Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-supervised Object and Action Localization

Krishna Kumar Singh and Yong Jae Lee
University of California, Davis
Introduction
Why not use the fully-supervised approach?

Requires expensive, error-prone bounding box annotations. Thus, it’s not scalable
Weakly-supervised approach

- Visual classification and localization tasks
- Visual attribute localization
- Requires less detailed annotations compared to the fully-supervised approach
Weakly-supervised approach

- Supervision is provided at the image-level. It is scalable.

- Most weakly-supervised object localization approaches mine discriminative features or patches in the data that frequently appear in one class and rarely in other classes.
Prior attempts to improve weak object localization

Select multiple discriminative regions

[Song et al. NIPS 2014]
Prior attempts to improve weak object localization

Requires additional labeled videos

[Singh et al. CVPR 2016]

Transfer tracked objects from videos to images
Prior attempts to improve weak object localization

Global average pooling to encourage network to look at all relevant parts.

[Zhou et al. CVPR 2016]
Hide-and-Seek

- If we randomly remove some patches from the image, the model must seek other relevant parts.

- Hide-and-Seek only alters the input image.
Hide-and-Seek

Full image

Randomly hidden patches
Related Work

- Weakly-supervised object localization
- Masking pixels or activations
- Action localization
Related Work

- **Weakly-supervised object localization**

  Other network architectures have been designed for weakly-supervised object detection, still rely on a classification objective and thus fail to capture the full extent of an object.

- **Masking pixels or activations**

- **Action localization**
Related Work

- Weakly-supervised object localization

- **Masking pixels or activations**
  - In the paper, image regions are masked during training.

- Action localization
Related Work

- Weakly-supervised object localization
- Masking pixels or activations
- Action localization
Related Work

- Weakly-supervised object localization

- Masking pixels or activations

- **Action localization**
  - Fully-supervised methods/Weak-supervised methods/approach in the paper
Approach
Hide-and-Seek (HaS) for:

- Weakly-supervised object localization in images
- Weakly-supervised temporal action localization in videos
Divide the training image into a grid with a patch size of $S \times S$

Training image with label “dog”
Divide the training image into a grid with a patch size of $S \times S$.

Training image with label “dog”
Randomly hide patches

Training image with label “dog”

Epoch 1
Randomly hide patches

Training image with label “dog”

Epoch 1

Epoch 2
Randomly hide patches

Training image with label “dog”
Feed each hidden image to image classification CNNs

Training image with label “dog”
During testing feed full images into trained network

Test image

Trained CNN

Class Activation Map (CAM)
Predicted label: “dog”
Generating a Class Activation Map (CAM)

Setting the hidden pixel values

Set the RGB value of a hidden pixel to be the mean RGB vector of the entire dataset:

\[ \mathbf{v} = \mu = \frac{1}{N_{\text{pixels}}} \sum_j x_j \]

where \( j \) indexes all pixels in the entire training dataset and \( N_{\text{pixel}} \) is the total number of pixels in the dataset.
Hide-and-Seek (HaS) for:

- Weakly-supervised object localization in images
- Weakly-supervised temporal action localization in videos
Divide training video into contiguous frame segments of size $S$
Divide training video into contiguous frame segments of size $S$
Feed each hidden video to action classification CNN
Feed each hidden video to action classification CNN

Training video “high-jump”
Training video “high-jump”

During testing feed full video into trained network
Other applications of HaS

- Weakly-supervised semantic segmentation
- Image classification
- Emotion recognition and age/gender estimation
- Person re-identification

Experiments and Results
Dataset

For object localization in images -

- ILSVRC 2016
- 1000 classes
- 1.2 million images with class labels for training
Dataset

For action localization in video -

- THUMOS 2014 validation data
- 1010 untrimmed videos, 101 classes
- Train over all classes
- Evaluate 20 classes with temporal annotations
- Each video can contain multiple instances of a class
Metrics

For Object localization -

1) **Top-1 Loc:**
   Predicted class correct and bounding box > 50% IoU with ground truth

2) **GT-known Loc:**
   Bounding box > 50% IoU with ground truth of known class

3) **Top-1 Clas:**
   Classification accuracy
Metrics

For action localization -

1) Mean average precision (mAP) for evaluation
2) Prediction is correct if IoU > $\Theta$
3) $\Theta = \{ 0.1, 0.2, 0.3, 0.4, 0.5 \}$
4) Assume ground truth class label is known
Qualitative results of object localization
Qualitative results of action localization

Slide courtesy of Krishna Kumar Singh, UC Davis
Qualitative results of action localization

Slide courtesy of Krishna Kumar Singh, UC Davis
### Results of object localization

<table>
<thead>
<tr>
<th>Methods</th>
<th>GT-known Loc</th>
<th>Top-1 Loc</th>
<th>Top-1 Clas</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet-GAP [61]</td>
<td>54.90²</td>
<td>36.25</td>
<td>60.23</td>
</tr>
<tr>
<td>AlexNet-HaS-16</td>
<td>57.86</td>
<td>36.77</td>
<td>57.97</td>
</tr>
<tr>
<td>AlexNet-HaS-32</td>
<td><strong>58.75</strong></td>
<td>37.33</td>
<td>57.94</td>
</tr>
<tr>
<td>AlexNet-HaS-44</td>
<td>58.55</td>
<td>37.54</td>
<td>58.10</td>
</tr>
<tr>
<td>AlexNet-HaS-56</td>
<td>58.43</td>
<td>37.34</td>
<td>58.13</td>
</tr>
<tr>
<td>AlexNet-HaS-Mixed</td>
<td>58.68</td>
<td><strong>37.65</strong></td>
<td>58.68</td>
</tr>
<tr>
<td>GoogLeNet-GAP [61]</td>
<td>58.41²</td>
<td>43.60</td>
<td><strong>71.95</strong></td>
</tr>
<tr>
<td>GoogLeNet-HaS-16</td>
<td>59.83</td>
<td>44.62</td>
<td>70.49</td>
</tr>
<tr>
<td>GoogLeNet-HaS-32</td>
<td><strong>60.29</strong></td>
<td><strong>45.21</strong></td>
<td>70.70</td>
</tr>
<tr>
<td>GoogLeNet-HaS-44</td>
<td>60.11</td>
<td>44.75</td>
<td>70.34</td>
</tr>
<tr>
<td>GoogLeNet-HaS-56</td>
<td>59.93</td>
<td>44.78</td>
<td>70.37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Takeaway:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomly selecting the hidden patch size gives the best result.</td>
</tr>
</tbody>
</table>

Table 1. Localization accuracy on ILSVRC validation data with different patch sizes for hiding. Our Hide-and-Seek always performs better than AlexNet-GAP [61], which sees the full image.
Results of object localization

<table>
<thead>
<tr>
<th>Methods</th>
<th>GT-known Loc</th>
<th>Top-1 Loc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backprop on AlexNet [38]</td>
<td>-</td>
<td>34.83</td>
</tr>
<tr>
<td>AlexNet-GAP [61]</td>
<td>54.90</td>
<td>36.25</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>58.68</strong></td>
<td><strong>37.65</strong></td>
</tr>
<tr>
<td>AlexNet-GAP-ensemble</td>
<td>56.91</td>
<td>38.58</td>
</tr>
<tr>
<td>Ours-ensemble</td>
<td><strong>60.14</strong></td>
<td><strong>40.40</strong></td>
</tr>
<tr>
<td>Backprop on GoogLeNet [38]</td>
<td>-</td>
<td>38.69</td>
</tr>
<tr>
<td>GoogLeNet-GAP [61]</td>
<td>58.41</td>
<td>43.60</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>60.29</strong></td>
<td><strong>45.21</strong></td>
</tr>
</tbody>
</table>

Table 2. Localization accuracy on ILSVRC val data compared to state-of-the-art. Our method outperforms all previous methods.

Takeaway:

Averaging the CAM and class probabilities gives the best performance.
Results of object localization - Comparative Study

Table 3. Our approach outperforms Dropout [44] for localization.

<table>
<thead>
<tr>
<th>Methods</th>
<th>GT-known Loc</th>
<th>Top-1 Loc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>58.68</td>
<td>37.65</td>
</tr>
<tr>
<td>AlexNet-dropout-trainonly</td>
<td>42.17</td>
<td>7.65</td>
</tr>
<tr>
<td>AlexNet-dropout-traintest</td>
<td>53.48</td>
<td>31.68</td>
</tr>
</tbody>
</table>

Takeaway: HaS method much better at improving localization performance than dropout.
Results of object localization - Comparative Study

<table>
<thead>
<tr>
<th>Methods</th>
<th>GT-known Loc</th>
<th>Top-1 Loc</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet-GAP</td>
<td>54.90</td>
<td>36.25</td>
</tr>
<tr>
<td>AlexNet-Avg-HaS</td>
<td>58.43</td>
<td>37.34</td>
</tr>
<tr>
<td>AlexNet-GMP</td>
<td>50.40</td>
<td>32.52</td>
</tr>
<tr>
<td>AlexNet-Max-HaS</td>
<td><strong>59.27</strong></td>
<td><strong>37.57</strong></td>
</tr>
</tbody>
</table>

Table 4. Global average pooling (GAP) vs. global max pooling (GMP). Unlike [61], for Hide-and-Seek GMP still performs well for localization. For this experiment, we use patch size 56.

Takeaway: GMP performs better as HaS already trains network for better localization.
Results of object localization - Comparative Study

<table>
<thead>
<tr>
<th>Methods</th>
<th>GT-known Loc</th>
<th>Top-1 Loc</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet-GAP</td>
<td>54.90</td>
<td>36.25</td>
</tr>
<tr>
<td>AlexNet-HaS-conv1-5</td>
<td>57.36</td>
<td>36.91</td>
</tr>
<tr>
<td>AlexNet-HaS-conv1-11</td>
<td><strong>58.33</strong></td>
<td><strong>37.38</strong></td>
</tr>
</tbody>
</table>

Table 5. Applying Hide-and-Seek to the first conv layer. The improvement over [61] shows the generality of the idea.

Takeaway: HaS method intuition can be applied to even convolution layer filter outputs to give the same boost in performance.
Results of object localization - Comparative Study

<table>
<thead>
<tr>
<th>Methods</th>
<th>GT-known Loc</th>
<th>Top-1 Loc</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet-HaS-25%</td>
<td>57.49</td>
<td>37.77</td>
</tr>
<tr>
<td>AlexNet-HaS-33%</td>
<td>58.12</td>
<td>38.05</td>
</tr>
<tr>
<td>AlexNet-HaS-50%</td>
<td>58.43</td>
<td>37.34</td>
</tr>
<tr>
<td>AlexNet-HaS-66%</td>
<td>58.52</td>
<td>35.72</td>
</tr>
<tr>
<td>AlexNet-HaS-75%</td>
<td>58.28</td>
<td>34.21</td>
</tr>
</tbody>
</table>

Table 6. Varying the hiding probability. Higher probabilities lead to decrease in Top-1 Loc whereas lower probability leads to smaller GT-known Loc. For this experiment, we use patch size 56.

Takeaway: There is a trade off between localization and classification accuracy wrt hiding probability.
Results of action localization

<table>
<thead>
<tr>
<th>Methods</th>
<th>IOU thresh = 0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video-full</td>
<td>34.23</td>
<td>25.68</td>
<td>17.72</td>
<td>11.00</td>
<td>6.11</td>
</tr>
<tr>
<td>Video-HaS</td>
<td><strong>36.44</strong></td>
<td><strong>27.84</strong></td>
<td><strong>19.49</strong></td>
<td><strong>12.66</strong></td>
<td><strong>6.84</strong></td>
</tr>
</tbody>
</table>

Table 7. Action localization accuracy on THUMOS validation data. Across all 5 IoU thresholds, our Video-HaS outperforms the full video baseline (Video-full).
Classification results for higher capacity network

<table>
<thead>
<tr>
<th></th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>94.19</td>
<td>94.66</td>
<td>94.87</td>
</tr>
<tr>
<td>HaS</td>
<td>94.97</td>
<td>95.41</td>
<td>95.53</td>
</tr>
</tbody>
</table>

“Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond” (Singh, 2018).
Using Hide-and-Seek as data augmentation improves performance of various vision tasks

Image classification +1.1%  
[He et al. 2015]

Semantic segmentation +1.3%  
[Long et al. 2015]

Face recognition tasks +1%  
(emotion, age, gender) [Khorrami et al. 2015]

Person Reidentification +1.6%  
[Zhong et al. 2018]
Our approach improves image classification when objects are partially-visible
Fail Cases

AlexNet-GAP
[Zhou et al. CVPR 2016]

Ours

Merging spatially-close instances together

Localizing co-occurring context

Slide courtesy of Krishna Kumar Singh, UC Davis
Fail Cases

Our approach can fail by localizing co-occurring context

Slide courtesy of Krishna Kumar Singh, UC Davis
Strength and weakness

Strength -

1) The Hide and Seek method can be applied to any architecture
2) Better than Dropout for localization problem
3) Can be used as form of data augmentation to improve other tasks like segmentation, face recognition, etc.

Weakness -

1) The classification accuracy decreases for lower capacity networks
2) Spatially close instances and co-occurring contexts cause method to fail
3) Current method will not suffice for videos with multiple action labels
Future work

1) The patch size and hiding probabilities are hyper-parameters.
2) Dynamically learn patch size and hiding probability during training.
Thank You
Any Questions?
## Implementation - Object localization

<table>
<thead>
<tr>
<th>Models</th>
<th>Conv layer</th>
<th>Train epochs</th>
<th>Learning rate</th>
<th>Batch Norm</th>
<th>Cam Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet-GAP</td>
<td>512, (3x3), stride = 1, pad = 1</td>
<td>55</td>
<td>0.01 → 0.0001</td>
<td>Yes</td>
<td>20%</td>
</tr>
<tr>
<td>GoogLeNet-GAP</td>
<td>1024, (3x3), stride = 1, pad = 1</td>
<td>40</td>
<td>0.01 → 0.0001</td>
<td>Yes</td>
<td>30%</td>
</tr>
</tbody>
</table>
Implementation - Action localization

**Sample 2000 features @ 10 feats/sec**
- Model trained on Sports 1 Million
- Compute C3D fc7 Features

**Segment video**
- \( F_{\text{segment}} = 100, \ P_{\text{hide}} = 0.5 \)
- 20 equal length segments

**Feed C3D features to CNN**
- 500 kernels, (1x1), stride = 1
- Mean C3D for hidden frames

**Prediction**
- Threshold = 50% of max CAM
- All continuous segments after thresholding