

Image Segmentation with U-Net

Welcome to the final assignment of Week 3! You'll be building your own U-Net, a type of CNN designed for quick, precise image segmentation, and using it to predict a label for every single pixel in an image - in this case, an image from a self-driving car dataset.

This type of image classification is called semantic image segmentation. It's similar to object detection in that both ask the question: "What objects are in this image and where in the image are those objects located?," but where object detection labels objects with bounding boxes that may include pixels that aren't part of the object, semantic image segmentation allows you to predict a precise mask for each object in the image by labeling each pixel in the image with its corresponding class. The word "semantic" here refers to what's being shown, so for example the "Car" class is indicated below by the dark blue mask, and "Person" is indicated with a red mask:



Figure 1: Example of a segmented image

As you might imagine, region-specific labeling is a pretty crucial consideration for self-driving cars, which require a pixel-perfect understanding of their environment so they can change lanes and avoid other cars, or any number of traffic obstacles that can put peoples' lives in danger.

By the time you finish this notebook, you'll be able to:

- Build your own U-Net
- Explain the difference between a regular CNN and a U-net
- Implement semantic image segmentation on the CARLA self-driving car dataset
- Apply sparse categorical crossentropy for pixelwise prediction

Onward, to this grand and glorious quest!

Table of Content

- [1 - Packages](#)
- [2 - Load and Split the Data](#)
 - [2.1 - Split Your Dataset into Unmasked and Masked Images](#)