Fast Template Evaluation using Vector Quantization

Amin Sadeghi, David Forsyth (msadeghi2.daf@illinois.edu)

**Motivation**
- Object detection systems run many templates all over the image. Each template is sensitive to a certain pattern and computes a score at each location.

**Fast Evaluation of Legacy Templates**
- **HOG Features**
  - Use Haar-like features for object detection.
  - Fast computation of HOG features.

- **Vector Quantization For Compression**
  - Replace each cell with an index. (e.g., 0-255)
  - More arithmetic operations in a second.
  - No need for floating point, use 16-bit fixed point operations.
  - No multiplication is involved; only addition.

- **Cascade of Templates**
  - Template evaluation is always a bottleneck.
  - Prior techniques to speed up template evaluation:
    - Random-access score computation
    - Faster operations, fixed point operations
    - Compute look-up tables from legacy detectors
    - About 100 times memory saving while training
    - Can access all training examples on RAM while training

**Fast Deformation Estimate**
- Instead of global search over a range of deformations, search locally using hit clipping.

**Conclusion**
- One order of magnitude speedup for template evaluation
- Ten orders of magnitude speedup for object detection

**Properties**
- Random-access score computation
- Compatible with Cascade
- No multiplications, only additions
- Faster operations. Fixed point operations
- Backward Compatibility
- Compute look-up tables from legacy detectors
- About 100 times memory saving while training
- Can access all training examples on RAM while training

**Further Applications**
- Faster training
- Real-time object detection

**Further Speedup**
- Why compute HOG features?

**Code Library**
http://vision.cs.illinois.edu/fvq/

**Fast Code Available for**
- Deformable Part Model Object Detection
- Fast Exemplar Template Evaluation
- General Template Evaluation

- Implemented in MATLAB/C++ using fast mex Els
  - Parallel processing and SIMD operations
  - Start by downloading the library and running demo.m

**Computation Cost Model**
- **HOG features computation**
- **Per Image preprocess**
- **Per Category preprocess**
- **Per (Image x Category)** processes

**PASCAL 2007 Dataset**
- Average end-to-end time to detect 20 object categories

**Results**
- **Computation Time vs. Estimation Error**
  - PCA
  - VQ

**Fast Evaluation of Legacy Templates**
- **HOG Features**
  - Image
  - HOG features

- **Vector Quantization For Compression**
  - Replace each cell with an index. (e.g., 0-255)

- **Cascade of Templates**
  - Random-access score computations allows for efficient cascade implementation.

- **Fast Deformation Estimate**
  - Instead of global search over a range of deformations, search locally using hit clipping.

**Fast Template Evaluation using Vector Quantization**

**Computation Time vs. Estimation Error**

<table>
<thead>
<tr>
<th>Computation Cost Model</th>
<th>HOG features computation</th>
<th>Per Image preprocess</th>
<th>Per Category preprocess</th>
<th>Per (Image x Category) processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average end-to-end time to detect 20 object categories</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Motivation**
- Object detection systems run many templates all over the image. Each template is sensitive to a certain pattern and computes a score at each location.

**Fast Evaluation of Legacy Templates**
- **HOG Features**
  - Use Haar-like features for object detection.
  - Fast computation of HOG features.

- **Vector Quantization For Compression**
  - Replace each cell with an index. (e.g., 0-255)
  - More arithmetic operations in a second.
  - No need for floating point, use 16-bit fixed point operations.
  - No multiplication is involved; only addition.

- **Cascade of Templates**
  - Template evaluation is always a bottleneck.
  - Prior techniques to speed up template evaluation:
    - Random-access score computation
    - Faster operations, fixed point operations
    - Compute look-up tables from legacy detectors
    - About 100 times memory saving while training
    - Can access all training examples on RAM while training

**Fast Deformation Estimate**
- Instead of global search over a range of deformations, search locally using hit clipping.

**Conclusion**
- One order of magnitude speedup for template evaluation
- Ten orders of magnitude speedup for object detection

**Properties**
- Random-access score computation
- Compatible with Cascade
- No multiplications, only additions
- Faster operations. Fixed point operations
- Backward Compatibility
- Compute look-up tables from legacy detectors
- About 100 times memory saving while training
- Can access all training examples on RAM while training

**Further Applications**
- Faster training
- Real-time object detection

**Further Speedup**
- Why compute HOG features?

**Code Library**
http://vision.cs.illinois.edu/fvq/

**Fast Code Available for**
- Deformable Part Model Object Detection
- Fast Exemplar Template Evaluation
- General Template Evaluation

- Implemented in MATLAB/C++ using fast mex Els
  - Parallel processing and SIMD operations
  - Start by downloading the library and running demo.m

**Computation Cost Model**
- **HOG features computation**
- **Per Image preprocess**
- **Per Category preprocess**
- **Per (Image x Category)** processes

**PASCAL 2007 Dataset**
- Average end-to-end time to detect 20 object categories

**Results**
- **Computation Time vs. Estimation Error**
  - PCA
  - VQ