A Compact and Discriminative Face Track Descriptor

Omkar M Parkhi, Karen Simonyan, Andrea Vedaldi, Andrew Zisserman
Recognising and verifying faces in videos

Recognition

Verification

same
different
VF$^2$: a new compact face track descriptor

**Face track**: sequence of face detections in consecutive frames.

- **Discriminative**
- **Useful for different tasks (Recognition, Verification)**
- **Extremely compact**
Large scale face retrieval

Example of a typical target dataset

- 5 years of evening news programs
- 10,000 hrs of broadcast
- 20 Million frames,
- 2.1 Million face tracks
- Real time performance

http://www.robots.ox.ac.uk/~vgg/research/on-the-fly/

- 30 frames per track on average
- Typical 4000D descriptor → 1 TB
- Our descriptor → 270 MB
Outline

1. Dense feature computation
2. Fisher Vector encoding
3. Video and jittered pooling
4. Compression by metric learning
5. Binarisation
6. Results
1. Dense feature computation

- **Input:** a face track
  - Aligned or unaligned
  - No facial landmarks required (eyes, nose, etc.)

- **Output:** a set of local features
  - Extracted from all frames
  - Dense **RootSIFT** at multiple scales
  - 64-D PCA
1. Dense feature computation
2. Fisher Vector encoding
3. Video and jittered pooling
4. Compression by metric learning
5. Binarisation
6. Results
2. Fisher Vector encoding

GMM

Dense SIFT

\[ \mathbf{v}_k = \frac{1}{M \sqrt{\pi_k}} \sum_{i=1}^{M} \gamma_k(\mathbf{x}_i) \frac{\mathbf{x}_i - \mu_k}{\sigma_i} \]

\[ \mathbf{u}_k = \frac{1}{M \sqrt{2\pi_k}} \sum_{i=1}^{M} \gamma_k(\mathbf{x}_i) \left( \frac{\mathbf{x}_i - \mu_k}{\sigma_i} - 1 \right)^2 \]

FV encoding \( \Phi = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{u}_1 \\ \mathbf{v}_2 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{v}_K \\ \mathbf{u}_K \end{bmatrix} \)

+ sqrt-L2 normalisation

GMM

Hard Assignment

Gaussians \((\mu_k, \Sigma_k)\)

Assignment Hard

Dense SIFT

\[ \mathbf{x}_i \]

\[ \gamma_k(\mathbf{x}_i) \]

\[ \mu_k \]
2. Fisher Vector Encoding

Gaussian components as part detectors

Spatial (x,y) Augmentation
1. Dense feature computation

2. Fisher Vector encoding

3. Video and jittered pooling

4. Compression by metric learning

5. Binarisation

6. Results
3. Video and jittered pooling

- Typically each frame is pooled independently
- Complex inference procedures combining multiple descriptors
- Large memory footprint

[Sivic et al. CVPR 09, Everingham et. al IVC 09,, Wolf et al. CVPR 2011]
3. Video and jittered pooling

- Single descriptor per track
  - Smaller memory footprint
  - Easy to use
  - Improved performance

[Application to Action Recognition: Oneata, Verbeek, Schmid ICCV 2013]
3. Video and jittered pooling

- Data augmentation
  - Data augmentation without training set increase
  - Improvement in the performance

[Paulin et al. CVPR 2014]
Outline

1. Dense feature computation
2. Fisher Vector encoding
3. Video and jittered pooling
4. Compression by metric learning
5. Binarisation
6. Results
4. Metric Learning

Learn to discriminate faces

\[ d_W^2(x, y) = \| Wx - Wy \|^2 \]

same person

\[ d_W^2(u, v) = \| Wu - Wv \|^2 < b \]

different people

[Simonyan, Parkhi, Vedaldi, Zisserman BMVC 2013]
Outline

1. Dense feature computation

2. Fisher Vector encoding

3. Video and jittered pooling

4. Compression by metric learning

5. Binarisation

6. Results
5. Binarisation

Parseval Tight Frame

- Low-dimensional real-valued descriptor $\rightarrow$ high dimensional binary
- 4x decrease in memory footprint (128D real $\rightarrow$ 1024D binary)
- Fast distance computation
- Alternative binarisation methods could be used

[Jégou et al. ICASSP 2012, Simonyan et al. PAMI 2014]
Outline

1. Dense feature computation
2. Fisher Vector encoding
3. Video and jittered pooling
4. Compression by metric learning
5. Binarisation
6. Results
Face Verification

YouTube Faces Dataset

- **Face verification in videos**
  - 3,425 videos of 1,595 celebrities
  - Videos collected from internet
  - Wide pose, expression and illumination variation
  - 10 splits of 600 pairs of videos
  - **Restricted setting:** Use provided pairs
  - **Unrestricted setting:** Free to form own pairs.

[Wolf, Hassner, Moaz CVPR 2011]
YouTube Faces Dataset

Face Verification

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Pool (Soft assignment FV)</td>
<td>17.3</td>
</tr>
<tr>
<td>Video Pool (Soft assignment FV)</td>
<td>15</td>
</tr>
<tr>
<td>Video Pool hard assignment fv</td>
<td>16.2</td>
</tr>
<tr>
<td>Video Pool + Jittered Pool</td>
<td>14.2</td>
</tr>
<tr>
<td>Video Pool. + Binar. 1024 bit + jitt.</td>
<td>13.4</td>
</tr>
<tr>
<td>Video Pool. + Joint sim. + jitt.</td>
<td>12.3</td>
</tr>
</tbody>
</table>

Error
## Face Verification

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGBS &amp; SVM-APEM FUSION</td>
<td>21.2</td>
</tr>
<tr>
<td>STFRD &amp; PMML</td>
<td>19.9</td>
</tr>
<tr>
<td>VSOF &amp; OSS (Adaboost)</td>
<td>20</td>
</tr>
<tr>
<td>DDML (Combined)</td>
<td>18.5</td>
</tr>
<tr>
<td>VF(^2) 1024D (binary)</td>
<td>13.4</td>
</tr>
<tr>
<td>VF(^2) 256D</td>
<td>12.3</td>
</tr>
<tr>
<td>Deep Face (facebook.com)</td>
<td>8.6</td>
</tr>
</tbody>
</table>

Requires additional training data.
Oxford Buffy Dataset

Weakly supervised face classification

“Buffy The Vampire Slayer”
- Face tracks from 7 episodes of season 5.
- Both frontal and profile detections
- Weak supervision from transcript and subtitles
- Multi Class classification for every episode

[Everingham et al. IVC 2009, Sivic et al. CVPR 2009]
Oxford Buffy Dataset

Weakly supervised classification

Sivic et al. (HOG RBF MKL) 0.81
VF\(^2\) (GMMs trained on Buffy) 0.81
VF\(^2\) (GMMs trained on YTF) 0.8
VF\(^2\) (GMMs trained on YTF) + Jitt. Pool 1024D 0.86
VF\(^2\) (GMMs trained on YTF 2048b) 0.82

Avg. AP

0.79 0.808 0.825 0.843 0.86
Very simple yet powerful face track descriptor

- Track descriptor in 128 bytes
  - Face landmarks and alignment not required
  - One descriptor per track

- State of the art/comparable results on multiple tasks
  - YouTube Faces Dataset
  - Oxford Buffy Dataset

- Can be trained with very small amount of data
  - Extremely easy to compute

- Code online soon.

Questions?