Fisher Vector image representation

Machine Learning and Category Representation 2013-2014
Jakob Verbeek, December 13, 2013

Course website:
http://lear.inrialpes.fr/~verbeek/MLCR.13.14
Fisher vector image representation

• An alternative to bag-of-words image representation introduced in
  *Fisher kernels on visual vocabularies for image categorization*
  

• FV in comparison to the BoW representation
  – Both FV and BoW are based on a visual vocabulary, with assignment of patches to visual words
  – FV based on Mixture of Gaussian clustering of patches, BoW based on k-means clustering
  – FV Extracts a larger image signature than the BoW representation for a given number of visual words
Fisher vector representation: Motivation

- Suppose we want to refine a given visual vocabulary

- Bag-of-word histogram stores # patches assigned to each word
  - Need more words to refine the representation
  - But this directly increases the computational cost
  - And leads to many empty bins: redundancy
Fisher vector representation: Motivation

- Feature vector quantization is computationally expensive
- To extract visual word histogram for a new image
  - Compute distance of each local descriptor to each k-means center
  - run-time $O(NKD)$ : linear in
    - $N$: nr. of feature vectors ~ $10^4$ per image
    - $K$: nr. of clusters ~ $10^3$ for recognition
    - $D$: nr. of dimensions ~ $10^2$ (SIFT)

- So in total in the order of $10^9$ multiplications per image to obtain a histogram of size 1000

- Can this be done more efficiently ?!
  - Yes, extract more than just a visual word histogram
Fisher vector representation in a nutshell

• Instead, the Fisher Vector also records the mean and variance of the points per dimension in each cell
  – More information for same # visual words
  – Does not increase computational time significantly
  – Leads to high-dimensional feature vectors

• Even when the counts are the same, the position and variance of the points in the cell can vary
Image representation using Fisher kernels

• General idea of Fischer vector representation
  ▶ Fit probabilistic model to data \( p(X; \Theta) \)
  ▶ Represent data with derivative of data log-likelihood
    “How does the data want that the model changes?”
    \[
    G(X, \Theta) = \frac{\partial \log p(x; \Theta)}{\partial \Theta}
    \]

• Mixture of Gaussians to model the local (SIFT) descriptors
  \( X = \{x_n\}_{n=1}^{N} \)
  \[
  L(X, \Theta) = \sum_n \log p(x_n)
  \]
  \[
  p(x_n) = \sum_k \pi_k N(x_n; m_k, C_k)
  \]
  ▶ Define mixing weights using the soft-max function
    \[
    \pi_k = \frac{\exp \alpha_k}{\sum_k \exp \alpha_k}
    \]
Image representation using Fisher kernels

- Mixture of Gaussians to model the local (SIFT) descriptors

\[ L(\Theta) = \sum_n \log p(x_n) \]
\[ p(x_n) = \sum_k \pi_k N(x_n; m_k, C_k) \]

- The parameters of the model are

\[ \Theta = \{\alpha_k, m_k, C_k\}_{k=1}^K \]

- where we use diagonal covariance matrices

- Concatenate derivatives to obtain data representation

\[ G(X, \Theta) = \left( \frac{\partial L}{\partial \alpha_1}, \ldots, \frac{\partial L}{\partial \alpha_K}, \frac{\partial L}{\partial m_1}, \ldots, \frac{\partial L}{\partial m_K}, \frac{\partial L}{\partial C_1^{-1}}, \ldots, \frac{\partial L}{\partial C_K^{-1}} \right)^T \]
Image representation using Fisher kernels

- Data representation

\[ G(X, \Theta) = \left( \frac{\partial L}{\partial \alpha_1}, \ldots, \frac{\partial L}{\partial \alpha_K}, \frac{\partial L}{\partial m_1}, \ldots, \frac{\partial L}{\partial m_K}, \frac{\partial L}{\partial C^{-1}_1}, \ldots, \frac{\partial L}{\partial C^{-1}_K} \right)^T \]

- In total \( K(1+2D) \) dimensional representation, since for each visual word / Gaussian we have

  Count (1 dim):  
  \[ \frac{\partial L}{\partial \alpha_k} = \sum_n (q_{nk} - \pi_k) \]

  More/less patches assigned to visual word than usual?

  Mean (D dims):  
  \[ \frac{\partial L}{\partial m_k} = C_k^{-1} \sum_n q_{nk} (x_n - m_k) \]

  Center of assigned data relative to cluster center

  Variance (D dims):  
  \[ \frac{\partial L}{\partial C^{-1}_k} = \frac{1}{2} \sum_n q_{nk} (C_k - (x_n - m_k)^2) \]

  Variance of assigned data relative to cluster variance

With the soft-assignments:  
\[ q_{nk} = p(k|x_n) = \frac{\pi_k p(x_n|k)}{p(x_n)} \]
Illustrative example in 2d

New Data Points

Gradient with respect to mean
Function approximation view

- Suppose our local descriptors are 1 dimensional for simplicity
  - Vocabulary quantizes the real line
- Suppose we use a linear function, e.g., for image classification
  - **BoW**: locally constant function
    \[
    f(x;w) = \sum_{k=1}^{K} x_k w_k
    \]
  - **FV**: locally constant + linear + quadratic function
    \[
    f(x;w) = \sum_{k=1}^{K} \left[ \frac{\partial L}{\partial \alpha_k} \frac{\partial L}{\partial \mu_k} \frac{\partial L}{\partial C_k^{-1}} \right]^T w_k
    \]
Images from categorization task PASCAL VOC

- Yearly evaluation from 2005 to 2012 for image classification
Fisher vectors: classification performance VOC'07

- Fisher vector representation yields better performance for a given number of Gaussians / visual words than Bag-of-words.
- For a fixed dimensionality Fisher vectors perform better, and are more efficient to compute.
Bag-of-words vs. Fisher vector image representation

• Bag-of-words image representation
  ▶ Off-line: fit k-means clustering to local descriptors
  ▶ Represent image with histogram of visual word counts: K dimensions

• Fischer vector image representation
  ▶ Off-line: fit MoG model to local descriptors
  ▶ Represent image with gradient of log-likelihood: K(2D+1) dimensions

• Computational cost similar:
  ▶ Both compare N descriptors to K visual words (centers / Gaussians)

• Memory usage: higher for fisher vectors
  ▶ Fisher vector is a factor (2D+1) larger, e.g. a factor 257 for SIFTs!
    • For 1000 visual words the FV has 257,000 dimensions
  ▶ However, because we store more information per visual word, we can generally obtain same or better performance with far less visual words
FV normalization

- Normalization with Fisher information matrix $F = E_{p(x)}[G(X, \Theta)G(X, \Theta)^T]$  
  - Invariance w.r.t. re-parametrization, e.g. does not matter if we use standard dev., variance, or inverse-variance parameter

\[
\tilde{G}(X, \Theta) = F^{-1/2} \left( \frac{\partial L}{\partial \alpha_1}, ..., \frac{\partial L}{\partial \alpha_K}, \frac{\partial L}{\partial m_1}, ..., \frac{\partial L}{\partial m_K}, \frac{\partial L}{\partial C_1^{-1}}, ..., \frac{\partial L}{\partial C_K^{-1}} \right)^T
\]

- Power normalization to reduce sparseness
  - Element-wise signed-power $\tilde{z} = \text{sign}(z)|z|^p$
  - Typically power set to 1/2, i.e. signed-square-root

- L2 normalization to make scales comparable
  - Eliminates effect of the number of patches
  - Increase FV magnitude for “typical” images with small gradient
  - Divide FV by its L2 norm
FV normalization, effect on performance

- Power normalization to reduce sparseness
- L2 normalization to make scales comparable
- Can also use Spatial Pyramids
  - compute FV per spatial cell, and concatenate
  - Here: 1x1, 2x2, 3x1 grid over image, 8 cells total

<table>
<thead>
<tr>
<th>PN</th>
<th>$\ell_2$</th>
<th>SP</th>
<th>SIFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>49.6</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>57.9 (+8.3)</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>54.2 (+4.6)</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>51.5 (+1.9)</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>59.6 (+10.0)</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>58.8 (+10.2)</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>57.3 (+7.7)</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>61.8 (+12.2)</td>
</tr>
</tbody>
</table>

- Power + L2 normalization most important
- Spatial Pyramid also helps, but increases FV size by a factor 8
PCA projection of local features

- We used diagonal variances
  - Assumes dimensions are de-correlated
  - Not true for most local descriptors, like SIFT

- Perform PCA on the descriptors to de-correlate them
  - Possibly also reduce the dimension too

- Effect on image classification performance
Reading material

- A recent overview article on the Fisher Vector representation

  - Image Classification with the Fisher Vector: Theory and Practice
    Jorge Sanchez; Florent Perronnin; Thomas Mensink; Jakob Verbeek
    International Journal of Computer Vision, springer, 2013