Aggregating local image descriptors for large-scale retrieval and classification

Cordelia Schmid
LEAR – INRIA Grenoble
Aggregating local descriptors

• Set of n local descriptors $\rightarrow$ 1 vector

• Popular approach: bag of features, often with SIFT features

• Recently improved aggregation schemes
  – Fisher vector [Perronnin & Dance ‘07]
  – VLAD descriptor [Jegou, Douze, Schmid, Perez ‘10]
  – Supervector [Zhou et al. ‘10]
  – Sparse coding [Wang et al. ’10, Boureau et al.’10]

• Use in very large-scale retrieval and classification
Towards large-scale image search

- Each image is represented by one vector
  - Bag-of-features, Fisher vector, GIST

- Vector compression to reduce storage requirement and search time
Aggregation of local descriptors

- Most popular approach: BoF representation [Sivic & Zisserman 03]
  - sparse vector
  - highly dimensional
  $\rightarrow$ significant dimensionality reduction introduces loss

- Vector of locally aggregated descriptors (VLAD) [Jegou et al. 10]
  - non sparse vector
  - fast to compute
  - excellent results with a small vector dimensionality

- Fisher vector [Perronnin & Dance 07]
  - probabilistic version of VLAD
  - initially used for image classification
  - comparable or improved performance over VLAD for image retrieval
VLAD: vector of locally aggregated descriptors

- Learn a vector quantifier ($k$-means): $c_1, \ldots, c_i, \ldots c_k$, with $c_i$ centroid of dim. $d$

- For a given image
  - assign each descriptor to closest center $c_i$
  - accumulate (sum) descriptors per cell
    $$v_i := v_i + (x_j - c_i)$$
    measure repartition of vectors within a cell

- VLAD of dimension $D = k \times d$
  (k typically between 16 and 256)

- The vector is square-root + L2-normalized

[Jegou, Douze, Schmid, Perez, CVPR'10]
Fisher vector

- Use a Gaussian Mixture Model as vocabulary
- Statistical measure of the descriptors of the image w.r.t the GMM
- Derivative of likelihood w.r.t. GMM parameters

GMM parameters:
- $w_i$ weight
- $\mu_i$ mean
- $\sigma_i$ co-variance (diagonal)

Translated cluster $\rightarrow$
large derivative on $\mu_i$ for this component

[Perronnin & Dance 07]
Fisher vector

FV formulas:

\[
G_{\mu,i}^X = \frac{1}{T \sqrt{w_i}} \sum_{t=1}^{T} \gamma_t(i) \left( \frac{x_t - \mu_i}{\sigma_i} \right)
\]

\[
G_{\sigma,i}^X = \frac{1}{T \sqrt{2w_i}} \sum_{t=1}^{T} \gamma_t(i) \left[ \frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right]
\]

\(\gamma_t(i)\) = soft-assignment of patch \(x_t\) to Gaussian \(i\)

Fisher Vector = concatenation of per-Gaussian gradient vectors

For image retrieval in our experiments:
- only deviation wrt mean, dim: \(K \times D\) [\(K\) number of Gaussians, \(D\) dim of descriptor]
- variance does not improve for comparable vector length
We compare Fisher, VLAD and BoF on INRIA Holidays Dataset (mAP %)

Holidays Dataset
- 500 query images + 991 annotated true positives
- most images are holiday photos of friends and family
- 1 million & 10 million distractor images from Flickr
- Vocabulary construction on a different Flickr set
- Evaluation metric: mean average precision (in [0,1], bigger = better)
**VLAD/Fisher/BOF performance and dimensionality reduction**

- We compare Fisher, VLAD and BoF on INRIA Holidays Dataset (mAP %)
- Dimension is reduced to $D'$ dimensions with PCA

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>$K$</th>
<th>$D$</th>
<th>$D' = D$</th>
<th>$D'=2048$</th>
<th>$D'=512$</th>
<th>$D'=128$</th>
<th>$D'=64$</th>
<th>$D'=32$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOW</td>
<td>1000</td>
<td>1000</td>
<td>40.1</td>
<td>43.5</td>
<td>44.4</td>
<td>43.4</td>
<td>40.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20000</td>
<td>20000</td>
<td>43.7</td>
<td>44.9</td>
<td>45.2</td>
<td>44.4</td>
<td>41.8</td>
<td></td>
</tr>
<tr>
<td>Fisher ($\mu$)</td>
<td>16</td>
<td>1024</td>
<td>54.0</td>
<td>54.6</td>
<td>52.3</td>
<td>49.9</td>
<td>46.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>4096</td>
<td>59.5</td>
<td>61.0</td>
<td>56.5</td>
<td>52.0</td>
<td>48.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>16384</td>
<td>62.5</td>
<td>57.0</td>
<td>53.8</td>
<td>50.6</td>
<td>48.6</td>
<td></td>
</tr>
<tr>
<td>VLAD</td>
<td>16</td>
<td>1024</td>
<td>52.0</td>
<td>52.7</td>
<td>52.6</td>
<td>50.5</td>
<td>47.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>4096</td>
<td>55.6</td>
<td>59.8</td>
<td>55.7</td>
<td>52.3</td>
<td>48.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>16384</td>
<td>58.7</td>
<td>56.7</td>
<td>54.2</td>
<td>51.3</td>
<td>48.1</td>
<td></td>
</tr>
<tr>
<td>GIST</td>
<td>960</td>
<td></td>
<td>36.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Observations:**
- Fisher, VLAD better than BoF for a given descriptor size
- Choose a small $D$ if output dimension $D'$ is small
- Performance of GIST not competitive

[Jegou, Perronnin, Douze, Sanchez, Perez, Schmid, PAMI'12]
Compact image representation

- Aim: improving the tradeoff between
  - search speed
  - memory usage
  - search quality

- Approach: joint optimization of three stages
  - local descriptor aggregation
  - dimension reduction
  - indexing algorithm

Image representation
- VLAD / Fisher

PCD + PQ codes

(Non) – exhaustive search
Product quantization for nearest neighbor search

- Vector split into \( m \) subvectors: \( y \rightarrow [y_1 | \cdots | y_m] \)

- Subvectors are quantized separately by quantizers \( q(y) = [q_1(y_1)| \cdots | q_m(y_m)] \)
  where each \( q_i \) is learned by \( k \)-means with a limited number of centroids

- Example: \( y = 128 \)-dim vector split in 8 subvectors of dimension 16
  - each subvector is quantized with 256 centroids \( \rightarrow \) 8 bit
  - very large codebook \( 256^8 \approx 1.8 \times 10^{19} \)

\[ \text{16 components} \]
\[ \begin{array}{cccccccc}
  y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\
\end{array} \]

\[ \text{256 centroids} \]
\[ q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6 \quad q_7 \quad q_8 \]

\[ \text{8 bits} \]
\[ q_1(y_1) \quad q_2(y_2) \quad q_3(y_3) \quad q_4(y_4) \quad q_5(y_5) \quad q_6(y_6) \quad q_7(y_7) \quad q_8(y_8) \]

\( \Rightarrow 8 \) subvectors \( \times 8 \) bits = 64-bit quantization index

[Jegou, Douze, Schmid, PAMI’11]
Optimizing the dimension reduction and quantization together

- Fisher vectors undergoes two approximations
  - mean square error from PCA projection
  - mean square error from quantization
- Given k and bytes/image, choose D’ minimizing their sum

Results on Holidays dataset:
- there exists an optimal D’
- 16 byte best results for k=64
- 320 byte best results for k=256
Joint optimization of Fisher and dimension reduction-indexing

- For Fisher
  - The larger $k$, the better the raw search performance
  - But large $k$ produce large vectors, that are harder to index

- Optimization of the vocabulary size
  - Fixed output size (in bytes)
  - $D'$ computed from $k$ via the joint optimization of reduction/indexing
  - Only $k$ has to be set

→ end-to-end parameter optimization
Results on the Holidays dataset with various quantization parameters

![Graph showing the relationship between ADC parameters and mAP for different miniBOF configurations. The x-axis represents the number of bytes, and the y-axis represents mAP. Different markers correspond to different miniBOF configurations with varying values of K.](image)
Comparison to the state of the art

Datasets:
- INRIA Holidays dataset, score: mAP (%)
- University of Kentucky benchmark (UKB)
  - 10200 images, 4 images per objects
  - score: number of relevant images retrieved in the first 4 positions, max 4
Comparison to the state of the art

<table>
<thead>
<tr>
<th>Method</th>
<th>Method: $K=20,000$</th>
<th>Method: $K=200,000$</th>
<th>miniBOF [12]</th>
<th>FV $K=64$, spectral hashing 128 bits</th>
<th>VLAD, $K=16$, ADC 16x8 [23]</th>
<th>VLAD, $K=64$, ADC 32x10 [23]</th>
<th>FV $K=8$, binarized [22]</th>
<th>FV $K=64$, binarized [22]</th>
<th>FV $K=64$, ADC 16x8 ($D'=96$)</th>
<th>FV $K=256$, ADC 256x10 ($D'=2048$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bytes</td>
<td>UKB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>BOW</td>
<td>10364</td>
<td>2.87</td>
<td>43.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>12886</td>
<td>2.81</td>
<td>54.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td>miniBOF</td>
<td>20</td>
<td>2.07</td>
<td>25.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>2.72</td>
<td>40.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>2.83</td>
<td>42.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td>FV</td>
<td>16</td>
<td>2.57</td>
<td>39.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>3.10</td>
<td>49.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>2.79</td>
<td>46.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>520</td>
<td>3.21</td>
<td>57.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td><strong>3.10</strong></td>
<td><strong>50.6</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td><strong>3.47</strong></td>
<td><strong>63.4</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.6</td>
<td>63.4</td>
</tr>
</tbody>
</table>

Large scale experiments (10 million images)

- With the product quantizer
  - Exhaustive search with ADC: 0.29s
  - Non-exhaustive search with IVFADC: 0.014s

IVFADC -- Combination with an inverted file
Large scale experiments (10 million images)

Timings
IVFADC: 0.02s
Conclusion

- Competitive search accuracy with a few dozen bytes per indexed image

- Tested on 220 million video frames
  - extrapolation for 1 billion images: 20GB RAM, query < 1s on 8 cores

- Code on-line available Software for Fisher computation and PQ-codes
  - http://lear.inrialpes.fr/software
Image classification

- Image classification: assigning a class label to the image

- Car: present
- Cow: present
- Bike: not present
- Horse: not present
...
Image classification

- Image classification: assigning a class label to the image

- Object localization: define the location and the category

Car: present
Cow: present
Bike: not present
Horse: not present...

Location
Category
Difficulties: within object variations

Variability: Camera position, Illumination, Internal parameters

Within-object variations
Difficulties: within class variations
Image classification

- **Given**
  Positive training images containing an object class

  ![Motorcycle](image1.png) ![Motorcycle](image2.png) ![Motorcycle](image3.png)

  Negative training images that don’t

  ![Grass](image4.png) ![Airplane](image5.png) ![Office](image6.png)

- **Classify**
  A test image as to whether it contains the object class or not

  ![Motorcycle](image7.png)
Bag-of-features – Origin: texture recognition

- Texture is characterized by the repetition of basic elements or *textons*

Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001
Bag-of-features – Origin: texture recognition

Universal texton dictionary

histogram
Bag-of-features – Origin: bag-of-words (text)

- Orderless document representation: frequencies of words from a dictionary
- Classification to determine document categories

Bag-of-words

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>People</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Sculpture</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Bag-of-features for image classification

[Csurka et al., ECCV Workshop’04], [Nowak, Jurie & Triggs, ECCV’06],
[Zhang, Marszalek, Lazebnik & Schmid, IJCV’07]
Bag-of-features for image classification

**Step 1**
- Extract regions
- Compute descriptors

**Step 2**
- Find clusters and frequencies
- Compute distance matrix

**Step 3**
- Classification

\[ d(S_i, S_j) \]
Step 1: feature extraction

- Scale-invariant image regions + SIFT (see previous lecture)
  - Affine invariant regions give “too” much invariance
  - Rotation invariance for many realistic collections “too” much invariance

- Dense descriptors
  - Improve results in the context of categories (for most categories)
  - Interest points do not necessarily capture “all” features

- Color-based descriptors

- Shape-based descriptors
Dense features

- Multi-scale dense grid: extraction of small overlapping patches at multiple scales
- Computation of the SIFT descriptor for each grid cells
- Exp.: Horizontal/vertical step size 3-6 pixel, scaling factor of 1.2 per level
Bag-of-features for image classification

Step 1
Extract regions

Step 2
Compute descriptors
Find clusters and frequencies

Step 3
Compute distance matrix
Classification

SVM
Step 2: Quantization

Visual vocabulary

Clustering
### Examples for visual words

<table>
<thead>
<tr>
<th>Category</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplanes</td>
<td><img src="image1" alt="Airplanes Images" /> <img src="image2" alt="Airplanes Images" /> <img src="image3" alt="Airplanes Images" /> <img src="image4" alt="Airplanes Images" /></td>
</tr>
<tr>
<td>Motorbikes</td>
<td><img src="image5" alt="Motorbikes Images" /> <img src="image6" alt="Motorbikes Images" /> <img src="image7" alt="Motorbikes Images" /> <img src="image8" alt="Motorbikes Images" /></td>
</tr>
<tr>
<td>Faces</td>
<td><img src="image9" alt="Faces Images" /> <img src="image10" alt="Faces Images" /> <img src="image11" alt="Faces Images" /> <img src="image12" alt="Faces Images" /></td>
</tr>
<tr>
<td>Wild Cats</td>
<td><img src="image13" alt="Wild Cats Images" /> <img src="image14" alt="Wild Cats Images" /> <img src="image15" alt="Wild Cats Images" /> <img src="image16" alt="Wild Cats Images" /></td>
</tr>
<tr>
<td>Leaves</td>
<td><img src="image17" alt="Leaves Images" /> <img src="image18" alt="Leaves Images" /> <img src="image19" alt="Leaves Images" /> <img src="image20" alt="Leaves Images" /></td>
</tr>
<tr>
<td>People</td>
<td><img src="image21" alt="People Images" /> <img src="image22" alt="People Images" /> <img src="image23" alt="People Images" /> <img src="image24" alt="People Images" /></td>
</tr>
<tr>
<td>Bikes</td>
<td><img src="image25" alt="Bikes Images" /> <img src="image26" alt="Bikes Images" /> <img src="image27" alt="Bikes Images" /> <img src="image28" alt="Bikes Images" /></td>
</tr>
</tbody>
</table>
Step 2: Quantization

- Cluster descriptors
  - K-means
  - Gaussian mixture model

- Assign each visual word to a cluster
  - Hard or soft assignment

- Build frequency histogram
Image representation

- each image is represented by a vector, typically 1000-4000 dimension, normalization with L1/L2 norm
- fine grained – represent model instances
- coarse grained – represent object categories
Bag-of-features for image classification

- Extract regions
- Compute descriptors
- Find clusters and frequencies

Step 1

Step 2

Step 3

Compute distance matrix
Classification

SVM
Step 3: Classification

- Learn a decision rule (classifier) assigning bag-of-features representations of images to different classes
Training data

Vectors are histograms, one from each training image

positive

negative

Train classifier, e.g. SVM
Kernels for bags of features

- Histogram intersection kernel: \( I(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i)) \)

- Generalized Gaussian kernel:
  \[
  K(h_1, h_2) = \exp\left( -\frac{1}{A} D(h_1, h_2)^2 \right)
  \]

  - \( D \) can be Euclidean distance \( \rightarrow \) RBF kernel

  - \( D \) can be \( \chi^2 \) distance
    \[
    D(h_1, h_2) = \sum_{i=1}^{N} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)}
    \]

- Earth mover’s distance
Combining features

• SVM with multi-channel chi-square kernel

\[ K(H_i, H_j) = \exp \left( - \sum_{c \in \mathcal{C}} \frac{1}{A_c} D_c(H_i, H_j) \right) \]

- Channel \( c \) is a combination of detector, descriptor
- \( D_c(H_i, H_j) \) is the chi-square distance between histograms
  \[ D_c(H_1, H_2) = \frac{1}{2} \sum_{i=1}^{m} \left[ \frac{(h_{1i} - h_{2i})^2}{(h_{1i} + h_{2i})} \right] \]
  - \( A_c \) is the mean value of the distances between all training sample
- Extension: learning of the weights, for example with Multiple Kernel Learning (MKL)

Multi-class SVMs

• Various direct formulations exist, but they are not widely used in practice. It is more common to obtain multi-class SVMs by combining two-class SVMs in various ways.

• One versus all:
  – Training: learn an SVM for each class versus the others
  – Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value

• One versus one:
  – Training: learn an SVM for each pair of classes
  – Testing: each learned SVM “votes” for a class to assign to the test example
Why does SVM learning work?

- Learns foreground and background visual words

  foreground words – high weight

  background words – low weight
Localization according to visual word probability

Illustration

foreground word more probable

background word more probable
Illustration

A linear SVM trained from positive and negative window descriptors

A few of the highest weighted descriptor vector dimensions (= 'PAS + tile')

+ lie on object boundary (= local shape structures common to many training exemplars)
Bag-of-features for image classification

- Excellent results in the presence of background clutter
Examples for misclassified images

Books- misclassified into faces, faces, buildings

Buildings- misclassified into faces, trees, trees

Cars- misclassified into buildings, phones, phones
Bag of visual words summary

• Advantages:
  – largely unaffected by position and orientation of object in image
  – fixed length vector irrespective of number of detections
  – very successful in classifying images according to the objects they contain

• Disadvantages:
  – no explicit use of configuration of visual word positions
  – poor at localizing objects within an image
Evaluation of image classification

- PASCAL VOC [05-10] datasets

- PASCAL VOC 2007
  - Training and test dataset available
  - Used to report state-of-the-art results
  - Collected January 2007 from Flickr
  - 500,000 images downloaded and random subset selected
  - 20 classes
  - Class labels per image + bounding boxes
  - 5011 training images, 4952 test images

- Evaluation measure: average precision
PASCAL 2007 dataset
PASCAL 2007 dataset

Dining Table  Dog  Horse  Motorbike  Person

Potted Plant  Sheep  Sofa  Train  TV/Monitor
Evaluation

- **Average Precision [TREC]** averages precision over the entire range of recall
  - Curve interpolated to reduce influence of “outliers”

- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall
Results for PASCAL 2007

• Winner of PASCAL 2007 [Marszalek et al.] : mAP 59.4
  – Combination of several different channels (dense + interest points, SIFT + color descriptors, spatial grids)
  – Non-linear SVM with Gaussian kernel

• Multiple kernel learning [Yang et al. 2009] : mAP 62.2
  – Combination of several features
  – Group-based MKL approach

• Combining object localization and classification [Harzallah et al.’09] : mAP 63.5
  – Use detection results to improve classification

• .....
Spatial pyramid matching

- Add spatial information to the bag-of-features
- Perform matching in 2D image space

[Lazebnik, Schmid & Ponce, CVPR 2006]
Related work

Similar approaches:

Subblock description [Szummer & Picard, 1997]
SIFT [Lowe, 1999]
GIST [Torralba et al., 2003]
Locally orderless representation at several levels of spatial resolution
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution

level 0

level 1
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution

level 0

level 1

level 2
Spatial pyramid matching

- Combination of spatial levels with pyramid match kernel [Grauman & Darell’05]
- Intersect histograms, more weight to finer grids
Scene dataset [Labzenik et al.’06]

Coast  Forest  Mountain  Open country  Highway  Inside city  Tall building  Street

Suburb  Bedroom  Kitchen  Living room  Office

Store  Industrial

4385 images
15 categories
Scene classification

<table>
<thead>
<tr>
<th>L</th>
<th>Single-level</th>
<th>Pyramid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0(1x1)</td>
<td>72.2±0.6</td>
<td></td>
</tr>
<tr>
<td>1(2x2)</td>
<td>77.9±0.6</td>
<td>79.0 ±0.5</td>
</tr>
<tr>
<td>2(4x4)</td>
<td>79.4±0.3</td>
<td>81.1 ±0.3</td>
</tr>
<tr>
<td>3(8x8)</td>
<td>77.2±0.4</td>
<td>80.7 ±0.3</td>
</tr>
</tbody>
</table>
Category classification – CalTech101

<table>
<thead>
<tr>
<th>L</th>
<th>Single-level</th>
<th>Pyramid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0(1x1)</td>
<td>41.2±1.2</td>
<td></td>
</tr>
<tr>
<td>1(2x2)</td>
<td>55.9±0.9</td>
<td>57.0 ±0.8</td>
</tr>
<tr>
<td>2(4x4)</td>
<td>63.6±0.9</td>
<td>64.6 ±0.8</td>
</tr>
<tr>
<td>3(8x8)</td>
<td>60.3±0.9</td>
<td>64.6 ±0.7</td>
</tr>
</tbody>
</table>
Discussion

• Summary
  – Spatial pyramid representation: appearance of local image patches + coarse global position information
  – Substantial improvement over bag of features
  – Depends on the similarity of image layout

• Extensions
  – Flexible, object-centered grid
Large-scale image classification

- Image classification: assigning a class label to the image

  ![Image classification example]

  - Car: present
  - Cow: present
  - Bike: not present
  - Horse: not present
  ...

- What makes it large-scale?
  - number of images
  - number of classes
  - dimensionality of descriptor

  **IMAGENET** has 14M images from 22k classes
Large-scale image classification

• Image descriptors
  – Fisher vector (high dimensional)
  – Normalization: square-rooting or latent MOG+ L2 normalization
    [Image categorization using Fisher kernels of non-iid image models, Cinbis, Verbeek, Schmid, CVPR’12] [Perronnin’10]

• Classification approach
  – Linear classifiers
  – One versus rest classifier
  – Stochastic gradient descent optimization
    [Towards good practice in large-scale learning for image classification, Perronnin, Akata, Harchaoui, Schmid, CVPR’12]
Evaluation image description

- Comparing on PASCAL VOC’07 linear classifiers with
  - Fisher vector
  - Sqrt transformation of Fisher vector
  - Latent GMM of Fisher vector

- Sqrt transform + latent MOG models lead to improvement

- State-of-the-art performance obtained with linear classifier
Evaluation image description

Fisher versus BOF vector + linear classifier on Pascal Voc’07

• Fisher improves over BOF
• Fisher comparable to BOF + non-linear classifier
• Limited gain due to SPM on PASCAL
• Sqrt helps for Fisher and BOF
• [Chatfield et al. 2011]

<table>
<thead>
<tr>
<th>SPM</th>
<th>Method</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>BoW</td>
<td>20.1</td>
<td>29.0</td>
<td>36.2</td>
<td>40.7</td>
<td>44.1</td>
</tr>
<tr>
<td>No</td>
<td>SqrtBoW</td>
<td>21.0</td>
<td>29.5</td>
<td>37.4</td>
<td>41.3</td>
<td>46.1</td>
</tr>
<tr>
<td>No</td>
<td>LatBoW</td>
<td>22.9</td>
<td>30.1</td>
<td>38.9</td>
<td>41.2</td>
<td>44.5</td>
</tr>
<tr>
<td>Yes</td>
<td>BoW</td>
<td>37.1</td>
<td>40.1</td>
<td>42.4</td>
<td>46.4</td>
<td>48.9</td>
</tr>
<tr>
<td>Yes</td>
<td>SqrtBoW</td>
<td>37.8</td>
<td>41.2</td>
<td>44.6</td>
<td>47.8</td>
<td>51.6</td>
</tr>
<tr>
<td>Yes</td>
<td>LatBoW</td>
<td>39.3</td>
<td>41.7</td>
<td>45.3</td>
<td>48.7</td>
<td>52.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPM</th>
<th>Method</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>MoG</td>
<td>49.2</td>
<td>51.5</td>
<td>53.0</td>
<td>54.4</td>
<td>55.0</td>
<td>55.9</td>
</tr>
<tr>
<td>No</td>
<td>SqrtMoG</td>
<td>51.9</td>
<td>54.7</td>
<td>56.2</td>
<td>58.2</td>
<td>58.8</td>
<td>60.2</td>
</tr>
<tr>
<td>No</td>
<td>LatMoG</td>
<td>52.3</td>
<td>55.3</td>
<td>56.5</td>
<td>58.6</td>
<td>59.5</td>
<td>60.3</td>
</tr>
<tr>
<td>Yes</td>
<td>MoG</td>
<td>53.2</td>
<td>55.4</td>
<td>56.2</td>
<td>57.0</td>
<td>57.3</td>
<td>57.6</td>
</tr>
<tr>
<td>Yes</td>
<td>SqrtMoG</td>
<td>56.1</td>
<td>57.7</td>
<td>58.9</td>
<td>60.4</td>
<td>60.5</td>
<td>60.8</td>
</tr>
<tr>
<td>Yes</td>
<td>LatMoG</td>
<td>57.3</td>
<td>58.8</td>
<td>59.4</td>
<td>60.4</td>
<td>60.6</td>
<td>60.7</td>
</tr>
</tbody>
</table>
Large-scale image classification

- **Classification approach**
  - One-versus-rest classifiers
  - stochastic gradient descent (SGD)
  - At each step choose a sample at random and update the parameters using a sample-wise estimate of the regularized risk

- **Data reweighting**
  - When some classes are significantly more populated than others, rebalancing positive and negative examples
  - Empirical risk with reweighting

\[
\frac{\rho}{N_+} \sum_{i \in I_+} L_{OVR}(x_i, y_i; w) + \frac{1 - \rho}{N_-} \sum_{i \in I_-} L_{OVR}(x_i, y_i; w)
\]

\[
\rho = \frac{1}{2} \quad \text{Natural rebalancing, same weight to positive and negatives}
\]
Experimental results

• Datasets
  – ImageNet Large Scale Visual Recognition Challenge 2010 (ILSVRC)
    • 1000 classes and 1.4M images
  – ImageNet10K dataset
    • 10184 classes and ~ 9 M images
Experimental results

• Features: dense SIFT, reduced to 64 dim with PCA

• Fisher vectors
  – 256 Gaussians, using mean and variance
  – Spatial pyramid with 4 regions
  – Approx. 130K dimensions (4x [2x64x256])
  – Normalization: square-rooting and L2 norm

• BOF: dim 1024 + R=4
  – 4960 dimensions
  – Normalization: square-rooting and L2 norm
Importance of re-weighting

- Plain lines correspond to w-OVR, dashed one to u-OVR
- $\beta$ is number of negatives samples for each positive, $\beta=1$ natural rebalancing
- Results for ILSVRC 2010

- Significant impact on accuracy
- For very high dimensions little impact
One-versus-rest works

- Different classification methods
- 256 Gaussian Fisher vector + SP with R=4 (dim 130k)
- BOF dim=1024 + SP with R=4 (dim 4000)
- Results for ILSVRC 2010

<table>
<thead>
<tr>
<th>Top-1</th>
<th>BOV</th>
<th>FV</th>
<th>w-OVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>26.4</td>
<td>45.7</td>
<td></td>
</tr>
</tbody>
</table>
Impact of the image signature size

- Fisher vector (no SP) for varying number of Gaussians + different classification methods, ILSVRC 2010

- Performance improves for higher dimensional vectors
Large-scale experiment on ImageNet10k

<table>
<thead>
<tr>
<th></th>
<th>u-OVR</th>
<th>w-OVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOV 4K-dim</td>
<td>3.8</td>
<td>7.5</td>
</tr>
<tr>
<td>FV 130K-dim</td>
<td>16.7</td>
<td>19.1</td>
</tr>
</tbody>
</table>

- Significant gain by data re-weighting, even for high-dimensional Fisher vectors
- $w$-OVR $>$ $u$-OVR
- Improves over state of the art: 6.4% [Deng et. al] and WAR [Weston et al.]
Large-scale experiment on ImageNet10k

- Illustration of results obtained with w-OVR and 130K-dim Fisher vectors, ImageNet10K top-1 accuracy
Conclusion

- **Stochastic training**: learning with SGD is well-suited for large-scale datasets

- **One-versus-rest**: a flexible option for large-scale image classification

- **Class imbalance**: optimize the imbalance parameter in one-versus-rest strategy is a must for competitive performance
Conclusion

- State-of-the-art performance for large-scale image classification

- Code on-line available at http://lear.inrialpes.fr/software

- Future work
  - Beyond a single representation of the entire image
  - Take into account the hierarchical structure