Bag-of-features for category classification

Cordelia Schmid
Category recognition

- Image classification: assigning a class label to the image

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

- Image classification: assigning a class label to the image
  
  ![Image Classification Example](image)

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

- Object localization: define the location and the category
  
  ![Object Localization Example](image)

  - Location
  - Category
Difficulties: within object variations

Variability: Camera position, Illumination, Internal parameters

→ Within-object variations
Difficulties: within-class variations
Category recognition

• Robust image description
  – Appropriate descriptors for categories

• Statistical modeling and machine learning for vision
  – Use and validation of appropriate techniques
Why machine learning?

• Early approaches: simple features + handcrafted models
• Can handle only few images, simples tasks

Why machine learning?

- Early approaches: manual programming of rules
- Tedious, limited and does not take into account the data

Figure 3. A system developed in 1978 by Ohta, Kanade and Sakai [33, 32] for knowledge-based interpretation of outdoor natural scenes. The system is able to label an image (c) into semantic classes: S-sky, T-tree, R-road, B-building, U-unknown.

Why machine learning?

- Today lots of data, complex tasks

  - Internet images, personal photo albums
  - Movies, news, sports

- Instead of trying to encode rules directly, learn them from examples of inputs and desired outputs
Types of learning problems

- Supervised
  - Classification
  - Regression
- Unsupervised
- Semi-supervised
- Active learning
- ....
Supervised learning

• Given training examples of inputs and corresponding outputs, produce the “correct” outputs for new inputs

• Two main scenarios:
  
  - **Classification**: outputs are discrete variables (category labels). Learn a decision boundary that separates one class from the other
  
  - **Regression**: also known as “curve fitting” or “function approximation.” Learn a continuous input-output mapping from examples (possibly noisy)
Unsupervised Learning

- Given only *unlabeled* data as input, learn some sort of structure

- The objective is often more vague or subjective than in supervised learning. This is more an exploratory/descriptive data analysis
Unsupervised Learning

- Clustering
  - Discover groups of “similar” data points
Unsupervised Learning

• **Quantization**
  – Map a continuous input to a discrete (more compact) output
Unsupervised Learning

- **Dimensionality reduction, manifold learning**
  - Discover a lower-dimensional surface on which the data lives
Other types of learning

- **Semi-supervised learning**: lots of data is available, but only small portion is labeled (e.g. since labeling is expensive)
Other types of learning

• **Semi-supervised learning**: lots of data is available, but only small portion is labeled (e.g. since labeling is expensive)
  – Why is learning from labeled and unlabeled data better than learning from labeled data alone?
Other types of learning

- **Active learning**: the learning algorithm can choose its own training examples, or ask a “teacher” for an answer on selected inputs.
Category recognition

- Image classification: assigning a class label to the image

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

- Supervised scenario: given a set of training images
Image classification

• Given
  Positive training images containing an object class
  Negative training images that don’t

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

?
Bag-of-features for image classification

- Origin: texture recognition
  - Texture is characterized by the repetition of basic elements or *textons*

Texture recognition

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

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

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>common people</td>
<td>sculpture</td>
<td>sculpture common sculpture</td>
<td>common</td>
</tr>
<tr>
<td></td>
<td>people</td>
<td></td>
<td>sculpture</td>
<td>people</td>
</tr>
<tr>
<td></td>
<td>common</td>
<td></td>
<td></td>
<td>common</td>
</tr>
<tr>
<td></td>
<td>people</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Extract regions → Compute descriptors → Find clusters and frequencies → Compute distance matrix → Classification

[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

Step 3: Compute distance matrix, Classification

[Nowak, Jurie & Triggs, ECCV'06], [Zhang, Marszalek, Lazebnik & Schmid, IJCV'07]
Step 1: feature extraction

- Scale-invariant image regions + SIFT (see lecture 2)
  - 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 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
Step 2: Quantization
Step 2: Quantization

Visual vocabulary

Clustering
<table>
<thead>
<tr>
<th>Examples for visual words</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Airplanes</strong></td>
</tr>
<tr>
<td><img src="image1" alt="Airplanes" /></td>
</tr>
<tr>
<td><img src="image2" alt="Airplanes" /></td>
</tr>
<tr>
<td><img src="image3" alt="Airplanes" /></td>
</tr>
<tr>
<td><img src="image4" alt="Airplanes" /></td>
</tr>
<tr>
<td><strong>Motorbikes</strong></td>
</tr>
<tr>
<td><img src="image5" alt="Motorbikes" /></td>
</tr>
<tr>
<td><img src="image6" alt="Motorbikes" /></td>
</tr>
<tr>
<td><img src="image7" alt="Motorbikes" /></td>
</tr>
<tr>
<td><img src="image8" alt="Motorbikes" /></td>
</tr>
<tr>
<td><strong>Faces</strong></td>
</tr>
<tr>
<td><img src="image9" alt="Faces" /></td>
</tr>
<tr>
<td><img src="image10" alt="Faces" /></td>
</tr>
<tr>
<td><img src="image11" alt="Faces" /></td>
</tr>
<tr>
<td><img src="image12" alt="Faces" /></td>
</tr>
<tr>
<td><strong>Wild Cats</strong></td>
</tr>
<tr>
<td><img src="image13" alt="Wild Cats" /></td>
</tr>
<tr>
<td><img src="image14" alt="Wild Cats" /></td>
</tr>
<tr>
<td><img src="image15" alt="Wild Cats" /></td>
</tr>
<tr>
<td><img src="image16" alt="Wild Cats" /></td>
</tr>
<tr>
<td><strong>Leaves</strong></td>
</tr>
<tr>
<td><img src="image17" alt="Leaves" /></td>
</tr>
<tr>
<td><img src="image18" alt="Leaves" /></td>
</tr>
<tr>
<td><img src="image19" alt="Leaves" /></td>
</tr>
<tr>
<td><img src="image20" alt="Leaves" /></td>
</tr>
<tr>
<td><strong>People</strong></td>
</tr>
<tr>
<td><img src="image21" alt="People" /></td>
</tr>
<tr>
<td><img src="image22" alt="People" /></td>
</tr>
<tr>
<td><img src="image23" alt="People" /></td>
</tr>
<tr>
<td><img src="image24" alt="People" /></td>
</tr>
<tr>
<td><strong>Bikes</strong></td>
</tr>
<tr>
<td><img src="image25" alt="Bikes" /></td>
</tr>
<tr>
<td><img src="image26" alt="Bikes" /></td>
</tr>
<tr>
<td><img src="image27" alt="Bikes" /></td>
</tr>
<tr>
<td><img src="image28" alt="Bikes" /></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
Gaussian mixture model (GMM)

- Mixture of Gaussians: weighted sum of Gaussians

\[ p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \]

where \( \mathcal{N}(x; \mu, \Sigma) = (2\pi)^{(-d/2)}|\Sigma|^{-1/2} \exp \left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right) \)
Hard or soft assignment

• K-means $\rightarrow$ hard assignment
  – Assign to the closest cluster center
  – Count number of descriptors assigned to a center

• Gaussian mixture model $\rightarrow$ soft assignment
  – Estimate distance to all centers
  – Sum over number of descriptors

• Represent image by a frequency histogram
• Each image is represented by a vector, typically 1000-4000 dimension
• fine grained – represent model instances
• coarse grained – represent object categories
Bag-of-features for image classification

Step 1: Extract regions
Step 2: Compute descriptors and find clusters and frequencies
Step 3: Compute distance matrix and 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

Train classifier, e.g. SVM
• Assign input vector to one of two or more classes
• Any decision rule divides input space into *decision regions* separated by *decision boundaries*
Nearest Neighbor Classifier

- Assign label of nearest training data point to each test data point

Voronoi partitioning of feature space for 2-category 2-D and 3-D data

from Duda et al.
For a new point, find the k closest points from training data
Labels of the k points “vote” to classify
Works well provided there is lots of data and the distance function is good

\[ k = 5 \]
Linear classifiers

• Find linear function \((hyperplane)\) to separate positive and negative examples

\[ x_i \text{ positive: } x_i \cdot w + b \geq 0 \]

\[ x_i \text{ negative: } x_i \cdot w + b < 0 \]

Which hyperplane is best?
• Generalization is not good in this case:

• Better if a margin is introduced:
Support vector machines

• Find hyperplane that maximizes the *margin* between the positive and negative examples

\[
\begin{align*}
\text{x}_i \text{ positive (}y_i = 1): & \quad \text{x}_i \cdot \text{w} + b \geq 1 \\
\text{x}_i \text{ negative (}y_i = -1): & \quad \text{x}_i \cdot \text{w} + b \leq -1
\end{align*}
\]

For support vectors, \( \text{x}_i \cdot \text{w} + b = \pm 1 \)

The margin is \( \frac{2}{||\text{w}||} \)
Nonlinear SVMs

- Datasets that are linearly separable work out great:

  ![Linear Separable Datasets]

  ```plaintext
  • But what if the dataset is just too hard?
  ```

  ![Dataset困難]

  ```plaintext
  • We can map it to a higher-dimensional space:
  ```

  ![High-Dimensional Mapping]
Nonlinear SVMs

• General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:

\[ \Phi: x \rightarrow \varphi(x) \]
The kernel trick: instead of explicitly computing the lifting transformation $\varphi(x)$, define a kernel function $K$ such that

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$$

This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i K(x_i, x) + b$$
Kernels for bags of features

- Hellinger kernel \( K(h_1, h_2) = \sum_{i=1}^{N} \sqrt{h_1(i)h_2(i)} \)

- 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, \( \chi^2 \) distance etc.

\[
D_{\chi^2}(h_1, h_2) = \sum_{i=1}^{N} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)}
\]
Combining features

- SVM with multi-channel chi-square kernel

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

- 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

- Red circles: foreground word more probable
- Green circles: background word more probable
Illustration

A linear SVM trained from positive and negative window descriptors

A few of the highest weighed 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]
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution
Pyramid match kernel

- Weighted sum of histogram intersections at multiple resolutions (linear in the number of features instead of cubic)

\[ \bigcap \approx \]

optimal partial matching between sets of features
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>
## Retrieval examples

<table>
<thead>
<tr>
<th>(a) kitchen</th>
<th>living room</th>
<th>living room</th>
<th>living room</th>
<th>office</th>
<th>living room</th>
<th>living room</th>
<th>living room</th>
<th>living room</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b) kitchen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) store</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(d) tall bldg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e) tall bldg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(f) inside city</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>

Bag-of-features approach by Zhang et al.’07: 54 %
Easiest and hardest classes

- Sources of difficulty:
  - Lack of texture
  - Camouflage
  - Thin, articulated limbs
  - Highly deformable shape
**Evaluation BoF – spatial**

**Image classification results on PASCAL’07 train/val set**

<table>
<thead>
<tr>
<th>(SH, Lap, MSD) x (SIFT,SIFTC) spatial layout</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.53</td>
</tr>
<tr>
<td>2x2</td>
<td>0.52</td>
</tr>
<tr>
<td>3x1</td>
<td>0.52</td>
</tr>
<tr>
<td>1,2x2,3x1</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Spatial layout not dominant for PASCAL’07 dataset
Combination improves average results, i.e., it is appropriate for some classes
Image classification results on PASCAL’07 train/val set for individual categories

<table>
<thead>
<tr>
<th>Category</th>
<th>1</th>
<th>3x1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheep</td>
<td>0.339</td>
<td>0.256</td>
</tr>
<tr>
<td>Bird</td>
<td>0.539</td>
<td>0.484</td>
</tr>
<tr>
<td>DiningTable</td>
<td>0.455</td>
<td>0.502</td>
</tr>
<tr>
<td>Train</td>
<td>0.724</td>
<td>0.745</td>
</tr>
</tbody>
</table>

Results are category dependent!

⇒ Combination helps somewhat
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
Recent extensions

- **Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification. J. Yang et al., CVPR’09.**
  - Local coordinate coding, linear SVM, excellent results in 2009 PASCAL challenge

- **Learning Mid-level features for recognition, Y. Boureau et al., CVPR’10.**
  - Use of sparse coding techniques and max pooling
Recent extensions

• Efficient Additive Kernels via Explicit Feature Maps, A. Vedaldi and Zisserman, CVPR’10.
  – approximation by linear kernels

• Improving the Fisher Kernel for Large-Scale Image Classification, Perronnin et al., ECCV’10
  – More discriminative descriptor, power normalization, linear SVM
Fisher vector image representation

- Mixture of Gaussian/ k-means stores nr of points per cell

- Fisher vector adds 1st & 2nd order moments
  - More precise description of regions assigned to cluster
  - Fewer clusters needed for same accuracy
  - Per cluster store: mean and variance of data in cell
  - Representation 2D times larger, at same computational cost
  - High dimensional, robust representation
Fisher vector image representation

\[ X = \{x_t, t = 1 \ldots T\} \] is the set of T i.i.d. D-dim local descriptors (e.g. SIFT) extracted from an image:

\[ u_\lambda(x) = \sum_{i=1}^{K} w_i u_i(x) \] is a Gaussian Mixture Model (GMM) with parameters \( \lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \ldots N\} \) trained on a large set of local descriptors: a **visual vocabulary**

FV formulas:

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

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

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

FV formulas:

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

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

Soft BOV formula:

$\frac{1}{T} \sum_{t=1}^{T} \gamma_t(i)$

Like the (original) BOV the FV is an average of local statistics.

The FV extends the BOV and includes higher-order statistics (up to 2\textsuperscript{nd} order)

Results on VOC 2007: BOV = 43.6 \% $\rightarrow$ FV = 57.7 \% $\rightarrow$ \sqrt{FV} = 62.1 \%
Large-scale image classification

- Image classification: assigning a class label to the image

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

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

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

<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>

- Fisher improves over BOF
- Fisher comparable to BOF + non-linear classifier
- Limited gain due to SPM on PASCAL
- Sqrt helps for Fisher and BOF
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

• Significant impact on accuracy
• For very high dimensions little impact

- 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
One-versus-rest works

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

<table>
<thead>
<tr>
<th>Top-1</th>
<th>BOV</th>
<th>FV</th>
<th>w-OVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>26.4</td>
<td>45.7</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
- \( w\text{-OVR} > u\text{-OVR} \)
- 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

(a) Star Anise (92.45%)  
(b) Geyser (85.45%)  
(c) Pulp Magazine (83.01%)  
(d) Carrycot (81.48%)  
(e) European gallinule (15.00%)  
(f) Sea Snake (10.00%)  
(g) Paintbrush (4.68%)  
(h) Mountain Tent (0.00%)
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