Vehicle Image masking Challenge

A. Jeevithan(C00289092)

***Abstract*— Masking the Vehicle image for given image is well known problem. In this work we have been asked to mask the vehicle segment from the given humongous dataset. Then the U-net architecture was used to train the model. For back propagation binary cross entropy is used.**

***Keywords*— Convolution Neural Network(CNN), and Dice Similarity Coefficient(DSC)**

I. Introduction

As with any big purchase full information and transparency are key. While most everyone describes buying a used car as frustrating,its just as annoying to sell one,especially on line. Shoppers want to know everything about the car but they must rely on often blurry pictures and little information,keeping used car sales a largely inefficient, local industry.[1]

"Carvana", a successful on line used car startup, has seen opportunity to build long term trust with consumers and streamline the on line buying process.

An interesting part of their innovation is a custom rotating photo studio that automatically captures and process 16 standard images each vehicle in their inventory. While carvana takes high quality photos bright refelctions and cars with similar colors as the background causes automation errors, which requires a skilled photo editor to change.

Here the challenge was removes the photo studio background.

The proposed methodology to solve this problem is using UNET neural network architecture to mask the vehicle through image segmentation.

II. Dataset

A. Dataset

Data set contains a large number of car images(as.jpg file).Each car has exactly 16 images, each one taken at different angles.Each car has a unique id and images are named acording to id_01.jpgFor the training set, you are provided a.gif file that contains the manually cutout mask for each image.The task is to automatically segment the cars in the images in the test set.

.

In that they have given 5088 images with dimension (1280 x 1918 pixels) training set. There is an example of one training set image and the mask for that image is shown in Fig.1 and

Fig.2 respectively. Those are given by under the name of "**/train/".**

They also provide test set images under the name "**/test/**". Those images are only high quality images, there were no mask images provided.

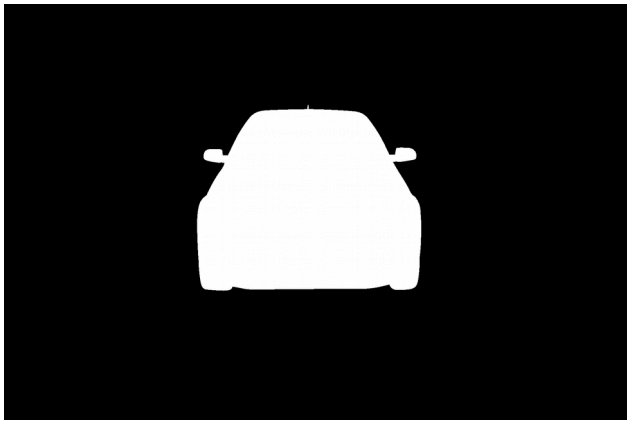
For our approach from the train set 432 images are used for testing purposes. The test data is given without mask for prediction for the trained network.

B. *Difficulties*

Training image is quit big to fit and train in GPU so I had to processes the image. I resized the the entire image set to 128 x 128 which is perfectly fitted to my GPU.

III. Architecture And Method

A. Architecture



architecture[2] is well known architecture for the medical images. In this work I implement the same architecture.

In my work I had to implement Convolution Neural Network(CNN), We have gone through the various CNN architecture for my project and I found that “U-net

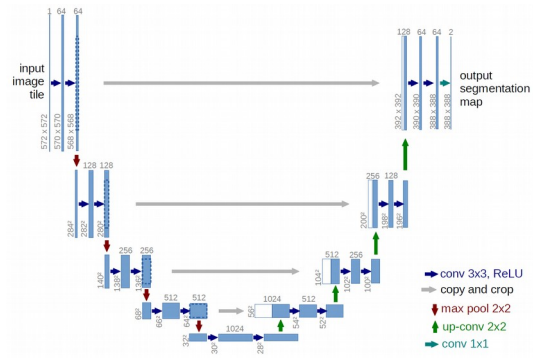


Fig 3: U-net architecture

- Convolution kernel size is 3 x 3
- Stride size (2 x 2)
- Activation function is relu
- binary cross entropy for loss.

A. *Accuracy.*

For the back propagation Dice Similarity Coefficient(DSC) used as loss function.

The Dice Similarity coefficient is often used to quantify the performance of image segmentation methods[3]. Where you annotate some ground truth region in your image and then make an automated algorithm to do it. You validate the algorithm by calculating the DSC, which is a measure of how similar the objects are. So it is the size of the overlap of the two segmentations divided by the total size of the two objects. Using the same terms as describing accuracy, the Dice score is:

$$\text{Dice\_coefficient}(A,B) = \frac{2|AB|}{|A|+|B|} \qquad (1)$$

B. *Method*

Data set has contains ~5000 training images with their masks (labels). Each image is 1280x1918 is quit big it didn’t fit in to memory then I resized all the images to 128x128 which is the size we could able to train the model without any hindrance.

I used pixel wise prediction in our research because regardless of size of the car and position(Translation invariant) of the vehicle we need to predict the nerve segmentation.

To use our memory effectively we used Stochastic Gradient Decent (SGD) which is a method helps to train the model as mini batches.

If I use small learning rate (alpha)There are more possibilities to get stuck in a global minimum when I try minimize the loss function or if you use high learning rate model won’t converge ,to avoid this scenario I used decaying learning rate (beginning it will be high and with epoch increases it will decay).

Literally I need to mask vehicles which is not shown yet to the model, so my model should be able to predict rather than look up. To do that our model shouldn't be over fitted, if that so rather than predict model will try to look up. If you feed new images in to the model it may be fail to predict. To avoid that, we used drop out in each layer. Drop out basically mutes the weights randomly for representations. And also we used validation test when we training the image, validation test would tell the capability of the model in predicting new images with optimal generalization.

- In our work we used stochastic gradient decent(mini batch size=5) because to fit our data in memory.
- In our work we used Adam optimizer as gradient decent optimizer.
- We made 20% of our training data as validation data.
- To avoid over fitting we added 20% dropout at each layer.
- In our work we trained the data for 4000 epoch.

IV. Results

The network loss of the trained network as mentioned in methodology is shown in Fig.4. And the accuracy for training set and the validation set is shown in Fig.5.

Some of the test images with predicted images are shown in Fig.6.

In the Table 1, the pixel wise performance of the network classifier is given, there

TP: Annotated as vehicles pixel is predicted as vehicle pixel (both normalized value of the pixel equal to 1)

TN: Annotated as no vehicles pixel is predicted as no nerve pixel (both normalized value of the pixel equal to 0)

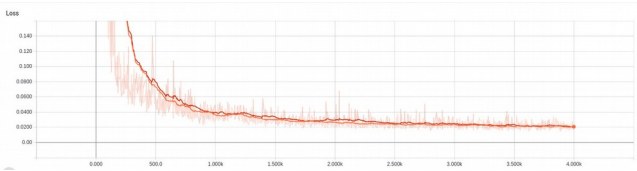
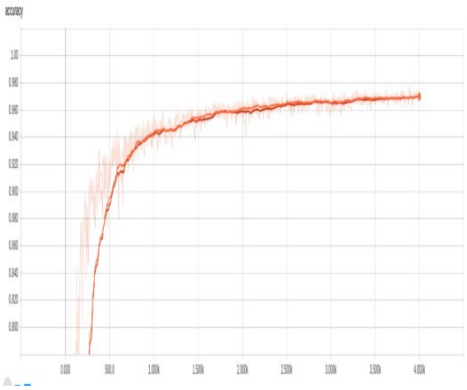


Fig.4. Loss respect to epoch



FN: Annotated as vehicels pixel is predicted as no vehicle pixel (in normalized scale annotated as 1 predicted as 0)

FP: Annotated as no vehicles pixel is predicted as vehicle pixel (in normalized scale annotated as 0 predicted as 1)

V. Problem Encountered

When I first implemented I got erroneous prediction and there were no improvement after it reached to 60 % accuracy , then I changed the upsampling method it worked and training accuracy improved.

VI. Conclusion

Then we can train the architecture by implementing

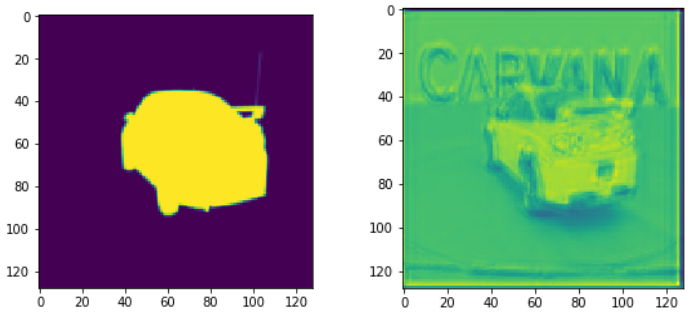
- L2 regularization
- changing kernel sizes as 5 x 5, 4 x 4
- Adding 1 x 1 convolution layer
- Changing the loss function.

Even further changing batch size and the epoch to check the model behavior.

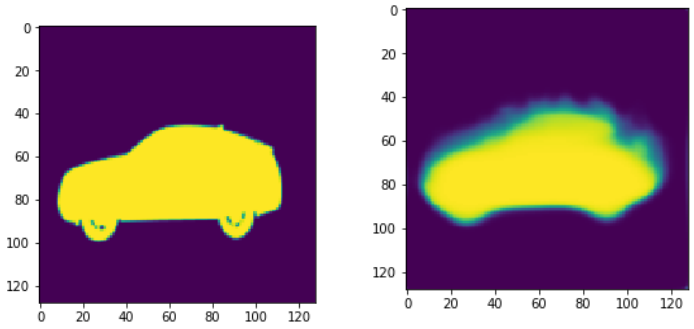
Table.1: Architecture 's pixel wise performance of the classifier

Dicecoefficient(Avg)	*Accuracy (avg)*	*Precision*	*NPV*	*TNR*	*TPR*
91.54	0.958	0.84	0.95	0.99	0.99

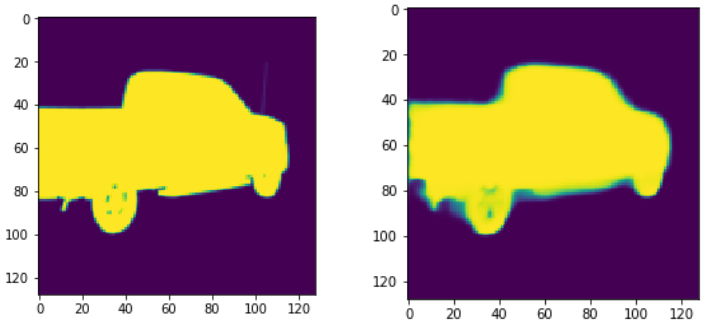
Training Visualization with step progress



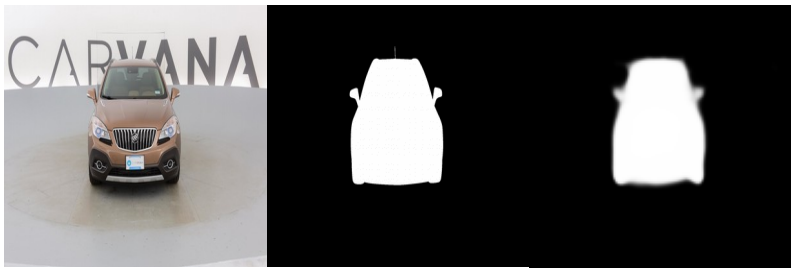
At step 100 original mask and Predicted mask



At step 300 original mask and Predicted mask



At step 3000 original mask and Predicted mask



Original image Original mask Predicted mask

Predicted mask Original image Original mask

Reference

Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *arXiv preprint arXiv:1606.00915* (2016).

Li, Bing, and Scott T. Acton. "Active contour external force using vector field convolution for image segmentation." *IEEE transactions on image processing* 16.8 (2007): 2096-2106.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, Cham, 2015.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." *arXiv preprint arXiv:1511.00561* (2015).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.