Bag-of-features models 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
...

![Image showing a car and a cow]
Category recognition

• Image classification: assigning a class label to the image

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

• Object localization: define the location and the 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
  
  Negative training images that don’t

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

?
Bag-of-features – Origin: texture recognition

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

Bag-of-features – Origin: texture recognition

histogram

Universal texton dictionary
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>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-words
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

Step 2: Compute descriptors and find clusters and frequencies

Step 3: Compute distance matrix

Classification

SVM
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 cell
- Exp.: Horizontal/vertical step size 3 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></th>
<th>Airplanes</th>
<th>Motorbikes</th>
<th>Faces</th>
<th>Wild Cats</th>
<th>Leaves</th>
<th>People</th>
<th>Bikes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1" alt="Airplanes Images" /></td>
<td><img src="image2" alt="Motorbikes Images" /></td>
<td><img src="image3" alt="Faces Images" /></td>
<td><img src="image4" alt="Wild Cats Images" /></td>
<td><img src="image5" alt="Leaves Images" /></td>
<td><img src="image6" alt="People Images" /></td>
<td><img src="image7" 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
K-means clustering

- Minimizing sum of squared Euclidean distances between points $x_i$ and their nearest cluster centers

- **Algorithm:**
  - Randomly initialize K cluster centers
  - Iterate until convergence:
    - Assign each data point to the nearest center
    - Recompute each cluster center as the mean of all points assigned to it

- Local minimum, solution dependent on initialization

- Initialization important, run several times, select best
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 → hard assignment**
  - Assign to the closest cluster center
  - Count number of descriptors assigned to a center

- **Gaussian mixture model → 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, normalization with L1/L2 norm
• fine grained – represent model instances
• coarse grained – represent object categories
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 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
Linear classifiers

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

\[
\begin{align*}
\text{x}_i \text{ positive: } & \quad \text{x}_i \cdot \text{w} + b \geq 0 \\
\text{x}_i \text{ negative: } & \quad \text{x}_i \cdot \text{w} + b < 0
\end{align*}
\]
Linear classifiers - margin

- Generalization is not good in this case:

- Better if a margin is introduced:
Nonlinear SVMs

• Datasets that are linearly separable work out great:

• But what if the dataset is just too hard?

• We can map it to a higher-dimensional space:
Nonlinear SVMs

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

- *The kernel trick*: instead of explicitly computing the lifting transformation $\phi(x)$, define a kernel function $K$ such that

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

- 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

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

[J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study, IJCV 2007]
Combining features

• For linear SVMs
  – Early fusion: concatenation the descriptors
  – Late fusion: learning weights to combine the classification scores

• Theoretically no clear winner

• In practice late fusion give better results
  – In particular if different modalities are combined
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
Illustration

Localization according to visual word probability

- Red circle: foreground word more probable
- Green circle: background word more probable
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
  – no model of the object location
Evaluation of image classification

• PASCAL VOC [05-12] 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

<table>
<thead>
<tr>
<th>Category</th>
<th>Example Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aeroplane</td>
<td><img src="image1.png" alt="Aeroplane" /></td>
</tr>
<tr>
<td>Bicycle</td>
<td><img src="image2.png" alt="Bicycle" /></td>
</tr>
<tr>
<td>Bird</td>
<td><img src="image3.png" alt="Bird" /></td>
</tr>
<tr>
<td>Boat</td>
<td><img src="image4.png" alt="Boat" /></td>
</tr>
<tr>
<td>Bottle</td>
<td><img src="image5.png" alt="Bottle" /></td>
</tr>
<tr>
<td>Bus</td>
<td><img src="image6.png" alt="Bus" /></td>
</tr>
<tr>
<td>Car</td>
<td><img src="image7.png" alt="Car" /></td>
</tr>
<tr>
<td>Cat</td>
<td><img src="image8.png" alt="Cat" /></td>
</tr>
<tr>
<td>Chair</td>
<td><img src="image9.png" alt="Chair" /></td>
</tr>
<tr>
<td>Cow</td>
<td><img src="image10.png" alt="Cow" /></td>
</tr>
</tbody>
</table>
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
Precision/Recall

- Ranked list for category A:

A, C, B, A, B, C, C, A; in total four images with category A
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

• Adding objectness boxes [Sanchez at al.’12]: mAP 66.3
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

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]

<table>
<thead>
<tr>
<th>Coast</th>
<th>Forest</th>
<th>Mountain</th>
<th>Open country</th>
<th>Highway</th>
<th>Inside city</th>
<th>Tall building</th>
<th>Street</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Coast" /></td>
<td><img src="image" alt="Forest" /></td>
<td><img src="image" alt="Mountain" /></td>
<td><img src="image" alt="Open country" /></td>
<td><img src="image" alt="Highway" /></td>
<td><img src="image" alt="Inside city" /></td>
<td><img src="image" alt="Tall building" /></td>
<td><img src="image" alt="Street" /></td>
</tr>
<tr>
<td><img src="image" alt="Suburb" /></td>
<td><img src="image" alt="Bedroom" /></td>
<td><img src="image" alt="Kitchen" /></td>
<td><img src="image" alt="Living room" /></td>
<td><img src="image" alt="Office" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Store" /></td>
<td><img src="image" alt="Industrial" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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>office</td>
<td>inside city</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) store</td>
<td>mountain</td>
<td>forest</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(d) tall bldg</td>
<td>inside city</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e) tall bldg</td>
<td>inside city</td>
<td>mountain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(f) inside city</td>
<td></td>
<td>tall bldg</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>
**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></td>
</tr>
<tr>
<td>3x1</td>
<td></td>
</tr>
<tr>
<td>1,2x2,3x1</td>
<td></td>
</tr>
</tbody>
</table>
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
Evaluation BoF - spatial

Image classification results on PASCAL’07 train/val set for individual categories

<table>
<thead>
<tr>
<th></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

• Recent extensions
  – Flexible, object-centered grid
    • Shape masks [Marszalek’12] => additional annotations
  – Weakly supervised localization of objects
    • [Russakovsky et al.’12]
Recent extensions

- Efficient Additive Kernels via Explicit Feature Maps
  [Perronnin et al.’10, Maji and Berg’09, A. Vedaldi and Zisserman’10]

- 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]

- Improved performance + linear SVM
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:

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

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

\[\gamma_t(i) = \text{soft-assignment of patch } x_t \text{ to Gaussian i}\]

Fisher Vector = concatenation of per-Gaussian gradient vectors

For image retrieval in our experiments:
- only deviation wrt mean, dim: K*D [K number of Gaussians, D dim of descriptor]
- variance does not improve for comparable vector length
Image classification with Fisher vector

- Dense SIFT
- Fisher vector (k=32 to 1024, total dimension from approx. 5000 to 160000)
- Normalization
  - square-rooting
  - L2 normalization
  - [Perronnin’10], [Image categorization using Fisher kernels of non-iid image models, Cinbis, Verbeek, Schmid, CVPR’12]

- Classification approach
  - Linear classifiers
  - One versus rest classifier
Image classification with Fisher vector

- Evaluation 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

<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>60.4</td>
<td>60.6</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
- [Chatfield et al. 2011]
Large-scale image classification

**IMAGENET** has 14M images from 22k classes

**Standard Subsets**
- ImageNet Large Scale Visual Recognition Challenge 2010 (ILSVRC)
  - 1000 classes and 1.4M images
- ImageNet10K dataset
  - 10184 classes and ~ 9 M images

(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%)
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

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

\[\rho = \frac{1}{2}\] Natural rebalancing, same weight to positive and negatives
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
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
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
Experimental results for ILSVRC 2010

- Features: dense SIFT, reduced to 64 dim with PCA
- 256 Gaussian Fisher vector using mean and variance + SP (3x1) (4x [2x64x256] ~ 130k dim), square-root + L2 norm
- BOF dim=1024 + SP (3x1) (dim 4000), square-root + L2 norm
- Different classification methods

<table>
<thead>
<tr>
<th></th>
<th>w-OVR</th>
<th>MUL</th>
<th>RNK</th>
<th>WAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOV</td>
<td>26.4</td>
<td>22.7</td>
<td>20.8</td>
<td>24.1</td>
</tr>
<tr>
<td>FV</td>
<td>45.7</td>
<td>46.2</td>
<td>46.1</td>
<td>46.1</td>
</tr>
</tbody>
</table>
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>

Top-1 accuracy

- 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
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