# Gaussian Processes for Face Recognition

**Workshop on Gaussian Processes for Feature Extraction**

**Gaussian Process Summer School**

---

**Chaochao Lu**

Multimedia Lab, Dept. of Information Engineering
The Chinese University of Hong Kong

September 18, 2014

# Outline

- Face Recognition
- Review of GPs and Its Related Models
- GaussianFace Model
- Learned Bayesian Face Model
- Thinking in GPs for Computer Vision

# Outline

- Face Recognition
- Review of GPs and Its Related Models
- GaussianFace Model
- Learned Bayesian Face Model
- Thinking in GPs for Computer Vision

# Face Recognition

## 1. What Is Face Recognition?

- Face is the most common biometric used by humans.
- Face recognition aims to identify or authenticate individuals by comparing their face against a database of known faces and looking for a match.
- Face recognition can be traced back to the 1960s. The technology first captured the public's attention from the media reaction to a trial implementation at the January 2001 Super Bowl.

# Face Recognition

## 1. What Is Face Recognition?

- Face Verification: validate a claimed identity based on the query face image (1:1 matching)


- Face Identification: identify a person by computers based on a query face image (1:N matching)






# Face Recognition

## 2. Challenges for Face Recognition

- Large intra-personal variations of poses, illuminations, aging, occlusions, makeups, hair styles and expressions.

Figure: Many faces of Lady Gaga.

# Face Recognition

## 2. Challenges for Face Recognition

- Small extra-personal variations: different people look similar.

Figure: Miss Korean 2013 Contestants All Look Identical.
(http://www.inquisitr.com/636319/miss-korea-2013-contestants-all-look-identical-say-redditors)

- Small sample size problems: each person only has a few training examples while features are usually in very high dimensional space. It is easy for the classifier to overfit the training set.

# Face Recognition

## 3. Applications

- Access Control


http://www.nec.com/en/global/solutions/safety/face_recognition/NeoFaceWatch.html

# Face Recognition

## 3. Applications

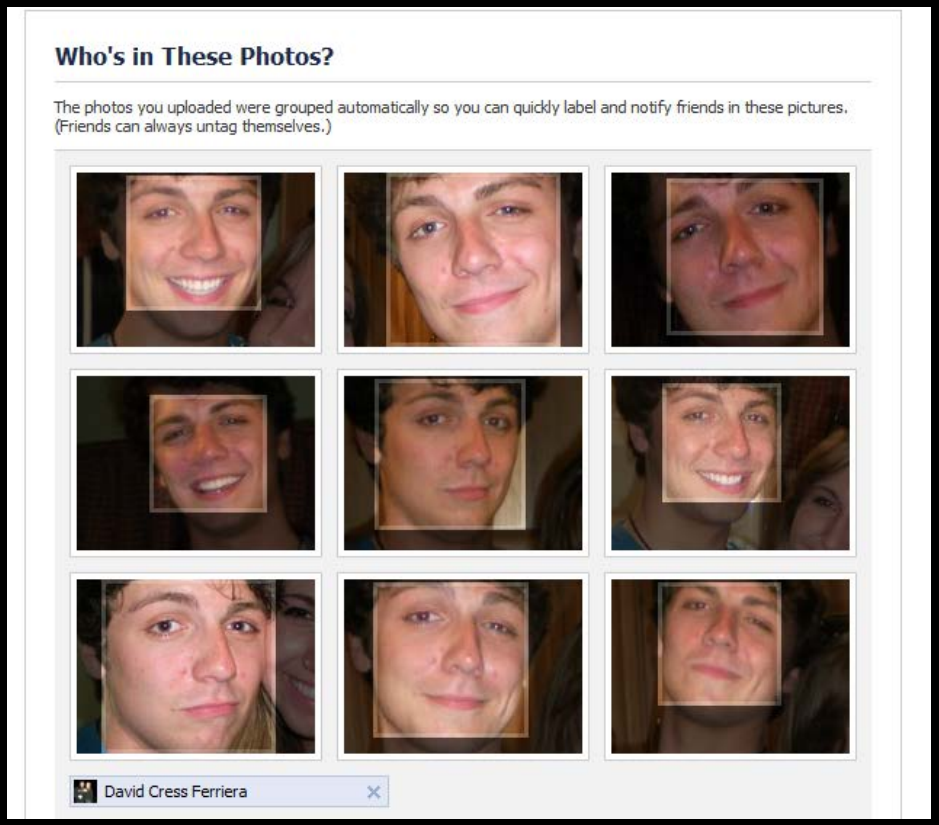
- Surveillance
  (support law enforcement, identify missing children, criminal investigations, etc.)


http://www.nec.com/en/global/solutions/safety/face_recognition/NeoFaceWatch.html

# Face Recognition

## 3. Applications

- Facebook's Automatic Face Tag Suggestion




http://blog.timesunion.com/highschool/facebook-recognizes-your-face-stop-it-now/19944

# Face Recognition

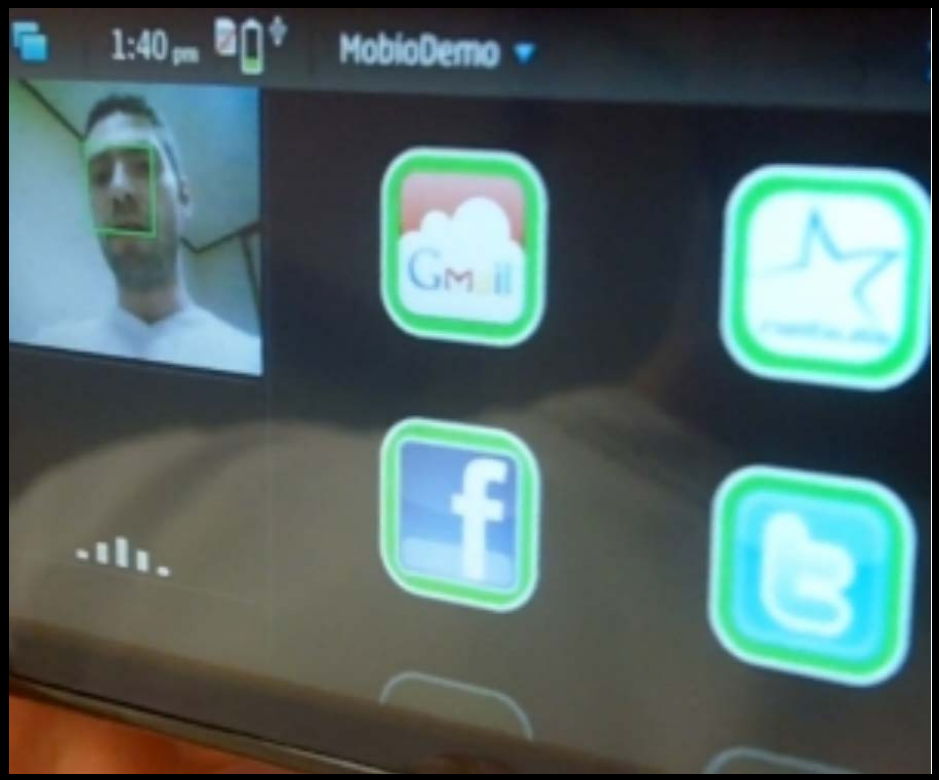
## 3. Applications

- EasyToon: An Easy and Quick Tool to Personalize a Cartoon Storyboard Using Family Photo Album (ACM MM'08)

# Face Recognition

## 3. Applications

- Security
  (Log into Twitter/Facebook with your face)




http://vimeo.com/16660014

# Face Recognition

## 3. Applications

- Augmented ID


https://www.youtube.com/watch?v=tb0pMeg1UN0

# Face Recognition

## 3. Applications

- Game (CrowsComing)


https://www.youtube.com/watch?v=XIq47P97fpI

# Face Recognition

## 4. Current State of Research

Nowadays, the Labeled Faces in the Wild (LFW) dataset is the most challenging benchmark for face verification. (Huang et al. 2007)


LFW Samples

# Face Recognition

## 4. Current State of Research

LFW Dataset:

- All of these images are collected from the Web.
- This dataset contains 13,233 uncontrolled face images of 5749 public figures with variety of pose, lighting, expression, race, ethnicity, age, gender, clothing, hairstyles, and other parameters.
- Before my GaussianFace, the accuracy rate has been improved from 60.02% (Turk et al. 1991) to 97.35% (Taigman et al. 2014) since LFW is established in 2007.
- Human-Level Performance: 97.53% (cropped images) and 99.20% (original images) (Kumar et al. 2009)

# Face Recognition

## 4. Current State of Research

- Top Five Performance on LFW

| Method | Accuracy | Author | Training data |
|---|---|---|---|
| **GaussianFace** | $\mathbf{0.9852 \pm 0.0066}$ | **CUHK** | 200K image pairs |
| DeepID | $0.9745 \pm 0.0026$ | CUHK | 200K images |
| DeepFace | $0.9735 \pm 0.0025$ | Facebook | 4000K images |
| LearnedBayesian | $0.9665 \pm 0.0031$ | CUHK | 60K images |
| FR+FCN | $0.9645 \pm 0.0025$ | CUHK | 90K images |

(Here: GaussianFace (Lu et al. 2014), DeepID (Sun et al. 2014), DeepFace (Taigman et al. 2014), LearnedBayesian (Lu et al. 2014), FR+FCN (Zhu et al. 2014)).

NB: Human-Level Performance: 97.53% (cropped images).

# Face Recognition

## 4. Current State of Research

- Top Five Performance on LFW

| Method | Accuracy | Author | Training data |
|---|---|---|---|
| **GaussianFace** | $\mathbf{0.9852 \pm 0.0066}$ | **CUHK** | 200K image pairs |
| DeepID | $0.9745 \pm 0.0026$ | CUHK | 200K images |
| DeepFace | $0.9735 \pm 0.0025$ | Facebook | 4000K images |
| LearnedBayesian | $0.9665 \pm 0.0031$ | CUHK | 60K images |
| FR+FCN | $0.9645 \pm 0.0025$ | CUHK | 90K images |

(Here: GaussianFace (Lu et al. 2014), DeepID (Sun et al. 2014), DeepFace (Taigman et al. 2014), LearnedBayesian (Lu et al. 2014), FR+FCN (Zhu et al. 2014)).

NB: Human-Level Performance: 97.53% (cropped images).

# Face Recognition

## 4. Current State of Research

- Top Five Performance on LFW

| Method | Accuracy | Author | Training data |
|---|---|---|---|
| **GaussianFace** | **$0.9852 \pm 0.0066$** | **CUHK** | 200K image pairs |
| DeepID | $0.9745 \pm 0.0026$ | CUHK | 200K images |
| DeepFace | $0.9735 \pm 0.0025$ | Facebook | 4000K images |
| LearnedBayesian | $0.9665 \pm 0.0031$ | CUHK | 60K images |
| FR+FCN | $0.9645 \pm 0.0025$ | CUHK | 90K images |

(Here: GaussianFace (Lu et al. 2014), DeepID (Sun et al. 2014), DeepFace (Taigman et al. 2014), LearnedBayesian (Lu et al. 2014), FR+FCN (Zhu et al. 2014)).

NB: Human-Level Performance: 97.53% (cropped images).

# Face Recognition

## 4. Current State of Research

- Top Five Performance on LFW

| Method | Accuracy | Author | Training data |
|---|---|---|---|
| **GaussianFace** | $\mathbf{0.9852 \pm 0.0066}$ | **CUHK** | 200K image pairs |
| DeepID | $0.9745 \pm 0.0026$ | CUHK | 200K images |
| DeepFace | $0.9735 \pm 0.0025$ | Facebook | 4000K images |
| LearnedBayesian | $0.9665 \pm 0.0031$ | CUHK | 60K images |
| FR+FCN | $0.9645 \pm 0.0025$ | CUHK | 90K images |

(Here: GaussianFace (Lu et al. 2014), DeepID (Sun et al. 2014), DeepFace (Taigman et al. 2014), LearnedBayesian (Lu et al. 2014), FR+FCN (Zhu et al. 2014)).

NB: Human-Level Performance: 97.53% (cropped images).

# Face Recognition

## 4. Current State of Research

- Top Five Performance on LFW

| Method | Accuracy | Author | Training data |
|---|---|---|---|
| **GaussianFace** | $\mathbf{0.9852 \pm 0.0066}$ | **CUHK** | 200K image pairs |
| DeepID | $0.9745 \pm 0.0026$ | CUHK | 200K images |
| DeepFace | $0.9735 \pm 0.0025$ | Facebook | 4000K images |
| LearnedBayesian | $0.9665 \pm 0.0031$ | CUHK | 60K images |
| FR+FCN | $0.9645 \pm 0.0025$ | CUHK | 90K images |

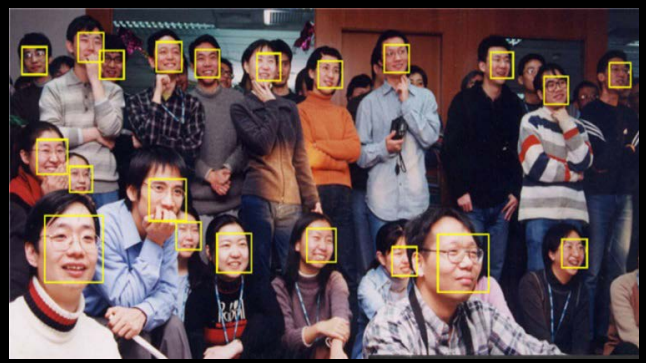
DeepID2: 99.15% using 200 deep ConvNets trained on 200K images with 21 facial landmarks alignment. (Sun et al. 2014)

NB: Human-Level Performance: 97.53% (cropped images).

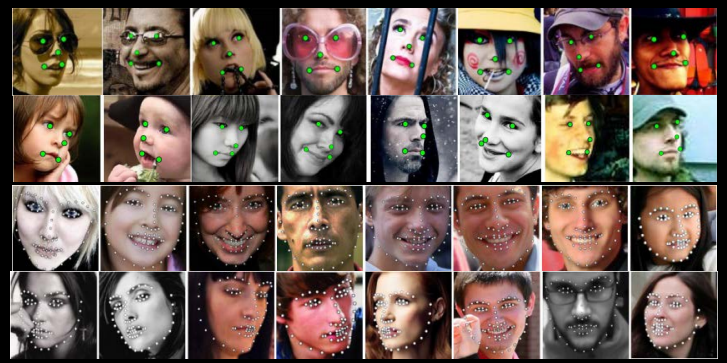
# Face Recognition

## 5. Face Recognition Pipeline


- Face detection: Given an image, the face region is localized by a face detection algorithm.
- Face alignment: The landmarks on the face automatically located by face alignment algorithm.



Xiao et al. ICCV'07



Zhang et al. ECCV'14

# Face Recognition

## 5. Face Recognition Pipeline


- Preprocessing: Given an image, the face region is localized by a face detection algorithm.
  (a) Geometric rectification: Face images are cropped and transformed such that the two eye centers and the mouth centers are at fixed positions. Background and hair are removed.
  (b) Photometric rectification: reduce the effect of lighting variations.

# Face Recognition

## 6. Different Approaches

- Features
  a) Global Features
     - Principal Component Analysis (PCA)
     - Independent Component Analysis (ICA)
     - Convolutional Neural Networks (CNN)
  b) Local Features
     - Local Binary Pattern (LBP)
     - Learning-based Descriptor (LE)
     - Gabor Wavelet

- Similarity Measure
  - Euclidian Distance
  - Neural Networks
  - Elastic Graph Matching
  - Template Matching

# Outline

- Face Recognition
- Review of GPs and Its Related Models
- GaussianFace Model
- Learned Bayesian Face Model
- Thinking in GPs for Computer Vision

# Review of GPs and Its Related Models

1. Gaussian Processes (GPs)
2. Gaussian Process Regression (GPR)
3. Gaussian Processes for Clustering
4. Gaussian Process Latent Variable Model (GPLVM)
5. Discriminative GPLVM (DGPLVM)

# Review of GPs and Its Related Models

## 1. Gaussian Processes (GPs)

- Infinite set of variables: $\{f(x) : x \in \mathcal{X}\}$
- All possible mappings from $\mathcal{X}$ to $\mathbb{R}$: $\mathcal{F}$
- For any $f(\cdot) \in \mathcal{F}$, we have

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) \end{bmatrix}\right),$$

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)), \tag{1}$$

$$m(x) = \mathbb{E}[x], \tag{2}$$

$$k(x, x') = \mathbb{E}[(x - m(x))(x' - m(x'))^\top]. \tag{3}$$

# Review of GPs and Its Related Models

## 2. Gaussian Process Regression (GPR)

- GPR Model

$$y^i = f(x^i) + \epsilon^i, \quad i = 1, \ldots, m. \tag{4}$$

where

$$\begin{aligned} f(\cdot) &\sim \mathcal{GP}(0, k(\cdot, \cdot)), \\ \epsilon^i &\sim \mathcal{N}(0, \sigma^2). \end{aligned} \tag{5}$$

- Bayesian training

$$P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = \int p(\mathbf{Y}|\mathbf{f}) p(\mathbf{f}|\mathbf{X}) \mathrm{d}\mathbf{f} \tag{6}$$

# Review of GPs and Its Related Models

## 2. Gaussian Process Regression (GPR)

- Given test points

$$T = \{(x_*^i, y_*^i)\}_{i=1}^{m_*}. \tag{7}$$

- Prediction

$$\mathbf{y}_* \mid \mathbf{y}, \mathbf{X}, \mathbf{X}_* \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \tag{8}$$

where

$$\boldsymbol{\mu}_* = K(\mathbf{X}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1}\mathbf{y},$$

$$\boldsymbol{\Sigma}_* = K(\mathbf{X}_*, \mathbf{X}_*) + \sigma^2 \mathbf{I} - K(\mathbf{X}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1} K(\mathbf{X}, \mathbf{X}_*).$$

# Review of GPs and Its Related Models

## 3. Gaussian Processes for Clustering (Kim et al. 2007)

- The principle of GP clustering is based on the key observation that the variances of predictive values are smaller in dense areas and larger in sparse areas.

$$\sigma^2(x_*) = \mathbf{K}_{**} - \mathbf{K}_* \tilde{\mathbf{K}}^{-1} \mathbf{K}_*^\top. \tag{9}$$

- To perform clustering, the following dynamic system associated with the above equation can be written as

$$F(x) = -\nabla \sigma^2(x). \tag{10}$$

# Review of GPs and Its Related Models

## 4. Gaussian Process Latent Variable Model (GPLVM) (Lawrence 2003)

- The posterior can be written as

$$p(\mathbf{Z}, \boldsymbol{\theta}|\mathbf{X}) = \frac{1}{\mathcal{Z}_a} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta}) p(\mathbf{Z}) p(\boldsymbol{\theta}), \tag{11}$$

where $\mathcal{Z}_a$ is a normalization constant, the uninformative priors over $\boldsymbol{\theta}$, and the simple spherical Gaussian priors over $\mathbf{Z}$ are introduced as follows,

$$p(\boldsymbol{\theta}) = \prod_i \theta_i, \tag{12}$$

$$p(\mathbf{Z}) = \prod_i \exp\left(\frac{\|\mathbf{z}_i\|^2}{2}\right). \tag{13}$$

# Review of GPs and Its Related Models

## 5. Discriminative GPLVM (DGPLVM) (Urtasun et al. 2007)

- Using an informative prior over the latent space $\mathbf{Z}$.
- Discriminative methods: LDA (linear) and GDA (non-linear).
- LDA and GDA try to maximize the between-class separability and minimize with-class variability by maximizing

$$J(\mathbf{Z}) = \mathrm{Tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right), \tag{14}$$

where $\mathbf{S}_w$ and $\mathbf{S}_b$ are the within- and between- class matrices:

$$\mathbf{S}_w = \sum_{i=1}^{L} \frac{N_i}{N} \left(\mathbf{M}_i - \mathbf{M}_0\right)\left(\mathbf{M}_i - \mathbf{M}_0\right)^\top, \tag{15}$$

$$\mathbf{S}_b = \sum_{i=1}^{L} \frac{N_i}{N} \left[\frac{1}{N_i}\sum_{k=1}^{N_i} \left(\mathbf{x}_k^i - \mathbf{M}_i\right)\left(\mathbf{x}_k^i - \mathbf{M}_i\right)^\top\right]. \tag{16}$$

# Review of GPs and Its Related Models

## 5. Discriminative GPLVM (DGPLVM) (Urtasun et al. 2007)

- The informative prior over the latent space $\mathbf{Z}$ is interpreted as

$$p(\mathbf{Z}) = \frac{1}{\mathcal{Z}_b} \exp\left(-\frac{1}{\sigma^2} J(\mathbf{Z})\right), \tag{17}$$

where $\mathcal{Z}_b$ is a normalization constant, and $\sigma^2$ represents a global scaling of the prior.

- Given $\mathbf{x}_*$, $\mathbf{z}_*$ can be obtained by optimizing

$$\mathcal{L}_{Inf} = \frac{\|\mathbf{x}_* - \mu(\mathbf{z}_*)\|^2}{2\sigma^2(\mathbf{z}_*)} + \frac{D}{2} \ln \sigma^2(\mathbf{z}_*) + \frac{1}{2}\|\mathbf{z}_*\|^2, \tag{18}$$

where

$$\mu(\mathbf{z}_*) = \mu + \mathbf{X}^\top \mathbf{K}^{-1} \mathbf{K}_*,$$
$$\sigma^2(\mathbf{z}_*) = \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}^{-1} \mathbf{K}_*.$$

# Outline

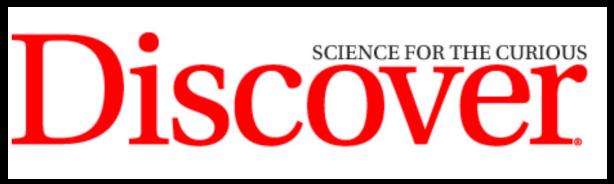
- Face Recognition
- Review of GPs and Its Related Models
- GaussianFace Model
  (Paper: Surpassing Human-Level Face Verification Performance on LFW with GaussianFace, arXiv:1404.3840)
- Learned Bayesian Face Model
- Thinking in GPs for Computer Vision

# GaussianFace Model

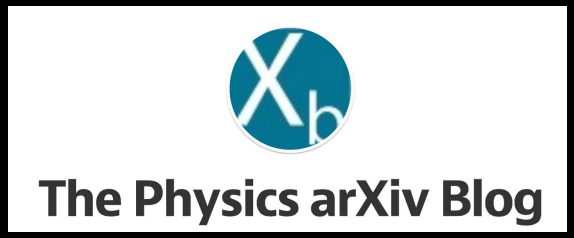
## News Coverage of GaussianFace














# GaussianFace Model

## Limitations of Current Face Verification Methods

- Most existing face verification methods assume that the training data and the test data are drawn from the same feature space and follow the same distribution. When the distribution changes, these methods may suffer a large performance drop.

- Most existing face verification methods require some assumptions to be made about the structures of the data, they cannot work well when the assumptions are not valid. Moreover, due to the existence of the assumptions, it is hard to capture the intrinsic structures of data using these methods.

# GaussianFace Model

## Key Ideas

we propose the Multi-Task Learning approach based on DGPLVM , named GaussianFace, for face verification.

- GaussianFace model is a non-parametric Bayesian kernel method, and can adapt its complexity flexibly to the complex data distributions in the real-world, without any heuristics or manual tuning of parameters.
- To enhance discriminative power, we introduced a more efficient equivalent form of Kernel Fisher Discriminant Analysis to DGPLVM.
- In order to take advantage of more data from multiple source-domains to improve the performance in the target-domain, we introduce the multi-task learning constraint to DGPLVM.
- To speed up the process of inference and prediction, we exploited the low rank approximation method.

# GaussianFace Model

## DGPLVM Reformulation

- A more efficient equivalent form of KFDA (Kim et al. 2006)

$$J^* = \frac{1}{\lambda}\left(\mathbf{a}^\top \mathbf{K}\mathbf{a} - \mathbf{a}^\top \mathbf{K}\mathbf{A}(\lambda \mathbf{I}_n + \mathbf{A}\mathbf{K}\mathbf{A})^{-1}\mathbf{A}\mathbf{K}\mathbf{a}\right), \tag{19}$$

where

$$\mathbf{a} = [\frac{1}{n_+}\mathbf{1}_{N_+}^\top, -\frac{1}{N_-}\mathbf{1}_{N_-}^\top],$$

$$\mathbf{A} = \text{diag}\Big(\frac{1}{\sqrt{N_+}}(\mathbf{I}_{N_+} - \frac{1}{N_+}\mathbf{1}_{N_+}\mathbf{1}_{N_+}^\top), \frac{1}{\sqrt{N_-}}(\mathbf{I}_{N_-} - \frac{1}{N_-}\mathbf{1}_{N_-}\mathbf{1}_{N_-}^\top)\Big).$$

Here, $\mathbf{I}_N$ denotes the $N \times N$ identity matrix and $\mathbf{1}_N$ denotes the length-$N$ vector of all ones in $\mathbb{R}^N$.

# GaussianFace Model

## Multi-task Learning Constraint

- We extend the mutual entropy to multiple distributions as follows,

$$\mathcal{M} = H(p_t) - \frac{1}{S}\sum_{i=1}^{S} H(p_t|p_i), \tag{20}$$

where $H(\cdot)$ is the marginal entropy, $H(\cdot|\cdot)$ is the conditional entropy, $S$ is the number of source tasks, $\{p_i\}_{i=1}^{S}$, and $p_t$ are the probability distributions of source tasks and target task, respectively.

# GaussianFace Model

## Our GaussianFace Model

- $S$ source-domain datasets $\{\mathbf{X}_1, \ldots, \mathbf{X}_S\}$, a target-domain data $\mathbf{X}_T$.
- The Automatic Relevance Determination (ARD) kernel is used,

$$k_{\boldsymbol{\theta}}(\mathbf{z}_i, \mathbf{z}_j) = \theta_0 \exp\Big(-\frac{1}{2}\sum_{m=1}^{d}\theta_m(\mathbf{z}_i^m - \mathbf{z}_j^m)^2\Big) + \theta_{d+1} + \frac{\delta_{\mathbf{z}_i,\mathbf{z}_j}}{\theta_{d+2}}, \quad (21)$$

- For each dataset, learning the DGPLVM is equivalent to optimizing

$$p(\mathbf{Z}_i, \boldsymbol{\theta}|\mathbf{X}_i) = \frac{1}{\mathcal{Z}_a} p(\mathbf{X}_i|\mathbf{Z}_i, \boldsymbol{\theta}) p(\mathbf{Z}_i) p(\boldsymbol{\theta}), \quad (22)$$

where

$$p(\mathbf{X}_i|\mathbf{Z}_i, \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^{ND}|\mathbf{K}|^D}} \exp\Big(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{X}_i\mathbf{X}_i^\top)\Big),$$

$$p(\mathbf{Z}_i) = \frac{1}{\mathcal{Z}_b} \exp\Big(-\frac{1}{\sigma^2}J^*\Big).$$

# GaussianFace Model

## Our GaussianFace Model (Cont'd)

- According to the multi-task learning constraint, we can attain

$$\mathcal{M} = H(p(\mathbf{Z}_T, \theta|\mathbf{X}_T)) - \frac{1}{S}\sum_{i=1}^{S} H(p(\mathbf{Z}_T, \theta|\mathbf{X}_T)|p(\mathbf{Z}_i, \theta|\mathbf{X}_i)). \quad (23)$$

- Learning GaussianFace model amounts to minimizing the follow marginal likelihood,

$$\mathcal{L}_{Model} = -\log p(\mathbf{Z}_T, \theta|\mathbf{X}_T) - \beta\mathcal{M}. \quad (24)$$

We can optimize the model with respect to the hyper-parameters $\theta$ and the latent positions $\mathbf{Z}_i$ by the Scaled Conjugate Gradient (SCG) technique.

# GaussianFace Model

## Speedup

We use the anchor graphs method to speed up this process.

$$\underbrace{\mathbf{K} \approx \mathbf{Q}\mathbf{Q}^\top}_{n \times n} \xrightarrow{\text{Woodbury identity}} \underbrace{\mathbf{K} \approx \mathbf{Q}^\top \mathbf{Q}}_{q \times q}, \mathbf{Q} \in \mathbb{R}^{n \times q}, q \ll n. \quad (25)$$

- Speedup on Inference

$$(\lambda \mathbf{I}_n + \mathbf{A}\mathbf{K}\mathbf{A})^{-1} \approx \lambda^{-1}\mathbf{I}_n - \lambda^{-1}\mathbf{A}\mathbf{Q}(\lambda \mathbf{I}_q + \mathbf{Q}^\top \mathbf{A}\mathbf{A}\mathbf{Q})^{-1}\mathbf{Q}^\top \mathbf{A},$$

$$\mathbf{K}^{-1} \approx (\mathbf{K} + \tau \mathbf{I})^{-1} \approx \tau^{-1}\mathbf{I}_n - \tau^{-1}\mathbf{Q}(\tau \mathbf{I}_q + \mathbf{Q}^\top \mathbf{Q})^{-1}\mathbf{Q}^\top.$$

- Speedup on Prediction

$$(\mathbf{K} + \mathbf{W}^{-1})^{-1} \approx \mathbf{W} - \mathbf{W}\mathbf{Q}(\mathbf{I}_q + \mathbf{Q}^\top \mathbf{W}\mathbf{Q})^{-1}\mathbf{Q}^\top \mathbf{W}.$$

# GaussianFace Model

## GaussianFace Model For Face Verification

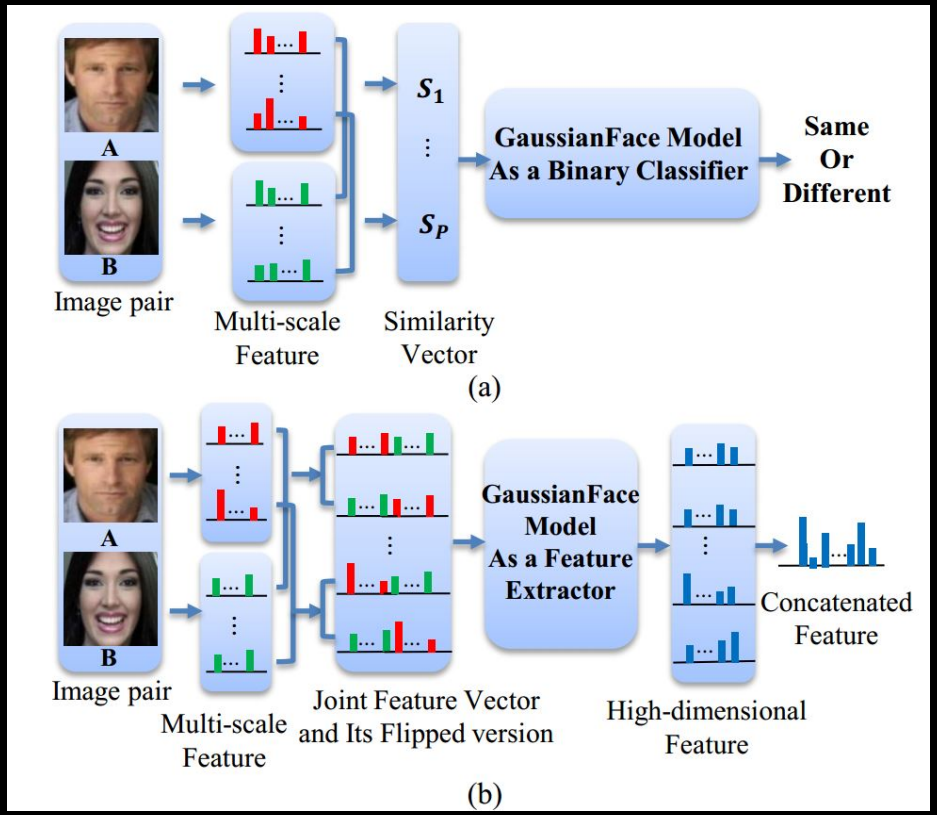



Figure: Two approaches based on GaussianFace model for face verification. (a) GaussianFace model as a binary classifier. (b) GaussianFace model as a feature extractor.

# GaussianFace Model

## GaussianFace Model as a Binary Classifier

S1: Given any un-seen face pair, compute its similarity vector $\mathbf{x}_*$,

S2: Estimate its latent representation $\mathbf{z}_*$,

S3: Predict whether the pair is from the same person as follows

$$\bar{\pi}(f_*) = \int \pi(f_*) p(f_*|\mathbf{X}, \mathbf{y}, x_*) \mathrm{d}f_*. \tag{26}$$

We prescribe the sigmoid function $\pi(\cdot)$ to be the cumulative Gaussian distribution $\Phi(\cdot)$, so

$$\bar{\pi}_* = \Phi\Big(\frac{\bar{f}_*(\mathbf{z}_*)}{\sqrt{1+\sigma^2(\mathbf{z}_*)}}\Big), \tag{27}$$

where $\sigma^2(\mathbf{z}_*) = \mathbf{K}_{**} - \mathbf{K}_* \tilde{\mathbf{K}}^{-1} \mathbf{K}_*^\top$ and $\bar{f}_*(\mathbf{z}_*) = \mathbf{K}_* \mathbf{K}^{-1} \hat{\mathbf{f}}$.

# GaussianFace Model

## GaussianFace Model as a Feature Extractor

S1: Estimate the latent representations of the training data,

S2: Group the latent data points into clusters automatically:

$$F(x) = -\nabla\sigma^2(x). \tag{28}$$

where

$$\sigma^2(x_*) = \mathbf{K}_{**} - \mathbf{K}_*\tilde{\mathbf{K}}^{-1}\mathbf{K}_*^\top.$$

S3: Suppose that we finally obtain $C$ clusters, we can compute

$$\{\mathbf{c}_i\}_{i=1}^C : \text{the centers of these clusters}$$

$$\{\Sigma_i^2\}_{i=1}^C : \text{the variances of these clusters}$$

$$\{w_i\}_{i=1}^C : \frac{\#\text{ of latent data points from the i-th cluster}}{\#\text{ all latent data points}}$$

# GaussianFace Model

## GaussianFace Model as a Feature Extractor (Cont'd)

S4: Obtain the corresponding probability $p_i$ and variance $\sigma_i^2$ of $\mathbf{c}_i$,

S5: For any un-seen pair of face images, compute its joint feature vector $\mathbf{x}_*$ for each patch pair,

S6: Estimate its latent representation $\mathbf{z}_*$,

S7: Compute its first-order and second-order statistics to the centers,

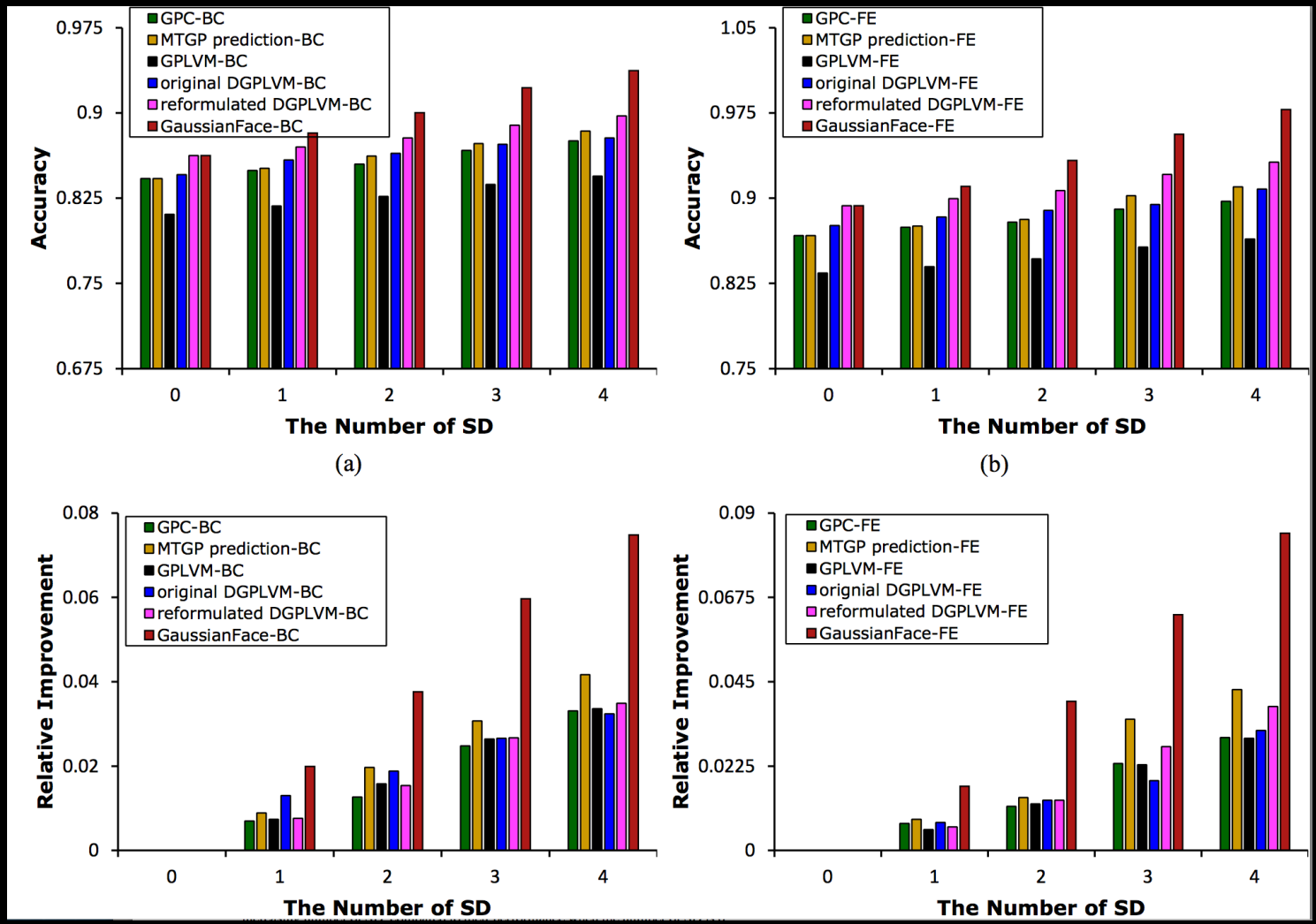
S8: Obtain the corresponding probability $p_i$ and variance $\sigma_i^2$ of $\mathbf{z}_*$,

S9: Each patch is represented by the high-dimensional facial feature $\hat{\mathbf{z}}_* = [\Delta_1^1, \Delta_1^2, \Delta_1^3, \Delta_1^4, \ldots, \Delta_C^1, \Delta_C^2, \Delta_C^3, \Delta_C^4]^\top$, where $\Delta_i^1 = w_i\left(\frac{\mathbf{z}_* - \mathbf{c}_i}{\Sigma_i}\right)$, $\Delta_i^2 = w_i\left(\frac{\mathbf{z}_* - \mathbf{c}_i}{\Sigma_i}\right)^2$, $\Delta_i^3 = \log\frac{p_*(1-p_i)}{p_i(1-p_*)}$, $\Delta_i^4 = \frac{\sigma_*^2}{\sigma_i^2}$,

S10: Concatenate all of the new high-dim features from each patch pair to form the final new high-dim feature for the face pair.

# GaussianFace Model

## Comparisons with Other MTGP/GP Methods



# GaussianFace Model

## Comparisons with Other BC/FE

| The Number of SD | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| SVM | 83.21 | 84.32 | 85.06 | 86.43 | 87.31 |
| LR | 81.14 | 81.92 | 82.65 | 83.84 | 84.75 |
| Adaboost | 82.91 | 83.62 | 84.80 | 86.30 | 87.21 |
| **GaussianFace-BC** | **86.25** | **88.24** | **90.01** | **92.22** | **93.73** |

| The Number of SD | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| K-means | 84.71 | 85.20 | 85.74 | 86.81 | 87.68 |
| RP Tree | 85.11 | 85.70 | 86.45 | 87.52 | 88.34 |
| GMM | 86.63 | 87.02 | 87.58 | 88.60 | 89.21 |
| **GaussianFace-FE** | **89.33** | **91.04** | **93.31** | **95.62** | **97.79** |

# GaussianFace Model

## Comparison with the state-of-art Methods

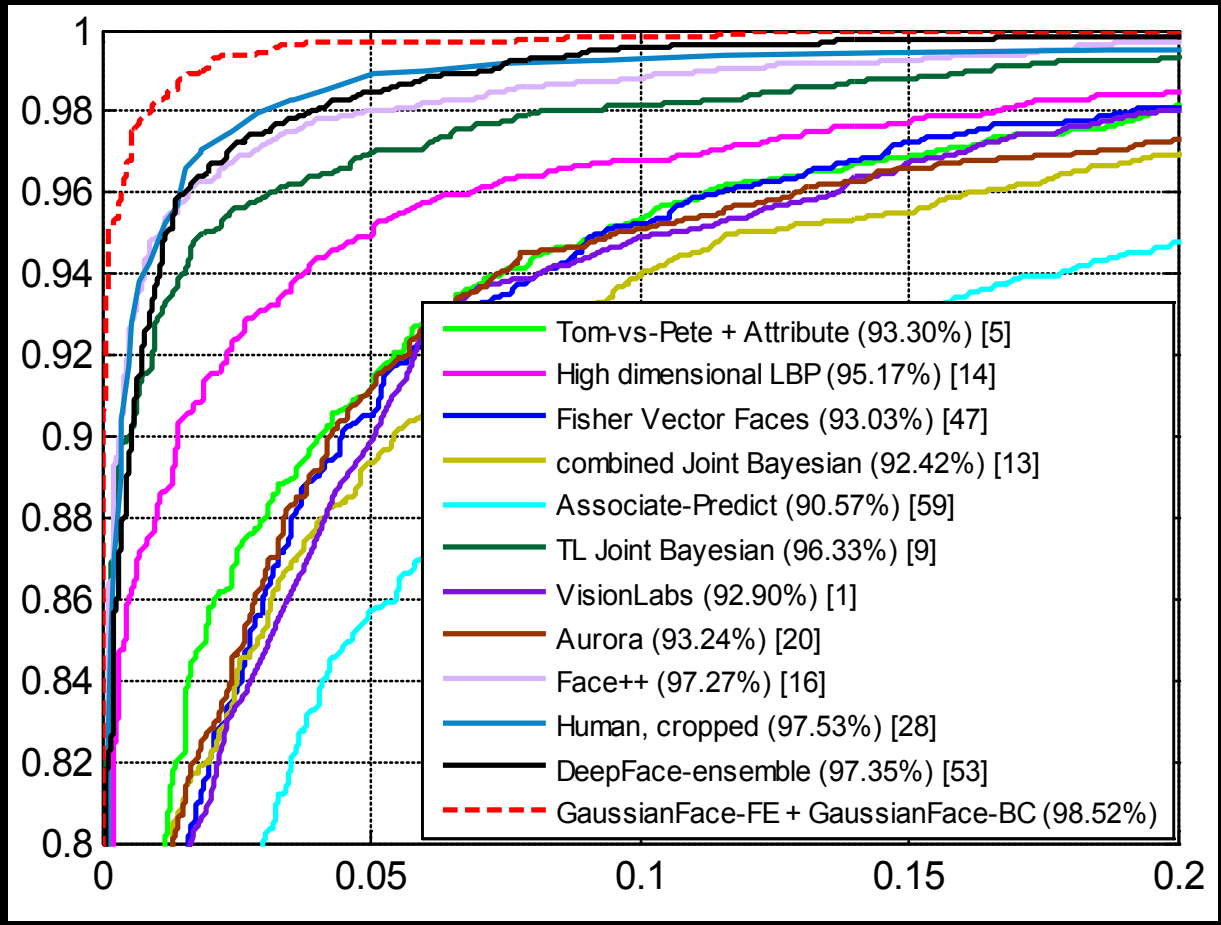



Figure: The ROC curve on LFW.

# Outline

- Face Recognition
- Review of GPs and Its Related Models
- GaussianFace Model
- Learned Bayesian Face Model
  (Paper: Learning the Face Prior for Bayesian Face Recognition, ECCV 2014)
- Thinking in GPs for Computer Vision

# Learned Bayesian Face Model

## Classic Bayesian Face Recognition (Moghaddam et al. 2000)

- The difference $\Delta = x_1 - x_2$ of two faces $x_1$ and $x_2$,
- It classifies $\Delta$ as intra-personal variations $\Omega_I$ or extra-personal variations $\Omega_E$,
- Based on the MAP (Maximum a Posterior) rule, the similarity measure between $x_1$ and $x_2$ can be expressed by

$$s(x_1, x_2) = \log \frac{p(\Delta|\Omega_I)}{p(\Delta|\Omega_E)}, \tag{29}$$

where both $p(\Delta|\Omega_I)$ and $p(\Delta|\Omega_E)$ are assumed to follow one multivariate Gaussian distribution.

# Learned Bayesian Face Model

## Problems in Classic Bayesian Face Recognition

**Problem 1**

It is based on the difference of a given face pair, which discards the discriminative information and reduce the separability.

**Problem 2**

The distributions of $p(\Delta|\Omega_I)$ and $p(\Delta|\Omega_E)$ are oversimplified, assuming one multivariate Gaussian distribution can cover large variations in facial poses, illuminations, expressions, aging, occlusions, makeups and hair styles in the real world.

**Problem 1** has been addressed (Chen et al. 2012), where the joint distribution of $\{x_1, x_2\}$ is directly modeled as a Gaussian.

In this paper, we focus on solving **Problem 2**.

# Learned Bayesian Face Model

## Key Idea

To overcome **Problem 2**, we propose a method to automatically learn the conditional distributions of $\{x_1, x_2\}$.

- We exploit the properties of Manifold Relevance Determination (MRD) (Damianou et al. 2012) and extend it to learn the identity subspace for $\{x_1, x_2\}$ automatically and accurately.
- Based on the structure of the learned identity subspace, we propose to flexibly estimate Gaussian mixture densities for $\{x_1, x_2\}$ with Gaussian process regression.

Since the subspace only contains the identity information, the learned density can fully reflect the distribution of identities of face pairs $\{x_1, x_2\}$ in the observation space.

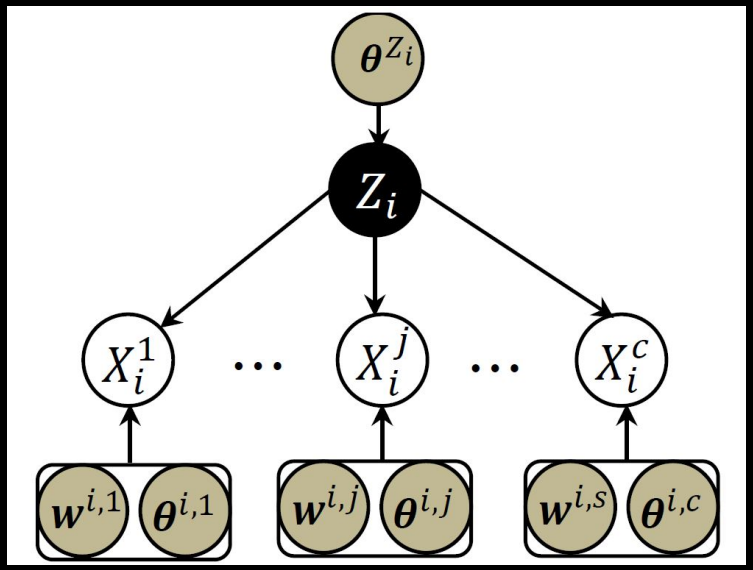
# Learned Bayesian Face Model

## Properties of MRD

- It can learn a factorized latent variable representation of multiple observation spaces;
- Each latent variable is either associated with a private space or a shared space;
- It is a fully Bayesian model and allows estimation of both the dimensionality and the structure of the latent representation to be done automatically.

# Learned Bayesian Face Model

## The Model of MRD (Damianou et al. 2012)




$$p(X_i^1, \cdots, X_i^c | Z_i, \theta^{X_i}) = \prod_{j=1}^{c} \int p(X_i^j | F^{i,j}) p(F^{i,j} | Z_i, \mathbf{w}^{i,j}, \theta^{i,j}) \mathrm{d}F^{i,j}$$

$$p(X_i^1, \cdots, X_i^c | \theta^{X_i}, \theta^{Z_i}) = \int p(X_i^1, \cdots, X_i^c | Z_i, \theta^{X_i}) p(Z_i | \theta^{Z_i}) \mathrm{d}Z_i.$$

# Learned Bayesian Face Model

## The Model of MRD (Cont'd) (Damianou et al. 2012)

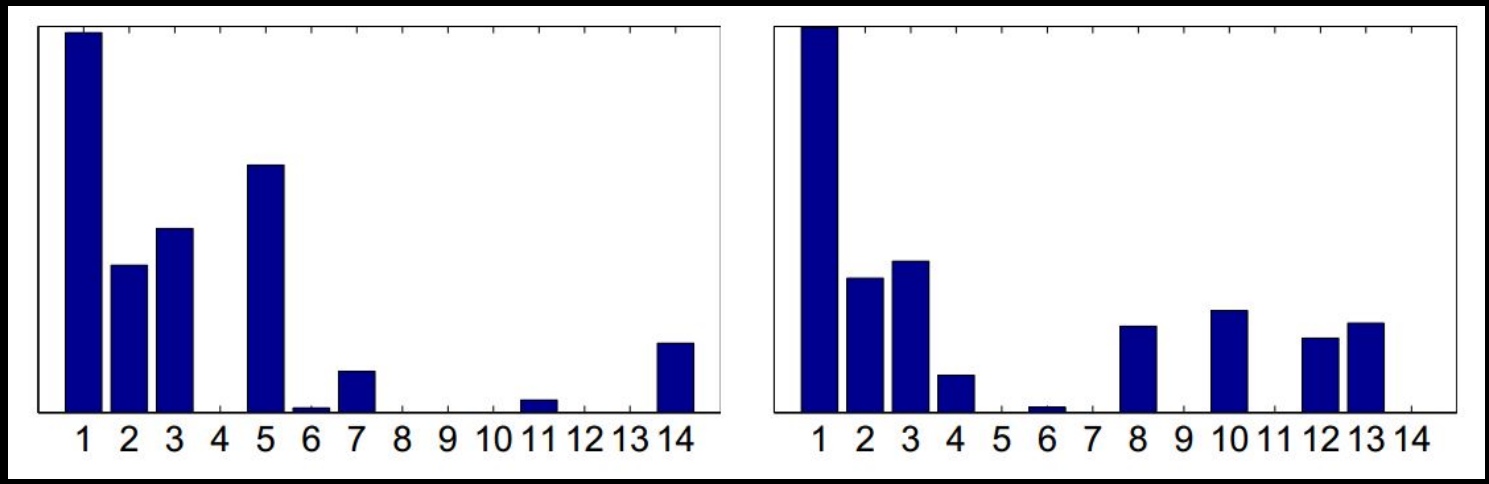



Figure: The ARD weights in the case with two views. (Damianou et al. 2012)

To make each individual lie in the identity subspace with the same dimension $Q_S$, we let $Q_S = \min(Q_S^1, \cdots, Q_S^M)$. For $Q_S^i > Q_S$, we only select the dimensions with $Q_S$ largest ARD weights.

# Learned Bayesian Face Model

## The Construction of Training Set for Bayesian Face

- For each individual, we can construct the following $n_i \times c$ correspondences between the identity subspace and the observation space,

$$\begin{bmatrix} \{z_1^i, x_1^{i,1}\} & \cdots & \{z_1^i, x_1^{i,j}\} & \cdots & \{z_1^i, x_1^{i,c}\} \\ \vdots & & \vdots & & \vdots \\ \{z_n^i, x_n^{i,1}\} & \cdots & \{z_n^i, x_n^{i,j}\} & \cdots & \{z_n^i, x_n^{i,c}\} \\ \vdots & & \vdots & & \vdots \\ \{z_{n_i}^i, x_{n_i}^{i,1}\} & \cdots & \{z_{n_i}^i, x_{n_i}^{i,j}\} & \cdots & \{z_{n_i}^i, x_{n_i}^{i,c}\} \end{bmatrix}.$$

- $K$ matched pairs and $K$ mismatched pairs, denoted by $\Pi_1$ and $\Pi_2$, can be generated using the following criterion,

$$\pi^k = \{[z_a^{i_a}, z_b^{i_b}], [x_a^{i_a,j_a}, x_b^{i_b,j_b}]\}, \quad k = 1, \ldots, K \tag{30}$$

where $\pi^k \in \Pi_1$ when $i_a = i_b$ and $\pi^k \in \Pi_2$ when $i_a \neq i_b$.

# Learned Bayesian Face Model

## Gaussian Mixture Modeling with GPR

- If $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z})$, then the distribution of $\mathbf{x}$ can be approximated by the following Gaussian distribution,

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x}), \tag{31}$$

  with $\boldsymbol{\mu}_\mathbf{x} = \mathbf{C}\bar{\mathbf{k}}$, and $\boldsymbol{\Sigma}_\mathbf{x} = (\bar{k} - \mathrm{Tr}(\mathbf{K}^{-1}\bar{\mathbf{K}}))\mathbf{I} + \mathbf{C}(\bar{\mathbf{K}} - \bar{\mathbf{k}}\bar{\mathbf{k}}^\top)\mathbf{C}^\top$,
  where $\mathbf{C} = [\mathbf{x}^1, \ldots, \mathbf{x}^K]\mathbf{K}^{-1}$, $\bar{\mathbf{k}} = \mathbb{E}[\mathbf{k}]$, $\bar{\mathbf{K}} = \mathbb{E}[\mathbf{k}\mathbf{k}^\top]$,
  $\mathbf{k} = [\hat{k}(\mathbf{z}^1, \mathbf{z}), \ldots, \hat{k}(\mathbf{z}^K, \mathbf{z})]^\top$, $\mathbf{K} = [\hat{k}(\mathbf{z}^a, \mathbf{z}^b)]_{a,b=1..K}$ and
  $\bar{k} = \hat{k}(\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\mu}_\mathbf{z})$.

- $p(\mathbf{z}) = \sum_{l=1}^{L} \lambda_l \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\mathbf{z}^l, \boldsymbol{\Sigma}_\mathbf{z}^l) \longmapsto p(\mathbf{x}) = \sum_{l=1}^{L} \lambda_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_\mathbf{x}^l, \boldsymbol{\Sigma}_\mathbf{x}^l)$

# Learned Bayesian Face Model

## The Leave-set-out Method

- To estimate the parameters of the covariance function on the training set $\{\mathbf{z}^k, \mathbf{x}^k\}_{k=1}^K$, we can maximize,

$$\mathcal{L}(\boldsymbol{\theta}^{\mathcal{K}}) = \sum_{k=1}^{K} \ln p(\mathbf{x}^k) = \sum_{k=1}^{K} \ln \sum_{l=1}^{L} \lambda_l \mathcal{N}(\mathbf{x}^k | \boldsymbol{\mu}_{\mathbf{x}}^l, \boldsymbol{\Sigma}_{\mathbf{x}}^l). \tag{32}$$

- We propose the leave-set-out (LSO) method to prevent overfitting,

$$\mathcal{L}_{LSO}(\boldsymbol{\theta}^{\mathcal{K}}) = \sum_{l=1}^{L} \sum_{k \in I_l} \ln \sum_{l' \neq l} \lambda_{l'} \mathcal{N}(\mathbf{x}^k | \boldsymbol{\mu}_{\mathbf{x}}^{l'}, \boldsymbol{\Sigma}_{\mathbf{x}}^{l'}). \tag{33}$$

- We use the scaled conjugate gradients to optimize $\mathcal{L}_{LSO}(\boldsymbol{\theta}^{\mathcal{K}})$ with respect to $\boldsymbol{\theta}^{\mathcal{K}}$.

# Learned Bayesian Face Model

## Handling Large Poses

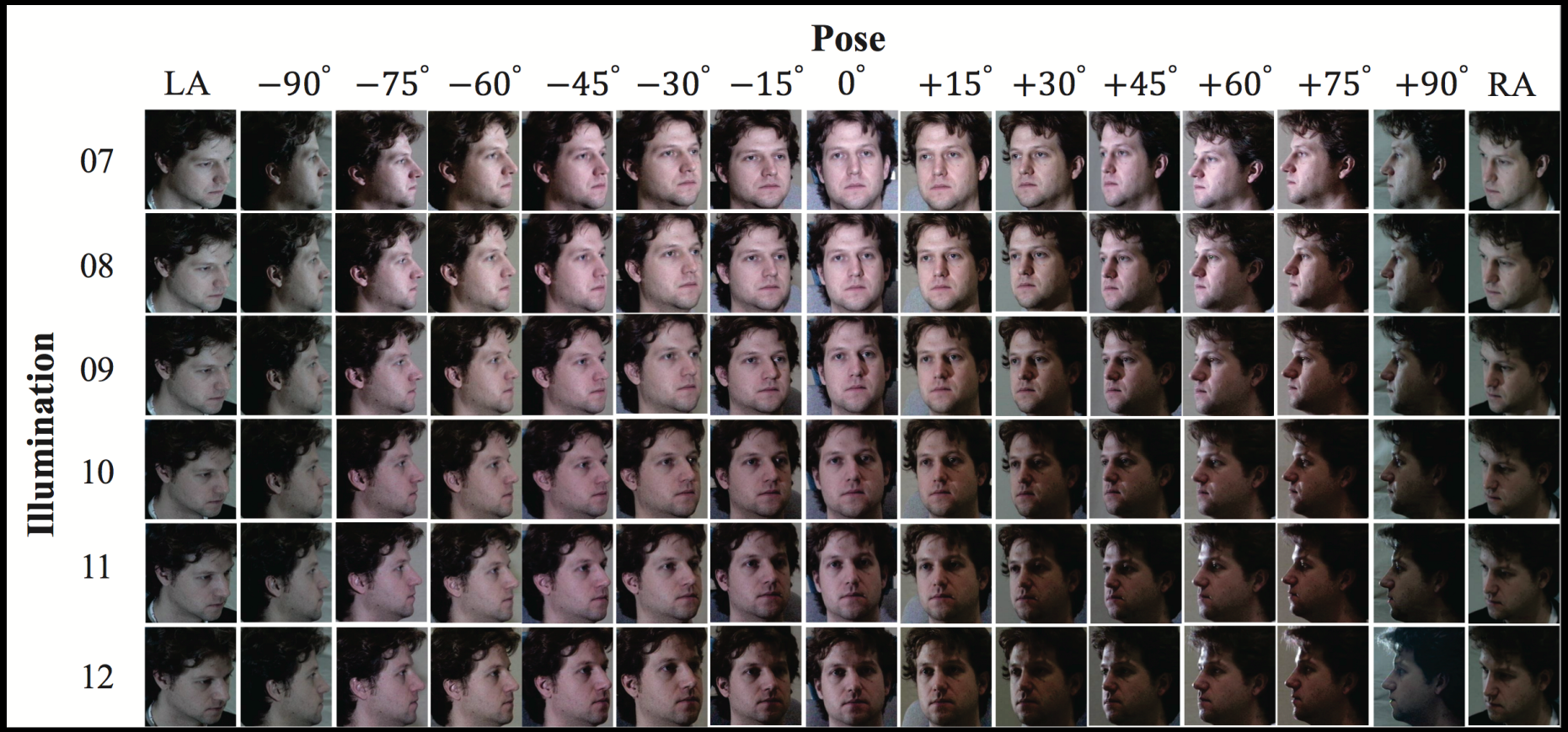



Figure: Samples on Multi-PIE.

# Learned Bayesian Face Model

## Handling Large Poses

| Pose Pairs | APEM | ELF | CBVT | TFA | LLR | MvDA | Ours |
|---|---|---|---|---|---|---|---|
| $\{0°, +60°\}$ | 65.3 | 77.4 | 86.7 | 89.1 | 85.4 | 86.4 | **93.6** |
| $\{0°, +75°\}$ | 51.7 | 63.9 | 79.2 | 86.5 | 74.7 | 82.3 | **91.2** |
| $\{0°, +90°\}$ | 40.1 | 38.9 | 70.1 | 82.4 | 64.2 | 73.6 | **88.5** |
| $\{+15°, +75°\}$ | 60.2 | 75.1 | 81.6 | 86.5 | 82.3 | 75.4 | **89.1** |
| $\{+15°, +90°\}$ | 45.8 | 55.2 | 75.2 | 81.2 | 78.6 | 79.3 | **89.2** |
| $\{+30°, +90°\}$ | 41.2 | 57.3 | 73.2 | 84.4 | 79.1 | 77.2 | **90.3** |

Table: Results (%) on the Multi-PIE dataset.

# Learned Bayesian Face Model

## Handling Large Occlusions


Figure: Samples on AR.

| Method | SRC | SMRFs | GSRC | Ours |
|---|---|---|---|---|
| Accuracy (%) | 87.13 | 92.42 | 94.38 | **96.23** |

Table: Results on the AR dataset.

# Learned Bayesian Face Model

## Comparison with the state-of-art Methods

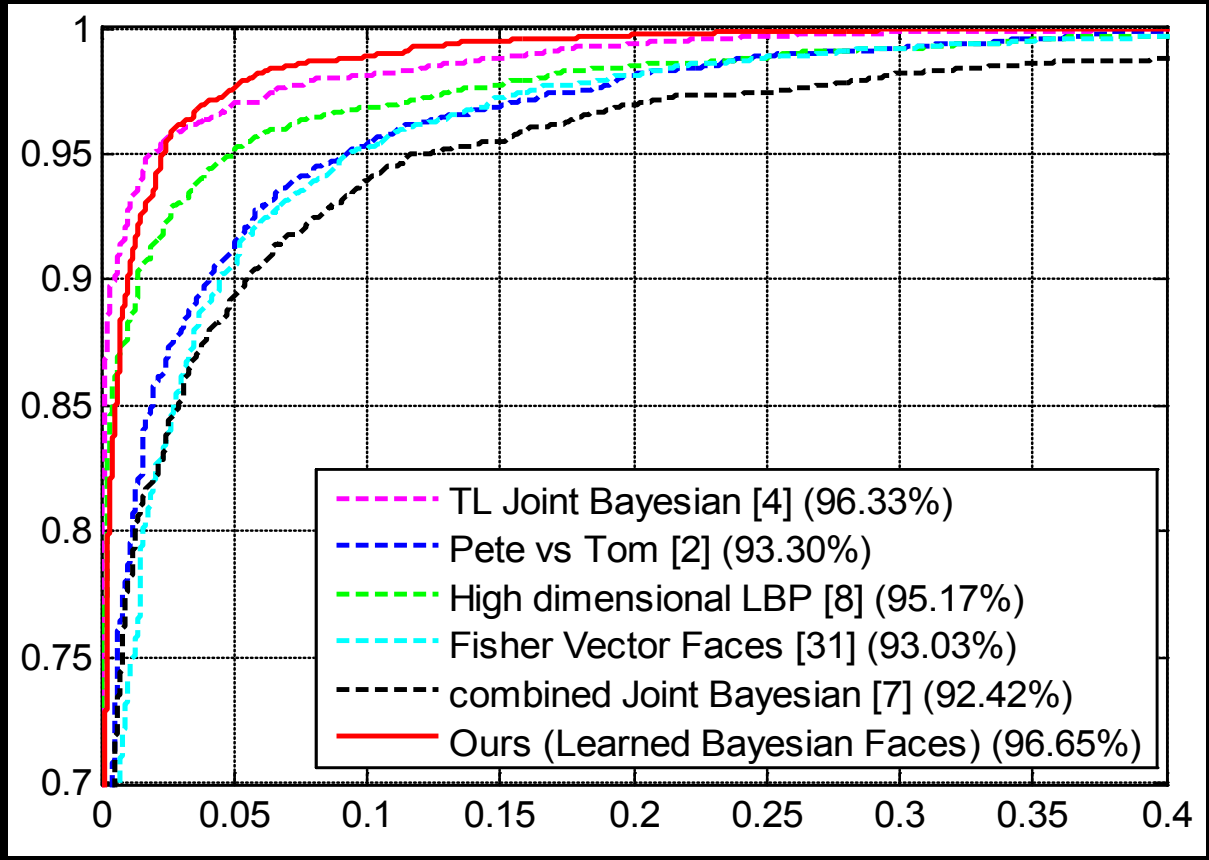



Figure: The ROC curve on LFW.

# Outline

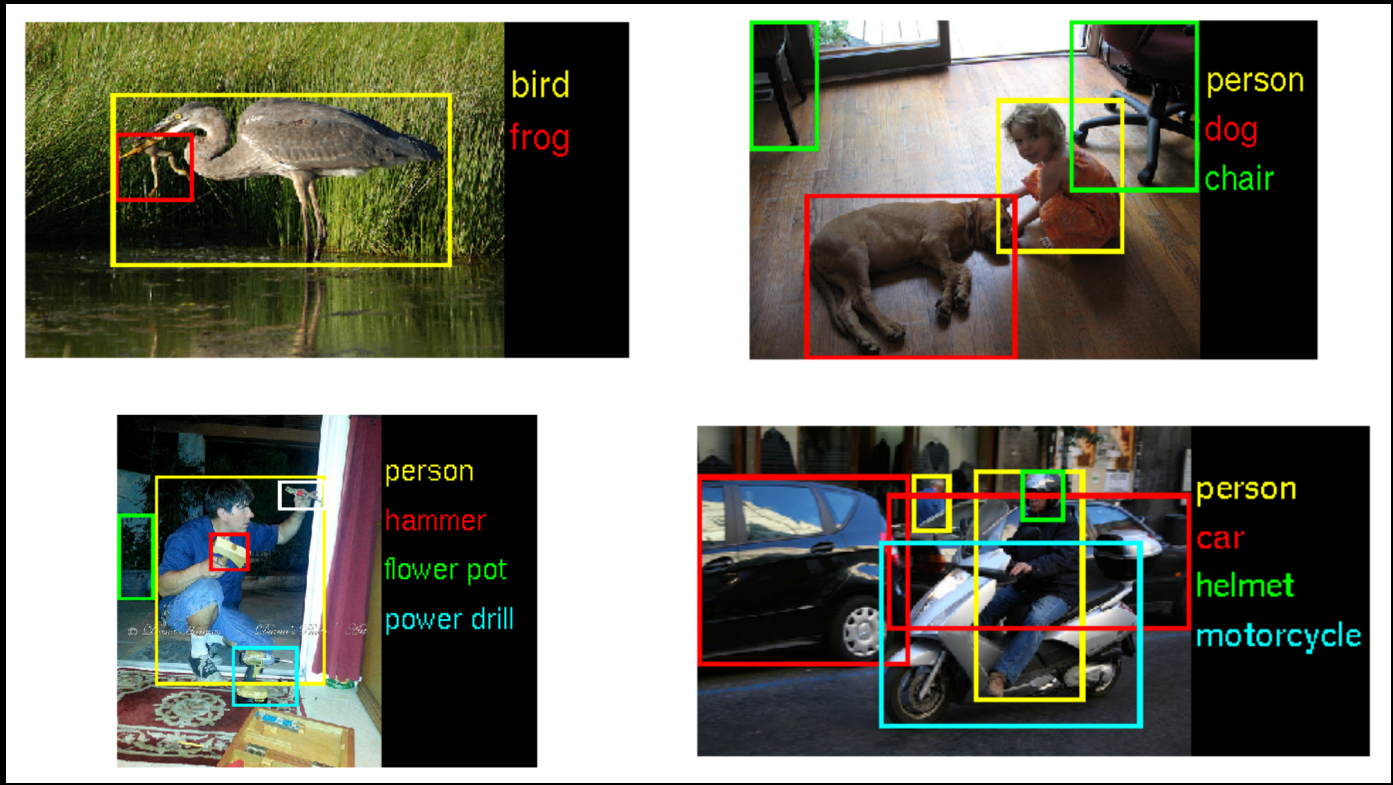
- Face Recognition
- Review of GPs and Its Related Models
- GaussianFace Model
- Learned Bayesian Face Model
- Thinking in GPs for Computer Vision

# Thinking in GPs for Computer Vision

## 1. Large Scale Visual Recognition Challenge 2014

**(Russakovsky⋆ and Deng⋆ et al. 2014)**

a) A detection challenge on fully labeled data for 200 categories of objects.



# Thinking in GPs for Computer Vision

## 1. Large Scale Visual Recognition Challenge 2014

**(Russakovsky⋆ and Deng⋆ et al. 2014)**

a) A detection challenge on fully labeled data for 200 categories of objects.

| | | PASCAL VOC 2012 | ILSVRC 2014 |
|---|---|---|---|
| Number of object classes | | 20 | **200** |
| Training | Num images | 5717 | 456567 |
| | Num objects | 13609 | 478807 |
| Validation | Num images | 5823 | 20121 |
| | Num objects | 13841 | 55502 |
| Testing | Num images | 10991 | 40152 |
| | Num objects | --- | --- |

# Thinking in GPs for Computer Vision

## 1. Large Scale Visual Recognition Challenge 2014

**(Russakovsky$^\star$ and Deng$^\star$ et al. 2014)**

a) A detection challenge on fully labeled data for 200 categories of objects.

**Task 1b: Object detection with additional training data**

**Ordered by number of categories won**

| Team name | Entry description | Description of outside data used | **Number of object categories won** | mean AP |
|---|---|---|---|---|
| GoogLeNet | Ensemble of detection models. Validation is 44.5% mAP | Pretraining on ILSVRC12 classification data. | 142 | 0.439329 |
| CUHK DeepID-Net | Combine multiple models described in the abstract without contextual modeling | ImageNet classification and localization data | 29 | 0.406659 |
| Deep Insight | Combination of three detection models | Three CNNs from classification task are used for initialization. | 27 | 0.404517 |
| UvA-Euvision | Deep learning with outside data | ImageNet 1000 | 1 | 0.354213 |
| Berkeley Vision | R-CNN baseline | The CNN was pre-trained on the ILSVRC 2013 CLS dataset. | 1 | 0.345213 |

NB: the newest result in our lab is 45%.

# Thinking in GPs for Computer Vision

## 1. Large Scale Visual Recognition Challenge 2014

**(Russakovsky$^\star$ and Deng$^\star$ et al. 2014)**

b) An image classification plus object localization challenge with 1000 categories.

- Training data: 1.2 million images from 1000 categories
- Validation data: 150,000 images with the presence or absence of 1000 categories
- Test data: 150,000 images with the presence or absence of 1000 categories

# Thinking in GPs for Computer Vision

## 1. Large Scale Visual Recognition Challenge 2014

**(Russakovsky* and Deng* et al. 2014)**

b) An image classification plus object localization challenge with 1000 categories.

**Ordered by classification error**

| Team name | Entry description | **Classification error** | Localization error |
|---|---|---|---|
| GoogLeNet | No localization. Top5 val score is 6.66% error. | 0.06656 | 0.606257 |
| VGG | a combination of multiple ConvNets, including a net trained on images of different size (fusion weights learnt on the validation set); detected boxes were not updated | 0.07325 | 0.256167 |
| VGG | a combination of multiple ConvNets, including a net trained on images of different size (fusion done by averaging); detected boxes were not updated | 0.07337 | 0.255431 |
| VGG | a combination of multiple ConvNets (by averaging) | 0.07405 | 0.253231 |
| VGG | a combination of multiple ConvNets (fusion weights learnt on the validation set) | 0.07407 | 0.253501 |
| MSRA Visual Computing | Multiple SPP-nets further tuned on validation set (B) | 0.0806 | 0.354924 |

# References


- M. A. Turk and A. P. Pentland. Face Recognition Using Eigenfaces. CVPR, 1991.
- Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to Human-Level Performance in Face Verification. CVPR, 2014.
- N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. ICCV, 2009.
- C. Lu, and X. Tang. Surpassing Human-Level Face Verification Performance on LFW with GaussianFace. arXiv:1404.3840, 2014.
- C. Lu, and X. Tang. Learning the Face Prior for Bayesian Face Recognition. ECCV, 2014.
- Y. Sun, X. Wang, and X. Tang. Deep Learning Face Representation from Predicting 10,000 Classes. CVPR, 2014.
- Y. Sun, X. Wang, and X. Tang. Deep Learning Face Representation by Joint Identification-Verification. arXiv:1406.4773, 2014.
- Z. Zhu, P. Luo, X. Wang, and X. Tang. Recover Canonical-View Faces in the Wild with Deep Neural Networks. arXiv:1404.3543, 2014
- G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, 2007.

# References


- C. E. Rasmussen, and C. Williams. Gaussian Processes for Machine Learning. the MIT Press, 2006.
- N. Lawrence. Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data. NIPS, 2004.
- A. Damianou, C. Ek, M. Titsias, and N. Lawrence. Manifold Relevance Determination. ICML, 2012.
- D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian Face Revisited: A Joint Formulation. ECCV, 2012.
- O. Russakovsky$^\star$, J. Deng$^\star$, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg and Li Fei-Fei. ($\star$= equal contribution) ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575, 2014.
- H. Kim, and J. Lee. Clustering based on Gaussian Processes. Neural Computation, 2007.
- R. Urtasun, and T. Darrell. Discriminative Gaussian process latent variable model for classification. ICML, 2007.
- S. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel fisher discriminant analysis. ICML, 2006.
- B. Moghaddam, T. Jebara, A. Pentland. Bayesian face recognition. Pattern Recognition, 2000.

Thank you

Q & A