A PROJECT REPORT ON

Image Segmentation on Vehicles

As part of the Capstone Project

Submitted to Udacity in partial fulfillment of the

Machine Learning Engineer Nanodegree

Contents

Abstract

In the field of computer vision, segmentation tasks are the earliest and a field of extensive research. Based on the requirements, these can be basic object classification to deciphering multiple classes in complex environments like in the Cityscapes dataset. In this report, we go over using a basic U-Net architecture for the semantic segmentation task. We then use pretrained custom model for this task provided in AWS SageMaker and train over our dataset to achieve accurate vehicular segmentation of the Carvana dataset. Based on performance, the Pyramid Scene Parsing Network (PSP Net) architecture is chosen with an accuracy of 99.2% is selected for our model compared to 97.8% accuracy when using the Fully Convolutional Network

Keywords: Fully Convolutional Network, Pyramid Scene Parsing Network, Computer Vision, Semantic Segmentation

Introduction

As the name suggests, the goal of the semantic image segmentation is to label elements in the image to their corresponding class. Under the hood, the modeling technique labels each pixel of the image with a corresponding class of what is being represented. A more complex objective is to carry out image segmentation, wherein not only are the pixels classified, but also to distinguish various objects related to the same class.

Segmentation of the images have various applications such as detecting road signs (Maldonado-Bascon et al. 2007), Advanced Driver Assistance Systems (ADAS) or self-driving car in transportation industry. In medical industry, it is being used to detect brains and tumors (Moon et al. 2002), and detecting and tracking medical instruments in operations (Wei et al. 1997), colon crypts segmentation (Cohen et al. 2015), etc. There are many other uses of it, like in video surveillance, land use and land cover classification (Huang et al. 2002), object detection tasks (Delmerico et al. 2011), etc.

Approaches in image semantic segmentation include unsupervised and supervised approaches. In unsupervised setting, we utilize K-means clustering, SVM etc. to help annotating dataset for prediction. Presently, we are provided with the labelled dataset, and there are plenty of approaches for this as we discuss below.

Before the discovery of Convolutional Neural Networks (CNNs), research in computer vision area mainly relied on extracting different features within an image. There are Variety of feature extraction methods for semantic segmentation. For example, there are several methods dealing in corner detection Shi-Tomasi (Shi et al. 1994), Harris Corners (Derpanis 2004), Adaptive and Generic corner detection based on the Accelerated Segment Test AGAST (Mair et al. 2010) and Multiscale AGAST (Leutenegger et al. 2011) Detector, Features from Accelerated Segment Test (FAST) (Rosten and Drummond 2005) and its Enhanced Repeatability modification FAST- ER (Rosten et al. 2010). Then there are features to utilize pixel and brightness like Similar brightness in Univalue Segment Assimilating Nucleus (SUSAN) (Smith and Brady 1997), Pixel color, Sub-pixel Corner (Medioni and Yasumoto 1987). Then there are techniques using Local features like Scale-Invariant Feature Transform (SIFT) (Lowe 2004), Local Binary Pattern (LBP) (He and Wang 1990), Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005; Bourdev et al. 2010), etc. Lastly, there are some uniques techniques like Speeded-up robust features (SURF) (Bay et al. 2008),Bag-Of-Visual-words (BOV) (Csurka et al. 2004) that detect words and make patches on image, Poselets (Brox et al. 2011) for pose detection, and Textons (Zhu et al. 2005) to identify basic image shape.

The first application of using deep learning for semantic segmentation was using CNN, which took advantage of identifying various patterns from pixel levels all the way upto the bigger image elements. The basic architecture applies convolutions and pools the data to learn low-level patterns while increasing the number of channels to increase the expressiveness of the model. The downsizing of the image due to pooling poses the needs for resizing the image to original dimensions, which is handled in 2 ways. First is the upsampling, wherein we upsample using bilinear calculations, or simply increase the size in respective directions keeping the same values without any learning parameters. The second approach is to apply transpose convolutions, which applies the kernel to adapt to the images being presented. Since the kernel has learnable parameters, it can further help in preserving the image features since the vector operations by upsampling leads to smoothing of the output image provided. This entire network

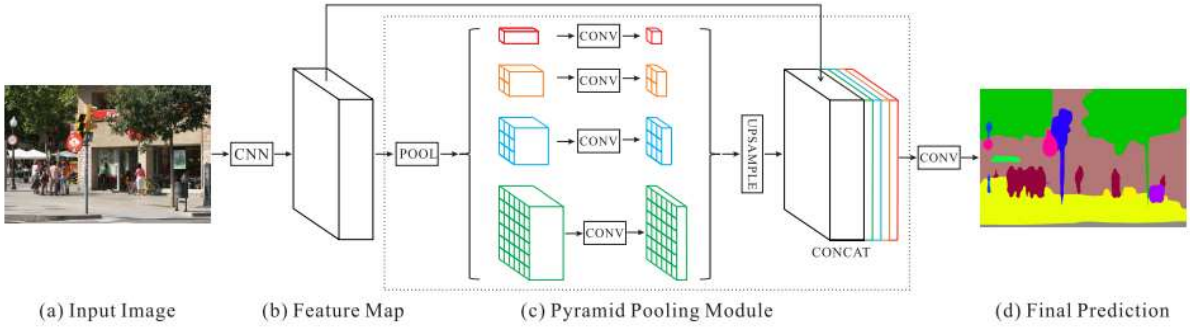


Figure 1: Architecture of PSP Net which has ability to utilize scene context for predicting

is called as an auto-encoder, wherein the convolutional layers are referred as an encoder while the deconvolution of these are called as the decoder.

The Fully Convolutional Network (FCN) applies the above method as well as augmenting the deconvolution layers by adding the channels of convolution layer with corresponding size known as skip connections (Long et al. 2016). These skip connections aid in boosting the network performance while preserving the image vocabulary for the model. Most of today's State-of-the-art scene parsing frameworks utilize the fully convolutional network (FCN) owing to their capability of dynamic object understanding in the encoding-decoding phase. For predictions, often a combination of various number of layers in the 2 phases are tested to arrive at the best architecture for a specific use case. However, they often face the issue of discombobulating image classes with similar structure/shape.

This led to the conception of Pyramid Scene Parsing Network (PSP Net) as seen in Figure 1, which tries to determine the image class by taking scene's context into account (Zhao et al. 2016).

Methods and Discussions

The training for the U-Net was carried out on GPU provided by Google Colab. For the analysis of SageMaker's modules, compute was carried out on AWS SageMaker's "ml.p2.xlarge" instances as it's the least compute architecture need to train an image based model. Keeping this in mind, the number of epochs was selected to be 5 for training both the methods.

Dataset

The dataset is provided by a start-up called Carvana. The need for image segmentation here is that even though 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 (refer to Annexure 1 for details). The objective is to develop an algorithm that automatically removes the photo studio background instead of relying on manual editing. This will allow Carvana to superimpose cars on a variety of backgrounds. Detailed overview as well as the dataset is available at: https://www.kaggle.com/c/carvana-image-masking-challenge

As mentioned in the proposal, the dataset provided is quite large since it contains images. For present evaluation and to monitor our metrics, we'll utilize the training images and annotations

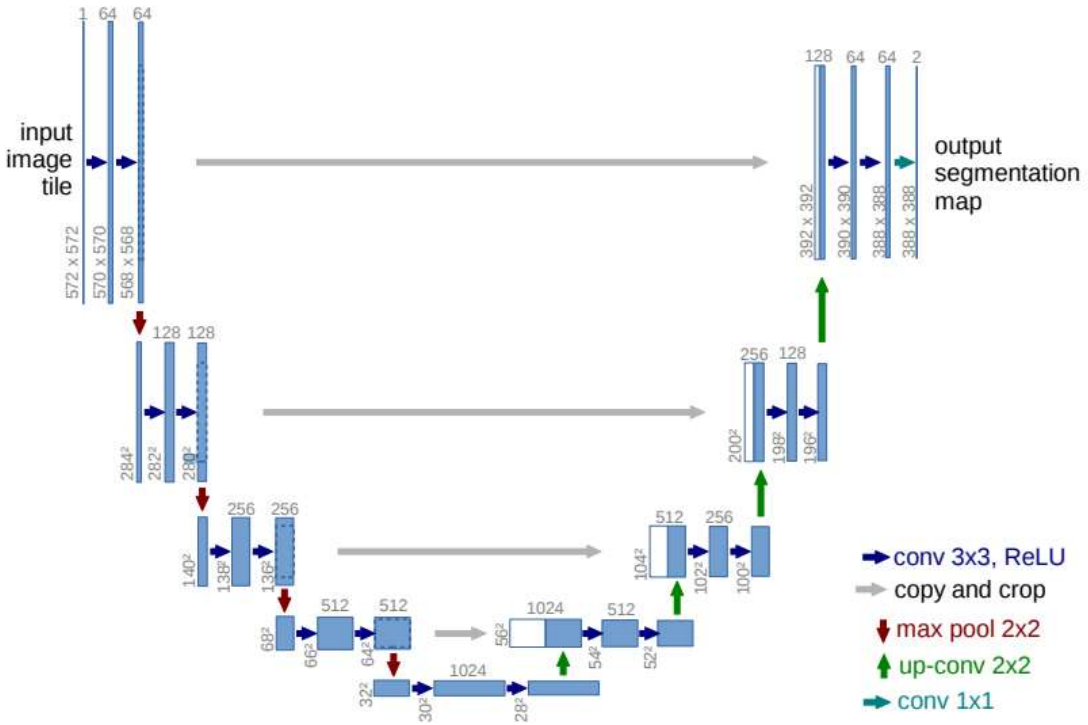


Figure 2: U-Net architecture used for the PyTorch model

provided as images, splitting them into: train/validation/test set, and then evaluate the results. The images provided are in 'jpeg' while annotations are in 'gif' format. For the Sagemaker and PyTorch model, appropriate pre-processing steps are carried out.

Metrics

The segmentation data objective is binary classification of image to whether a pixel is part of the car or not. In semantic segmentation, the Dice Coefficient or the F1 Score is used as evaluation metrics as is commonly done in segmentation of 2 classes.

$$\text{Dice Coefficient} = \frac{2 * \text{Area of Overlap}}{\text{Total number of pixels in both images}} = \frac{2TP}{2TP + FP + FN}$$

The default models on Sagemaker is monitored on two metrics, the pixel accuracy and a metric similar to the Dice Coefficient called the mean Intersection-Over-Union (mIOU).

$$\text{mIOU} = \frac{TP}{TP + FP + FN}$$

These metrics values slightly differ, but both the metrics converge similarly

Models

The first approach relied on the U-Net (Ronneberger et al. 2015) based architecture (as seen in Figure 2), which introduced the auto-encoding concept in image segmentation. The implementation used the same model architecture and number of layers as referred in the paper.

For the SageMaker models, both the FCN and PSP Net used the model architecture with a

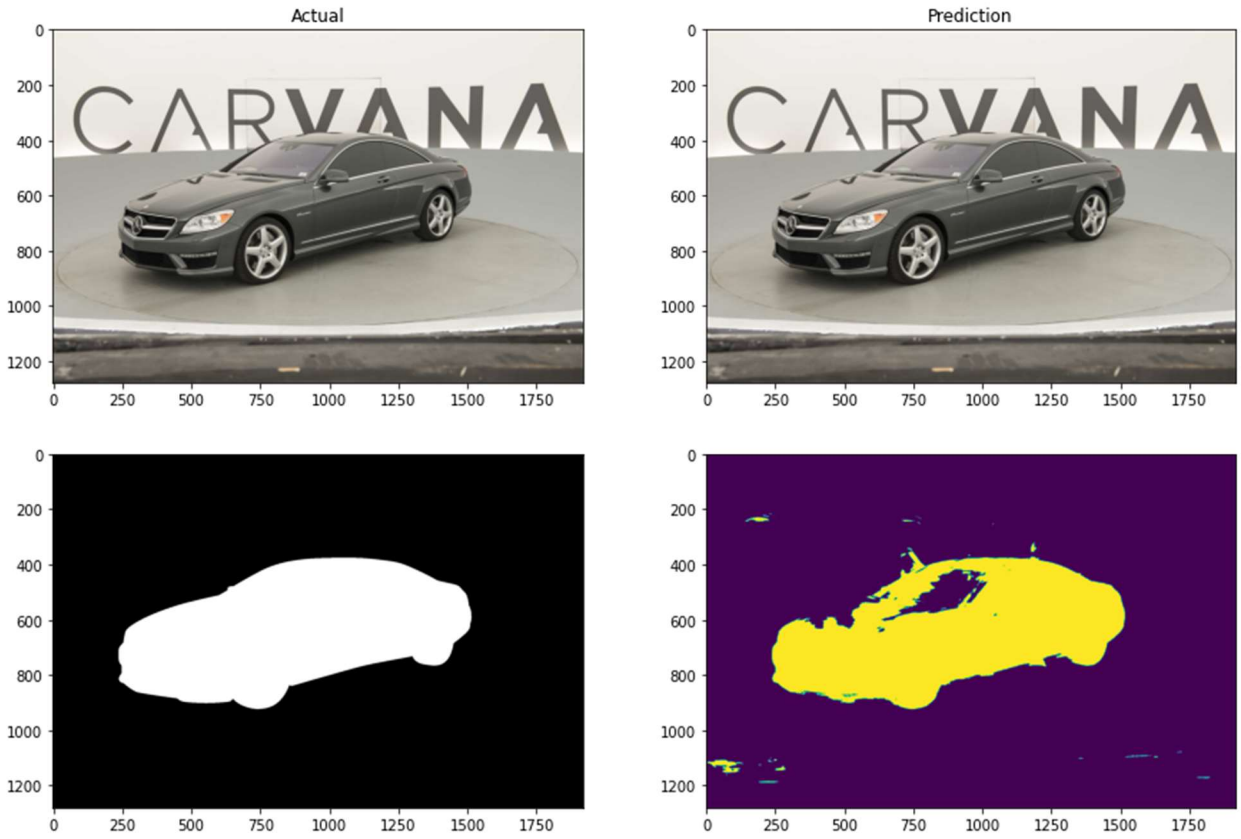


Figure 3: Semantic Segmentation results after 2 epochs on the U-Net model

backbone of Resnet-50. As the application of task is to carry out segmentation in various environments, the pre-trained models were used to achieve quicker implementation and adaption of results in new backgrounds.

Results

U-Net model

The first approach to handling the problem was using the custom PyTorch model based on the U-Net architecture. This initial approach was to get the baseline results of the model. Since the train setting did not use pre-trained weights, the model performance was not good enough. Although, it must be remarked that this model was trained only for 2 epochs (owing to limited speed in Colab's GPU instance). Future work involving making it a deeper model can boost performance. The results are shown in Figure 3.

SageMaker's in-built models

From the "semantic-segmentation" module of AWS architecture, both the model architectures were tested on same hyperparameters for 5 epochs each. Both used pre-trained weights which helped them to converge in limited number of epochs. Refer to the second Annexure section for the image results with the PSP Net model. The result with FCN looked similar but had some more instances of misclassifications.

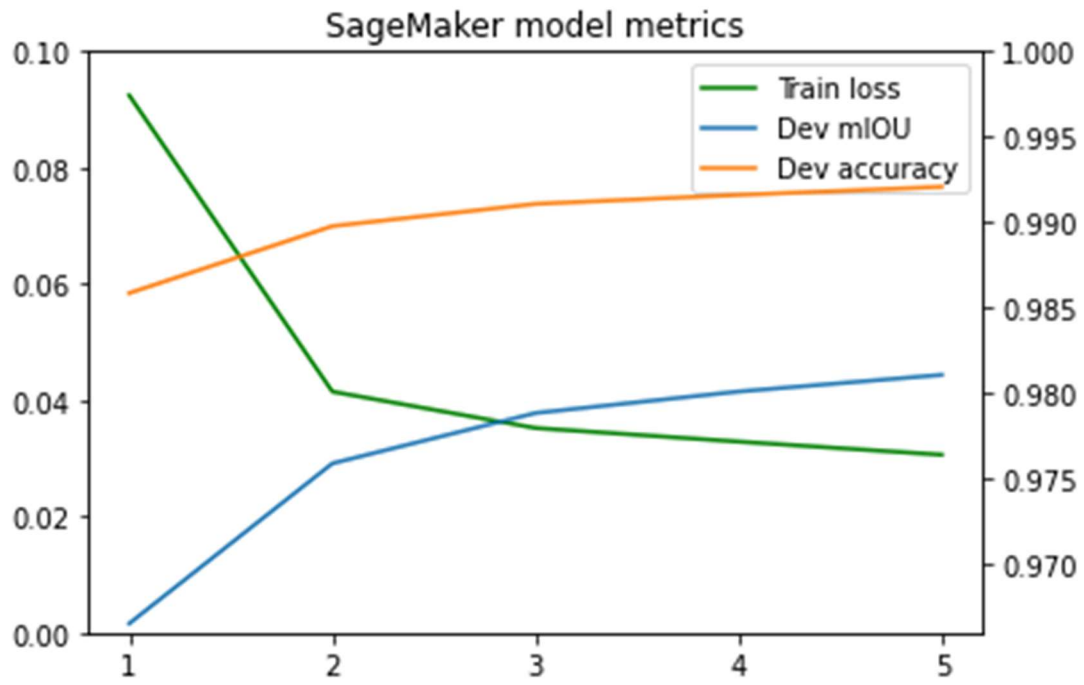


Figure 4: Semantic Segmentation results after 2 epochs on the PSP Net model

All the 3 models were converging and as an example, the metrics for the PSP Net model are visualized in Figure 4. The Table 1 enlists the results of the 3 models.

Model	**Final Dice Coefficient**	**Epochs**
U-Net	0.824	2
FCN	0.978	5
PSP Net	0.992	5

Table 1: Comparison of model performances

Conclusions

The main thing to observe is that using deep pre-trained model on tasks which are not more complex than the one on original dataset generalize well and we get high metrics accuracy right from the start of the model fitting. In general, utilizing the right kind of architecture helps in improving the model performance. We can observe the various methods having different advantages and longer training times can increase the performance on the dataset. The dataset has diverse range of cars, but all of them have similar background. For segmentation, this might mean the model can overfit on the data and classify only the specific background to class 0. This can lead to wrong predictions during production stage where images for different environments would be encountered

References

Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). Comput Vis Image Underst 110(3):346–359

Bourdev L, Maji S, Brox T, Malik J (2010) Detecting people using mutually consistent poselet activations. Comput Vis ECCV 2010:168–181

Brox T, Bourdev L, Maji S, Malik J (2011) Object segmentation by alignment of poselet activations to image contours. In: Proceedings of the 2011 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2225–2232

Cohen A, Rivlin E, Shimshoni I, Sabo E (2015) Memory based active contour algorithm using pixel-level classified images for colon crypt segmentation. Comput Med Imaging Graph 43:150–164

Csurka G, Dance C, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV, vol 1. Prague, pp 1–2

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005, vol 1. IEEE, pp 886–893

Derpanis KG (2004) The harris corner detector. York University, Toronto

He DC, Wang L (1990) Texture unit, texture spectrum, and texture analysis. IEEE Trans Geosci Remote Sens 28(4):509–512

Huang C, Davis L, Townshend J (2002) An assessment of support vector machines for land cover classification. Int J Remote Sens 23(4):725–749

Leutenegger S, Chli M, Siegwart RY (2011) Brisk: binary robust invariant scalable keypoints. In: 2011 IEEE international conference on computer vision (ICCV). IEEE, pp 2548–2555

Long J, Shelhamer E, Darrell T (2016) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 79(10):1337–1342

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110

Mair E, Hager G, Burschka D, Suppa M, Hirzinger G (2010) Adaptive and generic corner detection based on the accelerated segment test. Comput Vis ECCV 2010:183–196

Maldonado-Bascon S, Lafuente-Arroyo S, Gil-Jimenez P, Gomez-Moreno H, López-Ferreras F (2007) Roadsign detection and recognition based on support vector machines. IEEE Trans Intell Transp Syst 8(2):264– 278

Medioni G, Yasumoto Y (1987) Corner detection and curve representation using cubic b-splines. Comput Vis Graph Image Process 39(3):267–278

Moon N, Bullitt E, Van Leemput K, Gerig G (2002) Automatic brain and tumor segmentation. Med Image Comput Comput Assist Interv MICCAI 2002:372–379

Rosten E, Drummond T (2005) Fusing points and lines for high performance tracking. In: Proceedings of the tenth IEEE international conference on computer vision, 2005. ICCV 2005, vol 2. IEEE, pp 1508–1515

Rosten E, Porter R, Drummond T (2010) Faster and better: a machine learning approach to corner detection. IEEE Trans Pattern Anal Mach Intell 32(1):105–119

Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

Smith SM, Brady JM (1997) Susana new approach to low level image processing. Int J Comput Vis 23(1):45–78

Shi J et al (1994) Good features to track. In: Proceedings of the 1994 IEEE computer society conference on CVPR'94 computer vision and pattern recognition. IEEE, pp. 593–600

Wei GQ, Arbter K, Hirzinger G (1997) Automatic tracking of laparoscopic instruments by color coding. In: CVRMed-MRCAS'97. Springer, Berlin, pp 357–366

Zhao H, Shi J, Qi X, Wang X, Jia J (2016) Pyramid scene parsing network. arXiv preprint arXiv:1612.01105

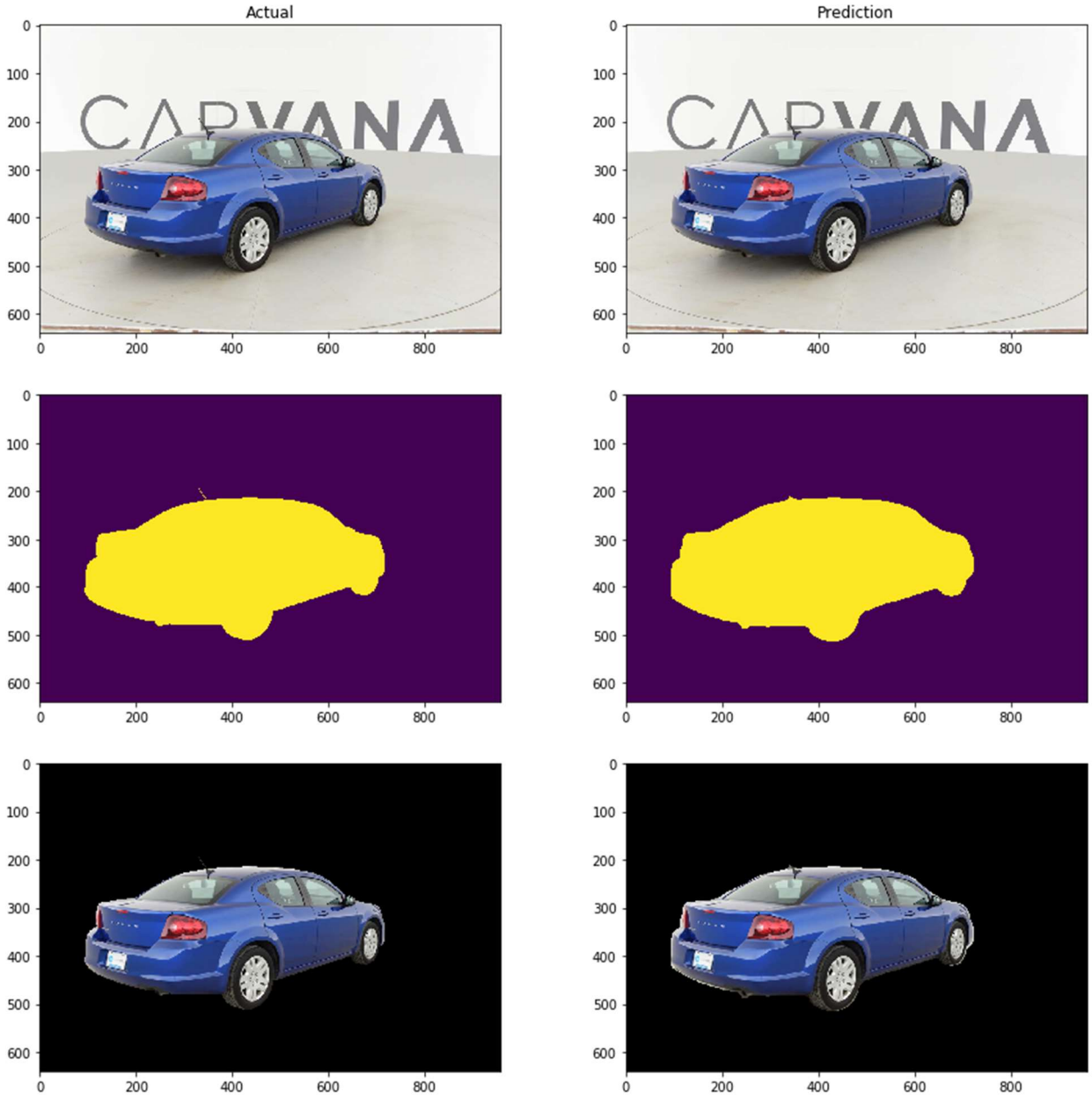
Zhu SC, Guo CE, Wang Y, Xu Z (2005) What are textons? Int J Comput Vis 62(1):121–143

Annexures

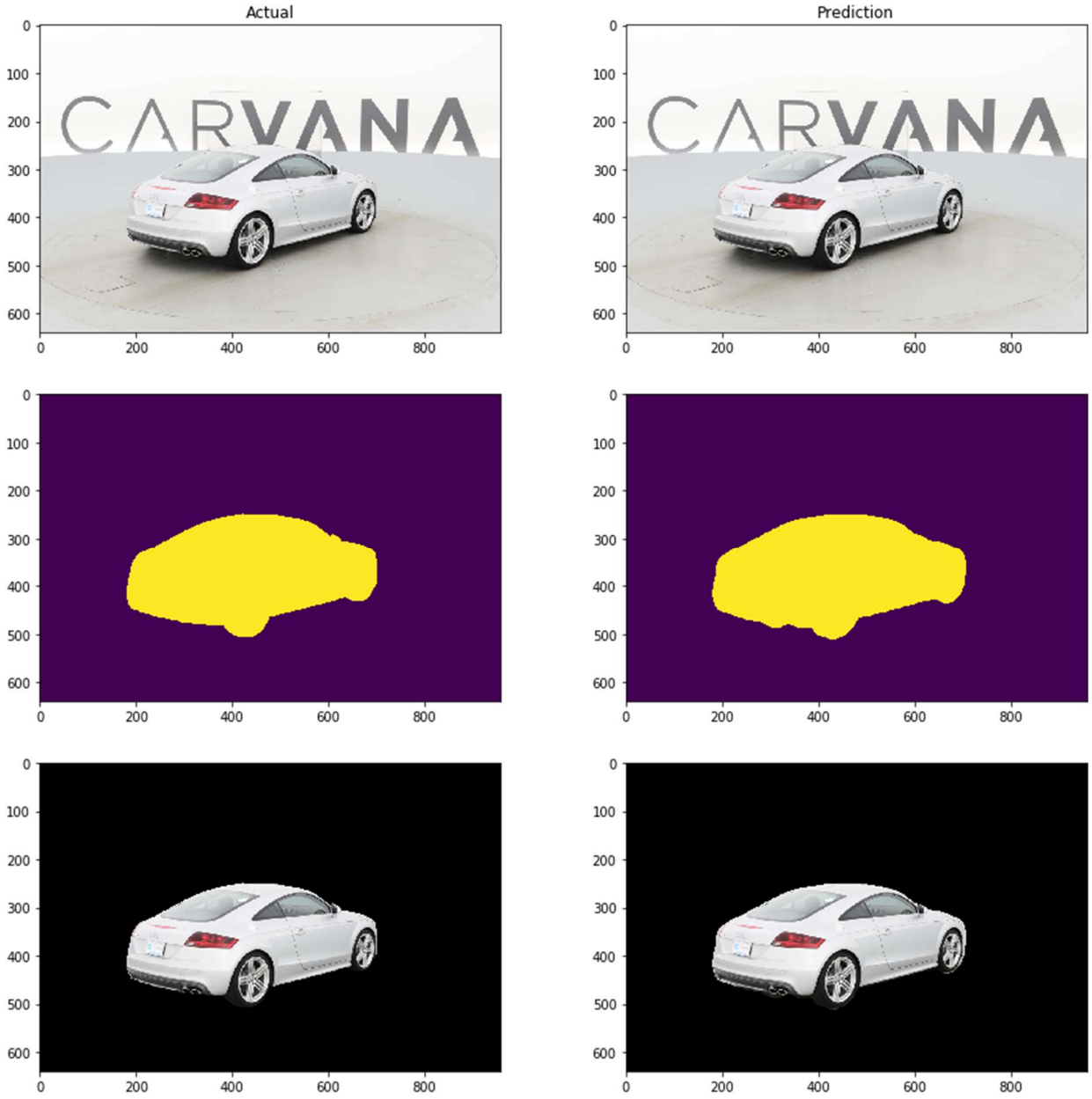
1. Sample image from the dataset with the required binary masking attempts shown

Figure 5: Dataset description

2. Adding two examples of segmentation prediction using the best performing PSP Net model



(a)



(b)

Figure 6: Comparison of actual and predicted image, mask and the masked image for two random image samples