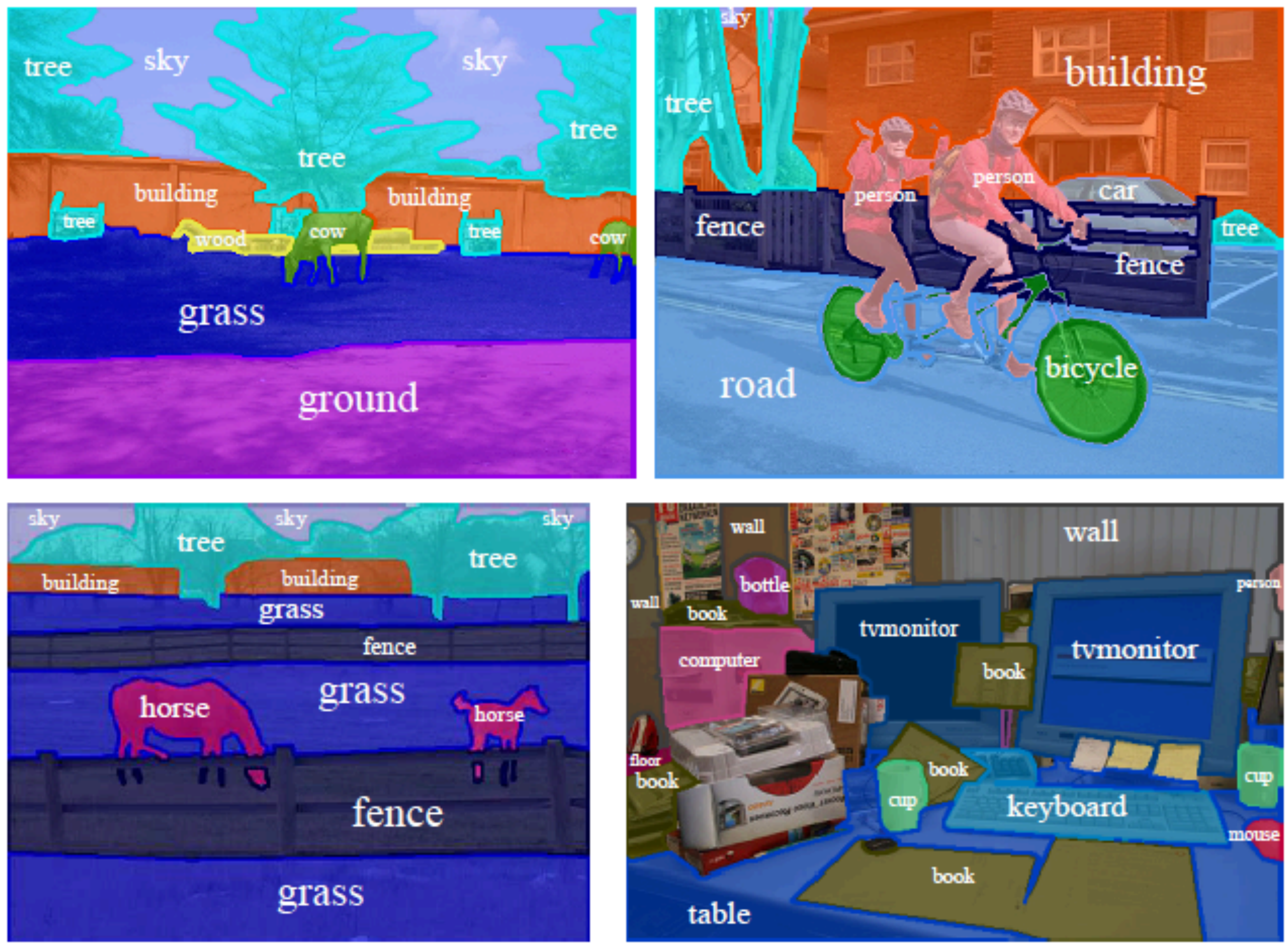
Semantic segmentation

D.A. Forsyth

The problem

- Tag each pixel with a class name for some set of classes

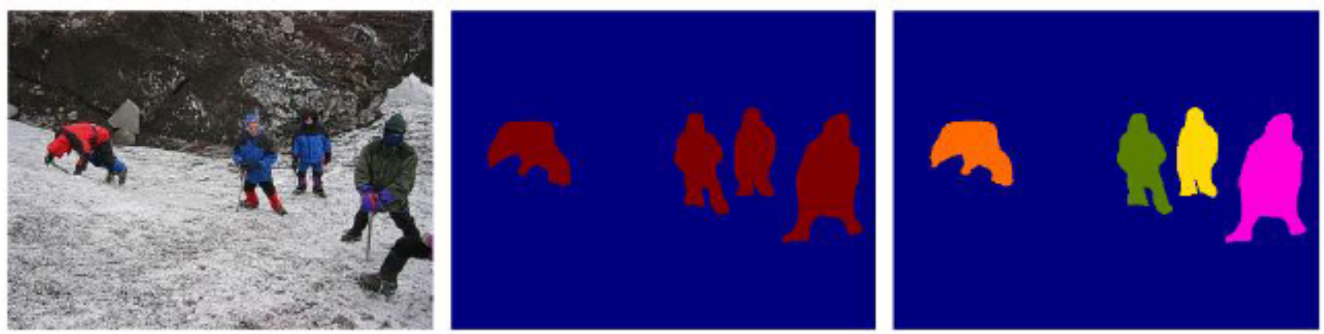


Mottaghi et al

OR

- Tag every pixel,
  - BUT different instances of the same class get different tags

instance-level labelling



pixel-level labelling

Issues

- Label distributions are skewed
  - Pascal 2010
    - from Mottaghi et al 14

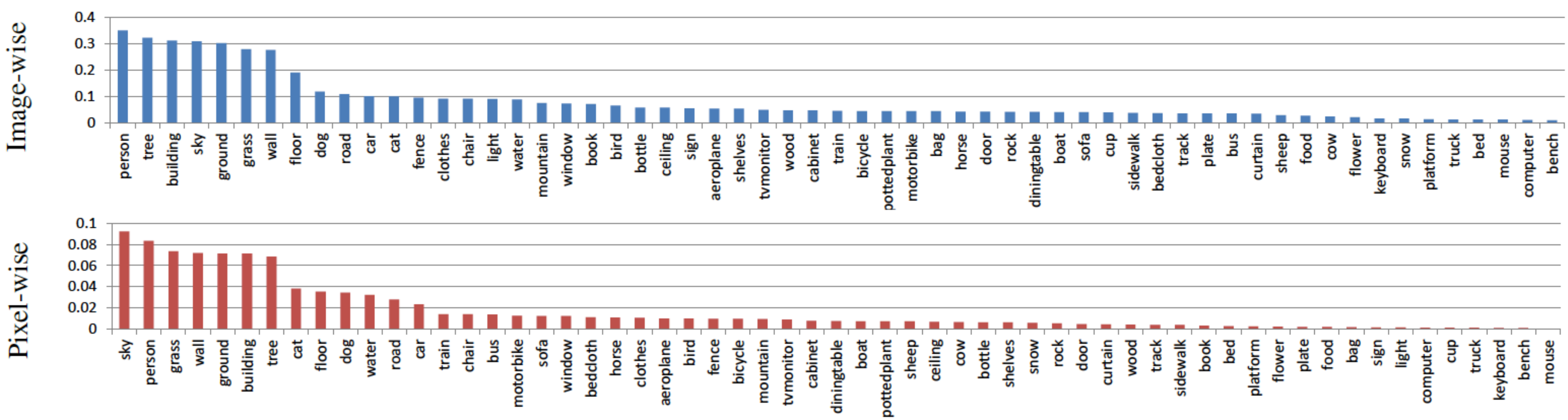
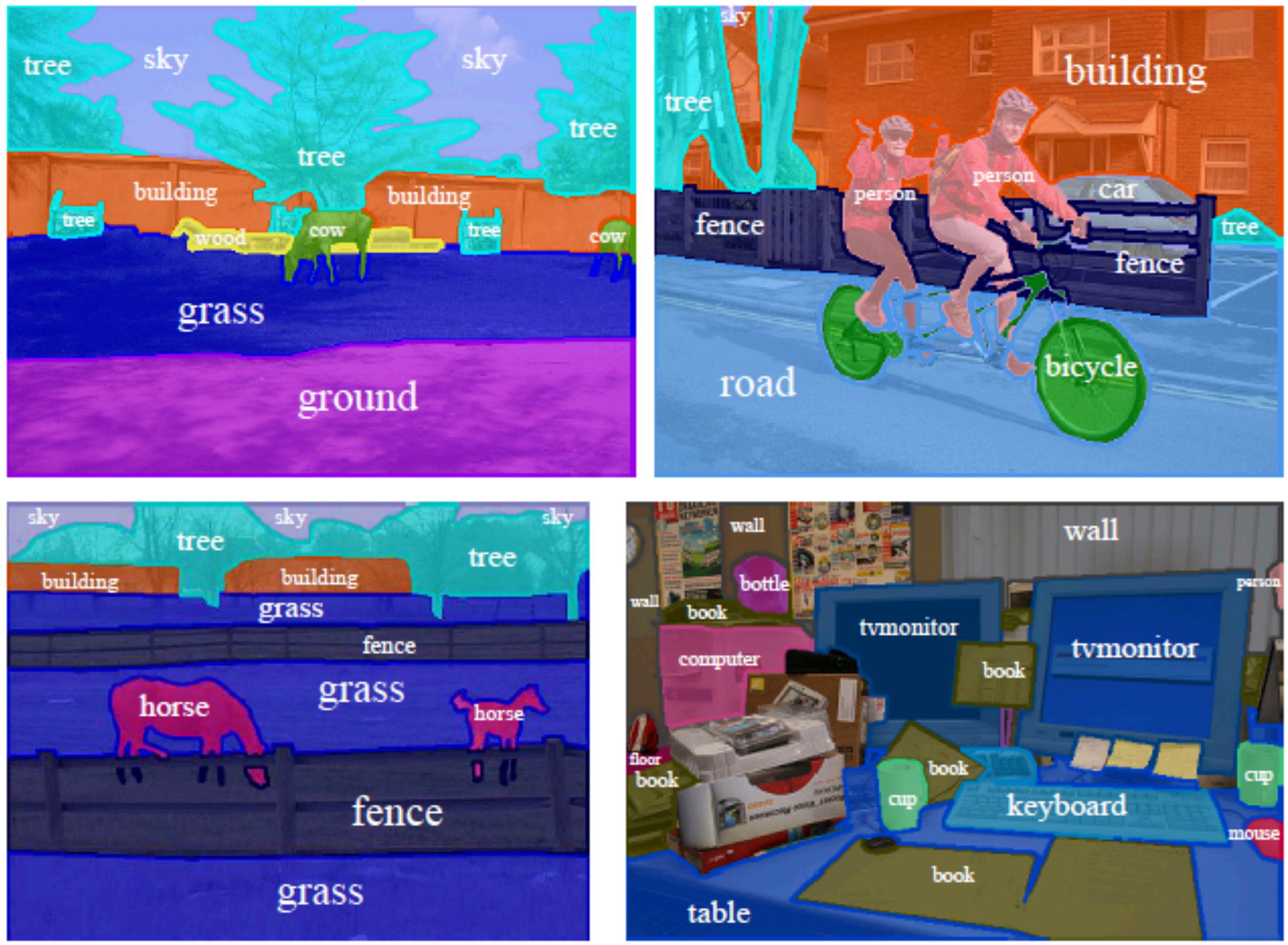


Figure 2. Distribution of pixels and images for the 59 most frequent categories. See text for the statistics.

Issues

- Some ambiguity in labelling

This is a pixel-level labelling



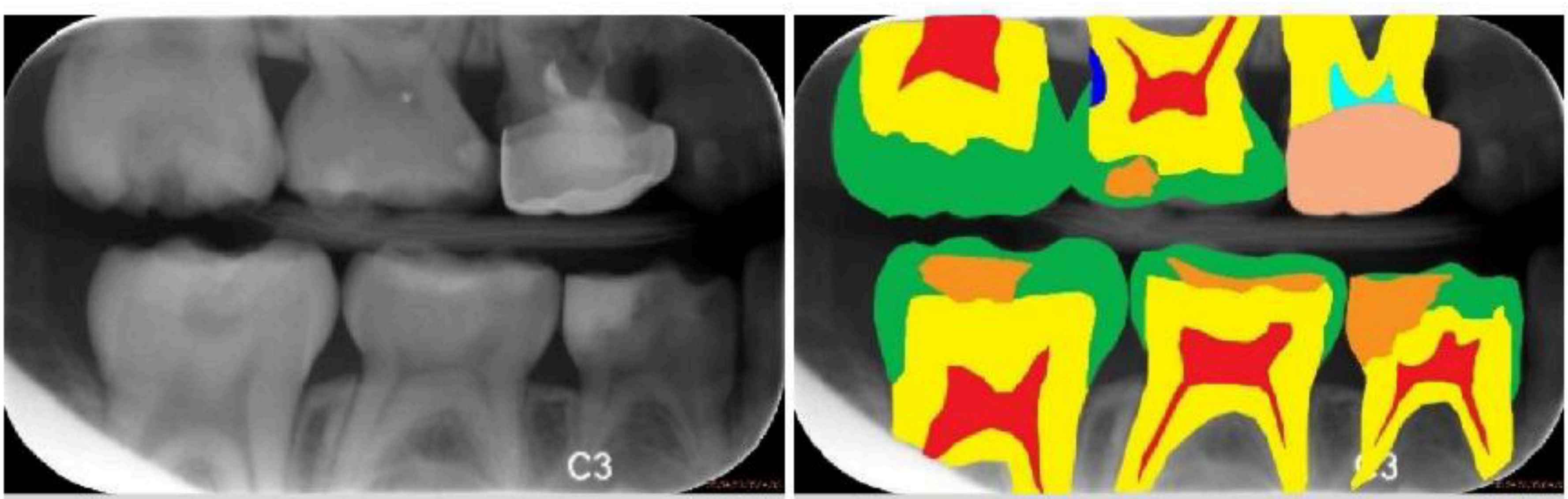
More issues

- Data
- Spatial models
- Appearance models
- Managing scale, context, etc.

Why bother?

Driving (maybe - why everything?)

Why bother?

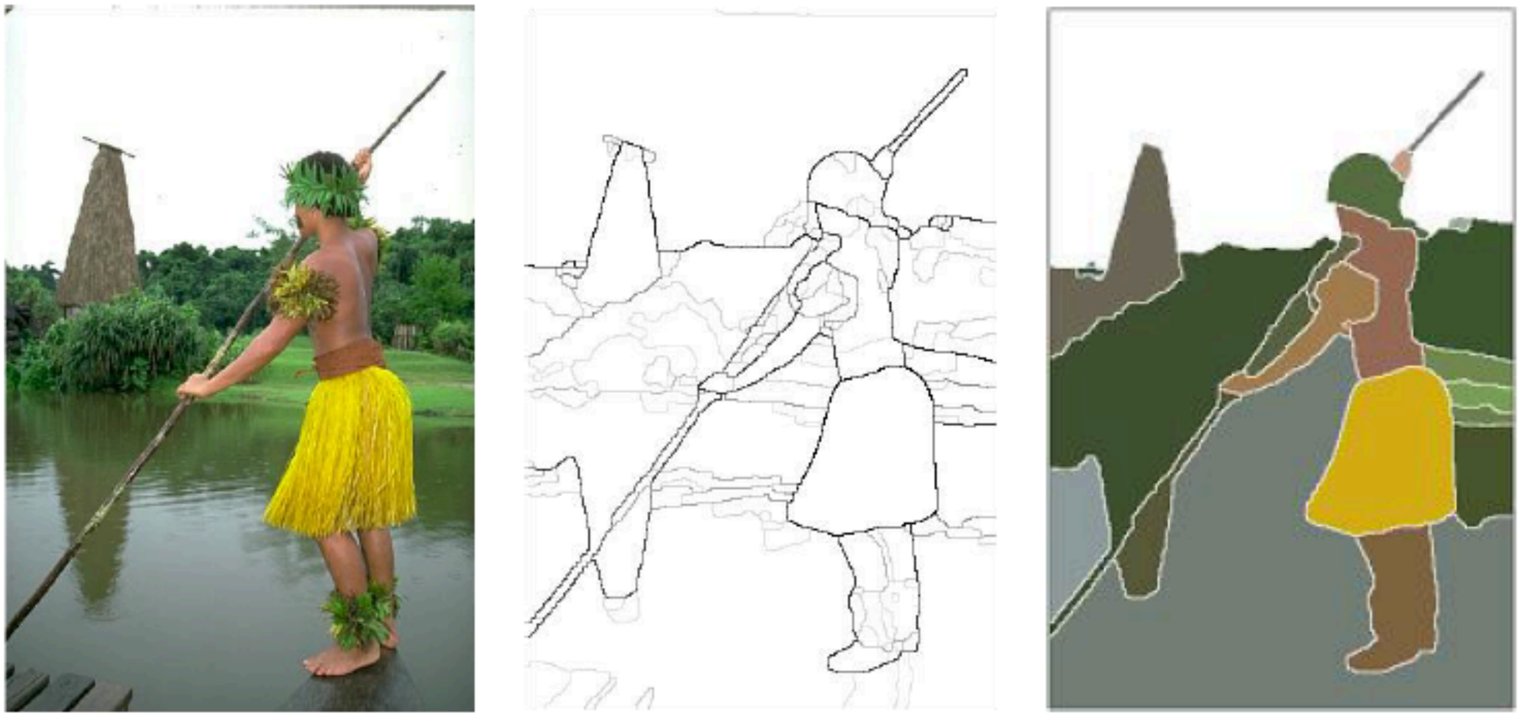


Medical applications (compelling)

Important variants

- Partial semantic segmentation
  - some pixels unlabelled
- Thing segmentation
  - label “things”
    - count nouns (car, person, dog…)
- Stuff segmentation
  - label “stuff”
    - mass nouns (grass, sky, water…)
- Panoptic segmentation
  - each pixel gets a label
  - each instance of a count noun gets a different label (person-a, etc)
    - I *think* MS-COCO and Cityscapes use the term differently

Contrast with segmentation



Learning a semantic segmenter should be *MUCH* easier
cause you KNOW what label each pixel should have
and labels transfer across images

Evaluation

To assess performance, we rely on the standard Jaccard Index, commonly known as the PASCAL VOC intersection-over-union metric IoU = TP / (TP+FP+FN) [1], where TP, FP, and FN are the numbers of true positive, false positive, and false negative pixels, respectively, determined over the whole test set. Owing to the two semantic granularities, i.e. classes and categories, we report two separate mean performance scores: $IoU_{category}$ and IoU_{class}. In either case, pixels labeled as void do not contribute to the score.

Evaluation, II

It is well-known that the global IoU measure is biased toward object instances that cover a large image area. In street scenes with their strong scale variation this can be problematic. Specifically for traffic participants, which are the key classes in our scenario, we aim to evaluate how well the individual instances in the scene are represented in the labeling. To address this, we additionally evaluate the semantic labeling using an instance-level intersection-over-union metric iIoU = iTP / (iTP+FP+iFN). Again iTP, FP, and iFN denote the numbers of true positive, false positive, and false negative pixels, respectively. However, in contrast to the standard IoU measure, iTP and iFN are computed by weighting the contribution of each pixel by the ratio of the class' average instance size to the size of the respective ground truth instance. It is important to note here that unlike the instance-level task below, we assume that the methods only yield a standard per-pixel semantic class labeling as output. Therefore, the false positive pixels are not associated with any instance and thus do not require normalization. The final scores, $\text{iIoU}_{\text{category}}$ and $\text{iIoU}_{\text{class}}$, are obtained as the means for the two semantic granularities.

Cityscapes

Many variants

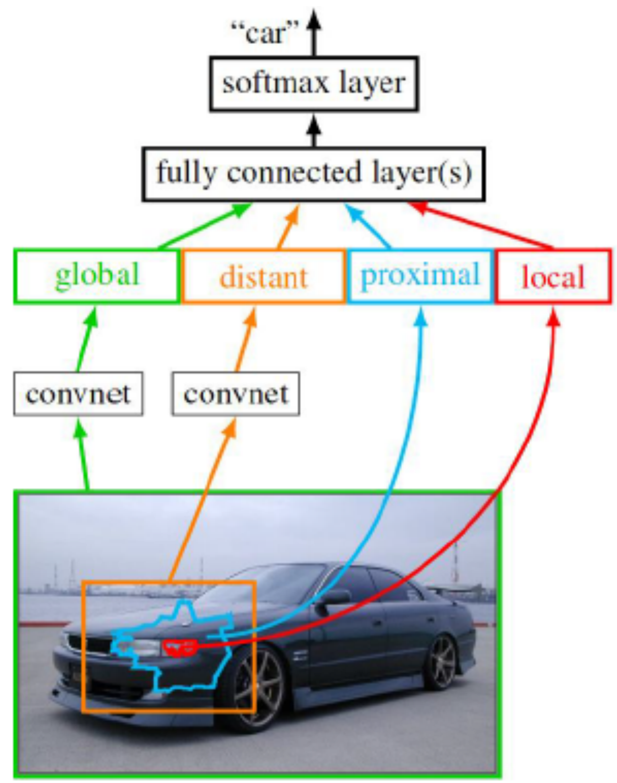
- Segment into polygons (rather than label pixels)
  - might do some form of matching first
- Instance segmentation
  - need to keep track of which instance

(Some) Datasets

- Cityscapes
  - https://www.cityscapes-dataset.com/benchmarks/
- Pascal VOC 2010 context
  - https://cs.stanford.edu/~roozbeh/pascal-context/
- Kitti
  - http://www.cvlibs.net/datasets/kitti/eval_semantics.php
  - also see other annotations at bottom of page
- Mapillary vistas
  - https://research.mapillary.com/img/publications/ICCV17a.pdf
- MS COCO
  - http://cocodataset.org/#panoptic-2018

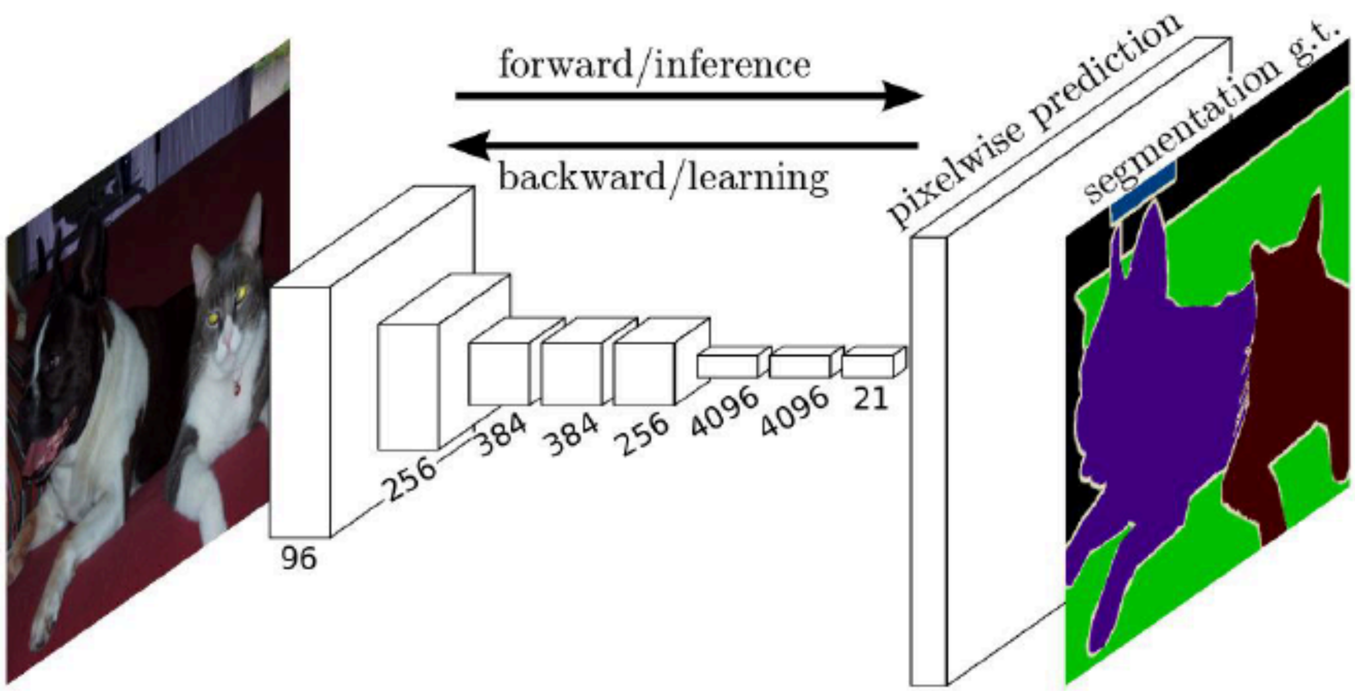
Early ideas

- Label pixel using
  - its appearance
  - features for context, etc.
    - proximal
    - distant
    - global
  - etc



Procedure

- Fully convolutional network
  - with very large receptive fields
  - some skip connections
- Train with cross-entropy loss



Long et al

Procedure, II

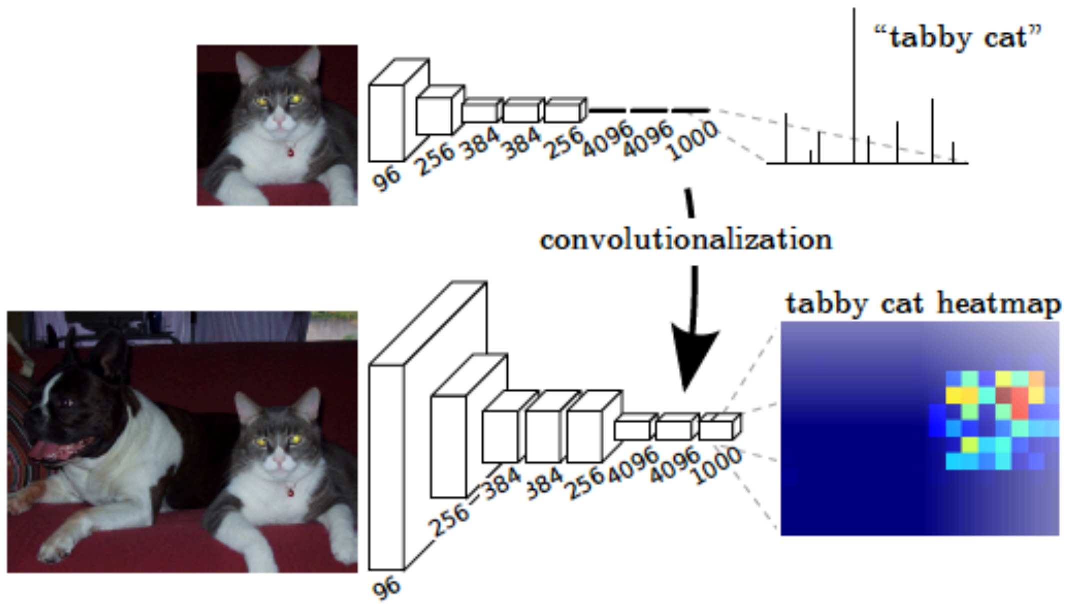


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Procedure, III

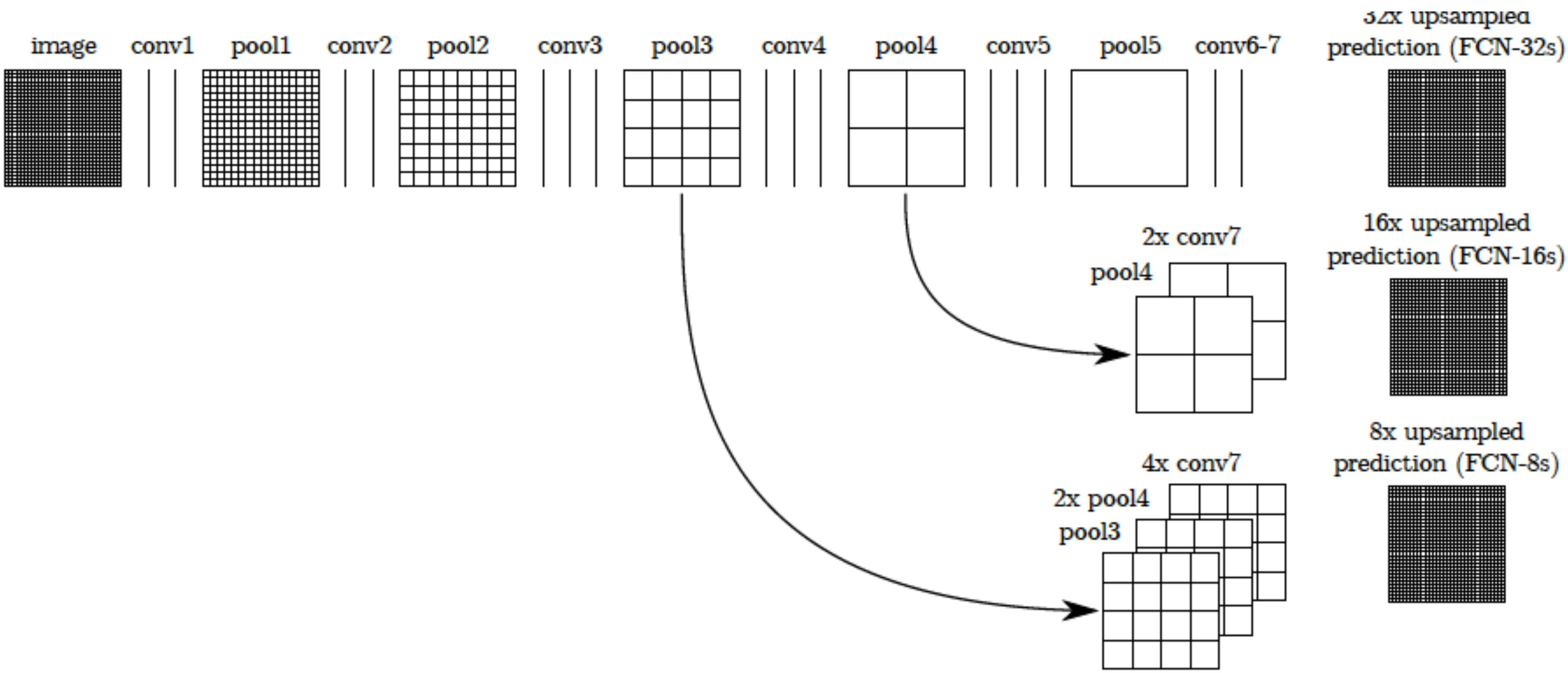


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines. First row (FCN-32s): Our single-stream net, described 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.

Procedure, IV

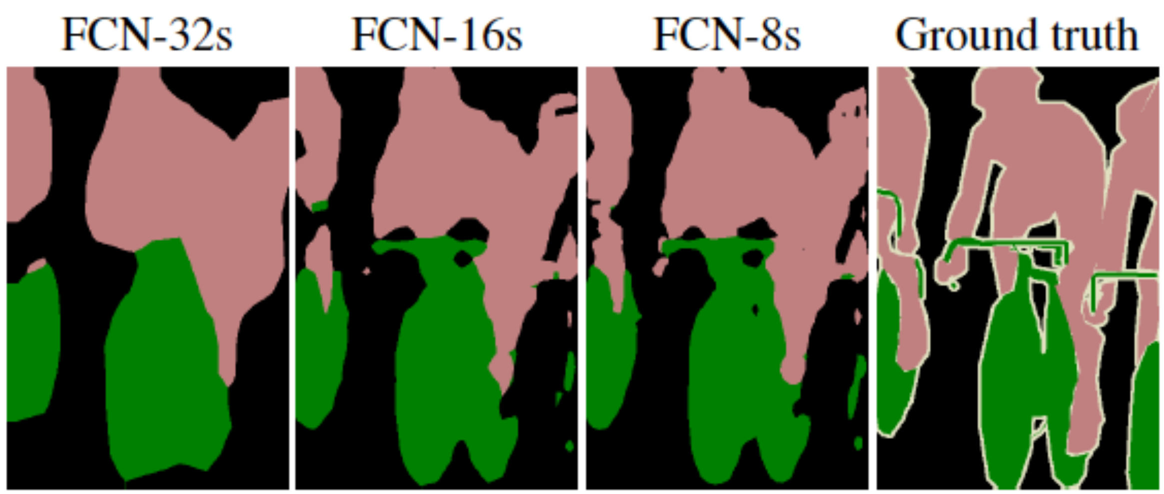


Figure 4. Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

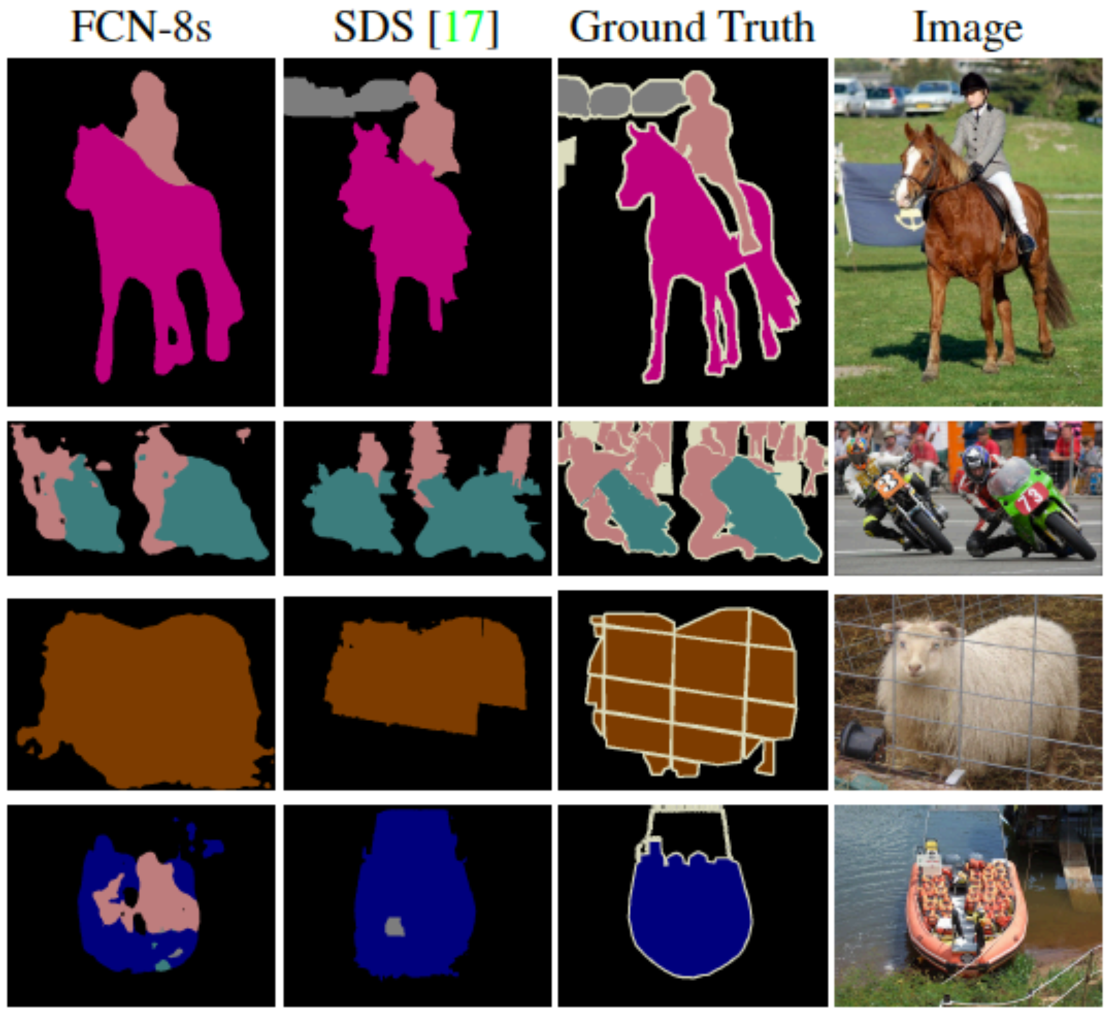


Figure 6. Fully convolutional segmentation nets produce state-of-the-art performance on PASCAL. The left column shows the output of our highest performing net, FCN-8s. The second shows the segmentations produced by the previous state-of-the-art system by Hariharan *et al*. [17]. Notice the fine structures 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.

Spatial constraints on regions

- Q: do we need them?
- A: (jury is out)
  - Yes:
    - thing regions need to form structures
    - stuff regions need to be coherent
  - No:
    - pixel appearance is dispositive
    - anyhow, the model learns a prior from all the data it sees

- Q: how do we build them?
  - A: Fully connected CRF's
  - A: GAN machinery

CRF obstacles

- Which pixels are connected to which?
  - and with what energy?
  - tricky to learn
  - neighbors only is weak in practice
- How to do inference with multiple labels?
  - Two very important tricks apply in the easy case
    - other cases very hard
      - alpha-expansion
      - alpha-beta swap