Overview

• Instance level search
  – Local features and matching
  – Efficient visual recognition

• Image classification & object localization
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
  - Car: present
  - Cow: present
  - Bike: not present
  - Horse: not present
  ...

- Object localization: define the location and the category
Image classification

• Given

Positive training images containing an object class

![Motorcycle](image1)

![Motorcycle](image2)

![Motorcycle](image3)

Negative training images that don’t

![Crushed paper](image4)

![Airplane](image5)

![Office](image6)

• Classify

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

![Motorcycle](image7)
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

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

[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

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

- Scale-invariant image regions + SIFT
  - 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
- Compute distance matrix: $d(S_i, S_j)$

**Step 3**
- Classification

SVM
Step 2: Quantization
Step 2: Quantization

Clustering
Step 2: Quantization

Visual vocabulary

Clustering
<table>
<thead>
<tr>
<th>Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplanes</td>
<td><img src="image1" alt="Airplanes Examples" /> <img src="image2" alt="Airplanes Examples" /></td>
</tr>
<tr>
<td>Motorbikes</td>
<td><img src="image3" alt="Motorbikes Examples" /> <img src="image4" alt="Motorbikes Examples" /></td>
</tr>
<tr>
<td>Faces</td>
<td><img src="image5" alt="Faces Examples" /> <img src="image6" alt="Faces Examples" /></td>
</tr>
<tr>
<td>Wild Cats</td>
<td><img src="image7" alt="Wild Cats Examples" /> <img src="image8" alt="Wild Cats Examples" /></td>
</tr>
<tr>
<td>Leaves</td>
<td><img src="image9" alt="Leaves Examples" /> <img src="image10" alt="Leaves Examples" /></td>
</tr>
<tr>
<td>People</td>
<td><img src="image11" alt="People Examples" /> <img src="image12" alt="People Examples" /></td>
</tr>
<tr>
<td>Bikes</td>
<td><img src="image13" alt="Bikes Examples" /> <img src="image14" alt="Bikes Examples" /></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
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 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

Decision boundary

Zebra

Non-zebra
Training data

Vectors are histograms, one from each training image

Train classifier, e.g. SVM
Classifiers

- K-nearest neighbor classifier

- Linear classifier
  - Support Vector Machine

- Non-linear classifier
  - Kernel trick
  - Explicit lifting
Classification

- 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
k-Nearest Neighbors

- 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

\[ \mathbf{x}_i \text{ positive (} y_i = 1\): } \mathbf{x}_i \cdot \mathbf{w} + b \geq 1 \]

\[ \mathbf{x}_i \text{ negative (} y_i = -1\): } \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \]

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

The margin is \( 2 / ||\mathbf{w}|| \)
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:

\[ \Phi: x \rightarrow \phi(x) \]
Nonlinear SVMs

- *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
  \[ K(h_1, h_2) = \exp\left(-\frac{1}{A} D(h_1, h_2)^2\right) \]
  \[ D_{\chi^2}(h_1, h_2) = \sum_{i=1}^{N} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)} \]

- \( D \) can be Euclidean distance, \( \chi^2 \) distance etc.
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

- foreground word more probable
- background word more probable
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

bikes  books  building  cars  people  phones  trees
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
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
Overview

• Instance level search
  – Local features and matching
  – Efficient visual recognition

• Image classification & *object localization*
Recognition

- **Classification**
  - Object present/absent in an image
  - Often presence of a significant amount of background clutter

- **Localization / Detection**
  - Localize object within the frame
  - Bounding box or pixel-level segmentation
Pixel-level object classification
Difficulties

- Intra-class variations
- Scale and viewpoint change
- Multiple aspects of categories
Approaches

• Intra-class variation
  => Modeling of the variations, mainly by learning from a large dataset, for example by SVMs

• Scale + limited viewpoints changes
  => multi-scale approach or invariant local features

• Multiple aspects of categories
  => separate detectors for each aspect, front/profile face, build an approximate 3D “category” model
Approaches

- Localization (bounding box)
  - Sliding window approach

- Localization (segmentation)
  - Pixel-based +MRF
Localization with sliding window

Training

Positive examples

Negative examples

Description + Learn a classifier
Localization with sliding window

Testing at multiple locations and scales

Find local maxima, non-maxima suppression
Haar Wavelet / SVM Human Detector

Training set (2k positive / 10k negative)

Haar wavelet descriptors

Support vector machine

1326-D descriptor

test

descriptors

Multi-scale search

results

Test image

[Papageorgiou & Poggio, 1998]
Which Descriptors are Important?

32x32 descriptors

16x16 descriptors

Mean response difference between positive & negative training examples

Essentially just a coarse-scale human silhouette template!
Some Detection Results
The Viola/Jones Face Detector

• A seminal approach to real-time object detection
  • Training is slow, but detection is very fast

• Key ideas
  – Integral images for fast feature evaluation
  – Boosting for feature selection
  – Attentional cascade for fast rejection of non-face windows


P. Viola and M. Jones. Robust real-time face detection. IJCV 57(2), 2004.
Image Features

“Rectangle filters”

Value =

\[ \sum (\text{pixels in white area}) - \sum (\text{pixels in black area}) \]
Fast computation with integral images

- The *integral image* computes a value at each pixel \((x,y)\) that is the sum of the pixel values above and to the left of \((x,y)\), inclusive.

- This can quickly be computed in one pass through the image.
Feature selection

• For a 24x24 detection region, the number of possible rectangle features is $\sim 160,000!$
Feature selection

• For a 24x24 detection region, the number of possible rectangle features is \(~160,000\)!

• At test time, it is impractical to evaluate the entire feature set

• Can we create a good classifier using just a small subset of all possible features?

• How to select such a subset?
Boosting

• Boosting is a classification scheme that works by combining weak learners into a more accurate ensemble classifier

• Training consists of multiple boosting rounds
  • During each boosting round, we select a weak learner that does well on examples that were hard for the previous weak learners
  • “Hardness” is captured by weights attached to training examples

Boosting for face detection

- First two features selected by boosting:

This feature combination can yield 100% detection rate and 50% false positive rate
Attentional cascade

- We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows.

- Positive response from the first classifier triggers the evaluation of a second (more complex) classifier, and so on.

- A negative outcome at any point leads to the immediate rejection of the sub-window.

![Diagram of attentional cascade with classifiers and outcomes](attachment:image.png)
The implemented system

• Training Data
  • 5000 faces
    – All frontal, rescaled to 24x24 pixels
  • 300 million non-faces
    – 9500 non-face images
  • Faces are normalized
    – Scale, translation

• Many variations
  • Across individuals
  • Illumination
  • Pose
Result of Face Detector on Test Images
Profile Detection
Profile Features
Summary: Viola/Jones detector

- Rectangle features
- Integral images for fast computation
- Boosting for feature selection
- Attentional cascade for fast rejection of negative windows
- Available in open CV
Histogram of Oriented Gradient Human Detector

- Descriptors are a grid of local Histograms of Oriented Gradients (HOG)
- Linear SVM for runtime efficiency
- Tolerates different poses, clothing, lighting and background
- Assumes upright fully visible people

[Dalal & Triggs, CVPR 2005]
Human detection