Hypercolumns for object segmentation and fine-grained localization

(2015)

Hariharan, Bharath Arbelez et al.
Resume

March 1, 2019

---

1 Introduction

Recognition algorithms based on convolutional networks (CNNs) typically use the output of the last layer as a feature representation. However, the information in this layer may be too coarse spatially to allow precise localization. On the contrary, earlier layers may be precise in localization but will not capture semantics. To get the best of bothworlds, we define the hypercolumn at a pixel as the vector of activations of all CNN units above that pixel. Using hypercolumns as pixel descriptors gives better results in keypoints detection and object segmentation (instance segmnetation).

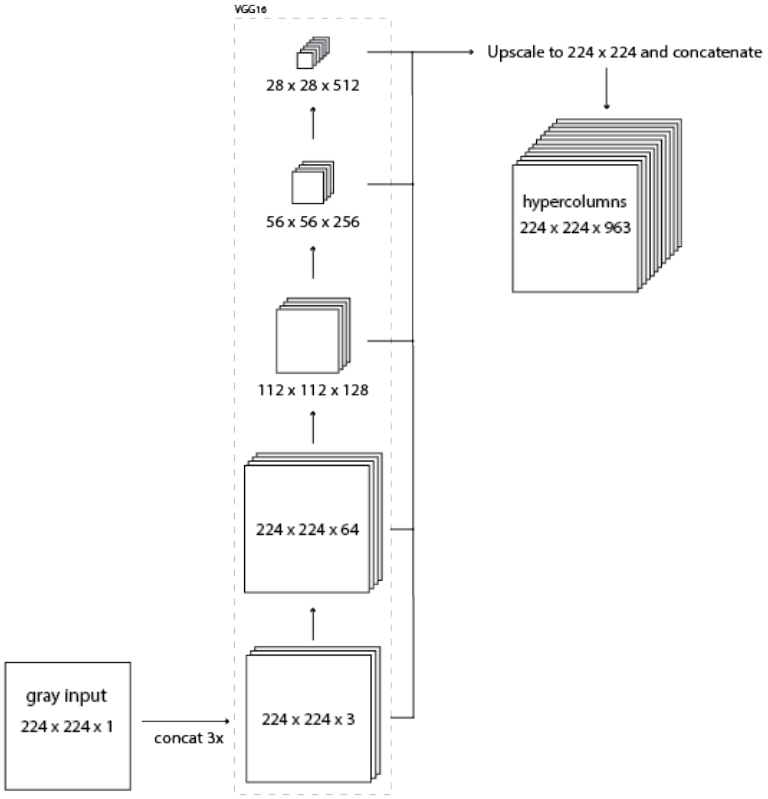
Usage of hypercolumns:

- `https://tinyclouds.org/colorize/`
- `http://blog.christianperone.com/2016/01/convolutional-hypercolumns-in-python/`

2 Hypercolumns

Many algorithms using features from CNNs (Convolutional Neural Networks) usually use the last FC (fully-connected) layer features in order to extract information about certain input. However, the information in the last FC layer may be too coarse spatially to allow precise localization (due to sequences of maxpooling, etc.), on the other side, the first layers may be spatially precise but will lack semantic information. To get the best of both worlds, the authors define the hypercolumn of a pixel as the vector of activations of all CNN units above that pixel.

The first step on the extraction of the hypercolumns is to feed the image into the CNN (Convolutional Neural Network) and extract the feature map activations for each location of the image. The tricky part is when the feature maps are smaller than the input image, for instance after a pooling operation, the authors of the paper then do a bilinear upsampling of the feature map in order to keep the feature maps on the same size of the input. There are also the issue with the FC (fully-connected) layers, because you cant isolate units semantically tied only to one pixel of the image, so the FC activations are seen as 11 feature maps, which means that all locations shares the same information regarding the FC part of the hypercolumn. All these activations are then concatenated to create the hypercolumn. For instance, if we take the VGG-16:



This means that each pixel of the image will have a 192-dimension hypercolumn vector. This hypercolumn is really interesting because it will contain information about the first layers (where we have a lot of spatial information but little semantic) and also information about the final layers (with little spatial information and lots of semantics). Thus this hypercolumn will certainly help in a lot of pixel classification.