Car damage detection

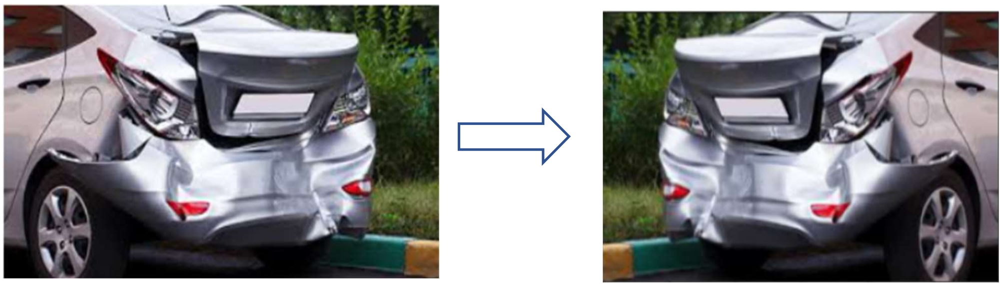
Detecting the value of a car before selling is very crucial as it is important for both parties involved in purchasing and selling of a car. Therefore, deep learning can come in handy when detecting the price reduction by capturing the level of damage a car has gone through. In this report, the aim focuses on detecting the external damage of a car and mainly highlights the pre-processing techniques used when detecting the damage of a car.

Pre-processing techniques

In terms of data, the data required for detecting damages in a car is different from usual machine learning pre-processing. The data for car damage detection could be trained only using images and therefore there must be images of different type of damages in cars. This process is to train the model to be able to identify the damage of a certain car when the picture is uploaded for detection. The different type of damages includes and not limited to damage areas such as front damage, rear damage, side damage, tail lamp damage, head lamp damage, glass shatter,

Although Deep learning is an efficient approach to detect damages, the resources for such model development is limited. For the purpose of building and training a car damage detection model, there needs to be a huge dataset with images of damages in cars as mentioned earlier and this would take up more computational time as the dataset gets bigger. One could think that it is easy to add images of cars and make it a huge dataset. However, truth is that car damages detection is a specific domain and there are not many publicly available datasets with images and labelling.

A model that is widely used is convolutional neural network, CNN but small dataset would not be sufficient to train a CNN model. Therefore, one way to process this limitation is by applying data augmentation to the collected images. Data augmentation can be done to synthetically enlarge the dataset by applying random flips to the images in the range of -20 and 20 degrees. Besides that, including different range of zoom to the images can help to create more images. Image below shows an example of data augmentation.



We could manually increase the number of images in such way and add a label to the images to make it easy for the CNN training later. However, as mentioned earlier, the computational time plays an important role in building this model as it can take up to days and even weeks to run and complete. Therefore, it is essential to reduce the running time of a model to allow it to predict and be able to contribute to the production as quick as possible.

A method to reduce the time and get best performance is by resizing the images. Resizing is to help and downscale or upscale the images to match dimensions of the smallest image available in the dataset. This would help tremendously when running the model. Another effective way to reduce running time is by applying transfer learning. This basically means using pre-trained CNN models which was trained previously with a larger dataset. By transfer learning, the weights and architectures of the pre-trained model can be applied for other specific tasks. However, the features when applying CNN to a specific task should be similar or else it will require a rebuild from scratch. 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 [1]. When the source and target data are similar, transfer learning helps to improve the performance.

Trade-offs between models

Previous section discussed about some pre-processing techniques to collect and optimize a dataset in order to build a car damage detection model and ways to reduce time when producing such model. This section focuses on comparison of some CNN models that could help in damage detection. CNN is an image classification model that detects object by using bounding boxes. However, CNN has its limitation when it has multiple objects in a picture because CNN detects one object at a time using bounding boxes and this could be an issue if a picture uploaded has more than one object.

This is where region based CNN comes in handy to help and detect the object in an image. A typical CNN can only tell you the class of the objects but not where they are located. It is actually possible to regress bounding boxes directly from a CNN but that can only happen for one object at a time.

In R-CNN the CNN is forced to focus on a single region at a time because that way interference is minimized because it is expected that only a single object of interest will dominate in a given region. The regions in the R-CNN are detected by selective search algorithm followed by resizing so that the regions are of equal size before they are fed to a CNN for classification and bounding box regression. Although R-CNN is a good model, the challenge is that R-CNN is slow when training. It extracts 2000 regions for every image based on selective search. The processing time is unimaginable in this case if the dataset is huge. This leads to Fast R-CNN where object classification and object detection is done with deep convnets. Unlike R-CNN, Fast R-CNN uses a single deep ConvNet to extract features for the entire image once. That sums up the difference between R-CNN and Fast R-CNN where one uses 2000 convnets and the later uses only a single deep convnet. Besides that, Fast R-CNN uses multi-task loss to achieve an end to end training of Deep ConvNets which increases the detection accuracy. Although the time can be reduced by using Fast R-CNN, it can still be time consuming when dataset is large. That led to Faster R-CNN that does not use selective search like selective search but uses region proposal network. Selective search is where images are grouped based on region size and region proposal network takes an image of any size as input and outputs a set of rectangular object proposals each with an object score.

Mask R-CNN on the other hand is a model that does instance segmentation. It is able to highlight objects in the image regardless of whether the objects are similar. Mask R-CNN also adds colour pixels in the bounding box of the specific class which is an added advantage when compared to Faster R-CNN. Mask R-CNN use Fully Convolutional Network which is a popular algorithm for doing semantic segmentation. This model uses various blocks of convolution and max pool layers to first decompress an image to 1/32th of its original size. It then makes a class prediction at this level of granularity.

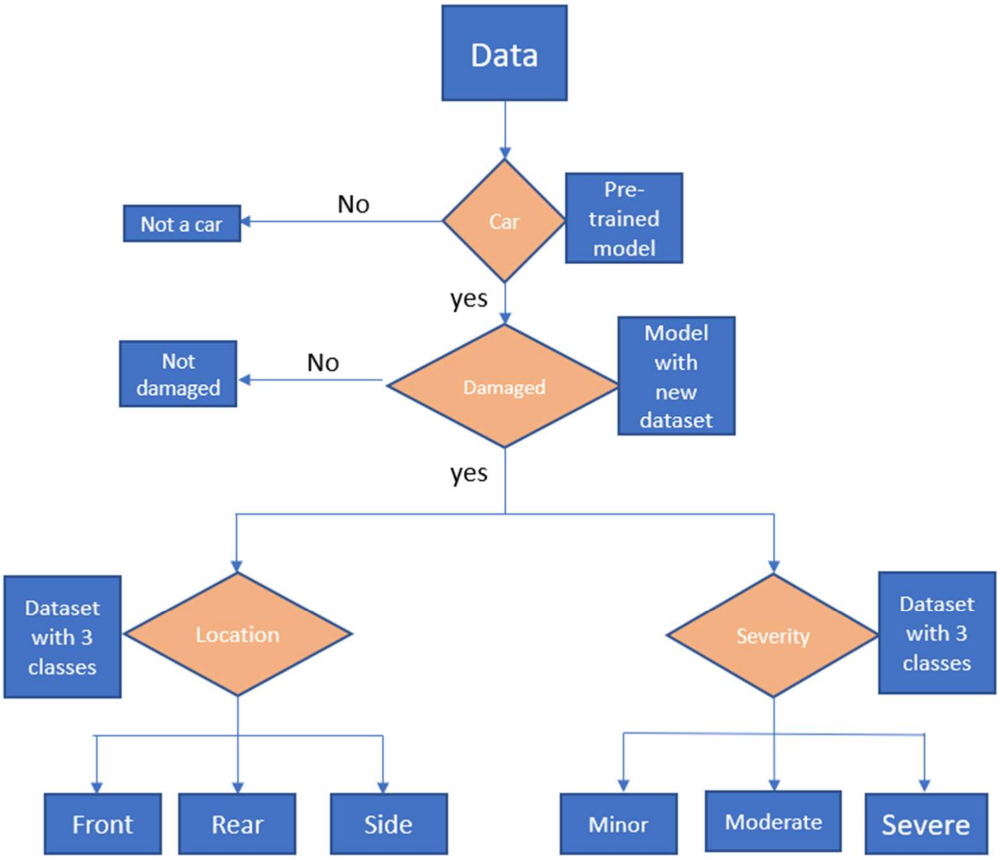
YOLO (you only look once) is the final model to be discussed in this section. YOLO is a real-time object detection system that applies a single neural network to the full image. This network divides the image into regions and predicts the bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

One of the advantages of YOLO is that it looks at the whole image during the test time, so its predictions are informed by global context in the image. Unlike R-CNN, which requires thousands of networks for a single image, YOLO makes predictions with a single network. This makes this algorithm extremely fast when compared to other models above. YOLO is popular because it achieves high accuracy while also being able to run in real-time. The algorithm “only looks once” at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes

Models	**Trade-offs**
CNN	Detection of a single object in an image
R-CNN	Image classification as well as localization for multiple objects in an image.
Fast R-CNN	Faster than R-CNN and efficient but still time consuming when dataset is large since it uses selective search algo
Faster R-CNN	More Efficient than Fast R-CNN as it is using RPN
Mask R-CNN	Combination of Faster R-CNN and Fully Convolutional Network
YOLO	Fast and reliable as it that it looks at the whole image during the test time

In summary of the above models, Mask R-CNN and YOLO would be two of the best models that can be used when detecting damages in car. Besides that, applying transfer learning together with one of these models would help reduce time and give good output as well as accuracy.

Flow Chart



Flow chart above would be one possible way to detect cars that have damage and judge the severity of the condition. This model could serve the business goal to sell cars with proper price based on the output of it.

References

1. https://www.analyticsvidhya.com/blog/2018/07/building-mask-r-cnn-model-detecting-damage-cars-python/
2. https://towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-1d141eba6e04
3. https://www.analyticsvidhya.com/blog/2018/07/building-mask-r-cnn-model-detecting-damage-cars-python/
4. Kalpesh Patil, Mandar Kulkarni and Shirish Karanade: DEEP LEARNING BASED CAR DAMAGE CLASSIFICATION, pp 1-7.
5. https://medium.com/analytics-vidhya/car-damage-classification-using-deep-learning-d29fa1e9a520
6. https://medium.com/ai-techsystems/detecting-car-damage-using-deep-learning-781ffc643414
7. https://algoanalytics.com/vehicleDamage.html