Car Damage Detection and Classification

Conference Paper · July 2020

DOI: 10.1145/3406601.3406651

2 authors:

Phyu Mar Kyu
King Mongkut's Institute of Technology Ladkrabang

Kuntpong Woraratpanya
King Mongkut's Institute of Technology Ladkrabang

Car Damage Detection and Classification

Phyu Mar Kyu[1]
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
62606003@kmitl.ac.th

Kuntpong Woraratpanya
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
kuntpong@it.kmitl.ac.th

ABSTRACT

Nowadays, the proliferation of automobile industries is directly related to the increasing number of car incidents. So, insurance companies are facing many simultaneous claims and solving claims leakage. The sense of Artificial Intelligence (AI) based on machine learning and deep learning algorithms can help to solve these kinds of problem for insurance industries. In this paper, we apply deep learning-based algorithms, VGG16 and VGG19, for car damage detection and assessment in real-world datasets. The algorithms detect the damaged part of a car and assess its location and then its severity. Initially, we discover the effect of domain-specific pre-trained CNN models, which are trained on an ImageNet dataset, and followed by fine-tuning, because some of the categories can be fine-granular to get our specific tasks. Then we apply transfer learning in pre-trained VGG models and use some techniques to improve the accuracy of our system. We achieve the accuracy of 95.22% of VGG19 and 94.56% of VGG16 in the damaged detection, the accuracy of 76.48% of VGG19 and 74.39% of VGG16 in damage localization, the accuracy of 58.48% of VGG19 and 54.8% of VGG16 in damage severity with the combination of transfer learning and L2 regularization. From their results, the performance of VGG19 is better than VGG16. After analyzing and implementing our models, we find out that the results of using transfer learning and L2 regularization can work better than those of fine-tuning.

CCS CONCEPTS

• Computing methodologies • Machine learning • Machine learning approaches • Neural networks

KEYWORDS

Damage assessment, Deep learning, Machine learning, Transfer learning, Pre-trained VGG

IAIT2020, July 1–3, 2020, Bangkok, Thailand

ACM ISBN 978-1-4503-7759-1/20/07…$15.00
https://doi.org/10.1145/3406601.3406651

ACM Reference format:
Phyu Mar Kyu and Kuntpong Woraratpanya. 2020. Car Damage Detection and Classification. In *Proceedings of International Conference on Advances in Information Technology (IAIT2020), July 1-3, 2020, Bangkok, Thailand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3406601.3406651

1 Introduction

The insurance industry is one of the first industries invested in innovation, the latest technology and artificial intelligence (AI) [1]. In today's world, when the rate of car accidents is increasing, car insurance companies waste millions of dollars annually, due to claims leakage. The sense of AI technology based on machine learning and deep learning can help problems such as analyzing and processing data, detecting frauds, lessening risks and automating claim process in insurance industries [2]. So, insurance firms have looked for faster damage assessment and agreement of claims.

However, a development of modern applications to overcome such problems is still challenging, especially in applying deep learning for car damage assessment. Deep learning is an efficient approach for solving complex tasks, but it needs more resources for model development, i.e., for training a model, deep learning requires a huge dataset and takes more computation time. To realize deep learning approach for car damaged assessment, this paper focuses on two challenges for creating an efficient model: (i) car damaged datasets for training and (ii) reduction of computation time.

Since car damaged assessment is a specific domain, it is lack of publicly available datasets for car damaged images with labelling. Training a model with a small dataset is the most challenging. In this case, [3] demonstrated significant progress on how to solve classification problems when the small dataset is not enough to train a CNN model. Using data augmentation by manually collecting and labelling data on the Web can solve this problem [2, 4, 5, 6].

A reduction of model training time is also the most challenge. Typically, a traditional CNN model can be very time-consuming to perform image classification tasks and identify the correct weights for the network by multiple forward and backward iterations. This process may take days or even weeks to complete it using GPUs. Fortunately, the model training time can be reduced by mean of pre-trained CNN models, which have been previously trained on large benchmark datasets like an ImageNet dataset [7]. We can freely download their weights and apply their architectures for other specific tasks via transfer learning.

Moreover, pre-trained CNN models can be used as a feature extractor and a fine-tuned. However, their frameworks are very complicated to understand because the variance is intensity. Jeffrey [8] found out a way how to focus on the impact of certain hyper-parameters and exploring theory to adapt them. In the same way, the learning strategy of training the model with k-epoch and evaluating its performance can get the best learner parameters when the validation performance converging towards the right expected values. This way can help to adjust the regularization, which tries to defeat the overfitting problem and the influence of hyper-parameters [9]. From 2014 to the present, deep learning has been excellent in image classification from 2014 up to the present with the extensive use of data and computing resources based on transfer learning solutions [8, 10, 11, 12] and how to modify VGG model with batch normalization [13]. A comprehensive review of deep learning based on transfer learning is provided by [14]. According to [15], they proposed an end-to-end system with a transfer learning based on CNN models on an ImageNet dataset to perform different tasks of localization and detection but not calculate the level of damage part. The similarity in papers [2, 5, 16], they also trained CNN model with both of transfer learning and ensemble learning by comparing with the result of fine-tuning in the pre-trained CNN model on an ImageNet dataset focusing on the accuracy of damage detection. Mahavir Dwivedi [4] applied the YOLO object detection model [17] to train and detect damage region as their important pipeline to improve their performances of damage detection. After all of that, most papers just focus on how to study about CNN models to detect damaged part with many techniques via transfer learning.

In this paper, we used the pre-trained VGG model [11] for VGG16 and VGG19. Both VGG16 andVGG19 won the first and second places in 2014 ILSVRC challenge on the object localization and classification tasks with the error rate of 7.32% and 25.32%. Their unique parts are the stack of convolutional layers with a very small receptive field of 3x3, a 2×2 pixel window of max-pooling and three fully-connected layers. The configurations of their fully connected layers are the same as AlexNet [10]. They can cover the receptive field larger than AlexNet even both of their output features maps are the same size in their last pooling layer. Moreover, their model weights are freely available to load and use for our specific tasks. In addition to this, we do not need to prepare our new datasets with annotation boxes by using them. We just manually collect both damaged and undamaged car images on the Web using selenium and apply data augmentation to enlarge datasets. To define every damaged part with their severity, we have to consider each damaged part into minor, moderate or severe according to Libertymutual.com [18]. Then, we utilise our models with transfer learning and L2 regularization to reduce training time-consuming and the overfitting problem. Using L2 regularization, we do not need to concern with explicit feature selection [19].After that, we apply fine-tuning to adjust some hyper-parameters in our models because we want to compare its result with the result of using transfer learning and L2 regularization, which one is most suitable for our system. To sum up, performances of damage detection, location and severity using transfer learning with L2 regularization are better than that of fine-tuning in our system.

The rest of this paper is organized as follows. In section 2, we describe our proposed methods. The experimental results are reported and discussed in section 3, and finally, the conclusion and future work are presented in section 4.

2 Proposed Methods

Responding to our two challenges of (i) car damaged datasets for training and (ii) reduction of computation time we use the following proposed methods in our system.

2.1 Dataset Description

Generally, the phase of data preparation is very time consuming depending on the requirement of the labelling data. But we did not need to do data cleaning since we used pre-trained VGG because it just requires the original images as input. Using cross-validation to estimate our models would need too much computational time because it is very expensive to train VGG for a long time. Therefore, we decided to split the dataset randomly into separate sets for training (80%) and validation (20%).We randomly put train and validation sets, because creating and training with an ensemble of models would have taken very much time. After training multiple times with different splits, this split proved useful. Finally, the train and test were split such that they have similar images. We created our three datasets based on 1150 car damaged images, which consist of different types of car damage when there is no openly obtainable dataset for car damage classification. To reach our classification procedure, we needed to have our three datasets. So, we manually collected damaged and undamaged car images on the Web using selenium. We describe our three collected datasets in the following.

- ImageNet dataset—Having 12 sub-trees: mammal, bird, fish, reptile, amphibian, vehicle, furniture, musical instrument, geological formation, tool, flower, fruit.
- Dataset 1—Train and validation sets with undamaged and damaged cars.
- Dataset 2—Train and validation sets with damaged cars.
- Dataset 3—Train and validation sets with damaged cars.

2.2 Defining Damage Level

To define every damaged part with their severity, we had to consider each damaged part into minor, moderate or severe according to Libertymutual.com. The classification of car damaged levels is as follows [18].

- Minor Damage—Scratches headlight or dent in the hood of a car.
- Moderate Damage—Large dents in hood, fender or door of a car.
- Severe Damage—Broken axes, bent or twisted frames and destroy air bags of a car.

2.3 Data Augmentation

Data augmentation can develop the size and quality of training datasets for deep learning models. Moreover, the augmented data will correspond to a more wide-ranging set of possible data points, decreasing the distance between the training and validation set, as well as testing sets. Connor Shorten [6] surveyed of data augmentation, a solution to the data-space problem of limited data, to improve the execution of their models and expand limited datasets to take benefit of the abilities of big data.

According to the lack of car damaged datasets for training, we use data augmentation to artificially expand and adapt our small datasets, to improve their performance and decrease their tolerance to the overfitting issue during training. Therefore, we apply it randomly rotation, zooming, dimension shift and flipping renovation plans to differ the generated data.

2.4 Pre-trained VGG

Andrew Zisserman introduced VGG [11]: VGG16 and VGG19 and submitted to Large Scale Visual Recognition Challenge 2014 (ILSVRC-2014). Their VGG model achieves 92.7% of top-5 test accuracy on an ImageNet dataset. Their model won the first and second places in 2014 ILSVRC challenge on object localization and image classification tasks with the error rate of 7.32% and 25.32%. VGG16 has 13 layers of convolutional layers and 3 layers of fully connected layers; VGG19 has 16 layers of convolutional layers and 3 layers of fully connected layers. They use their inputs as a fixed 224×224 RGB image, which passes through a stack of their convolutional layers, where they use filters as a very small 3×3 receptive field, and also utilize the max-pooling, which is performed over a 2×2 window with stride 2. The stride and padding of all convolutional layers are fixed to one pixel. The configurations of their fully connected three layers are the same as AlexNet [10]. They can cover the receptive field larger than AlexNet even both of their output features maps are the same size in their last pooling layer.

A reduction of model training time is the most challenge in deep learning. Typically, a traditional VGG model, such as VGG16 and VGG19, can be very time-consuming to perform image classification tasks and identify the correct weights for the network by multiple forward and backward iterations. This process may take days or even weeks to complete it using GPUs. Fortunately, the model training time can be reduced by a mean of pre-trained VGG, which have been previously trained on large benchmark datasets like an ImageNet dataset [7]. We can freely download their weights and apply their architectures for our specific new tasks via transfer learning. By using them, we do not need to prepare our new datasets with annotation boxes. Moreover, pre-trained VGG can be used as a feature extractor and a fine-tuned. However, their frameworks are very complicated to understand because the variance is intensity. Therefore, we used the pre-trained VGG: pre-trained VGG16 and pre-trained VGG19, with transfer learning, L2 regularization and fine-tuning to reduce training time-consuming, defeat the overfitting problem and adjust the influence of hyper-parameters in our models. We will explain more about transfer learning, the influence of hyper-parameters and L2 regularization as follows.

2.5 Transfer Learning

Typically, machine learning is used to train a CNN model on a specific feature space and the same distribution. A widespread assumption of training, validation and test data must have identical feature spaces with the underlying distribution for a specific task. On the other hand, this opinion may not hold, and thus model need to be rebuild from the scratch if features and distribution change, how arduous process to collect related training data and rebuild it again. Transfer learning is one of the solutions that can solve this case [12]. Every deep learning model trains and places each task from the ground up, while transfer learning concentrates on feature extraction and appropriate data from source tasks and then applies the required data to a target task [2]. When the source and target data are similar, as a result, transfer learning may improve the performance of target tasks.

We applied transfer learning to the pre-trained VGG models to defeat the training times and the overfitting problem on our small datasets and solve classification, regression and clustering problems. The most important fact was that we did not need to train neural networks from scratch; we could take the existing network into our desirable tasks. So, it could significantly reduce the training times when we used the weights of pre-trained VGG models. Furthermore, it had demonstrated significant progress on how to solve classification problems when the small dataset was not enough to train a CNN model [3]. The classes of the pre-trained VGG models are the source tasks, and the detected damaged parts of their locations, and their damaged levels are the target tasks in our system.

2.6 Influence of Hyper-parameters

Adapting the hyper-parameters based on diagnostics in a theoretically way can help to obtain good results with a limited number of tasks. Keeping track of the loss and metric functions during our training datasets can determine good specifications of the learning rate: which helps to control the changing of weights for our network, batch size: which refers to the number of training samples propagated through the network, and the amount of data augmentation in our system. The smaller the learning rate, the larger the batch size and the more stable the learning process but using a large batch size is more expensive. Fine-tuning transferred parameters can give great results according to other related papers. Thus, we fine-tuned the last layers of pre-trained VGG models to adjust hyper-parameters in our models.

2.7 Regularization

Regularization is a way to control the model complexity by preventing the overfitting problem in machine learning. It can limit the splits to avoid redundant classes in k-means clustering, the tree depth and new features (branches) in random decision forests, and the model complexity (weights) in neural networks. Many techniques of regularization are L1 regularization, L2 regularization, dropouts, early stopping, batch normalization. Among them, L1 and L2 regularization techniques are well-known in deep learning. Mathematically, their regularization terms: L1 regularization term of $\lambda_1|\omega|$ and L2 regularization term of $\frac{1}{2}\lambda_2\omega^2$ are added to the loss function. The key dissimilarity between them is that L1 regularization contracts the less important feature's coefficient to zero by removing some features. So, it is just suitable for feature selection in a huge number of features. However, Karpathy [19] proved that L2 regularization was expected to give the best performance and no need to concern with explicit feature selection. In this paper, we used L2 regularization to fit the overfitting problem in our system.

3 Experiments and Discussion

The tasks of the experiments are: (i) to recognize a car or not a car, (ii) to detect every part of a car image to be damaged or undamaged, (iii) to detect the location of the damaged part of the car where it locates in front, rear or side and (iv) to assess the severity of the damaged part of the car how minor, moderate or serve level it is. Responding to our two challenges and the above tasks, we setup the experiments, as schematically shown in Figure 1, applying our proposed methods, and discuss the experimental results as follows.

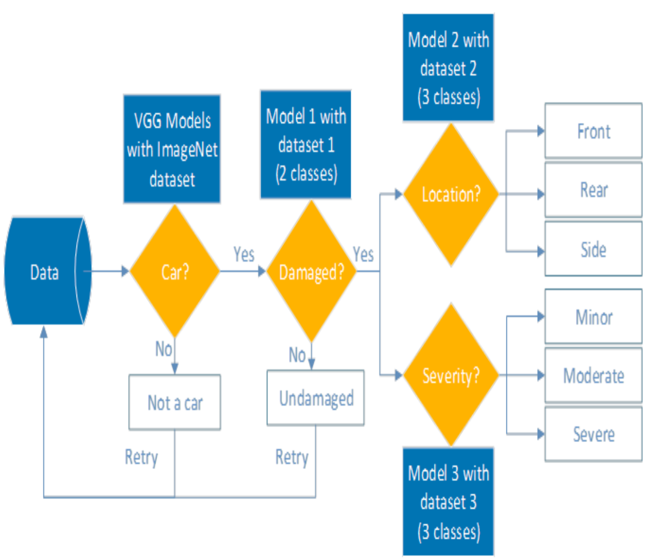

Figure 1: A flow chart of developing car damage assessment pipelines

Figure 1 shows the overall system of car damage assessment pipelines. There are four phases with three models, which are based on three datasets. Dataset 1consists of two classes to complete task 2; dataset 2 includes three classes to do task 3; dataset 3composes of three classes to perform task 4 in our system. In the first phase, we get input data from our datasets, and then we choose one of the pre-trained VGG models trained on an ImageNet dataset to recognize as a car or not a car. After choosing the model and testing it, we move on to the second phase. In this phase, we create model 1 and train it with our dataset 1 to determine an input data is a damaged or undamaged car. When we reach the third phase, we also train a model 2 with dataset 2 to detect the location of the damaged part of a car. When we arrive the final phase, a model 3 uses dataset 3 to assess the severity of the damaged part of a car. After all of those, we finish all tasks and accept some satisfying results.

3.1 Experimental Results

We use three different metrics: precision, recall, and F1-score to estimate the performance of our different transfer learning models such as VGG16 and VGG19. The higher those matrices are the best our model outperforms.

Table 1 shows the values of those different matrices in damage detection, damage location and damage severity in both VGG16 and VGG19 models. In the performance of damage detection, there call of 94% and 91% in both VGG16 and VGG19 means that they recognize those percentage of the damaged cars. We achieve the precision of 94%, 71% and 61% in damage detection, damage location and damage severity in VGG16 respectively. VGG19 get the precision of 91%, 71% and 59% in the performance of damage detection, location and severity. In Table 2, we explain about the overall accuracy of our models. We accept the accuracy of 95.22% and 94.56% in the damaged detection, the accuracy of 76.48% and 74.39% in damage localization, the accuracy of 57.89% and 54.80% in damage severity respectively by using both of transfer learning and regularization in VGG19 and VGG16 models. All performances accuracy of VGG19 are better than VGG16 even its values of matrices are not larger than VGG16. This means that sometimes the values of matrices are not reliable when there is a wrong prediction image. We observe that some data are difficult to predict. For example, the prediction of difference between a dent and scratch is very clear, whereas the damaged boundary between small and medium dent images is hard to define. We describe the prediction of validation data resulting from the three evaluations of the confusion matrices of VGG16 and VGG19, as shown in Figure 2 and 3. Each number represents the prediction number of validation data. For all three tasks, we use the respective models with L2 regularization with its value 0.001 because it overfits less when we use this value in our models. From their results, the performance of VGG19 is better than VGG16. All of the above, our pre-trained models not only detect damaged part but also assess its location and severity.

Table 1: Performance analysis of car damage assessment

	Performance of damage detection			Performance of damage location			Performance of damage severity		
Pre-trained VGG	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
VGG16	0.94	0.94	0.94	0.71	0.69	0.69	0.61	0.55	0.53
VGG19	0.91	0.91	0.91	0.71	0.66	0.66	0.59	0.54	0.51

Table 2: Accuracy of car damage assessment

	Performance of damage detection			Performance of damage location			Performance of damage severity		
Pre-trained VGG	**Witho ut L2**	**With L2**	**Fine-tuning**	**Without L2**	**With L2**	**Fine-tuning**	**Without L2**	**With L2**	**Fine-tuning**
VGG16	0.9456	0.9456	0.9283	0.7030	0.7439	0.7342	0.5338	0.5480	0.5268
VGG19	0.9457	0.9522	0.9086	0.7039	0.7648	0.7318	0.5731	0.5789	0.5614

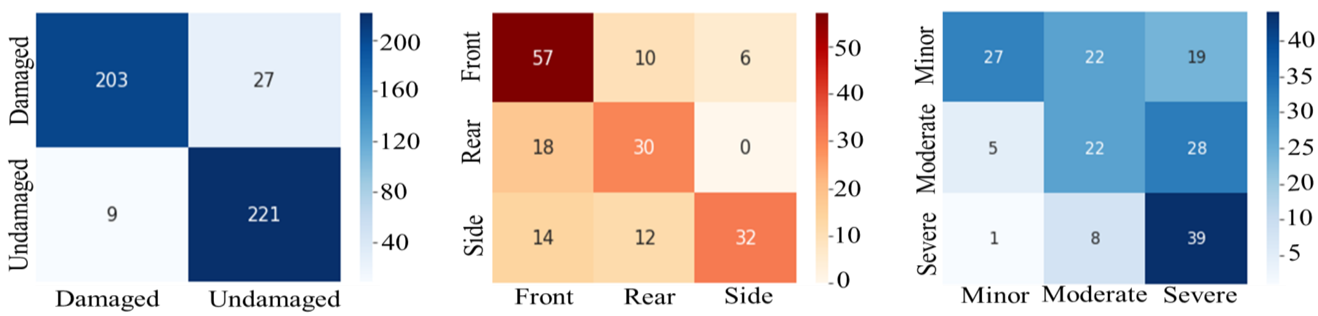

Figure 2: Confusion matrices for car damage assessment of VGG16

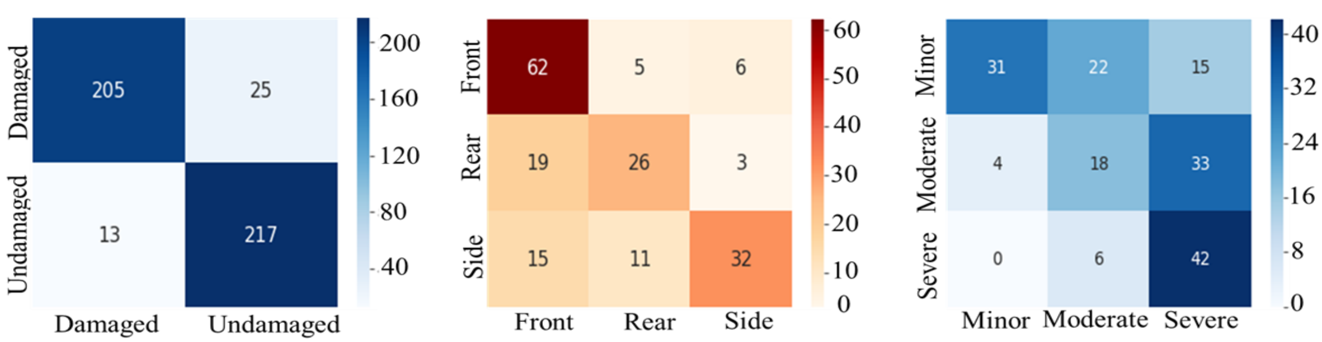

Figure 3: Confusion matrices for car damage assessment of VGG19

4 Conclusion and Future Work

We described applicable deep learning-based algorithms for car damage assessment. We created new datasets when there is no openly obtainable dataset for car damage classification. What is more, we experimented with the deep learning-based pre-trained VGG models from random initialization. Those models followed by supervised fine-tuning and transfer learning with L2 regularization technique to fit our specific tasks. We observed that training with a small dataset is not sufficient to get the best accuracy based on deep learning approach. In addition to this, it was not enough just using L2 regularization technique in our system. After analyzing our models, we find out that the results of using transfer learning and regularization can work better than those of fine-tuning. After that, the performances of VGG19 are better than VGG16. All of the above, our pre-trained VGG models not only detect damaged part of a car but also assess its location and severity.

Regarding our proposed models, we still face the overfitting problem in our models. Thus, in future work, we will utilize other types of regularization techniques and other pre-trained CNN models with a large dataset to fit that problem. If we have higher quality datasets, including the features of a car (make, model and the year of manufacture), location information, type of damaged part and repair cost, we can predict the cost of a car damaged part to be more reliable and accurate.

REFERENCES

[1] Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud. 2019. Extreme Gradient Boosting MachineLearning Algorithm For Safe Auto Insurance Operations. In2019 IEEE International Conference of Vehicular Electronicsand Safety (ICVES). IEEE, 1–5.

[2] Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud. 2019. A very deep transfer learning modelfor vehicle damage detection and localization. In2019 31st International Conference on Microelectronics (ICM). IEEE,158–161.

[3] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neuralnetworks?. InAdvances in neural information processing systems. 3320–3328.

[4] Mahavir Dwivedi, Malik Hashmat Shadab, SN Omkar, Edgar Bosco Monis, Bharat Khanna, and Satya Ranjan. [n.d.].Deep Learning Based Car Damage Classification and Detection. ([n. d.]

[5] Kalpesh Patil, Mandar Kulkarni, Anand Sriraman, and Shirish Karande. 2017. Deep learning based car damageclassification. In2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 50–54.

[6] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning.Journal ofBig Data6, 1 (2019), 60

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical imagedatabase. In2009 IEEE conference on computer vision and pattern recognition. IEEE, 248–255.

[8] Jeffrey de Deijn. 2018. Automatic Car Damage Recognition using Convolutional Neural Networks. (2018).

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition. 770–778.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neuralnetworks. InAdvances in neural information processing systems. 1097–1105.

[11] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition.arXiv preprint arXiv:1409.1556(2014).

[12] Srikanth Tammina. [n.d.]. Transfer learning using VGG-16 with Deep Convolutional Neural Network for ClassifyingImages. ([n. d.]).

[13] Syukron Abu Ishaq Alfarozi, Kitsuchart Pasupa, Hiromichi Hashizume, Kuntpong Woraratpanya, and MasanoriSugimoto. 2019. Robust and Unified VLC Decoding System for Square Wave Quadrature Amplitude Modulation UsingDeep Learning Approach.IEEE Access7 (2019), 163262–163276.

[14] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning.IEEE Transactions on knowledge and dataengineering22, 10 (2009), 1345–1359.

[15] Ranjodh Singh, Meghna P Ayyar, Tata Sri Pavan, Sandeep Gosain, and Rajiv Ratn Shah. 2019. Automating Car InsuranceClaims Using Deep Learning Techniques. In2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM).IEEE, 199–207.

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

[17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time objectdetection. InProceedings of the IEEE conference on computer vision and pattern recognition. 779–788.

[18] WA Rukshala Harshani and Kaneeka Vidanage. 2017. Image processing based severity and cost prediction of damagesin the vehicle body: A computational intelligence approach. In2017 National Information Technology Conference (NITC).IEEE, 18–21.

[19] Andrej Karpathy. 2017. Course notes of convolutional neural networks for visual recognition.