Visual Codebook

Tae-Kyun Kim
Sidney Sussex College
Visual Words

• Visual words are base elements to describe an image.
• Interest points are detected from an image
  – Corners, Blob detector, SIFT detector
• Image patches are represented by descriptor
  – SIFT (Scale-Invariant Feature Transform) or Raw pixel intensity

\[ \mathbf{x} \in \mathbb{R}^D \]
Building a Visual Codebook (Dictionary)

• Visual words (real-valued vectors) can be compared using Euclidean distance:

\[ E(x_1, x_2) = \sqrt{\sum_d (x_1^d - x_2^d)^2} \]

• These vectors are divided into groups which are similar, essentially clustered together, to form a codebook.
K-means Clustering

• The two steps repeated until no vector changes membership:
  1. Compute a cluster center for each cluster as the mean of the cluster members
  2. Reassign each data point to the cluster whose center is nearest

• The cluster centers (mean values) now form a visual dictionary.
Histogram of Visual Words

- Every visual word is compared with codewords and assigned to the nearest codeword.

- Histogram bins are codewords and each bin counts the number of words assigned to the codeword.

“bag of words”
Categorisation

**Learning**
- Feature detection & representation
- Image representation

**Recognition**
- Codewords dictionary
- Category models (and/or) classifiers
- Category decision
Categorisation

- Histogram matching of a pair of images $P$ and $Q$ is

$$\mathcal{I}(P, Q) = \sum_{n=1}^{K} \min(H^{(n)}(P), H^{(n)}(Q))$$

- Based on HMK, we can use various classifiers such as kNN, SVM, Naïve-Bayes classifiers or pLSA, LDA.

Class assignment by

$$\text{class} \left( \arg\max_i \mathcal{I}(P_i, Q) \right)$$
Summary of K-means Codebook

- The histogram is 1-D, a highly compact and robust representation of images.
- The histogram representation greatly facilitates modelling images and their categories.
- Codeword assignment (quantisation process) by K-means is time-demanding.
- K-means is an unsupervised learning method.
Case Study:
Video (action) Categorisation by RF Codebook

In collaboration with Tsz Ho Yu
Demo video: Action categorisation

no action
Learning

- Interest point detection
- Bag of semantic textons
- Classification

Recognition

- FASTEST Corner
- Descriptor/Codebook
  - Spatiotemporal Semantic Texton Forest
  - kNN Classifier

- Category decision

- Powerful Discriminative Codebook
- Extremely Fast
Relation to the previous

<table>
<thead>
<tr>
<th></th>
<th>2D → 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>Images</td>
</tr>
<tr>
<td><strong>Interest Point</strong></td>
<td>Corners</td>
</tr>
<tr>
<td><strong>Descriptor</strong></td>
<td>Patches</td>
</tr>
<tr>
<td><strong>Codebook</strong></td>
<td>SIFT</td>
</tr>
<tr>
<td><strong>Representation</strong></td>
<td>K-means</td>
</tr>
<tr>
<td><strong>Classifier</strong></td>
<td>Histogram of visual words</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Relation to the previous

*Patches ↔ Space-Time Volumes*

Image (2D)  \[ \begin{aligned} &\text{Patch} \\
&\text{Patch} \end{aligned} \]

Video (3D)  \[ \begin{aligned} &\text{Video} \\
&\text{Video} \end{aligned} \]

Cuboid  \[ \begin{aligned} &\text{Cuboid} \\
&\text{Cuboid} \end{aligned} \]  

or

\[ \mathbf{x} \in \mathbb{R}^D \]

“visual words”
Data Set and Interest Point
# Action data set

- [http://www.nada.kth.se/cvap/actions/](http://www.nada.kth.se/cvap/actions/)
- 25 subjects, 2391 sequences – the largest public action DB

<table>
<thead>
<tr>
<th></th>
<th>walking</th>
<th>jogging</th>
<th>running</th>
<th>boxing</th>
<th>Hand waving</th>
<th>Hand clapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td><img src="image1.png" alt="image" /></td>
<td><img src="image2.png" alt="image" /></td>
<td><img src="image3.png" alt="image" /></td>
<td><img src="image4.png" alt="image" /></td>
<td><img src="image5.png" alt="image" /></td>
<td><img src="image6.png" alt="image" /></td>
</tr>
<tr>
<td>s2</td>
<td><img src="image1.png" alt="image" /></td>
<td><img src="image2.png" alt="image" /></td>
<td><img src="image3.png" alt="image" /></td>
<td><img src="image4.png" alt="image" /></td>
<td><img src="image5.png" alt="image" /></td>
<td><img src="image6.png" alt="image" /></td>
</tr>
<tr>
<td>s3</td>
<td><img src="image1.png" alt="image" /></td>
<td><img src="image2.png" alt="image" /></td>
<td><img src="image3.png" alt="image" /></td>
<td><img src="image4.png" alt="image" /></td>
<td><img src="image5.png" alt="image" /></td>
<td><img src="image6.png" alt="image" /></td>
</tr>
<tr>
<td>s4</td>
<td><img src="image1.png" alt="image" /></td>
<td><img src="image2.png" alt="image" /></td>
<td><img src="image3.png" alt="image" /></td>
<td><img src="image4.png" alt="image" /></td>
<td><img src="image5.png" alt="image" /></td>
<td><img src="image6.png" alt="image" /></td>
</tr>
</tbody>
</table>
Space-Time Interest Point

- In Dollar et al. 2005, separable linear filters are applied.
- The response function is

\[ R = (I * G * h_{ev})^2 + (I * G * h_{od})^2 \]

- \( G(x, y; \sigma) \) is 2D Gaussian smoothing kernel
- \( h_{ev} / h_{od} \) are a pair of 1D Gabor filters temporally applied as

\[ h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2} \quad h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2} \]
Space-Time Interest Point

• For multiple scales, it runs the detector over a set of spatial and temporal scales.

• Interest points are detected as local maxima of the response-function.
Matlab Demo:

Space-Time Interest Points
Background:
Randomised Decision Forest
Binary Decision Trees

- feature vector $\mathbf{v} \in \mathbb{R}^D$
- split functions $f_n(\mathbf{v})$
- thresholds $t_n$
- Classifications $P_n(c)$

Slide credits to J. Shotton
Toy Learning Example

• Try several lines, chosen at random

• Keep line that best separates data
  – information gain

• Recurse

- feature vectors are $x, y$ coordinates: $\mathbf{v} = [x, y]^T$
- split functions are lines with parameters $a, b$: $f_n(\mathbf{v}) = ax + by$
- threshold determines intercepts: $t_n$
- four classes: purple, blue, red, green

Slide credits to J. Shotton
Toy Learning Example

- Try several lines, chosen at random

- Keep line that best separates data
  - information gain

- Recurse

- feature vectors are $x$, $y$ coordinates: $v = [x, y]^T$
- split functions are lines with parameters $a$, $b$: $f_n(v) = ax + by$
- threshold determines intercepts: $t_n$
- four classes: purple, blue, red, green

Slide credits to J. Shotton
Toy Learning Example

- Try several lines, chosen at random
- Keep line that best separates data
  - information gain
- Recurse

- feature vectors are $x, y$ coordinates: $\mathbf{v} = [x, y]^T$
- split functions are lines with parameters $a, b$: $f_n(\mathbf{v}) = ax + by$
- threshold determines intercepts: $t_n$
- four classes: purple, blue, red, green

Slide credits to J. Shotton
Toy Learning Example

- Try several lines, chosen at random
- Keep line that best separates data — information gain
- Recurse

- feature vectors are $x$, $y$ coordinates: $\mathbf{v} = [x, y]^T$
- split functions are lines with parameters $a$, $b$: $f_n(\mathbf{v}) = ax + by$
- threshold determines intercepts: $t_n$
- four classes: purple, blue, red, green

Slide credits to J. Shotton
Generalisation

Overfit

Reasonably Smooth
Generalisation

tree $t_1$  tree $t_2$  ...... tree $t_T$

Reasonably Smooth
Randomized Tree Learning

- **Train data**: \( \{ (x_i, y_i) \}_{i=1}^N \) \( y_i \in \{1, 2, \ldots, C\} \)
- **Recursive algorithm**
  - set \( I_n \) of training examples that reach node \( n \) is split:
    
    \[
    \begin{align*}
    \text{left split} & \quad I_1 = \{ i \in I_n \mid f(v_i) < t \} \\
    \text{right split} & \quad I_r = I_n \setminus I_1
    \end{align*}
    \]
- **Features** \( f \) and thresholds \( t \) chosen at random
- **Choose** \( f \) and \( t \) to maximize gain in information

\[
\Delta E = -\frac{|I_1|}{|I_n|} E(I_1) - \frac{|I_r|}{|I_n|} E(I_r)
\]
A Forest of Trees: Learning

- Forest is an ensemble of decision trees

- Bagging (Bootstrap AGGregatING)
  - For each set, it randomly draws examples from the uniform dist. allowing duplication and missing.
A Forest of Trees: Recognition

- Forest is an ensemble of decision trees

classification is

\[ P(c|v) = \sum_{t=1}^{T} P_t(c|v) \]
Fast Evaluation

- Flat structure
- Tree structure

5 times speed up
Demo video: Fast evaluation
Random Forest Codebook
Video Cuboid Features

Video pixel $i$ gives cuboid $p$
(13x13x19 pixels in experiments)

Tree split function

If $f(p) > \text{threshold}$
goto Left
Else
  goto Right

$f(p) = p_{x_1,y_1,t_1}$
$f(p) = p_{x_1,y_1,t_1} - p_{x_2,y_2,t_2}$
$f(p) = p_{x_1,y_1,t_1} + p_{x_2,y_2,t_2}$

$x \in \mathbb{R}^D$
Randomized Forest Codebook

Example Cuboids
Histogram of Randomized Forest Codewords

“bag of words”

Extremely Fast
Powerful Discriminative Codebook
Matlab Demo: 

Random Forest Codebook
### Action Categorisation Result

- **Action categorization accuracy (%) on KTH dataset**
  - 6 types of actions, 25 subjects of 4 scenarios by leave-one-out protocol

<table>
<thead>
<tr>
<th>Action</th>
<th>Walking</th>
<th>Running</th>
<th>Jogging</th>
<th>Handwaving</th>
<th>Handclapping</th>
<th>Boxing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>.79</td>
<td>.01</td>
<td>.14</td>
<td>.00</td>
<td>.06</td>
<td>.00</td>
</tr>
<tr>
<td>Running</td>
<td>.01</td>
<td>.88</td>
<td>.11</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td>Jogging</td>
<td>.11</td>
<td>.36</td>
<td>.52</td>
<td>.00</td>
<td>.01</td>
<td>.00</td>
</tr>
<tr>
<td>Handwaving</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.93</td>
<td>.01</td>
<td>.06</td>
</tr>
<tr>
<td>Handclapping</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.77</td>
<td>.23</td>
</tr>
<tr>
<td>Boxing</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**Images:**
- Walking
- Running
- Jogging
- Handwaving
- Handclapping
- Boxing
Matlab Demo:

Action Categorisation
Take-home Message

• Use of visual codebook greatly facilitates image and video categorisation tasks.
• The RF codebook is extremely fast as it only uses a small number of features in trees.
  – c.f. K-means codebook is in a flat structure. It compares an input with every codewords, which is time-demanding.
• The RF codebook is also a powerful discriminative codebook. It combines multiple decision trees showing good generalisation.
  – c.f. K-means is an unsupervised learning method.
Questions?
Action Recognition Result

- **Action categorization accuracy (%) on KTH dataset**
  - 6 types of actions, 25 subjects of 4 scenarios by leave-one-out protocol

```
<table>
<thead>
<tr>
<th>Action</th>
<th>walking</th>
<th>running</th>
<th>jogging</th>
<th>hand-waving</th>
<th>hand-clapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>walking</td>
<td>.79</td>
<td>.01</td>
<td>.14</td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td>running</td>
<td>.01</td>
<td>.88</td>
<td>.11</td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td>jogging</td>
<td>.11</td>
<td>.36</td>
<td>.52</td>
<td>.00</td>
<td>.01</td>
</tr>
<tr>
<td>hand-waving</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.93</td>
<td>.01</td>
</tr>
<tr>
<td>hand-clapping</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.77</td>
</tr>
<tr>
<td>boxing</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>
```

- quantisation effect!
Pyramid Match Kernel

• PMK acts on semantic texton histogram
  – matches P and Q in *learned* hierarchical histogram space
  – deeper node matches are more important

\[
K(P, Q) = \sum_{d=1}^{D} \frac{1}{2^{D-d+1}} (I_d - I_{d+1})
\]

- increased similarity at depth \(d\)
- depth weight

- leaf nodes
- split nodes

- \(d=1\)
- \(d=2\)
- \(d=3\)
Categorisation

• Histogram matching of a pair of videos $P$ and $Q$ is

$$\mathcal{I}_d(P, Q) = \sum_{n=1}^{b_d} \min (H_d(P^{(n)}), H_d(Q^{(n)}))$$

• Based on HMK, we can use various classifiers such as kNN, SVM, Naïve-Bayes classifiers or pLSA, LDA.

NN classification by

$$\arg \max_c \mathcal{I}_d (P_c, Q)$$
Action Recognition Result

- Action categorization accuracy (%) on KTH dataset
  - 6 types of actions, 25 subjects of 4 scenarios by leave-one-out protocol

![Image of action categorization results]

<table>
<thead>
<tr>
<th>Action</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>box</td>
<td>.98</td>
</tr>
<tr>
<td>hclp</td>
<td>1.0</td>
</tr>
<tr>
<td>hwav</td>
<td>0.97</td>
</tr>
<tr>
<td>jog</td>
<td>0.90</td>
</tr>
<tr>
<td>run</td>
<td>0.12</td>
</tr>
<tr>
<td>walk</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Note: The table shows the accuracy percentages for each action category.