Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation (2015)

Seunghoon Hong, Hyeonwoo Noh, Bohyung Han
Notes

1 Introduction

The authors propose a decouped architecture that learns seperate netwrok for both image classification using only image labels and segmentation with pixel wise labels, with a bridging layer that facilitates and reduces the search space for the segmentation by exploiting class-specific activation maps.

The proposed decoupled DNN is appropriate for a semi-supervised setting by exploiting heterogenous annotations with a small number of strong annotations as well as weak annotations, it contains two seperate networks, one for classification and the other for segmentation, each one is repsonsible for predicting the corresponding labels, and with the bridging layers that dilivers class specific information from the classification network to the segmentation network to enable it to focus on a single label indentified by the classification net at a time, each network is trainied seperately using the image and pixel level labels, and then the two are decoupled and gives the possibility to obtain good segmentation results with a very limited number of examples, i.e. 5 or 10.

2 Algorithm

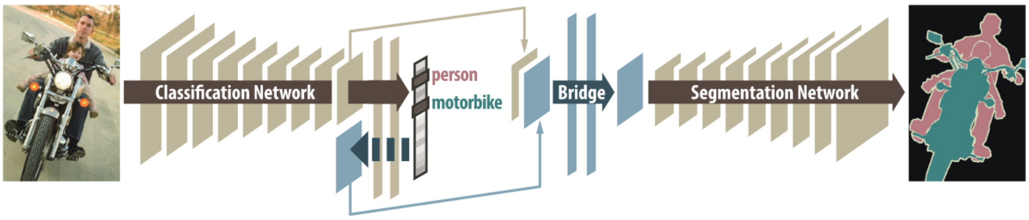


Figure 1: The architecture of the proposed network. While classification and segmentation networks are decoupled, bridging layers deliver critical information from classification network to segmentation network.

Given an input image, classification network identifies labels associated with the image, and segmentation network produces pixel-wise segmentation corresponding to each identified label, with a bridging layers between the two networks that delivers class-specific information from the classification network to segmentation network, the two networks are then optimized using seperate objective function with a deoupled task, first, given a large number of image lebeled examples, they train the classification network, and then using a small number of pixel wise images and extensive data augmentation, they optimize the segmentation network and bridging layers.

2.1 Architecture

Classification network Using a VGG-16 with a set of weights θ_c, the net takes an input image x and outputs a normalized score vector $S\left(\mathrm{x};\theta_c\right) \in \mathrm{R}^L$ as scores of the inputs for predefined classes

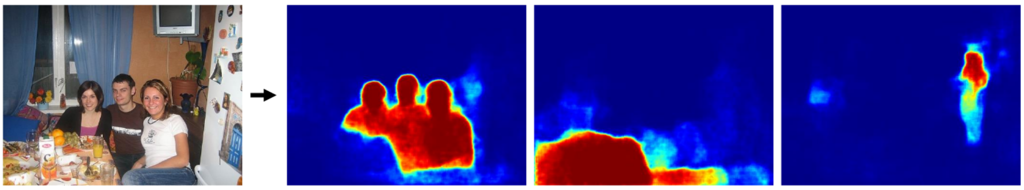
C, the objective of the classification network is writted as, with e_c is the classification loss and $\mathbf{y}_i$ is the one hot vector:

$$\min_{\theta_c} \sum_i e_c \left(\mathbf{y}_i, S\left(\mathbf{x}_i; \theta_c\right)\right)$$

Segmentation Network The segmentation network (A deconvolution network, the upsampling is done by the unpooling layers) takes as input class specific activation maps obtained by the bridging layer from the classification network, and output two channel class-specific segmentation masks (one for the foreground and one for the background) $M\left(\mathbf{g}_i^l; \theta_s\right)$, with a ground truth segmentation mask $\mathbf{z}_i^l$ for a class l of a given image, we can formulate the learning objective, which is a binary classification between the two ground truth (z_b and z_l) masks and the two predicted masks (M_b and M_l) for each image i:

$$\min_{\theta_s} \sum_i e_s \left(\mathbf{z}_i^l, M\left(\mathbf{g}_i^l; \theta_s\right)\right)$$

And here is an example of the output mask for each class, each one is a combination of the background and foreground mask by taking the max:

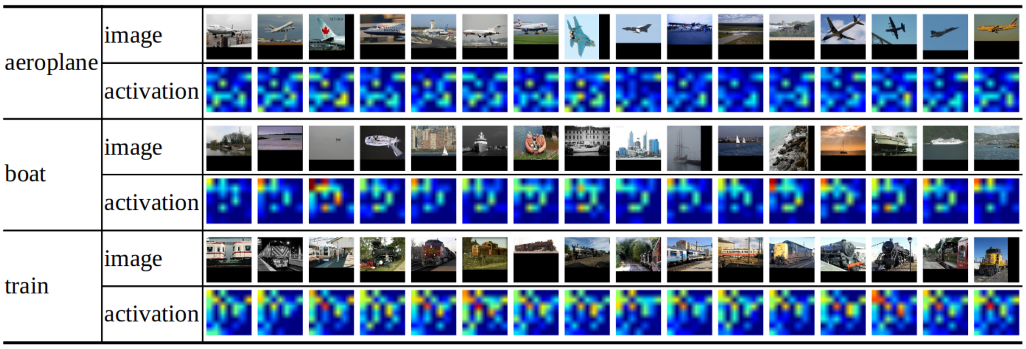


$\mathcal{L}_i^* = \{$person, table, plant$\}$ $M_f(\mathbf{g}_i^{\text{person}})$ $M_f(\mathbf{g}_i^{\text{table}})$ $M_f(\mathbf{g}_i^{\text{plant}})$

Figure 3: Input image (left) and its segmentation maps (right) of individual classes.

Bridging Layers The goal of the bridging layer is to encode spatial information of a specific class and pass it to the segmentation network, first the authors choose the activation of the *pool5* of the VGG16 that still preverve the spatial information of the input, this gives us the activations $\mathbf{f}_{\text{spat}}$ which are not class specific, to add class specific information, we create saliency maps, this is done by backpropagating the classification error for a given class l (during training, we choose the correct class) up to the pool5 layer we've chosen, and this given us $\mathbf{f}_{\text{cls}}^l$ which are class specific, we them concatenate both over the channel dimension and pass them through a fully connected layer to obtain the optimal combination of both, and pass the results to the segmentation network, during inference we get all the possible class spefici activation and pass them to the segmentation network, and obtain the segmentation results per class and using argmax we can get one mask for all classes.

And here we see the passes activations from the classification network to the segmentation network for a given class:



3 Experiments

Table 1: Evaluation results on PASCAL VOC 2012 validation set.

# of strongs	DecoupledNet	WSSL-Small_FoV [8]	WSSL-Large-FoV [8]	DecoupledNet-Str	DeconvNet [12]
Full	67.5	63.9	**67.6**	67.5	67.1
25 (×20 classes)	**62.1**	56.9	54.2	50.3	38.6
10 (×20 classes)	**57.4**	47.6	38.9	41.7	21.5
5 (×20 classes)	**53.1**	-	-	32.7	15.3

Table 2: Evaluation results on PASCAL VOC 2012 test set.

Models	bkg	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	prsn	plnt	sheep	sofa	train	tv	mean
DecoupledNet-Full	91.5	78.8	39.9	78.1	53.8	68.3	83.2	78.2	80.6	25.8	62.6	55.5	75.1	77.2	77.1	76.0	47.8	74.1	47.5	66.4	60.4	66.6
DecoupledNet-25	90.1	75.8	41.7	70.4	46.4	66.2	83.0	69.9	76.7	23.1	61.2	43.3	70.4	75.7	74.1	65.7	46.2	73.8	39.7	61.9	57.6	62.5
DecoupledNet-10	88.5	73.8	40.1	68.1	45.5	59.5	76.4	62.7	71.4	17.7	60.4	39.9	64.5	73.0	68.5	56.0	43.4	70.8	37.8	60.3	54.2	58.7
DecoupledNet-5	87.4	70.4	40.9	60.4	36.3	61.2	67.3	67.7	64.6	12.8	60.2	26.4	63.2	69.6	64.8	53.1	34.7	65.3	34.4	57.0	50.5	54.7