Machine Learning Engineer Nanodegree

Shashank Kumar Tekriwal

1st April, 2018

Image Segmentation for Outfit Detection

Domain Background

In order to further investigate the domain of deep learning, I decided to explore the field of image segmentation. The output of an image classification model is a discrete probability distribution: one number between 0 and 1—a probability—for each class the model is trained to recognize. The objective of an image segmentation model is to segment an image and classify various objects of interest within that image. Instead of predicting a single probability distribution for the whole image, the image is divided into a number of blocks and each block is assigned its own probability distribution. Image segmentation typically generates a label image the same size as the input whose pixels are color-coded according to their classes.

Clothing parsing is a specific form of semantic segmentation, where the categories are one of the clothing items, such as t-shirt. Clothing parsing has been actively studied in the vision community, perhaps because of its unique and challenging problem setting, and also because of its tremendous value in the real-world application. This problem is very challenging due to the large diversity of fashion items and the absence of pixel-level tags, which make the traditional fully supervised algorithms inapplicable.

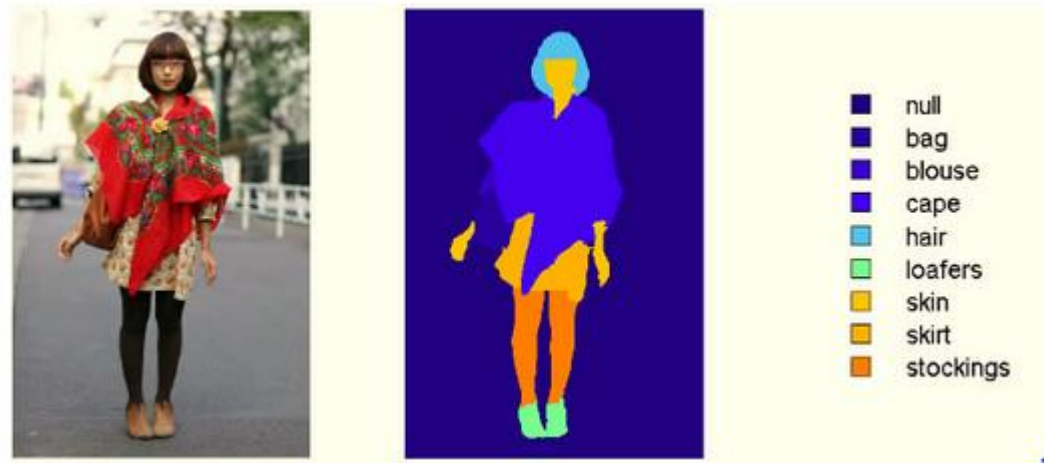
Some ideas regarding fashion parsing have been discussed in these papers, but I will be only implementing basic and simple models for the purpose of learning.

1. https://arxiv.org/pdf/1703.01386.pdf
2. http://hi.cs.waseda.ac.jp/~esimo/publications/SimoSerraACCV2014.pdf
3. http://vision.is.tohoku.ac.jp/~kyamagu/ja/research/clothing_parsing/

Problem Statement

The problem statement can be posed as the detection and labelling of different categories of clothes within an image that consists of a single front-facing person with visible full body. Given below is a sample of input image and a corresponding output image (ground truth) that the model should be capable of learning and reproducing.

The input images include an exclusive person in fashionable clothes. Apart from clothing categories, the model should also be able to learn to classify skin and hair, thus making the problem even more complicated.



Datasets and Input

I will be primarily using the *clothing-co-parsing* dataset hosted here: https://github.com/bearpaw/clothing-co-parsing. This contains above 1000 images with pixel level annotations, which can be directly used for training. The labels are distributed over 59 classes, some of which are shown in the image above.

Additionally, I will be working on the *colorful-fashion-parsing-dataset* hosted here: https://sites.google.com/site/fashionparsing/dataset. The CFPD dataset contains above 2000 annotated images, across 23 labels.

The goal is to produce *segmentation* models for both of these datasets separately and evaluate how close they can get to the ground labels.

Benchmark Model

There is already a lot of advanced work done in this field using techniques like *outfit-encoders* and *conditional random fields* to optimize to further fine-tune the output of deep neural networks. Various strategies and their state-of-the-art outputs have been discussed in this paper here: https://arxiv.org/pdf/1703.01386.pdf

This paper can be used as a reference benchmark model against my outputs.

Evaluation Metrics

As I will be training a CNN, *loss (Euclidean Loss)* and *accuracy*, can be used to evaluate the model.

$$Euclidean\ Loss = \frac{1}{2N}\sum_{i=1}^{N}\|y_i - x_i\|_2^{\ 2}\ , where$$

x_i are the predicted values, y_i $are\ the\ observed\ values$, and N $is\ the\ batch\ size$.

$$Accuracy = \frac{Number\ of\ pixels\ correctly\ classified}{Total\ number\ of\ pixels}$$

The average accuracy over the test set will be used to measure the performance of our model.

Project Design and Solution Statement

I will be training deep convolutional neural networks for segmentation, in particular, *fully convolutional networks* – that have advantage of being independent of the input size. This link https://devblogs.nvidia.com/image-segmentation-using-digits-5/ describes how we can convert classical classification architectures (e.g. *alexnet and vgg-16*) into their fully convolutional counterpart for segmentation. The coursework by *stanford* hosted here http://cs231n.github.io/ provides excellent resources to further complement my knowledge in this domain. I propose to follow the following three steps and observe their effects on the output.

1. Use semantic segmentation networks – FCNs (*at least fcn-32s, fcn-16s and fcn-8s)* and learn them from the scratch using random weight initialization. Their network definition along with other successful architectures is described in the link below: https://github.com/shelhamer/fcn.berkeleyvision.org

2. Use *Transfer Learning* and fine-tune the weights of some pre-trained network. Training an entire convolutional network from scratch (with random initialization) is difficult because it is relatively rare to have a dataset of sufficient size. Instead, it is common to pre-train a network on a very large dataset (e.g. ImageNet which contains 1.2 million images with 1000 categories), and then fine-tune those pre-trained weights as an initialization for the task of interest.

3. Use *Conditional Random Field* as a post-processing step to further refine the outputs from the above learnt networks. CRF helps to estimate the posterior distribution given predictions from our network and raw RGB features that are represented by our image. It does that by minimizing the energy function which are defined by the user. In our case it takes into account the spatial closeness of pixels and their similarity in RGB feature space (intensity space). On a very simple level, it uses RGB features to make prediction more localized – for example the border is usually represented as a big intensity change – this acts as a strong factor that objects that lie on different side of this border belong to different classes. Details of implementation can be find here: http://warmspringwinds.github.io/tensorflow/tf-slim/2016/12/18/image-segmentation-with-tensorflow-using-cnns-and-conditional-random-fields/

I believe I have done enough ground work required to get started for a project like this. I sincerely hope this proposal meets the required expectations and qualifies for the Capstone Project.