EasyChair Preprint

№ 3008

Deep Learning Based Car Damage Classification and Detection

Hashmat Shadab Malik, Mahavir Dwivedi, S. N. Omakar, Satya Ranjan Samal, Aditya Rathi, Edgar Bosco Monis, Bharat Khanna and Ayush Tiwari

March 22, 2020

Deep Learning Based Car Damage Classification and Detection

Abstract. In this paper, we designed and implemented a car damage classification/detection pipeline, which can be used by insurance companies to automate the process of vehicle insurance claims. The recent advances in computer vision largely due to the adoption of fast, scalable and end to end trainable Convolutional Neural Networks(CNN's) makes it technically feasible to recognize vehicle damages using deep convolutional networks. We manually collected and annotated images from various online sources using web crawler containing different types of vehicle damages. Due to the relatively small size of our dataset, we used models pre-trained on a large and diverse dataset to avoid overfitting and learn more general features. Using CNN models pretrained on ImageNet dataset and applying preprocessing techniques to improve the performance of our system, we were able to achieve accuracy of 96.39%, significantly better than results achieved in the past on a similar test-set. Furthermore to detect the region of damage we used state-of-the-art YOLO object detector and achieving a maximum map score of 77.78 % on the held-out test set, demonstrating that the model was able to successfully recognise different vehicle damages. In addition to this , we also propose a pipeline for a more robust identification of the damage in vehicles by combining the tasks of classification and detection. Overall these results pave the way for further research in this problem domain and we believe collection of a more diverse dataset would be sufficient to implement an automated vehicle damage identification system in the near future

Keywords: Car damage classification/detection, Pre-trained CNN models, YOLO object detector.

1. Introduction

Nowadays car insurance companies deal with frequent insurance claims. Using manually validation on large scale of claims, companies are not able to meet the speedy requirement for insurance claim processing resulting in Claim Leakage[1]. Claim Leakage is simply defined as lost money through claims management inefficiencies that ultimately result from failures in existing processes (manual and automated). Claim amount primarily depends on the damage type and damaged part of the car, so we need an automated system for the car insurance claim processing as used by some startups[2] which can efficiently classify and detect damage and helps to minimize the claim leakage. This motivates us to explore several means of automating this process , of which computer vision could be an alternative.

The revolution because of Krizhevsky and others leaves the field of computer vision open for further research. Convolutional Neural Networks have shown superior accuracy on image classification tasks, as shown on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and are dominating most of the problems faced in computer vision. CNNs have also been applied on the problem of object detection, such as R-CNN [3] that are significantly better than the conventional feature-based detectors in terms of accuracy.In RCNN selective search [4] is used to propose regions which are more likely to have an object. These proposed regions are then resized to pass through a CNN with SVM classifiers to assign the type of object in the region. However as for each proposal generated by selective search algorithm, we have to perform a forward pass through the CNN, it is quite slow. Further improvements in the architecture came in Fast R-CNN [5] which instead of passing proposed region through the CNNs processes the whole image to extract a feature map. Feature map is used for object proposals with the help of a region of interest (ROI) pooling layer. In addition, faster R-CNN [6] improves the architecture further by proposing Region Proposal Networks (RPNs) that share convolutional layers with object detection networks.
Architecture's following the approach of [3] for object detection use regions to localize objects within the image and tend to look at the only those regions of image which have a higher chance of containing an object. In contrast to this technique, YOLO framework [7] uses the whole image simultaneously for detection. Furthermore, it's significantly faster than R-CNN family of architectures.Yolov3 [8] is currently the best among different available models in the YOLO family, it is improved by making the network bigger and adding residual networks by adding shortcut connections.
Taking this into consideration, we used CNNs for classification/detection of vehicle damages. For this problem statement we created our own dataset by collecting images from internet and manually annotating them. We focused on seven commonly observed car damage types such as smashed, scratch, bumper dent, door dent, head lamp

broken, glass shatter and tail lamp broken.We started with basic CNN model, pre-training a CNN using autoencoder followed by fine-tuning image-classification models pretrained on the ImageNet dataset. We observed that Transfer learning performs best. To increases the robustness of the system, we propose a architectural pipeline that will first extracts the damaged region of the vehicle from the image and then passes it to the trained classifier network. As described earlier, for detection purposes, We also use YOLOv3 [8], an object detector that uses features learned by a deep convolutional neural network to detect an object.

This paper is organized as follows. In section 2 we briefly review related work in the field of car damage classification/detection . The collected dataset is introduced in section 3. Section 4 deals with transfer learning and the results achieved on various deep learning based models used for classification of car damage. Section 5 deals with techniques used to further improve the accuracy of the CNN based models. In section 6 we implement a technique known as "Class Activation Mapping" to visualize the region used by CNN's to identify a specific class. Section 7 & 8 deal with damage detection and proposed pipeline for inference of results in a more robust manner respectively. Conclusion and references are given in section 9 and 10 respectively.

2. Related Works

Several damage detection approaches have been proposed applied to car body damage detection. Srimal.al[9] propose to use 3D CAD models to handle automatic vehicle damage detection via photograph.To identify damage in a car they use a library of undamaged 3D CAD models of vehicles as ground truth information. In principle, image edges which are not present in the 3D CAD model projection can be considered to be vehicle damage. An Anti-fraud System for Car Insurance Claim Based on Visual Evidence - they proposed an approach to generate robust deep features by locating the damages accurately and used YOLO to locate the damage regions. Gontscharov.al [10] tries to solve vehicle body damage by using multi sensor-data fusion. Keyence Vision[11] proposed an industrial solution for car damage by hail by applying a high-resolution Multi-camera vision system. Cha. et al.[12] adopt image based deep learning to detect crack damages in concrete, the methodology used is - acquiring images with the help of camera, then the preprocessing stage where the acquired images undergo scaling and segmentation, and finally to get the shape of crack, feature extraction is done, while [13] adopted a phase-based optical flow and unscented Kalman filters. A. Mohan and S. Poobal. study and review Crack detection using image processing. Based on the analysis, they conclude that more number of researchers have used the camera type image for the analysis with a better segmentation algorithm [14].

3. Dataset Description

As far as we know there is no publicly available dataset for car damage classification, for instance , since vehicles have very reflective metallic bodies & the photographs are taken in an uncontrolled environment , it is very challenging to apply standard computer vision techniques in this context, therefore we created our own dataset by collecting images using web crawling as done by [15]. We manually filtered and categorised images into 7 commonly observed damage types as shown below in Table 1. We also collected images belonging to No Damage class. Some sample images of our dataset are shown in Figure 1 where each column from left to right represents a damage types bumper dent, scratches, door dent, glass shatter, head-lamp broken, tail-lamp broken and smashed respectively.

3.1 Data Augmentation

Training our model with small dataset may cause overfitting, so to overcome this problem and improve our model we used data augmentation to increase dataset size. We enlarged the dataset to approx. 4X by applying rotation-range of 20 degrees, shear-range of 0.2, zoom-range of 0.2 and horizontal-flip.For classification, we randomly split the data into 80%-20% where 80% was used for training and 20% was used for testing.

Figure 1: Sample images of different damage-types

Table 1. Dataset Description

Classes	Train Size	Aug. Train Size	Test Size
Bumper dent	150	750	30
Scratch	112	560	22
Door dent	146	730	25
Glass shatter	104	520	25
Head-lamp broken	107	535	20
Tail-lamp broken	39	195	11
Smashed	256	1280	30
No damage	947	4735	225

4. Transfer Learning

To avoid the problem of overfitting on small datasets, instead of training the CNN models from scratch we can use Transfer learning which has shown a significant improvement on classification problems when the dataset available is scarce[16] [17]. So it is more common to train a CNN on an already available large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories) and then transfer this knowledge to the target task with the intuition that some knowledge may be common between the source domain and target domain. Learning features from a large and diverse input data distribution has been found to be more effective than using features from a data distribution specific to just the target class. Instead of just pretraining our model on a large car dataset, it is better to learn features on a more diverse dataset.[15]. After training on Imagenet dataset, we retrain the classifier on top of the CNN on our dataset. We also fine-tune all the layers of the CNN while keeping in mind that the earlier layers learn more generic features that are common in all classification tasks[18] .

4.1 Alexnet

AlexNet [19] was designed by Alex Krizhevsky , is one of the deep ConvNets designed to deal with complex scene classification task on Imagenet data.The Network had a very similar architecture to LeNet [20], but was deeper, bigger, and featured Convolutional Layers stacked on top of each other. It contains eight layers, the first five are convolutional layers, some of them followed by max-pooling layers, and the last three are fully connected layers. To reduce overfitting alexnet uses another technique called dropout that was introduced by G.E. Hinton in a paper [21] in 2012.

4.2 VGG19

The best-performing submissions to the ILSVRC2013 (Zeiler & Fergus, 2013; Sermanet et al., 2014) utilised smaller receptive window size and smaller stride of the first convolutional layer. VGG-19[18] address another important aspect of ConvNet architecture design – its depth. They fixed other parameters of the architecture, and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3×3) convolution filters in all layers.As a result, came up with significantly more accurate ConvNet architecture.

4.3 Inception V3

The InceptionV3[22] deep convolutional architecture also known as GoogleNet , was developed by Google.With 42 layers, lower error rate is obtained and make it become the 1st Runner Up for image classification in ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2015. All the architectures prior to Inception, performed convolution on the spatial and channel wise domain together. This model is comprised of a basic unit referred to as an "Inception cell" in which we perform a series of convolutions at different scales and subsequently aggregate the results. By performing the 1x1 convolution, the inception block is doing cross-channel correlations, ignoring the spatial dimensions. This is followed by cross-spatial and cross-channel correlations via the 3x3 and 5x5 filters. All of these layers then go through dimension reduction to end up in 1×1 convolutions.

4.4 MobileNets

As consumer technology becomes thinner and lighter, a rising interest in small, efficient neural networks for mobile applications has developed [23] [24] [25].
MobileNets [26] is based on the concept of factorised convolution, which factorises a standard convolution into a depthwise convolution and a point wise convolution, originally introduced in [27]. Considering M as the number of input channels, N as the number of output channels and Dk as the spatial dimension of the kernel which is assumed to be square, the reduction in computation is:

$$\frac{1}{n^2} + \frac{1}{D_k^2} \qquad [26]$$

4.5 ResNet50

Unlike traditional sequential network architectures such as AlexNet, OverFeat, and VGG, ResNet is instead a form of “exotic architecture” that relies on micro-architecture modules (also called “network-in-network architectures”). First introduced by He et al. in their 2015 paper[28], the ResNet architecture has become a seminal work, demonstrating that extremely deep networks can be trained using standard SGD (and a reasonable initialization function) through the use of residual modules. We used ResNet50 pretrained on the ImageNet dataset for Feature Extraction.
Keeping the Conv layer freezed we trained the densely connected classifier on the augmented data. In order to make the model generalize well with our dataset, we fine-tuned it. In which we unfreezed a few of the top layers of the

freezed model used for feature extraction, and jointly trained the fully connected classifier and these top layers. Validation accuracy increased in all the models. Accuracy of the models with and without data augmentation is shown in Table 2.

Table 2. Test accuracy of different pre-trained (on Imagenet) models

Model	**Parameters**	**Acc. (without Aug)**	**Acc.(with Aug.)**
AlexNet	60M	82.71	89.89
VGG19	144M	93.2	94.9
Inception V3	5M	66.26	74.18
ResNet50	25.6M	89.58	90.26
MobileNets V1.0	4.2M	69.6	70.8

5. Further Improvements

To further enhance the accuracy and speed up the training process, we implemented various techniques that have demonstrated significant improvements than the conventional techniques.

Finding the optimal learning rate region is utmost important as it drastically affects the performance and speed of the network. We used the technique developed by Leslie N. Smith [29], where we simply keep increasing the learning rate from a very small value, until the loss stops decreasing. We then choose an optimal learning rate by estimating “reasonable bounds”. Further also varying the learning rate cyclically between reasonable boundary values helps in achieving improved classification accuracy and often in fewer iterations. This can be understood by keeping in mind the intuition that as we get closer to the optimal weights, we might want to take smaller steps.To get a more accurate and stable model, we must find weight space that is robust to small changes to the weight and in turn generalise well to the unseen data. And to prevent from getting stuck in a ‘spikey’ region of weight space, the cyclic restarts of learning rate can help to jump to a different weight landscape and avoid getting stuck in a region which will not generalise well.

The initial layers of the pre-trained models can be thought of learning more general-purpose features [30]. So, while fine-tuning our models, we would be expecting the initial layers to go through less changes in order to accomodate our dataset. Keeping this in mind we will use different learning rates for different layers: increasing the learning rate with the depth of the network.

We have differentiated our models into three blocks; and then accordingly we chose a learning rate for each block. It is referred to as differential learning rates(DLR)[31].While data augmentation during training time significantly improves the performance of the model, we can also use it during inference. Instead of making prediction on a single image, we will be making predictions on a number of augmented versions of the original image and taking the average prediction as the final answer.

All the networks ran for a total of 10 epochs when training only the fully connected layers and 15 epochs when training all the layers

Table 3. Test accuracy using these techniques

Model	Acc(training FC layers only)	Acc(training all layers with DLR annealing)	Precision	Recall	F-beta Score	Acc(Using test time augmentation)
VGG16	90.97% (10 epochs)	93.81% (15 epochs)	0.907	0.886	0.891	94.84%
VGG19	90.46% (10 epochs)	95.36% (15 epochs)	0.922	0.908	0.914	95.87%
Resnet34	90.20% (10 epochs)	93.29% (15 epochs)	0.857	0.830	0.837	93.88%
Resnet50	91.75% (10 epochs)	96.13% (15 epochs)	0.928	0.922	0.922	96.39%

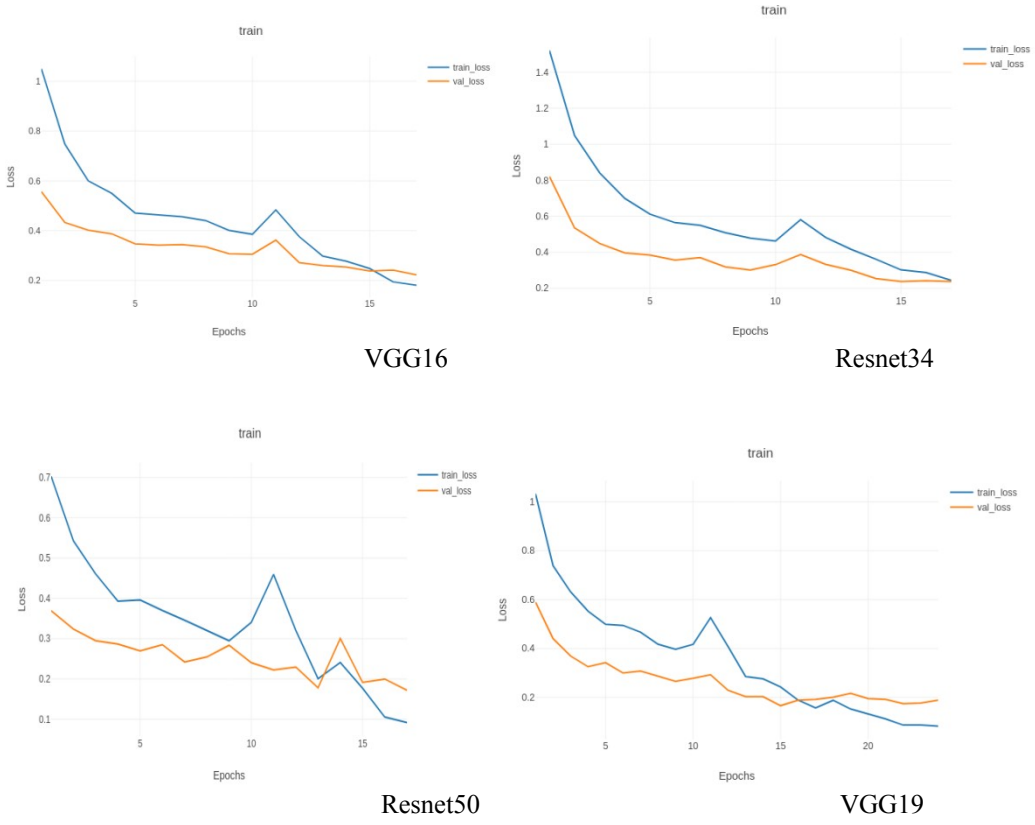

Figure 2: Training and validation loss plots of various model

The confusion matrix of Resnet34 in **figure 3** shows that it predicted the classes correctly to large extent. Some of the misclassification are merely due to large inter-class similarity between the classes such as head-lamp broken, tail-lamp broken and glass-shatter because all three involve broken glasses.The uncertainty in tail lamp broken is also because of the small percentage of images in this category. Moreover, smashed cars contains dents, so some of the images which belong to bumper dent were misclassified as smashed. Zoomed viewpoints of images involving dents may also have led to misclassification of bumper dent as door dent. Out of the 388 testing images 27 images are misclassified and 361 images are classified correctly.

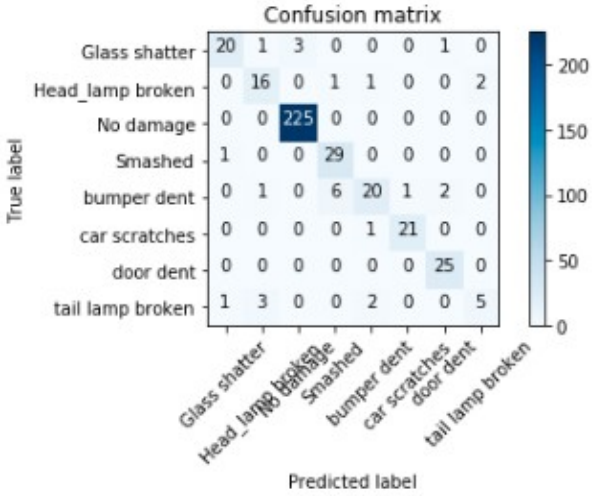

Figure 3: Confusion matrix For Resnet34

6. Class Activation Mapping:

CAM (Class Activation Mapping) technique developed in the 2015[32] allows one to visualize which pixels of the original image are responsible for predicting the corresponding class, highlighting the discriminative object parts detected by the CNN.Moreover,We can be more certain that the CNN is working properly and gain some intuitions where the CNNs are failing as it localizes the main object in the image.It enables classification-trained CNN models to learn to perform object localization, without using any bounding box annotations.

Figure 4: Examples of CAMs, few images of Damaged Cars , and their class activation maps with them. We observe that the damaged regions of the images are often highlighted. Apart from this, similar observations are recorded in other samples.

We generated class activation maps (CAM)[ ] using global average pooling (GAP) in CNNs . A class activation map for a particular category indicates the discriminative image regions used by the CNN to identify that category.

7. Damage Detection Using YOLOv3

Yolov3 consists of a 106 layer architecture [8]. It improves on version 2 by using residual connections and upsampling. .[YOLOv3: An Incremental Improvement]. It uses the darkNet architecture. Darknet-53 performs on par with state-of-the-art classifiers but with fewer floating point operations and more speed. Darknet is 1.5X faster than Resnet-101 and 2X faster than Resnet-152 while achieving comparable performance.

We first tested the object detection capabilities of Yolov3 by only labelling damaged areas of the image. After achieving satisfactory results**(table 6)**, we split the data into 6 categories : (1)Bumper Dent (BD) , (2)Car Scratches (CR), (3)Door Dent (DD), (4)Glass Shatter (GS), (5)Lamp Broken (LB), (6)Smashed (SM). As observed, the Average Precision of the "Car scratches" and "Lamp Broken" classes is comparatively lower because of fewer images in the training dataset.

Results of Yolov3 were validated across a set consisting of images of 416 * 416 Resolution. The Mean Average Precision score obtained was 74.23 %. When tested across a set of 608 * 608 images, the precision was improved by 3.5 percent to 77.78 % which shows that the model accuracy can be improved during inference by passing a higher resolution image through the network trained on relatively lower resolution; this increases the precision and makes it possible to detect small objects.

Table 4. Results on YOLOv3 (416x416) Map score: 0.7423

Threshold	Precision	Recall	F1-score	Avg. IOU
0.00	0.59	0.81	0.68	43.22%
0.25	0.82	0.73	0.77	61.25%
0.40	0.84	0.71	0.77	62.99%
0.50	0.86	0.70	0.77	64.18% .18%
0.70	0.88	0.69	0.77	65.70%

Table 5. Result on YOLOv3 (608x608) Map score 0.7778

Threshold	Precision	Recall	F1-score	Avg. IOU
0.40	0.70	0.80	0.74	50.21%

AP scores: **BD**: 77.16% **CR:** 59.21% **DD:** 88.21% **GS:** 71.07% **LB:** 64.56% **SM:** 85.19%

Figure 5: Sample Images of Damage Detection via YOLOv3

Table 6. Results on damage detection Map score: 0.663

Threshold	Precision	Recall	F1-score	Avg. IOU
0.00	0.65	0.80	0.72	45.76%
0.25	0.81	0.78	0.80	57.20%
0.40	0.82	0.77	0.79	57.66%
0.50	0.82	0.77	0.79	57.66%
0.70	0.84	0.75	0.79	59.53%

8. Pipeline for Insurance Claim Process

Keeping in mind the importance of accuracy and hassle free results for the users of our model, we have proposed a two-step process to combine object detection and classification. The basic strategy is to use YOLOv3 framework to detect the damaged region and then classifying that region using a CNN model trained on the damage dataset.

Initially, the Customer will upload an image to the server using an insurance claim app. Then the image will be passed through an object detector; in our case it is YOLOv3. The detector will be able to detect the region of the vehicle which is damaged. After localizing the damaged region, the proposed region will go through a CNN trained on damaged dataset to classify the type of damage. Furthermore, images aggregated overtime can be used to increase the size of dataset and in turn increase the accuracy of the system by training on a more diverse dataset.

The current accuracy on detecting damages using YOLOv3 on our dataset is depicted in Table 6 . Taking in consideration the relatively small size of our dataset, the results are very encouraging, and are able to learn features that are common in all the different type of damages. We believe only increasing the size of the dataset would be enough to increase the accuracy further. So as we keep storing images uploaded by customers and then training our system on them,it would improve the durability of the pipeline in detecting regions of damage as it will lead to a large dataset which eventually allow the network to learn more significant features which generalise well to inter-class variations.

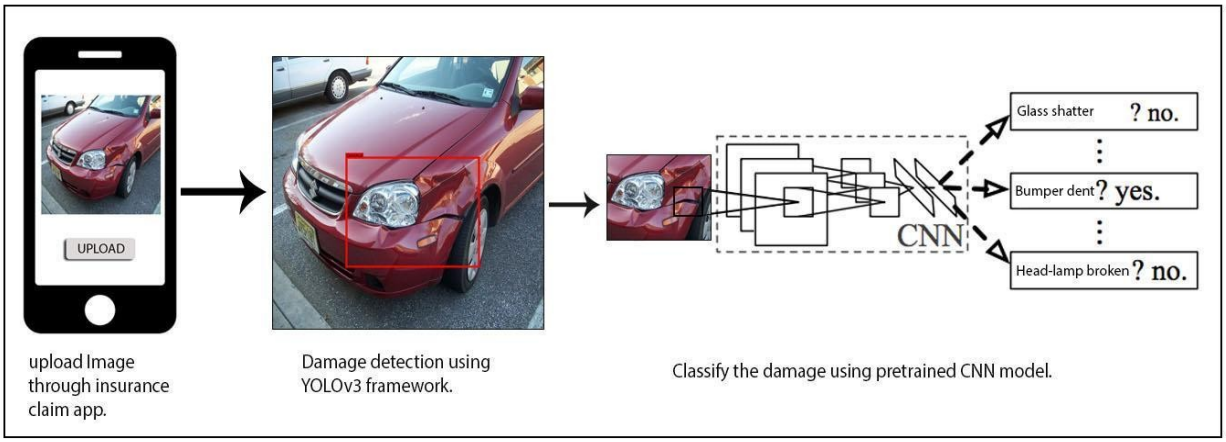

Figure 6: Pipeline Overview

9. Conclusion

Stating the two most prudent quotes of our time by Andrew Ng , "AI is the new electricity", and Clive Humby "Data is the new oil", based on this motivation we here amalgamate both data and AI to provide a novel approach for automating the vehicle damage insurance claims. In this paper, we demonstrate a mechanism to classify/detect damage in vehicles. For this, we manually collected versatile dataset from the internet through running web crawler on various search engines like google and bing, and used deep learning models for the damage classification task. Combining transfer learning with cyclic learning rates for training neural networks we were able to outperform the current state-of-the-art in vehicle damage classification by a significant margin. We are also successful in detecting the damaged part of the vehicle using YOLO framework. Even though our dataset is comparatively small by the standard of other deep learning datasets but several quantitative evaluations show the power and potential of the proposed approach. We have also proposed a practical system application which can fully automate the insurance claim process and can profit many car insurance companies. We believe a larger dataset would improve the results and make this system ready for much more dynamic real-life scenarios.

10. References

[1]http://www.ey.com/publication/vwluassets/ey-doesyour-firm-need-a-claims-leakage-study/ey-does-yourfirm-need-a-claim -leakage-study.pdf

[2] https://tractable.ai/

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation," in The IEEE
Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

[4] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M.Smeulders, Selective search for object recognition," International Journal of Computer Vision, vol. 104, pp. 154{171, Sep 2013.

[5] R. Girshick, Fast r-cnn, in The IEEE International Conference on Computer Vision (ICCV), December 2015.

[6] S. Ren, K.He, R.Girshick, and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 1137{1149, June 2017.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: Uni_ed, real-time object detection," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[8] J. Redmon and A. Farhadi, Yolov3: An incremental improvement," 2018.

[9] S. Jayawardena, Image based automatic vehicle damage detection. PhD thesis, College of Engineering and Computer Science (CECS), 12 2013.

[10] S. Gontscharov, H Baumgartel, A.Kneifel, and K.-L. Krieger, Algorithm development for minor damage identification in vehicle bodies using adaptive sensor data processing," Procedia Technology, vol. 15, pp. 586 {
594,2014. 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering.

[11]. Multi-camera vision system inspects cars for dents caused by hail

[12] Y.-J. Cha, W. Choi, and O. B''uy''uk''ozt''urk, Deep learning-based crack damage detection using convolutional neural networks," Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361{378, 2017.

[13] Y.-J. Cha, J. Chen, and O. B''uy''uk''ozt''urk, Output-only computer vision based damage detection using phase-based optical ow and unscented kalman _lters," Engineering Structures, vol. 132, pp. 300 { 313, 2017.

[14] A. Mohan and S. Poobal, Crack detection using image processing: A critical review and analysis," Alexandria Engineering Journal, vol. 57, no. 2, pp. 787 { 798, 2018.

[15] K. Patil, M. Kulkarni, A. Sriraman, and S. Karande, Deep learning based car damage classi_cation," in 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 50{54, Dec 2017.2

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, How transferable are features in deep neural networks?," in Advances in Neural Information Processing Systems 27 (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3320{3328, Curran Associates, Inc., 2014.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

[18] K. Simonyan and A. Zisserman, Very deep convolutional networks for
large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classi_cation with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097{1105, Curran Associates, Inc., 2012.

[20] Y. Lecun, L. Bottou, Y. Bengio, and P. Ha_ner, Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, pp. 2278{2324, Nov 1998.

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors," CoRR, vol. abs/1207.0580, 2012.

[22] C. Szegedy, V. Vanhoucke, S. Io_e, J. Shlens, and Z. Wojna, Rethinking the inception architecture for computer vision," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[23] J. Jin, A. Dundar, and E. Culurciello, Flattened convolutional neural networks for feedforward acceleration," CoRR, vol. abs/1412.5474, 2014.
[24] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio,
Quantized neural networks: Training neural networks with low precision weights and activations," Journal of Machine Learning Research, vol. 18, no. 187, pp. 1{30, 2018.
[25] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, Xnor-net: Imagenet classi_cation using binary convolutional neural networks," in Computer Vision { ECCV 2016 (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 525{542, Springer International Publishing, 2016.
[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, Mobilenets: E_cient convolutional neural networks for mobile vision applications," CoRR, vol. abs/1704.04861, 2017.3
[27].L. Sifre. Rigid-motion scattering for image classification. PhD thesis, Ph. D. thesis, 2014.1,3
[28] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
[29] L. N. Smith, Cyclical learning rates for training neural networks," in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464{472, March 2017.
[30] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks," in Computer Vision { ECCV 2014 (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 818{833, Springer International Publishing, 2014.
[31]. Fastai deep learning course lesson 1. https://course.fast.ai/videos/?lesson=1
[32]. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, Learning deep features for discriminative localization," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.4