Computer Vision – Carvana Image Masking

STAT 4984 – Deep Learning, Spring 2022

Pranav Thiriveedhi

PID: pthiriv792

ABSTRACT

In the project, I am challenged with developing an algorithmic based deep learning model that automatically removes the photo studio background of Carvana car photo. This will allow Carvana to superimpose cars on a variety of backgrounds. I will be analyzing a dataset of photos, covering different vehicles with a wide variety of year, make, and model combinations. I will also be using given car photo masks with the task of generating machine learning based masks so I can impose a background on any car photo.

Keywords:

MOTIVATION PROBLEM STATEMENT

Carvana is a car dealership website that takes a multitude of vehicles and lists them online for customers to purchase and have delivered without even having to get up. One of Carvana's selling points is the advertisement and cover for each vehicle they list on the website. Most of the cars, if not all of them, come with a 360-degree view of the car imposed on a basic Carvana designed background. The problem for this project, is taking that view of images, and taking out the background so that the website can superimpose the car images in a variety of backgrounds. This would allow the company to advertise their vehicles in a creative way, different from the basic backgrounds that most of the other car-selling competitors provide.

This problem is interesting to me for a few reasons. First, you can't obviously take out the backgrounds manually for each car. There are thousands and thousands of cars on the website, so you would need an algorithm that automatically removes the backgrounds. Researching and creating an algorithm to run this task seems really interesting and would help me learn a lot more about deep learning and real-world application uses of certain types of deep learning. Another reason that I find this problem to be interesting is the type of data I would be working with. I am a car enthusiast, so working with the images of different vehicles over a wide range of makes and models and years is fun and interesting to myself. Working with that collection of data and knowing its real-world application definitely helps in understanding what I need to do and what I need to do to make sure this project is successful.

GITHUB REPOSITORY LINK

https://github.com/prans792/stat4984

DESCRIPTION OF DATA (INCLUDING PLOTS)

I am using the dataset of images from the Kaggle challenge website. The Kaggle website link is listed here: $(https://www.kaggle.com/c/carvana-image-masking-challenge/data?select =$

$29bb3ece3180_11.jpg$)

The data set in the listed website contains a large number of car images (as .jpg files). Each car has exactly 16 images, each one taken at different angles. Each car has a unique id and images are named according to $id_01.jpg$, $id_02.jpg$ ... $id_16.jpg$, which are used to create a 360-degree view of the car. Using the 16 photos for each car would be difficult, so setting the input as a stack containing these 16 images as a 3d input as one 3d input image would be the correct thing to do.

The 9 file folders listed in the website challenge page include the data I would need to complete this project. More specifically, I would just need five of the folders full of data. These files include the train and test folders, the $train_hq$ and $test_hq$ folders, and the $train_masks$ folder of data. An example of the photos listed in the data set is shown below:

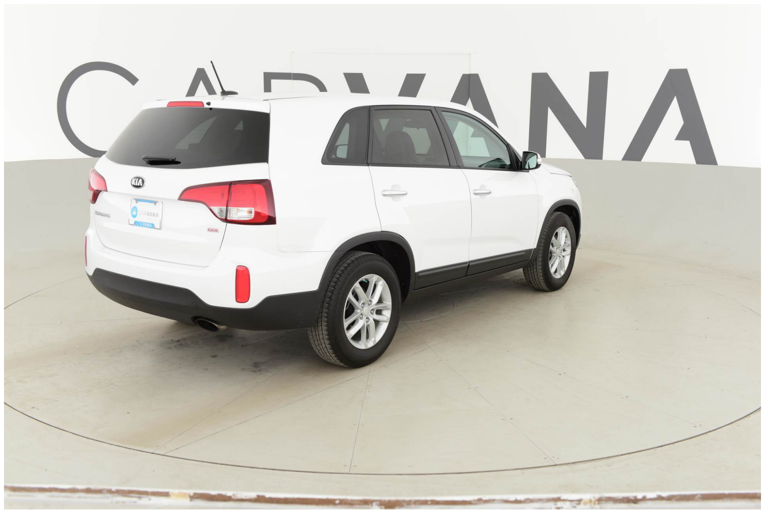

Figure 1. Sample Image from Data Provided

ARCHITECTURE CHOICE (CNN)

The type of deep learning neural network I have decided to use is Mask-R-CNN, which is a type of convolutional neural network used to perform image segmentation. Using this specific architecture of deep learning is especially very useful in automatically computing pixel-wise masks for a specific object in an image. This allows us to segment the foreground (object) from the background, which is exactly what we are achieving to accomplish in the project.

Convolutional Neural Networks, or CNNs, are the route I will be taking to accomplish the tasks of this project. There are a lot of deep learning architectures other than CNNs. They include RNNs, LSTMs, GAMs, etc. Convolutional Neural Networks sounds like a very complex deep learning architecture, but after a brush up on the structure of it and a proper understanding, the architecture makes a lot of sense and is simple to develop. CNNs mainly consist of three types of layers:

1. Convolutional layers, where a kernel (or filter) of weights is convolved to the algorithm to extract certain features like a specific object in a picture.
2. Nonlinear layers, which apply an activation function on feature maps in order to enable the modeling of non-linear functions by the network.

3. Pooling layers, the last part of a CNN, which replaces a small neighborhood of a feature map with some statistical information (mean, max, etc.) about the neighborhood and reduce spatial resolution.

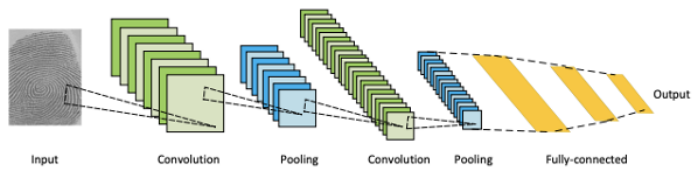

Figure 2. Map of Convolutional Neural Network

The above picture is a map of a convolutional neural network and how it works.

Out of all the different deep learning architectures, in the tree of the various CNNs, there is a specific type called R-CNN, which stands for regional convolutional neural network. This specific type of CNN has proven successful in the field of object detection, which is exactly the name of the game we are playing.

For this project, and the data given, we can train the algorithm using the reference pictures and true masks. The true masks will help the neural network understand what we are trying to mask, remove from the picture. From there, with a better trained algorithm, we can develop a predicted mask of the background and automatically apply a picture of any car to any background. This algorithm would basically be a back version of cutting the car out of the standard carvana image and pasting it into any background scenery that the company would want to use.

PRESENTATION OF RESULTS

The first thing I did was attach the input image with it's corresponding true mask. An example of that is below:

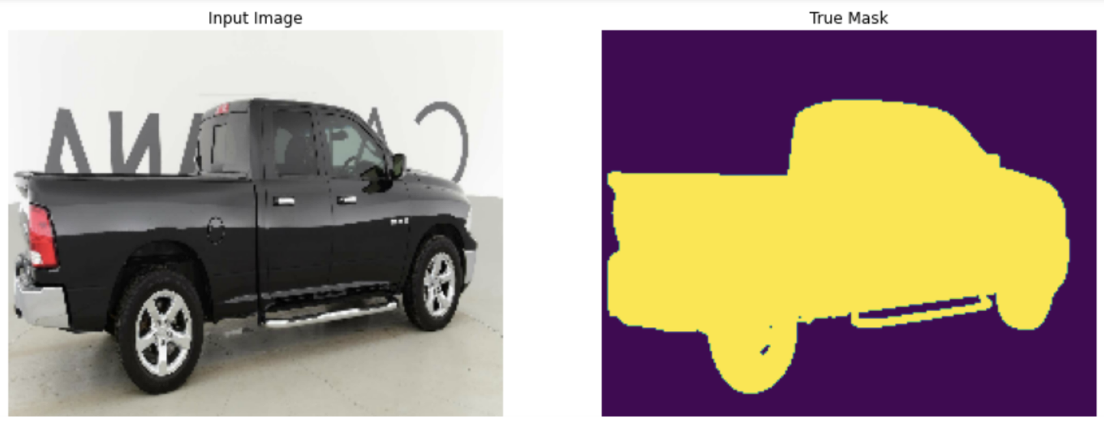

Figure 3. Input image, true mask

Was able to use a convolutional neural network type architecture that was trained and used image segmentation to output a pixel-wise mask of the image of which will help us create a picture without the car, only the empty background mask. Below is an example of the output I have received:

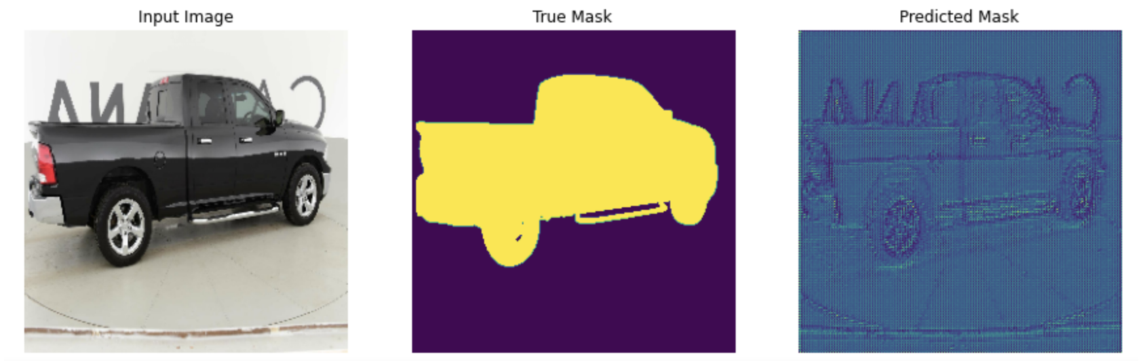

Figure 4. Input image, true mask, and predicted mask

After recieving this output, and going thru further conditioning using the training data and epochs of 15 steps, I was able to fit a better predicted mask model. An example is attached below:

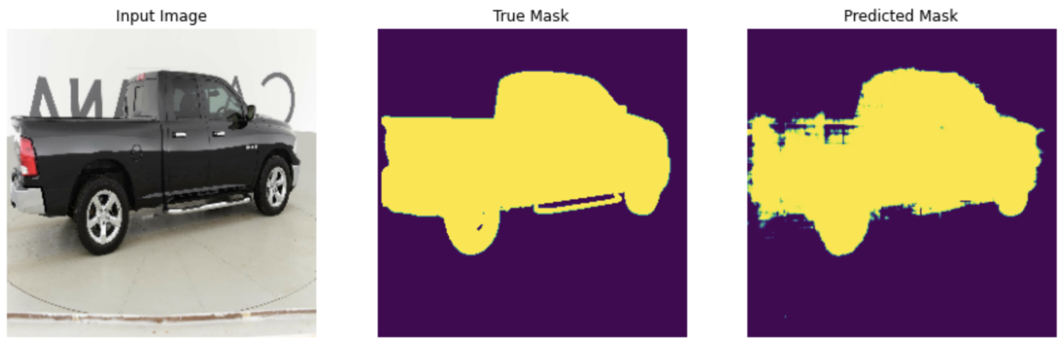

Figure 5. Input image, true mask, and predicted mask

ANALYSIS OF RESULTS

In this section, I will go thru each section of my code and explain what I did to achieve the results I have presented above:

1. The first thing I did was importing the necessary libraries. This includes the standard Python libraries and the ML related libraries.

2. Next, I prepared the train sets and train masks, which are what we are working with. Alongside each car picture and car mask, I attached an ID so the images would correlate with each other in Data Frame notation.

3. I then created a final table of the ID, car photo, and it's given car mask.

4. After the organization of that table, I created three functions to normalize the images (both the car input images and the true masks given alongside the data).

5. Then, I split the data from the main data frame into a train data frame and a test data frame

6. After that, I created a display function to relay the images of the car and the image of the car mask side by side. These are all given data points so far.
7. Next, I created a machine learning model using U-net image segmentation. Then, I trained that model and set it to visualize a deep learning made predicted mask.
8. Then, I created a model summary, which showed that there were 4,657,441 trainable parameters and 1,843,904 non-trainable parameters.
9. I finally then further trained the model and fit it so that the predicted mask becomes more and more similar to the true mask given.
10. I got a final accuracy of 99.31 percent after going thru 15 epochs.

If you look at figure 5, of the presentation of results section, you can see that the predicted mask is very similar to the true mask that was given in the data. Although it is very close, the mask could look way better. With a better implementation of the model I selected, I feel I could get a better image of the mask.

INSIGHTS/DISCUSSIONS RELATED TO PROJECT

At the end of this project I was able to create a mask of the car image, that is over 99 percent accurate. This mask was created using U-net image segmentation and can be used to superimpose the picture of the car over any background.

Image segmentation is a powerful artificial intelligence technique that helps in object detection, background and foreground segregation, object tracking, object analysis, and detecting anomalies. Image segmentation segments the image into small fragments based on the image's specific parameters, which are characteristic of the chosen area of the image. The majority of computer vision projects use image segmentation as it's backbone. It is a vital aspect of detecting objects and is the reason I used it for this project.

Link 6 in the reference section provides a deep understanding of image segmentation and it's use in industry. It is a great discussion of it's tools and techniques and proves how important image segmentation is in the real world.

REFERENCES

1. https://www.kaggle.com/c/carvana-image-masking-challenge/overview
2. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
3. https://www.carvana.com/
4. https://pyimagesearch.com/2020/09/28/image-segmentation-with-mask-r-cnn-grabcut-and-opencv/
5. https://arxiv.org/pdf/2001.05566.pdf
6. https://medium.com/projectpro/u-net-convolutional-networks-for-biomedical-image-segmentation-435699255d26