LID Challenge: Weakly Supervised Semantic Segmentation

3d place solution

NoPeopleAllowed: The 3 step approach to weakly supervised semantic segmentation

Mariia Dobko, Ostap Viniavskyi, Oles Dobosevych
UCU & SoftServe team
The Machine Learning Lab at Ukrainian Catholic University, SoftServe
Outline

- Problem description
- Competition
- Approach architecture
  - Step 1. CAM generation via classification
  - Step 2. IRNet for CAM improvements
  - Step 3. Segmentation
- Postprocessing
- Results
- Conclusions
Problem description

A key bottleneck in building a DCNN-based segmentation models is that they typically require pixel level annotated images during training. Acquiring such data demands an expensive, and time-consuming effort.

We develop a method that has a high performance in segmentation task while also saves time and expenses by using only image-level annotations.

Image-level annotations

- 15 times faster to label
- > 25 times cheaper
  - 0.035$ per image for class,
  - 3.45$ for segmentation
LID Challenge Dataset

- **Multilabel multiclass**
- **200 classes + background**
- **456,567 training images**
  - validation: 4,690
  - test: 10,000
- **Pixel-wise labels are provided for validation set only**
- **No pixel-wise annotations** are allowed for training
Challenges

- **High imbalance** in classes: ‘person’, ‘bird’, ‘dog’
- **Missing labels**
- **Subset of 2014** has better labels for ‘person’, than the whole dataset
## Previous works

### Expectation-Maximization methods
- Multiple Instance Learning methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
<th>Code available?</th>
<th>Train/test code</th>
<th>Code framework</th>
<th>VOC2012 mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-FCN (Pathak et al., 2014)</td>
<td>2015</td>
<td>Y</td>
<td>Train/test</td>
<td>MatConvNet</td>
<td>25.7</td>
</tr>
<tr>
<td>CCNN (Pathak et al., 2015)</td>
<td>2015</td>
<td>Y</td>
<td>Train/test</td>
<td>Caffe</td>
<td>35.3</td>
</tr>
<tr>
<td>EM-Adapt (Papandreou et al., 2015)</td>
<td>2015</td>
<td>Y</td>
<td>Train/test</td>
<td>Caffe, TensorFlow</td>
<td>38.2</td>
</tr>
<tr>
<td>DCSM w/o CRF (Shimoda and Yanai, 2016)</td>
<td>2016</td>
<td>Y</td>
<td>Test</td>
<td>Caffe</td>
<td>40.5</td>
</tr>
<tr>
<td>DCSM w/ CRF (Shimoda and Yanai, 2016)</td>
<td>2016</td>
<td>Y</td>
<td>Test</td>
<td>Caffe</td>
<td>44.1</td>
</tr>
<tr>
<td>BFBP (Saleh et al., 2016)</td>
<td>2016</td>
<td>N</td>
<td>No</td>
<td>-</td>
<td>46.6</td>
</tr>
<tr>
<td>SEC (Kolesnikov and Lampert, 2016b)</td>
<td>2016</td>
<td>Y, Caffe, TensorFlow</td>
<td>Train/test</td>
<td>Caffe, TensorFlow</td>
<td>50.7</td>
</tr>
<tr>
<td>WILDCAT + CRF (Durand et al., 2017)</td>
<td>2017</td>
<td>Y</td>
<td>Train/test</td>
<td>PyTorch</td>
<td>43.7</td>
</tr>
<tr>
<td>SPN (Kwak et al., 2017)</td>
<td>2017</td>
<td>Y</td>
<td>Custom layer only</td>
<td>Keras</td>
<td>50.2</td>
</tr>
<tr>
<td>AE-PSL (Wei et al., 2017)</td>
<td>2017</td>
<td>N</td>
<td>No</td>
<td>-</td>
<td><strong>55.0</strong></td>
</tr>
<tr>
<td>PRM (Zhou et al., 2018)</td>
<td>2018</td>
<td>Y</td>
<td>Test</td>
<td>PyTorch</td>
<td>53.4</td>
</tr>
<tr>
<td>DSRG (VGG16) (Huang et al., 2018)</td>
<td>2018</td>
<td>Y, Caffe, TensorFlow</td>
<td>Train/test</td>
<td>Caffe, TensorFlow</td>
<td>59.0</td>
</tr>
<tr>
<td>PSA (DeepLab) (Ahn and Kwak, 2018)</td>
<td>2018</td>
<td>Y</td>
<td>Train/test</td>
<td>PyTorch</td>
<td>58.4</td>
</tr>
<tr>
<td>MDC (Wei et al., 2018)</td>
<td>2018</td>
<td>N</td>
<td>No</td>
<td>-</td>
<td>60.4</td>
</tr>
<tr>
<td>DSRG (ResNet101) (Huang et al., 2018)</td>
<td>2018</td>
<td>Y, Caffe, TensorFlow</td>
<td>Train/test</td>
<td>Caffe, TensorFlow</td>
<td>61.4</td>
</tr>
<tr>
<td>PSA (ResNet38) (Ahn and Kwak, 2018)</td>
<td>2018</td>
<td>Y</td>
<td>Train/test</td>
<td>PyTorch</td>
<td><strong>61.7</strong></td>
</tr>
<tr>
<td>FickleNet (Lee et al., 2019)</td>
<td>2019</td>
<td>N</td>
<td>No</td>
<td>-</td>
<td>61.2</td>
</tr>
<tr>
<td>IRNet (Ahn et al., 2019)</td>
<td>2019</td>
<td>Y</td>
<td>Train/test</td>
<td>PyTorch</td>
<td><strong>63.5</strong></td>
</tr>
</tbody>
</table>

---

Chan et al. A Comprehensive Analysis of Weakly-Supervised Semantic Segmentation in Different Image Domains
Our approach architecture

Step 1
- Classification CNN
- GRADCAM
- Multiscale CAM
- Dense CRF

Step 2
- IRNet

Step 3
- Segmentation
- TTA
Step 1. CAM generation via classification

Input

- 72k - train, 12k validation
- balanced dataset
- no person class

Results

Zhou et al. Learning deep features for discriminative localization
Step 1. CAM generation via classification

Tested approaches

- ResNet50 vs. VGG16 → ResNet produces artifacts
- VGG16 with additional 4 conv layers
- GRADCAM vs. GRADCAM++ → GRADCAM++ usually gives just slightly better results

Chattopadhyay et al. Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks
Step 2. IRNet for CAM improvements

**Input**
- Select most confident maps
- Threshold CAMs into confident BG, confident FG and unconfident regions

**Results**

Ahn et al. *Weakly supervised learning of instance segmentation with inter-pixel relations.*

Figure 2. Overall architecture of IRNet.
IRNet

IRNet’s two branches:
1 - learns the displacement field
2 - learns class boundaries

\[
\mathcal{L} = \mathcal{L}_{fg}^D + \mathcal{L}_{bg}^D + \mathcal{L}_B.
\]

Losses for Displacement fields (foreground & background)

Loss for class boundary detection

Ahn et al. Weakly supervised learning of instance segmentation with inter-pixel relations.
IRNet. Class Boundary Detection

\[ a_{i,j} = 1 - \max_{k \in \mathcal{P}_{i,j}} \mathcal{B}(x_k) \]

\[
\mathcal{L}^B = - \sum_{(i,j) \in \mathcal{P}_{fg}^+} \frac{\log a_{i,j}}{2|\mathcal{P}_{fg}|} - \sum_{(i,j) \in \mathcal{P}_{bg}^+} \frac{\log a_{i,j}}{2|\mathcal{P}_{bg}|} - \sum_{(i,j) \in \mathcal{P}^-} \frac{\log(1 - a_{i,j})}{|\mathcal{P}^-|}
\]

Ahn et al. Weakly supervised learning of instance segmentation with inter-pixel relations.
Step 3 - Segmentation

DeepLab v3+

Input

- 352x352 input images
- Strong augmentations
- ~42k images for training

Results

Chen et al. Encoder-decoder with atrous separable convolution for semantic image segmentation.
Test Time Augmentations (TTAs) are added after the segmentation step. The combination of 2 types of different TTAs, with one having 3 parameters, results in a total of 6 predictions, which are averaged by mean.
Secret insights

- **VGG** is better for CAM generation as **ResNet** gives artifacts
- **Decrease the output stride** of VGG by removing some of the max pooling operations
- **Confident** and **unconfident** regions for IRNet
- **Multiscale CAM** give a large improvement
- **Dense CRF** doesn’t require training, helps to rectify boundaries
- **TTA** after segmentation step drastically improves the results
- Replace stride with dilation in DeepLabv3+ to **decrease the output stride**
Metrics

Classification Quality

- F-1 score

\[ F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]

Segmentation Quality

- Mean IoU

\[ m\text{IoU} = \frac{1}{k} \sum_{i=1}^{k} \frac{TP_{ii}}{\sum_{j=1}^{k} FN_{ij} + \sum_{j=1}^{k} FP_{ij} - TP_{ii}} \]

- Pixel Accuracy
- Mean Accuracy

Step 1. Classification

Step 2-3. IRnet & Segmentation
# Quantitative Results

<table>
<thead>
<tr>
<th>Model</th>
<th>IRNet threshold</th>
<th>TTA</th>
<th>Person CAM</th>
<th>Mean IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepLabv3+ encoder: ResNet50</td>
<td>0.3</td>
<td>No</td>
<td>No</td>
<td>36.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td><strong>39.64</strong></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>No</td>
<td></td>
<td>37.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td>39.58</td>
</tr>
<tr>
<td>DeepLabv3+ encoder: ResNet101</td>
<td>0.5</td>
<td>No</td>
<td></td>
<td>36.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td>37.15</td>
</tr>
</tbody>
</table>

* wasn’t submitted

**Validation set**

Experiments with different architectures and parameters on the 3rd step
## Quantitative Results

### Test set:

- **DeepLabv3+**
- **TTA** (Horizontal Flip, Multi-scaling)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Participant team</th>
<th>Mean IoU</th>
<th>Mean accuracy</th>
<th>Pixel accuracy</th>
<th>Last submission at</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cvl</td>
<td>45.18</td>
<td>59.62</td>
<td>80.46</td>
<td>1 day ago</td>
</tr>
<tr>
<td>2</td>
<td>VL-task1</td>
<td>37.73</td>
<td>60.15</td>
<td>82.98</td>
<td>2 days ago</td>
</tr>
<tr>
<td>3</td>
<td>UCU &amp; SoftServe</td>
<td>37.34</td>
<td>54.87</td>
<td>83.64</td>
<td>2 days ago</td>
</tr>
<tr>
<td>4</td>
<td>IOnlyHaveSevenDays</td>
<td>36.24</td>
<td>68.27</td>
<td>84.10</td>
<td>2 days ago</td>
</tr>
<tr>
<td>5</td>
<td>play-njupt</td>
<td>31.90</td>
<td>46.07</td>
<td>82.63</td>
<td>1 month ago</td>
</tr>
<tr>
<td>6</td>
<td>xingxiao</td>
<td>29.48</td>
<td>48.66</td>
<td>80.82</td>
<td>1 month ago</td>
</tr>
<tr>
<td>7</td>
<td>hagenbreaker</td>
<td>22.50</td>
<td>39.92</td>
<td>77.38</td>
<td>19 days ago</td>
</tr>
<tr>
<td>8</td>
<td>go-g0</td>
<td>19.80</td>
<td>38.30</td>
<td>76.21</td>
<td>20 days ago</td>
</tr>
<tr>
<td>9</td>
<td>lasthours-try</td>
<td>12.56</td>
<td>24.65</td>
<td>64.35</td>
<td>1 day ago</td>
</tr>
<tr>
<td>10</td>
<td>WH-ljs</td>
<td>7.79</td>
<td>16.59</td>
<td>62.52</td>
<td>2 days ago</td>
</tr>
</tbody>
</table>
Open questions

Different types of **regularization** added to the first step → Improve the **classification**

**Downsampling** was used to balance data → **Upsampling** or **combination** of both should be tested

Adding person class labels to the other steps of pipeline →
   Ability to provide better results for a class which is highly present in data, though severely mislabeled

Mean IoU per class allows to obtain high score even when some classes are skipped →
   **A different metric** or combination of metrics should be chosen as a premier for this task
Thank you for attention!

dobko_m@ucu.edu.ua
viniavskyi@ucu.edu.ua
dobosevych@ucu.edu.ua

presentation