Semantic Segmentation in Computer Vision

Credits: Nanodegree Program in Computer Vision by Udacity

Table of Contents

Introduction

Semantic segmentation goes one step forward from object detection algorithms such as YOLO or SSD. The difference resides in Semantic Segmentation attempts to classify every pixel in the image, rather than just creating a bounding box surrounding the detected object.

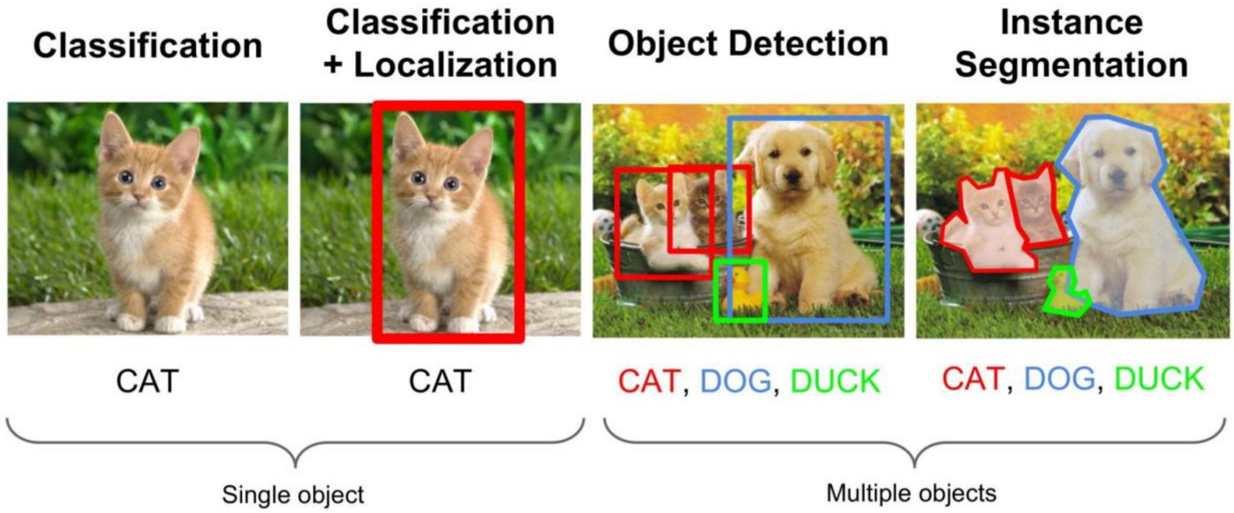

Figure 1. Evolution of complexity for object recognition tasks in computer vision

There is still a discrete set of categories the network can classify, much like in a classification task. However, instead of assigning a single class to an image, we want to assign a class to every pixel in that image. So, how do we approach this task?

Fully-Convolutional Networks FCN

Since our goal is to classify every pixel, we are this time concerned about spatial information, something that did not happened in image classification.

For instance, one of the motivations of Geoffrey Hinton to invent Capsule Nets is that regular CNNs don't make a distinction between the two faces in Figure 2 when attempting to classify it as a face. Why? Simply because it is still able to find the patterns (edges, corners…) that helps it identify it is a face.

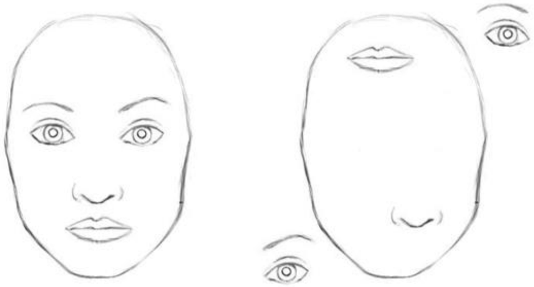

Figure 2. Spatial inconsistency on regular CNNs

Because of being then concerned about spatial information, we can no longer use down sampling by pooling layers or fully-connected layers at the end of the network. Instead, we need an architecture that only consists on convolutional layers – therefore the name fully convolutional network.

Training

This FCN would take in an image that has true labels attached to each pixel, so every pixel is labeled as a category: grass or cat or sky, and so on.

Then we pass that input through a stack of convolutional layers that preserve the spatial size of the input, by convolving the input volume with zero padding.

Then, the final convolutional layer outputs a tensor that has dimensions CxHxW, where C is the number of categories we have. The output volume dimension C shows the probabilities of this single pixel to be a cat, grass, sky, ...

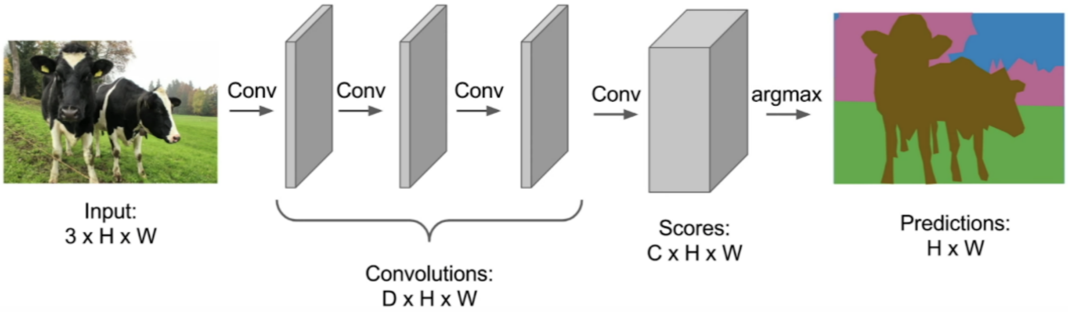

Figure 3. FCN

This pixel level classification can be done all at once, and then train this network by assigning a loss function to each pixel in the image and doing backpropagation as usual. So, if the network makes an error and classifies a single pixel incorrectly, it will go back and adjust the weights in the convolutional layers until that error is reduced.

Drawbacks

First of all, there is a lot of work to do to label every pixel of every image to perform the training. Furthermore, it intuitively sounds very computational costly to have to maintain spatial convolution at every layer. The solution to this is a slight modification to the network architecture.

Downsampling – FCN – Upsampling

It is normally preferred to perform a downsampling on the feature maps through pooling operations, then stack the FCN at that reduced volume sizes, and after that perform back an Upsampling to match the input image size so the classification can be done pixel-wise, as shown in FIG:

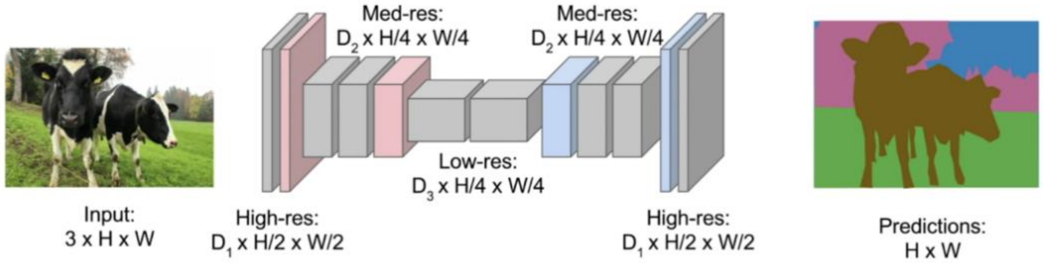

Figure 4. Downsampling – FCN - Upsampling

Scene understanding

One approach to scene understanding is to train multiple decoders, where each decoder trains on a separate task. Figure 5 propose an architecture where one decoder will be trained to perform segmentation mentioned before, while the other decoder could be trained to infer how far objects are.

These two abilities can be easily imagined for improving scene understanding on self-driving cars.

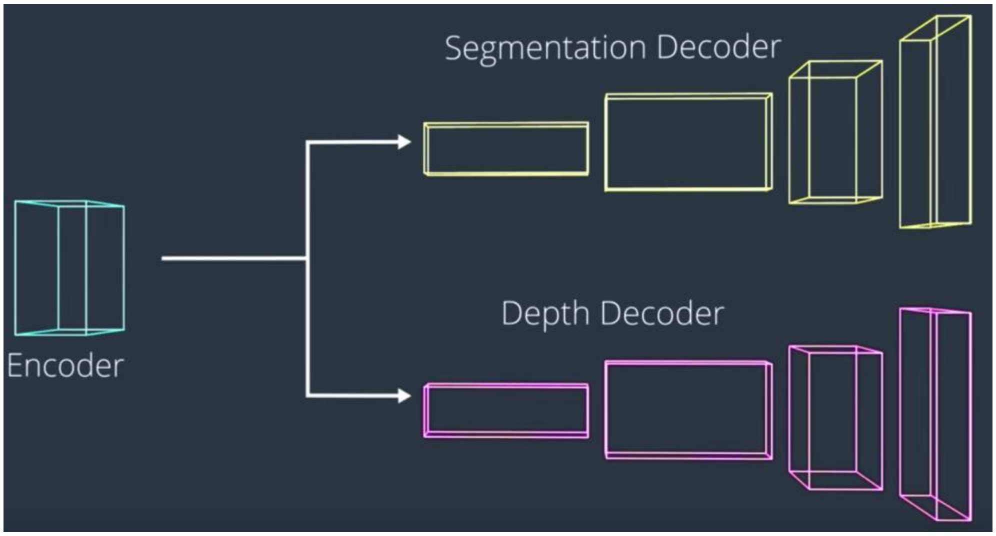

Figure 5. Scene understanding through multiple decoders

Metrics – IOU

How do we measure how good is the network at doing semantic segmentation?

The Intersection Over Union (IOU) measures the intersection section between the prediction and the real values of the pixels of the images shown in the left-hand side of Figure 6, divided by the union section of them, shown in right-hand size of Figure 6.

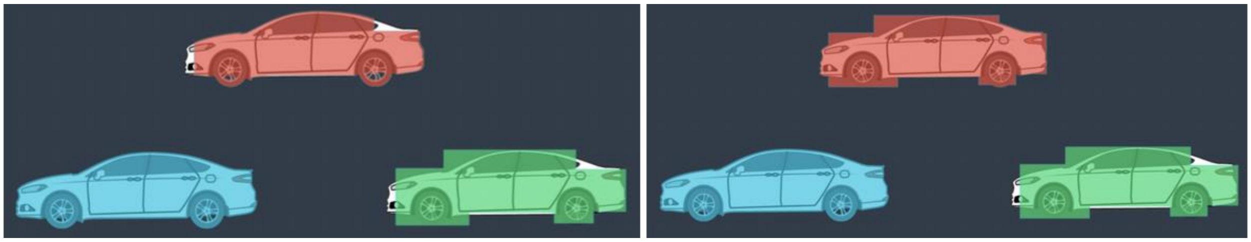

Figure 6. IOU

So, Intersection is an AND operation where only the pixels belonging to both counts, whereas Union is an OR operation where any point belonging to at least one of them counts.

Therefore, IOU always lies between 0 – 1.

Lastly, this value is computed class-wise. To have an idea of how the network performs, we can simply average the IOU for all classes.