Webly Supervised Learning of Convolutional Networks

Xinlei Chen, Abhinav Gupta
ICCV, 2015

LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, Jianxiong Xiao
arXiv, 2015

Presenter: Igor Janjic
11/12/2015
Machine Learning

Supervised $\iff$ Semi-Supervised $\iff$ Unsupervised
(Weakly Supervised)

Image source: Prof. Kai Arras
(Social Robotics Lab)
Why is unsupervised learning important?

Answer: For learning general representations
1. Most data is unlabeled or weakly labeled
2. The amount of supervision in a learning task depends on both the information content and noise level in the labels.
3. Unsupervised learning is used by the brain
Webly Supervised Learning of Convolutional Networks

Xinlei Chen, Abhinav Gupta
ICCV, 2015
The Problem

- Investigation of webly-supervised learning of CNNs
- Can CNNs be trained for easy categories using images retrieved by search engines and from the web at large?
- How well do webly-supervised CNNs generalize to vision tasks?
- Main contributions:
  - Trained a webly-supervised CNN on PASCAL VOC 2012 which outperforms ImageNet
  - Webly-supervised learning works for object localization and for training R-CNN style detectors
  - State-of-the-art performance on Pascal VOC data without training on VOC dataset
  - Competitive performance on scene classification
Motivation

- Supervised learning using CNNs has achieved state-of-the-art success in a variety of vision tasks
- How to improve performance?
  - Deeper networks are better, especially with more data
  - But human labeled data is expensive and inefficient
  - Web data is biased and noisy but is nearly infinite in scale (and continuously growing)
  - Exploit web data to train CNNs
  - Okay, but how?
Google vs Flickr

Image source: google.com

Image source: flickr.com
Bootstrapping

- The goal is to learn a model on Flickr-like images, but these images often have very noisy tags
- Bootstrap CNN training with easy, noise-free examples first and then follow with a more comprehensive learning procedure

Image source: Sinno Jialin Pan, Qiang Yang, "A Survey on Transfer Learning"
Approach

- Could naively train a CNN on random image/tag pairs from the web
- Instead, first train the CNN model from scratch using easy images downloaded from Google search queries
- Then finetune this representation using harder Flickr images under specific constraints determined by a relationship graph
- Use the confusion matrix from the initial training done on easy images as the relationships between labels
Confusion Matrix

- Suppose a classifier was trained to distinguish between cats, dogs, and rabbits
- Dataset contains 27 images of 8 cats, 6 dogs, and 13 rabbits
- Diagonals of confusion matrix are all of the correct guesses

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cat</td>
<td>Dog</td>
</tr>
<tr>
<td>Cat</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Dog</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Rabbit</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Example source: Andrea Vedaldi

Example source: wikipedia
Approach

- Three lists of categories are constructed from the ImageNet Challenge, SUN database, and NEIL knowledge base.
- These category names are used as Google search queries to construct a dataset of ~600 images/query.
- BVLC reference network is trained on this dataset in a supervised manner using Caffe.
- After convergence, the network has learned good low-level filters but is biased towards simple images.
- Now construct a dataset of more realistic Flickr images found using the same search queries.
Relationship Graph

- Object categories have various complex relationships such as hierarchies, dependencies, similarities, restrictions, etc., all of which together form an ontology representable as a graph.
- Approach taken in the paper is to simply use the confusion matrix as the relationships.
- For any pair of concepts \( i \) and \( j \), the relationship \( R_{ij} \) is defined as:
  \[
  R_{ij} = P(i | j) = \frac{\sum_{k \in C_i} \text{CNN}(j | I_k)}{|C_i|},
  \]
  where \( C_i \) is the index set for images that belong to concept \( i \), and given pixel values \( I_k \), \( \text{CNN}(j | I_k) \) is the network's belief on how likely image \( k \) belongs to concept \( j \).
- Choose only the top \( K = 5 \).
Relationship Graph

- Relationship graph is a way to characterize the label-flip noise
- For a class label $l_k$, softmax loss is

\[ L = \sum_k \sum_i R_{il_k} \log(CNN(i|I_k)), \]

- Relationship matrix $R$ is kept fixed after being learned using the initial network
Object Localization

• Need to clean web data and localize objects to train a R-CNN detector
• But CNN only distinguishes small set of classes and is spatially invariant
• Google images have centered-biased images so they are used as bounding box seeds
• Nearest Neighbor propagation to find neighbors and EdgeBox to find candidate windows
• Agglomerative clustering merges NN sets bottom up to form subcategories and R-CNN detector is trained on each category using all clustered bounding boxes
• Random patches from YFCC are used as negatives
• Positive bounding boxes are increased using EdgeBox and by using the relationship graph to expand the category
• Final SVM is trained
Implementation Details

• Networks are trained in Caffe
  ➢ Batch size is 256
  ➢ Learning rate starts at 0.01 and reduced by a factor of 10 every 150k iterations
  ➢ Training stops after 450k iterations
• 2,240 objects, 89 attributes, 874 scenes
• GoogleO (object-attribute network):
  ➢ ~1.5 million images from Google image search
  ➢ Later fine-tuned with ~1.2 million Flickr images with relationship graph regularization (FlickrG) and without (FlickrF) for 100k iterations and step size of 30k
  ➢ Baselines: CNN learned using Flickr images alone (FlickrS) and combined Google and Flickr images (GFAll)
• GoogleA (all-included network):
  ➢ ~2.1 million images from Google image search (add scene images)
Visualizing Confusion Matrix

- Diagonal of confusion matrix is ranked in descending order and 3 random categories are sampled from top, bottom, and middle of list
# Object Detection (PASCAL VOC 2007)

<table>
<thead>
<tr>
<th></th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>pers</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VOC 2007 test</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageNet [20]</td>
<td>57.6</td>
<td>57.9</td>
<td>38.5</td>
<td>31.8</td>
<td>23.7</td>
<td>51.2</td>
<td>58.9</td>
<td>51.4</td>
<td>20.0</td>
<td>50.5</td>
<td>40.9</td>
<td>46.0</td>
<td>51.6</td>
<td>55.9</td>
<td>43.3</td>
<td>23.3</td>
<td>48.1</td>
<td>35.3</td>
<td>51.0</td>
<td>57.4</td>
<td>44.7</td>
</tr>
<tr>
<td><strong>w/o VOC FT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoogleO [Obj.]</td>
<td>57.1</td>
<td>59.9</td>
<td>35.4</td>
<td>30.5</td>
<td>21.9</td>
<td>53.9</td>
<td>59.5</td>
<td>40.7</td>
<td>18.6</td>
<td>43.3</td>
<td>37.5</td>
<td>41.9</td>
<td>49.6</td>
<td>57.7</td>
<td>38.4</td>
<td>22.8</td>
<td>45.2</td>
<td>37.1</td>
<td>48.0</td>
<td>54.5</td>
<td>42.7</td>
</tr>
<tr>
<td>GoogleA [Obj. + Scc.]</td>
<td>54.9</td>
<td>58.2</td>
<td>35.7</td>
<td>30.7</td>
<td>22.0</td>
<td>54.5</td>
<td>59.9</td>
<td>44.7</td>
<td>19.9</td>
<td>41.0</td>
<td>34.5</td>
<td>40.1</td>
<td>46.8</td>
<td>56.2</td>
<td>40.0</td>
<td>22.2</td>
<td>45.8</td>
<td>36.3</td>
<td>47.5</td>
<td>54.2</td>
<td>42.3</td>
</tr>
<tr>
<td>FlickrS [Flickr Obj.]</td>
<td>50.0</td>
<td>55.9</td>
<td>29.6</td>
<td>26.8</td>
<td>18.7</td>
<td>47.6</td>
<td>56.3</td>
<td>34.4</td>
<td>14.5</td>
<td>35.9</td>
<td>33.3</td>
<td>34.2</td>
<td>43.2</td>
<td>52.2</td>
<td>36.7</td>
<td>21.5</td>
<td>43.3</td>
<td>31.6</td>
<td>48.5</td>
<td>48.4</td>
<td>38.1</td>
</tr>
<tr>
<td>GFAIl [All Obj., 1-stage]</td>
<td>52.1</td>
<td>57.8</td>
<td>38.1</td>
<td>25.6</td>
<td>21.2</td>
<td>47.6</td>
<td>56.4</td>
<td>43.8</td>
<td>19.6</td>
<td>42.6</td>
<td>30.3</td>
<td>37.6</td>
<td>45.1</td>
<td>50.8</td>
<td>39.3</td>
<td>22.9</td>
<td>43.5</td>
<td>34.2</td>
<td>48.3</td>
<td>52.2</td>
<td>40.5</td>
</tr>
<tr>
<td>FlickrF [2-stage]</td>
<td>53.9</td>
<td>60.7</td>
<td>37.0</td>
<td>31.6</td>
<td>23.8</td>
<td>57.7</td>
<td>60.8</td>
<td>44.1</td>
<td>20.3</td>
<td>46.5</td>
<td>31.5</td>
<td>39.8</td>
<td>49.7</td>
<td>59.0</td>
<td>41.6</td>
<td>23.0</td>
<td>44.4</td>
<td>36.2</td>
<td>49.9</td>
<td>56.2</td>
<td>43.4</td>
</tr>
<tr>
<td>FlickrG [2-stage, Graph]</td>
<td>55.3</td>
<td><strong>61.9</strong></td>
<td><strong>39.1</strong></td>
<td>29.5</td>
<td><strong>24.8</strong></td>
<td>55.1</td>
<td><strong>62.7</strong></td>
<td>43.5</td>
<td><strong>22.7</strong></td>
<td>49.3</td>
<td>36.6</td>
<td>42.7</td>
<td>48.9</td>
<td><strong>59.7</strong></td>
<td>41.2</td>
<td>25.4</td>
<td>47.7</td>
<td><strong>41.9</strong></td>
<td>48.8</td>
<td>56.8</td>
<td><strong>44.7</strong></td>
</tr>
<tr>
<td><strong>VOC-Scratch [2]</strong></td>
<td>49.9</td>
<td>60.6</td>
<td>24.7</td>
<td>23.7</td>
<td>20.3</td>
<td>52.5</td>
<td>64.8</td>
<td>32.9</td>
<td>20.4</td>
<td>43.5</td>
<td>34.2</td>
<td>29.9</td>
<td>49.0</td>
<td>60.4</td>
<td>47.5</td>
<td>28.0</td>
<td>42.3</td>
<td>28.6</td>
<td>51.2</td>
<td>50.0</td>
<td>40.7</td>
</tr>
<tr>
<td>ImageNet [20]</td>
<td>64.2</td>
<td><strong>69.7</strong></td>
<td><strong>50.0</strong></td>
<td><strong>41.9</strong></td>
<td><strong>32.0</strong></td>
<td>62.6</td>
<td>71.0</td>
<td><strong>60.7</strong></td>
<td><strong>32.7</strong></td>
<td>58.5</td>
<td>46.5</td>
<td><strong>56.1</strong></td>
<td>60.6</td>
<td>68.8</td>
<td>54.2</td>
<td>31.5</td>
<td>52.8</td>
<td>48.9</td>
<td>57.9</td>
<td>64.7</td>
<td><strong>54.2</strong></td>
</tr>
<tr>
<td><strong>w/VOC FT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GoogleO</td>
<td><strong>65.0</strong></td>
<td>68.1</td>
<td>45.2</td>
<td>37.0</td>
<td>29.6</td>
<td>65.4</td>
<td>73.8</td>
<td>54.0</td>
<td>30.4</td>
<td>57.8</td>
<td>48.7</td>
<td>51.9</td>
<td>64.1</td>
<td>64.7</td>
<td>54.0</td>
<td><strong>32.0</strong></td>
<td><strong>54.9</strong></td>
<td>44.5</td>
<td>57.9</td>
<td>64.0</td>
<td>53.1</td>
</tr>
<tr>
<td>GoogleA</td>
<td>64.2</td>
<td>68.3</td>
<td>42.7</td>
<td>38.7</td>
<td>26.5</td>
<td>65.1</td>
<td>72.4</td>
<td>50.7</td>
<td>28.5</td>
<td><strong>60.9</strong></td>
<td><strong>48.8</strong></td>
<td>51.2</td>
<td>60.2</td>
<td>65.5</td>
<td><strong>54.5</strong></td>
<td>31.1</td>
<td>50.5</td>
<td><strong>48.5</strong></td>
<td>56.3</td>
<td>60.3</td>
<td>52.3</td>
</tr>
<tr>
<td>FlickrG</td>
<td>63.7</td>
<td>68.5</td>
<td>46.2</td>
<td>36.4</td>
<td>30.2</td>
<td><strong>68.4</strong></td>
<td><strong>73.9</strong></td>
<td>56.9</td>
<td>31.4</td>
<td>59.1</td>
<td>46.7</td>
<td>52.4</td>
<td>61.5</td>
<td><strong>69.2</strong></td>
<td>53.6</td>
<td>31.6</td>
<td>53.8</td>
<td>44.5</td>
<td><strong>58.1</strong></td>
<td>59.6</td>
<td>53.3</td>
</tr>
<tr>
<td>VOC 2012 test</td>
<td>aero</td>
<td>bike</td>
<td>bird</td>
<td>boat</td>
<td>bottle</td>
<td>bus</td>
<td>car</td>
<td>cat</td>
<td>chair</td>
<td>cow</td>
<td>table</td>
<td>dog</td>
<td>horse</td>
<td>mbike</td>
<td>person</td>
<td>plant</td>
<td>sheep</td>
<td>sofa</td>
<td>train</td>
<td>tv</td>
<td>mAP</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-------</td>
<td>-----</td>
<td>-------</td>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>ImageNet</td>
<td>68.1</td>
<td>63.8</td>
<td>46.1</td>
<td>29.4</td>
<td>27.9</td>
<td>56.6</td>
<td>57.0</td>
<td>65.9</td>
<td>26.5</td>
<td>48.7</td>
<td>39.5</td>
<td>66.2</td>
<td>57.3</td>
<td>65.4</td>
<td>53.2</td>
<td>26.2</td>
<td>54.5</td>
<td>38.1</td>
<td>50.6</td>
<td>51.6</td>
<td>49.6</td>
</tr>
<tr>
<td>ImageNet-TV</td>
<td>73.3</td>
<td>67.1</td>
<td>46.3</td>
<td>31.7</td>
<td>30.6</td>
<td>59.4</td>
<td>61.0</td>
<td>67.9</td>
<td>27.3</td>
<td>53.1</td>
<td>39.1</td>
<td>60.5</td>
<td>70.9</td>
<td>57.2</td>
<td>26.1</td>
<td>59.0</td>
<td>40.1</td>
<td>56.2</td>
<td>54.9</td>
<td>52.3</td>
<td>52.4</td>
</tr>
<tr>
<td>GoogleO</td>
<td>72.2</td>
<td>67.3</td>
<td>46.0</td>
<td>32.3</td>
<td>31.6</td>
<td>62.6</td>
<td>62.5</td>
<td>66.5</td>
<td>27.3</td>
<td>52.1</td>
<td>38.9</td>
<td>64.0</td>
<td>59.1</td>
<td>71.6</td>
<td>58.0</td>
<td>27.2</td>
<td>57.6</td>
<td>41.3</td>
<td>56.3</td>
<td>53.7</td>
<td>52.4</td>
</tr>
<tr>
<td>FlickrG</td>
<td>72.7</td>
<td>68.2</td>
<td>47.3</td>
<td>32.2</td>
<td>30.6</td>
<td>62.3</td>
<td>62.6</td>
<td>65.9</td>
<td>28.1</td>
<td>52.2</td>
<td>39.5</td>
<td>65.1</td>
<td>60.0</td>
<td>71.7</td>
<td>58.2</td>
<td>27.3</td>
<td>58.0</td>
<td>41.5</td>
<td>57.2</td>
<td>53.8</td>
<td>52.7</td>
</tr>
</tbody>
</table>

Object Detection (PASCAL VOC 2012)
### Object Localization (PASCAL VOC 2007)

<table>
<thead>
<tr>
<th>VOC 2007 test</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVAN 14</td>
<td>14.0</td>
<td>36.2</td>
<td>12.5</td>
<td>10.3</td>
<td>9.2</td>
<td>35.0</td>
<td>35.9</td>
<td>8.4</td>
<td>10.0</td>
<td>17.5</td>
<td>6.5</td>
<td>12.9</td>
<td>30.6</td>
<td>27.5</td>
<td>6.0</td>
<td>1.5</td>
<td>18.8</td>
<td>10.3</td>
<td>23.5</td>
<td>16.4</td>
<td>17.1</td>
</tr>
<tr>
<td>GoogleO</td>
<td>30.2</td>
<td>34.3</td>
<td>16.7</td>
<td>13.3</td>
<td>6.1</td>
<td>43.6</td>
<td>27.4</td>
<td>22.6</td>
<td>6.9</td>
<td>16.4</td>
<td>10.0</td>
<td>21.3</td>
<td>25.0</td>
<td>35.9</td>
<td>7.6</td>
<td>9.3</td>
<td>21.8</td>
<td>17.3</td>
<td>31.0</td>
<td>18.1</td>
<td>20.7</td>
</tr>
<tr>
<td>GoogleA</td>
<td>29.5</td>
<td>38.3</td>
<td>15.1</td>
<td>14.0</td>
<td>9.1</td>
<td>44.3</td>
<td>29.3</td>
<td>24.9</td>
<td>6.9</td>
<td>15.8</td>
<td>9.7</td>
<td>22.6</td>
<td>23.5</td>
<td>34.3</td>
<td>9.7</td>
<td>12.7</td>
<td>21.4</td>
<td>15.8</td>
<td>33.4</td>
<td>19.4</td>
<td>21.5</td>
</tr>
<tr>
<td>FlickrG</td>
<td>32.6</td>
<td>42.8</td>
<td>19.3</td>
<td>13.9</td>
<td>9.2</td>
<td>46.6</td>
<td>29.6</td>
<td>20.6</td>
<td>6.8</td>
<td>17.8</td>
<td>10.2</td>
<td>22.4</td>
<td>26.7</td>
<td>40.8</td>
<td>11.7</td>
<td>14.0</td>
<td>19.0</td>
<td>19.0</td>
<td>34.0</td>
<td>21.9</td>
<td>22.9</td>
</tr>
<tr>
<td>FlickrG-EA</td>
<td><strong>32.7</strong></td>
<td><strong>44.3</strong></td>
<td>17.9</td>
<td>14.0</td>
<td><strong>9.3</strong></td>
<td><strong>47.1</strong></td>
<td>26.6</td>
<td>19.2</td>
<td>8.2</td>
<td>18.3</td>
<td>10.0</td>
<td>22.7</td>
<td>25.0</td>
<td>42.5</td>
<td>12.0</td>
<td>12.7</td>
<td><strong>22.2</strong></td>
<td>20.9</td>
<td>35.6</td>
<td>18.2</td>
<td>23.0</td>
</tr>
<tr>
<td>FlickrG-CE</td>
<td>30.2</td>
<td>41.3</td>
<td><strong>21.7</strong></td>
<td><strong>18.3</strong></td>
<td>9.2</td>
<td>44.3</td>
<td>32.2</td>
<td><strong>25.5</strong></td>
<td>9.8</td>
<td><strong>21.5</strong></td>
<td><strong>10.4</strong></td>
<td><strong>26.7</strong></td>
<td><strong>42.8</strong></td>
<td><strong>12.6</strong></td>
<td>13.3</td>
<td>20.4</td>
<td><strong>20.9</strong></td>
<td><strong>36.2</strong></td>
<td><strong>22.8</strong></td>
<td><strong>24.4</strong></td>
<td></td>
</tr>
</tbody>
</table>
Failure Modes

Image source: reference paper
Failure Modes

Image source: reference paper
Scene Classification (MIT-67)

<table>
<thead>
<tr>
<th>Indoor-67</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet [62]</td>
<td>56.8</td>
</tr>
<tr>
<td>OverFeat [42]</td>
<td>58.4</td>
</tr>
<tr>
<td>GoogleO [Obj.]</td>
<td>58.1</td>
</tr>
<tr>
<td>FlickrG [Obj.]</td>
<td>59.2</td>
</tr>
<tr>
<td>GoogleA [Obj. + Sce.]</td>
<td>66.5</td>
</tr>
</tbody>
</table>
Conclusion

- Two-stage bootstrapping approach to training CNNs using web data
- First train on easy images downloaded from Google which is used to initialize the network and the relationship graph
- Then finetune the network on Flickr images and use the relationship graph for regularization
- Network has similar results to ImageNet pre-trained CNN on VOC 2007 and outperforms on VOC 2012 for object detection
- Object localization and scene classification can be done webly-supervised
LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, Jianxiong Xiao
arXiv, 2015
Motivation

- Data hungry algorithms everywhere
- ImageNet is out-of-date
- 59 papers with “Deep” in the title in CVPR 2015

Slide credit: Fisher Yu
Problem Statement

• Goal is to build a dataset containing hundreds of millions of images

• Why?
  – Deep learning models usually have millions of parameters
  – Searching for the optimal settings requires a massive amount of training data with accurate labels
  – Human labeled data is expensive, inefficient, and contains mistakes
  – Partially automate data collection using deep learning methods and statistical guarantees to catch up with the progress of algorithms and computers
Gathering Images

- LSUN dataset has the same scene categories as SUN
- Images are acquired using Google search queries combined with 696 manually chosen common adjectives
- Around 600 million images downloaded
- Duplicate images are ignored (not removed)
Approach

(1) Random samples from unknown set

(2) Interface design & Human labelling

(3) Train classifier: Deep learning & SVM

(4) Propagate information & relabel unknowns

95% Precision

99% Recall

Image source: reference paper
Example: Train an object detector to detect cats in a scene containing 9 dogs and some cats. 4 dogs are correctly detected while 3 of the detected dogs are actually cats. $P = \frac{4}{7}$ while $R = \frac{4}{9}$. 

AMT Labeling Interface

Is this a kitchen? (red: no, green: yes)

Question & Explanation

Task description

Image source: reference paper
Lifetime of AMT Hit

AMT Hit

Instruction Example (15 images)

Random shuffle

Online Ground Truth (20 images)  Hidden Ground Truth (20 images)  New Images (150 images)

Look carefully at the examples and rectify answers until all correct.

Refine the labelling until the accuracy is higher than threshold.

Instruction → Load image to interface → Label examples → Correct? → Label other images → Pass online check? → Pass offline check? → Accept

Reject

Image source: reference paper
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Method</th>
<th>Positive images</th>
<th>Precision</th>
<th>Positive labels</th>
<th>Label Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Clustering</td>
<td>941,981</td>
<td>96.9%</td>
<td>9,515</td>
<td>1%</td>
</tr>
<tr>
<td>1</td>
<td>ConvNet Feature + SVM</td>
<td>1,918,913</td>
<td>96.4%</td>
<td>41,785</td>
<td>2.1%</td>
</tr>
<tr>
<td>2</td>
<td>ConvNet Feature + SVM</td>
<td>2,011,332</td>
<td>96.2%</td>
<td>88,259</td>
<td>4.4%</td>
</tr>
<tr>
<td>3</td>
<td>ConvNet Fine Tuning</td>
<td>2,140,763</td>
<td>96.1%</td>
<td>88,259</td>
<td>4.1%</td>
</tr>
<tr>
<td>4</td>
<td>ConvNet Fine Tuning</td>
<td>2,215,244</td>
<td>95.9%</td>
<td>147,453</td>
<td>6.6%</td>
</tr>
</tbody>
</table>
Experiments
Conclusion

- Datasets are a major roadblock to advancing progress in visual recognition.
- A large dataset called LSUN was created with millions of accurately labeled images.
- Simple experiments were demonstrated to show the datasets potential.
- Construction of the dataset is still underway.