Fully Convolutional Networks for Semantic Segmentation

Authors: *Jonathan Long , Evan Shelhamer, Trevor Darrell*

Presented by: Jason T. Smith

November 14th, 2018

Fully-Convolutional Networks (FCN)

- **What are they?**

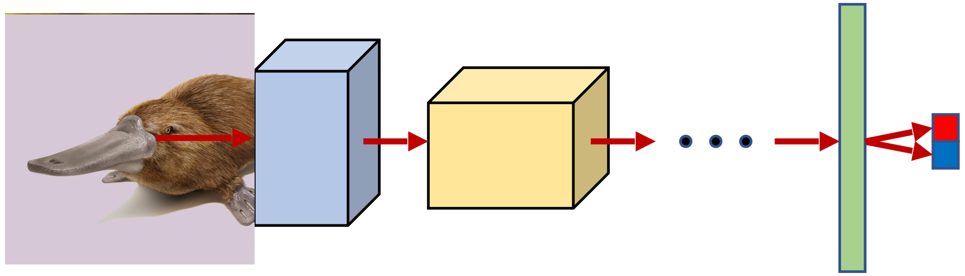
- Independent of image input size.

- **…Why?**

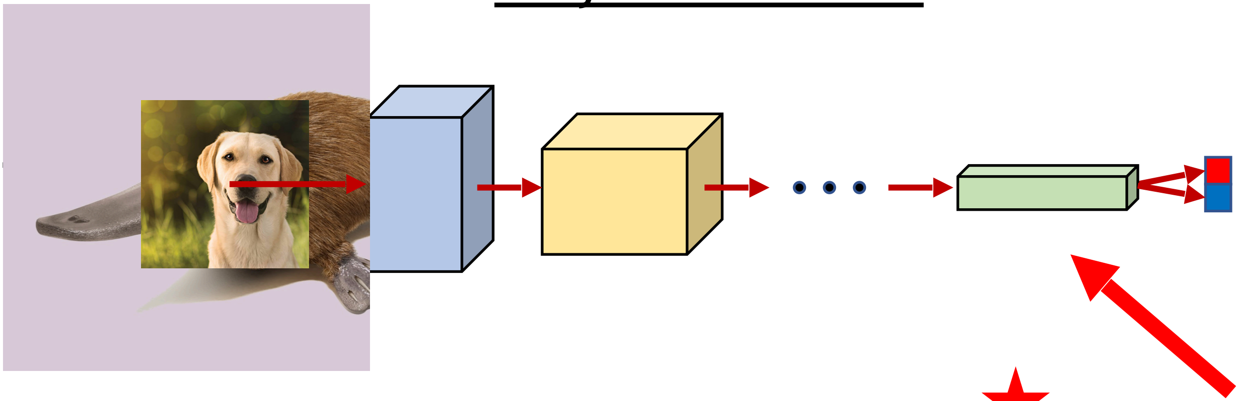
- Fully-connected layers require set input size.

- Simple form of FCN allows for the same sor of operation, but operates independent of input size.

“Fully-connected”



Fully-convolutional



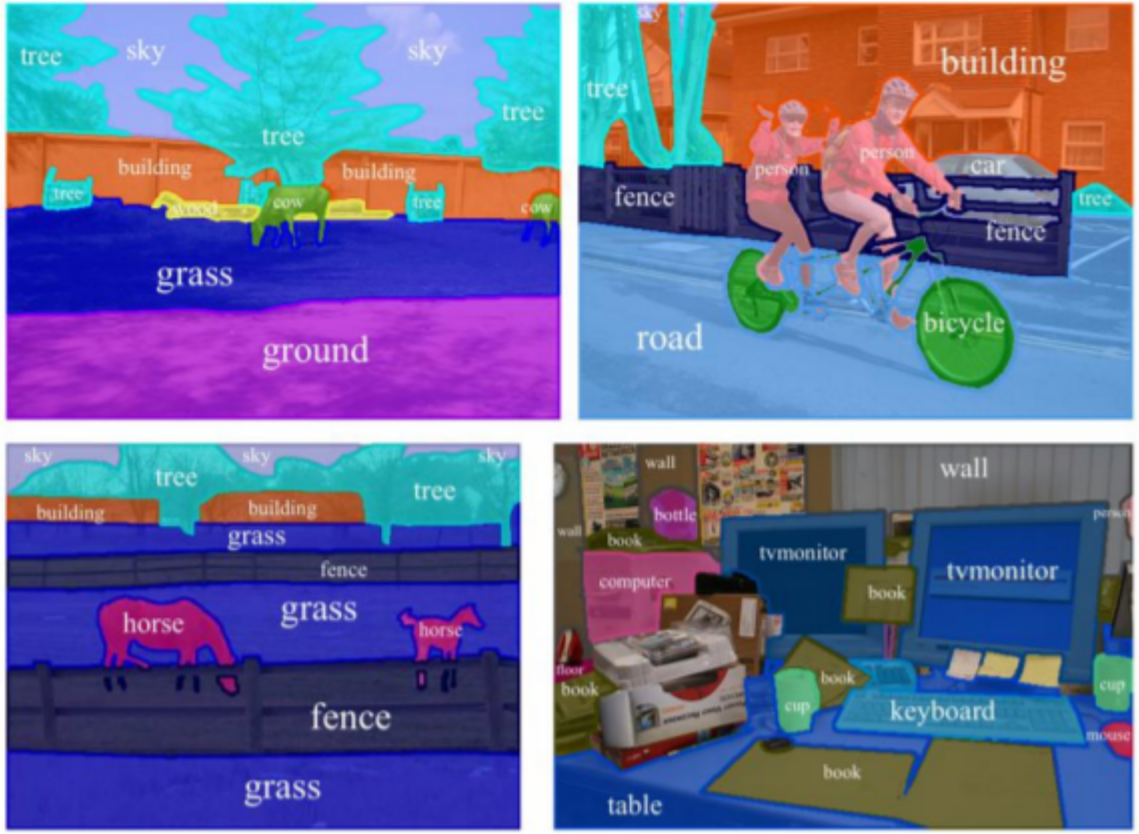
★Not quite as simple as this

Semantic Segmentation (end-to-end)

- **Semantics:** meaning.

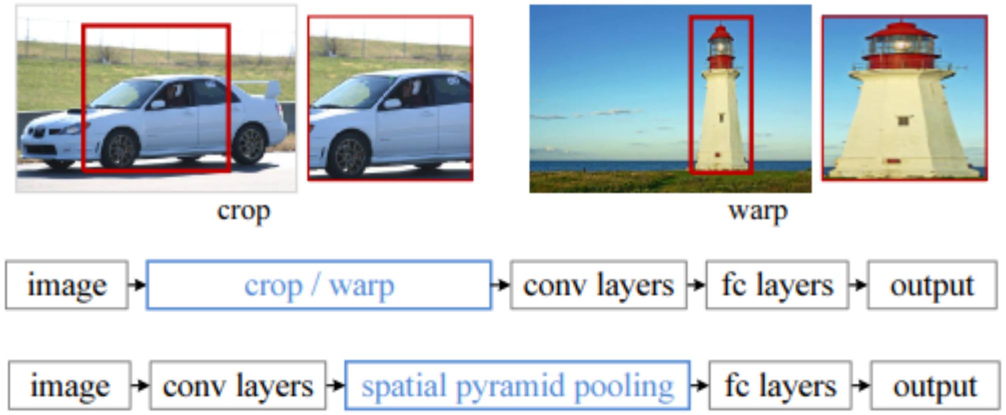
- **Semantic segmentation:** Classification (what) performed regionally (where).

- **End-to-end:** No introduction of SVM, HOG, or other intermediary into the workflow.

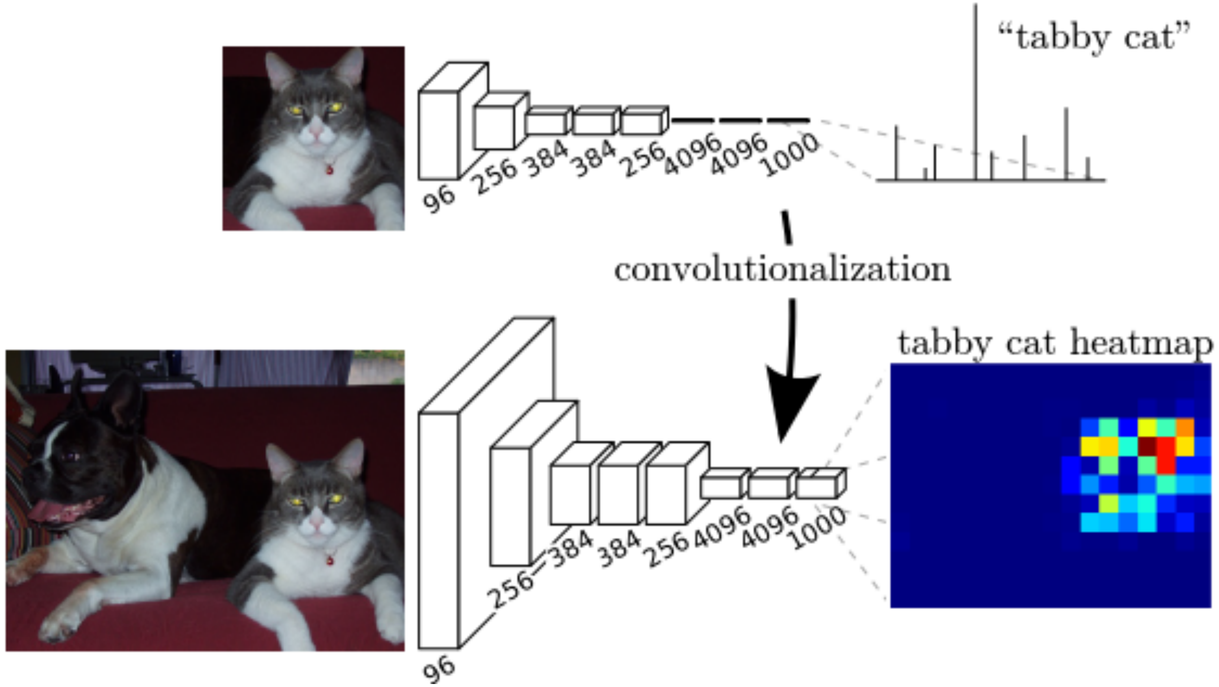


Other techniques to get around size constraint

- Cropping/warping
- Disregard the input size by **spatial pyramid pooling** right between the final conv output and the fully-connected layer![1]

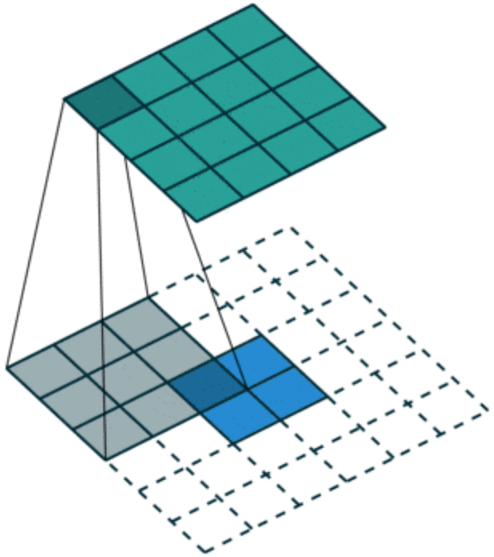


Connecting Outputs back to Pixel Location

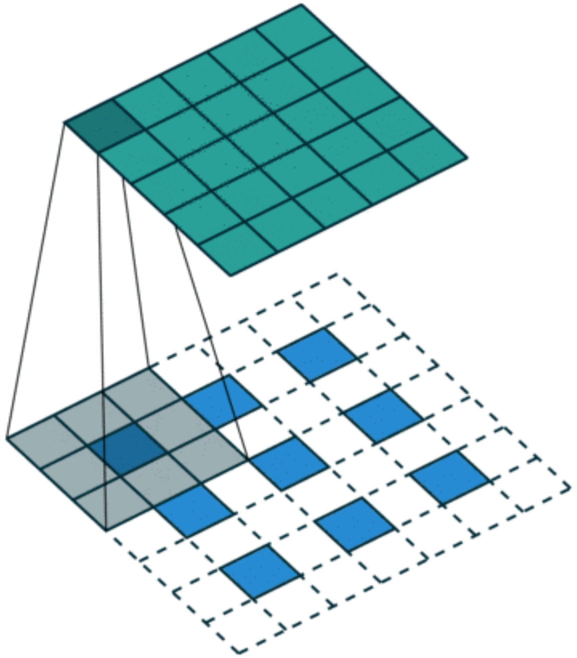


Ex. replacing AlexNet final layer (fully-connected) with 1000 (1x1) convolutions

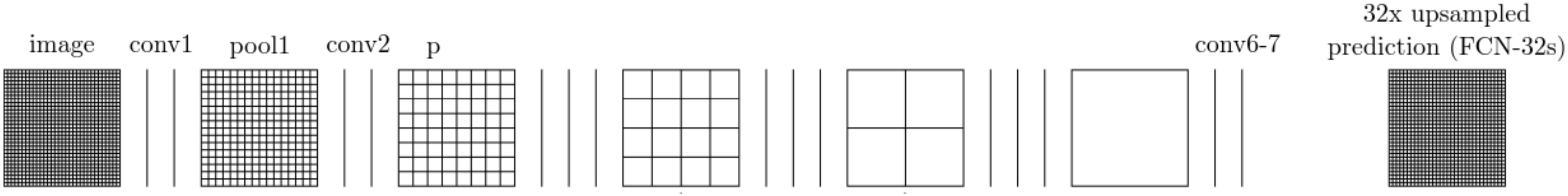
Deconvolution (upsample) visualization



Stride 2 case:



“Skip FCN



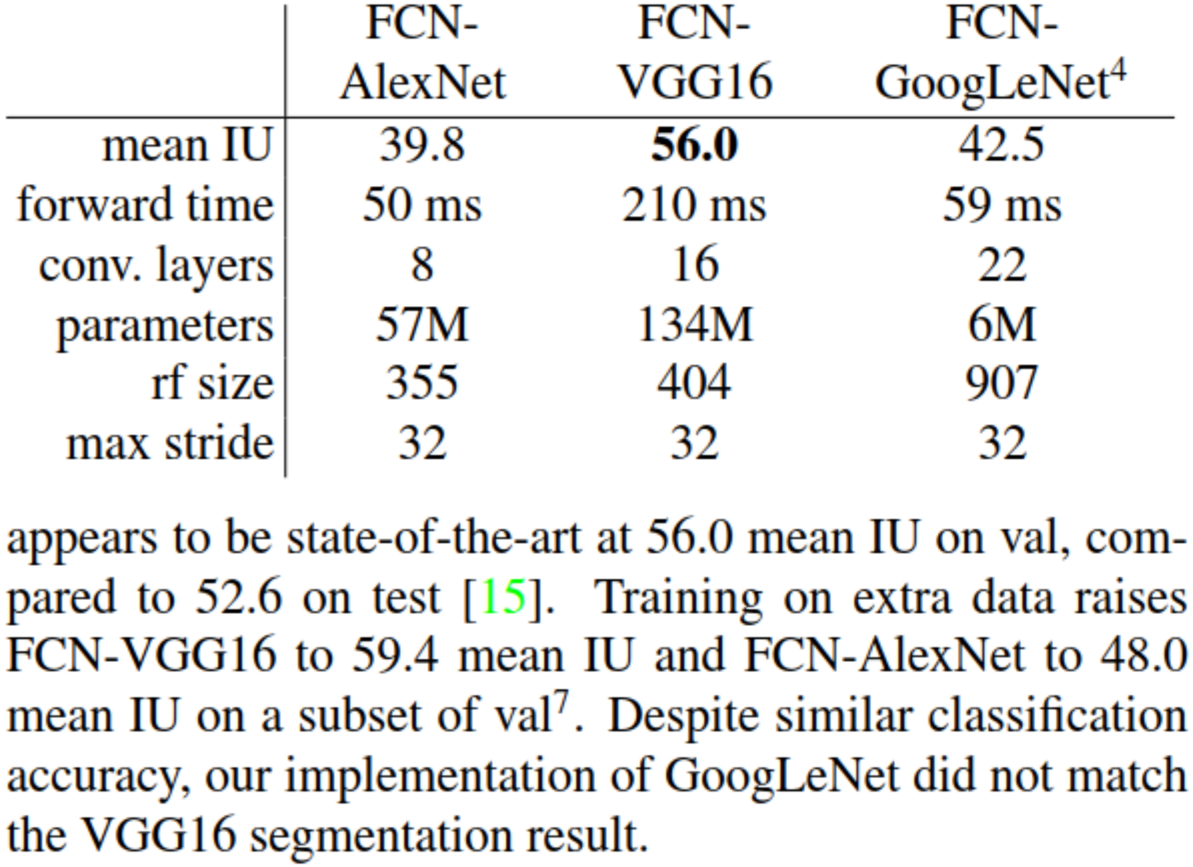
(FCN-32s) – Single stream (upsamples in single step)

(FCN-16s) – Combines predictions from pool 4 and final layer.

(FCN-8s) – Combines predictions from pool 3, pool 4 and final layer.

Adapting ILSVRC models into FCNs for dense predictions (FCN-32s)

- AlexNet, GoogLeNet and VGG-16.
- Replace final fully-connected layer with (1x1) convolutions.
    - 21 filters, each corresponding to a class of PASCAL VOC 2011 segmentation challenge (and background)
- Single deconvolution added for upsampling
    - 21 filters, each corresponding to a class of PASCAL VOC 2011 segmentation challenge (and background)



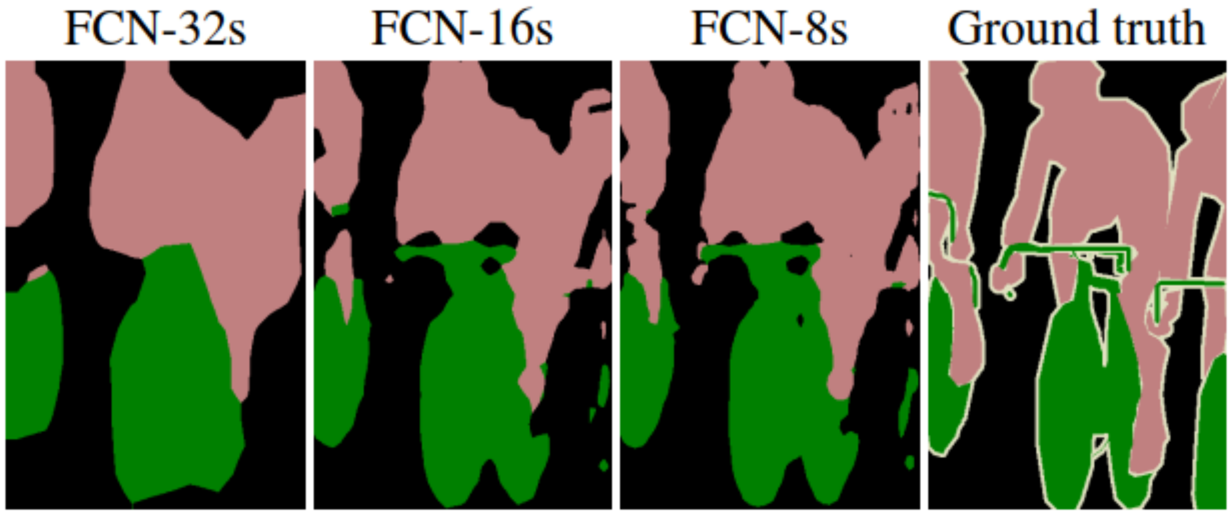
	FCN- AlexNet	FCN- VGG16	FCN- GoogLeNet[4]
mean IU	39.8	**56.0**	42.5
forward time	50 ms	210 ms	59 ms
conv. layers	8	16	22
parameters	57M	134M	6M
rf size	355	404	907
max stride	32	32	32

appears to be state-of-the-art at 56.0 mean IU on val, compared to 52.6 on test [15]. Training on extra data raises FCN-VGG16 to 59.4 mean IU and FCN-AlexNet to 48.0 mean IU on a subset of val[7]. Despite similar classification accuracy, our implementation of GoogLeNet did not match the VGG16 segmentation result.

Customized DAGs?

- Divide output stride in half (32 ⟶ 16)
- (1x1) convolution layer on top of *pool 4* for additional class-predictions.
- This is fused with final conv layer by a 2x upsampling and subsequent summation. (deemed "*deep jet*")

- FCN-8s following similar procedure, but reducing stride to 8 and including *pool 3* as well.



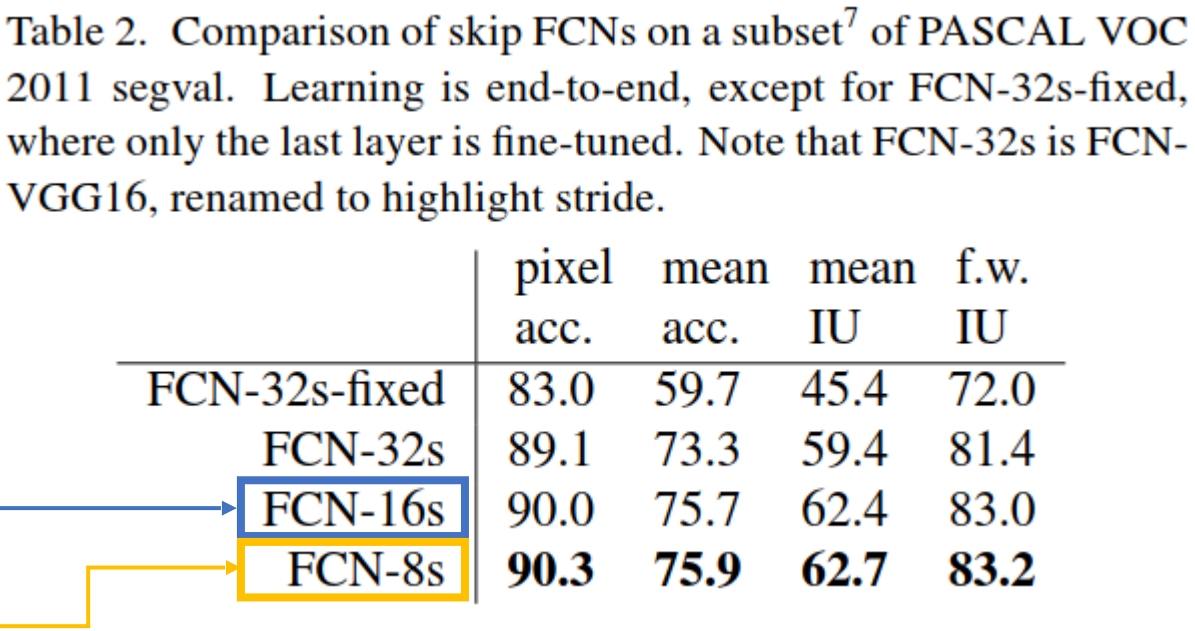


Table 2. Comparison of skip FCNs on a subset[7] of PASCAL VOC 2011 segval. Learning is end-to-end, except for FCN-32s-fixed, where only the last layer is fine-tuned. Note that FCN-32s is FCN-VGG16, renamed to highlight stride.

	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**

Training

- SGD with momentum.
- Minibatch size of 20.
- Fixed learning rates of 10^{-3}, 10^{-4}, and 5^{-5} for FCN-AlexNet, FCN-VGG16, and FCN-GoogLeNet, respectively, chosen by line search.
- We use momentum 0.9, weight decay of 5^{-4} or 2^{-4}, and doubled learning rate for biases, **although we found training to be sensitive to the learning rate alone**.
- Dropout was included where used in the original classifier nets.

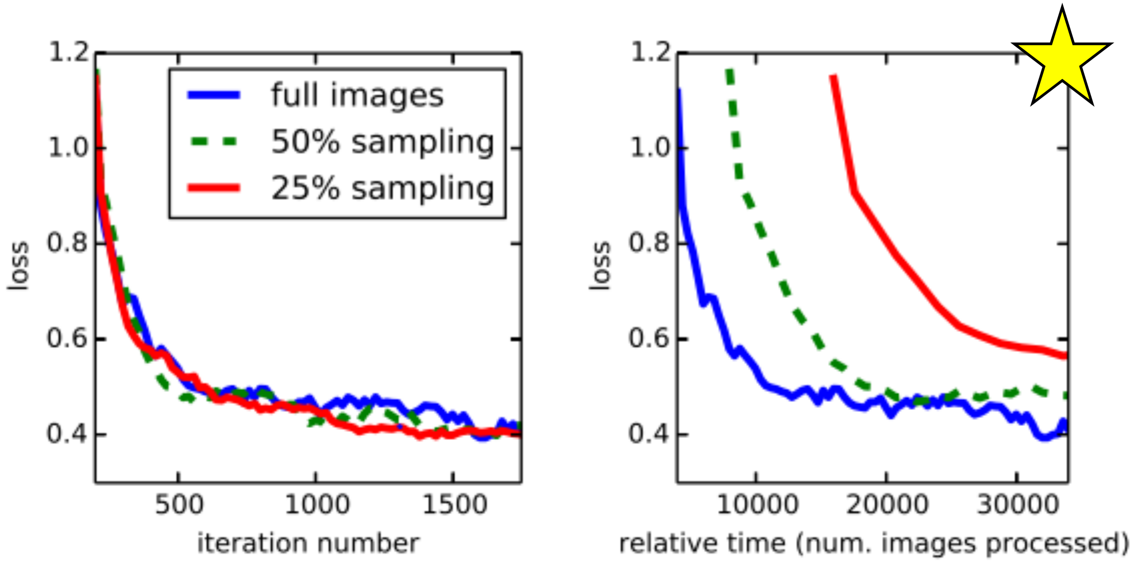
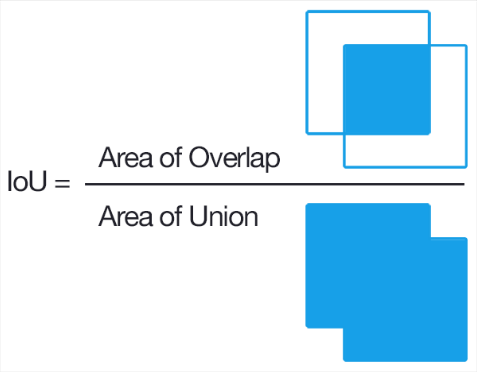


Figure 5. Training on whole images is just as effective as sampling patches, but results in faster (wall time) convergence by making more efficient use of data. Left shows the effect of sampling on convergence rate for a fixed expected batch size, while right plots the same by relative wall time.

Results – PASCAL VOC 2011/2012

- IU - Regional Intersection over Union



- *“Inference time is reduced 114× (convnet only, ignoring proposals and refinement) or 286× (overall).”*

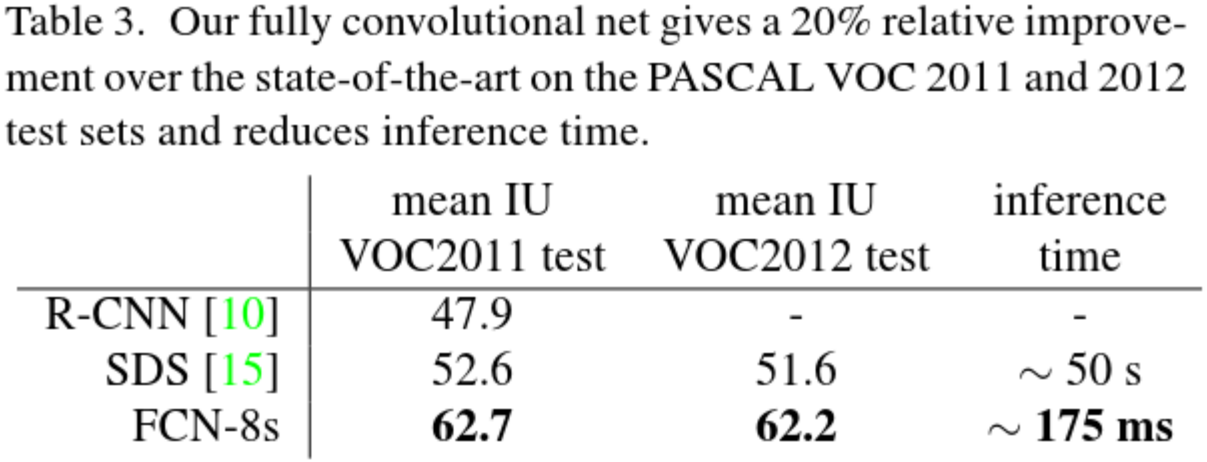


Table 3. Our fully convolutional net gives a 20% relative improvement over the state-of-the-art on the PASCAL VOC 2011 and 2012 test sets and reduces inference time.

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [10]	47.9	-	-
SDS [15]	52.6	51.6	$\sim$ 50 s
FCN-8s	**62.7**	**62.2**	$\sim$ **175 ms**

Results (…continued) – NYUDv2

- RGBD – Early fusion of color layers at input.
- HHA(?) – Depth embedding as horizontal disparity, height above ground, and the angle of the local surface normal with the inferred gravity direction.
- RGB-HHA – jointly trained fusion.

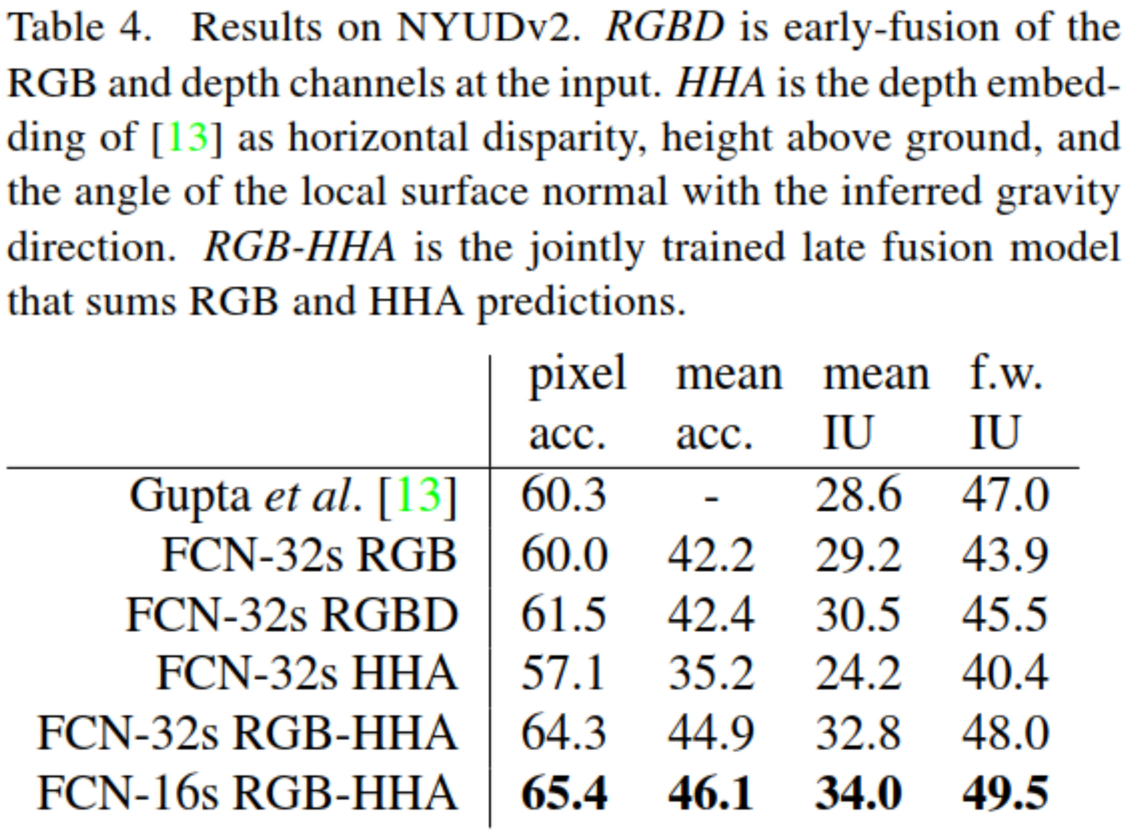


Table 4. Results on NYUDv2. *RGBD* is early-fusion of the RGB and depth channels at the input. *HHA* is the depth embedding of [13] as horizontal disparity, height above ground, and the angle of the local surface normal with the inferred gravity direction. *RGB-HHA* is the jointly trained late fusion model that sums RGB and HHA predictions.

	pixel acc.	mean acc.	mean IU	f.w. IU
Gupta *et al.* [13]	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**

Results (…continued) – SIFT Flow

- 2688 images, 33 semantic categories.
- 200 for test.

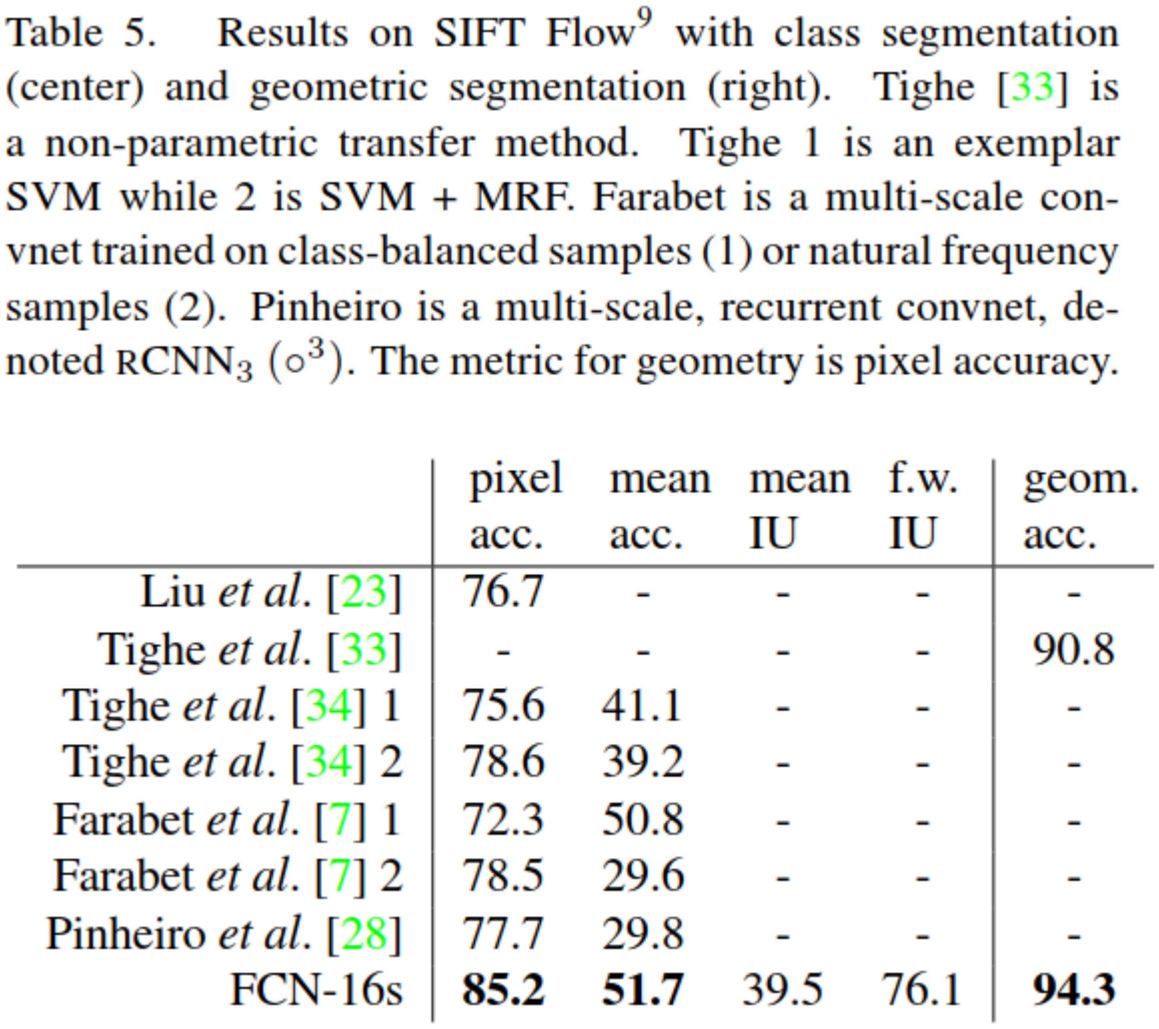


Table 5. Results on SIFT Flow[9] with class segmentation (center) and geometric segmentation (right). Tighe [33] is a non-parametric transfer method. Tighe 1 is an exemplar SVM while 2 is SVM + MRF. Farabet is a multi-scale convnet trained on class-balanced samples (1) or natural frequency samples (2). Pinheiro is a multi-scale, recurrent convnet, denoted RCNN$_3$ ($\circ^3$). The metric for geometry is pixel accuracy.

	pixel	mean	mean	f.w.	geom.
	acc.	acc.	IU	IU	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.* [7] 1	72.3	50.8	-	-	-
Farabet *et al.* [7] 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**

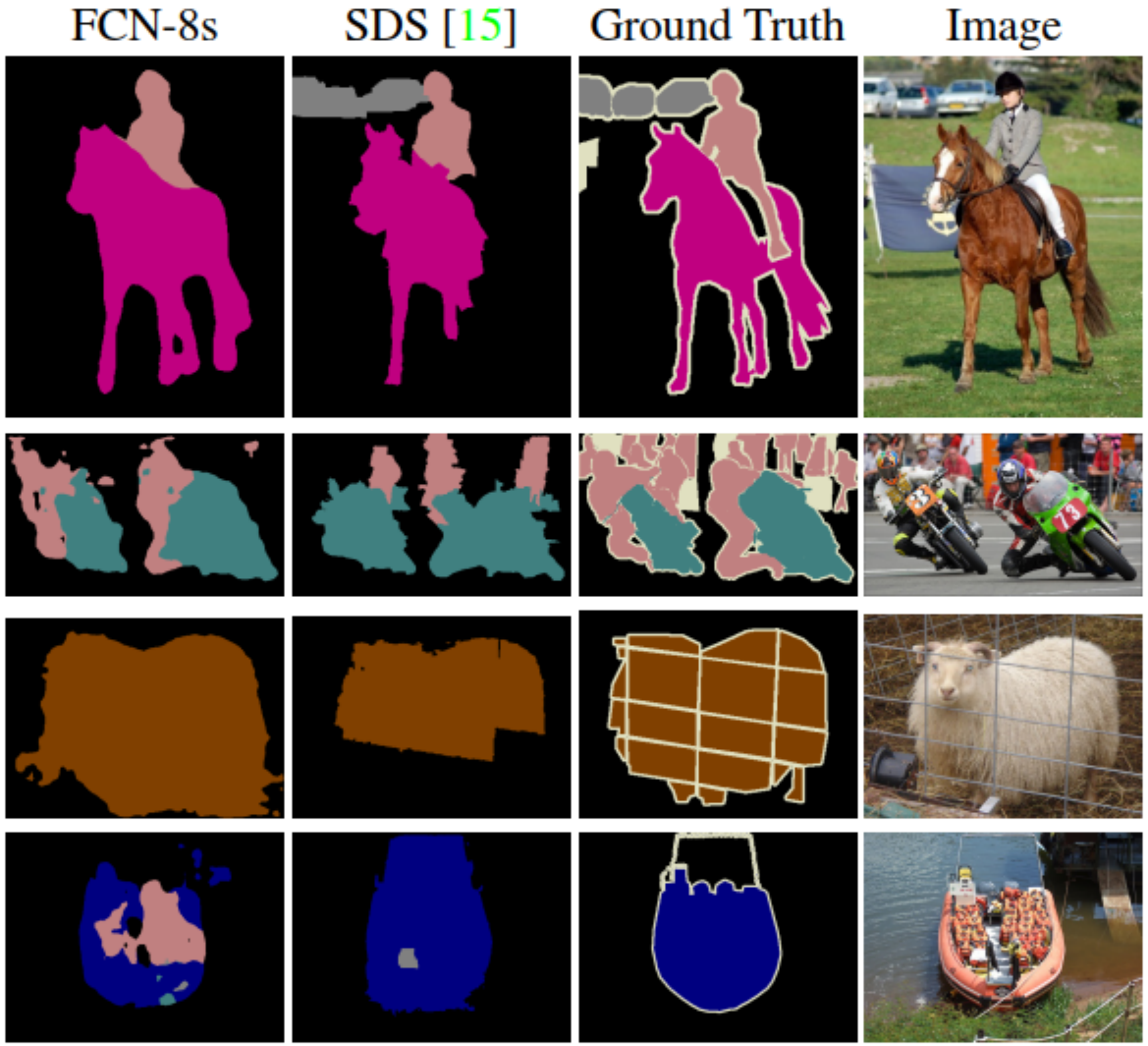


Figure 6. Fully convolutional segmentation nets produce state-of-the-art performance on PASCAL. The left column shows the output of our highest performing net, FCN-8s. The second shows the segmentations produced by the previous state-of-the-art system by Hariharan *et al*. [15]. Notice the fine structures recovered (first row), ability to separate closely interacting objects (second row), and robustness to occluders (third row). The fourth row shows a failure case: the net sees lifejackets in a boat as people.

Takeaways

- *“Fully convolutional training can balance classes by weighting or sampling the loss. Although our labels are mildly unbalanced (about 3/4 are background), we find class balancing unnecessary.”*
- *“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.”*
- Significant improvement (performance & speed-boosts) versus state-of-the-art in all three test cases.