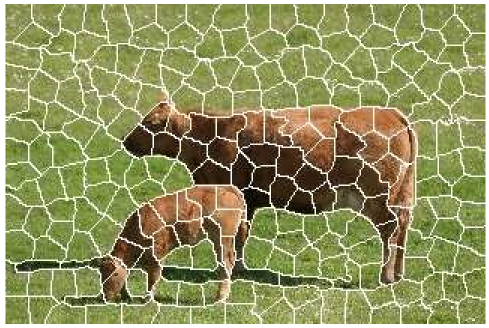
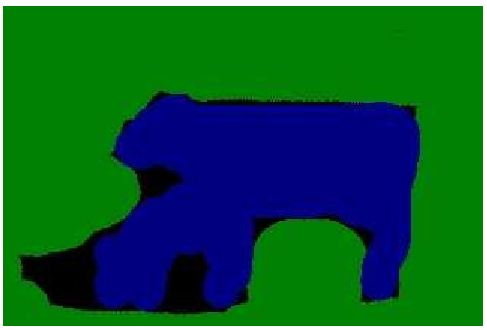
How to train and run multi-class image segmentation

Stephen Gould sgould@stanford.edu [mailto:sgould@stanford.edu]
August 2008

Background

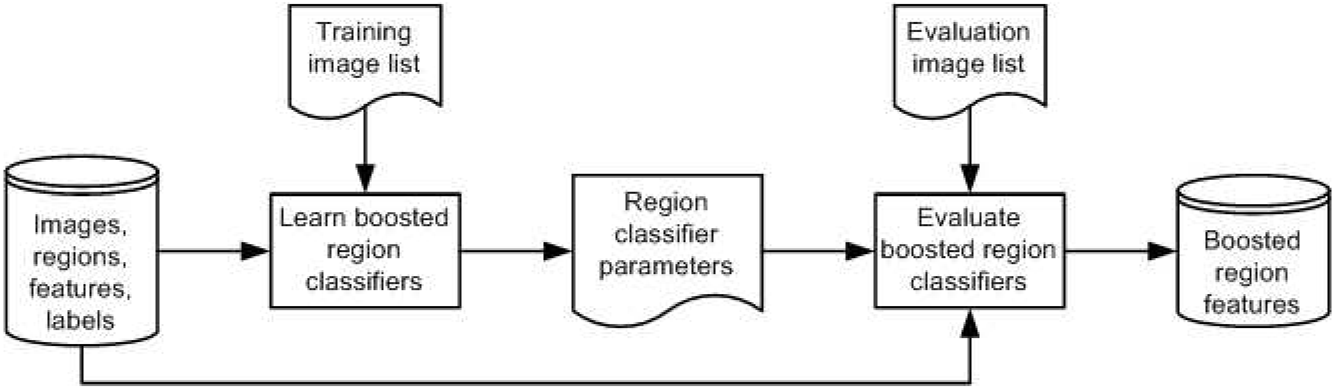
Multi-class image segmentation (or pixel labeling) aims to label every pixel in an image with one of a number of classes (e.g., grass, sky, water, etc). Since classifying every pixel can be computationally expensive, many state-of-the-art methods first over-segment the image into *superpixels* (or small coherent regions) and classify each region. The following figure shows an example of an image, its over-segmentation, and region labels.



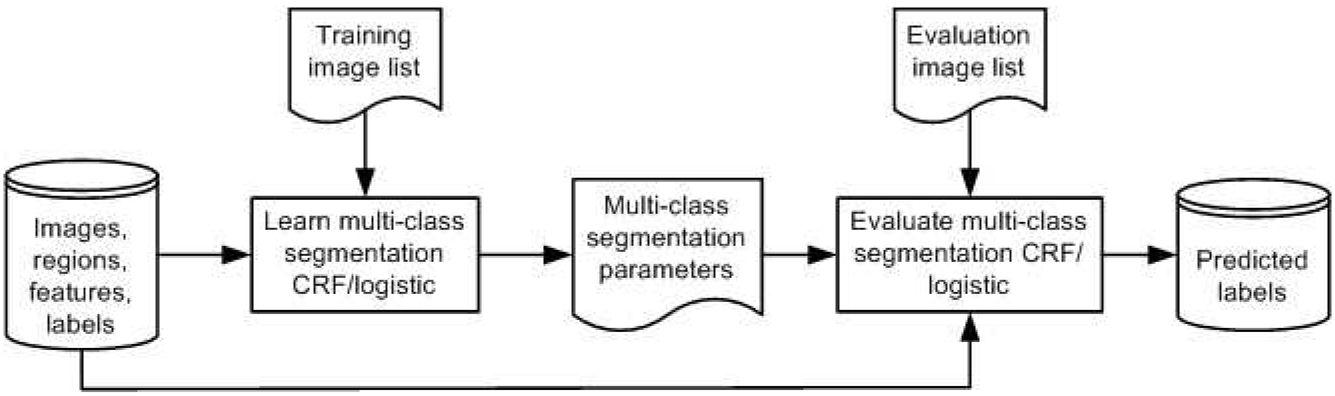


Our method first extracts appearance (color and texture), geometry and location features for each superpixel region.

We then learn boosted classifiers over these features for each region class.



Finally, we learn a CRF or logistic model using the output of the boosted classifiers as features.



A step-by-step guide to the process is provided below.

Step-by-step Guide

Data Preparation

The multi-class image segmentation algorithms that we describe take as input an image (`<base>.jpg`) and corresponding over-segmentation (`<base>.seg`) and ground-truth pixel labels (`<base>.txt`). The images are usually resized to approximately the same dimensions (e.g. 320-by-240). Over-segmentation and groundtruth labeling are described below.

Groundtruth Labeling

The groundtruth file (`<base>.txt`) corresponding to an image is a text file containing one number (integer) per pixel and arranged in a matrix the same size as the image. The integers correspond to different class labels and are zero-based. You can use the regionLabeler application to generate or view groundtruth files.

Over-Segmentation

The over-segmentation file (`<base>.seg`) corresponding to an image is a text file containing one number (integer) per pixel and arranged in a matrix the same size as the image. The integers correspond to different superpixels and are zero-based. Free online code such as that provided by Greg Mori [http://www.cs.sfu.ca/~mori/research/superpixels/code/] or Pedro Felzenszwalb [http://people.cs.uchicago.edu/~pff/segment/] can be used for this purpose. Note that in our models each pixel is assigned to one and only one superpixel.

Extracting Region Features

Once the image regions/superpixels have been defined, we extract appearance (color and texture), geometry and location based features for each superpixel. These are cached in features files (`<base>.features.txt`) which contain one feature vector per superpixel. If the features are to be used directly as input to the logistic/CRF multi-class segmentation model, then a bias term (constant feature) can be appended which helps to model prior class preferences.

During this step we also extract the groundtruth label for each superpixel. This is done by finding the maximum occurring pixel label within each superpixel. The labels are written to corresponding labels file (`<base>.labels.txt`). This can be skipped if groundtruth labels are not available (such as when using a previously trained multi-class segmentation model to label new images).

The following command processes all `.jpg` (and corresponding `.seg`) files in `$IMAGEDIR` and output feature and label files to `$OUTPUTDIR`.

```
bin/segImageExtractFeatures -v -o $OUTPUTDIR $IMAGEDIR
```

Building Boosted Region Classifiers

Performance can be greatly improved if learn a boosted classifier for each class instead of using the raw appearance, geometry and location features described above. In this step we learn a one-versus-all classifier for each groundtruth label using the cached superpixel features and labels from the previous step. Once the classifiers are learned new feature files (`<base>.boosted.txt`) are created containing the output score for each of the learned classifiers on each superpixel in an image. Bias terms are also included so that these features files can be used directly by the logistic/CRF models.

```
bin/segImageTrainBoostedFeatures -v -featuresDir $OUTPUTDIR -o ${OUTPUTDIR}/demo \
    -rounds 200 -splits 2 $TRAININGLIST
bin/segImageEvalBoostedFeatures -v -featuresDir $OUTPUTDIR -i ${OUTPUTDIR}/demo \
    -includeBias $NUMCLASSES $ALLIMAGESLIST
```

`$ALLIMAGESLIST` is the name of the file containing a list (one per line) of base names (image filenames without the `.jpg` extension) to be processed. The file can be created using the following shell commands

```
foreach i (`ls $IMAGEDIR/*.jpg`)
  echo $i:r >> $ALLIMAGESLIST
end
```

Learning the Multi-class Image Segmentation Model

Training involves learning the parameters of the logistic or CRF model. The logistic model assumes that each superpixel is independent and learns a multi-class logistic classifier based on the (raw or boosted) features. The CRF model adds a pairwise term between neighboring superpixels which acts to smooth the predicted labels.

```
bin/segImageTrainModel -v -o ${OUTPUTDIR}/demo.crf.model \
    -imgDir $IMAGEDIR -featuresDir $OUTPUTDIR \
    -model "CRF" -regNodes 1.0e-9 -regEdges 1.0e-2 $TRAININGLIST
```

The regularization parameters (`regNodes` and `regEdges`) should be set by cross-validation on the training set.

Evaluating the Multi-class Image Segmentation Model

After training, the accuracy of a model can be evaluated by comparing predicted labels against groundtruth labels on a hold out set of data (i.e., data that was not used during the training process). The model can also be used to label new images that will be used as input for some other vision task.

```
bin/segImageEvalModel -imgDir $IMAGEDIR -featuresDir $OUTPUTDIR \
    -model "CRF" ${OUTPUTDIR}/demo.crf.model $EVALLIST
```

Example Results

We ran experiments on the 21-class MSRC database [http://research.microsoft.com/vision/cambridge/recognition/MSRC_ObjCategImageDatabase_v2.zip]. We over-segmented each image into approximately 200 superpixels using the method of Ren and Malik (with code [http://www.cs.sfu.ca/~mori/research/superpixels/code/] provided by Greg Mori). We randomly split the images into 296 training and 295 test. Results are shown in the table below.

Experimental Results (Pixelwise Accuracy)	
Logistic on raw features	62.1%
CRF on raw features	67.3%
Logistic on boosted features	65.9%
CRF on boosted features	**76.0%**

Download dataset and training script: msrcsegexample.tar.gz (43MB).

See Also

svlSegImage, svlRegionFeatures, regionLabeler, segImageExtractFeatures, segImageTrainBoostedFeatures, segImageEvalBoostedFeatures, segImageTrainModel, segImageEvalModel

References

1. S. Gould, J. Rodgers, D. Cohen, G. Elidan, D. Koller, "Multi-Class Segmentation with Relative Location Prior," IJCV, 2008. *(The SVL implementation is a slight variation of the baseline model described in this paper.)*
2. G. Heitz, S. Gould, A. Saxena, D. Koller, "Cascaded Classification Models: Combining Models for Holistic Scene Understanding," NIPS, 2008.