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

Yixian Wang, Ethan Hou
Outline

- Background & Current Problems
- Paper’s Solution Intro.
- Approach & Implementation Details
- Experiment
Background

- **Weakly-supervised**
  - Def.: Limited amount of labeled data.
  - Advantages:
    - less detailed annotations
    - Compatibility with vast weakly-annotated visual data
  - Categories:
    - Incomplete Supervision
    - Inexact Supervision
    - Inaccurate Supervision
      Mixly Used
Background

Current Problem

- Localize only the most discriminative parts
Main Idea

- Hide patches in a training image randomly
- Force the network to seek other relevant parts
Main Idea

Advantage

- Modify only the input image
- Ability to work with any network designed for object localization.
- Ability to be easily generalized to different neural networks and tasks
Approach

● Object localizer
  ○ to predict both the category label & bounding box

● Hide Random Image Patches
  ○ Divide the training images into grid of fixed patches.
  ○ Hide patches randomly (0.5 Probability),
  ○ Force the network to focus on other relevant parts of the object.

* hide different patches for the same training images
Approach

- Hide Random Image Patches

Fig. Hide Different patches for the same image
Approach ---- Deal with Hidden Pixel Value

- Setting the hidden pixel values
  - Reason:
    - Activation distributions are different between training sets and test sets, due to the hidden patches in training images.

- Solution
  - Setting the value of hidden patches to mean value of the image over the entire dataset.
Approach

- Setting the hidden pixel values
  - Completely visible Filter
    \[ \sum_{i=1}^{k \times k} w_i^T x_i \]
  - Completely hidden Filter
    \[ \sum_{i=1}^{k \times k} w_i^T v \]
  - Partially hidden Filter
    \[ \sum_{m \in \text{visible}} w_m^T x_m + \sum_{n \in \text{hidden}} w_n^T v \]

Fig. Three types of activation
Approach

- Setting the hidden pixel values
  - Set the hidden value to the mean of all pixels

\[
\nu = \mu = \frac{1}{N_{\text{pixels}}} \sum_{j} x_i
\]

- Completely visible Filter

\[
\sum_{i=1}^{k \times k} w_i^T x_i
\]

- Completely hidden Filter

\[
\sum_{i=1}^{k \times k} w_i^T \nu \rightarrow \sum_{i=1}^{k \times k} w_i^T \mu
\]

- Partially hidden Filter

\[
\sum_{m \in \text{visible}} w_m^T x_m + \sum_{n \in \text{hidden}} w_n^T \mu
\]
What is Class Activation Map (CAM)

- Indicate and visualize the discriminative image regions used by the CNN to identify that category (heat map)

- Enables classification-trained CNNs to perform object localization without using bounding box annotations.
Class Activation Map (CAM) Example

- **Type of environment:**
  - outdoor

- **Scene categories:**
  - rock_arch (0.931)

- **Scene attributes:**
  - natural light, open area, boating, natural, ocean, swimming, rugged scene, no horizon, rock

- Informative region for predicting the category *rock_arch* is:
How to Generate CAM

Network Architecture

- CNN trained of classification
- GAP layer instead of fully-connected layer before the final output, then softmax layer.

How to Generate CAM

Class Activation Mapping

\[
CAM(c, I) = \sum_{i=1}^{M} W(c, i) \cdot F_i(I)
\]

\[F = \{F_1, F_2, \ldots, F_m\} \quad ---- \quad M \text{ feature maps}
\]

\[W \quad ---- \quad N \times M \text{ weight matrix}
\]

How to generate bounding box from CAM

- Thresholding -> segment the heatmap
- Regions of which the value is above 20% of the max value
- Bounding box covers the largest connected component.

**Goal**
- Predict the label of an action & its start and end time

**Approach**
- Hiding frames in video to improve action localization,

**Implementation**
- Sample video frames uniformly from video
- Divide samples into segments
- Hide segments randomly and feed to deep action localizer network
- Apply thresholding on generated CAM
Experiment ---- Object Localization

- **Architecture:**
  - Modified AlexNet and GoogLeNet
    - remove the fully-connected layers before the final output and replace them with GAP followed by a fully-connected softmax layer[*]
- **Dataset:**
  - ILSVRC 2016
- **Metrics:**
  - Top-1 Loc -> classification & localization
  - GT-known Loc -> localization
  - Top-1 Class -> classification

Experiment ---- Action Localization

- **Architecture:**
  - C3D + CNN
    - C3D generate feature maps that fed into a CNN with 2 conv layer followed by GAP and softmax

- **Dataset:**
  - THUMOS 2014

- **Metrics:**
  - mAP
C3D

3D convolutional operation

C3D network architecture

Object localization quantitative results

Patch Size

- N = [16, 32, 44, 56]
- Difference between sizes is trivial
- All better than baseline in localization
- Lower classification accuracy
Object localization quantitative results

Patch Size

- N = [16, 32, 44, 56]
- Lower classification accuracy
- **Mix Size**: randomly choose from 16, 32, 44, 56 and no hidden
- Classification accuracy improved (learn complementary info)
Object localization quantitative results

- Ensemble Model

Average CAM from different patch sizes
Object localization qualitative results

<table>
<thead>
<tr>
<th>Ground-truth:</th>
<th>Localization Result:</th>
</tr>
</thead>
</table>

- **Bounding Box (AlexNet-GAP)**
- **Heatmap (AlexNet-GAP)**
- **Bounding Box (Our HaS)**
- **Heatmap (Our HaS)**

![Object localization results](image-url)
Further Analysis

Comparison with **Dropout**

**Difference:**
- **Dropout:**
  - prevent overfitting
  - Drop pixels randomly
- **HaS:**
  - improve localization
  - Drop patches contiguously

**Results:**
- dropout produces inferior performance
- still show most discriminative part
  - Random drop
  - High probability see relevant parts
Further Analysis

GAP VS GMP

- GMP is inferior to GAP in baseline
- HaS GMP increase a lot and slightly outperform than Has GAP
- Reason:
  - GMP is forced to learn the less discriminative parts
  - GMP is more robust to noise.
Further Analysis

HaS in convolutional layers.

- Divide conv1 feature maps into a grid and hide patches of size 5 and 11.
- Proves that HaS can be generalized to convolutional layers
Further Analysis

Does Hiding Probability Make Difference?

- 50% in previous experiment
- Showing more pixels make classification performance better
- But decrease localization ability.

![Graph showing accuracy vs hiding probability](image)
## Further Analysis

### Action Localization Result

<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>36.44</td>
<td>27.84</td>
<td>19.49</td>
<td>12.66</td>
<td>6.84</td>
</tr>
</tbody>
</table>
Strength & Weakness

- **Strength**:  
  - Simple method, good result

- **Weakness**:  
  - Brief introduction to action localization part
Question?