CVPR, 2015

Fully convolutional networks for semantic segmentation

Long, Jonathan, Evan Shelhamer, and Trevor Darrell

Proceedings of the IEEE conference on computer vision and pattern recognition

- **Problem/Objective**
  - Semantic segmentation

- **Contribution/Key Idea**
  - Fully convolutional network ~에서 바뀌었다.
    1. Forward : transform 'fully connected layers' into 'convolution layers'.
    2. Backward : in-network upsampling layers.
    3. Whole image training.
    4. Skip architecture.

전유진

Fully convolutional networks for semantic segmentation

- Fully convolutional network
    1. Forward : transform ‘fully connected layers’ into ‘convolution layers’.

정의

Forward propagation of FCN : inference : produce coarse output maps.

문제점

Fully connected layers take fixed-sized inputs and produce nonspatial outputs.

해결

Transform fully connected layers into convolution layers.

결과

Take input of any size and output classification maps.

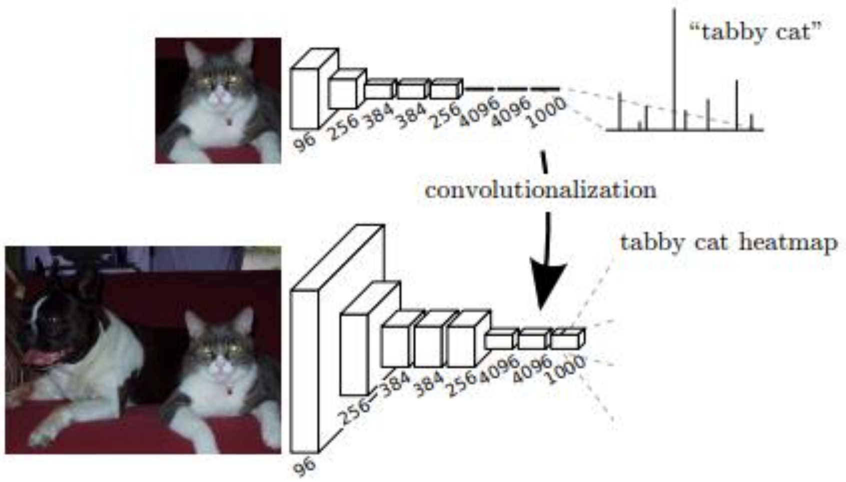


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Fully convolutional networks for semantic segmentation

- Fully convolutional network

  2. Backward : in-network upsampling layers.

이것도 forward과정임.

정의

Backward propagation of FCN : learning : connect coarse outputs back to dense pixels.

개선

Shift-and-stitch : input shifting and output interlacing

$\downarrow$

Upsampling : in-network, end-to-end learning

설명

Upsampling with factor f

= deconvolution with an output stride of f.

= convolution with fractional input stride of 1/f

Deconvolution이 아니라 이제는 Transposed convolution.

Subset.

Fully convolutional networks for semantic segmentation

- Fully convolutional network

    3. Whole image training.

배경

<u>Gradient computation</u> is driven by the training distribution.

개선

<u>Patchwise training</u>

↓

<u>Whole image training</u>

효과

Faster convergence for dense prediction.

By weighting the loss, class balance achieved & spatial correlation addressed.

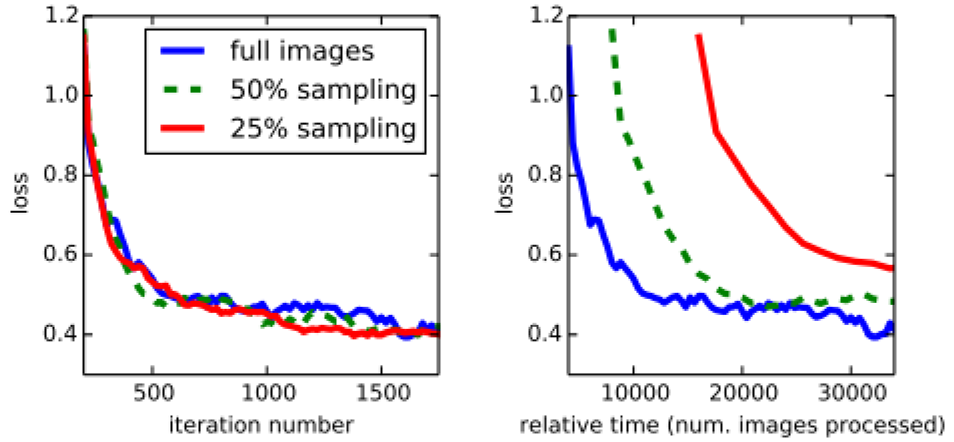


Figure 5. Training on whole images is just as effective as sampling patches, but results in faster (wall time) convergence by making more efficient use of data. Left shows the effect of sampling on convergence rate for a fixed expected batch size, while right plots the same by relative wall time.

Fully convolutional networks for semantic segmentation

- Fully convolutional network
  4. Skip architecture.

문제

Output is dissatisfyingly coarse.

원인

32 pixel stride at the final prediction layer limits the scale of detail in the upsampled output.

해결

Combine fine layers and coarse layers

결과

Model make local predictions that respect global structure.

Combine 예시. FCN-16s 구하는 과정.

-produce additional prediction by adding 1x1 convolution layer on top of pool4.
-initialize 2x upsampling to bilinear interpolation.
-fuse 2x upsampled prediction from pool5 with additional prediction.
-stride 16 predictions are upsampled back to the image.
-FCN-16s is learned end-to-end, initialized with the parameters of the last coarser net, which we now call FCN-32s.

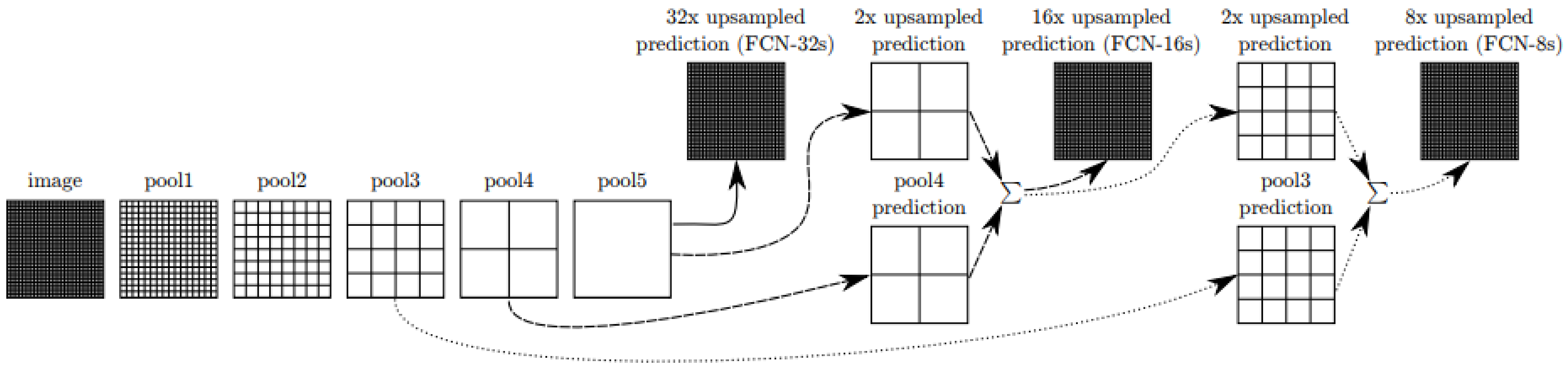


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (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. Dotted line (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.

결과
Model make local predictions that respect global structure.

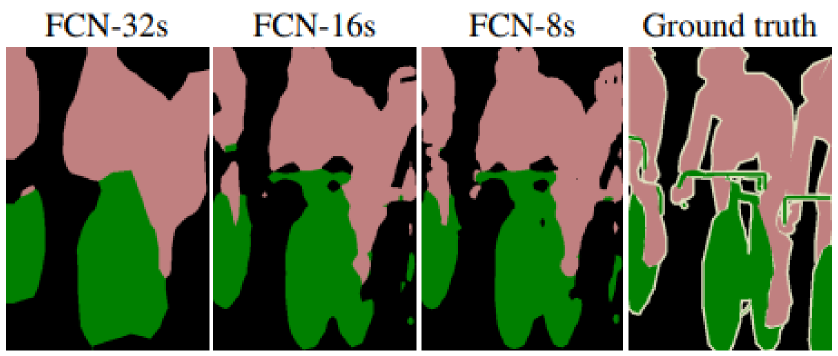


Figure 4. Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

The reason why
we do not continue fusing even lower layers
= minor additional improvement.

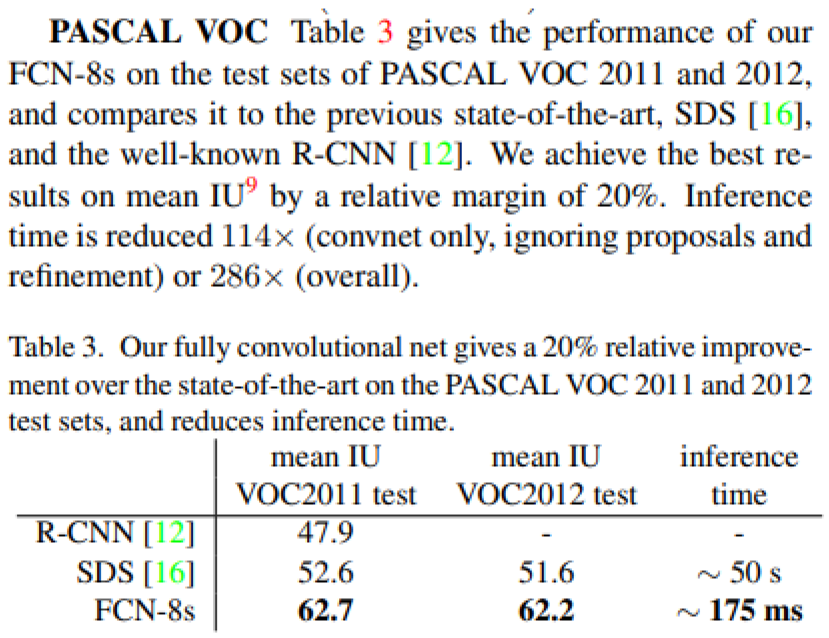
Table 2. Comparison of skip FCNs on a subset of PASCAL VOC2011 validation[7]. Learning is end-to-end, except for FCN-32s-fixed, where only the last layer is fine-tuned. Note that FCN-32s is FCN-VGG16, renamed to highlight stride.

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**

Fully convolutional networks for semantic segmentation

- **Results**

PASCAL VOC



PASCAL VOC Table 3 gives the performance of our FCN-8s on the test sets of PASCAL VOC 2011 and 2012, and compares it to the previous state-of-the-art, SDS [16], and the well-known R-CNN [12]. We achieve the best results on mean IU[9] by a relative margin of 20%. Inference time is reduced 114× (convnet only, ignoring proposals and refinement) or 286× (overall).

Table 3. Our fully convolutional net gives a 20% relative improvement over the state-of-the-art on the PASCAL VOC 2011 and 2012 test sets, and reduces inference time.

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [12]	47.9	-	-
SDS [16]	52.6	51.6	∼ 50 s
FCN-8s	**62.7**	**62.2**	∼ **175 ms**

Fully convolutional networks for semantic segmentation

- **Results**

NYUDv2

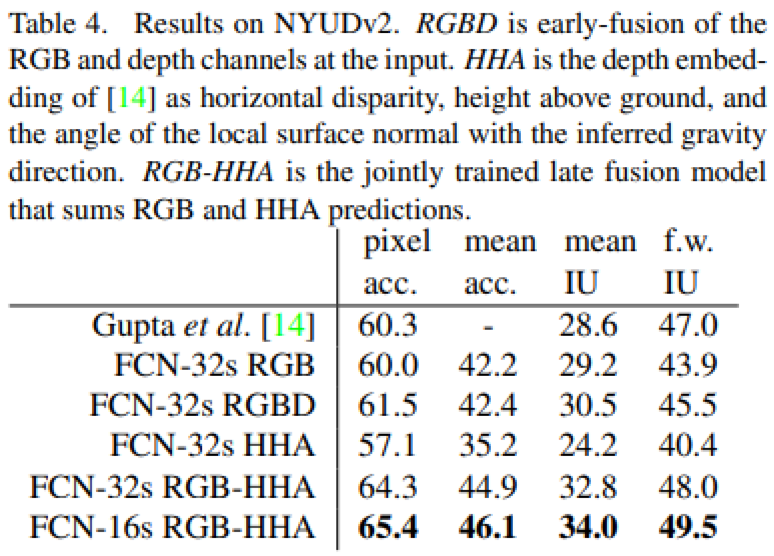


Table 4. Results on NYUDv2. *RGBD* is early-fusion of the RGB and depth channels at the input. *HHA* is the depth embedding of [14] as horizontal disparity, height above ground, and the angle of the local surface normal with the inferred gravity direction. *RGB-HHA* is the jointly trained late fusion model that sums RGB and HHA predictions.

	pixel acc.	mean acc.	mean IU	f.w. IU
Gupta *et al.* [14]	60.3	-	28.6	47.0
FCN-32s RGB	60.0	42.2	29.2	43.9
FCN-32s RGBD	61.5	42.4	30.5	45.5
FCN-32s HHA	57.1	35.2	24.2	40.4
FCN-32s RGB-HHA	64.3	44.9	32.8	48.0
FCN-16s RGB-HHA	**65.4**	**46.1**	**34.0**	**49.5**

Fully convolutional networks for semantic segmentation

- **Results**

SIFT Flow

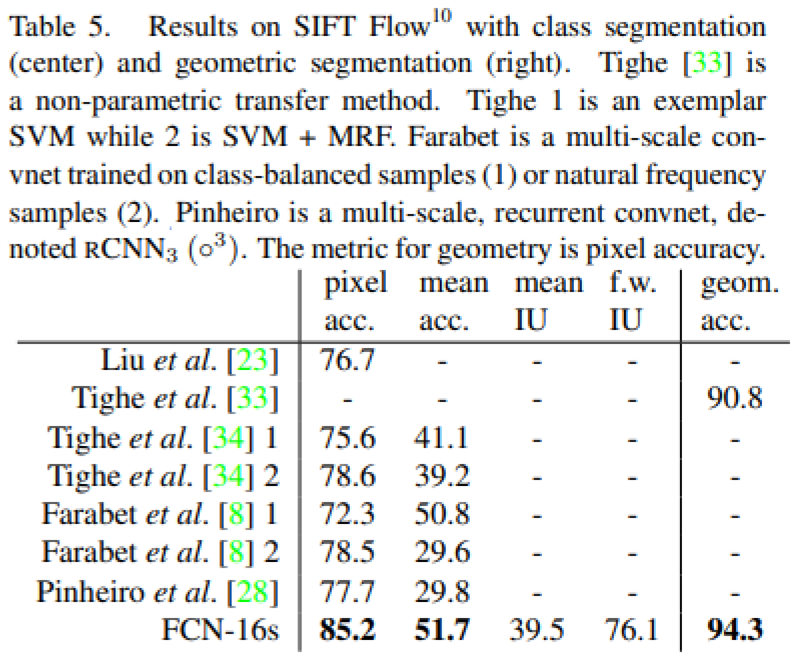


Table 5. Results on SIFT Flow[10] with class segmentation (center) and geometric segmentation (right). Tighe [33] is a non-parametric transfer method. Tighe 1 is an exemplar SVM while 2 is SVM + MRF. Farabet is a multi-scale convnet trained on class-balanced samples (1) or natural frequency samples (2). Pinheiro is a multi-scale, recurrent convnet, denoted RCNN$_3$ ($\circ^3$). The metric for geometry is pixel accuracy.

	pixel acc.	mean acc.	mean IU	f.w. IU	geom. acc.
Liu *et al.* [23]	76.7	-	-	-	-
Tighe *et al.* [33]	-	-	-	-	90.8
Tighe *et al.* [34] 1	75.6	41.1	-	-	-
Tighe *et al.* [34] 2	78.6	39.2	-	-	-
Farabet *et al.* [8] 1	72.3	50.8	-	-	-
Farabet *et al.* [8] 2	78.5	29.6	-	-	-
Pinheiro *et al.* [28]	77.7	29.8	-	-	-
FCN-16s	**85.2**	**51.7**	39.5	76.1	**94.3**