

Fisher Vector Faces (FVF) in the Wild

Karén Simonyan, Omkar Parkhi,
Andrea Vedaldi, Andrew Zisserman

Visual Geometry Group, University of Oxford

Objective

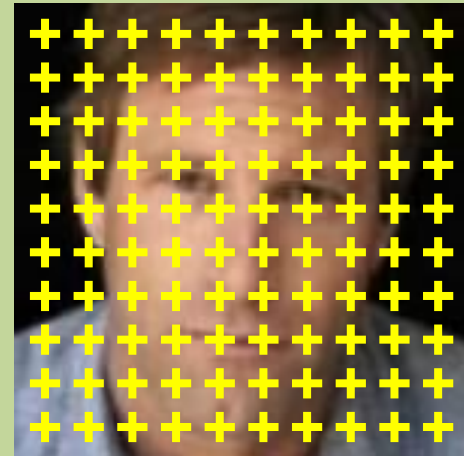
Face descriptor for recognition:

- dense sampling
- relevant face parts learnt automatically
- compact and discriminative

Conventional approach
(describe landmarks)

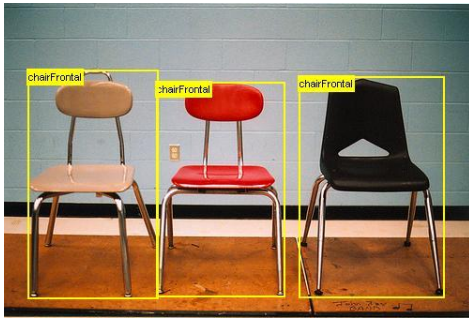


Our approach
(describe everything)



Motivation

- State-of-the-art image recognition pipeline:
 - **dense SIFT → Fisher vector encoding → linear SVM**
 - very competitive on (generic) image recognition tasks:
Caltech 101/256, PASCAL VOC, ImageNet ILSVRC
- Can it be applied to faces? Yes!



Application – Face Verification

«Is it the same person in both images?»



SAME



DIFFERENT

Labelled Faces in the Wild (LFW) dataset

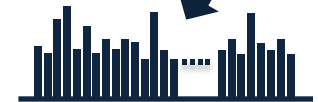
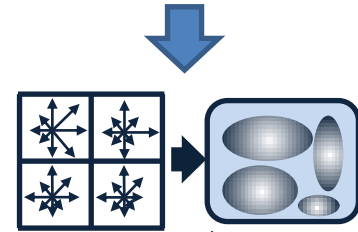
- large-scale: 13K images, 5.7K people
- collected using Viola-Jones face detector
- high variability in appearance
- several evaluation settings (restricted, unrestricted)

Pipeline Overview

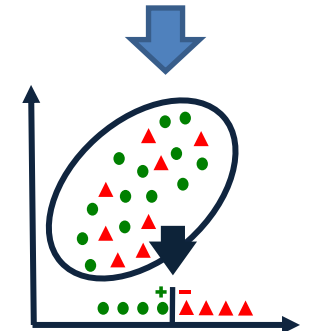
- Input: face image, e.g.
 - LFW + face alignment¹
 - pre-aligned: LFW-funneled, LFW-a
 - no alignment: just Viola-Jones detection!
- Output: Fisher Vector Face descriptor (FVF)
 - discriminative
 - compact



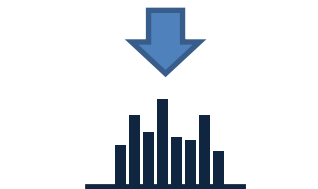
face image



face FV extraction



discriminative
projection



compact descriptor

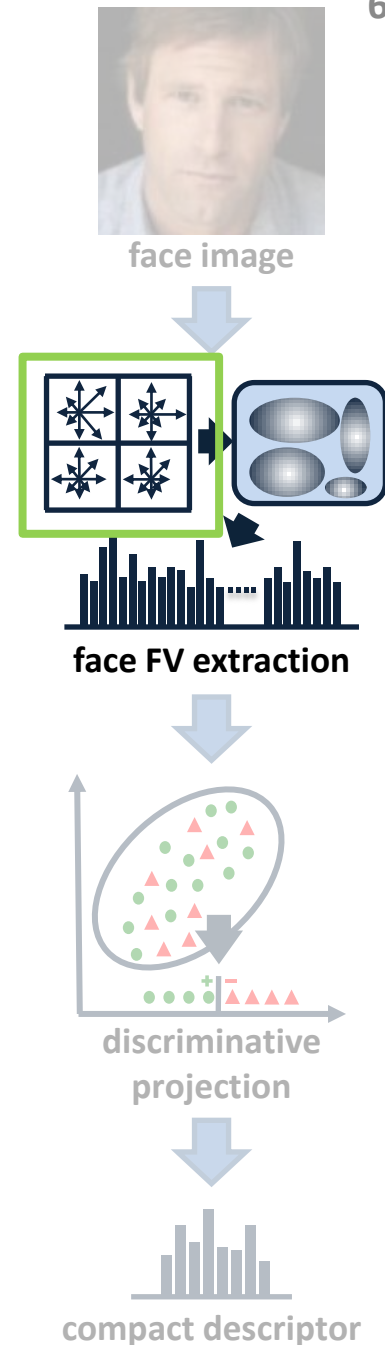
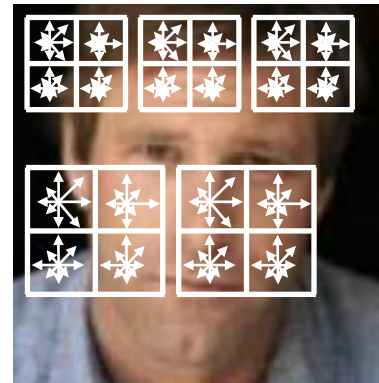
[1] "Taking the bite out of automatic naming of characters in TV video",
M. Everingham, J. Sivic, and A. Zisserman. IVC 2009.

Dense Features

face image → set of local features

Dense SIFT

- dense scale-space grid:
1 pix step, 5 scales
- 24x24 patch size
- rootSIFT¹ – explicit Hellinger kernel map
- 64-D PCA-rootSIFT
- augmented with (x,y): 66-D



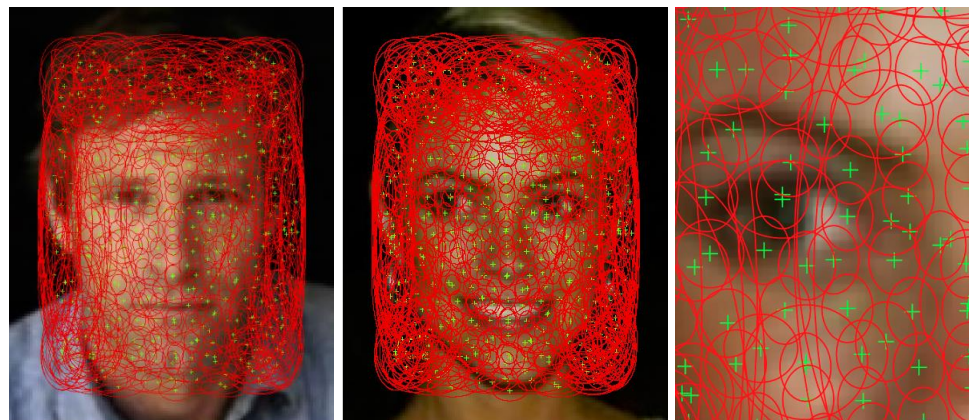
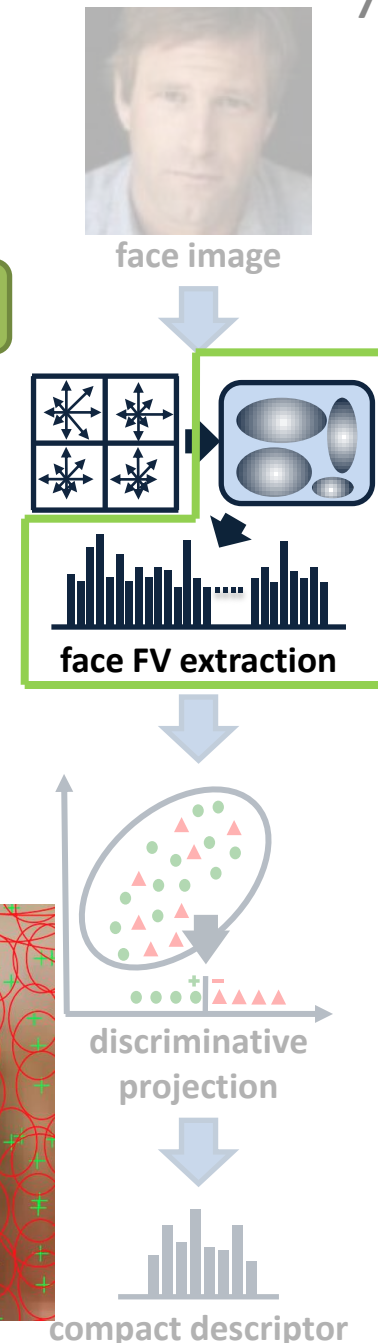
[1] “Three things everyone should know to improve object retrieval”,
R. Arandjelovic and A. Zisserman. CVPR, 2012.

Face Fisher Vector

set of local features \rightarrow high-dim Fisher vector

Fisher Vector (FV) encoding¹

- describes a set of local features in a single vector
- diagonal-covariance GMM as a codebook
 - appearance: SIFT
 - location: (x,y)
- GMM can be seen as a face model



ellipses – means & variances
of GMM's (x,y) components

Face Fisher Vector

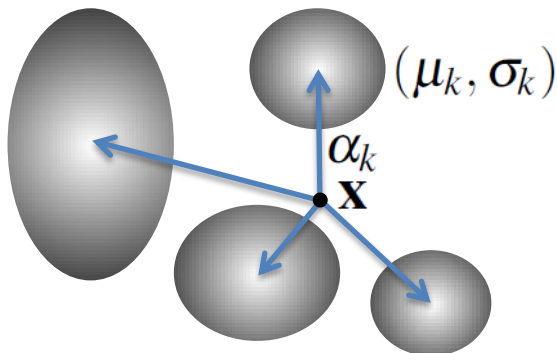
set of local features \rightarrow high-dim Fisher vector

- Image FV – normalised sum of feature FVs
- Feature FV – feature space location statistics:

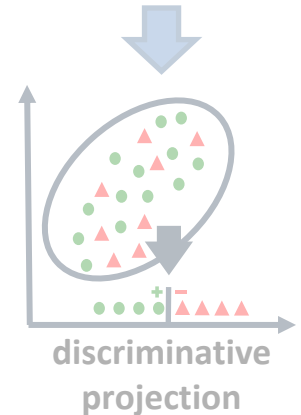
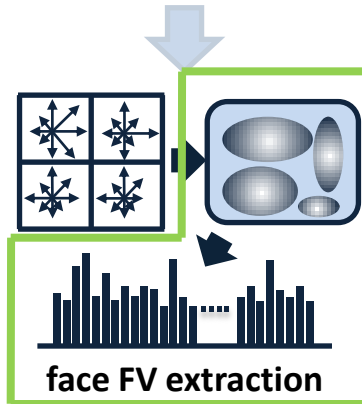
1st order stats (k-th Gaussian): $\Phi_k^{(1)} \sim \alpha_k \left(\frac{\mathbf{x} - \mu_k}{\sigma_k} \right)$

2nd order stats (k-th Gaussian): $\Phi_k^{(2)} \sim \alpha_k \left(\frac{(\mathbf{x} - \mu_k)^2}{\sigma_k^2} - 1 \right)$

soft-assignment to GMM



face image



Face Fisher Vector

set of local features \rightarrow high-dim Fisher vector

- Image FV – normalised sum of feature FVs
- Feature FV – feature space location statistics:

1st order stats (k-th Gaussian): $\Phi_k^{(1)} \sim \alpha_k \left(\frac{\mathbf{x} - \mu_k}{\sigma_k} \right)$

2nd order stats (k-th Gaussian): $\Phi_k^{(2)} \sim \alpha_k \left(\frac{(\mathbf{x} - \mu_k)^2}{\sigma_k^2} - 1 \right)$

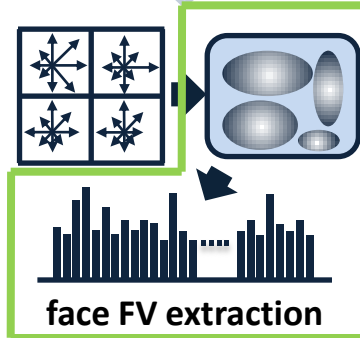
stacking

$$\phi(\mathbf{x}) = \left[\underbrace{\Phi_1^{(1)}}_{66\text{-D}}, \underbrace{\Phi_1^{(2)}}_{66\text{-D}}, \dots, \underbrace{\Phi_K^{(1)}}_{66\text{-D}}, \underbrace{\Phi_K^{(2)}}_{66\text{-D}} \right]$$

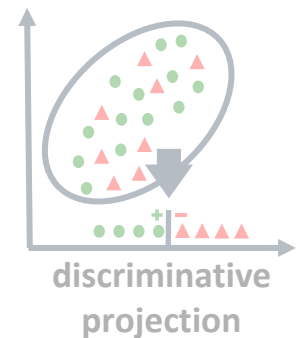
FV dimensionality: $66 \times 2 \times 512 = 67,584$
(for a mixture of 512 Gaussians)



face image



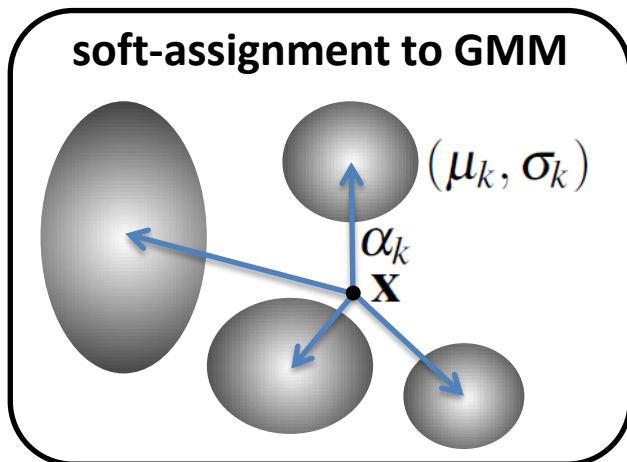
face FV extraction



discriminative projection



compact descriptor



soft-assignment to GMM

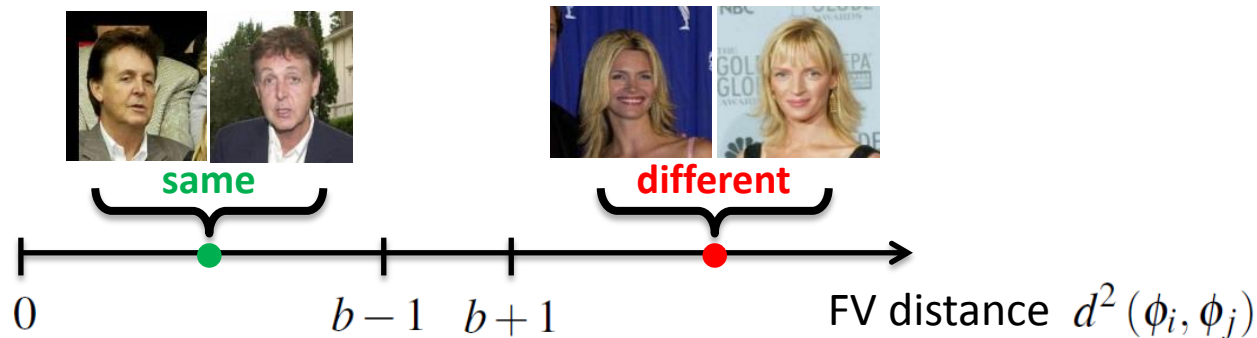
Distance Learning

high-dim FV \rightarrow low-dim face descriptor

- Large-margin distance constraints:

$$y_{ij} (b - d^2(\phi_i, \phi_j)) > 1$$

$y_{ij} = 1$ iff (i,j) is the same person, ϕ_i, ϕ_j – FV



- Distance models:

- low-rank Mahalanobis
- joint distance-similarity
- weighted Euclidean

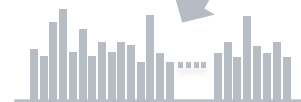
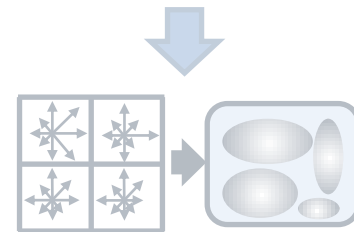
$$W = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$$

$$W = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix} \quad V = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$$

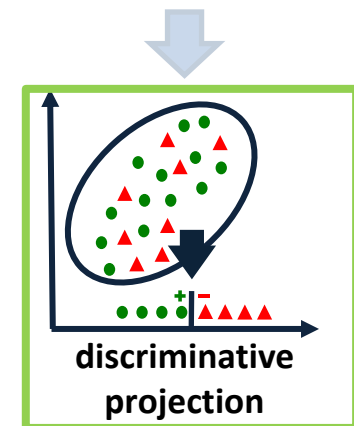
$$U = \begin{pmatrix} \text{---} \end{pmatrix}$$



face image



face FV extraction



discriminative projection



compact descriptor

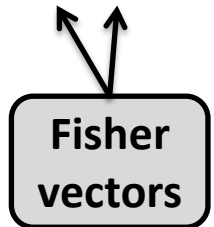
Projection Learning

- Low-rank Mahalanobis distance (projection W):

$$d_W^2(\phi_i, \phi_j) = \|W\phi_i - W\phi_j\|_2^2 = (\phi_i - \phi_j)^T W^T W (\phi_i - \phi_j)$$

$$W = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$$

- Large-margin objective: $\arg \min_W \sum_{i,j} \max [1 - y_{ij} (b - d_W^2(\phi_i, \phi_j)), 0]$
 - regularisation by $W \in \mathbb{R}^{p \times d}$, $p \ll d$
 - stochastic sub-gradient solver
 - initialised by PCA-whitening



- Models dependencies between FV elements
- Explicit dimensionality reduction



- Non-convex

Joint Distance-Similarity Learning

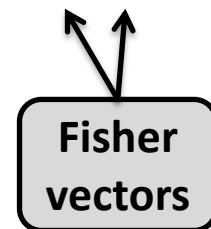
- Difference of low-rank distance and inner product¹ :

$$d_{W,V}^2(\phi_i, \phi_j) = \|W\phi_i - W\phi_j\|_2^2 - \langle V\phi_i, V\phi_j \rangle =$$

$$(\phi_i - \phi_j)^T W^T W (\phi_i - \phi_j) - \phi_i^T V^T V \phi_j$$

$W = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$
 $V = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$

- Large-margin objective: $\arg \min_{W,V} \sum_{i,j} \max [1 - y_{ij} (b - d_{W,V}^2(\phi_i, \phi_j)), 0]$
 - stochastic sub-gradient solver (as before)



+

- Models dependencies between FV elements
- More complex decision (distance) function

-

- Two low-dim representations (W & V projections)
- Non-convex

[1] "Blessing of dimensionality: high dimensional feature and its efficient compression for face verification", D. Chen, X. Cao, F. Wen, and J. Sun. CVPR, 2013.

Distance Learning

- Weighted Euclidean distance (diagonal Mahalanobis)

$$d_u^2(\phi_i, \phi_j) = \sum_k u_k \left(\phi_i^{(k)} - \phi_j^{(k)} \right)^2, \quad u_k \geq 0 \forall k$$

$$U = \text{diag}(u_1, \dots, u_n)$$

- Large-margin (SVM-like) objective:

$$\arg \min_{u_k \geq 0} \sum_{i,j} \max [1 - y_{ij} (b - d_u^2(\phi_i, \phi_j)), 0]$$

Fisher
vectors

+

- Convex, fast to train
- Less parameters → less training data needed

-

- Doesn't model dependencies between FV elements
- No dimensionality reduction

Effect of Parameters

SIFT density	GMM Size	Spatial Aug.	Desc. Dim.	Distance Function	Hor. Flip.	ROC-EER, %
2 pix	256		32768	diag. metric		89.0
2 pix	256	✓	33792	diag. metric		89.8
2 pix	512	✓	67584	diag. metric		90.6
1 pix	512	✓	67584	diag. metric		90.9
1 pix	512	✓	128	low-rank PCA-whitening		78.6
1 pix	512	✓	128	low-rank Mah. metric		91.4
1 pix	512	✓	256	low-rank Mah. metric		91.0
1 pix	512	✓	128	low-rank Mah. metric	✓	92.0
1 pix	512	✓	2×128	low-rank joint metric-sim.		92.2
1 pix	512	✓	2×128	low-rank joint metric-sim.	✓	93.1

Effect of FV parameters on accuracy @ ROC-EER¹ (LFW-unrestricted)

[1] “Is that you? Metric learning approaches for face identification”, Guillaumin et al., ICCV 2009.

Effect of Parameters

SIFT density	GMM Size	Spatial Aug.	Desc. Dim.	Distance Function	Hor. Flip.	ROC-EER, %
2 pix	256		32768	diag. metric		89.0
2 pix	256	✓	33792	diag. metric		89.8
2 pix	512	✓	67584	diag. metric		90.6
1 pix	512	✓	67584	diag. metric		90.9
1 pix	512	✓	128	low-rank PCA-whitening		78.6
1 pix	512	✓	128	low-rank Mah. metric		91.4
1 pix	512	✓	256	low-rank Mah. metric		91.0
1 pix	512	✓	128	low-rank Mah. metric	✓	92.0
1 pix	512	✓	2×128	low-rank joint metric-sim.		92.2
1 pix	512	✓	2×128	low-rank joint metric-sim.	✓	93.1

Effect of FV parameters on accuracy @ ROC-EER¹ (LFW-unrestricted)

Performance increases with:

- spatial augmentation, more Gaussians, higher density

Effect of Parameters

SIFT density	GMM Size	Spatial Aug.	Desc. Dim.	Distance Function	Hor. Flip.	ROC-EER, %
2 pix	256		32768	diag. metric		89.0
2 pix	256	✓	33792	diag. metric		89.8
2 pix	512	✓	67584	diag. metric		90.6
1 pix	512	✓	67584	diag. metric		90.9
1 pix	512	✓	128	low-rank PCA-whitening		78.6
1 pix	512	✓	128	low-rank Mah. metric		91.4
1 pix	512	✓	256	low-rank Mah. metric		91.0
1 pix	512	✓	128	low-rank Mah. metric	✓	92.0
1 pix	512	✓	2×128	low-rank joint metric-sim.		92.2
1 pix	512	✓	2×128	low-rank joint metric-sim.	✓	93.1

Effect of FV parameters on accuracy @ ROC-EER¹ (LFW-unrestricted)

Performance increases with:

- spatial augmentation, more Gaussians, higher density
- discriminative projection (also **500-fold** dimensionality reduction)

Effect of Parameters

SIFT density	GMM Size	Spatial Aug.	Desc. Dim.	Distance Function	Hor. Flip.	ROC-EER, %
2 pix	256		32768	diag. metric		89.0
2 pix	256	✓	33792	diag. metric		89.8
2 pix	512	✓	67584	diag. metric		90.6
1 pix	512	✓	67584	diag. metric		90.9
1 pix	512	✓	128	low-rank PCA-whitening		78.6
1 pix	512	✓	128	low-rank Mah. metric		91.4
1 pix	512	✓	256	low-rank Mah. metric		91.0
1 pix	512	✓	128	low-rank Mah. metric	✓	92.0
1 pix	512	✓	2×128	low-rank joint metric-sim.		92.2
1 pix	512	✓	2×128	low-rank joint metric-sim.	✓	93.1

Effect of FV parameters on accuracy @ ROC-EER¹ (LFW-unrestricted)

Performance increases with:

- spatial augmentation, more Gaussians, higher density
- discriminative projection (also **500-fold** dimensionality reduction)
- averaging across 4 combinations of horizontally flipped faces

Effect of Parameters

SIFT density	GMM Size	Spatial Aug.	Desc. Dim.	Distance Function	Hor. Flip.	ROC-EER, %
2 pix	256		32768	diag. metric		89.0
2 pix	256	✓	33792	diag. metric		89.8
2 pix	512	✓	67584	diag. metric		90.6
1 pix	512	✓	67584	diag. metric		90.9
1 pix	512	✓	128	low-rank PCA-whitening		78.6
1 pix	512	✓	128	low-rank Mah. metric		91.4
1 pix	512	✓	256	low-rank Mah. metric		91.0
1 pix	512	✓	128	low-rank Mah. metric	✓	92.0
1 pix	512	✓	2×128	low-rank joint metric-sim.		92.2
1 pix	512	✓	2×128	low-rank joint metric-sim.	✓	93.1

Effect of FV parameters on accuracy @ ROC-EER¹ (LFW-unrestricted)

Performance increases with:

- spatial augmentation, more Gaussians, higher density
- discriminative projection (also **500-fold** dimensionality reduction)
- averaging across 4 combinations of horizontally flipped faces
- combined distance-similarity score function

Effect of Face Alignment

- Robust w.r.t. alignment and crop:
 - LFW \rightarrow align & crop¹: 92.0%
 - LFW-deep-funneled² \rightarrow 150×150 crop: 92.0%
 - LFW-funneled³ \rightarrow 150×150 crop: 91.7%
 - LFW \rightarrow Viola-Jones crop (**no alignment**): 90.9%
- Good results without alignment
 - just run Viola-Jones and compute FVF!
 - might not hold for other datasets
- Setting: LFW-unrestricted, projection learning, horiz. flipping

[1] “Taking the bite out of automatic naming of characters in TV video”, Everingham et al., IVC 2009.

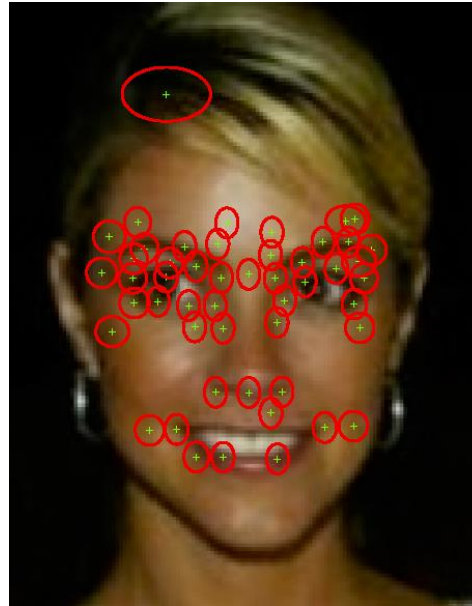
[2] “Learning to align from scratch”, Huang et al., NIPS 2012

[3] “Unsupervised joint alignment of complex images”, Huang et al., ICCV 2007

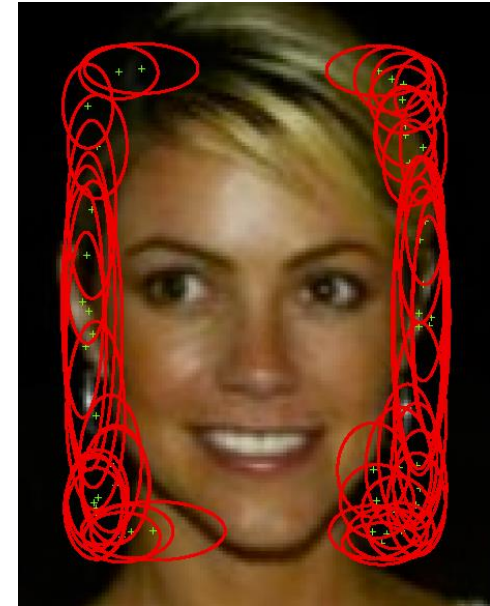
Learnt Model Visualisation



all Gaussians



**important
(top-50 Gaussians)**



**irrelevant
(bottom-50 Gaussians)**

Gaussian ranking (for visualisation):

GMM component \rightarrow FV sub-vector \rightarrow W sub-matrix \rightarrow its energy

**dimensionality
reduction projection**

$W =$

**1st
Gaussian**

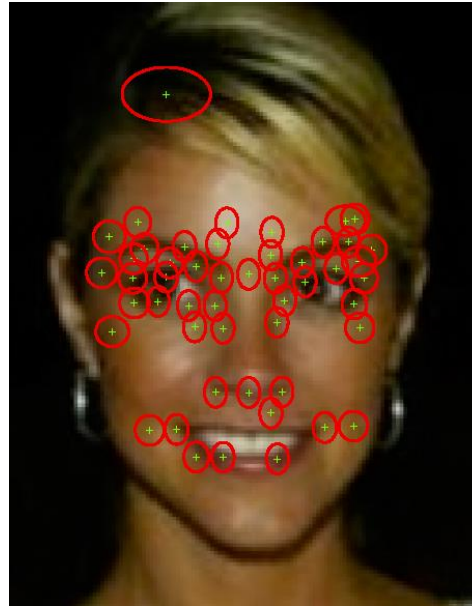
**2nd
Gaussian**

**512th
Gaussian**

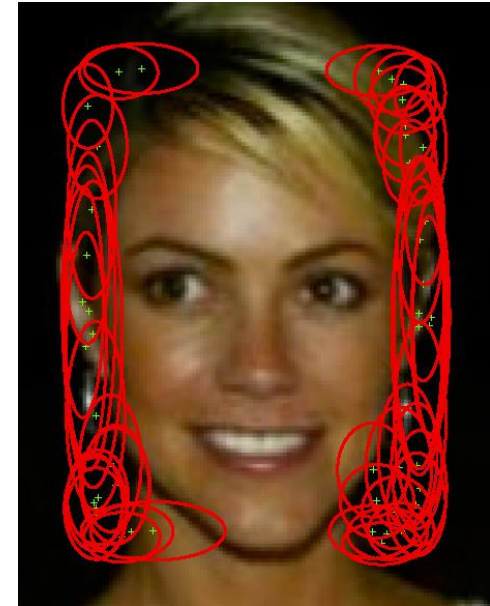
Learnt Model Visualisation



all Gaussians



important
(top-50 Gaussians)



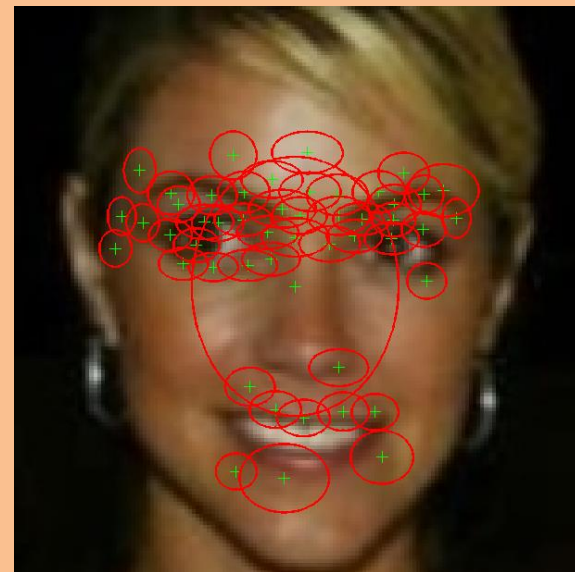
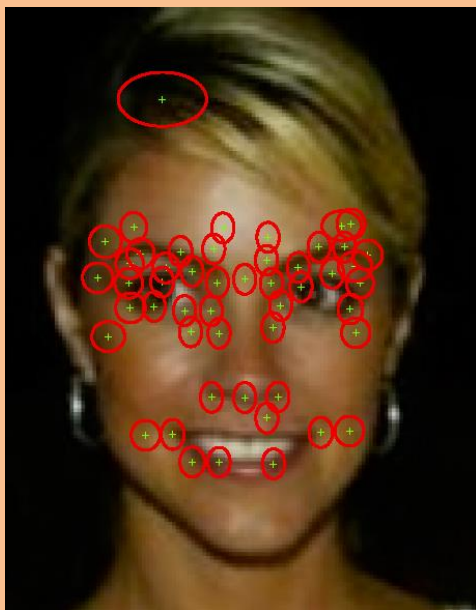
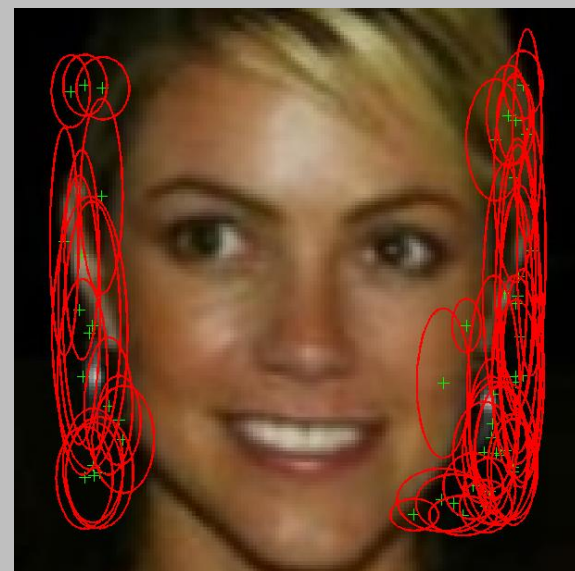
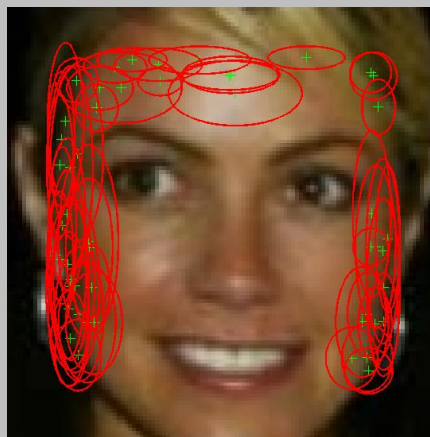
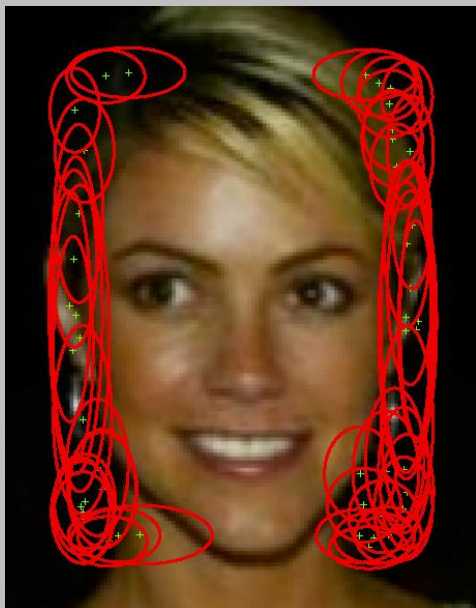
irrelevant
(bottom-50 Gaussians)

- High-ranked Gaussians (centre)
 - **match facial features** (weren't explicitly trained to do so)
 - fine localisation (low spatial variance)
- Low-ranked Gaussians (right)
 - cover background areas
 - loose localisation (high spatial variance)

LFW → alignment

LFW, no alignment
(Viola-Jones box)

LFW-funneled

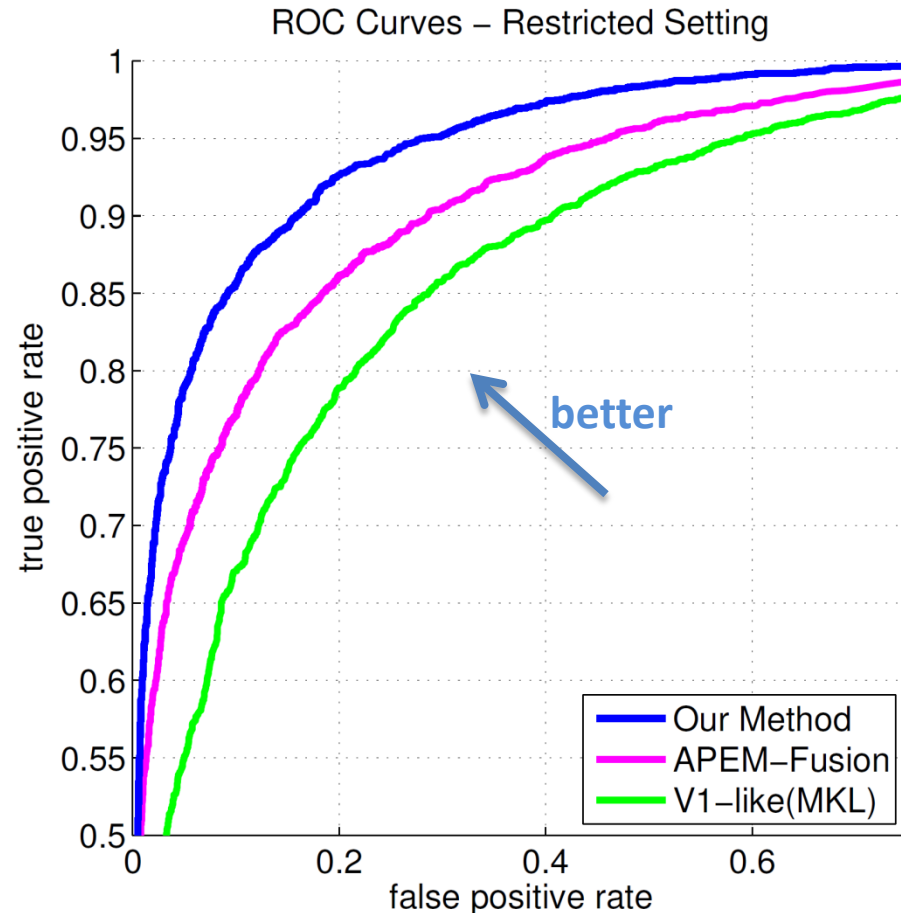
**important
(top-50
Gaussians)****irrelevant
(bottom-50
Gaussians)**

Results: LFW-restricted

Method	Mean Acc.
V1-like/MKL [26]	0.7935 ± 0.0055
PEM SIFT [19]	0.8138 ± 0.0098
APEM Fusion [19]	0.8408 ± 0.0120
Our Method	0.8747 ± 0.0149

verification accuracy

- no outside training data
- LFW-funneled images
 - 150×150 centre crop
- limited training data
 - just 5400 fixed image pairs
 - used diagonal metric (SVM)
- **state-of-the-art** accuracy: 87.47% vs 84.08%¹



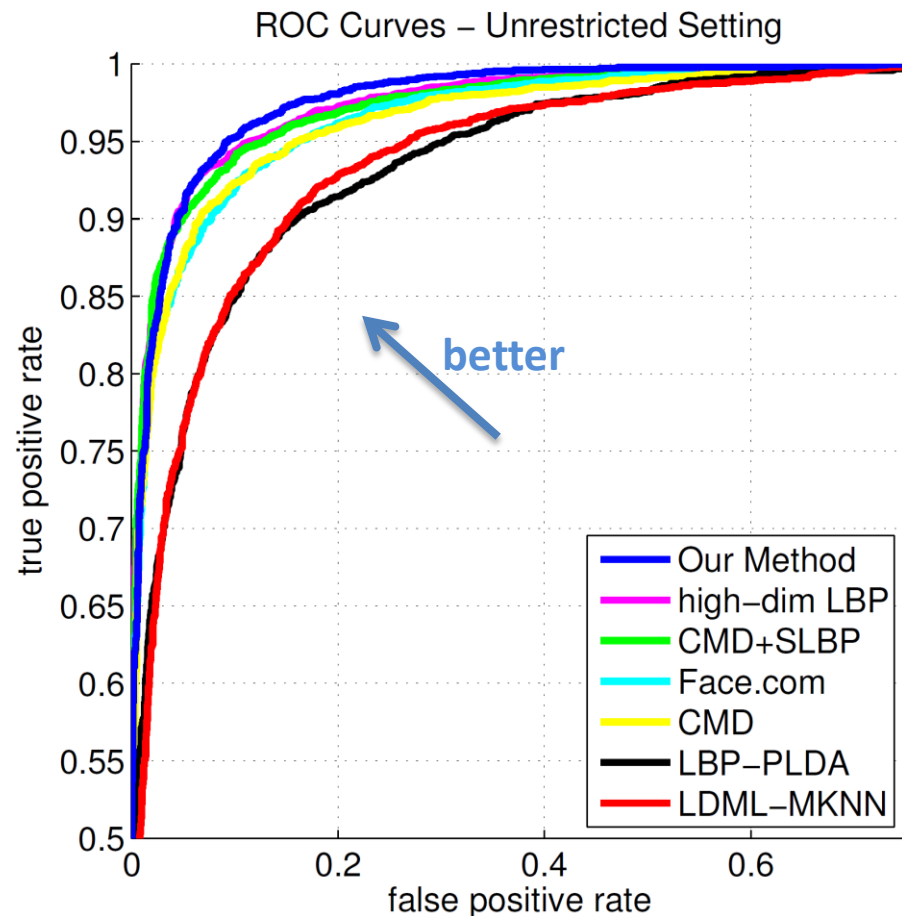
[1] “Probabilistic elastic matching for pose variant face verification”, H. Li, G. Hua, J. Brandt, and J. Yang. CVPR 2013.

Results: LFW-unrestricted

Method	Mean Acc.
LDML-MkNN [10]	0.8750 ± 0.0040
Combined multishot [32]	0.8950 ± 0.0051
Combined PLDA [20]	0.9007 ± 0.0051
face.com [31]	0.9130 ± 0.0030
CMD + SLBP [12]	0.9258 ± 0.0136
LBP multishot [32]	0.8517 ± 0.0061
LBP PLDA [20]	0.8733 ± 0.0055
SLBP [12]	0.9000 ± 0.0133
CMD [12]	0.9170 ± 0.0110
High-dim SIFT [6]	$0.9177 \pm \text{N/A}$
High-dim LBP [6]	0.9318 ± 0.0107
Our Method	0.9303 ± 0.0105

verification accuracy

- outside training data only for alignment [Everingham '09]
- any number of training image pairs
- matches **state-of-the-art** accuracy: 93.03% vs 93.18%¹



[1] “Blessing of dimensionality: high dimensional feature and its efficient compression for face verification”, D. Chen, X. Cao, F. Wen, and J. Sun. CVPR, 2013.

Summary

- **Fisher Vector Face (FVF)** representation
 - achieves state-of-the-art on LFW (restricted & unrestricted)
 - performs very well on top of different alignment schemes
- FVF is based on off-the-shelf techniques
 - dense SIFT (no need for sophisticated landmark detectors)
 - Fisher vector
 - discriminative dimensionality reduction