Image Segmentation and Scene Understanding

Xiaohe Shi
Supervisor: Stephen Gould
Outline

- Background
- Website for scene understanding
- Scene understanding model with relative location information
Background

Scene understanding:

- [Image of a sheep]
  → [Image of a sheep outline]

- [Image of a church]
  → [Image of a church outline with annotations: sky, building, tree]
Website for Scene Understanding

- Motivation:
  - For people to easily do scene understanding

- Different algorithms:
  - Pixel CRF, Region–based model...

- Different datasets:
  - 21–class MSRC dataset, 8–class Stanford Background dataset...
Welcome to the Scene Understanding website. Scene understanding is the task of getting a computer to automatically interpret the contents of an image. This can be done in a number of different ways ranging from describing the image textually to providing a fully annotated hierarchy of objects and their location within the image. In this project we focus on scene understanding by pixel labeling.

This website allows you to evaluate the performance of various scene understanding algorithms on your own data. To this end, you can submit images to our server and have the labeled image returned. If you are a computer vision researcher and would like us to include your method on this website, please email us.

You can also browse our current collection of images and results at http://sceneunderstanding.com.

Web Page Interface
Please select the image that you want to segment and choose the scene understanding model you want to evaluate. Click the Submit button to start processing the image.

image: [image selection]
model:
- Pairswise CRF on 21-class MSRC Dataset (Pairswise CRF models trained by Darwin 1.5)
- Unary Model on 21-class MSRC Dataset (Model with only unary potentials trained by Darwin 1.5)
- Pairswise CRF on 8-class Stanford Background Dataset (Pairswise CRF models trained by Darwin 1.5)
- Unary Model on 8-class Stanford Background Dataset (Model with only unary potentials trained by Darwin 1.5)

[Submit] (Warning: processing may take some time)
Model with Relative Location

- Relative location probability maps:
Train the model with relative location information:

1. Make initial predictions on training set by current model.
2. Generate relative location features.
3. Retrain the current model to incorporate relative location features.
Model with Relative Location

Make inference on test image with relative location information:

- Make initial predictions by old model

  Generate an relative location features

  Original local appearance features

  Relative Location Maps

Make final predictions using new model incorporating relative location
Generate relative location features

For each pixel $p_i$ in the image and each class $c$

\[ v_{c}^{other}(p_i) = \sum_{j \neq i, c \neq c_j} P(c | p_j)P(c | c_j, \hat{x}_i - \hat{x}_j, \hat{y}_i - \hat{y}_j) \]

\[ v_{c}^{self}(p_i) = \sum_{j \neq i, c = c_j} P(c | p_j)P(c | c_j, \hat{x}_i - \hat{x}_j, \hat{y}_i - \hat{y}_j) \]

Then normalize over all classes for each pixel and take logarithm of the normalized votes
Retrain the model

- Including relative location features

Diagram:
- Original local features
  - Boosted classifiers
  - Boosted responses
  - Logistic Regression
  - Unary potentials
- Relative location features
Retrain the model

- Including relative location features 2

Diagram:
- Original local features
- Boosted classifiers
- Boosted responses
- Logistic Regression
- Unary potentials
- Relative location features
## Experiment results (unary)

<table>
<thead>
<tr>
<th></th>
<th>Overall class accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Unary Model</td>
<td>0.781432</td>
</tr>
<tr>
<td>Model Trained by method 1</td>
<td>0.788123</td>
</tr>
<tr>
<td>Model Trained by method 2</td>
<td>0.784043</td>
</tr>
</tbody>
</table>
Experiment sample results

- Original model:

- Model with Rel. Loc.
Experiment sample results

- Original model:

- Model with Rel. Loc.: