Style-aware Mid-level Representation for Discovering Visual Connections in Space and Time

Paper Presentation By Bhavin Modi

Slides By- Yong Jae Lee, Alexei A. Efros, and Martial Hebert
Carnegie Mellon University / UC Berkeley
ICCV 2013
Long before the age of “data mining” ...

when?
(historical dating)

where?
(botany, geography)
when?
1972
where? Krakow, Poland

“The View From Your Window” challenge

Church of Peter & Paul
Visual data mining in Computer Vision

- Most approaches mine **globally consistent** patterns

**Low-level “visual words”**

**Object category discovery**
Visual data mining in Computer Vision

• Recent methods discover *specific* visual patterns

Mid-level visual elements
Problem

• Much in our visual world undergoes a **gradual change**

Temporal:
• Much in our visual world undergoes a *gradual change*.

**Spatial:**
Our Goal

• Mine mid-level visual elements in temporally- and spatially-varying data and model their “visual style”

where?
Geolocalization of StreetView images

[Kim et al. 2010, Fu et al. 2010, Palermo et al. 2012]

when?
Historical dating of cars

Key Idea

1) Establish connections

1926  

1947  

1975

“closed-world”

2) Model style-specific differences

1926  1947  1975
Approach
Unsupervised Discovery of Mid-Level Discriminative Patches
**Fig. 1.** The top two detected Visual Words (bottom) vs. Mid-level Discriminative Patches (top), trained without any supervision and on the same large unlabeled dataset.
Can we get nice parts without supervision?

- Idea 0: K-means clustering in HOG space
Still not good enough

- The SVM memorizes bad examples and still scores them highly
- However, the space of bad examples is much more diverse
- So we can avoid overfitting if we train on a training subset but look for patches on a validation subset
Why K-means on HOG fails?

• Chicken & Egg Problem
  – If we know that a set of patches are visually similar we can easily learn a distance metric for them
  – If we know the distance metric, we can easily find other members
Idea 1: Discriminative Clustering

- Start with K-Means
- Train a discriminative classifier for the distance function, using all other classes as negative examples
- Re-assign patches to clusters whose classifier gives highest score
- Repeat
Idea 2: Discriminative Clustering+

• Start with K-Means or kNN
• Train a discriminative classifier for the distance function, using Detection
• Detect the patches and assign to top k clusters
• Repeat
Can we get good parts without supervision?

• What makes a good part?
  – Must occur frequently in one class (representative)
  – Must not occur frequently in all classes (discriminative)
Discriminative Clustering+
Discriminative Clustering+
Idea 3: Discriminative Clustering++

• Split the discovery dataset into two equal parts (training and validation)
• Train on the training subset
• Run the trained classifier on the validation set to collect examples
• Exchange training and validation sets
• Repeat
Discriminative Clustering++

KMeans

Iter 1

Iter 2

Iter 3

Iter 4
**Algorithm 1 Discover Top n Discriminative Patches**

**Require:** Discovery set $\mathcal{D}$, Natural World set $\mathcal{N}$

1. $\mathcal{D} \Rightarrow \{D_1, D_2\}$; $\mathcal{N} \Rightarrow \{N_1, N_2\}$  $\triangleright$ Divide $\mathcal{D}, \mathcal{N}$ into equal sized disjoint sets
2. $S \leftarrow \text{rand\_sample}(D_1)$  $\triangleright$ Sample random patches from $D_1$
3. $K \leftarrow \text{kmeans}(S)$  $\triangleright$ Cluster patches using KMeans
4. **while not converged() do**
5.   **for all** $i$ **such that** $\text{size}(K[i]) \geq 3$ **do**  $\triangleright$ Prune out small ones
6.     $C_{\text{new}}[i] \leftarrow \text{svm\_train}(K[i], N_1)$  $\triangleright$ Train classifier for each cluster
7.     $K_{\text{new}}[i] \leftarrow \text{detect\_top}(C[i], D_2, m)$  $\triangleright$ Find top $m$ new members in other set
8.   **end for**
9. $K \leftarrow K_{\text{new}}; \ C \leftarrow C_{\text{new}}$
10. swap($D_1, D_2$); swap($N_1, N_2$)  $\triangleright$ Swap the two sets
11. **end while**
12. $A[i] \leftarrow \text{purity}(K[i]) + \lambda \times \text{discriminativeness}(K[i]) \ \forall \ i$  $\triangleright$ Compute scores
13. **return** select\_top($C, A, n$)  $\triangleright$ Sort according to scores and select top $n$ patches
Doublets: Discover second-order relationships

- Start with high-scoring patches
- Find spatial correlations to other (weaker patches)
- Rank the potential doublets on validation set
Doublets

Fig. 5. Cluster clean-up using "doublets". A visually non-homogeneous cluster (b) that has learned more than one concept, when coupled into a doublet with a high quality cluster (a), gets cleaned up (c).

Fig. 6. Examples of discovered discriminative "doublets" that were highly ranked.
Table 1. Quantitative Evaluation: Average Classification on MIT Indoor-67 dataset.
*Current state-of-the-art. †Best performance from various vocabulary sizes.
Coming Back
Mining style-sensitive elements

• Sample patches and compute nearest neighbors

[Dalal & Triggs 2005, HOG]
Mining style-sensitive elements

Patch | Nearest neighbors

- [Images of car patches and nearest neighbors]
Mining style-sensitive elements

Patch

Nearest neighbors

style-sensitive
Mining style-sensitive elements

Patch

Nearest neighbors

style-insensitive
### Mining style-sensitive elements

<table>
<thead>
<tr>
<th>Patch</th>
<th>Nearest neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Patch" /></td>
<td><img src="image2.png" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Patch" /></td>
<td><img src="image4.png" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Patch" /></td>
<td><img src="image6.png" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Patch" /></td>
<td><img src="image8.png" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td><img src="image9.png" alt="Patch" /></td>
<td><img src="image10.png" alt="Nearest neighbors" /></td>
</tr>
</tbody>
</table>
Mining style-sensitive elements

<table>
<thead>
<tr>
<th>Patch</th>
<th>Nearest neighbors</th>
<th>tight</th>
<th>uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1929</td>
<td>1927</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1937</td>
<td>1959</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>1947</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1946</td>
<td>1948</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1971</td>
<td>1972</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1973</td>
<td>1972</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1938</td>
<td>1939</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1940</td>
<td>1949</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mining style-sensitive elements

(a) Peaky (low-entropy) clusters
Mining style-sensitive elements

(b) Uniform (high-entropy) clusters
Making visual connections

• Take top-ranked clusters to build correspondences
Making visual connections

• Train a detector (HoG + linear SVM) [Singh et al. 2012]

Natural world “background” dataset
Making visual connections

Top detection per decade

[Singh et al. 2012]
Making visual connections

- We expect style to change gradually...

1920s

1930s

1940s

Natural world “background” dataset
Making visual connections

Top detection per decade

1920s  1930s  1940s  1950s  1960s  1970s  1980s  1990s
Making visual connections

Top detection per decade
Making visual connections

Initial model (1920s)

Initial model (1940s)

Final model

Final model
Results: Example connections
Training style-aware regression models

- Support vector regressors with Gaussian kernels
- Input: HOG, output: date/geo-location
Training style-aware regression models

• Train image-level regression model using outputs of visual element detectors and regressors as features
Figure 8. We visualize the styles that a single style-aware regressor has learned by averaging the predictions for each decade.
Results
Results: Date/Geo-location prediction

Crawled from www.cardatabase.net

• 13,473 images
• Tagged with year
• 1920 – 1999

Crawled from Google Street View

• 4,455 images
• Tagged with GPS coordinate
• N. Carolina to Georgia
Figure 6. Box plots showing date and location prediction error on the CarDb and EDb datasets, respectively. Lower values are better. Our approach models the subtle stylistic differences for each discovered element in the data, which leads to lower error rates.
Results: Date/Geo-location prediction

<table>
<thead>
<tr>
<th></th>
<th>Ours</th>
<th>Singh et al. [23, 4]</th>
<th>SP [13]</th>
<th>BOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>CarDb (years)</td>
<td>8.56</td>
<td>9.72</td>
<td>11.81</td>
<td>15.39</td>
</tr>
<tr>
<td>EDb (miles)</td>
<td>77.66</td>
<td>87.47</td>
<td>83.92</td>
<td>97.78</td>
</tr>
<tr>
<td>IMCDb (years)</td>
<td>13.53</td>
<td>15.32</td>
<td>17.06</td>
<td>18.65</td>
</tr>
</tbody>
</table>

Table 1. Mean absolute error on CarDb, EDb, and IMCDb for all methods. The result on IMCDb evaluates cross-dataset generalization performance. Lower values are better.
Results: Learned styles

Average of top predictions per decade
Extra: Fine-grained recognition

Mean classification accuracy on Caltech-UCSD Birds 2011 dataset

<table>
<thead>
<tr>
<th></th>
<th>Ours</th>
<th>Zhang et al. CVPR 2012</th>
<th>Berg, Belhumeur CVPR 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>41.01</td>
<td>28.18</td>
<td>56.89</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Zhang et al. ICCV 2013</th>
<th>Chai et al. ICCV 2013</th>
<th>Gavves et al. ICCV 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50.98</td>
<td>59.40</td>
<td>62.70</td>
</tr>
</tbody>
</table>

weak-supervision

strong-supervision
Conclusions

• Models *visual style*: appearance correlated with time/space

• First establish visual connections to create a *closed-world*, then focus on *style-specific differences*
Thank you!
References