Learning_Deconvolution_Network

Learning Deconvolution Network for Semantic Segmentation

CVPR 2015

2360+ citations

- FCN uses bilinear interpolation, for pixel-level labeling.
- CRF(Conditional Random Field) is optionally applied to the output map for fine segmentation to improve the boundary region of each prediction.

Crititical Limitations of Semantic Segmentation in FCN(paper):

- The network can handle only a single scale semantics within image due to the fixed-size receptive field.
- Therefore objects that are substantially larger or smaller than the receptive field may be fragmented or mislabeled.
- Because label prediction is done with only local information for large objects and the pixels that belong to the same object may have inconsistent labels.
- If the object is larger than the receptive field at the prediction layer(encoder output), then it will have disconnected blobs forming the predictions.
- If the object is smaller than the receptive field at the prediction layer(encoder output), then it may not get classified due to the influence of nearby pixels of other class.
- The detailed structures of an object are often lost or smoothed because the label map, input to the deconvolutional layer is too coarse and deconvolution procedure is overly simple.

(a) Inconsistent labels due to large object size

(b) Missing labels due to small object size

Contributions of the paper:

- *Learn a multi-layer deconvolution network, which is composed of deconvolution, upsampling and ReLU layers.*
- *Individual object proposals to obtain instance-wise segmentations, which are combined to form final semantic segmentation.*
- Deconvolution network is introduced to recontruct input images.
- Unpooling is used by utilizing pooled locations.
- Using the deconvolution network, the input image can be reconstructed from its feature representation.
- Deconvolution helps in visualizing activated features in trained CNN.

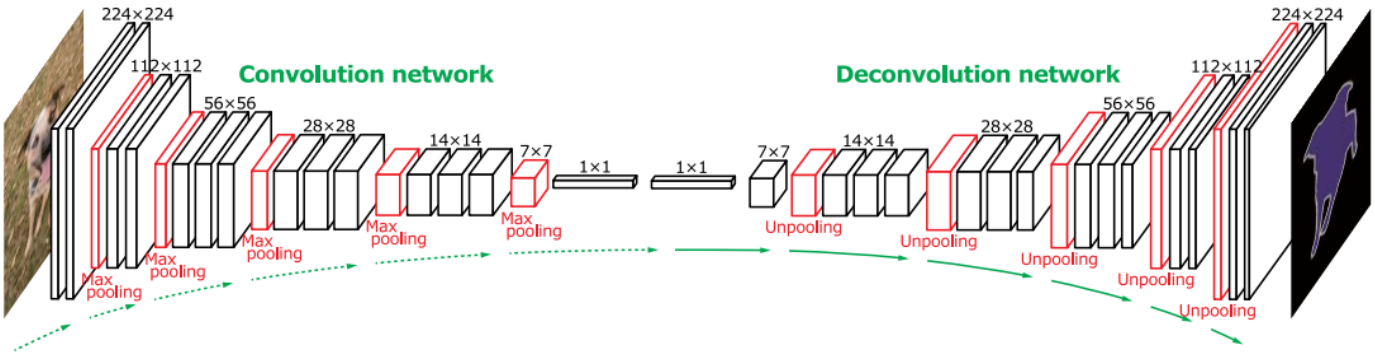

Figure 2. Overall architecture of the proposed network. On top of the convolution network based on VGG 16-layer net, we put a multi-layer deconvolution network to generate the accurate segmentation map of an input proposal. Given a feature representation obtained from the convolution network, dense pixel-wise class prediction map is constructed through multiple series of unpooling, deconvolution and rectification operations.

- 2 parts: Convolution and Deconvolution Networks.

- Convolution: Feature extractor; converts input image to multidimensional feature representation.
- Deconvolution: Shape generator; converts multidimensional feature representation to Object segmentation.
- Final output: probability map in the same size as the input image, indicating probability of each pixel that belongs to one of the predefined class.
- Encoding: Convolution,Pooling, and Rectification Layers
- Decoding: Deconvolution, Unpooling, and Rectification Layers

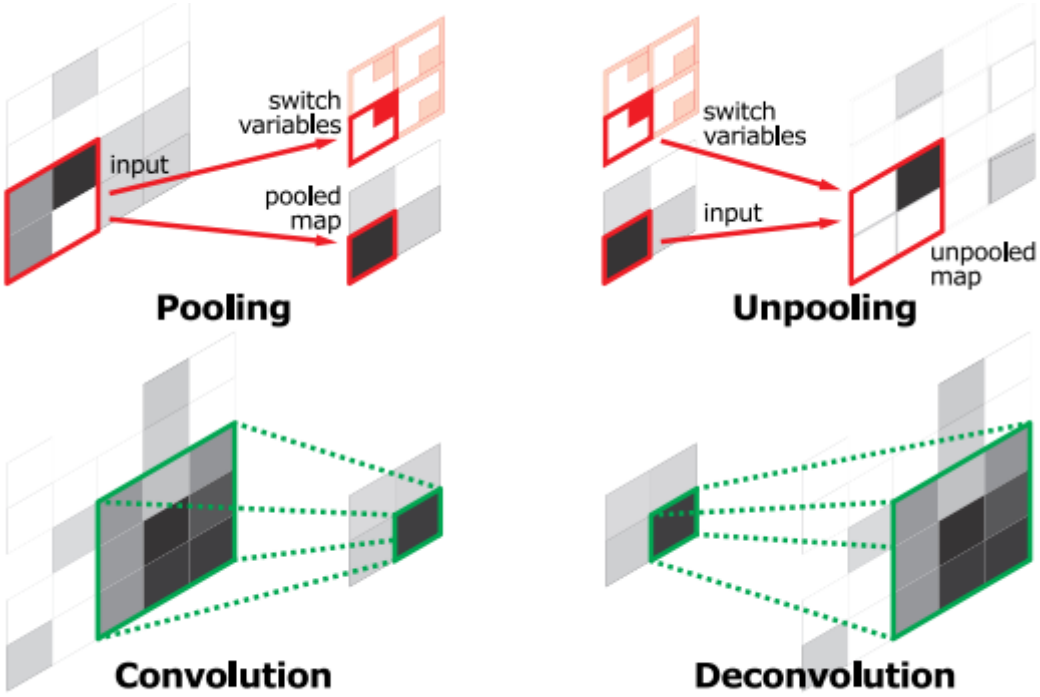

Figure 3. Illustration of deconvolution and unpooling operations.

- **Unpooling**:
  - *Pooling*: Used mainly to filter noisy activations in a lower layer by abstracting activations in a receptive field with a single representative value.
  - By **Pooling**, we lose spatial information within a receptive field.
  - **Unpooling**: The reverse of pooling to reconstruct the origina size of activations.
  - **Unpooling**: This is done using reusing the max-pool indices generated during pooling.
  - Max-pool indices are also called as switch variables.
- **Deconvolution**:
  - The output of an unpooling layer is an enlarged, but sparse activation map.
  - **The deconvolution layers densify the sparse activations obtained after unpooling through *convolution-like* operations with multiple learned filters.**
  - **Deconvolution layers associate a single input activation with multiple outputs. The output of the deconvolutional layer is an enlarged and dense activation map. We crop the bounday of the enlarged activatio map to keep the size of the output map identical to the one from the preceding unpooling layer.**

- The learned filteres in the deconv layers correspond to bases to reconstruct shape of an input object.
- Similar to convolution netowrk, the hierarchial structure of the deconv layers are used to capture different level of **shape details**.
- The filters in lower layers tend to caputre overall shape of an object while the class-specific fine details are encoded in the filters in higher layers.

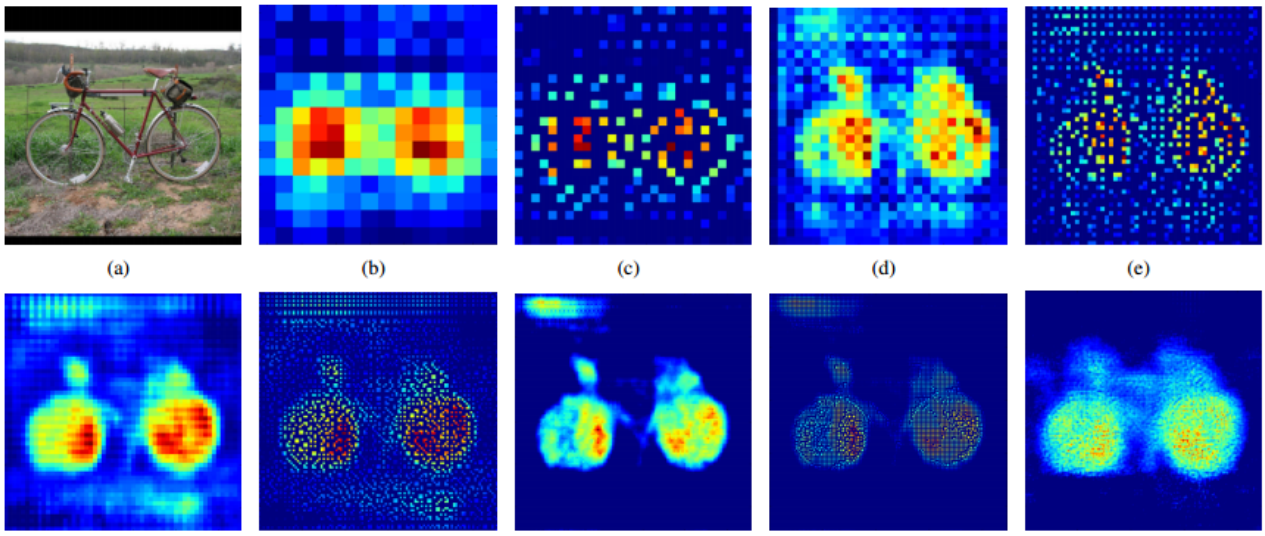

Figure 4. Visualization of activations in our deconvolution network. The activation maps from top left to bottom right correspond to the output maps from lower to higher layers in the deconvolution network. We select the most representative activation in each layer for effective visualization. The image in (a) is an input, and the rest are the outputs from (b) the last 14×14 deconvolutional layer, (c) the 28×28 unpooling layer, (d) the last 28×28 deconvolutional layer, (e) the 56×56 unpooling layer, (f) the last 56×56 deconvolutional layer, (g) the 112×112 unpooling layer, (h) the last 112×112 deconvolutional layer, (i) the 224×224 unpooling layer and (j) the last 224×224 deconvolutional layer. The finer details of the object are revealed, as the features are forward-propagated through the layers in the deconvolution network. Note that noisy activations from background are suppressed through propagation while the activations closely related to the target classes are amplified. It shows that the learned filters in higher deconvolutional layers tend to capture class-specific shape information.

Algorithm:

- Generate object segmentation masks using deep deconvolution network, where a dense pixel-wise class probability map is obtained by successive operations of unpooling, deconvolution, and rectification.

Note:

- Unpooling captures **example-specific** structures by tracing the original locations with strong activations back to image space.
- Deconvolution capture **class-specific** shapes
- Through deconvolutions, the activations closely related to the target classes are amplified while noisy acitivations from other regions are suppressed effectively.
- By the combination of unpooling and deconvolution, our network generates accurate segmentation maps.
- Instance-wise segmentation has a few advantages over image-level prediction. It handles object in various scales effectively and identifies fine details of object while the approaches with fixed-size receptive fields have troubles with these issues. It also reduces search space for prediction and reduces memory requirement for training.

Batch Normalization:

- Deep Learning is hard to optimize due to *internal-covariate-shift.*
- Input distributions in each layer change over iterations during training as the parameters of its previous layers are updated. This causes very deep networks to have the distribution shift being amplified through propagation through layers.
- **Batch Normalization** reduces the internal covariate shift by normalizing input distributions of every layer to the standard Gaussian distribution.
- We add batch-norm layer after every convolutional and deconvolutional layer.
- Without batch norm layer the network may end up in local minima.