Jay Kim

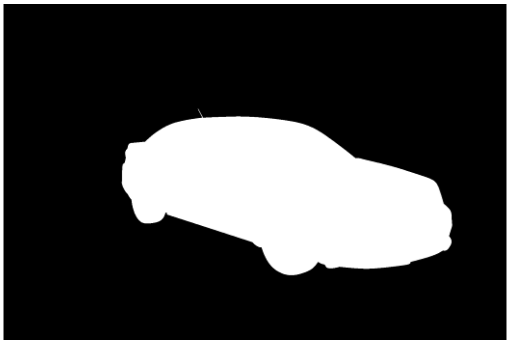
CMPUT 466

December 5, 2023

Dr. Lili Mou

Semantic Segmentation with Carvana Dataset

Semantic segmentation is a vital component in computer vision, allowing for precise object delineation within images. This project delves into the Carvana Image Masking Challenge, leveraging the Carvana dataset obtained from Kaggle. The primary objective is to develop robust semantic segmentation models for car detection and labeling. The dataset, consisting of 5088 car images, is divided into training (70%), validation (15%), and testing (15%) sets. In semantic segmentation, the goal is to classify each pixel in an image as belonging to a car or not. This project exclusively focuses on semantic segmentation, a crucial step in object detection.



The input is the dataset is the set of cars with 3 color channels and each car comes with its own black and white mask, which has one channel. The output of the model produces the same length and width size image but with one color channel to produce the segmentation mask, where it classifies each pixel as car or non-car.

In this experiment, we explore three distinct network models: CNN, DeepLabV3, and U-Net. CNN serves as the foundational or baseline model due to its simplicity. On the other hand, DeepLabV3 introduces advanced features, leveraging atrous convolution and varied backbone architectures. The model's architecture includes a convolutional backbone for fundamental feature extraction, an Atrous Spatial Pyramid Pooling (ASPP) module employing atrous convolutions to capture information at multiple scales, and a Global Average Pooling layer for holistic context aggregation. This combination enables the model to accurately segment cars in images by considering both local and global contextual information. The final convolutional layer integrates outputs from the ASPP module and Global Average Pooling, resulting in a comprehensive segmentation map. This approach enhances the model's ability to discern intricate details and contributes to its effectiveness in semantic segmentation tasks, particularly in the context of identifying cars within images.

The U-Net, another complex model, is derived from its distinctive U-shaped architecture, resembling the letter U when viewed from above. Its design encompasses both downsampling and upsampling pathways, facilitating the extraction of hierarchical features crucial for accurate segmentation. In the downsampling phase, DoubleConv blocks progressively reduce spatial dimensions while augmenting the number of features. The upsampling section employs ConvTranspose2d layers and DoubleConv blocks, incorporating skip connections to preserve high-resolution information. The architecture's bottleneck ensures the contraction and subsequent expansion of features, enhancing contextual understanding. A final 1x1 convolutional layer maps the learned features to the desired output channels, yielding a comprehensive segmentation map. U-Net's intricate design, with a focus on spatial detail preservation through skip connections, positions it as a key semantic segmentation model for effectively delineating cars in images within the context of this experiment.

For these three models, we applied BCEWithLogitsLoss (Binary Cross Entropy) as the loss function and the AdamW optimizer with a learning rate of 1e-4. Common hyperparameters include the learning rate, batch size, number of epochs, and AdamW optimizer. To change the parameters, it can be done within the “train_validate_test_Carvana.py”. A systematic search can be done by defining a search space; one can make a list of optimizers batch_size, learning_rate, and num_eopochs, then do a grid search to find the best parameters. For this experiment, CNN will be the baseline model because it is one of the rudimentary methods to solve semantic segmentation.

The primary evaluation metric employed is the Dice score, computed through the check_accuracy function in “utils.py”. The Dice score is a common metric in semantic segmentation tasks that gauges the similarity between the predicted segmentation masks and the corresponding ground truth masks. Specifically, the Dice score is calculated as $\frac{2*preds*y}{preds+y}$ and it is good if Dice score is closer to 1. A higher Dice score indicates a better overlap between the predicted and ground truth masks. Another evaluation metric, accuracy, is also computed, defined as $\frac{number\ of\ correct\ pixels}{number\ of\ pixels\ of\ preds\ and\ y}$. The key distinction between Dice and accuracy lies in the fact that the Dice score provides a more informative measure of accuracy, taking into account the spatial alignment of predicted and ground truth masks.

To run the file, please run “train_validate_test_Carvana.py” and make sure you have created the folders to store the generated images.

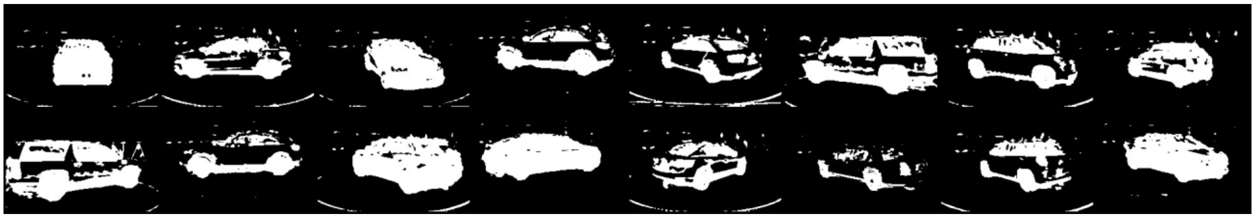
Results

All of the models had the 4 epochs with batch size of 16 and learning rate of 1e-4. Below is the table while performing training and validation and testing.

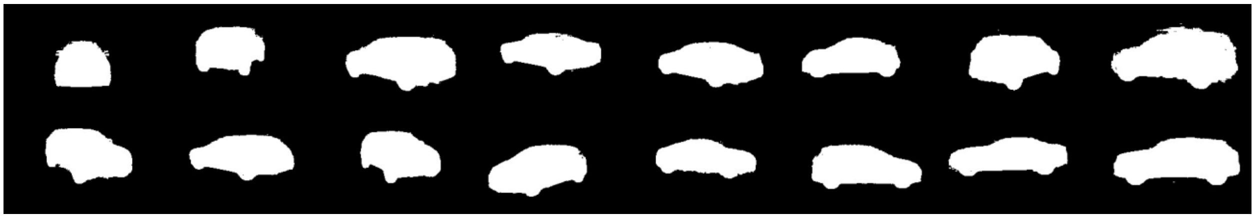
	Epoch 1		Epoch 2		Epoch 3		Epoch 4		TEST	
Model	Dice	Accuracy	Dice	Accuracy	Dice	Accuracy	Dice	Accuracy	Dice	Accuracy
CNN	0.666	88.33	0.717	90.23	0.750	91.12	0.783	91.81	0.818	92.81
DeepLabV3	0.971	98.78	0.974	98.91	0.980	99.17	0.984	99.30	0.984	99.30
U-Net	0.982	99.23	0.962	98.48	0.987	99.46	0.988	99.50	0.989	99.51

Below are the generated pictures of CNN, DeepLabV3, and UNET from the TEST set.

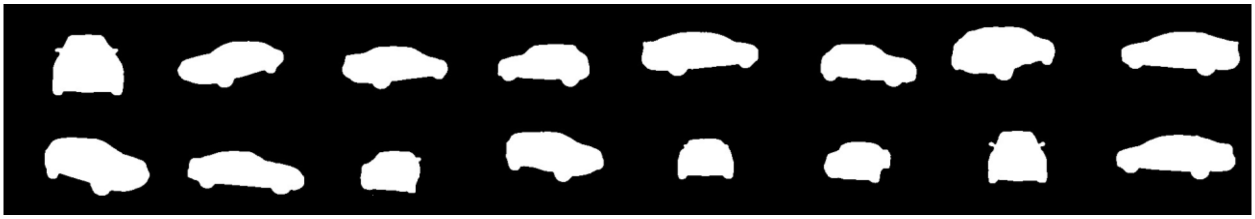
CNN



DeepLabV3



U-Net



Compared to our baseline model, DeepLabV3 and U-Net outperformed when generating the masks. CNN did manage to find the edges of the car, but it failed to shade the body of the car. DeepLabV3 and U-Net shows promising result. However, U-Net did slightly better than DeepLabV3; some of the test results, DeepLabV3 provided fuzzy car edges. Based on the DICE score, both models did a great job on capturing the car's mask.