Focusing Human Attention on the “Right” Visual Data

Kristen Grauman
Department of Computer Science
University of Texas at Austin

Work with Yong Jae Lee, Sudheendra Vijayanarasimhan, and Prateek Jain
Automating visual processing

- Recognizing objects and activities
- 3D Reconstruction
- Tracking people
- Image retrieval
- Detection

[Image retrieval example from Kulis & Grauman, Tumor image from Zhang et al.]
“Semi-automating” visual processing

Key question:
- Which visual data deserves human attention?
“Semi-automating” visual processing

We’ll consider two settings:

1. Supervised learning of object categories
2. Unsupervised video summarization

Key challenges:

• Predicting what is important
• Scaling to large-scale data collections
The importance of data in recognition

Best approaches today rely on discriminative learning.
The importance of data in recognition

• **Dataset creation**


• **Gathering annotations from “crowds”**


• **Active learning to focus human effort**

Active learning for image annotation
Active learning for image annotation

Intent: better models, faster/cheaper
Problem: “Sandbox” learning

Thus far, tested only in artificial settings:

• Unlabeled data already fixed, small scale, biased

• Computational cost ignored

• Really, “researcher in the loop”
Our idea: **Live** active learning

Large-scale active learning of object detectors with **crawled data** and **crowdsourced labels**.

*But how to scale active learning to massive unlabeled pools of data?*
SVM margin criterion for active selection

Select point nearest to hyperplane decision boundary for labeling.

\[ x^* = \arg\min_{x_i \in \mathcal{U}} |w^T x_i| \]

[Tong & Koller, 2000; Schohn & Cohn, 2000; Campbell et al. 2000]
Sub-linear time active selection

We propose a novel hashing approach to identify these most uncertain examples in sub-linear time.
Background: Locality-Sensitive Hashing

Probability a random hyperplane separates two unit vectors depends on the angle between them:

Corresponding hash function:

\[ h_r(x) = \begin{cases} 
1, & \text{if } r^T x \geq 0 \\
0, & \text{otherwise} 
\end{cases} \]

\[ r_i \sim \mathcal{N}(0, 1) \]

Probability of collision:

\[ \Pr(h_r(x_i) = h_r(x_j)) = 1 - \frac{1}{\pi} \cos^{-1}(x_i^T x_j) \]

Hashing a hyperplane query

To retrieve those points for which $|\mathbf{w}^T \mathbf{x}_i|$ is small, want probable collision for **perpendicular** vectors:

Assuming normalized data.

Should collide

Should not collide
Hashing a hyperplane query

We generate two independent random vectors $u$ and $v$:

- one to constrain angle between $x$ and $w$
- one to constrain angle between $x$ and $-w$

Collision likely only if neither vector splits

For parallel vectors

Unlikely to split and Likely to split

= Likely to split

For perpendicular vectors

Less likely to split and Less likely to split

= Unlikely to split
Hashing a hyperplane query

• We define an asymmetric 2-bit hash function:

\[
\begin{align*}
H_{-}\text{Hash family:} & \\
\hat{h}(z) &= \begin{cases} 
    h_{u,v}(z, z), & \text{if } z \text{ is a database point vector,} \\
    h_{u,v}(z, -z), & \text{if } z \text{ is a query hyperplane vector.}
\end{cases}
\end{align*}
\]

where 
\[
h_{u,v}(a, b) = [h_u(a), h_v(b)] = [\text{sign}(u^T a), \text{sign}(v^T b)]
\]

• We prove necessary conditions for locality sensitivity:

\[
\Pr[\hat{h}(w) = \hat{h}(x)] = \Pr[h_u(w) = h_u(x)] \Pr[h_v(-w) = h_v(x)]
\]

\[
= \frac{1}{4} - \frac{1}{\pi^2} \left( \theta_{x,w} - \frac{\pi}{2} \right)^2
\]

[Jain, Vijayanarasimhan & Grauman, NIPS 2010]
Hashing a hyperplane query

At each iteration of the learning loop, our hash functions map the current hyperplane directly to its nearest unlabeled points.
H-Hash result on Tiny Images

Images actively selected in first 9 iterations

Learning “airplane”

Learning “automobile”

Efficient active selection with pool of 1 Million unlabeled examples and 1000s of categories!
H-Hash result on Tiny Images

Accuracy improvements as more data labeled

Time spent searching for selection

By minimizing both selection and labeling time, obtain the best accuracy per unit time.
Live active learning

For 4.5 million unlabeled instances, 10 minutes machine time per iter, vs. 60 hours for a naïve scan.

[Vijayanarasimhan & Grauman CVPR 2011]
Live active learning results

Flickr test set

Outperforms status quo data collection approach
Live active learning results

First selections made when learning “boat”:

Ours: live active learning

Keyword+image baseline
PASCAL Visual Object Classes (VOC)

- “The” object detection benchmark
- Train/test data from Flickr

http://pascallin.ics.uci.edu/datasets/VOC/

PASCAL Live active learning results

Live learning improves the state-of-the-art for some of most difficult PASCAL VOC categories:

<table>
<thead>
<tr>
<th></th>
<th>bird</th>
<th>boat</th>
<th>dog</th>
<th>potted plant</th>
<th>sheep</th>
<th>chair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>15.8*</td>
<td>18.9*</td>
<td>25.3*</td>
<td>11.6*</td>
<td>28.4*</td>
<td>9.1*</td>
</tr>
<tr>
<td>Previous best</td>
<td>15.3</td>
<td>16.8</td>
<td>21.5</td>
<td>14.6</td>
<td>23.9</td>
<td>17.9</td>
</tr>
</tbody>
</table>

Our approach’s efficiency makes live learning feasible

<table>
<thead>
<tr>
<th></th>
<th>Active selection</th>
<th>Training</th>
<th>Detection per image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours + active</td>
<td>10 mins</td>
<td>5 mins</td>
<td>150 secs</td>
</tr>
<tr>
<td>LSVM [Felzenszwalb et al. 2009]</td>
<td>3 hours</td>
<td>4 hours</td>
<td>2 secs</td>
</tr>
<tr>
<td>SP+MKL [Vedaldi et al. 2009]</td>
<td>93 hours</td>
<td>&gt; 2 days</td>
<td>67 secs</td>
</tr>
</tbody>
</table>

Previous best : [Vedaldi et al. ICCV 2009] or [Felzenszwalb et al. PAMI 2009]
Summary so far

- Active training for object recognition
- Breaking free from “sandbox” learning requires new large-scale learning algorithms
- Live active learning challenges the status quo in data collection

Main contributions:
- Hyperplane hashing for sub-linear time active selection
- First autonomous live learning results
“Semi-automating” visual processing

We’ll consider two settings:
1. Supervised learning of object categories
2. Unsupervised video summarization

Key challenges:
• Predicting what is important
• Scaling to large-scale data collections
**Problem:** Summarizing egocentric videos

**Input:** Egocentric video of the camera wearer’s day

**Output:** Storyboard summary
Potential applications of egocentric video summarization

Steve Mann life logger; RHex Hexapedal Robot, Penn’s GRASP Laboratory
Existing approaches to video summarization


• They are indifferent to the impact that each object has on generating the “story” of the video.
Important person/object discovery

• **Our idea**: Discover important people and objects for egocentric video summarization
  
  – "**Important**": things with which the camera wearer has significant interaction
  
  – Develop novel egocentric and high-level saliency features to train a *category-independent* important person/object detector
  
  – Produce a concise visual summary driven by those detections

[Lee, Ghosh, Grauman, CVPR 2012]
Approach overview

1) Crowd-source important person/object annotations

2) Design features to train an importance detector

3) Given a new video, segment it into unique temporal events

4) For each event, discover important people and objects

5) Create a compact storyboard summary that encapsulates the main people and objects

[Lee, Ghosh, Grauman, CVPR 2012]
Data collection

- 15 fps, 320 x 480 resolution
- 10 videos, 3-5 hrs in length; total of 37 hrs
- Four subjects: one undergraduate, two grad students, and one office worker
Crowdsourcing training data

First task: watch a short clip, and describe in text the essential people or objects necessary to create a summary.
Crowdsourcing training data

- **Second task**: draw polygons around any described person or object *obtained from the first task* in sampled frames
Input: egocentric video

• Uncontrolled setting prohibits reliable space-time segmentation
Learning region importance

Egocentric features:

- distance to hand
- distance to frame center
- frequency
Learning region importance

Egocentric features:
- distance to hand
- distance to frame center
- frequency

Object features:
- Candidate region’s appearance, motion
- Surrounding area’s appearance, motion
- “Object-like” appearance, motion
  [Endres et al. ECCV 2010, Lee et al. ICCV 2011]

Region features: size, width, height, centroid
Learning region importance

\[ I(r) = \beta_0 + \sum_{i=1}^{N} \beta_i x_i(r) + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \beta_{i,j} x_i(r)x_j(r) \]

- Regressor to learn and predict a region’s *degree* of importance
- Expect significant *interactions* between the features
- For training: \( I(r) = \frac{|GT \cap r|}{|GT \cup r|} \)
- For testing: predict \( I(r) \) given \( x_i(r) \)'s
Segmenting the video into events

Events allow summary to include multiple instances of a person/object that is central in multiple contexts in the video
Discovering an event’s key people/objects

1. Event A
2. Score and group regions in event
3. Select representative region with highest $I(r)$

Procedure:
1. Collect training data
2. Learn Importance
3. Segment video into events
4. Discover important regions
5. Storyboard summary
Generating a storyboard summary

- Display event boundaries and frames of the selected important people and objects
Results: Important region prediction

- **Ours**
- **Object-like [Carreira, 2010]**
- **Object-like [Endres, 2010]**
- **Saliency [Walther, 2005]**

Good predictions
Results: Important region prediction

Ours
Object-like [Carreira, 2010]
Object-like [Endres, 2010]
Saliency [Walther, 2005]

Failure cases

Precision

Recall

Important (Ours): 0.26
Object-like [24]: 0.14
Object-like [35]: 0.08
Saliency [156]: 0.04
Results: Egocentric video summarization

Alternative methods for comparison

Uniform keyframe sampling (12 frames)  

[Liу & Kender, 2002] (12 frames)
Results: Egocentric video summarization

- Our summaries include more important objects with fewer frames
Results: Egocentric video summarization

<table>
<thead>
<tr>
<th></th>
<th>Much better</th>
<th>Better</th>
<th>Similar</th>
<th>Worse</th>
<th>Much worse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imp. captured</td>
<td>31.25%</td>
<td>37.5%</td>
<td>18.75%</td>
<td>12.5%</td>
<td>0%</td>
</tr>
<tr>
<td>Overall quality</td>
<td>25%</td>
<td>43.75%</td>
<td>18.75%</td>
<td>12.5%</td>
<td>0%</td>
</tr>
</tbody>
</table>

- User studies to compare ours vs. uniform sampling
  1. Which summary captures important objects/people better?
  2. Which provides a better overall summary?
Results: Egocentric video summarization

Generating a storyboard map
Summary

• Learn to focus human attention on the right data
  – Actively train object detector with human in the loop
  – Summarize videos for fast human consumption

• Semi-automating computer vision tasks → new applications in large-scale visual analysis