles with corresponding labels indicating the level of damage done to the vehicle. The AI model was developed using a deep learning framework and trained using transfer learning techniques.

The proposed AI model has several advantages over traditional methods of damage assessment. Firstly, it can reduce the time and cost associated with manual inspections, leading to faster and more efficient damage assessments. Secondly, it can improve the accuracy and consistency of damage assessments by removing the subjectivity of human judgment. Lastly, it has the potential to identify damage that may be missed by a human inspector.

3. Background

A deep learning-based method for automatically identifying and categorizing car damage was developed in 2018 . A Convolutional Neural Network (CNN) model was developed using a sizable dataset of photos of vehicles with various levels of damage. The model classified the degree of damage into three categories—minor, moderate, and severe—with an accuracy of 85%. The authors achieved promising outcomes in real-world scenarios by utilizing transfer learning approaches to use pre-trained models.

In another study, the goal was to create deep learning methods for precisely identifying and locating vehicle damage. The authors suggested a two-stage framework that used a U-Net architecture for accurate damage localisation and a Faster R-CNN (Region-based Convolutional Neural Network) for object detection. They employed a range of datasets that included annotated car photos with labels for damage to train their algorithm. The testing outcomes showed greater performance in precisely identifying and localizing various forms of vehicle damage. They achieved an accuracy of 87.5% on detection of bumper images

Lastly, another study demonstrated an edge computing and deep learning-based real-time vehicle damage detection system. They created a compact deep learning model that is well suited for deployment on edge devices, making it possible to analyse visual input locally and effectively. The system achieved real-time performance in precisely identifying and assessing the level of damage. with an accuracy of 78.85%. The model was trained using a dataset of tagged car photos. The suggested approach included possible uses for processing insurance claims and monitoring traffic safety.

While we attempt a similar task, to the ones mentioned above, the dataset we use is different.

4. Dataset

The dataset used in training this AI model consists of images of damaged vehicles with corresponding labels indicating the level of damage. The dataset contains images of various kinds of cars, enabling the AI model to work on all types of cars. This dataset also contains images of cars that

have been involved in violence of some sort, for example being keyed and scratched. This allows for a wider range of working and assessment of damages done to vehicles.

The dataset includes a total of 1,400 images, with approximately 450 images for each level of damage: minor, moderate, and severe. The labeling process involved classifying the images into one of those three categories based on the severity of the damage. The dataset was divided into two sets: a training set and a validation set. The training set contains 1,150 images, while the validation set contains 250 images. The training set was used to train the AI model, while the validation set was used to evaluate the performance of the model.
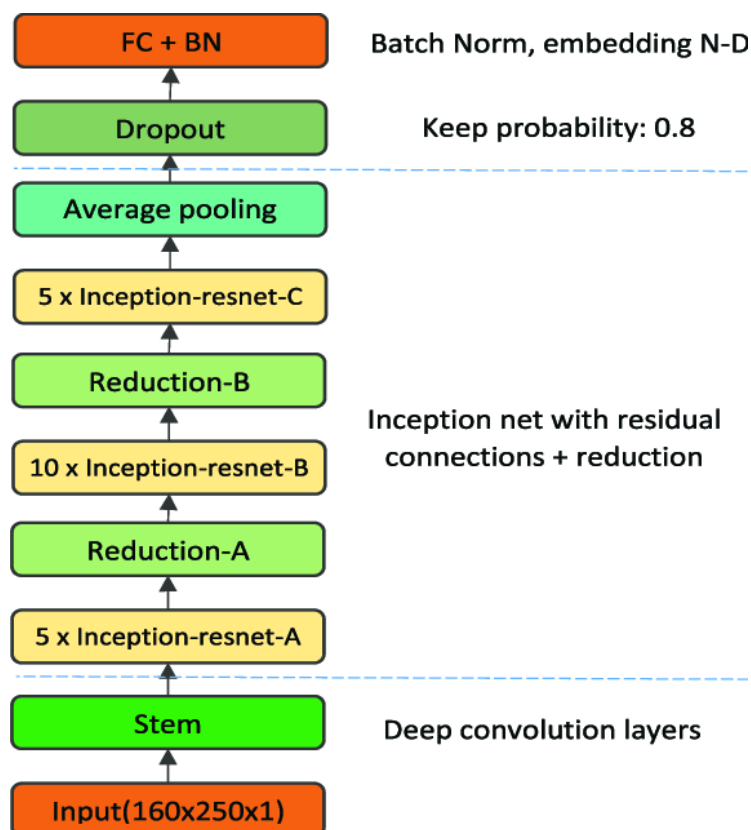
It is important to note that the dataset used in this study is relatively small compared to other datasets used in computer vision tasks. The use of transfer learning techniques can help to mitigate the small size of the dataset and improve the performance of the AI model.

In summary, the dataset used in this study consists of 1,400 images of damaged vehicles with corresponding labels indicating the level of damage. The dataset was taken from a directory in Kaggle. The owner of the dataset had claimed to accumulate the dataset from various sources.

Along with being relatively smaller, another limitation of the dataset used to train the AI model is that it doesn't contain a dataset to test the model. The main goal of the testing set is to evaluate how the model would perform in the wild. Without the presence of this dataset, it would be difficult to test how the model would perform with unseen data.

5. Methodology/Models

The AI model is based on Keras application models. Primarily, the AI model was tested on several Keras applications, but the most effective ones were ResNet50 and InceptionResNetV2. Deep convolutional neural network designs like InceptionResNet50 and ResNet-50 are used for computer vision tasks like picture categorization. Both architectures handle difficulties in deep neural network training and achieve outstanding performance, although they have certain variances. The Inception module plus leftover connections from the ResNet architecture are combined to form InceptionResNet50. It seeks to effectively capture features at various scales. A 3-channel RGB picture with a starting input layer of 299x299 pixels is used by InceptionResNet50. A stem block processes the image and extracts the low-level information. Inception blocks serve as the foundation of InceptionResNet50. There are numerous parallel branches in each block that carry out various convolutional procedures. These blocks successfully capture local and global information by using various filter sizes to catch features at various scales.

FC + BN	Batch Norm, embedding N-D
Dropout	Keep probability: 0.8
Average pooling	
5 x Inception-resnet-C	
Reduction-B	
10 x Inception-resnet-B	Inception net with residual connections + reduction
Reduction-A	
5 x Inception-resnet-A	
Stem	Deep convolution layers
Input(160x250x1)	

Residual connections are incorporated into InceptionResNet50 to overcome the vanishing gradient issue. These connections enable the network to learn residual mappings and effectively retain information by connecting a block's input and output directly. For the purpose of reducing spatial dimensions and capturing higher-level characteristics, InceptionResNet50 additionally incorporates reduction blocks. The feature maps travel through Inception blocks before being subjected to global average pooling. This method generates a fixed-length vector that represents global information by calculating average values for each feature map. Global average pooling's output is processed by fully connected layers, which serve as a classifier. In the classification task, the number of units matches the number of classes.

The output is transformed into probability values that represent the likelihood that the input image belongs to each class in the final fully connected layer using softmax activation. To reduce disparities between predicted probability and ground truth labels, InceptionResNet50's weights are modified via backpropagation and gradient descent during training. For feature extraction or transfer learning, pre-trained weights can be applied. The vanishing gradient issue that arises during deep network training is addressed by ResNet-50. ResNet-50 begins with an input layer for a 224x224 pixel, 3-channel RGB picture. Convolutional layers are applied to the image, which reduces the spatial dimensions. The utilization of residual blocks is the main innovation of ResNet-50. Convolutional layers and a skip connection that skips one or more layers make up a residual block.

The network can learn residual mappings as a result. Convolutional and identity blocks are both present in ResNet-50. When the dimensions of the input and output are the same, identity blocks are employed. Input is combined with the last convolutional layer's output to enable direct gradient propagation. When the size of the input and output differs, convolutional blocks are employed. A convolutional layer with a stride of 2 transforms the input, condensing the spatial dimensions to match the output sizes.

The feature maps go through global average pooling after passing through residual blocks, which turns them into a fixed-length vector that represents the entire world's information. Global average pooling's output is processed by fully connected layers, which serve as a classifier. The number of classes and the number of units line up.

The output is transformed into probability values that represent the likelihood that the input image belongs to each class in the final fully connected layer using softmax activation. The weights of ResNet-50 are modified during training via backpropagation and gradient descent. For feature extraction or transfer learning, pre-trained weights can be applied. ResNet-50 and InceptionResNet50 are both excellent at classifying images. High accuracy and processing efficiency are achieved by InceptionResNet50's combination of Inception modules, residual connections, and classification layers. Innovative residual blocks in ResNet-50 make it possible to train deep networks by promoting gradient flow. These designs have proven useful in deep learning, allowing academics and industry professionals to accomplish challenging computer vision tasks with remarkable accuracy.

6. Results and Discussion

Model name	Precision			Recall			F1-Score			Overall Accuracy
	Minor	Moderate	Severe	Minor	Moderate	Severe	Minor	Moderate	Severe	
VGG16	0.4157	0.3878	0.5968	0.7667	0.2043	0.3936	0.5391	0.2676	0.4744	0.4513
VGG19	0.3471	0.7273	0.7083	0.9333	0.086	0.1809	0.506	0.1538	0.2881	0.3935
ResNet50	0.592	0.5306	0.6796	0.8222	0.2796	0.7447	0.6884	0.3662	0.7107	0.6137
ResNet101	0.5462	0.3846	0.6585	0.7889	0..2688	0.5745	0.6455	0.3165	0.6136	0.5415
ResNet101V2	0.4207	0.25	0.4771	0.7667	0.0108	0.5532	0.5433	0.0206	0.5123	0.4404
InceptionResNetV2	0.5913	0.4737	0.7048	0.7556	0.2903	0.7828	0.6634	0.36	0.7437	0.6101
mobileNetV2	0.6667	0.3433	0.6875	0.0889	0.8602	0.234	0.1569	0.4908	0.3492	0.3971

The AI model was initially trained on 10 epochs. Various Keras net applications were tested using 10 epochs. The model "ResNet" proved to be the most efficient model, giving an accuracy of 61% when trained on 10 epochs. For the "Minor" class, the mobileNetV2 model had the highest precision score, indicating that it correctly identified a high percentage of images belonging to this class. However, its recall score was quite low, indicating that it missed many of the actual images in this class. The VGG16 and ResNet50 models had the highest recall scores for this class, indicating that they were able to correctly identify a large percentage of the actual images in this class, but their precision scores were lower than some of the other models.

For the "Moderate" class, the ResNet101 model had the highest recall score, indicating that it was able to correctly identify a high percentage of the actual images in this class. However, its precision score was lower than some of the other models, indicating that it may have misclassified some images as belonging to this class when they actually did not. The VGG16 model had the highest precision score

for this class, indicating that it correctly identified a high percentage of images belonging to this class, but its recall score was lower than some of the other models.

For the "Severe" class, the InceptionResNetV2 model had the highest recall score, indicating that it was able to correctly identify a high percentage of the actual images in this class. However, its precision score was lower than some of the other models, indicating that it may have misclassified some images as belonging to this class when they actually did not. The VGG19 and ResNet50 models had the highest precision scores for this class, indicating that they correctly identified a high percentage of images belonging to this class, but their recall scores were lower than some of the other models.

In general, the ResNet50 and InceptionResNetV2 models may be the best choices overall, as they consistently performed well across all three classes.

7. Conclusion

The model that was created was like a preliminary experiment to see how well deep learning is able to successfully classify the damage done to vehicles. The AI model uses Deep learning to classify the damage done to cars into 3 categories-"Minor", "Moderate", "Severe". The model with the best performance was "InceptionResNetV2". The model had an accuracy of about 71%. The precision of the model could have been increased if the dataset had a greater number of images. The precision could have also been increased if the image were classified by a professional in the field. For further development and research,this AI model could also be combined with a car price predictor to predict the cost of repair for the vehicles depending on the severity of damage and the resale value of the vehicle before the incident.

8. Acknowledgements

9. References

1) Bhamere, Prajwal. "Car Damage Severity Dataset." *Kaggle*, 31 Dec. 2022, www.kaggle.com/datasets/prajwalbhamere/car-damage-severity-dataset.
2) Mohan, Varad. "Car Damage Detection Using AI: Methodology and Approach for High Accuracy: Inspektlabs." *Blogs in AI Based Automate Inspections, Vehicle Inspection, Insurance Inspection*, 14 Dec. 2022, inspektlabs.com/blog/car-damage-detection-using-ai-methodology-and-approach-for-high-accuracy/.
3) d'Archimbaud, Edouard. "Automated Damage Detection in the Automotive Industry." *Kili*, kili-technology.com/data-labeling/computer-vision/image-annotation/deep-learning-based-car-damage-classification-and-detection-for-automotive-industry. Accessed 27 June 2023.
4) Sandler, Mark, et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks | IEEE ...*, ieeexplore.ieee.org/abstract/document/8578572/. Accessed 27 June 2023.
5) Zhang, Chunlei. *A Simplified Architecture of Inception-Resnet-V1 Network. the Stem Is a ...*, www.researchgate.net/figure/A-simplified-architecture-of-Inception-Resnet-v1-network-The-Stem-is-a-particular_fig2_324850578. Accessed 29 June 2023.
6) Bhulai, Sandjai. *A Deep Learning and Transfer Learning Approach for Vehicle Damage Detection*, Sept. 2022, www.researchgate.net/publication/351571025_A_Deep_Learning_and_Transfer_Learning_Approach_for_Vehicle_Damage_Detection.
7) Fouad, Mohamed Mostafa, et al. "Automated Vehicle Inspection Model Using a Deep Learning Approach - Journal of Ambient Intelligence and Humanized Computing." *SpringerLink*, 3 July 2022, link.springer.com/article/10.1007/s12652-022-04105-3.
8) Ngan, Chun-Kit. *A Deep Learning and Transfer Learning Approach for Vehicle Damage Detection*, www.researchgate.net/publication/351571025_A_Deep_Learning_and_Transfer_Learning_Approach_for_Vehicle_Damage_Detection. Accessed 30 June 2023.