A Deep Learning and Transfer Learning Approach for Vehicle Damage Detection

Lin Li
Seattle University, Seattle, WA 98122
lil@seattleu.edu

Koshin Ono
Seattle University, Seattle, WA 98122
onok1@seattleu.edu

Chun-Kit Ngan
Worcester Polytechnic Institute, Worcester, MA 01609
cngan@wpi.edu

Abstract

According to the U.S. Department of Transportation, there is an average of six million motor vehicle crashes every year in the United States. For insurance companies, it is very time-consuming and expensive to process claims for detecting and classifying vehicle damages; thus, deep learning techniques have been used to automate this process to reduce the time and the cost. In this paper, Mask R-CNN is used for image segmentation to identify and crop vehicles from images. Then a convolutional neural network (CNN) model is built to classify whether or not the vehicles have damages. In addition, transfer learning is utilized in both image segmentation and classification phases to help build the models for vehicle detection and damage detection, using the pre-trained weights from Microsoft COCO dataset and ImageNet, respectively. 864 images of damaged vehicles collected from public websites, such as Google Images, are used in this research. The experiment on the detection of bumper damages has achieved 87.5% accuracy.

1 Introduction

On average, there are six million car accidents every year in the U.S., according to the U.S. Department of Transportation. As part of the vehicle insurance claim process, the insurance company needs to detect and classify the vehicle damages. To make the damage detection and classification faster and more cost-efficient, it is critical to develop approaches for automated detection and classification of vehicle damages from images in the claims.

With the development of deep learning, in the areas of image classification such as detection and classification of diseases in medical images (Anwar et al. 2018; Li et al. 2014; Lam et al. 2018), deep learning models, such as convolutional neural network (CNN), have been more popular and more often used than traditional machine learning models such as support vector machine (SVM). One major reason is that in the big data era, the performance of deep learning models can be significantly improved by using a large number of images, whereas the performance of traditional machine learning models does not usually show a big improvement by significantly increasing the number of images. However, there are some challenges in the application of deep learning for image classification. First, to achieve high performance, deep learning models need to be trained on a large number of images. However, it is difficult to get a large quantity of images in some areas. For example, for the vehicle damage detection problem in this paper, it is hard to obtain thousands of damaged vehicle images that are publicly available. Second, with a big size of training data, deep learning requires high computational power to train a model. The model training process is time-consuming. Third, deep learning models, such as CNN, have many parameters and hyperparameters, which require a lot of efforts to tune.

To overcome the challenges of training deep learning models from scratch, transfer learning has been recently proposed and developed. Transfer learning is one of the popular machine learning techniques that focuses on developing and training a model, as a starting point, based upon the knowledge gained in general domains and then reuse the same model trained on a specific domain to solve a specific task. Researchers have explored transfer learning and its benefits by trying various ways to fine-tune the pre-trained models (Tajbakhsh et al. 2016; Das, and Kumar 2018). In (Tajbakhsh et al. 2016), the authors indicate that transfer learning produces accuracy that is better than the accuracy of training CNN from scratch. With the help of transfer learning, deep learning can be used on a limited amount of training data. Transfer learning has shown a better performance in classification problems to avoid the overfitting issue when the training dataset is limited (Oquab et al. 2014), because it takes advantage of the pre-trained models that have already been trained on large datasets such as ImageNet (Deng et al. 2009) and Microsoft COCO (Lin, et al. 2014) and transfers the knowledge learned from the pre-trained models to the target tasks that have limited datasets. Since the weights are pre-trained, the learning process would require less computational power and less time, while the tuning of parameters and hyperparameters would take less work.

Although deep learning and transfer learning have been used by researchers for image classification, few researches have been done using these approaches in vehicle damage detection and classification. Some existing researches do not

use machine learning in the detection of car damages. For example, in (Jayawardena, 2013), 3D CAD models of undamaged vehicles are proposed to obtain ground truth information to help detect vehicles with mild damage in the photograph. In principle, image edges which are not present in the 3D CAD model projection can be considered as vehicle damage. The authors in (Gontscharov et al. 2014) have proposed an approach to use adaptive sensor data processing for minor damage identification. A sensor network is integrated into the vehicle body, and multi sensor-data fusion of the signals from these sensors is used for the subsequent reasoning of damage detection.

There are some existing APIs that can be used for image classifications using machine learning, such as Google's Cloud Vision API and Microsoft's Azure Custom Vision API. These APIs can utilize the computation power on the cloud platforms; hence the training only takes a few minutes. However, the performance of these APIs on the vehicle damage detection problem is not ideal. Besides adjusting probability thresholds, users do not have ability to fine-tune the models for specific problems to increase the performance. Hence the customized models that are for vehicle damage detection and classification are needed.

In (Patil et al. 2017), for car damage classification, due to the lack of large amount of vehicle damage datasets and in order to prevent the model from suffering the overfitting issue, in addition to applying data augmentation, transfer learning is used on six CNNs, including Cars, Inception, AlexNet, VGG16, VGG19, and ResNet, pre-trained on ImageNet and then further trained on the car damage dataset collected from the web. An ensemble classifier is then built on top of the set of pre-trained classifiers. Using the transfer learning, the car damage classification accuracy is only about 75% on average among all the pre-trained CNN models. This unsatisfactory performance is due to the various damaged car locations on images. Hence, we need an approach that can identify and crop the damaged car on image before performing the classification.

In (Li at al. 2018), the authors have presented an approach to generate robust deep features by locating the damages in the images. YOLO is modified to train and identify damage region in the vehicles. In (Dhieb et al. 2019), the Mask R-CNN model is utilized to locate and visualize the damage on vehicle images, where the model weights are initialized from a pre-trained model based on Microsoft COCO dataset. Then the Inception-ResNet pre-trained model on ImageNet is employed and followed by a CNN model to classify the damages. The authors in (Malik et al. 2020) conduct vehicle damage detection using YOLOv3 (Redmon, and Farhadi 2018) and then perform damage classification using CNN models pre-trained on ImageNet. However, all these works directly feed the images that contain the vehicles to train models for damage detection and classification, without detecting and cropping the vehicles and removing irrelevant items in the images. Thus, the detection performance is not satisfactory due to lack of considering a pre-trained segmentation model in the framework. When users take pictures of damaged vehicles to submit a claim to insurance companies, it is likely that the images contain some irrelevant items such as road, trees, and buildings. To reduce the noises in the images, these irrelevant items should be removed before the images are fed into deep learning models.

Taking the above problems into consideration, we propose to develop a workflow that contains a pre-trained Mask R-CNN to detect and segment the damaged car and then pass the segmented images to the pre-trained CNN models for the damage classification. Our work starts from the image segmentation to locate vehicles from images. Mask R-CNN is used by applying weights pre-trained on Microsoft COCO dataset to identify and crop vehicles from the original images. Then CNN models pre-trained on ImageNet are retrained to detect whether or not there is a damage on the vehicles. The intensive experiments have been done for the detection of bumper damages with transfer learning, using eight different CNN architectures such as VGG16, VGG19 and ResNet50. 864 images of damaged vehicles collected from the public websites, such as Google Images, are used in this research. With data cleaning and data augmentation, it turns out that VGG16 has achieved the best performance 87.5%.

The paper is organized as follows. We describe the dataset and the approach used in this work, including Mask R-CNN with transfer learning for image segmentation to detect vehicles and CNN with transfer learning for the detection of damages on vehicles in Section 2. Section 3 presents the experimental results of detection of vehicles, detection of damages on vehicles, and the results of damage detection without identifying and cropping vehicles. In Section 4, we draw conclusions and discuss the future work.

2 Data and Approach

2.1 Dataset

Since there is no publicly available dataset of damaged vehicle images, we collect the images by web scraping from websites such as Google Images. The web scraping process is automated by developing an image scraping script using Python and Beautiful Soup library. The damaged vehicle images include various types of vehicles such as sedan, SUV, minivan, and more. The image resolution ranges from 224 × 224 pixels to 4000 × 6000 pixels. The damages can be divided into different types, including bumper damage, glass damage, severe frame damage, and so on. In total, 864 images are collected and manually labeled (e.g., no damage, bumper damage, glass damage, etc.).

2.2 Detection of Vehicles

Fig. 1 describes the workflow of our approach. The detection of vehicles by cropping cars in the image is discussed in this section, while section 2.3 discusses the detection of damages from the cropped images.

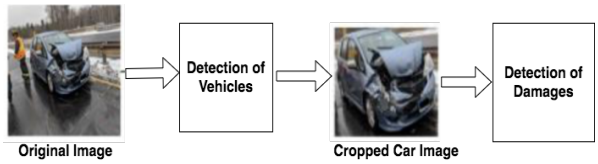

Figure 1 Workflow of our approach

Many images in the dataset have noises in the background. These noises include people standing next to vehicle, buildings, road signs, and more. It is important to remove those noises and then detect and crop the vehicles before feeding the cropped images into the learning CNN models. To achieve this purpose, a Mask R-CNN model is trained for image segmentation, that is, to detect the vehicles in the images. Fig. 2 shows the data pre-processing for the Mask R-CNN model before starting the training process.

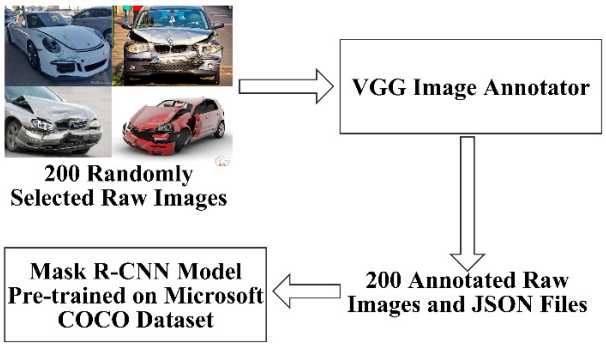

Figure 2 Data Pre-processing for the Mask R-CNN model

We use Python 3, Keras, and TensorFlow in the implementation. 200 images are randomly selected from the 864 images as the training data. We then use VGG Image Annotator to create polygon boundaries and manually add the masks around vehicles in each image in the training data. VGG Image Annotator provides a simple user interface for the users to create boundaries and outputs coordinates of pixels in the vehicles in all annotated images to a JSON file. Both the JSON file that has the vehicle coordinates and the original images in the training data are fed to the Mask R-CNN model for training. To avoid the overfitting issue, due to the limited training dataset, transfer learning is applied by using the pre-trained weights on Microsoft COCO dataset, which contains over 300,000 images and 1.5 million object instances (Lin, et al. 2014). After the training process, we use the remaining portion of the dataset, i.e., 664 images, to evaluate the model performance that is discussed in section 3.1.

2.3 Detection of Damages

After detecting the vehicles from the original images, the vehicles are cropped. The detection of damages is then conducted on the cropped vehicle images but not the original images. Fig. 3 shows the data pre-processing for the CNN models before the training, validation, and testing processes.

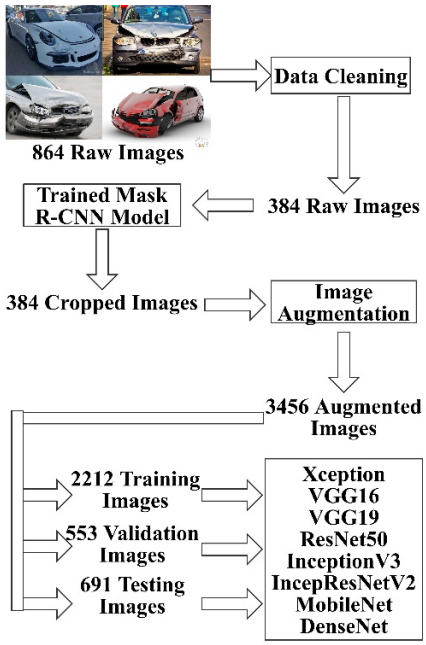

Figure 3 Data Pre-processing for the CNN models

First of all, we perform data cleaning. We notice that the variety of angles from which the damage images are taken may confuse the CNN during classification. For instance, bumper damages taken from front-view or from up to 45-degree angled view may look similar. However, the same bumper damages taken from 90-degree angle or from side-view would be completely different from neural network's perspective. Therefore, we group images by angles and remove any images that are over 45-degree angle from the data set. After data cleaning, the number of raw images is 384, which are then passed to the trained Mask R-CNN model to detect and crop the vehicles.

To deal with the small dataset problem, image augmentation is utilized to increase the number of images. The techniques that we use for data augmentation include Gaussian Noise, Gaussian Blur, Flip, Contrast, Hue, Add (i.e. add random values to pixel intensities), Multiply (i.e. multiply pixel intensities by some values), and Sharp. With all these data augmentation techniques, the number of images is increased to 3,456. 20% of the data is used for testing. Out of the other 80% of the data, 80% is used for training, and 20% is used for validation. That is to say, 2212, 553, and 691 images are used for training, validation, and testing, respectively.

A CNN model is used for the detection of vehicle damages. Images are broken into smaller pieces - feature filters. In convolutional layer, each filter would slide across the input image and create a stack of filtered images. Normalization through the activation function, i.e. Rectified

Linear Units (ReLUs), is utilized to step through each filtered image and convert negative intensity values of pixels to zero. A pooling layer is used to reduce the number of parameters and computation in the CNN model. The final layer is a fully connected layer using the Softmax activation function. In addition, the backpropagation is used to determine the weights.

Instead of training a CNN model from scratch, we also use transfer learning in the detection of damages. The CNN model pre-trained on ImageNet dataset is used. ImageNet contains more than 14 million images which belong to more than 20,000 classes (Deng et al. 2009). Due to the fact that the vehicle damages are not part of the data in ImageNet, we re-train the model by unfreezing more layers, such as five and ten layers. Instead of just using the ResNet architecture, we also employ seven more pre-trained CNN architectures, shown in Fig. 3, such as Xception, VGG16 and VGG 19. It turns out that VGG 16 can achieve the best accuracy, as shown in the experimental results in section 3.2.

3 Experimental Results

3.1 Results of Detection of Vehicles

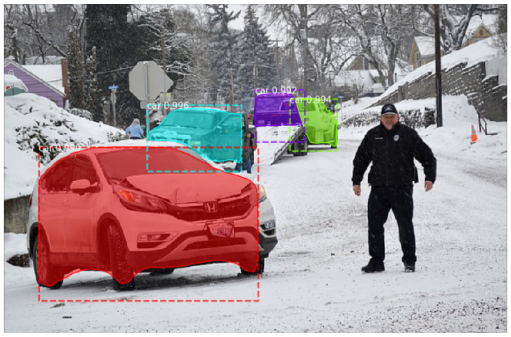

Figure 4 Vehicle detection result of a sample image

The Mask R-CNN model trained with transfer learning has achieved 90% accuracy in detecting vehicles in images. Fig. 4 shows one of the sample results of vehicle detection on 664 images when applying the Mask R-CNN model. In Fig. 4, even though there are noises, such as people, trees and road signs, the model only detects vehicles in the image. All vehicles are correctly captured, except that the back of the towing truck is also captured as a separate vehicle. Our experiments also show that the Mask R-CNN model can detect vehicles which body is not completely taken in the images. For example, Fig. 5 shows that the model is able to detect vehicles in the images, even though the SUV on the left-hand side of image only shows its rear. The model can also capture some of the vehicles in the parking lot in the background of the image. Since the bounding boxes (i.e. the colorful boxes with the dotted outlines in Fig. 4 and Fig. 5) always capture the majority part of vehicles but not the edges of vehicles, we have enlarged 5% of the size of the bounding boxes when cropping vehicles to include edges of the vehicles. After vehicles are detected, they are automatically cropped from the picture. Fig. 6 shows one of the original sample images and its auto detected and cropped image.

Figure 5 Vehicle detection result of another sample image

Figure 6 A sample original image is shown on the left; its auto-detected and cropped image is on the right.

3.2 Results of Damage Detection

Table 1 The accuracy of 8 different CNN architectures without transfer learning (TL) and with transfer learning by unfreezing the last 1 layer, 5 layers, and 10 layers

	Without TL	**TL (1 Layer Unfrozen)**	**TL (5 Layers Unfrozen)**	**TL (10 Layers Unfrozen)**
Xception	63.08%	74.21%	78.12%	63.28%
VGG16	66.60%	79.16%	**87.50%**	78.64%
VGG19	62.50%	76.56%	84.89%	67.18%
ResNet50	59.14%	71.09%	77.08%	78.12%
InceptionV3	60.93%	77.34%	84.14%	73.17%
IncepResNetV2	57.81%	69.53%	77.08%	66.66%
MobileNet	53.64%	72.65%	76.82%	71.35%
DenseNet	48.17%	74.73%	73.43%	64.84%

In our work, we conduct intensive experiments using eight different CNN architectures on 691 images to detect one type of the vehicle damages – bumper damages. We also train each CNN architecture by unfreezing different number of layers. For example, out of all the possible combinations, using VGG 16 by unfreezing last five layers of its architecture has achieved the highest accuracy 87.5%, as shown in Table

1. Fig. 7 presents the training accuracy and testing accuracy of VGG16 by unfreezing the last 5 layers of the pre-trained model.

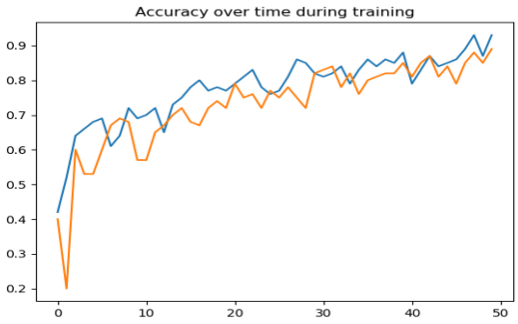

Figure 7 The accuracy of VGG16 by unfreezing the last 5 layers. The blue curve is the training accuracy, and the orange curve represents the testing accuracy.

3.3 Results of Damage Detection without Cropping Vehicles

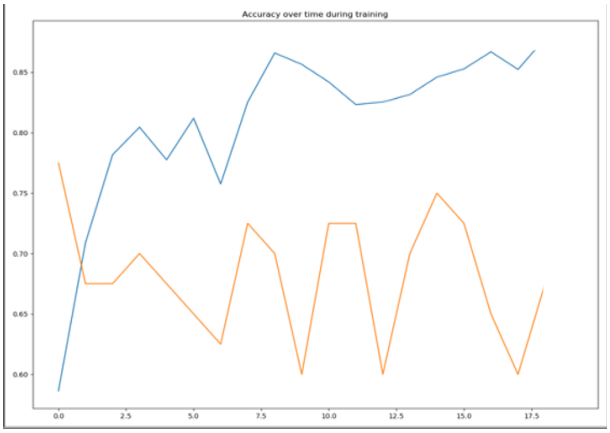

Figure 8 The accuracy of CNN (without transfer learning) for bumper detection with no detection and cropping of vehicles. The blue curve is the training accuracy, and the orange curve represents the testing accuracy.

To see how removing noises from images and cropping vehicles can improve performance of damage detection, we also conduct the experiments using a CNN model with the ResNet50 architecture with the original images as the input. The model is implemented using Keras with a TensorFlow backend. In order to speed up the learning process, we use the cross entropy as the cost function and use Adam as the optimizer. The original images are directly feed into the model for the damage detection, i.e., the bumper detection, without going through the step of detection of vehicles and the cropping of vehicles. Fig. 8 shows the training accuracy and testing accuracy of the model without using transfer learning. The training accuracy keeps increasing and approaching 90%, but the testing accuracy remains around an average of 65%, indicating an overfitting issue.

To overcome the overfitting issue, we also use transfer learning to train a model with its pre-trained weights. The ImageNet pre-trained model is used, where we freeze every layer but not the last layer. The last layer is the dense layer that determines whether or not an image contains a bumper. Again, the image is an original image without going through the step of detection of vehicles and the cropping of vehicles. With transfer learning, the accuracy is increased to 69.8%, which is lower than the performance of using the same model (with transfer learning by unfreezing the last 1 layer) but feeding into the vehicle images cropped from the original images (as shown in Table 1).

4 Discussion and Conclusions

In this paper, we present an approach that utilizes transfer learning to build a pre-trained Mask R-CNN model for the detection of vehicles from the original images, crops the vehicles from the images, and feeds the cropped images into a pre-trained CNN model for detecting the vehicle damages. From the performance comparison of bumper damage detection between the experiments with and without detecting and cropping vehicles, it can be seen that the vehicle detection step is important to remove the noises such as building, road signs, and so on. Our proposed approach can automatically detect and crop the vehicles, avoiding the overhead of manual detection and cropping.

For vehicle detection, the reason we choose Mask R-CNN for instance segmentation over a model for object detection only is that Mask R-CNN provides better accuracy in detecting objects than object detection models like Fast-RCNN with the implementation of ROI align in ROI pooling. In Fast R-CNN, quantized ROI pooling is used, which may lead to data loss, whereas in Mask R-CNN, non-quantized method – ROI Align is used to prevent data loss and leads to more accurate result (Huang et al. 2017; He et al. 2017).

As shown in Table 1, transfer learning can achieve a better performance than training a model from scratch, when the size of the data set is limited. In the experiment for damage detection without cropping vehicles, transfer learning also has achieved a better performance than training a CNN model from scratch. Also, Table 1 shows that with most of the eight CNN architectures in the experiment, transfer learning with the last 5 layers unfrozen can all achieve a better performance than that with the last 1 layer unfrozen and that with the last 10 layers unfrozen. Particularly, VGG16 with the last 5 layers unfrozen has achieved the best accuracy. This is possibly due to the fact that the vehicle damage is not part of ImageNet, so more layers need to be unfrozen to re-train and re-tune the model. In transfer learning, if the classification tasks require the model to look at unfamiliar images, it is common to re-train multiple layers. However, if too many layers are unfrozen, the pre-trained model loses its benefits (i.e., early layers represent simple shapes in the images and can be re-used to solve different image

classification problems) and the performance will decrease. In addition, a possible reason that VGG16 performs better than VGG19 is that there is an unstable, vanishing gradient happening within the neural net. As we increase the number of layers in a network, the layer towards the input will be affected less by the error calculation occurring at the output as going back through the network. As it gets closer to input, it might not change any weights because there are too many layers.

As shown in Fig. 8, the overfitting occurs in the detection of damages, possibly due to the limited number of datasets. But with transfer learning, the overfitting issue is resolved. Fig. 7 also shows that there is no overfitting when transfer learning is used, where we observe a healthy increase in both training accuracy and testing accuracy.

To our best knowledge, there is not yet a free tool to automatically detect vehicle damage. Hence, we adopt the APIs that are used for image classification for the purpose of car damage detection. The two major APIs are Google's Cloud Vision API and Microsoft's Azure Custom Vision (ACV) API, which provide similar capabilities. We use ACV API in our experiment to see its performance in detecting vehicle damage. Even though ACV provides a user-friendly web interface (UWI), it requires each image to be labeled manually within its UWI or users need to use its API to upload image with labels. We developed a Python script to feed all the labeled 864 images to the API. The result ends up being only 61.2% accuracy in detecting vehicle damages. The benefit of using the tool is that it utilizes the cloud computation power from the company, so the training process is fast. However, users do not have the ability to fine-tune the model except for adjusting the probability threshold.

In the future, there are four major tasks. First, we will improve the models that will be able to detect vehicle damages in the images that are taken from the over 45-degree angle. Second, we will add a step between vehicle detection and damage detection to detect the target areas (e.g. for bumper damage detection, the bumper will be detected and cropped, and then the bumper image will be fed into the CNN model for damage detection). To achieve this purpose, the current Mask R-CNN model in our work for detecting vehicles will need to be re-trained and re-tuned so that the new Mask R-CNN model will be able to detect target areas such as bumpers. Third, we will create the models for detecting each type of vehicle damages such as glass damage, engine damage, and so on. Finally, we will build an ensemble model to determine the severity of the vehicle damages based on all types of damages detected in the vehicles.

References

Anwar, Syed Muhammad, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. "Medical image analysis using convolutional neural networks: a review." Journal of medical systems 42, no. 11 (2018): 226.

Das, S. D., & Kumar, A. (2018, October 18). "Bird Species Classication using Transfer Learning with Multistage Training". In Cornell University Computer Vision and Pattern Recognition. Retrieved November 11, 2018, from https://arxiv.org/abs/1810.04250v2

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In 2009 IEEE conference on computer vision and pattern recognition, pp. 248-255. Ieee, 2009.

Dhieb, Najmeddine, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud. "A very deep transfer learning model for vehicle damage detection and localization." In 2019 31st International Conference on Microelectronics (ICM), pp. 158-161. IEEE, 2019.

Gontscharov, Sergei, Hauke Baumgärtel, Andre Kneifel, and Karl-Ludwig Krieger. "Algorithm development for minor damage identification in vehicle bodies using adaptive sensor data processing." Procedia Technology 15 (2014): 586-594.

K. He, G. Gkioxari, P. Doll´ar, and R. Girshick. Mask R-CNN. In International Conference on Computer Vision (ICCV), 2017.1,2,3,4,6,8

J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade- offs for modern convolutional object detectors. In CVPR,2017.2,3,4,6,7.

Jayawardena, Srimal. "Image based automatic vehicle damage detection." (2013).

Lam, Carson, Darvin Yi, Margaret Guo, and Tony Lindsey. "Automated detection of diabetic retinopathy using deep learning." AMIA summits on translational science proceedings 2018 (2018): 147.

Li, Pei, Bingyu Shen, and Weishan Dong. "An anti-fraud system for car insurance claim based on visual evidence." arXiv preprint arXiv:1804.11207 (2018).

Li, Qing, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. "Medical image classification with convolutional neural network." In 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 844-848. IEEE, 2014.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.

Hashmat Shadab Malik, Mahavir Dwivedi, S. N. Omakar, Satya Ranjan Samal, Aditya Rathi, Edgar Bosco Monis, Bharat Khanna, Ayush Tiwari, "Deep Learning Based Car Damage Classification and Detection", EasyChair Preprint no. 3008, 2020.

Oquab, Maxime, Leon Bottou, Ivan Laptev, and Josef Sivic. "Learning and transferring mid-level image representations using convolutional neural networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1717-1724. 2014.

Patil, Kalpesh, Mandar Kulkarni, Anand Sriraman, and Shirish Karande. "Deep learning based car damage classification." In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 50-54. IEEE, 2017.

Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767, 2018.

N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1299–1312, 2016.