# Kaggle Carvana Image Masking Challenge

Introduction:

The objective of this Kaggle problem is to automatically identify the boundaries of the car in an image given labelled input and output training images.

Performance indicator:

This performance of the masking algorithm is evaluated on the mean [Dice coefficient](https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient) which can be used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The formula is given by:

where X is the predicted set of pixels and Y is the ground truth. The value of D lies between 0 and 1 indicating none and perfect match respectively.

Solution strategy:

Image masking is a semantic segmentation problem which describes the process of associating each pixel of an image with a certain class label. Keeping in view the recent developments in Computer Vision and the use of deep learning models to solve similar problems, a Convolutional Neural Network design would be a reasonably good choice to train a model that can predict the mask of the test objects. To kick-start the implementation, a fully convolutional CNN upsample-downsample architecture is proposed. The details are given in the following section. The solution is implemented on the Kaggle kernel using python. The Deep learning model is set up using Keras library.

Methodology:

The implemented model consists of a battery of Convolutional and Maxpooling layers for the first downsampling region followed by a series of Upsampling and Convolutional layers. The model architecture is summarized in figure 1.

The model therefore consists of around 3 million hyperparameters. The choice of the network depth is somewhat arbitrary governed mostly by the maximum resolution of the image and the number of hyperparameters. The Maxpooling filter size was kept small to preserve localization of features. During one of the testing phases, it was seen that the usage of larger filter always yielded poorer results. Therefore a 2 by 2 layer is implemented in the final model. For convolutional layer, not much experimentation was done primarily due to shortage in time. A fully connected layer is avoided here as the benefits of adding them can be achieved by deeper convolutional layers.

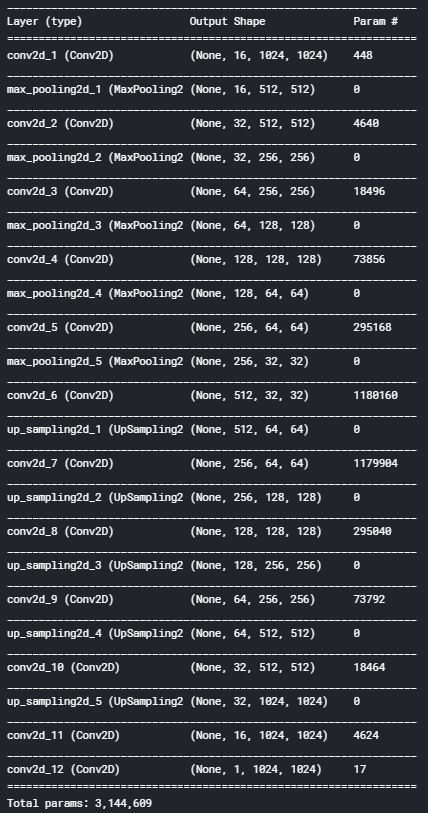


Figure 1: Network architecture

Results:

The results of the model are visualized for the training, validation and test samples.

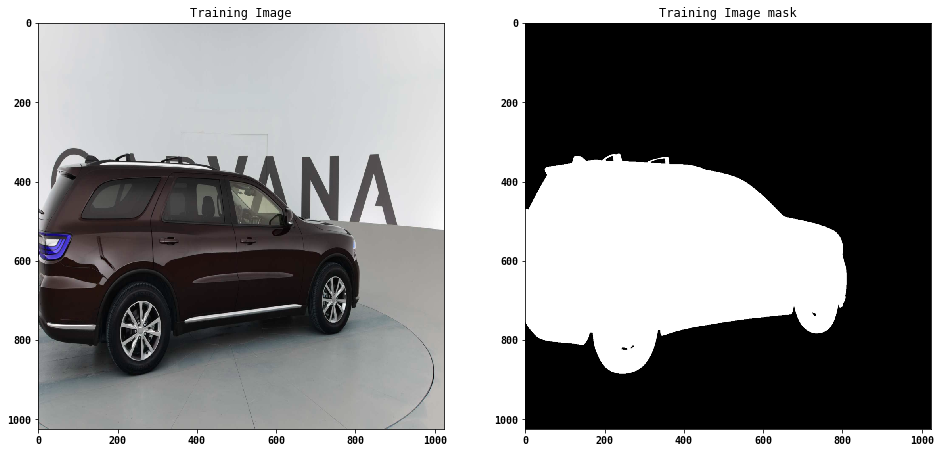


Figure 2: Training image and its corresponding mask

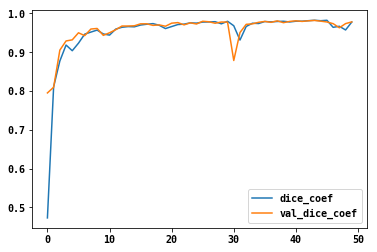


Figure 3:Train and validation dice coefficients with epochs



Figure 4: Validation and test samples

Discussion:

Following observations were made regarding the results obtained from the model:

* The validation and the training performance appear to be highly correlated
* The performance parameter ‘dice coefficient’ is quite sensitive to the perceived notion of a ‘good mask’
* Model was not learning better even after long epoch times possibly due to:
  + Model hyperparameters may be not right
  + The model architecture could be inadequate to reach the desired performance
    - For ex: Upsampling- could be an issue as ‘Keras’ does not allow ‘Max unpooling’ layer which would have been suitable to match the Upsampling unpooling to match in pixel location corresponding to the pooling operation.
  + The model may not have been fully trained due to several unknown reasons
* Regularization schemes like dropout is not included as the validation error does not seem to decrease on improved training error
* Validation error seems at certain times is smaller than training error- possibly due to small batch size. Larger batches however require more training time.

Ideas to improve the accuracy of the model:

A few ideas are proposed that could potentially lead to a better solution

1. Advanced architectures like R-CNN/mask may be an overkill for this binary image segmentation
2. Using the features of the car models to sample train sets and even add information into the DL model
3. Using a preprocessing stage can be employed to simplify the network, resulting in a more robust model.

Final score of the challenge

To be updated soon