Abstract

A U-Net model proposed by Ronneberger et al. which resembles an autoencoder but with convolutions instead of a fully connected layer was used to solve Carvana's image masking competition in Kaggle. The U-Net was trained only with data set provided and without any pretrained layers because it is a binary segmentation model.

Background

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

An interesting part of their innovation is a custom rotating photo studio that automatically captures and processes 16 standard images of each vehicle in their inventory. While Carvana takes high quality photos, bright reflections and cars with similar colors as the background cause automation errors, which requires a skilled photo editor to change[1].

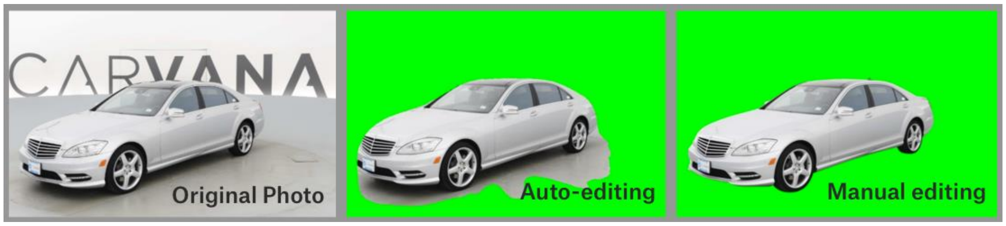


Figure 1: Background removing process of Carvana

We are challenged to develop an algorithm that automatically removes the photo studio background. This will allow Carvana to superimpose cars on a variety of backgrounds. A dataset of photos which cover different vehicles with a wide variety of year, make, and model combinations.

Proposed Method

- **Proposed Model Architecture**

The U-Net model proposed by Ronneberger et al. was chosen.

U-Net model resembles an autoencoder but with convolutions instead of a fully connected layer. It improves upon the "fully convolutional" architecture primarily through expanding the capacity of the decoder module of the network[3].

There is an encoding part with the convolution of decreasing dimension and decoder part with increasing dimension as shown in figure 2.

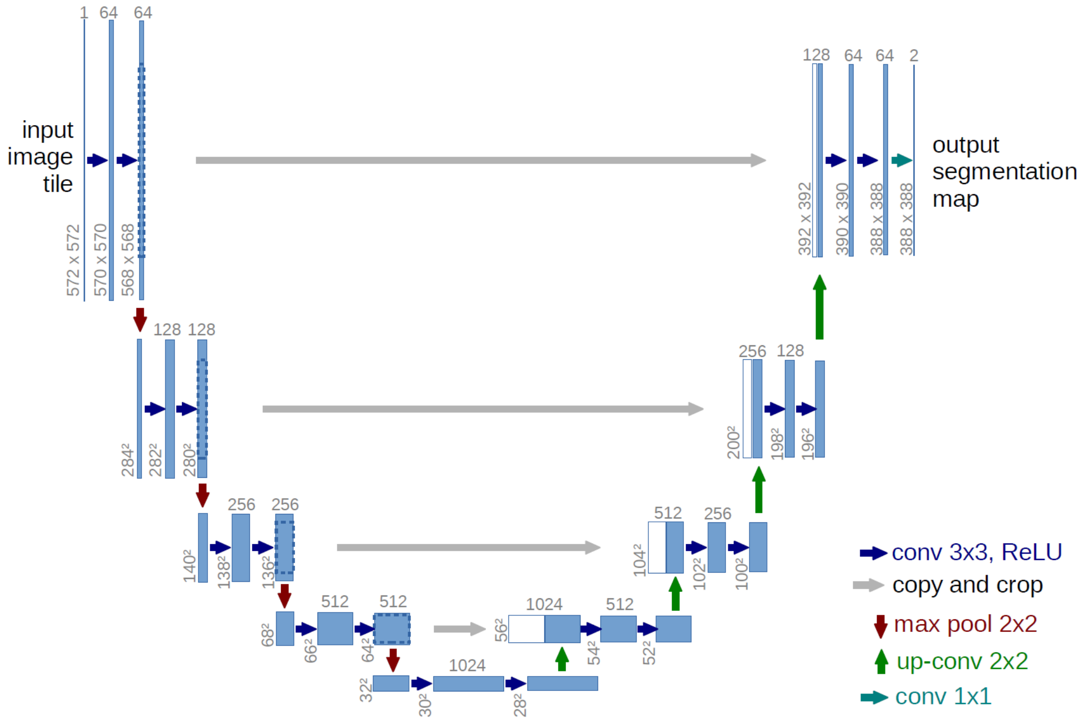


Figure 2: U-Net architecture proposed by Ronneberger et al.

- **Up-sampling Method**

A transpose convolution allows for us to develop a learned up-sampling. We take a single value from the low-resolution feature map and multiply all the weights in our filter by this value, projecting those weighted values into the output feature map as shown in figure 3.

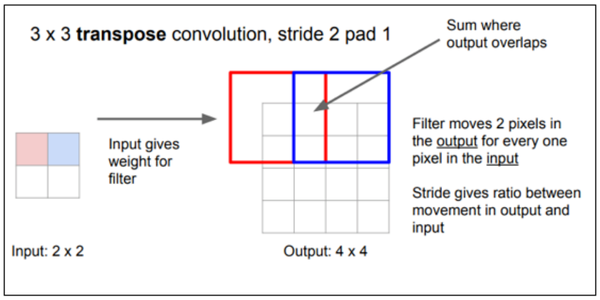


Figure 3: transpose convolution method

- **Performance matrix**

The performance matrix is a mean Dice coefficient or f1 score. The Dice coefficient can be used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth[2].

The formula is given by:

$$\frac{2\sum_{pixels} y_{true} y_{pred}}{\sum_{pixels} y_{true}^2 + \sum_{pixels} y_{pred}^2}$$

where y_{pred} is the predicted set of pixels and y_{true} is the ground truth.

- **Loss Function**

  The used loss functions are cross entropy loss and dice loss.

  Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.

  The formula of cross-entropy loss is given by:

$$\sum_{pixels} -(y_{true} \log(y_{pred}) + \left(1 - y_{true}\right) log(1 - y_{pred}))$$

  The loss function is a soft Dice loss which can be minimized because we directly use the predicted probabilities instead of thresholding and converting them into a binary mask[2].

  The formula of dice loss is given by:

$$1 - \frac{2\sum_{pixels} y_{true} y_{pred}}{\sum_{pixels} y_{true}^2 + \sum_{pixels} y_{pred}^2}$$

  With respect to the neural network output, the numerator is concerned with the common activations between our prediction and target mask, whereas the denominator is concerned with the quantity of activations in each mask separately.

Experiments

1. Data Preprocessing

a. Resizing

All the image was resized to 512*512.

b. Augmentation

It was used to increase training data size and it has regularization effect.

- horizontal flipping,
- image shifting,
- color shifting

c. Normalization

The image pixel was divided by 255 for normalization. This is because https://stackoverflow.com/questions/20486700/why-we-always-divide-rgb-values-by-255

2. Model architecture

a. Model summary

Table 1: Model summary

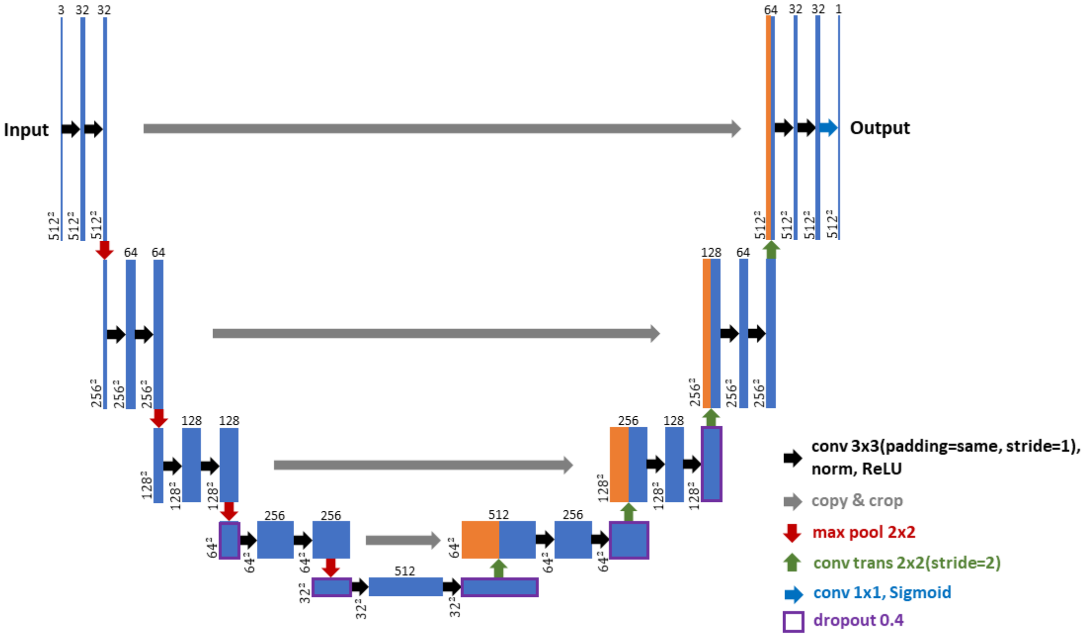


Figure 4 shows the layer details of modified U-Net.

b. Optimizer

Adaptive Moment Estimation(Adam) which is a combination of gradient descent with momentum & RMSProp with learning rate decay.

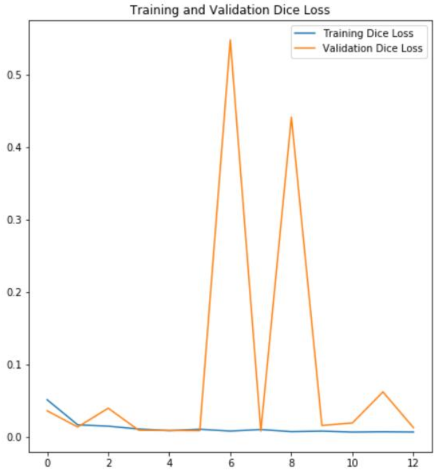
c. Loss

Sum of binary cross entropy loss and soft dice loss.

3. Results

The model got the best validation loss at 8th epoch with training loss of 0.0198 and validation loss of 0.0148.

The model showed a promising segmentation result as shown in figure 5 & 6.



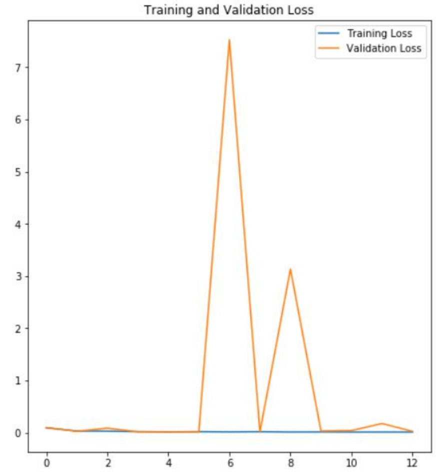


Figure 5: History of training & validation loss

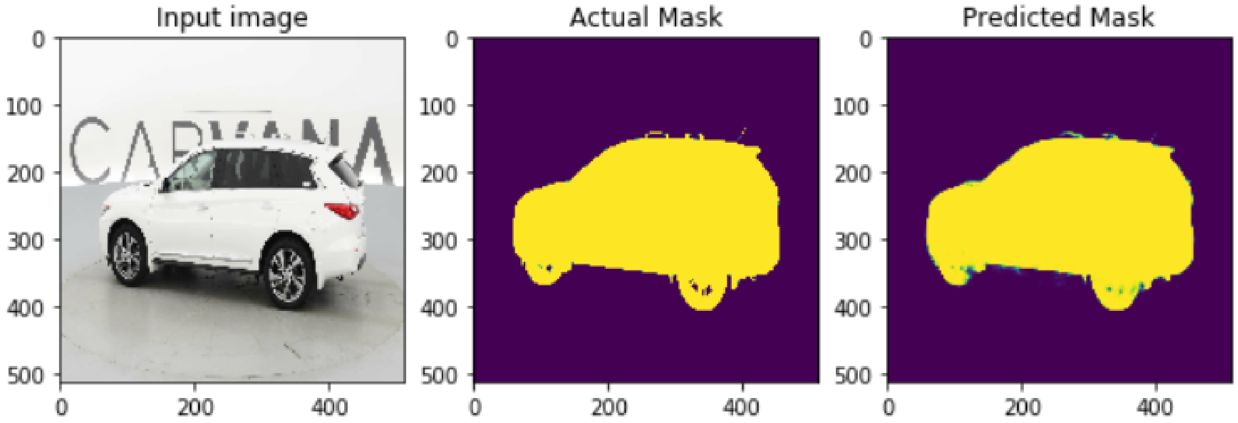


Figure 6: Result of actual mask vs predicted mask of training

Conclusion

The U-Net model showed a promising segmentation result. Further improvement step might be needed to get better result. The improvement steps are hyperparameter tuning and error analysis.

References

1. Carvana Image Masking Challenge. Retrieved from https://www.kaggle.com/c/carvana-image-masking-challenge
2. Jordan, J. (2018, May 21). An overview of semantic image segmentation. Retrieved from https://www.jeremyjordan.me/semantic-segmentation/
3. Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

Appendix

- **Version control**

Library	**Version**
Python	3.6.6
TensorFlow	1.10
Skimage	0.15
Numpy	1.15.1

- **Github**
  https://github.com/ChuaHanChong/kaggle_carvana_image_masking_challenge