Towards Efficient Learning of Optimal Spatial Bag-of-Words Representations

Lu Jiang¹, Wei Tong¹, Deyu Meng², Alexander G. Hauptmann¹

¹ School of Computer Science, Carnegie Mellon University
² School of Mathematics and Statistics, Xi'an Jiaotong University
People

CMU Informedia Team

Wei Tong
Deyu Meng
Alexander G. Hauptmann
Outline

- Motivation
- Related Work
- Jensen - Shannon Tiling
- Experiment Results
- Conclusions
Outline

- Motivation
- Related Work
- Jensen - Shannon Tiling
- Experiment Results
- Conclusions
Spatial Bag-of-Words

- The Spatial Bag-of-Words (BoW) model has proven one of the most broadly used models in image and video retrieval.
- It divides an image/video into one or more smaller tiles.
- The image represented by the concatenated BoW histograms from all the tiles.
Spatial Pyramid Matching (SPM)

- Spatial Pyramid Matching is a robust extension to spatial BoW Model.
- Combine a set of predefined partitions (1x1, 2x2, 4x4, etc.)

```
[ ] + [ ] + [ ]
```

- But, are predefined representations in SPM sufficient for multimedia retrieval?
Spatial Pyramid Matching (SPM)

• Spatial Pyramid Matching is a robust extension to spatial BoW Model.
• Combine a set of predefined partitions (1x1, 2x2, 4x4, etc.)

• But, are predefined tilings in SPM sufficient for multimedia retrieval?
Spatial Pyramid Matching (SPM)

- Spatial Pyramid Matching is a robust extension to spatial BoW Model.
- Combine a set of predefined partitions (1x1, 2x2, 4x4, etc.)

But, are predefined representations SPM sufficient for multimedia retrieval?
IBM’s Talk @ TRECVID 12

Semantic Indexing

<table>
<thead>
<tr>
<th>Global Visual Features - Spatial Granularities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Color Correlogram</td>
</tr>
<tr>
<td>Color Histogram</td>
</tr>
<tr>
<td>Color Moments</td>
</tr>
<tr>
<td>Color Wavelet</td>
</tr>
<tr>
<td>Color Wavelet Texture</td>
</tr>
<tr>
<td>Fourier Polar Pyramid</td>
</tr>
<tr>
<td>Edge Histogram</td>
</tr>
<tr>
<td>GIST</td>
</tr>
<tr>
<td>Image Stats</td>
</tr>
<tr>
<td>Image Type</td>
</tr>
<tr>
<td>LBP Histogram</td>
</tr>
<tr>
<td>maxiThumbnail Vector</td>
</tr>
<tr>
<td>miniThumbnail Vector</td>
</tr>
<tr>
<td>Siftogram</td>
</tr>
<tr>
<td>Size Vector</td>
</tr>
<tr>
<td>Thumbnail Vector</td>
</tr>
<tr>
<td>Wavelet Texture</td>
</tr>
<tr>
<td>Curvelet Texture</td>
</tr>
</tbody>
</table>

SRI Sarnoff’s Talk @TRECVID 12

Multimedia Event Detection

Feature Pooling Using Fixed Spatial Patterns

- Objective
  - Limitation: Features aggregated from a whole frame contains more irrelevant data of an event
  - Goal: Extract event relevant information by pooling features from different parts of a frame
- Spatial pooling using fixed patterns
  - Aggregate features over a set of pre-defined regions as shown at
  - Implicitly encodes location information with visual-words for better
  - Fixed patterns are easy and fast to compute

Surveillance Event Detection

- Each frame is divided into a set of rectangular tiles or grids.
- The resulting BoW features are derived by concatenating the BoW features captured in each grid.
- Encode the adjusted spatial information in BoW.

Motivation

- Spatial Representation is **fundamental** to multimedia retrieval.
  - Semantic objects/concepts indexing.
  - Multimedia event retrieval.
  - Surveillance event detection, etc.
- Different spatial representations can **affects results considerably.**
Semi-Manual Approach

• A straightforward way to find optimal representations [1,2]:
  – Manually design representation candidates.
  – Verify the candidates by running the classifier.
• Cons:
  – Require manual effort.
  – Computationally infeasible to verify all the candidates.

Motivation

• Manually designing representations is never an easy thing.
• Our goal:
  – Automatically learn salient spatial representations from data.
  – Efficient enough to run on large-scale data.
Outline

- Motivation
- Related Work
  - Jensen - Shannon Tiling
- Experiment Results
- Conclusions
Comparison with Related Work

Existing studies learn the representations with the classifiers [3,4,5].

• Reasonable Improvements.
• Time consuming.
• Low cost-effective.
• 2,000 core hours for 2% MAP (worth doing?)

Comparison with Related Work

Existing studies learn the representations with the classifiers [3,4,5].

- Reasonable Improvements.
- Time consuming.
- Low cost-effective.
- 2,000 core hours for 2% MAP (worth doing?)

JS (Jensen-Shannon) - Tiling directly captures representations at lower BoW level, independent of the classifier.

- Decent improvements.
- Orders of magnitude faster.
- High cost-effective.

BoW Distribution
Comparison with Related Work

Existing Work learn the representations with the classifiers [3,4,5].

- **Embedded** method in feature selection.

JS Tiling directly captures them at lower BoW level, independent of the classifier.

- **Filter** method in feature selection.
- **Efficiency.**
- **Generalizability.**
Proposed Approach

• **JS(Jensen-Shannon)-Tiling** offers a solution because it is:
  – Learn salient representations automatically from data.
  – Applicably to large-scale datasets.

• It is **an important component** in CMU Teams' final submission in TRECVID 2012 Multimedia Event Detection[1].
Outline

- Motivation
- Related Work
- Jensen - Shannon Tiling
- Experiment Results
- Conclusions
Problem Formulation

- A mask is a predefined partition.
  
  (a) rectangle  (b) diamond  (c) hexagon  (d) ellipse

- More representations can be derived by combining the tiles in the mask.

- Each representation is called a tiling.
A mask is a predefined partition.

More representations can be derived by combining the tiles in the mask.

Each representation is called a tiling.
Problem Formulation

- Problem: Find optimal tilings for a given mask.
- Proposed approach:
  - Systematically generate all possible tilings from the given mask.
  - Efficiently evaluate each tiling without running classifiers.
Problem Formulation

• Problem: Find optimal tilings for a given mask.

• Proposed approach:
  – Systematically generate all possible tilings from the given mask.
  – Efficiently evaluate each tiling without running classifiers.
Tiling Definition

- Tiling can be defined based on the set-partition theory.
- Divide a set as a union of non-overlapping and non-empty subsets.

\[ \{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\} \]
Tiling Definition

• Tiling can be defined based on the set-partition theory.
• Divide a set as a union of non-overlapping and non-empty subsets.

\[ \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\} \]

• A tiling can be defined as:
  – A complete partition of mask into non-overlapping area.
  – Each partition (tile) is visually adjacent[3].
Tiling Definition

• Tiling can be defined based on the set-partition theory.
• Divide a set as a union of non-overlapping and non-empty subsets.

\{1, 2, 3, 4, 5, 6, 7, 8, 9\}

• A tiling can be defined as:
  - A complete partition of mask into non-overlapping area.
  - Each partition (tile) is visually adjacent[3].

(c) Not a tiling.

identical to the connected components in the graph.
Tiling Generation

NP-hard problem. But given reasonable masks, it is solvable.

Algorithm (Loop until termination):
1) Generate a set partition candidate;
2) Test whether this candidate obeys the adjacency constraint;

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>#Set Partition</th>
<th>#Tiling</th>
<th>#Equal Tiling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangle</td>
<td>2 x 2</td>
<td>15</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Rectangle</td>
<td>3 x 3</td>
<td>21147</td>
<td>1434</td>
<td>12</td>
</tr>
<tr>
<td>Rectangle</td>
<td>4 x 4</td>
<td>10480142147</td>
<td>1691690</td>
<td>225</td>
</tr>
<tr>
<td>Diamond</td>
<td>1 x 1</td>
<td>15</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Diamond</td>
<td>2 x 2</td>
<td>52</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Diamond</td>
<td>3 x 3</td>
<td>4213597</td>
<td>17326</td>
<td>23</td>
</tr>
<tr>
<td>Hexagon</td>
<td>1</td>
<td>52</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>Hexagon</td>
<td>1.5</td>
<td>4140</td>
<td>466</td>
<td>7</td>
</tr>
<tr>
<td>Ellipse</td>
<td>4</td>
<td>4140</td>
<td>344</td>
<td>5</td>
</tr>
<tr>
<td>Ellipse</td>
<td>8</td>
<td>4213597</td>
<td>5504</td>
<td>10</td>
</tr>
</tbody>
</table>

- Visual adjacency constraint **significantly reduces** the number of candidates.
Tiling Generation

NP-hard problem. But given reasonable masks, it is solvable.

Algorithm (Loop until termination):
1) Generate a set partition candidate;
2) Test whether this candidate obeys the adjacency constraint;

- Visual adjacency constraint significantly reduces the number of candidates.
Problem Formulation

- Problem: Find optimal tilings for a given mask.
- Proposed approach:
  - Systematically generate all possible tilings from the given mask.
  - **Efficiently evaluate each tiling without running classifiers.**
Tiling Evaluation

- Intuitively an optimal tiling would separate the positive and negative samples with the maximum distance.
- The distance is evaluated w.r.t Kullback-Leibler (KL) divergence.
- Symmetric version called Jensen-Shannon (JS) divergence.

$$\text{cost}(\mathcal{T}_\kappa) = \lambda |\mathcal{T}_\kappa(S)|^{-1} \sum_{i=0}^{\mathcal{T}_\kappa(S)} \frac{JS(D_i^+ \parallel D_i^-)}{|\mathcal{T}_\kappa(S)|}$$

- $\mathcal{T}_\kappa(S)$ is the tiling to evaluate.
- $D_i^+$ and $D_i^-$ average word distributions of positive and negative samples generated by the tiling.
Tiling Evaluation

• Consistent with the distribution separability principle in [6].

Tiling Evaluation

- Consistent with the distribution separability principle in [6].
- We prove that the negative JS divergence is approximately an upper bound of the training error of a weighted K-Nearest Neighbor classifier $K = N$.
- Justify why the computationally inexpensive divergence can be a proxy to the computationally expensive classifier.

\[
\text{Minimize } -\text{JS Complexity } O(N) \quad \text{Minimize } \text{KNN} (K=N) \quad \text{Complexity } O(N^2)
\]
Outline

- Motivation
- Related Work
- Jensen - Shannon Tiling
- Experiment Results
- Conclusions
Comparison with state-of-the-art

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>MAP</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-Scene</td>
<td>SPM [12]</td>
<td>83.5±0.5</td>
<td>80.8±0.6</td>
</tr>
<tr>
<td></td>
<td>Boureau et al. [2]</td>
<td>-</td>
<td>84.9±0.3</td>
</tr>
<tr>
<td></td>
<td>Sharma et al. [19]</td>
<td>85.5±0.7</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>van Gemert et al. [23]</td>
<td>-</td>
<td>76.7±0.4</td>
</tr>
<tr>
<td></td>
<td>Sharma et al. [18]</td>
<td>-</td>
<td>81.2±0.6</td>
</tr>
<tr>
<td></td>
<td>Yang et al. [27]</td>
<td>-</td>
<td>80.3±0.9</td>
</tr>
<tr>
<td></td>
<td>JS Tiling</td>
<td>88.0±0.3</td>
<td>85.3±0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>MAP</th>
<th>Min DCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SED</td>
<td>SPM [12]</td>
<td>22.8±1.0</td>
<td>89.0±1.5</td>
</tr>
<tr>
<td></td>
<td>Winner’11 [30]</td>
<td>23.8±0.8</td>
<td>87.2±1.0</td>
</tr>
<tr>
<td></td>
<td>JS Tiling</td>
<td>26.5±0.6</td>
<td>85.1±0.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>MAP(SIFT)</th>
<th>MAP(STIP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>SPM [12]</td>
<td>26.8</td>
<td>17.2</td>
</tr>
<tr>
<td></td>
<td>Winner’12 [29, 21]</td>
<td>27.3</td>
<td>18.7</td>
</tr>
<tr>
<td></td>
<td>JS Tiling</td>
<td>30.7</td>
<td>21.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOC</td>
<td>SPM [12]</td>
<td>52.5</td>
</tr>
<tr>
<td></td>
<td>Winner’07 [15]</td>
<td>54.2</td>
</tr>
<tr>
<td></td>
<td>Wang et al. [26]</td>
<td>55.1</td>
</tr>
<tr>
<td></td>
<td>Yang et al. [28]</td>
<td>59.6</td>
</tr>
<tr>
<td></td>
<td>JS Tiling</td>
<td>55.5</td>
</tr>
</tbody>
</table>

- **Consistently outperforms the SPM** across datasets on scene/object recognition and event detection.
- **Comparable or even better** results with existing methods.
Reasons for the Improvement

- 1) Capture more **salient spatial representations** than SPM.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Predefined Masks</th>
<th>Rectangle Masks</th>
<th>All Masks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tiling</td>
<td>Tiling</td>
<td>Tiling</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>Accuracy</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>MAP</td>
<td>MAP</td>
</tr>
<tr>
<td>1</td>
<td>79.5±0.7</td>
<td>80.4±0.7</td>
<td>82.4±0.4</td>
</tr>
<tr>
<td>2</td>
<td>79.4±0.6</td>
<td>80.4±0.4</td>
<td>81.4±0.4</td>
</tr>
<tr>
<td>3</td>
<td>78.6±0.4</td>
<td>80.0±0.6</td>
<td>80.8±0.5</td>
</tr>
<tr>
<td>4</td>
<td>77.5±0.2</td>
<td>79.9±0.5</td>
<td>80.9±0.3</td>
</tr>
<tr>
<td>5</td>
<td>77.8±0.5</td>
<td>79.5±0.7</td>
<td>80.4±0.7</td>
</tr>
</tbody>
</table>

Predefined tilings in SPM

Proposed Method

The results are on 15 scene category dataset.
Reasons for the Improvement

• 1) Capture more salient spatial representations than SPM.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Predefined Masks</th>
<th>Rectangle Masks</th>
<th>All Masks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tiling</td>
<td>Tiling</td>
<td>Tiling</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>Accuracy</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>MAP</td>
<td>MAP</td>
</tr>
<tr>
<td>1</td>
<td>79.5±0.7</td>
<td>80.4±0.7</td>
<td>82.4±0.4</td>
</tr>
<tr>
<td>2</td>
<td>79.4±0.6</td>
<td>80.4±0.4</td>
<td>81.4±0.4</td>
</tr>
<tr>
<td>3</td>
<td>78.6±0.4</td>
<td>80.0±0.6</td>
<td>80.8±0.5</td>
</tr>
<tr>
<td>4</td>
<td>77.5±0.2</td>
<td>79.9±0.5</td>
<td>80.9±0.3</td>
</tr>
<tr>
<td>5</td>
<td>77.8±0.5</td>
<td>79.5±0.7</td>
<td>80.4±0.7</td>
</tr>
</tbody>
</table>

• 2) Substantially \textbf{augment the choices of representations}.

<table>
<thead>
<tr>
<th>L</th>
<th>Spatial Pyramid</th>
<th>Rectangle Masks</th>
<th>All Masks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Accuracy</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>MAP</td>
<td>MAP</td>
</tr>
<tr>
<td>0</td>
<td>75.3±0.3</td>
<td>80.4±0.7</td>
<td>82.4±0.4</td>
</tr>
<tr>
<td>1</td>
<td>80.7±0.6</td>
<td>80.8±0.5</td>
<td>82.2±0.5</td>
</tr>
<tr>
<td>2</td>
<td>\textbf{80.8±0.6}</td>
<td>\textbf{81.4±0.6}</td>
<td>\textbf{82.7±0.6}</td>
</tr>
<tr>
<td>3</td>
<td>80.1±0.6</td>
<td>81.5±0.6</td>
<td>82.8±0.5</td>
</tr>
<tr>
<td>4</td>
<td>79.2±0.6</td>
<td>81.7±0.6</td>
<td>83.5±0.7</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>81.9±0.5</td>
<td>85.3±0.4</td>
</tr>
</tbody>
</table>

The results are on 15 scene category dataset.
Learned Tiling on SED dataset

- Heat maps are plotted based on manual annotations.
- Tilings are learned without using annotations.
- Learned tilings are more sensible than predefined tilings.
Runtime Comparison

- Compare the runtime with tiling selection by running classifiers.
- Search a space of 1,434 tilings.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>JS Tiling</th>
<th>Linear SVM</th>
<th>Kernel SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-scene</td>
<td>1.1(h)</td>
<td>1,314(h)</td>
<td>10,874(h)</td>
</tr>
<tr>
<td>SED</td>
<td>2.1(h)</td>
<td>2,629(h)</td>
<td>32,862(h)</td>
</tr>
<tr>
<td>MED</td>
<td>2.3(h)</td>
<td>4,541(h)</td>
<td>41,825(h)</td>
</tr>
<tr>
<td>Pascal VOC</td>
<td>1.6(h)</td>
<td>1,912(h)</td>
<td>22,346(h)</td>
</tr>
</tbody>
</table>

- A single core Intel Core i7 CPU@2.8GHz with 4G memory.
- **Orders of magnitude faster** than running classifiers.
- Substantiate the theoretical complexity analysis.
Outline

- Motivation
- Related Work
- Jensen - Shannon Tiling
- Experiment Results
- Conclusions
Summary

• A few messages to take away from this talk:
  – JS Tiling provides a efficient solution to automatically learn salient BoW representations for large-scale datasets.
  – JS Tiling consistently outperforms the spatial pyramid matching across datasets. Comparable or even better performance with existing methods.
Beyond BoW representation

• Tokyo TechCanon’s Talk @TRECVID 2012


• AXES’s Talk @TRECVID 2013

- Spatial Fisher vector (SFV)
  - (Krapac et al., ICCV, 2011)
  - encodes first and second moments of visual word locations
  - adds 6 entries for each visual word: \( \mu \) and \( \sigma \) for \((x, y, t)\) coordinates.

- Compared to spatial pyramids:
  - (Oneață et al., ICCV, 2013)
  - similar performance gain
  - SFV are more compact

Beyond spatial representation

- Temporal tiling
  - Determine optimal sliding window sizes.
Aspects to be Improved

• The tilings learned from different masks are not directly comparable. A practical trick:
  – Start with a number of masks.
  – Use JS-Tiling to find a couple of salient tilings from the huge search space.
  – Run classifiers on these tilings on the validation dataset, and fuse promising ones to obtain better performance.

• Sampling bias for small tiles (overestimate the distance).
  – Equal tiling can avoid this bias.
  – Study the smoothing function.
Acknowledgement

This work was partially supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.
THANK YOU.

Q&A?