A Fast Data Collection and Augmentation Procedure for Object Recognition

Benjamin Sapp
Ashutosh Saxena
Andrew Y. Ng

Stanford University
Goal of this talk

- Data
  - Training data quality/quantity has a significant impact on the performance of learning algorithms.
  - Often, even an inferior learning algorithm will outperform a superior one, if given more data to learn from.

- Data collection/creation/augmentation techniques.
  - Our approach

- Applications using large amounts of data
  - Object recognition
  - Robotics
Acquiring large, high quality data sets remains a bottleneck for scaling object identification to many categories.
- Manual collection
- Web search

Dataset creation/augmentation techniques:
- Computer graphics using 3D models.
- Our approach
Data Collection Techniques: Web Search

- E.g., LabelMe, Pascal VOC, Caltech-101/256
- Realistic examples in real-world environments are extremely rare
- Most results come from product catalogs with unrealistic backgrounds.
Data Collection Techniques: Web search

- E.g., LabelMe, Pascal VOC, Caltech-101/256
- Realistic examples in real-world environments are extremely rare
- Most results come from product catalogs with unrealistic backgrounds
- Define a *good example* to be:
  - a qualitatively good representative of the object class
  - without occlusion
  - in a real world environment
Data Collection Techniques: Web search

- In Caltech-256, only 21/201 watches are occur in a realistic setting:

- Only 2/10 object categories yielded more than 100 good results from a Google image search.

Top Google search results for “hammer”: 
Data Collection Techniques: Computer Graphics

- Everingham & Zisserman – face models.
- Saxena et al. 2006 - grasping objects.
Data collection techniques: Computer Graphics

Michels, Saxena & Ng, 2005.

- Predict depth for driving a car.
- Use computer graphics data.
Data Collection Techniques: Computer Graphics

Problems:

- Difficult to achieve photorealism
- Time consuming to generate single model
- Non-trivial to synthesize unique, intra-class objects from existing models.
Prior Work using Large amount of Data

- Natural Language

  Scaling to Very Very Large Corpora for Natural Language Disambiguation, Michele Banko and Eric Brill, 2001.

- Object recognition

  Learning methods for generic object recognition with invariance to pose and lighting, LeCun et al., 2004.

  (Did not address object recognition in real-world environments---when objects are not “uniform green”---against real, cluttered backgrounds.)
Our approach

- More data helps.

- How can we produce huge amounts of “realistic” data quickly?

- We propose a probabilistic framework for synthesizing large, effective datasets at a fraction of the human cost.
Our approach

Model a training example as generative probabilistic model over foreground, background and shadow components:

\[
P(S_i | I_{OBJ_i}, I_{FG_i}, I_{BG_i}, I_{GRN_i})
= \sum_{m_i \in \{fg, bg, sh\}} P(S_i | m_i, I_{OBJ_i}, I_{FG_i}, I_{BG_i}, I_{GRN_i}) \cdot P(m_i | I_{OBJ_i}, I_{FG_i}, I_{BG_i}, I_{GRN_i})
\]

**\(I_{OBJ}\) Image of object captured against green screen**

**\(I_{GRN}\) Blank green screen image**

**\(I_{BG}\) Background image (e.g., office environment)**

**\(I_{FG}\) Foreground texture image**

**\(\hat{S}\) Synthetic image inferred**
Probabilistic Model

\[ P(m_i \mid I_{OBJ_i}, I_{GRN_i}, I_{FG_i}, I_{BG_i}) \]
\[ \propto P(I_{GRN_i} \mid m_i, I_{OBJ_i}) P(I_{OBJ_i} \mid m_i) P(m_i) \]

- \( P(I_{OBJ_i} \mid m_i = \text{bg}) = \text{mixture of Gaussians} \)
- \( P(I_{OBJ_i} \mid m_i = \text{sh}) = \text{mixture of Gaussians} \)
- \( P(I_{OBJ_i} \mid m_i = \text{fg}) = \mathcal{N}(I_{OBJ_i}; \mu_{fg}, \Sigma_{fg}) \)

- \( P(I_{GRN_i} \mid m_i = \text{bg}, I_{OBJ_i}) = \mathcal{N}(I_{GRN_i}; I_{OBJ_i}, \Sigma_1) \)
- \( P(I_{GRN_i} \mid m_i = \text{sh}, I_{OBJ_i}) = \mathcal{N}(I_{GRN_i}; I_{OBJ_i} + \mu_s, \Sigma_1) \)
- \( P(I_{GRN_i} \mid m_i = \text{fg}, I_{OBJ_i}) = P(I_{GRN_i} \mid m_i = \text{fg}) = \text{mixture of Gaussians} \)

- \( P(m_i) = \text{uniform prior} \)
To obtain varied realistic images, we take a linear blend of existing components and new components:

\[
P(S_i|m_i = fg, I_{OBJ_i}, I_{FG_i}; w_{fg}) = \\
(1/Z) \exp \left( -\|S_i - w_{fg}^T [I_{OBJ_i}, I_{FG_i}]^T \|^2 / 2 \right)
\]

\[
P(S_i|m_i = bg, I_{OBJ_i}, I_{BG_i}; w_{bg}) = \\
(1/Z) \exp \left( -\|S_i - w_{bg}^T [I_{OBJ_i}, I_{BG_i}]^T \|^2 / 2 \right)
\]

\[
P(S_i|m_i = sh, I_{OBJ_i}, I_{BG_i}; w_{sh}) = \\
(1/Z) \exp \left( -\|S_i - w_{sh}^T [I_{OBJ_i}, I_{BG_i}]^T \|^2 / 2 \right)
\]

Sapp, Saxena & Ng, Stanford University
## Synthesis techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>$w^T$</th>
<th>$I^T_{FG, BG}$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unaltered</td>
<td>$[1, 0]$</td>
<td>$-$</td>
<td>Leave $bg$/$fg$ from $I_{OBJ}$ unaltered</td>
</tr>
<tr>
<td>white/black</td>
<td>$[0, 1]$</td>
<td>$[255, 255, 255]/[0, 0, 0]$</td>
<td>Replace $bg$/$fg$ with white/black</td>
</tr>
<tr>
<td>uniform</td>
<td>$[0, 1]$</td>
<td>$(u_1, u_2, u_3), u_i \sim U(0, 255)$</td>
<td>Randomly sample each pixel uniformly</td>
</tr>
<tr>
<td>noise</td>
<td>learnt</td>
<td>$I_{\text{noise}}$</td>
<td>$I_{\text{noise}} = I_{\text{uniform}} \ast G$, $G$ is a $3 \times 3$ Gaussian filter with $\sigma = 0.95$</td>
</tr>
<tr>
<td>corel</td>
<td>learnt</td>
<td>$I_{\text{corel}}$</td>
<td>Generic image database from the web, see Li03</td>
</tr>
<tr>
<td>office</td>
<td>learnt</td>
<td>$I_{\text{office}}$</td>
<td>Images collected from office environments.</td>
</tr>
<tr>
<td>NRT</td>
<td>learnt</td>
<td>$I_{\text{tex}}$</td>
<td>Near-Regular Texture database with 188 textures, see CMU-NRT</td>
</tr>
</tbody>
</table>

![Synthesis Techniques Example](image)
Speedup

- Synthetic data sets automatically generated from green screen data.

- Collected 200 objects/class on green screen, manually varying object pose between shots.

- Green screen collection took 19 minutes per class, vs. 175 minutes collecting in the wild – **9.2x speedup**

- Augmenting data sets yields further speedup: 135x.
Experiments

- Office object classification
  - Empirical analysis of synthesis techniques vs. training with real data
- 3D synthesis/tree classification
  - Synthesize training images of outdoor scenes to identify trees
- 100,000 examples
  - Augment training set to upper limits for large-scale training

(All datasets have equal number of positive and negative examples in test and training set.)
Experiment: Synthesis Techniques

- 10 office objects collected for experiments:

- Trained Gentleboost classifiers of C1 features (Serre, Wolf, Poggio 2005)
- Compared training with various synthetic techniques vs. training with real data.
- All results come from testing on real data
- Real and synthetic data available at:
  - http://ai.stanford.edu/~asaxena/robotdatacollection/
### Background Variations

<table>
<thead>
<tr>
<th>Object</th>
<th>unaltered</th>
<th>black</th>
<th>white</th>
<th>uniform</th>
<th>noise</th>
<th>corel</th>
<th>office</th>
<th>NRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mug</td>
<td>.727</td>
<td>.566</td>
<td>.530</td>
<td>.521</td>
<td>.882</td>
<td>.874</td>
<td>.904</td>
<td>.827</td>
</tr>
<tr>
<td>scissor</td>
<td>.822</td>
<td>.529</td>
<td>.516</td>
<td>.505</td>
<td>.937</td>
<td>.951</td>
<td>.952</td>
<td>.899</td>
</tr>
<tr>
<td>stapler</td>
<td>.760</td>
<td>.550</td>
<td>.568</td>
<td>.504</td>
<td>.846</td>
<td>.862</td>
<td>.913</td>
<td>.860</td>
</tr>
<tr>
<td>keyboard</td>
<td>.954</td>
<td></td>
<td></td>
<td></td>
<td>.944</td>
<td>.955</td>
<td>.950</td>
<td>.939</td>
</tr>
<tr>
<td>hammer</td>
<td>.886</td>
<td>.511</td>
<td>.515</td>
<td>.565</td>
<td>.941</td>
<td>.949</td>
<td>.977</td>
<td>.921</td>
</tr>
<tr>
<td>plier</td>
<td>.709</td>
<td>.509</td>
<td>.552</td>
<td>.520</td>
<td>.883</td>
<td>.888</td>
<td>.913</td>
<td>.795</td>
</tr>
<tr>
<td>fork</td>
<td>.612</td>
<td>.515</td>
<td>.517</td>
<td>.493</td>
<td>.670</td>
<td>.623</td>
<td>.668</td>
<td>.605</td>
</tr>
<tr>
<td>watch</td>
<td>.900</td>
<td>.541</td>
<td>.519</td>
<td>.494</td>
<td>.967</td>
<td>.970</td>
<td>.965</td>
<td>.962</td>
</tr>
<tr>
<td>flipphone</td>
<td>.729</td>
<td>.545</td>
<td>.574</td>
<td>.496</td>
<td>.765</td>
<td>.696</td>
<td>.757</td>
<td>.674</td>
</tr>
<tr>
<td>telephone</td>
<td>.940</td>
<td>.500</td>
<td>.502</td>
<td>.694</td>
<td>.954</td>
<td>.942</td>
<td>.958</td>
<td>.963</td>
</tr>
</tbody>
</table>

### Foreground Variations

<table>
<thead>
<tr>
<th>Object</th>
<th>unaltered</th>
<th>black</th>
<th>white</th>
<th>uniform</th>
<th>noise</th>
<th>corel</th>
<th>office</th>
<th>NRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mug</td>
<td>.727</td>
<td>.496</td>
<td>.497</td>
<td>.608</td>
<td>.959</td>
<td>.956</td>
<td>.954</td>
<td>.937</td>
</tr>
<tr>
<td>scissor</td>
<td>.822</td>
<td>.939</td>
<td>.885</td>
<td>.691</td>
<td>.954</td>
<td>.967</td>
<td>.941</td>
<td>.941</td>
</tr>
<tr>
<td>stapler</td>
<td>.760</td>
<td>.780</td>
<td>.749</td>
<td>.507</td>
<td>.942</td>
<td>.957</td>
<td>.953</td>
<td>.911</td>
</tr>
<tr>
<td>keyboard</td>
<td>.954</td>
<td>.500</td>
<td>.851</td>
<td>.850</td>
<td>.980</td>
<td>.978</td>
<td>.977</td>
<td>.972</td>
</tr>
<tr>
<td>hammer</td>
<td>.886</td>
<td>.962</td>
<td>.932</td>
<td>.742</td>
<td>.987</td>
<td>.972</td>
<td>.977</td>
<td>.973</td>
</tr>
<tr>
<td>plier</td>
<td>.709</td>
<td>.819</td>
<td>.757</td>
<td>.652</td>
<td>.872</td>
<td>.935</td>
<td>.921</td>
<td>.930</td>
</tr>
<tr>
<td>fork</td>
<td>.612</td>
<td>.618</td>
<td>.613</td>
<td>.537</td>
<td>.806</td>
<td>.758</td>
<td>.804</td>
<td>.720</td>
</tr>
<tr>
<td>watch</td>
<td>.900</td>
<td>.496</td>
<td>.498</td>
<td>.746</td>
<td>.974</td>
<td>.971</td>
<td>.967</td>
<td>.969</td>
</tr>
<tr>
<td>flipphone</td>
<td>.729</td>
<td>.497</td>
<td>.501</td>
<td>.554</td>
<td>.868</td>
<td>.830</td>
<td>.873</td>
<td>.803</td>
</tr>
<tr>
<td>telephone</td>
<td>.940</td>
<td>.491</td>
<td>.490</td>
<td>.615</td>
<td>.989</td>
<td>.982</td>
<td>.980</td>
<td>.973</td>
</tr>
</tbody>
</table>
Real vs. Synthetic Data, No Augmentation
Real vs. Synthetic Data, With Augmentation
Real vs. Synthetic Data, With Augmentation (Caltech 4)
3D data
3D data

- Want to label each pixel as tree or non-tree
- Have $|T| = 24$ tree, $|N| = 103$ non-tree images for training
- Automatically generate synthetic data by placing trees into non-tree images using depth information available from laser scans
- Can create $|T| \times |N|$ synthetic positive training examples, versus only $|T|$ real positive training examples
Tree Classifier Results

- Using $|N| + |T|$ real examples
- Using $|N| \times |T|$ synthetic examples

AUC vs Number of real training examples
100,000 Examples Setup

- Collected images of 100 unique coffee mugs at a range of poses covering the upper hemisphere
- Use a fast k-NN implementation with cover trees for classification
- Used Caltech-256 “coffee-mug” category for testing
- As a baseline, compared to training with 220 good examples of coffee mugs from the LabelMe dataset.
100,000 Examples Results

Performance

Number of training examples

Trained with synthetic data
Trained with real data
Robotics Application

[[Video]]
Questions?
More Caltech-256 (classification)

Figure 3: Object classification accuracies for various object classes using cover trees. The x-axis describes the training set size and the y-axis describes accuracy. The colors indicate: Green - Regular 1-NN, Cyan - Regular 10-NN, Blue - Bias Shifted 1-NN, Red - Bias Shifted 10-NN.
More Caltech-256 (detection)

Figure 2: Object detection accuracies for various object classes using cover trees. The x-axis describes the training set size and the y-axis describes accuracy. The colors indicate: Green - Regular 1-NN, Cyan - Regular 10-NN, Blue - Bias Shifted 1-NN, Red - Bias Shifted 10-NN.
Figure 4: Object classification accuracies for various object classes using Nister trees. The x-axis describes the training set size and the y-axis describes accuracy. The colors indicate: Red - Performance using 1350 green screen object images, Blue - Performance using generated data.