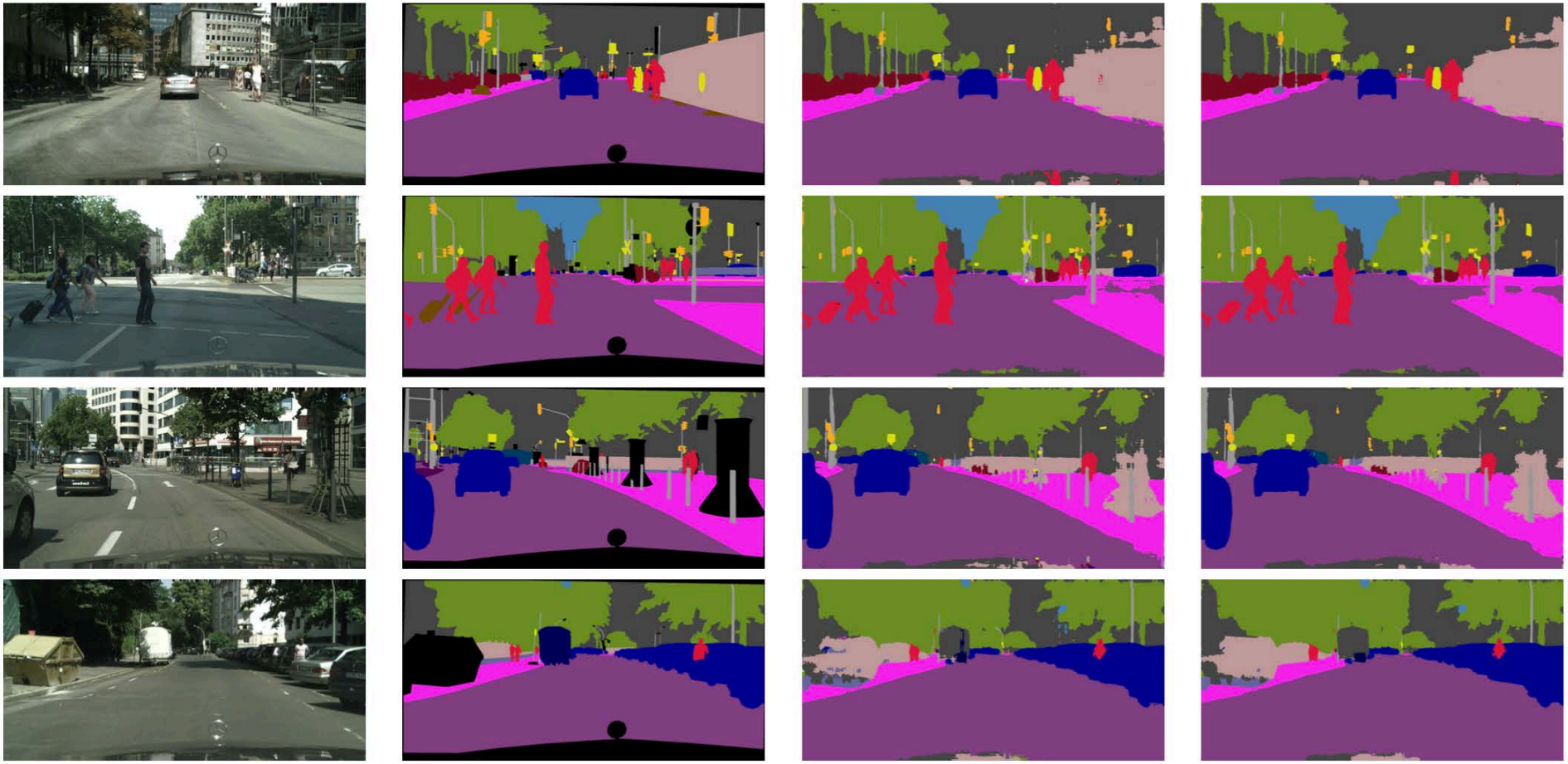
Semantic segmentation algorithms

Semantic segmentation



(a) Image (b) G.T. (c) Before CRF (d) After CRF

Leader board

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
DeepLabv2-CRF [?]	**79.7**	92.6	60.4	91.6	63.4	**76.3**	95.0	88.4	92.6	32.7	**88.5**	67.6	**89.6**	**92.1**	87.0	87.4	63.3	88.3	60.0	**86.8**	74.5	06-Jun-2016
CASIA_SegResNet_CRF_COCO [?]	79.3	**93.8**	42.2	93.1	68.6	75.3	**95.3**	**88.8**	92.5	36.5	84.3	64.2	86.8	87.8	87.5	**88.5**	**69.2**	89.7	64.1	**86.8**	**74.6**	03-Jun-2016
Adelaide_VeryDeep_FCN_VOC [?]	79.1	91.9	48.1	**93.4**	69.3	75.5	94.2	87.5	**92.8**	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	**90.1**	59.7	85.5	72.7	13-May-2016
LRR_4x_COCO [?]	78.7	93.2	44.2	89.4	65.4	74.9	93.9	87.0	92.0	42.9	83.7	**68.9**	86.5	88.0	**89.0**	87.2	67.3	85.6	64.0	84.1	71.5	16-Jun-2016
CASIA_IVA_OASeg [?]	78.3	**93.8**	41.9	89.4	67.5	71.5	94.6	85.3	89.5	38.1	88.4	64.8	87.0	90.5	84.9	83.3	67.5	86.9	**68.1**	83.4	74.0	21-May-2016
Oxford_TVG_HO_CRF [?]	77.9	92.5	59.1	90.3	**70.6**	74.4	92.4	84.1	88.3	36.8	85.6	67.1	85.1	86.9	88.2	82.6	62.6	85.0	56.3	81.9	72.5	16-Mar-2016
Adelaide_Context_CNN_CRF_COCO [?]	77.8	92.9	39.6	84.0	67.9	75.3	92.7	83.8	90.1	**44.3**	85.5	64.9	87.3	88.8	84.5	85.5	68.1	89.0	62.8	81.2	71.4	06-Nov-2015
CUHK_DPN_COCO [?]	77.5	89.0	**61.6**	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	22-Sep-2015
Adelaide_Context_CNN_CRF_COCO [?]	77.2	92.3	38.8	82.9	66.1	75.1	92.4	83.1	88.6	41.8	85.9	62.8	86.7	88.4	84.0	85.4	67.4	88.8	61.9	81.9	71.7	13-Aug-2015
DeepLab-CRF-Attention-DT [?]	76.3	93.2	41.7	88.0	61.7	74.9	92.9	84.5	90.4	33.0	82.8	63.2	84.5	85.0	87.2	85.7	60.5	87.7	57.8	84.3	68.2	03-Feb-2016
CentraleSuperBoundaries++ [?]	76.0	91.1	38.5	90.9	68.7	74.2	89.9	85.3	89.1	34.4	82.5	65.6	83.1	82.9	85.7	85.4	60.6	84.5	59.9	80.2	69.9	13-Jan-2016
LRR_4x_de_pyramid_VOC [?]	75.9	91.8	41.0	83.0	62.3	74.3	93.0	86.8	88.7	36.6	81.8	63.4	84.7	85.9	85.1	83.1	62.0	84.6	55.6	84.9	70.0	07-Jun-2016
DeepLab-CRF-Attention [?]	75.7	91.1	40.9	86.9	62.1	74.2	92.3	84.4	90.1	34.0	81.7	66.0	83.5	83.9	86.5	84.6	59.1	87.2	59.6	81.0	66.2	03-Feb-2016
CentraleSuperBoundaries [?]	75.7	90.3	37.9	89.6	67.8	74.6	89.3	84.1	89.1	35.8	83.6	66.2	82.9	81.7	85.6	84.6	60.3	84.8	60.7	78.3	68.3	01-Dec-2015
Adelaide_Context_CNN_CRF_VOC [?]	75.3	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	30-Aug-2015
MSRA_BoxSup [?]	75.2	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	18-May-2015
POSTECH_DeconvNet_CRF_VOC [?]	74.8	90.0	40.8	84.2	67.3	70.7	90.9	84.8	87.4	34.8	83.0	58.7	82.3	87.1	86.9	82.4	64.5	84.6	54.9	77.5	64.1	18-Aug-2015
MERL_UMD_Deep_GCRF_COCO [?]	74.8	89.9	42.6	90.0	65.0	69.2	89.9	83.9	88.2	31.3	81.8	66.4	82.9	81.1	85.7	83.4	58.4	88.4	56.7	77.7	64.3	15-Jan-2016
Oxford_TVG_CRF_RNN_COCO [?]	74.7	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	22-Apr-2015
DeepLab-MSc-CRF-LargeFOV-COCO-CrossJoint [?]	73.9	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	26-Apr-2015
MERL_DEEP_GCRF [?]	73.2	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	17-Oct-2015
Bayesian Dilation Network [?]	73.1	88.6	39.0	86.2	63.3	67.1	88.1	81.9	86.8	34.7	81.1	57.1	81.3	86.5	83.4	83.4	53.7	84.0	53.3	80.5	62.5	07-Jun-2016
DeepLab-CRF-COCO-LargeFOV [?]	72.7	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	18-Mar-2015
POSTECH_EDeconvNet_CRF_VOC [?]	72.5	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	22-Apr-2015
CCBM [?]	72.3	87.8	46.7	79.0	63.6	70.5	83.7	75.5	86.9	31.0	81.9	61.3	81.5	85.9	81.1	76.5	58.7	77.7	50.4	76.6	69.8	29-Nov-2015

Introduction

Fully Convolutional Networks for Semantic Segmentation

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley
{jonlong,shelhamer,trevor}@cs.berkeley.edu

Abstract

Convolutional networks are powerful visual models that yield hierarchies of features. We show that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation. Our key insight is to build "fully convolutional" networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. We define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. We adapt contemporary classification networks (AlexNet [22], the VGG net [34], and GoogLeNet [35]) into fully convolutional networks and transfer their learned representations by fine-tuning [5] to the segmentation task. We then define a skip architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed seg-

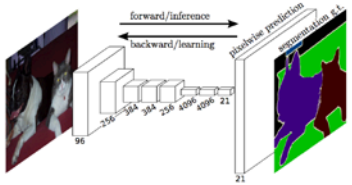


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

We show that a fully convolutional network (FCN) trained end-to-end, pixels-to-pixels on semantic segmentation exceeds the state-of-the-art without further machinery. To our knowledge, this is the first work to train FCNs end-to-end (1) for pixelwise prediction and (2) from supervised pre-training. Fully convolutional versions of existing networks predict dense outputs from arbitrary-sized inputs

Published as a conference paper at ICLR 2015

SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS

Liang-Chieh Chen
Univ. of California, Los Angeles
lcchen@cs.ucla.edu

George Papandreou *
Google Inc.
gpapan@google.com

Iasonas Kokkinos
CentraleSupélec and INRIA
iasonas.kokkinos@ecp.fr

Kevin Murphy
Google Inc.
kpmurphy@google.com

Alan L. Yuille
Univ. of California, Los Angeles
yuille@stat.ucla.edu

Learning Deconvolution Network for Semantic Segmentation

Hyeonwoo Noh Seunghoon Hong Bohyung Han
Department of Computer Science and Engineering, POSTECH, Korea
{hyeonwoonoh,maga33,bhhan}@postech.ac.kr

Abstract

We propose a novel semantic segmentation algorithm by learning a deep deconvolution network. We learn the network on top of the convolutional layers adopted from VGG 16-layer net. The deconvolution network is composed of deconvolution and unpooling layers, which identify pixel-wise class labels and predict segmentation masks. We apply the trained network to each proposal in an input image, and construct the final semantic segmentation map by combining the results from all proposals in a simple manner. The proposed algorithm mitigates the limitations of the existing methods based on fully convolutional networks by integrating deep deconvolution network and proposal-wise prediction; our segmentation method typically identifies detailed structures and handles objects in multiple scales naturally. Our network demonstrates outstanding performance in PASCAL VOC 2012 dataset, and we achieve the best accuracy (72.5%) among the methods trained without using Microsoft COCO dataset through ensemble with the fully convolutional network

(a) Inconsistent labels due to large object size

(b) Missing labels due to small object size

Figure 1. Limitations of semantic segmentation algorithms based on fully convolutional network. (Left) original image. (Center) ground-truth annotation. (Right) segmentations by [19]

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

Liang-Chieh Chen, George Papandreou, *Senior Member, IEEE*, Iasonas Kokkinos, *Member, IEEE*, Kevin Murphy, and Alan L. Yuille, *Fellow, IEEE*

Abstract—In this work we address the task of semantic image segmentation with Deep Learning and make three main contributions that are experimentally shown to have substantial practical merit. *First*, we highlight convolution with upsampled filters, or 'atrous convolution', as a powerful tool in dense prediction tasks. Atrous convolution allows us to explicitly control the resolution at which feature responses are computed within Deep Convolutional Neural Networks. It also allows us to effectively enlarge the field of view of filters to incorporate larger context without increasing the number of parameters or the amount of computation. *Second*, we propose atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales. ASPP probes an incoming convolutional feature layer with filters at multiple sampling rates and effective fields-of-views, thus capturing objects as well as image context at multiple scales. *Third*, we improve the localization of object boundaries by combining methods from DCNNs and probabilistic graphical models. The commonly deployed combination of max-pooling and downsampling in DCNNs achieves invariance but has a toll on localization accuracy. We overcome this by combining the responses at the final DCNN layer with a fully connected Conditional Random Field (CRF), which is shown both qualitatively and quantitatively to improve localization performance. Our proposed "DeepLab" system sets the new state-of-art at the PASCAL VOC-2012 semantic image segmentation task, reaching 79.7% mIOU in the test set, and advances the results on three other datasets: PASCAL-Context, PASCAL-Person-Part, and Cityscapes. All of our code is made publicly available online.

Index Terms—Convolutional Neural Networks, Semantic Segmentation, Atrous Convolution, Conditional Random Fields.

Fully Convolutional Networks for Semantic Segmentation

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley
{jonlong,shelhamer,trevor}@cs.berkeley.edu

Abstract

Convolutional networks are powerful visual models that yield hierarchies of features. We show that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation. Our key insight is to build "fully convolutional" networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. We define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. We adapt contemporary classification networks (AlexNet [22], the VGG net [34], and GoogLeNet [35]) into fully convolutional networks and transfer their learned representations by fine-tuning [5] to the segmentation task. We then define a skip architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed seg-

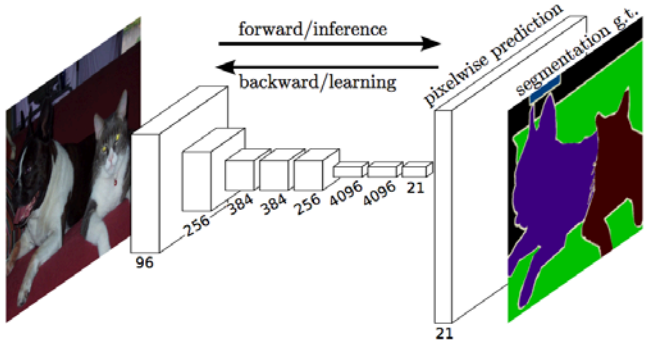


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

We show that a fully convolutional network (FCN) trained end-to-end, pixels-to-pixels on semantic segmentation exceeds the state-of-the-art without further machinery. To our knowledge, this is the first work to train FCNs end-to-end (1) for pixelwise prediction and (2) from supervised pre-training. Fully convolutional versions of existing networks predict dense outputs from arbitrary-sized inputs.

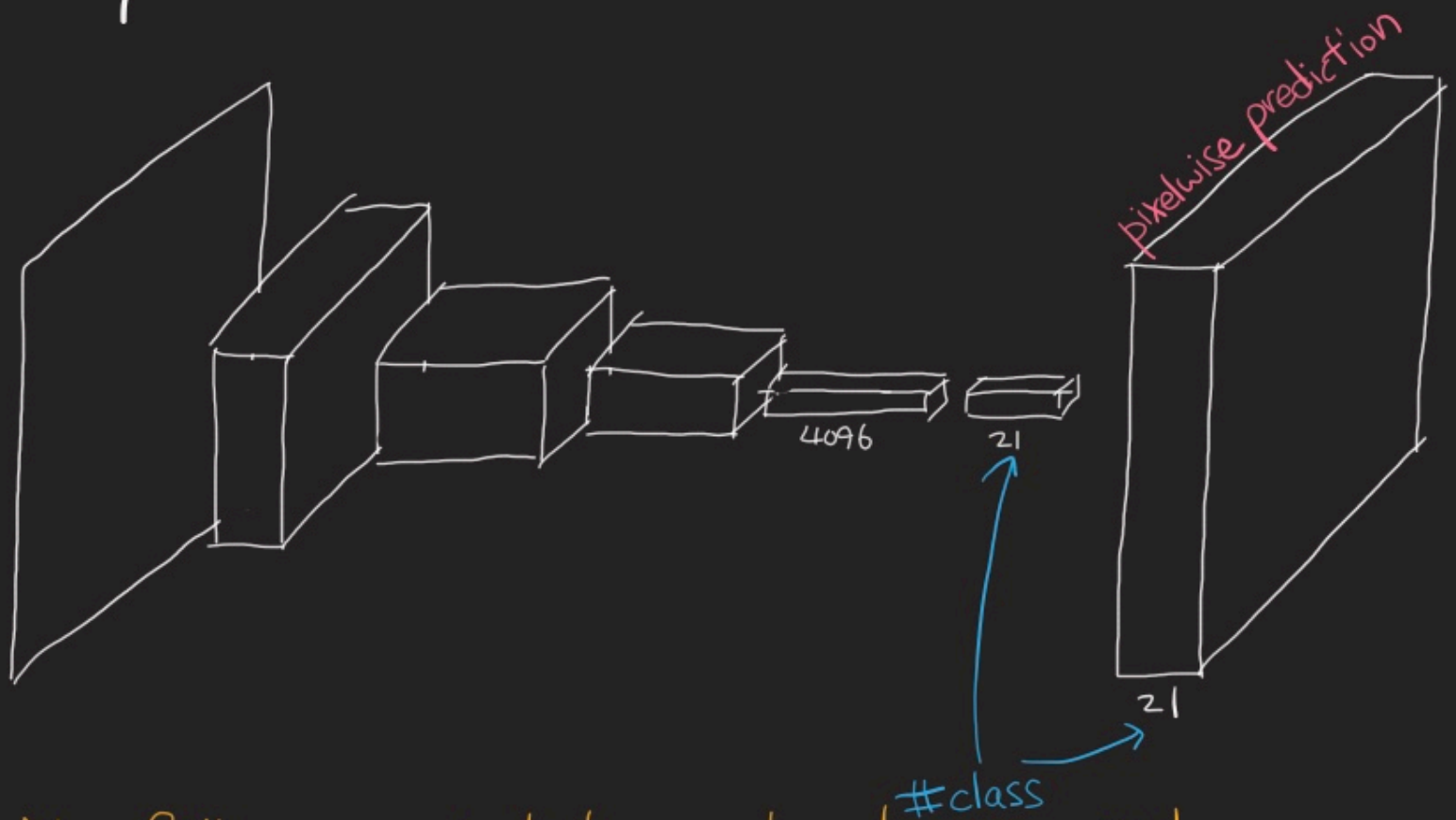
Fully Convolutional Networks for Semantic Segmentation

Fully convolutional network (FCN)

First pixelwise prediction with end to end learning

Fully convolutional versions of existing networks predict dense outputs from arbitrary-sized inputs.

Fully convolutional networks



No fully connected layer has been used.

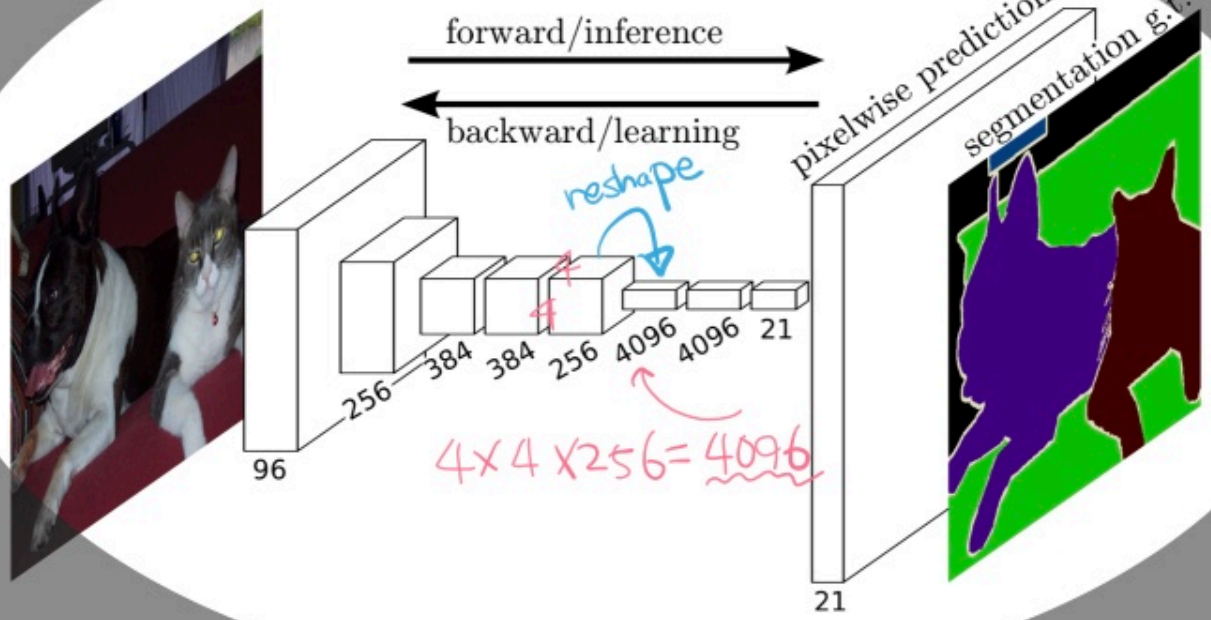
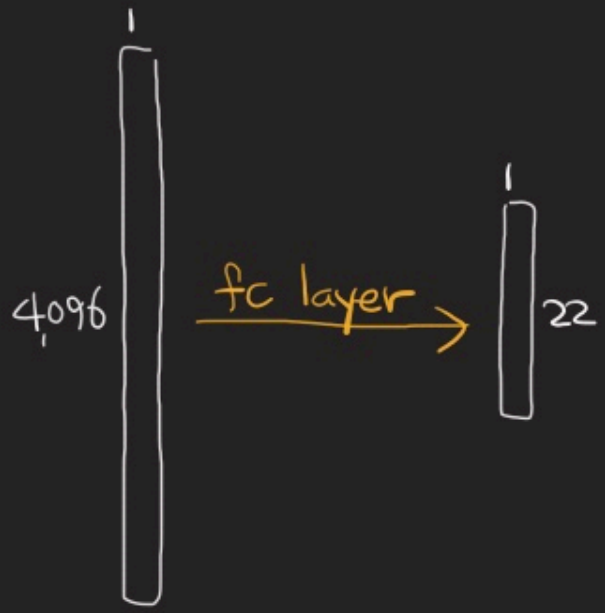


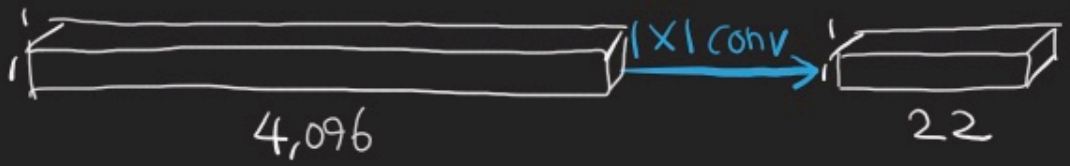
Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

We show that a fully convolutional network (FCN)



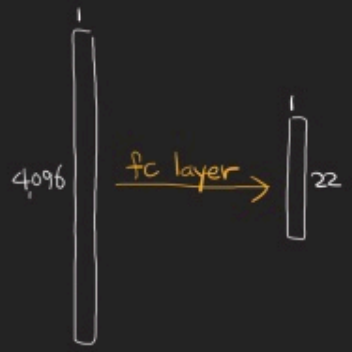
The number of parameters in the fc layer is

⇒ 4,096 × 22



The number of parameters in the 1×1 conv is

⇒ 1×1×4,096×22



The number of parameters in the fc layer is

⇒ 4,096 x 22

The number of parameter in the 1x1 conv is

⇒ 1x1x 4,096 x 22

In other words, fully connected layer is identical to the 1x1 convolution from 1x1 inputs to 1x1 outputs.

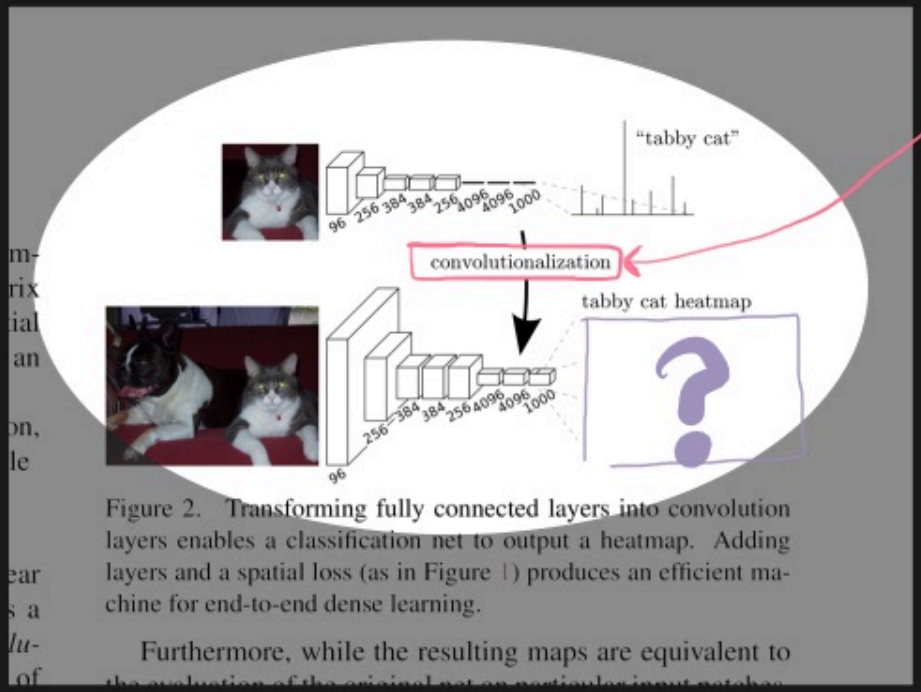


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

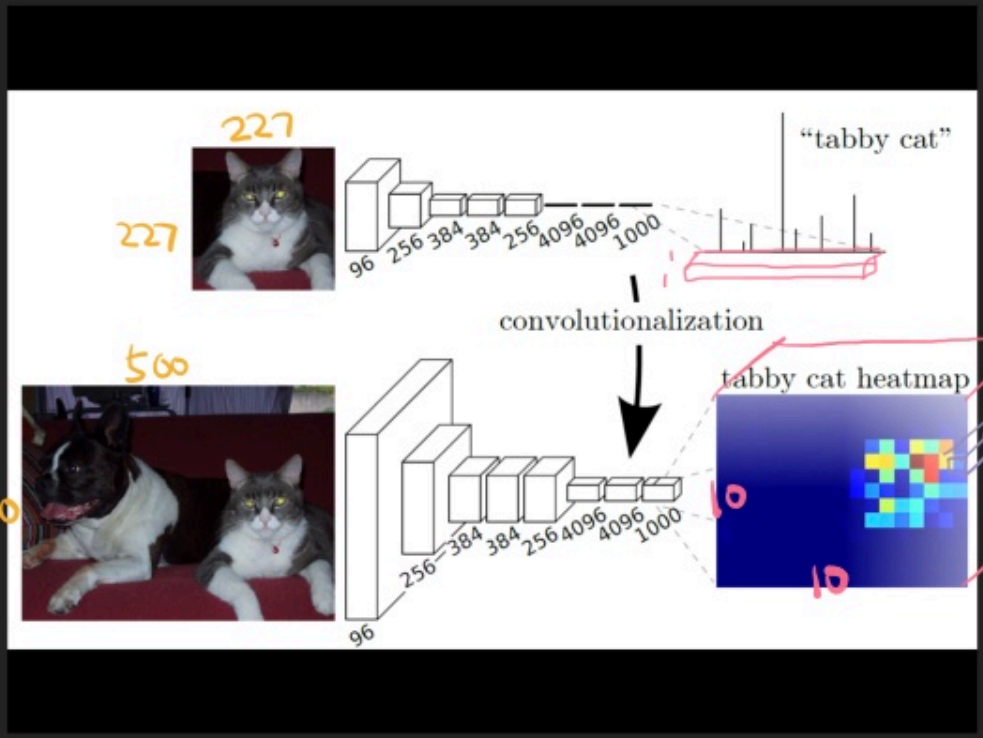
Furthermore, while the resulting maps are equivalent to

Convolutionalization

: Use 1x1 convolution instead of fully connected layer.

why?

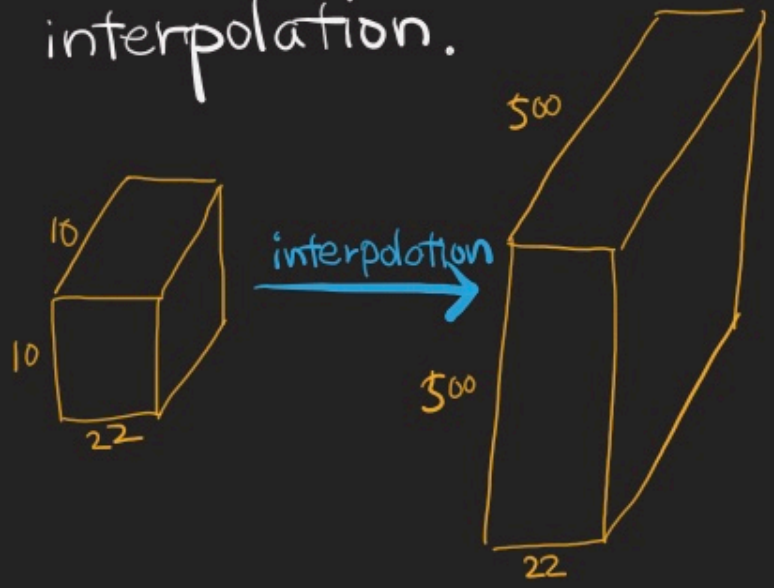
Transforming fully connect layers into convolution layers enables a classification net to output a heatmap.



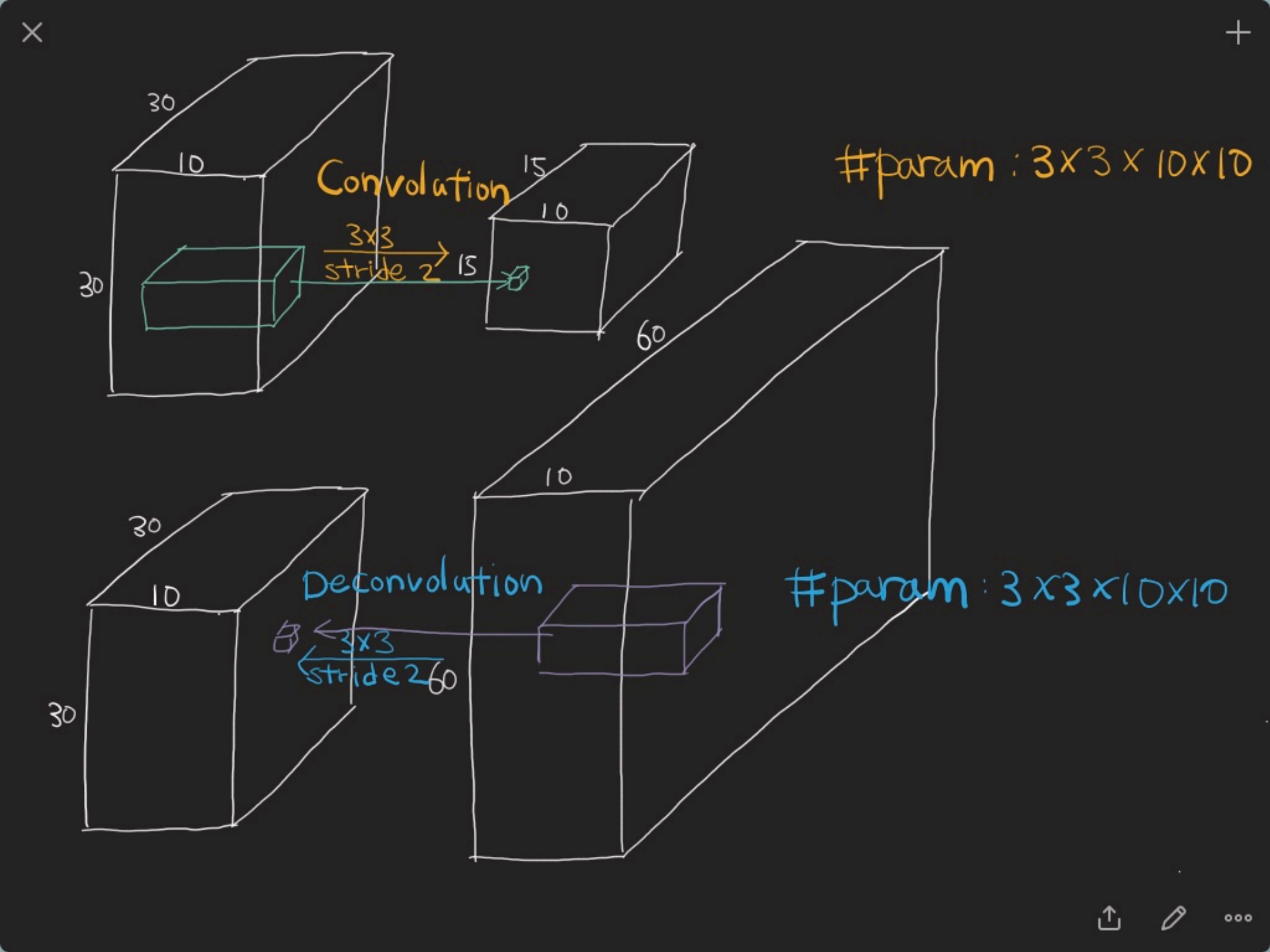
Run classification at each pixel!

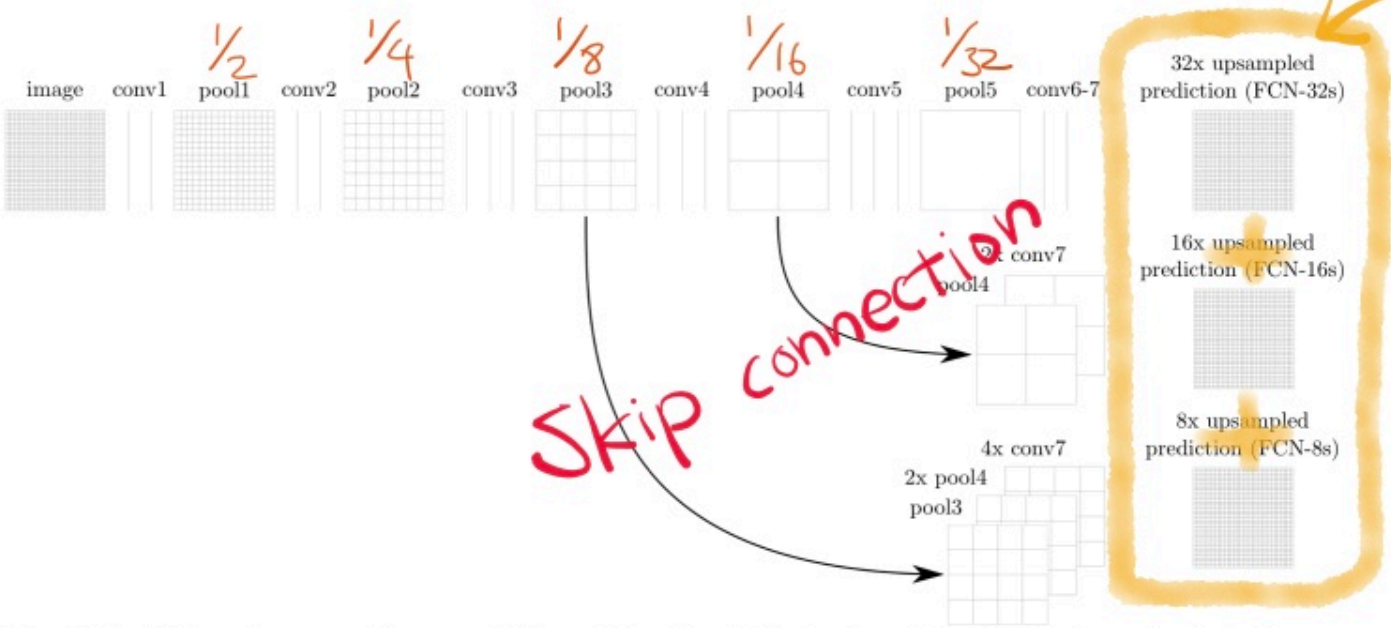
While the fully convolutional network can run with inputs of any size, the output dimensions are typically reduced by subsampling.

Another way to connect course output to dense pixels is interpolation.



Deconvolution is used for this purpose.


30
10
30
Convolution
3x3
stride 2
15
15
10
#param : 3x3x10x10
60
10
30
10
30
Deconvolution
3x3
stride 2
60
#param : 3x3x10x10



Dimensions are the same

Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines. First row (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Second row (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Third row (FCN-8s): Additional predictions from `pool3`, at stride 8, provide further precision.

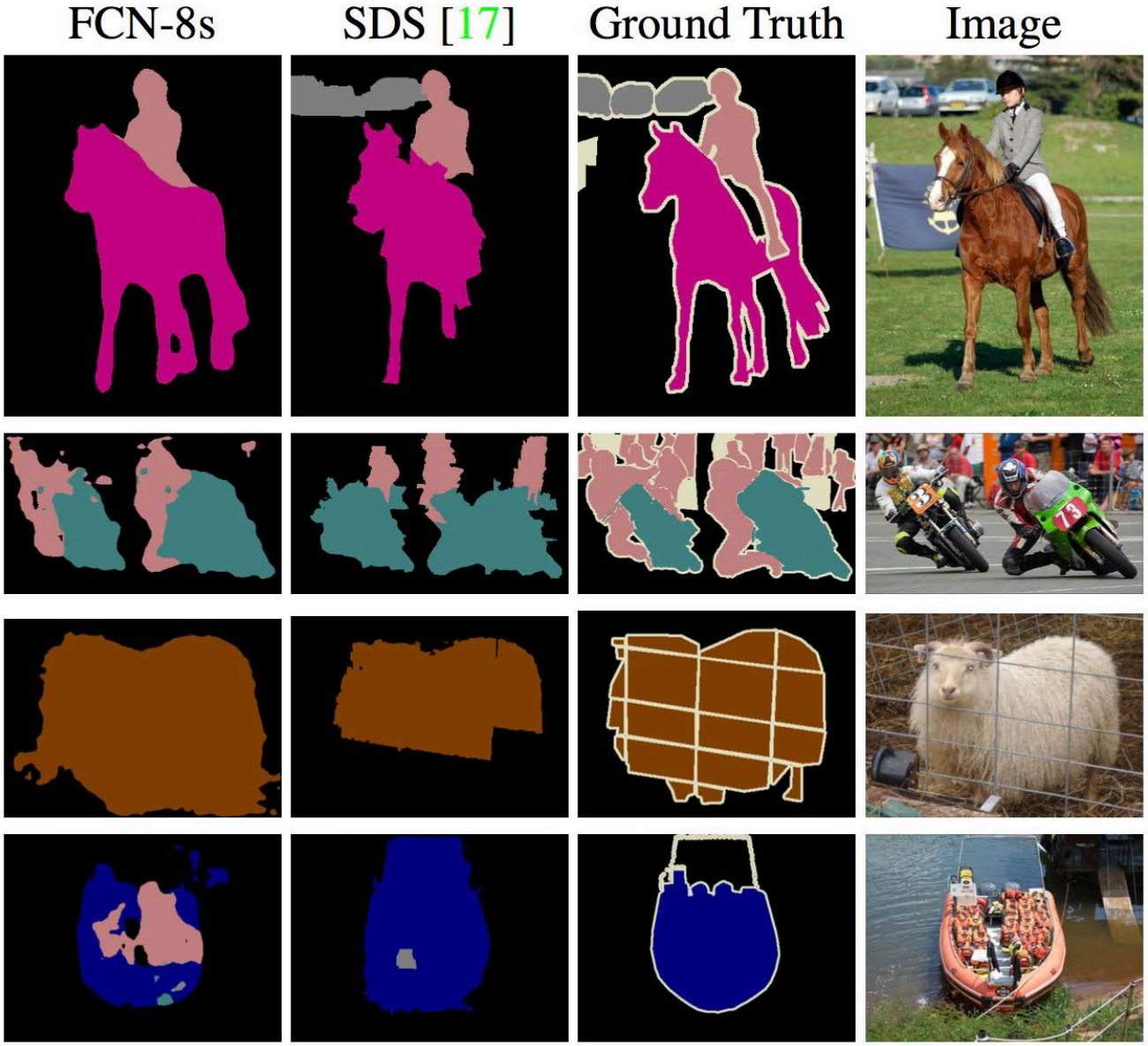
4.1. From classifier to dense FCN

We begin by convolutionalizing proven classification architectures as in Section 3. We consider the AlexNet[3] ar-

Table 1. We adapt and extend three classification convnets. We compare performance by mean intersection over union on the validation set of PASCAL VOC 2011 and by inference time (averaged over 20 trials for a 500 × 500 input on an NVIDIA Tesla K40c).

Performance increases as more predictions are used.

Results



SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS

Liang-Chieh Chen
Univ. of California, Los Angeles
lcchen@cs.ucla.edu

George Papandreou *
Google Inc.
gpapan@google.com

Iasonas Kokkinos
CentraleSupélec and INRIA
iasonas.kokkinos@ecp.fr

Kevin Murphy
Google Inc.
kpmurphy@google.com

Alan L. Yuille
Univ. of California, Los Angeles
yuille@stat.ucla.edu

Two problems

1. Signal downsampling

2. Spatial insensitivity

Two problems

1. Signal downsampling

   for efficient computation

   Comes from subsampling

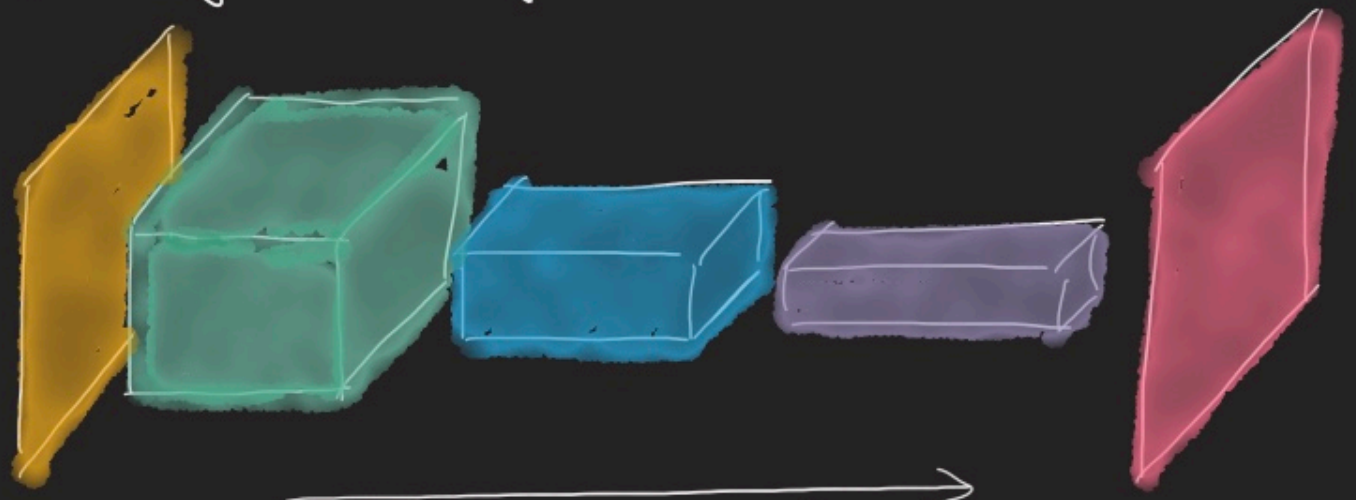
   → Atrous (with holes) algorithm

2. Spatial insensitivity

   Spatial invariance makes
   classification performance ↑
   spatial accuracy ↓

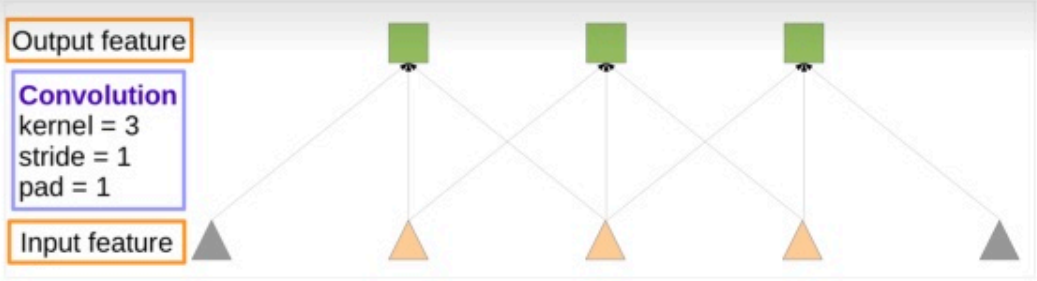
   → Conditional random field

Hole algorithm (Atrous convolution)

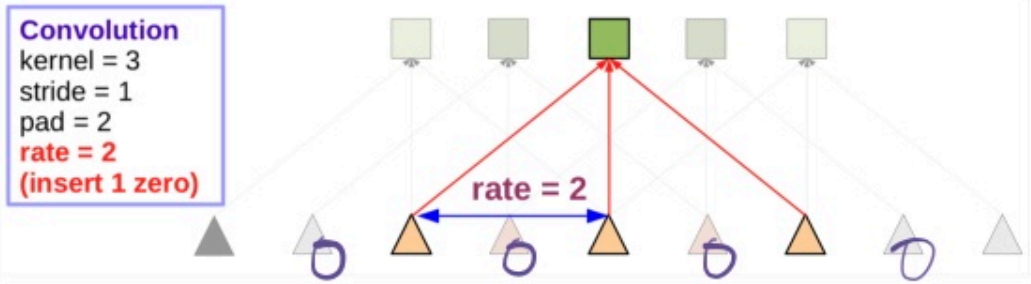


Maxpooling looses spatial information

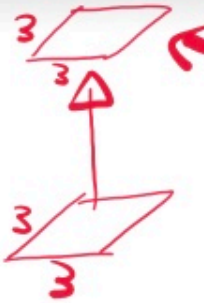
We want to get dense CNN feature maps



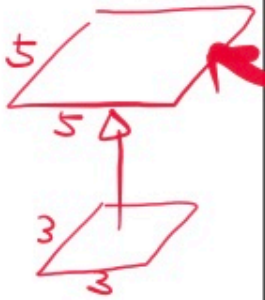
(a) Sparse feature extraction



(b) Dense feature extraction



low resolution



high resolution

Fig. 2: Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

This approach allows us to efficiently compute dense CNN feature maps at any target subsampling rate.

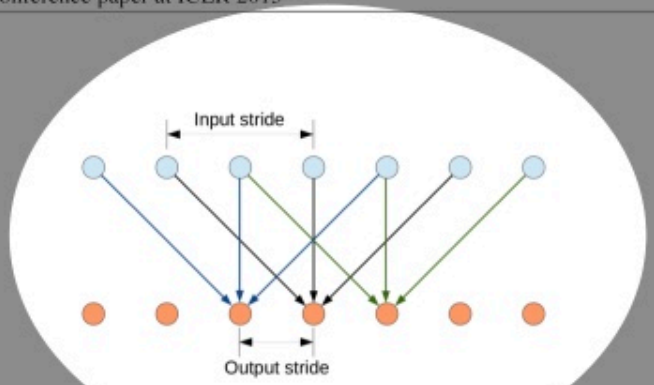


Figure 1: Illustration of the hole algorithm in 1-D, when *kernel_size* = 3, *input_stride* = 2, and *output_stride* = 1.

the last three convolutional layers and 4× in the first fully connected layer). We can implement this more efficiently by keeping the filters intact and instead sparsely sample the feature maps on which they are applied on using an input stride of 2 or 4 pixels, respectively. This approach, illustrated in Fig. 1 is known as the 'hole algorithm' ('atrous algorithm') and has been developed before for efficient computation of the undecimated wavelet transform (Mallat, 1999). We have implemented this within the Caffe framework (Jia et al., 2014) by adding to the *im2col* function (it converts multi-

Input stride

≧ Output stride

⇓

Make more dense feature map.

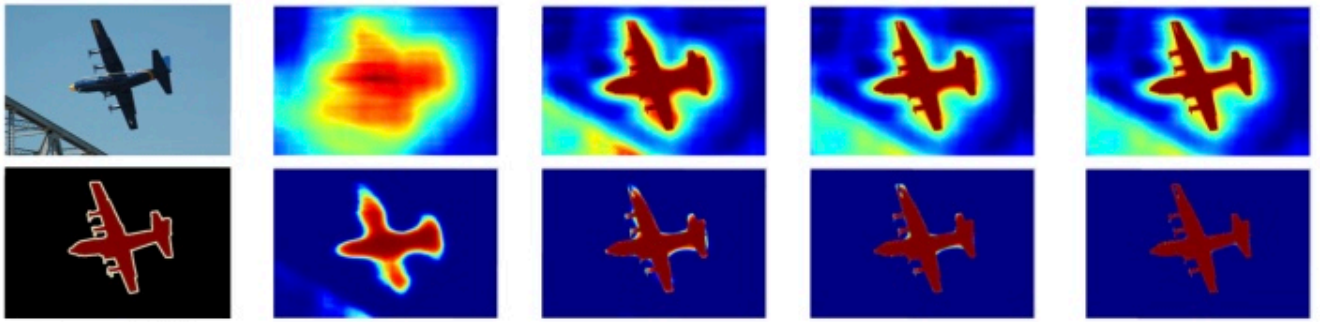
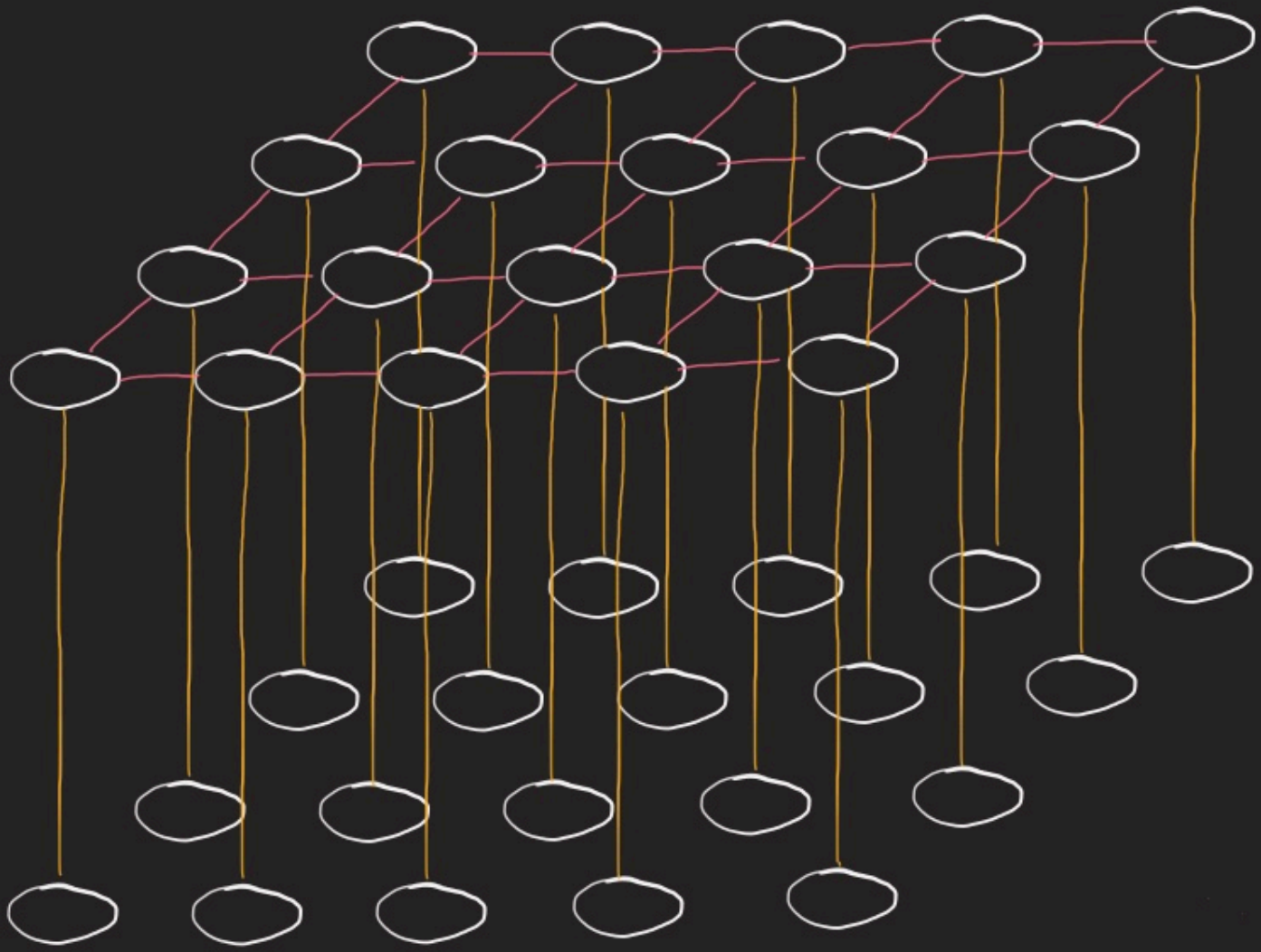


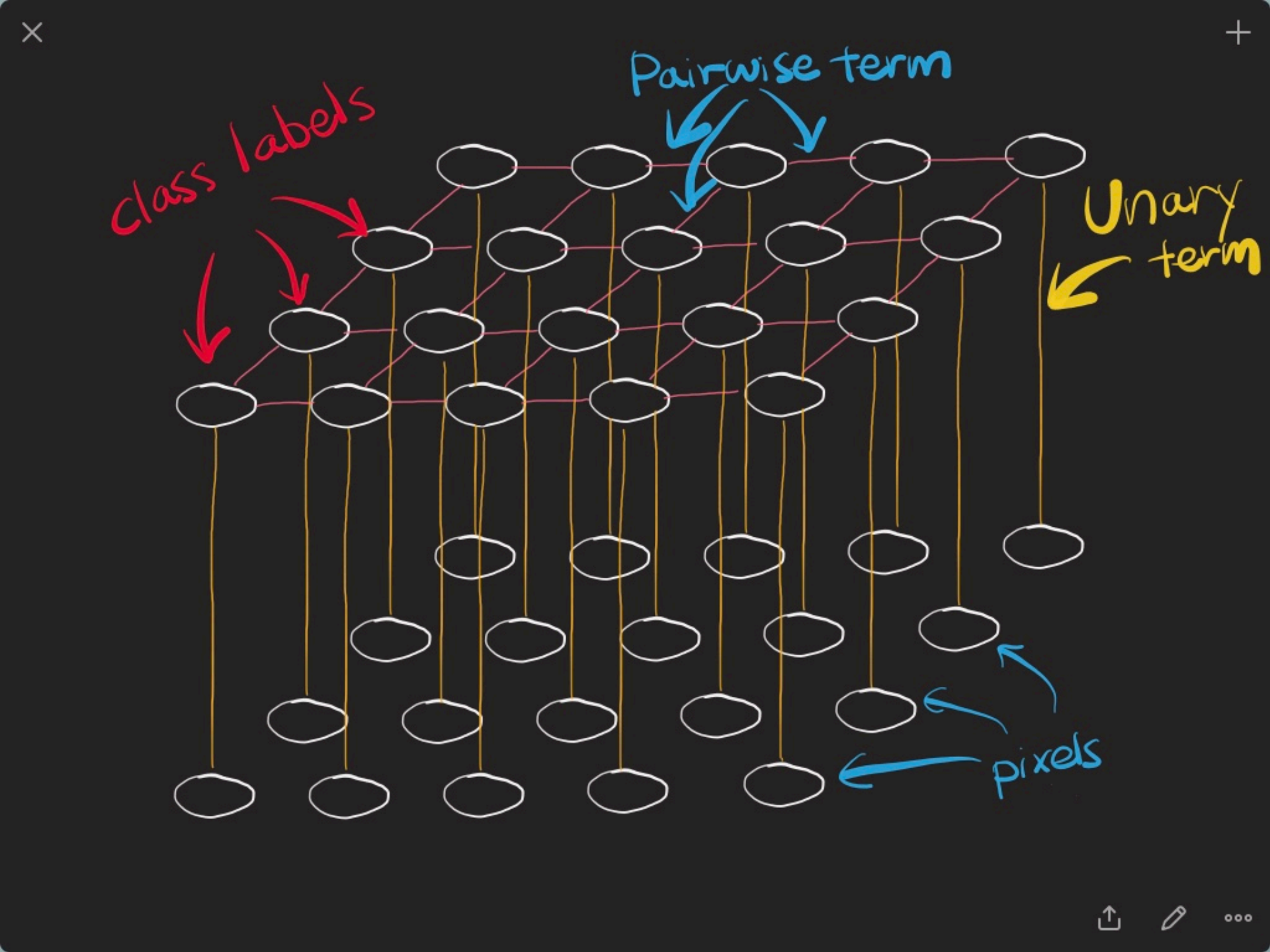
Figure 2: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane. We show the score (1st row) and belief (2nd row) maps after each mean field iteration. The output of last DCNN layer is used as input to the mean field inference. Best viewed in color.

Traditionally, conditional random fields (CRFs) have been employed to smooth noisy segmentation maps (Rother et al., 2004; Kohli et al., 2009). Typically these models contain energy terms that couple neighboring nodes, favoring same-label assignments to spatially proximal pixels. Qualitatively, the primary function of these short-range CRFs has been to clean up the spurious predictions of weak classifiers built on top of local hand-engineered features.

Compared to these weaker classifiers, modern DCNN architectures such as the one we use in this work produce score maps and semantic label predictions which are qualitatively different. As illustrated in Figure 2, the score maps are typically quite smooth and produce homogeneous classification results. In this regime, using short-range CRFs can be detrimental, as our goal should be to recover detailed local structure rather than further smooth it. Using contrast-sensitive potentials (Rother

Conditional random field




Pairwise term
class labels
Unary term
pixels

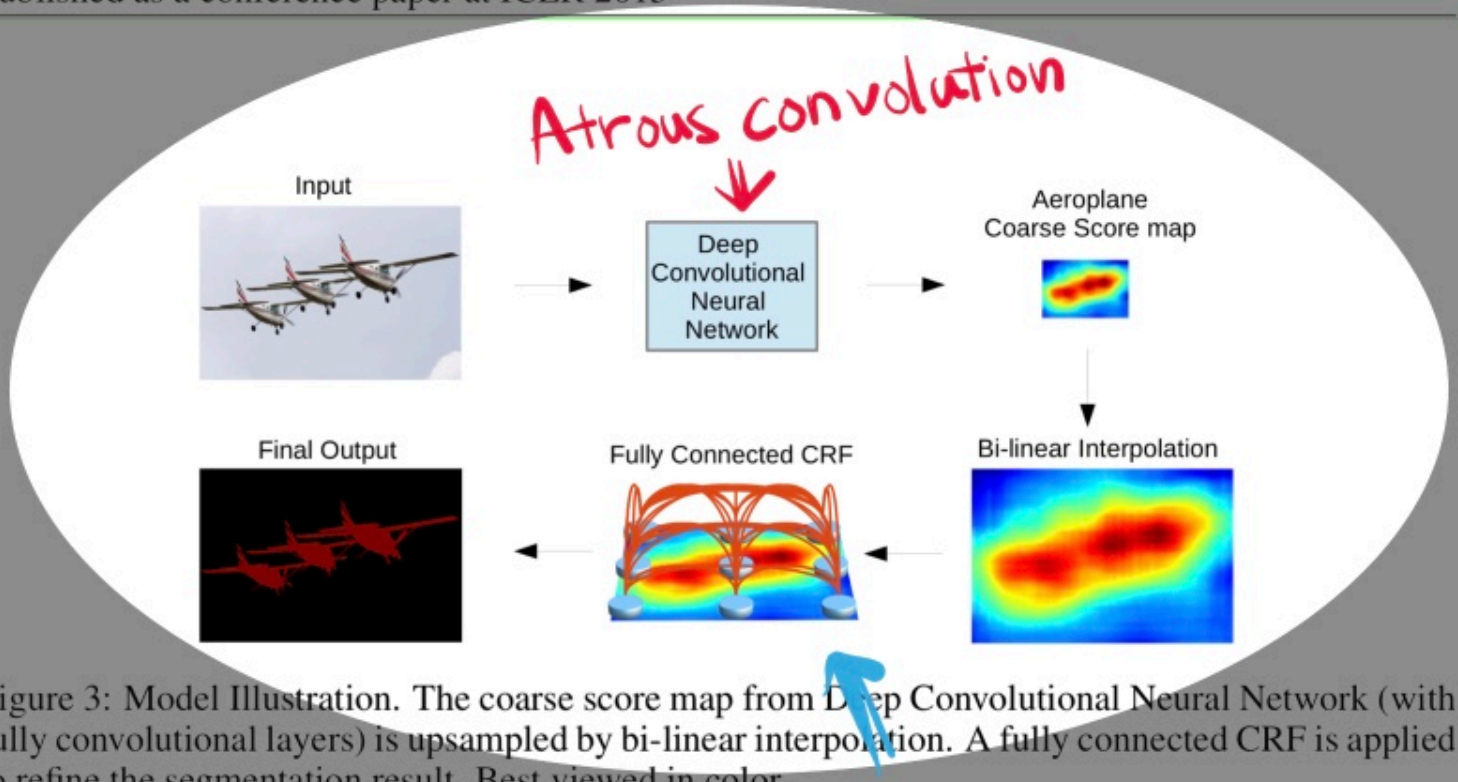
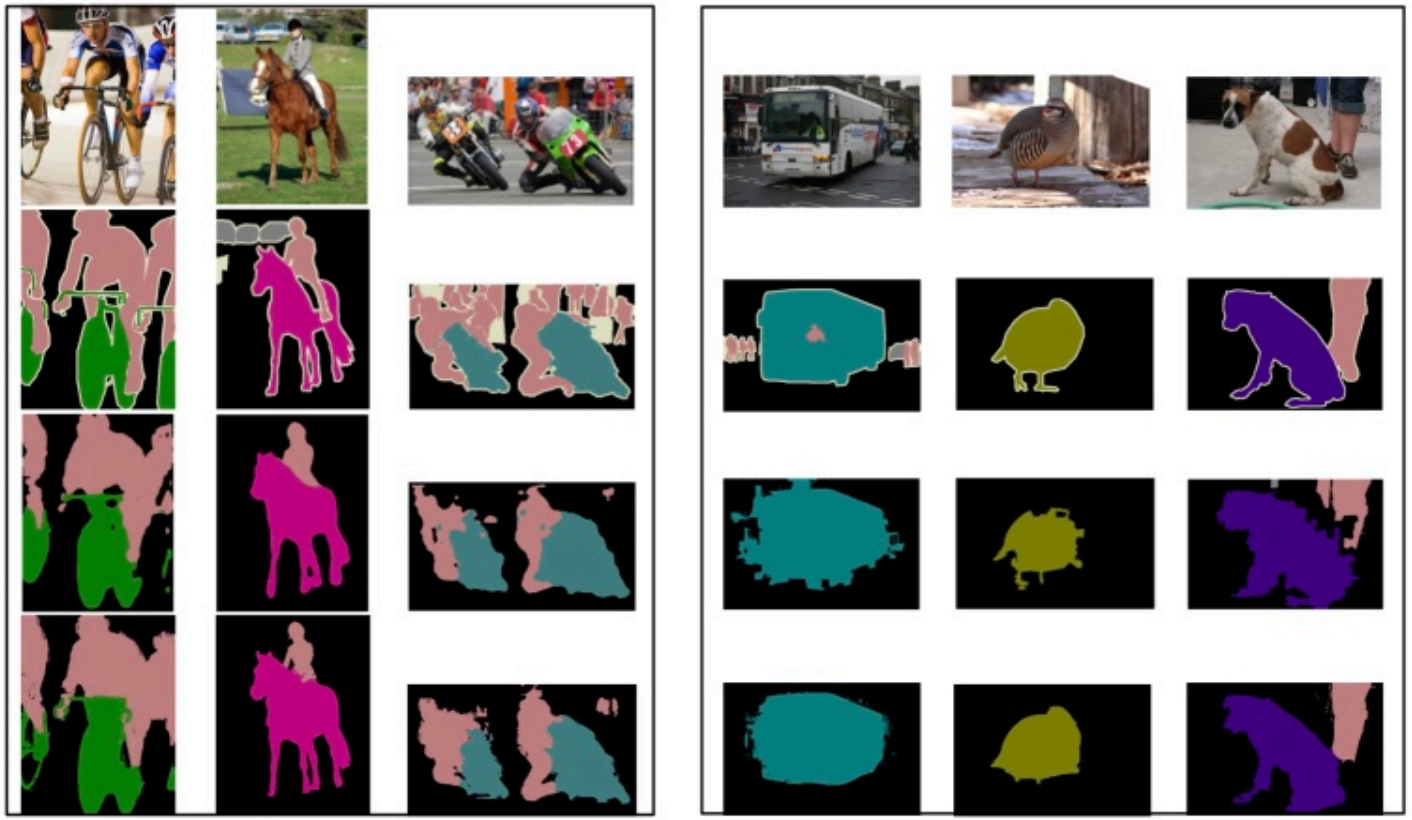


Figure 3: Model Illustration. The coarse score map from Deep Convolutional Neural Network (with fully convolutional layers) is upsampled by bi-linear interpolation. A fully connected CRF is applied to refine the segmentation result. Best viewed in color.

et al., 2004) in conjunction to local-range CRFs can potentially improve localization but still miss thin-structures and typically requires solving an expensive discrete optimization problem.

To overcome these limitations of short-range CRFs, we integrate into our system the fully connected CRF model of Krähenbühl & Koltun (2011). The model employs the energy function

of 2 pixels: bottom-right: trimap of 10 pixels). Quality of segmentation result within a band around the object boundaries for the proposed methods. (b) Pixelwise accuracy. (c) Pixel mean IOU.



(a) FCN-8s vs. DeepLab-CRF

(b) TTI-Zoomout-16 vs. DeepLab-CRF

Figure 6: Comparisons with state-of-the-art models on the val set. First row: images. Second row: ground truths. Third row: other recent models (Left: FCN-8s, Right: TTI-Zoomout-16). Fourth row: our DeepLab-CRF. Best viewed in color.

Learning Deconvolution Network for Semantic Segmentation

Hyeonwoo Noh Seunghoon Hong Bohyung Han
Department of Computer Science and Engineering, POSTECH, Korea
{hyeonwoonoh_,maga33,bhhan}@postech.ac.kr

Abstract

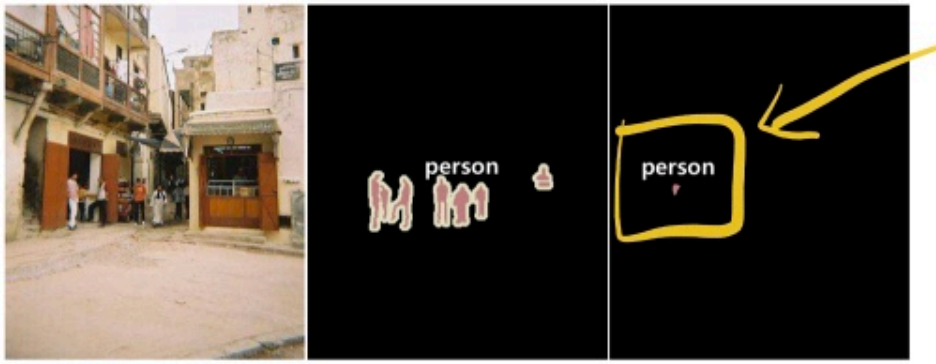
We propose a novel semantic segmentation algorithm by learning a deep deconvolution network. We learn the network on top of the convolutional layers adopted from VGG 16-layer net. The deconvolution network is composed of deconvolution and unpooling layers, which identify pixel-wise class labels and predict segmentation masks. We apply the trained network to each proposal in an input image, and construct the final semantic segmentation map by combining the results from all proposals in a simple manner. The proposed algorithm mitigates the limitations of the existing methods based on fully convolutional networks by integrating deep deconvolution network and proposal-wise prediction; our segmentation method typically identifies detailed structures and handles objects in multiple scales naturally. Our network demonstrates outstanding performance in PASCAL VOC 2012 dataset, and we achieve the best accuracy (72.5%) among the methods trained without using Microsoft COCO dataset through ensemble with the fully convolutional network.

(a) Inconsistent labels due to large object size

(b) Missing labels due to small object size

Figure 1. Limitations of semantic segmentation algorithms based on fully convolutional network. (Left) original image. (Center) ground-truth annotation. (Right) segmentations by [19]

(a) Inconsistent labels due to large object size



(b) Missing labels due to small object size

Needs a bigger receptive field

Needs a smaller receptive field.

Problem 1: Network has a predefined fixed size receptive field

→ Too small or too big objects are neglected.

Problem 2: Details are often lost

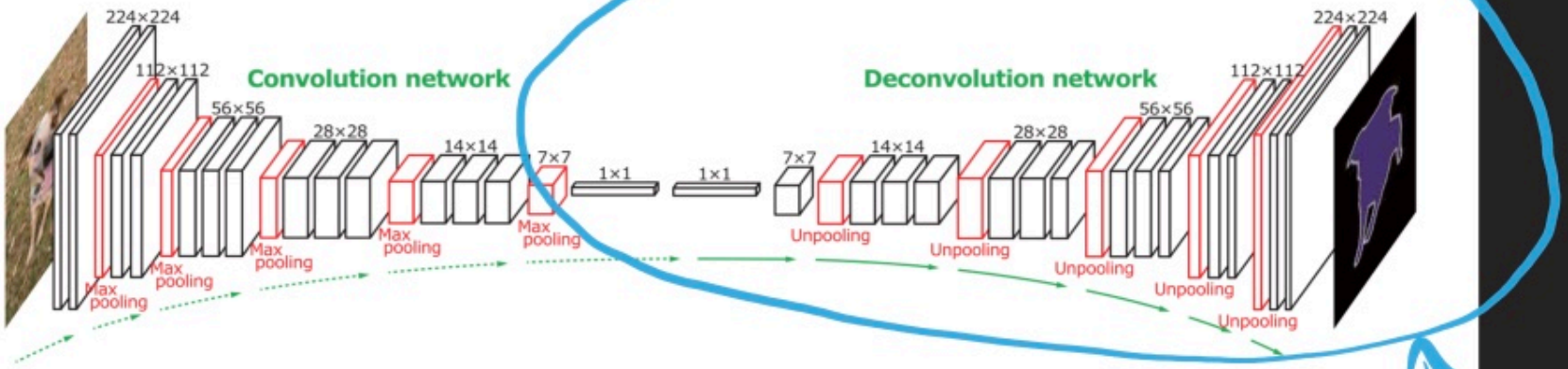
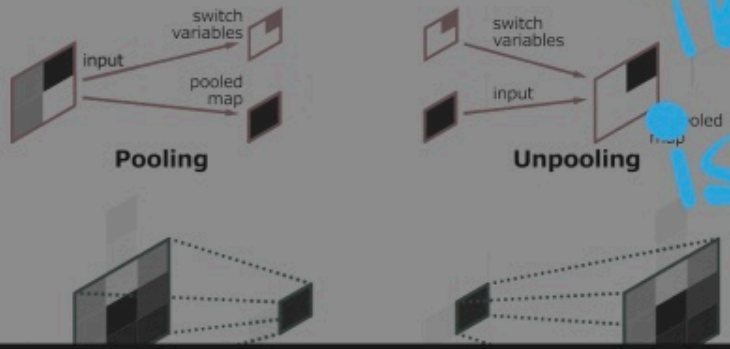


Figure 2. Overall architecture of the proposed network. On top of the convolution network based on VGG 16-layer net, we put a multi-layer deconvolution network to generate the accurate segmentation map of an input proposal. Given a feature representation obtained from the convolution network, dense pixel-wise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

erator that produces object segmentation from the feature extracted from the convolution network. The final output of the network is a probability map in the same size to input image, indicating probability of each pixel that belongs to one of the predefined classes.

We employ VGG 16-layer net [24] for convolutional part with its last classification layer removed. Our convolution network has 13 convolutional layers altogether, rectification and pooling operations are sometimes performed between convolutions, and 2 fully connected layers are augmented at the end to impose class-specific projection. Our



This part is new!

Network must be symmetric

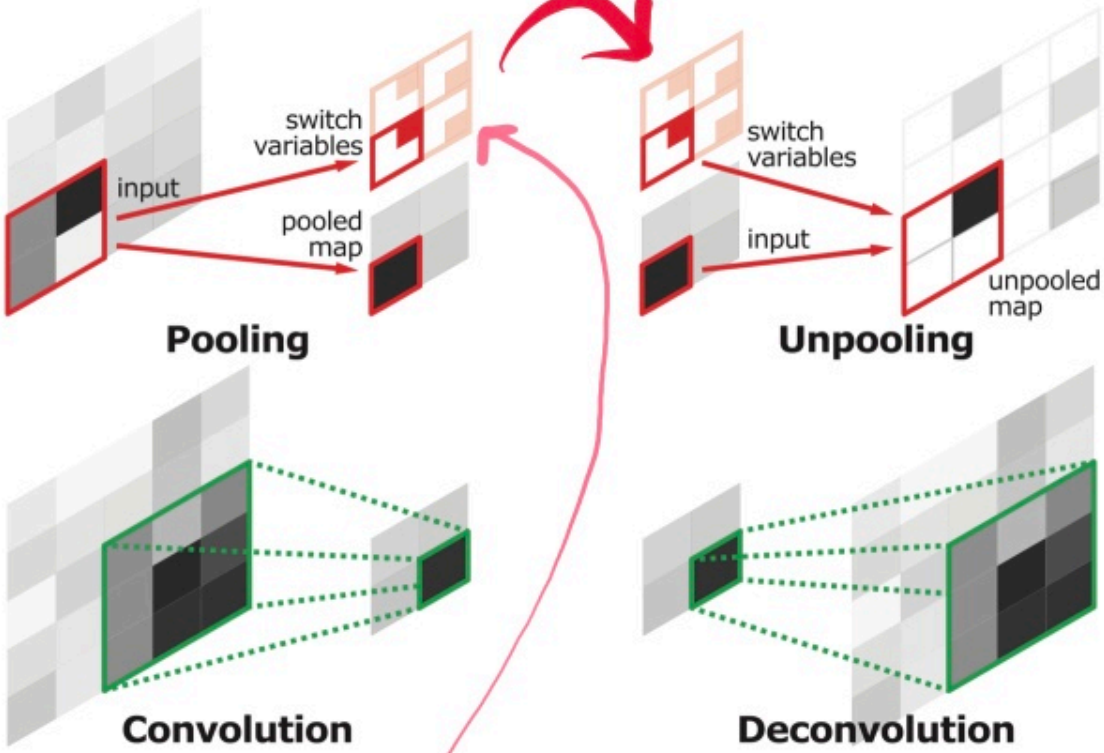
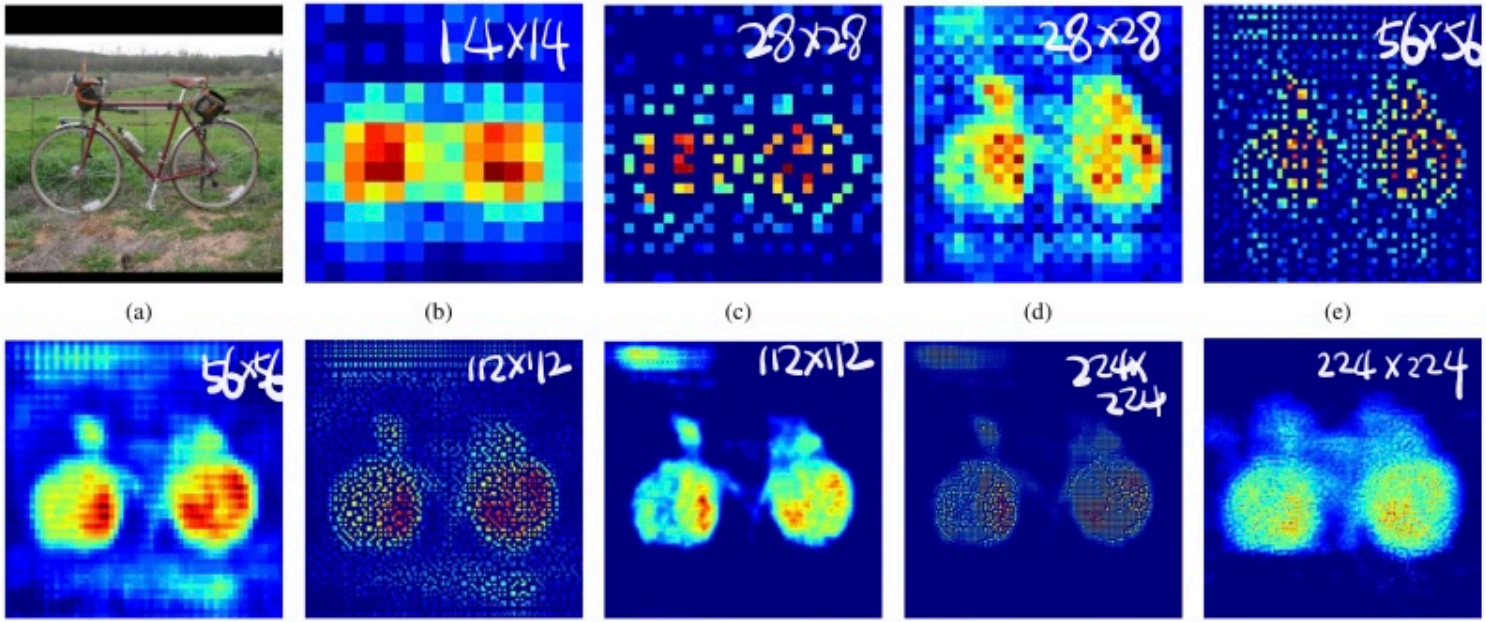


Figure 3. Illustration of deconvolution and unpooling operations.

Pooling is designed for filtering noise observations.

Spatial information is lost during pooling



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 4. Visualization of activations in our deconvolution network. The activation maps from (b) to (j) 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.

Batch normalization is important.

Two stage traing

Ensemble model is used

MSRA-CFM [ ]	87.7	35.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	61.8
FCN8s [ ]	91.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
TTI-Zoomout-16 [ ]	89.8	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF [ ]	93.1	84.4	**54.5**	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	**59.7**	82.2	50.4	73.1	63.7	71.6
DeconvNet	92.7	85.9	42.6	78.9	62.5	66.6	87.4	77.8	79.5	26.3	73.4	60.2	70.8	76.5	79.6	77.7	58.2	77.4	52.9	75.2	59.8	69.6
DeconvNet+CRF	92.9	87.8	41.9	80.6	63.9	67.3	**88.1**	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	**55.0**	75.2	61.5	70.5
EDeconvNet	92.9	88.4	39.7	79.0	63.0	67.7	87.1	**81.5**	84.4	27.8	76.1	61.2	78.0	79.3	83.1	79.3	58.0	82.5	52.3	80.1	64.0	71.7
EDeconvNet+CRF	93.1	**89.9**	39.3	79.7	63.9	**68.2**	87.4	81.2	**86.1**	28.5	**77.0**	62.0	79.0	**80.3**	**83.6**	80.2	58.8	**83.4**	54.3	**80.7**	65.0	**72.5**
* WSSL [ ]	93.2	85.3	36.2	**84.8**	61.2	67.5	84.7	81.4	81.0	**30.8**	73.8	53.8	77.5	76.5	82.3	**81.6**	56.3	78.9	52.3	76.6	63.3	70.4
* BoxSup [ ]	**93.6**	86.4	35.5	79.7	**65.2**	65.2	84.3	78.5	83.7	30.5	76.2	**62.6**	**79.3**	76.1	82.1	81.3	57.0	78.2	**55.0**	72.5	**68.1**	71.0

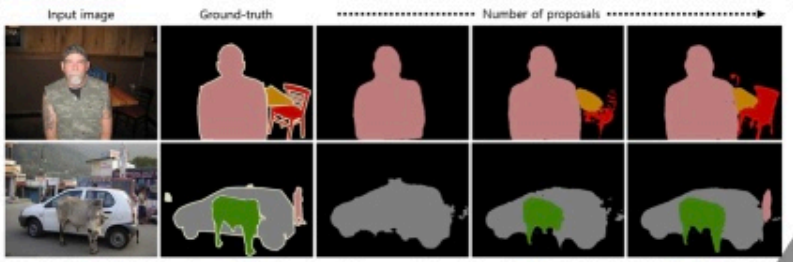


Figure 6. Benefit of instance-wise prediction. We aggregate the proposals in a decreasing order of their sizes. The algorithm identifies finer object structures through iterations by handling multi-scale objects effectively.

and FCN have complementary characteristics as discussed in Section 5.2. Our ensemble method denoted by EDeconvNet achieves the state-of-the-art accuracy among the methods trained only on PASCAL VOC 2012 augmented dataset.

Figure 6 demonstrates effectiveness of instance-wise prediction for accurate segmentation. We aggregate the proposals in a decreasing order of their sizes and observe the progress of segmentation. As the number of aggregated proposals increases, the algorithm identifies finer object struc-

7. Conclusion

We proposed a novel semantic segmentation algorithm by learning a deconvolution network. The proposed deconvolution network is suitable to generate dense and precise object segmentation masks since course-to-fine structures of an object is reconstructed progressively through a sequence of deconvolution operations. Our algorithm based on instance-wise prediction is advantageous to han-

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

Liang-Chieh Chen, George Papandreou, *Senior Member, IEEE,* Iasonas Kokkinos, *Member, IEEE,* Kevin Murphy, and Alan L. Yuille, *Fellow, IEEE*

Abstract—In this work we address the task of semantic image segmentation with Deep Learning and make three main contributions that are experimentally shown to have substantial practical merit. *First*, we highlight convolution with upsampled filters, or 'atrous convolution', as a powerful tool in dense prediction tasks. Atrous convolution allows us to explicitly control the resolution at which feature responses are computed within Deep Convolutional Neural Networks. It also allows us to effectively enlarge the field of view of filters to incorporate larger context without increasing the number of parameters or the amount of computation. *Second*, we propose atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales. ASPP probes an incoming convolutional feature layer with filters at multiple sampling rates and effective fields-of-views, thus capturing objects as well as image context at multiple scales. *Third*, we improve the localization of object boundaries by combining methods from DCNNs and probabilistic graphical models. The commonly deployed combination of max-pooling and downsampling in DCNNs achieves invariance but has a toll on localization accuracy. We overcome this by combining the responses at the final DCNN layer with a fully connected Conditional Random Field (CRF), which is shown both qualitatively and quantitatively to improve localization performance. Our proposed "DeepLab" system sets the new state-of-art at the PASCAL VOC-2012 semantic image segmentation task, reaching 79.7% mIOU in the test set, and advances the results on three other datasets: PASCAL-Context, PASCAL-Person-Part, and Cityscapes. All of our code is made publicly available online.

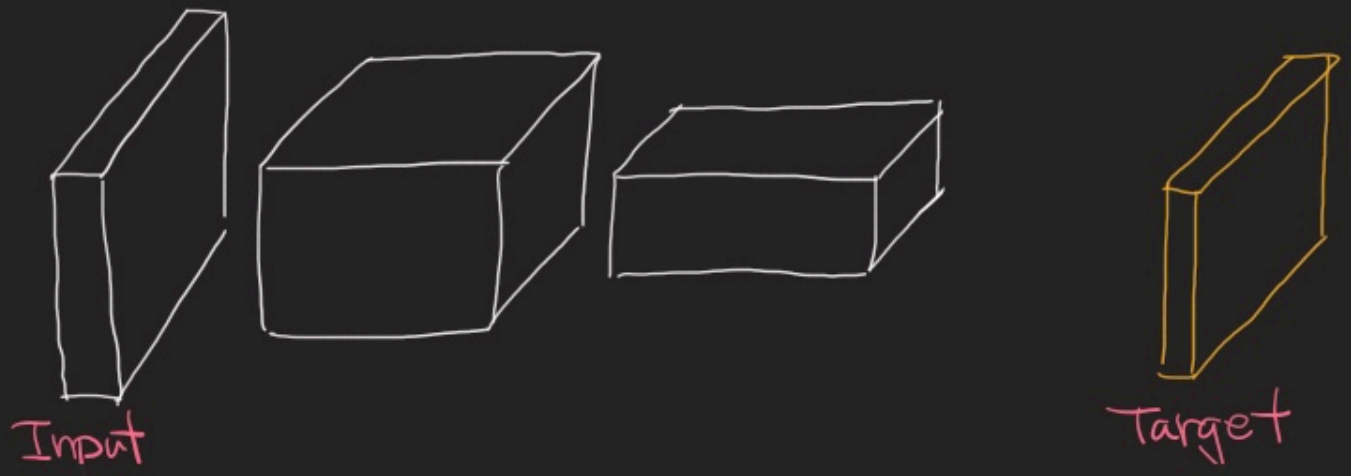
Index Terms—Convolutional Neural Networks, Semantic Segmentation, Atrous Convolution, Conditional Random Fields.

Consider three challenges

1) Reduced feature resolution

2) Existence of objects at multiple scales

3) Reduced localization accuracy



Consider three challenges

1) Reduced feature resolution

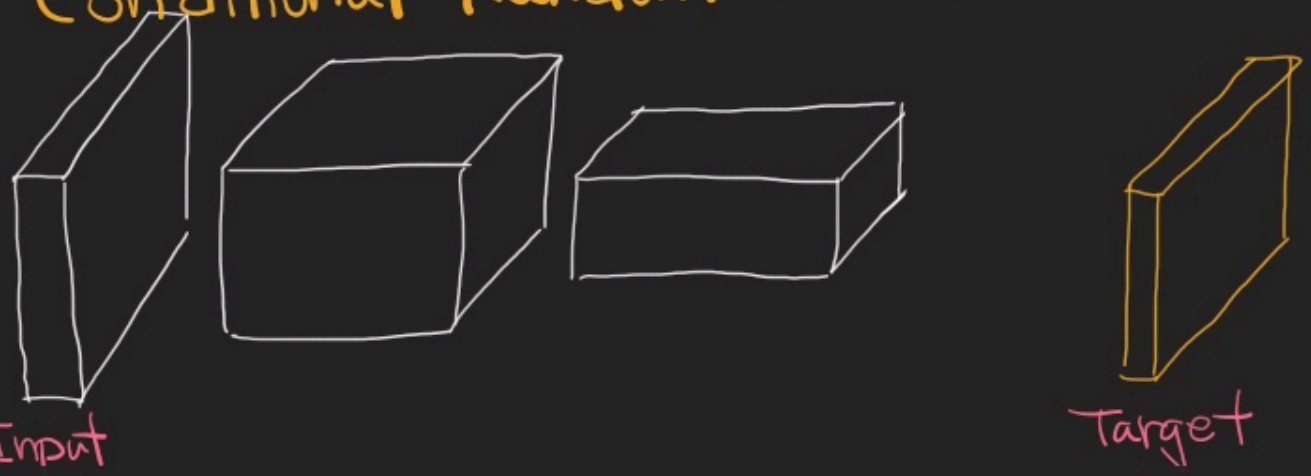
Atrous convolution

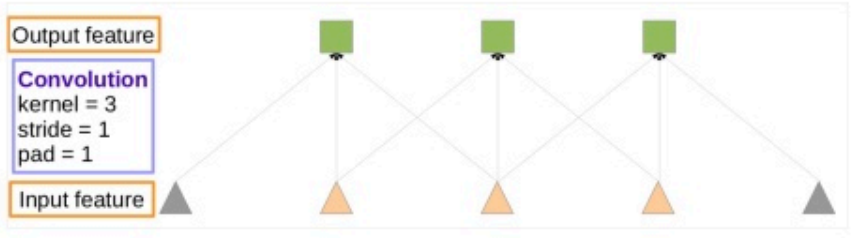
2) Existence of objects at multiple scales

Atrous spatial pyramid pooling

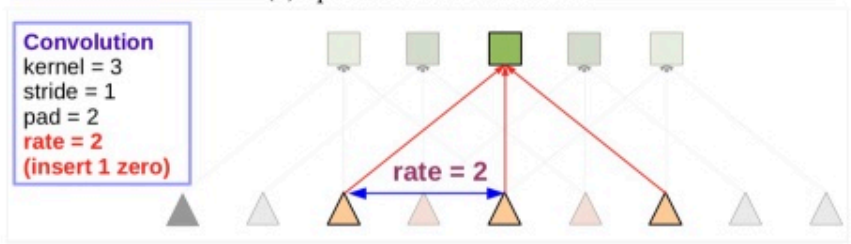
3) Reduced localization accuracy

Conditional Random Fields





(a) Sparse feature extraction



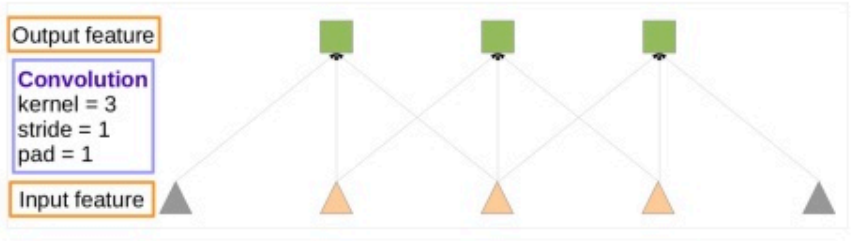
(b) Dense feature extraction

Fig. 2: Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

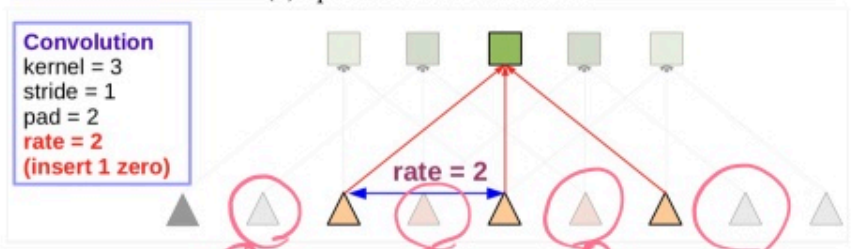
Atrous

= A + trous

"= holes" in French



(a) Sparse feature extraction



(b) Dense feature extraction

Fig. 2: Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

Atrous

= A + trous

= "holes" in French

Added holes (or zeros)

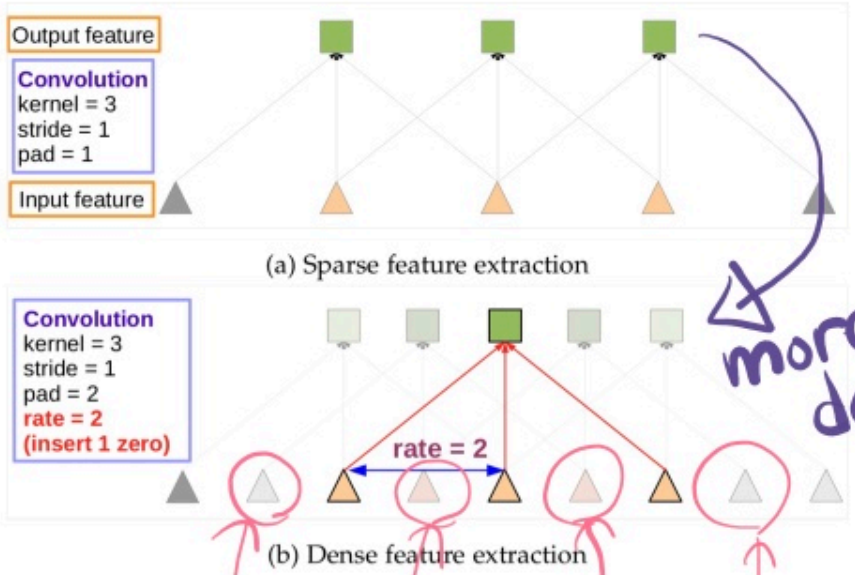


Fig. 2: Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

more dense !

Added holes (or zeros)

Atrous

= A + trous

= "holes" in French

The atrous convolution enables more dense features without requiring learning any extra parameters

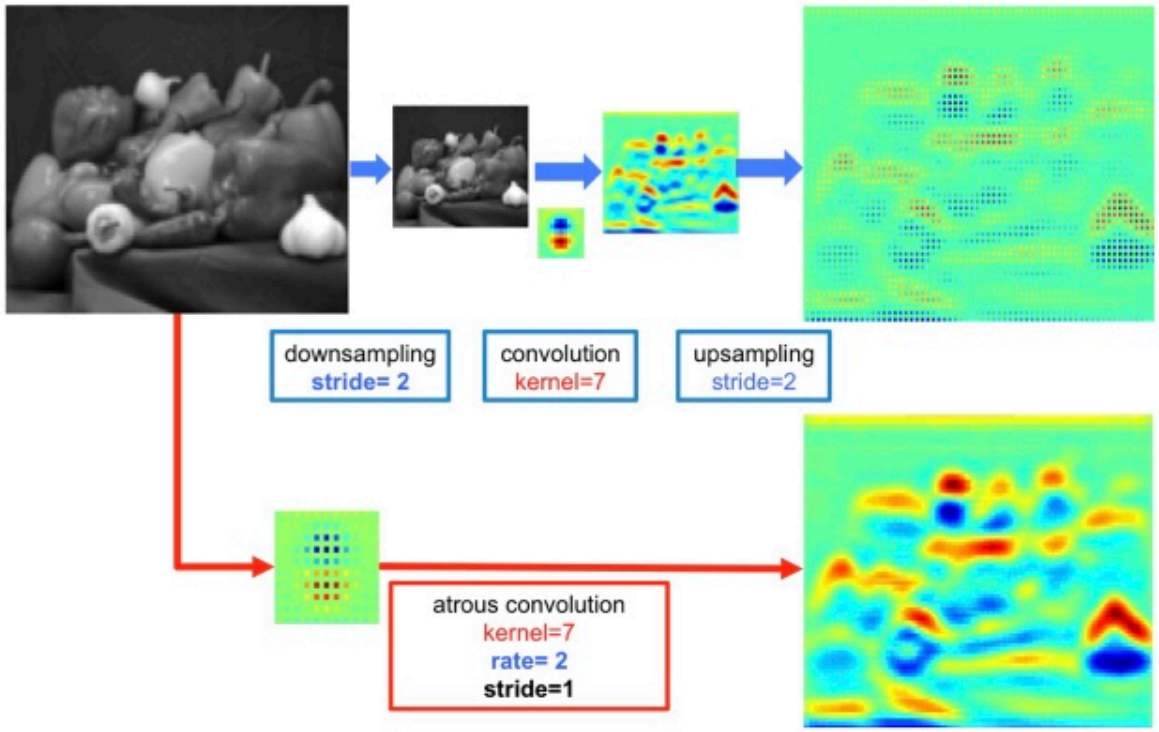


Fig. 3: Illustration of atrous convolution in 2-D. Top row: sparse feature extraction with standard convolution on a low resolution input feature map. Bottom row: Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

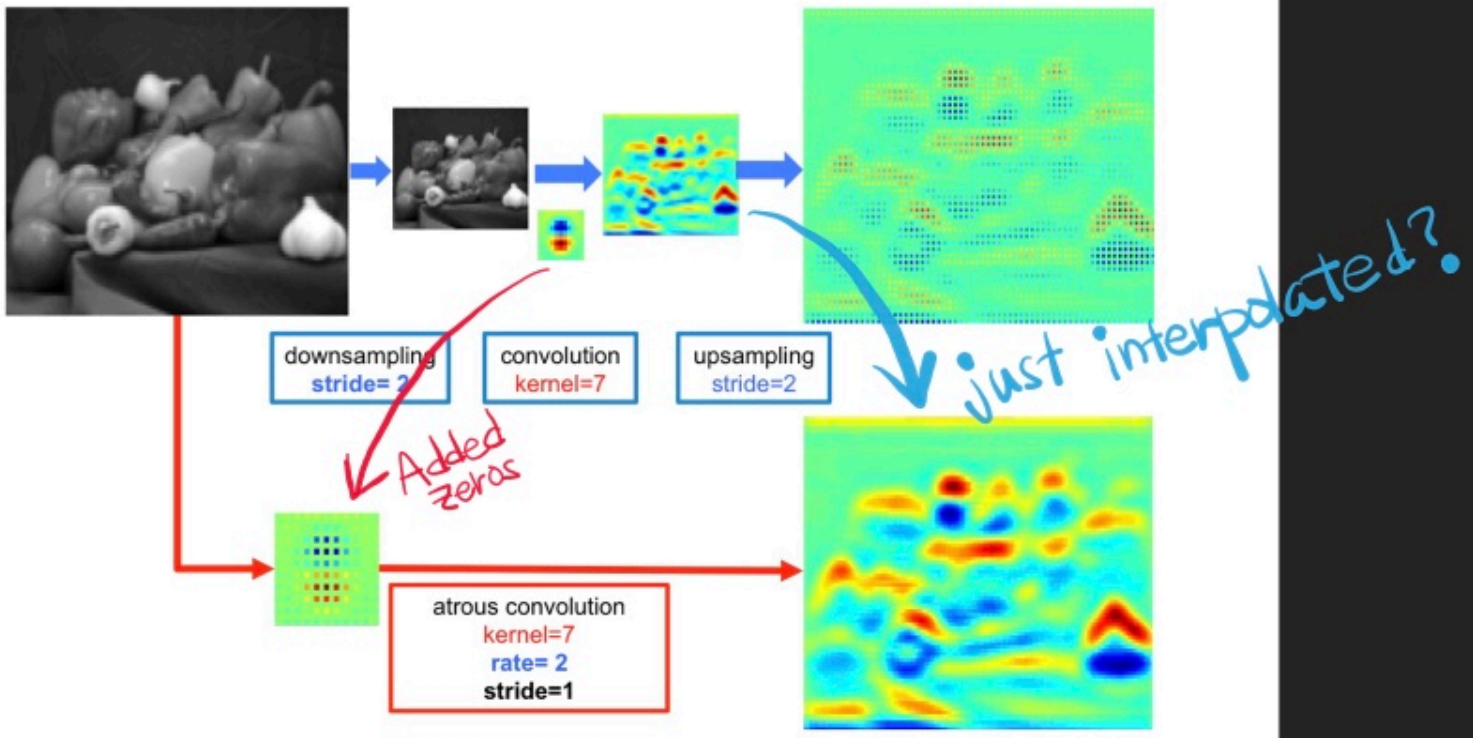


Fig. 3: Illustration of atrous convolution in 2-D. Top row: sparse feature extraction with standard convolution on a low resolution input feature map. Bottom row: Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

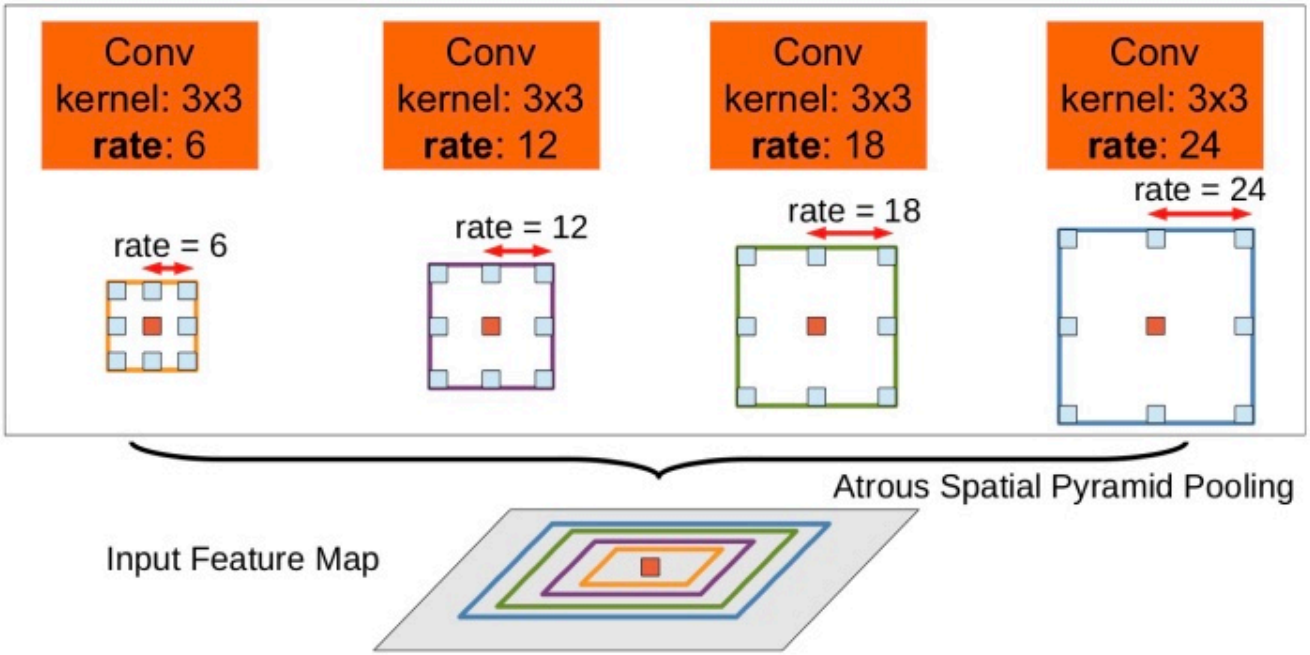
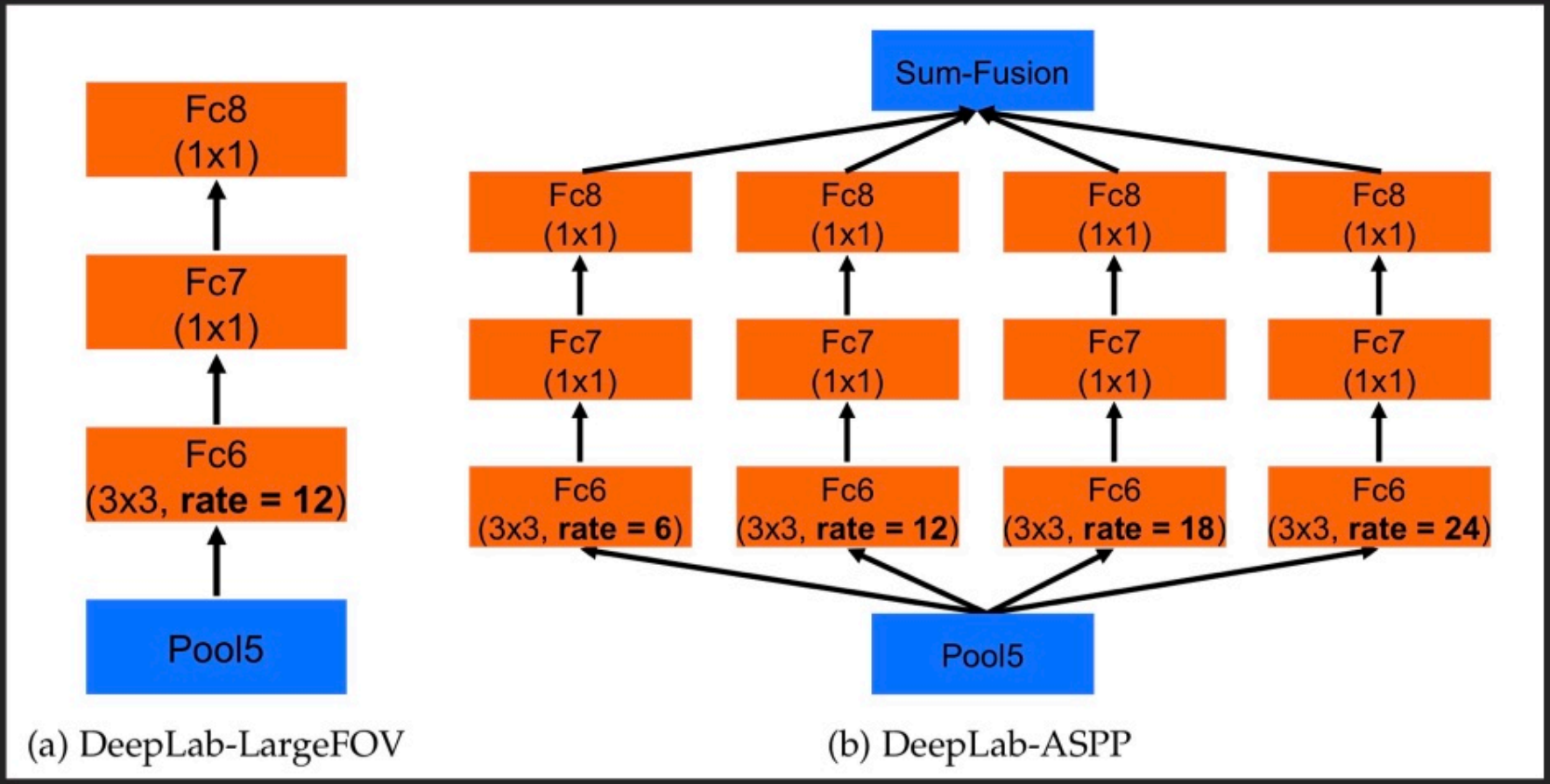


Fig. 4: Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

Enlarge the field-of-view



Atrous spatial pyramid pooling (ASPP)

Results

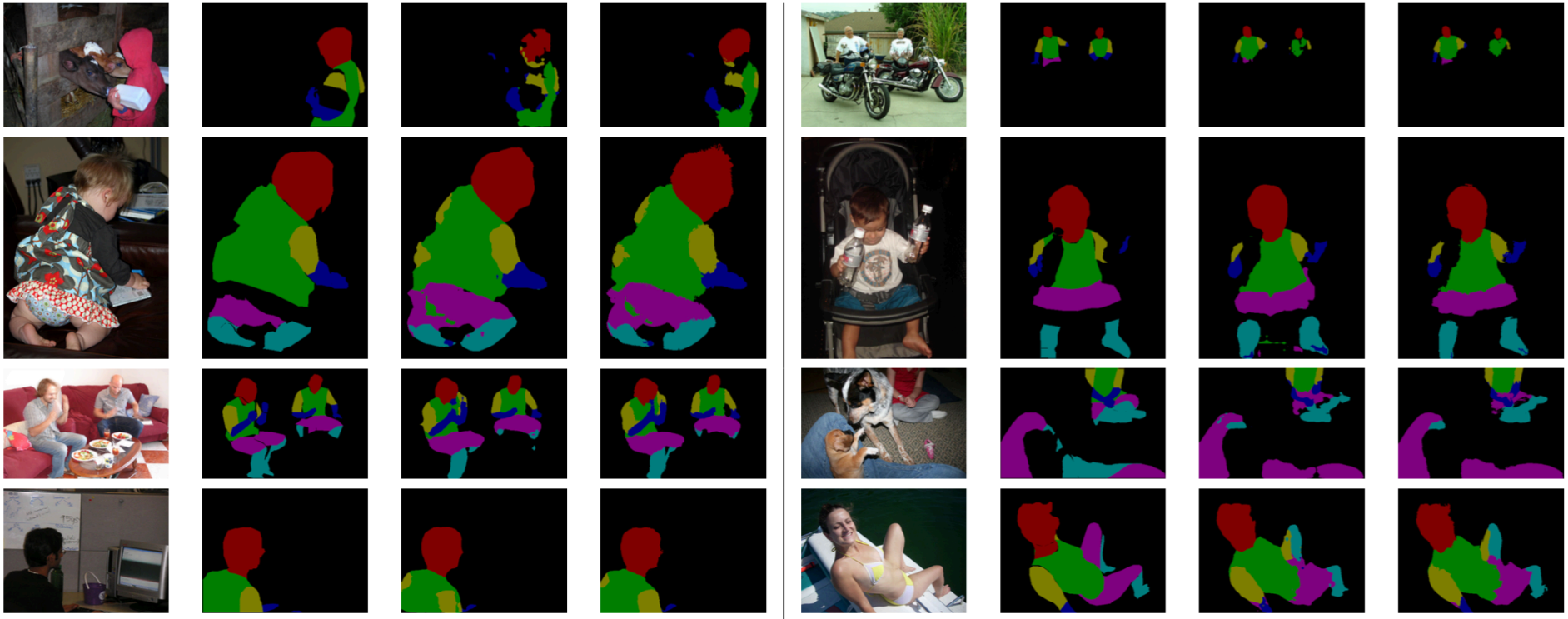


Fig. 12: PASCAL-Person-Part results. Input image, ground-truth, and our DeepLab results before/after CRF.

Results

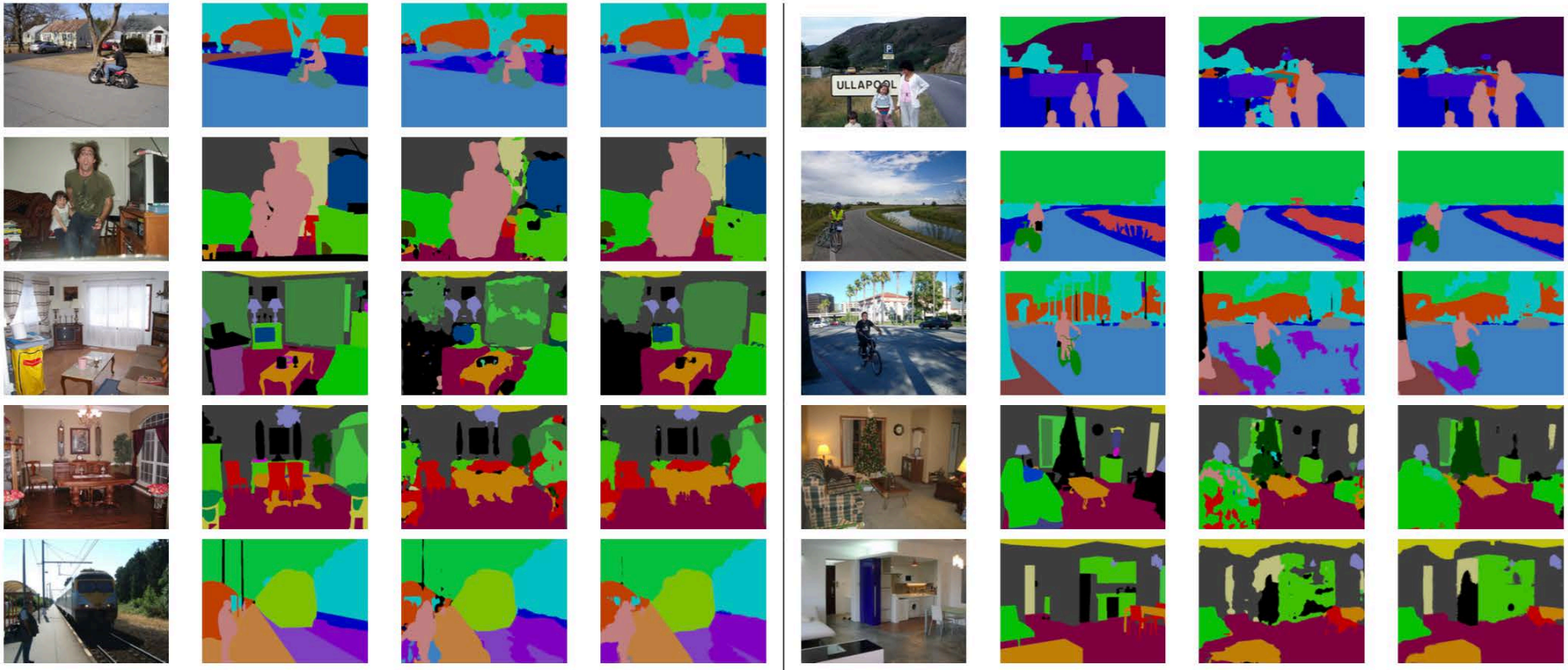


Fig. 11: PASCAL-Context results. Input image, ground-truth, and our DeepLab results before/after CRF.