City Forensics: Using Visual Elements to Predict Non-Visual City Attributes

Sean M. Arietta    Alexei A. Efros    Ravi Ramamoorthi    Maneesh Agrawala

Experiment presented by: Yu-Chuan Su and Paul Choi
Review

1. Discover visual elements which are indicative of attribute
2. Detect presence of visual elements
3. Predict value of attribute
Review: Training

1. Interpolate data to obtain attribute values over entire city

2. Build bank of SVMs to detect visual elements which are discriminative of attribute

3. Train attribute predictor from SVMs using Support Vector Regression
Review: Training

1. Interpolate data to obtain attribute values over entire city

2. **Build bank of SVMs to detect visual elements which are discriminative of attribute**

3. Train attribute predictor from SVMs using Support Vector Regression
Building an SVM bank

1. **Candidate selection**
   - Find clusters of image patches (visual elements) which are frequent and discriminative

2. **Initial SVM training**
   - Train SVMs for classifying each candidate visual element

3. **Iterative clustering**
   - Iteratively retrain SVMs using the previous top detections as the new positives
Experiment

- How does each step of the method help find more discriminative visual elements?
- Build a bank of SVMs for simple binary classification problems: duck or parrot, car or not car.
- Qualitatively evaluate visual elements found at each step.

Image credit: Doersch et al. 2012
Datasets

- Caltech-UCSD Birds-200-2011
- PASCAL VOC2012: cars
- UIUC Image Database for Car Detection
Our implementation

● Several patches at several scales from each training image
● VGG-16 features
● Candidate visual elements:
  ○ Don’t have too much spatial overlap with nearest neighbors
  ○ Have a high ratio of positive examples in nearest neighbors
● Discriminative training
  ○ For each candidate, train a Linear SVM to separate the 5 nearest neighbors from all negative examples
  ○ Re-train 3 more times, using the top 5 detections as the new positive examples
Expected results
Pascal VOC (Car v.s. Non-car)

Detected Element 1

kNN

ESVM

Iter. 1

Iter. 2

Iter. 3

Detected Element 2
UIUC_CAR

Detected Element 1

kNN

Detected Element 2

ESVM

Iter. 1

Iter. 2

Iter. 3
Caltech-UCSD Birds-200-2011

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Iter. 1</th>
<th>Iter. 2</th>
<th>Iter. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN</td>
<td>![Images]</td>
<td>![Images]</td>
<td>![Images]</td>
</tr>
<tr>
<td>ESVM</td>
<td>![Images]</td>
<td>![Images]</td>
<td>![Images]</td>
</tr>
<tr>
<td>Iter. 1</td>
<td>![Images]</td>
<td>![Images]</td>
<td>![Images]</td>
</tr>
<tr>
<td>Iter. 2</td>
<td>![Images]</td>
<td>![Images]</td>
<td>![Images]</td>
</tr>
<tr>
<td>Iter. 3</td>
<td>![Images]</td>
<td>![Images]</td>
<td>![Images]</td>
</tr>
</tbody>
</table>
Issue: converging to same elements
Issue: converging to same elements

- Imbalanced training data
- Large & diverse training data is necessary
- Method sensitive to meta-parameters
Issue: cluster drift
Issue: cluster drift

- Positive samples mining does not depend on class label
Conclusion

- **The method depends on**
  - Large & diverse training data
  - Careful per-dataset tuning
- **We successfully find informative parts in Pascal VOC**
- **Difficulties we encounter**
  - Training time
  - Imbalanced data
- **Possible improvements**
  - Hard negative mining
  - Use label in iterative training