

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/322671990>

# Deep Learning Based Car Damage Classification

Conference Paper · December 2017

DOI: 10.1109/ICMLA.2017.0-179

## CITATIONS

53

## READS

14,435

## 4 authors:



**Kalpesh Patil**

Indian Institute of Technology Bombay

2 PUBLICATIONS 61 CITATIONS

[SEE PROFILE](#)



**Mandar Kulkarni**

Flipkart

33 PUBLICATIONS 180 CITATIONS

[SEE PROFILE](#)



**Anand Sriraman**

Tata Consultancy Services Limited

5 PUBLICATIONS 130 CITATIONS

[SEE PROFILE](#)



**Shirish S. Karande**

Tata Consultancy Services Limited

86 PUBLICATIONS 708 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Data Driven Requirements Engineering [View project](#)



Knowledge distillation using unlabeled mismatched images [View project](#)

# Deep Learning Based Car Damage Classification

Kalpesh Patil\*, Mandar Kulkarni<sup>†</sup>, Anand Sriraman<sup>†</sup> and Shirish Karande<sup>†</sup>

\*IIT Bombay, Mumbai, India <sup>†</sup>TCS Innovation Labs, Pune, India

Email: kalpeshpatil@iitb.ac.in, {mandar.kulkarni3,anand.sriraman,shirish.karande}@tcs.com

**Abstract**—Image based vehicle insurance processing is an important area with large scope for automation. In this paper we consider the problem of car damage classification, where some of the categories can be fine-granular. We explore deep learning based techniques for this purpose. Initially, we try directly training a CNN. However, due to small set of labeled data, it does not work well. Then, we explore the effect of domain-specific pre-training followed by fine-tuning. Finally, we experiment with transfer learning and ensemble learning. Experimental results show that transfer learning works better than domain specific fine-tuning. We achieve accuracy of 89.5% with combination of transfer and ensemble learning.

## I. INTRODUCTION

Today, in the car insurance industry, a lot of money is wasted due to claims leakage [1] [2]. Claims leakage / Underwriting leakage is defined as the difference between the actual claim payment made and the amount that should have been paid if all industry leading practices were applied. Visual inspection and validation have been used to reduce such effects. However, they introduce delays in the claim processing. There have been efforts by a few start-ups to mitigate claim processing time [3] [4]. An automated system for the car insurance claim processing is a need of the hour.

In this paper, we employ Convolutional Neural Network (CNN) based methods for classification of car damage types. Specifically, we consider common damage types such as bumper dent, door dent, glass shatter, head lamp broken, tail lamp broken, scratch and smash. To the best of our knowledge, there is no publicly available dataset for car damage classification. Therefore, we created our own dataset by collecting images from web and manually annotating them. The classification task is challenging due to factors such as large inter-class similarity and barely visible damages. We experimented with many techniques such as directly training a CNN, pre-training a CNN using auto-encoder followed by fine-tuning, using transfer learning from large CNNs trained on ImageNet and building an ensemble classifier on top of the set of pre-trained classifiers. We observe that transfer learning combined with ensemble learning works the best. We also devise a method to localize a particular damage type. Experimental results validate the effectiveness of our proposed solution.

## II. RELATED WORKS

Deep learning has shown promising results in machine learning applications. In particular, CNNs perform well for computer vision tasks such as visual object recognition and detection [5] [6]. Application of CNNs to structural damage

assessment has been studied in [7]. The authors propose a deep learning based method for Structural Health Monitoring (SHM) to characterize the damage in the form of cracks on a composite material. Unsupervised representation is employed and results have been shown on a wide range of loading conditions with limited number of labeled training image data.

Most of the supervised methods need large amounts of labeled data and compute resources. Unsupervised pre-training techniques such as Autoencoders [8] have been proven to improve the generalization performance of the classifier in case of small number of labeled samples. For images, Convolutional AutoEncoders (CAE) [9] have shown promising results.

A very well known technique which has worked effectively in case of small labeled data is transfer learning [10] [11]. A network which is trained on a source task is used as a feature extractor for target task. There are many CNN models trained on Imagenet which are available publicly such as VGG-16 [12], VGG-19 [12], Alexnet [6], Inception [13], Cars [14], Resnet [15]. Transferable feature representation learned by CNN minimizes the effect of over-fitting in case of a small labeled set [10].

Traditional machine learning techniques have also been experimented for automated damage assessment. Jaywardena et al [16] proposed a method for vehicle scratch damage detection by registering 3D CAD model of undamaged vehicle (ground truth) on the image of the damaged vehicle. There has been attempts to analyze damage in geographical regions using satellite images [17] [18] [19]. To best of our knowledge, deep learning based techniques have not been employed for automated car damage classification, especially for the fine-granular damage classification.

## III. DATASET DESCRIPTION

Since there is no publicly available dataset for car damage classification, we created our own dataset consisting of images belonging to different types of car damage. We consider seven commonly observed types of damage such as bumper dent, door dent, glass shatter, head lamp broken, tail lamp broken, scratch and smash. In addition, we also collected images which belong to a no damage class. The images were collected from web and were manually annotated. Table I shows the description of the dataset.

### A. Data augmentation

It is known that an augmentation of the dataset with affine transformed images improves the generalization performance of the classifier. Hence, we synthetically enlarged the dataset

Classes	Train size	Aug. train size	Test size
Bumper Dent	186	1116	49
Door dent	155	930	39
Glass shatter	215	1290	54
Head-lamp broken	197	1182	49
Tail-lamp broken	79	474	21
Scratch	186	1116	46
Smash	182	1092	45
No damage	1271	7626	318

TABLE I  
Description of our dataset.

approx. five times by appending it with random rotations (between -20 to 20 degrees) and horizontal flip transformations.

For the classification experiments, the dataset was randomly split into 80%-20% where 80% was used for training and 20% was used for testing. Table I describes the size of our train and test sets.

Fig. 1 shows sample images for each class. Note that the classification task is non-trivial due to high inter-class similarity. Especially, since the damage does not cover the entire image (but a small section of it), it renders classification task even more difficult.

#### IV. TRAINING A CNN

In the first set of experiments, we trained a CNN starting with the random initialization. Our CNN architecture consist of 10 layers: Conv1-Pool1-Conv2-Pool2-Conv3-Pool3-Conv4-Pool4-FC-Softmax where Conv, Pool, FC and Softmax denotes convolution layer, pooling layer, fully connected layer and a softmax layer respectively. Each convolutional layer has 16 filters of size  $5 \times 5$ . A RELU non-linearity is used for every convolutional layer. The total number of weights in the network are around 423K. Dropout was added to each layer which is known to improve generalization performance. We trained a CNN on the original as well as on the augmented dataset.

Table II shows the result of the CNN training from random initialization. It can be seen that the data augmentation indeed helps to improve the generalization and provides better performance than just training on the original dataset.

We are aware that the data used for training the CNN (even after augmentation) is quite less compared to the number of parameters and it may result in overfitting. However, we performed this experiment to set a benchmark for rest of the experiments.

Method	Without Augmentation			With Augmentation		
	Acc	Prec	Recall	Acc	Prec	Recall
CNN	71.33	63.27	52.5	72.46	64.03	61.01
AE-CNN	73.43	67.21	55.32	72.30	63.69	59.48

TABLE II  
Test accuracy with CNN training and (CAE + fine-tuning).

##### A. Convolutional Autoencoder

Unsupervised pre-training is a well known technique, known to be useful in cases where training data is scarce [8].

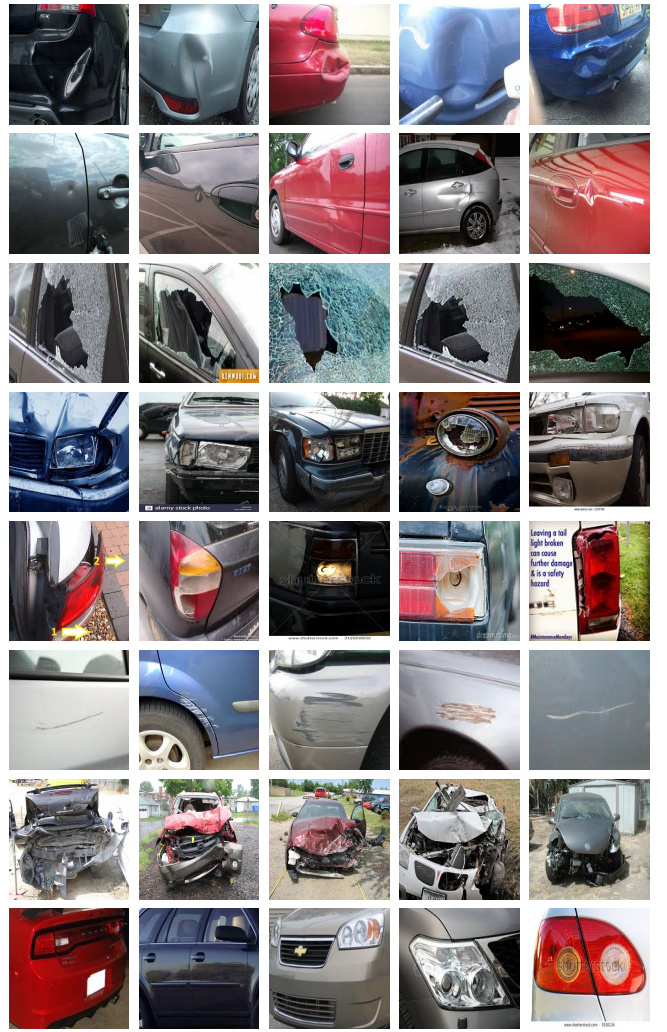


Fig. 1. Sample images for car damage types. Rows from top to bottom indicates damage types Bumper dent, Door dent, Glass shatter, Head-lamp broken, Tail-lamp broken, Scratch, Smash and No damage

The primary objective of an unsupervised learning method is to extract useful features from the unlabeled dataset by learning the input data distribution. They detect and remove input redundancies, and usually only preserve essential aspects of the data which tend to assist the classification task. A fully connected auto-encoders, especially in case of images, leads to large number of trainable parameters. Convolutional AutoEncoders (CAE) provide a better alternative because of less number of parameters due to sparse connections and weight sharing [9]. CAEs are trained in a layer wise manner where unsupervised layers can be stacked on top of each other to build the hierarchy. Each layer is trained independently of others where output of a previous layer acts as an input for the subsequent layer. Finally, the complete set of layers are stacked and fine-tuned by back-propagation using the cross-entropy objective function. Unsupervised initialization tend to avoid local minima and increase the networks performance stability.

For training a CAE, we used unlabeled images from Stan-

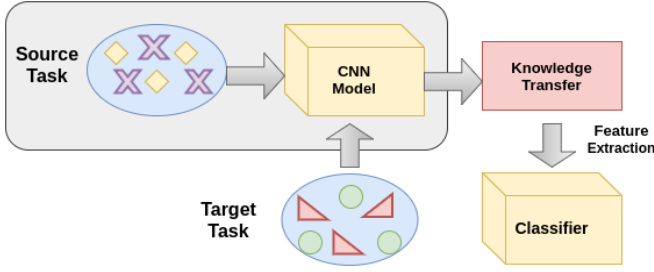


Fig. 2. Transfer learning setup used in our experiments. Source task is ImageNet classification while the target task is car damage classification.

ford car dataset [20]. The size of the dataset was synthetically increased by adding rotation and flip transformations. Since the target images belong to car damage type, we expect that learning the car specific features should help the classification task. The layers are then fine tuned using a smaller learning rate as compared to the training. The row, AE-CNN, in Table II shows the result with autoencoder pre-training. It can be seen that an autoencoder pre-training does help the classification task. A similar experiment was performed using augmented car damage images and there as well we see improvement in the test accuracy as compared to no pre-training.

## V. TRANSFER LEARNING

Transfer learning has shown promising results in case of small labeled data [10] [11]. In the transfer learning setting, knowledge from the source task is transferred to the target task. The intuition is that some knowledge is specific to individual domains, while some knowledge may be common between different domains which may help to improve performance for the target domain / task. However, in the cases where the source domain and target domain are not related to each other, brute-force transfer may be unsuccessful and can lead to the degraded performance. In our case, we use the CNN models which are trained on the Imagenet dataset. Since the Imagenet dataset contains car as a class, we expect the transfer to be useful which we extensively validate by experimenting with multiple pre-trained models. Fig. 2 shows the transfer learning experiment setup we use. Since we use pre-trained models which are trained for ImageNet, the Source task is the ImageNet classification. The pre-trained model is used as a feature extractor for Target task i.e. car damage images. Table III indicates the details of pre-trained networks used, their parameters and feature dimension.

We input car damage images to each network and extract their feature vectors. We then train a linear classifier on these features. We experimented with two linear classifiers, a linear SVM and a Softmax. In case of linear SVM, the penalty parameter  $C$  was set to 1 for all experiments. In case of the Softmax classifier, we used Adadelta optimization scheme and cross entropy loss. We trained the classifier for 100 epochs and chose the model with best classification performance. Also, since data augmentation helps the classifier in generalization, we train linear classifiers on augmented feature set as well.

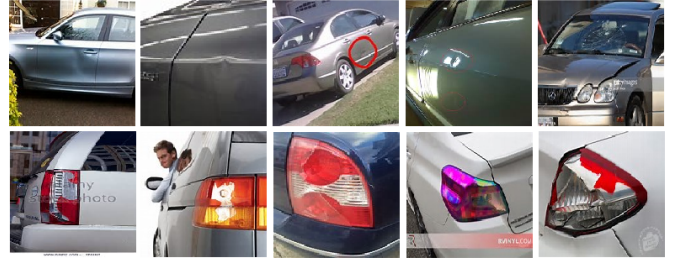


Fig. 3. Examples of test images mis-classified as 'no damage' class with Resnet. Note that the damaged portion is barely visible.

Table III indicates the accuracy, precision and recall when using these pre-trained models. It can be seen that the Resnet performs the best among all the pre-trained models. The data augmentation boosts the performance in most of the cases. During the experimentation, it was observed that the Softmax classifier works better than linear SVM and it is faster to train.

Surprisingly, the GoogleNet pre-trained model fine-tuned using car dataset, performed the worst. It indicates that only car object based features may not be effective for classifying damage types. The poor performance of autoencoder based approach may as well be due to this effect. It underlines the effectiveness of feature representation learned from large and diverse input data distributions.

We observe that the major factor in the mis-classifications is the ambiguity between damage and 'no damage' class. This is not surprising because, the damage of a part usually occupies a very small portion of the image and renders identification difficult even for the human observer. Fig. 3 shows few examples of test images of damage which are mis-classified as no damage.

### A. Ensemble method

To further improve the accuracy, we performed an experiment with ensemble of the pre-trained classifiers. For each training image, class probability predictions are obtained from multiple pre-trained networks. The weighted average of class posteriors is then used to obtain the final decision class. The weights to be used for the linear combination are learned by solving following least squares optimization

$$C = \frac{1}{N} \sum_{i=1}^N \|P_i w - g_i\|_2^2 \quad (1)$$

Here,  $P_i \in R^{m \times n}$  indicates the matrix of posteriors for the  $i^{th}$  training point,  $n$  indicates the number of pre-trained models used and  $m$  indicates the number of classes.  $w$  indicates the weight for each posterior and  $g_i$  indicates the (one-hot encoded) ground truth label for the  $i^{th}$  training data point.  $N$  is the total number of training points. The optimization is solved using gradient descent where learning rate is adjusted to yield the best test performance. Since Softmax performed the best, we use it for obtaining class posteriors. Table IV shows the result of the experiment. It can be seen that the ensemble



Model	Params	Dim	Without Augmentation						With Augmentation					
			Linear SVM			Softmax			Linear SVM			Softmax		
			Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall
Cars [14]	6.8M	1024	57.33	47.24	56.46	60.38	47.23	32.39	58.45	48.58	56.97	64.25	52.73	39.16
Inception [13]	5M.	2048	68.12	57.46	55.53	71.82	61.75	56.71	68.60	58.50	54.44	71.50	69.47	52.81
Alexnet	60 M.	4096	70.85	61.68	64.60	70.85	61.42	58.09	73.26	62.83	61.72	73.91	66.83	63.36
VGG-19 [12]	144M.	4096	82.77	78.62	73.16	84.22	80.76	73.60	82.29	76.30	70.60	83.90	80.74	73.41
VGG-16 [12]	138M.	4096	83.74	77.79	75.41	84.86	81.91	73.56	82.93	78.62	71.96	82.72	78.99	70.30
Resnet [15]	25.6M.	2048	86.31	80.87	78.30	<b>88.24</b>	84.38	81.10	87.92	84.40	78.94	87.92	83.68	79.47

TABLE III

Classification performance for transfer learning. Comparison of test accuracies with different pre-trained CNN models. Note that Resnet performs the best.

Ensemble	Without Augmentation			With Augmentation		
	Acc	Prec	Recall	Acc	Prec	Recall
Top-3	89.37	88.05	80.91	88.40	85.88	78.91
All	<b>89.53</b>	88.16	80.92	88.24	86.45	78.41

TABLE IV

Classification performance for Ensemble technique using Top-3 and All models

(Top-3 and All) works better than the individual classifiers, as expected.

### B. Damage localization

With the same approach, we can even localize the damaged portion. For each pixel in the test image, we crop a region of size  $100 \times 100$  around it, resize it to  $224 \times 224$  and predict the class posteriors. A damage is considered to be detected if the probability value is above certain threshold. Fig. 4 shows the localization performance for damage types such as glass shatter, smash and scratch with Resnet classifier and probability threshold of 0.9.

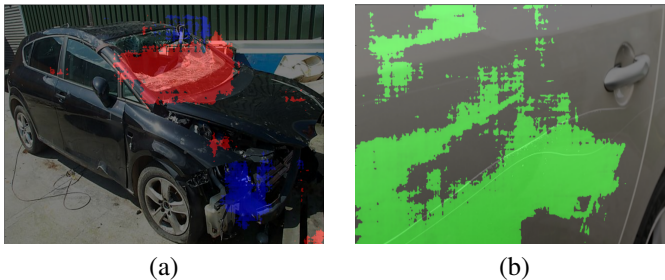


Fig. 4. Damage localization. (a) Glass shatter (red) and Smash (blue), (b) Scratch (green). Note that our approach is able to localize damage correctly.

## VI. CONCLUSION

In this paper, we proposed a deep learning based solution for car damage classification. Since there was no publicly available dataset, we created a new dataset by collecting images from web and manually annotating them. We experimented with multiple deep learning based techniques such as training CNNs from random initialization, Convolution Autoencoder based pre-training followed by supervised fine tuning and transfer learning. We observed that the transfer learning performed the best. We also note that only car specific

features may not be effective for damage classification. It thus underlines the superiority of feature representation learned from the large training set.

## REFERENCES

- [1] "http://www.ey.com/publication/vwluassets/ey-does-your-firm-need-a-claims-leakage-study/ey-does-your-firm-need-a-claims-leakage-study.pdf."
- [2] "https://www.irmi.com/articles/expert-commentary/controlling-claims-leakage-through-technology."
- [3] "http://www.sightcall.com/insurance-visual-claims-fight-fraud-and-reduce-costs/."
- [4] "http://www.tractable.io/."
- [5] B. Y. Lecun Y., Bottou L. and H. P., "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, 1998.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification 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., Curran Associates, Inc., 2012, pp. 1097–1105.
- [7] M. G. M. R. G. Soumalya Sarkar, Kishore K. Reddy, "Deep learning for structural health monitoring: A damage characterization application," in *Annual Conference of the Prognostics and Health Management Society*, 2016.
- [8] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.
- [9] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.
- [10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.
- [11] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Re-thinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.
- [14] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3973–3981.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.
- [16] S. Jayawardena *et al.*, "Image based automatic vehicle damage detection," Ph.D. dissertation, Australian National University, 2013.
- [17] F. Samadzadegan and H. Rastiveisi, "Automatic detection and classification of damaged buildings, using high resolution satellite imagery and vector data," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, pp. 415–420, 2008.

- [18] K. Kouchi and F. Yamazaki, "Damage detection based on object-based segmentation and classification from high-resolution satellite images for the 2003 boumerdes, algeria earthquake," in *Proceedings of the 26th Asian conference on Remote Sensing, Hanoi, Vietnam, 2005*.
- [19] E. Rathje and M. Crawford, "Using high resolution satellite imagery to detect damage from the 2003 northern algeria earthquake," in *13th World Conference on Earthquake Engineering, August, 2004*, pp. 1–6.
- [20] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 554–561.