Cascaded Nonlinear Shape Model Regression

Pedro Martins[1], Bruno Silva[1], Jorge Batista[2,1]

https://www.isr.uc.pt/~pedromartins
pedromartins@isr.uc.pt

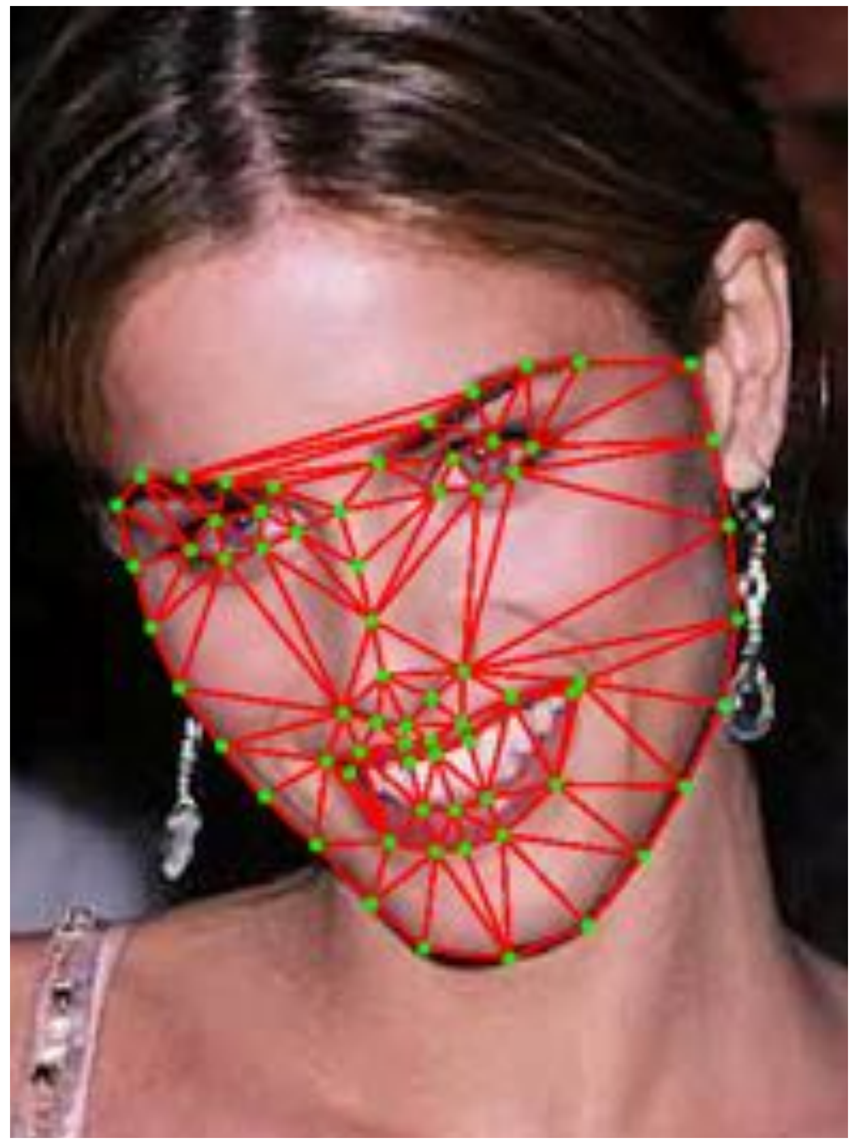
[1]Institute of Systems and Robotics (ISR)
University of Coimbra, Portugal

[2]Department of Electrical and Computer Engineering (DEEC)
University of Coimbra, Portugal

IEEE International Conference on Image Processing (ICIP) 2020, Abu Dhabi, United Arab Emirates

Introduction

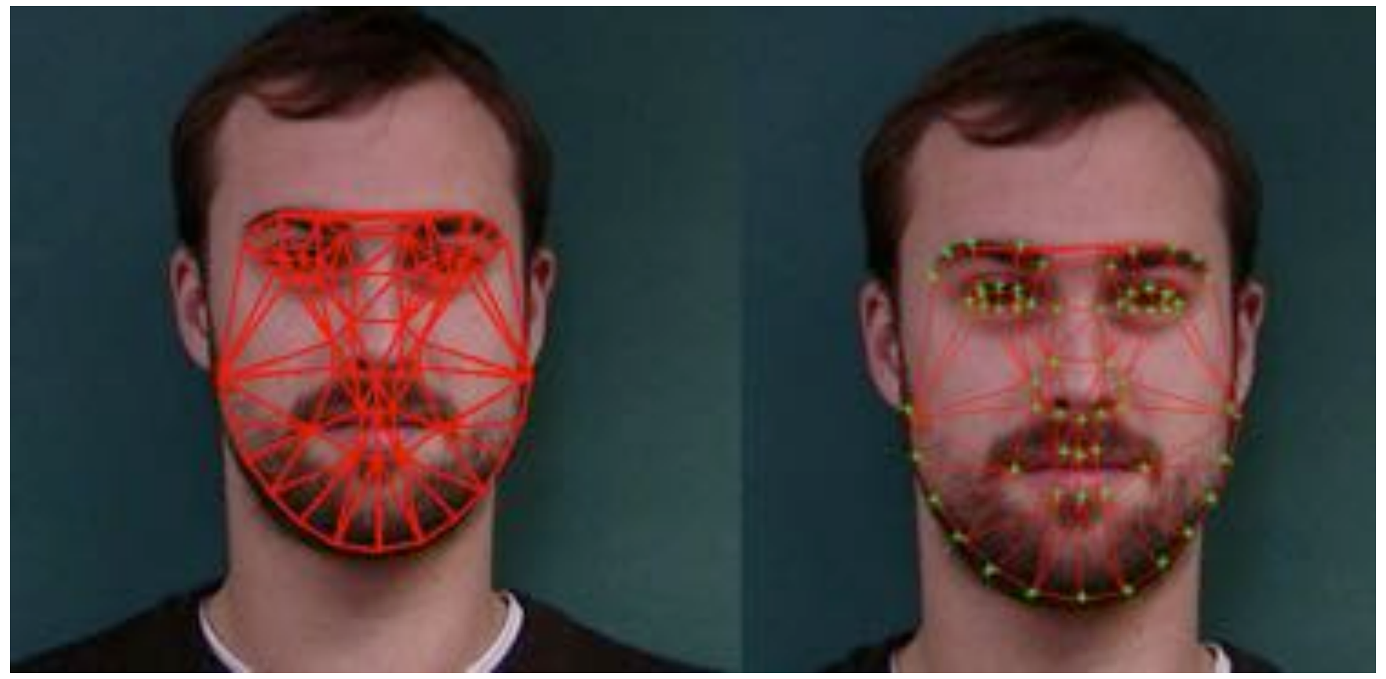
- Nonrigid face alignment (facial landmark localization) with deformable models.
- Range of Applications: Face recognition, emotion recognition, pose estimation, …
- Cascaded Regression Framework.
    - Gradient Descent vs Cascaded Regression.
    - Review of the standard formulation.
- Nonlinear Extension of the Cascaded Regression
    - Linear Shape Model.
    - Nonlinear Regression (via Convolution Neural Network).
- Evaluation Results (LFPW, LFW, HELEN, 300W datasets).



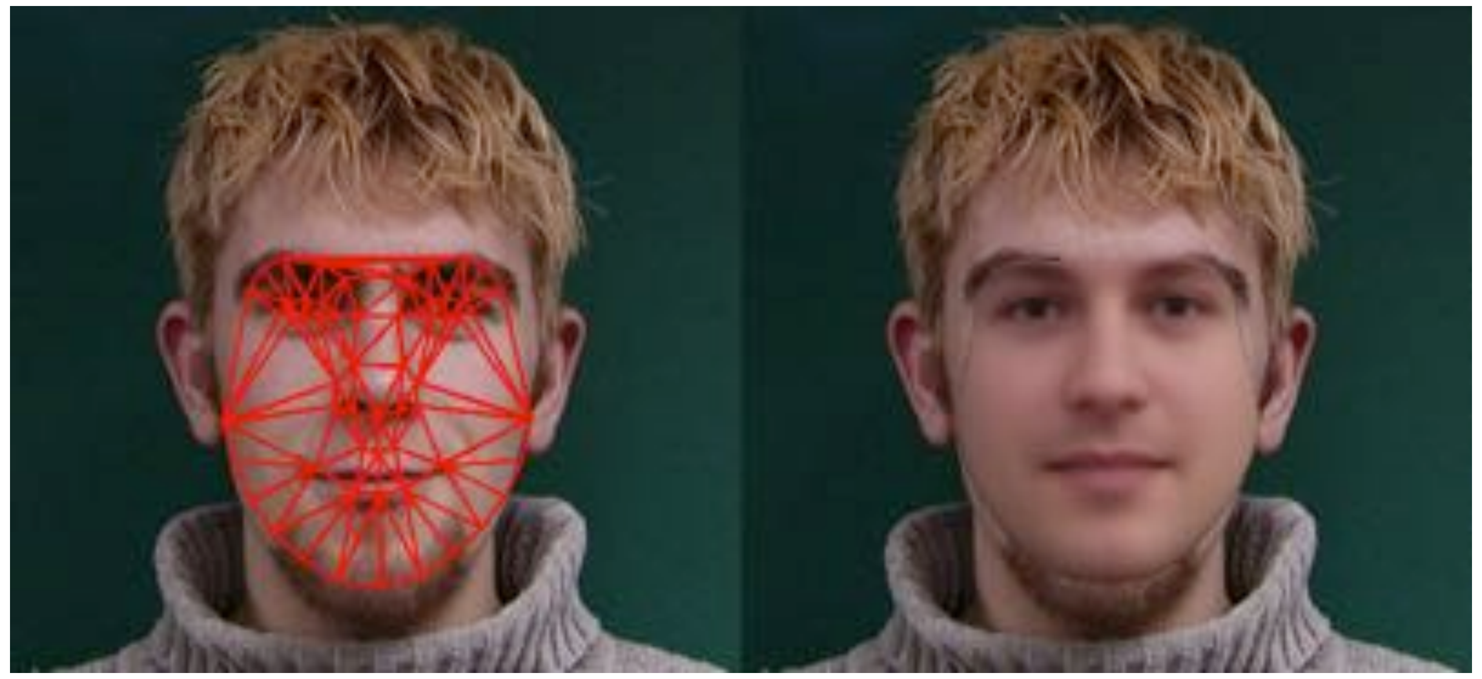
Face alignment example

Related Work

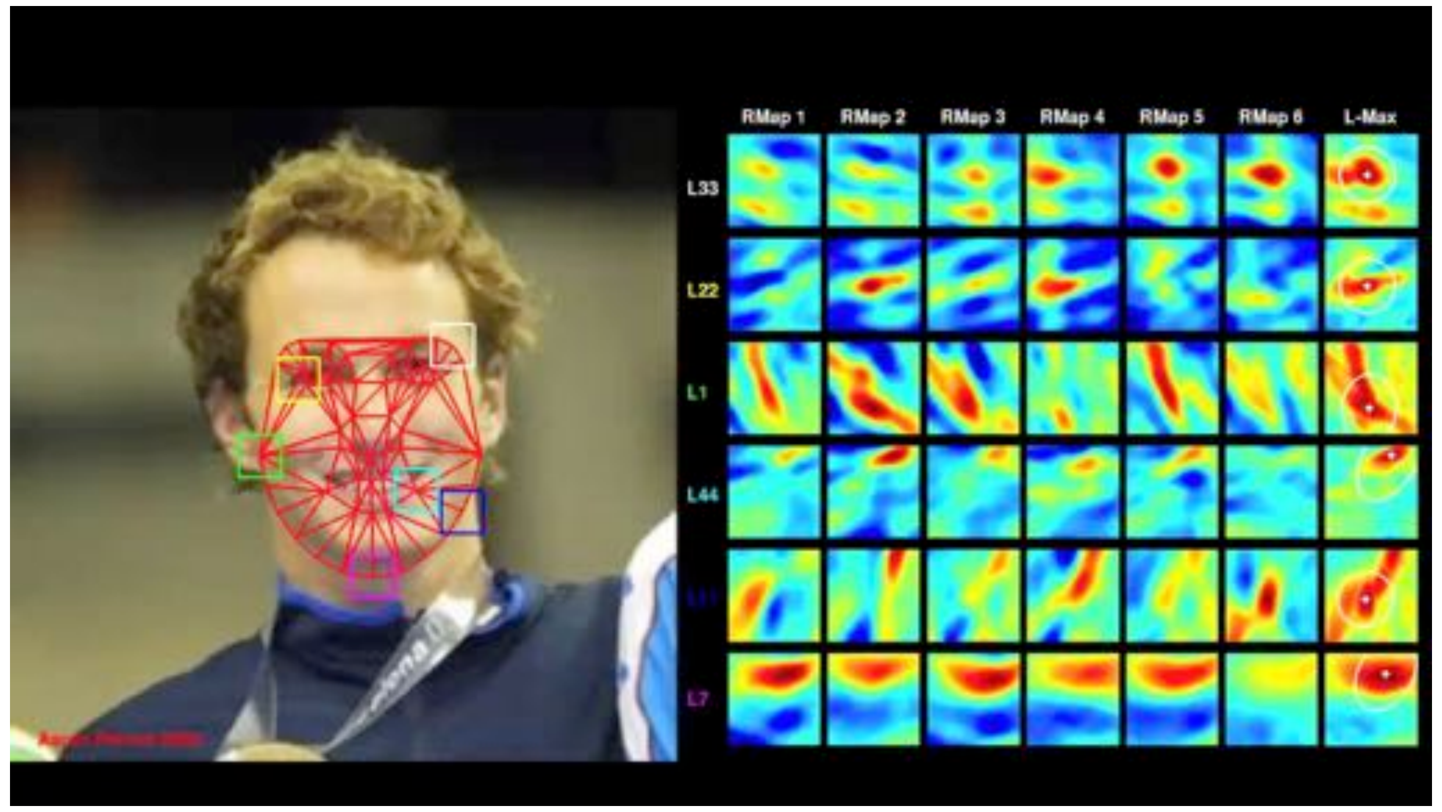
- Active Shape Model (ASM)
- Deformable Part Model (DPM)
- Active Appearance Model (AAM)
    - Project-Out Inverse Compositional (PO-IC)
    - Simultaneous Inverse Compositional (SIC)
- Constrained Local Model (CLM)
    - Convex Quadratic Fitting (CQF)
    - Subspace Constrained Mean-Shifts (SCMS)
    - Bayesian CLM (BCLM)
- Cascaded Regression (CR)
    - Supervised Descent Method (SDM)
    - Project-Out Cascade Regression (PO-CR)
    - Simultaneous Cascaded Regression (SCR)



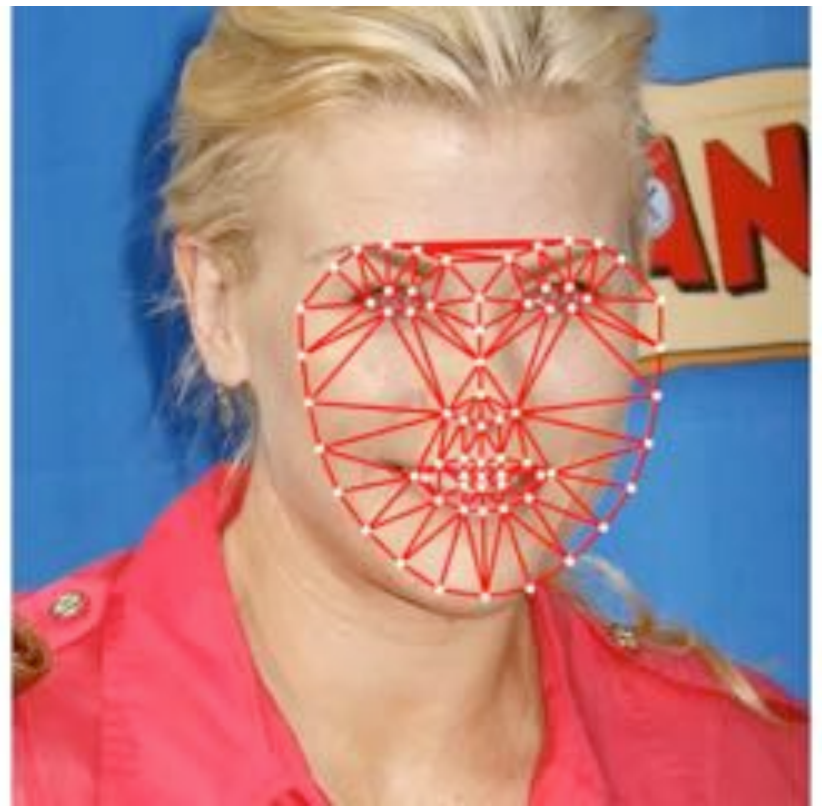
Active Shape Model (ASM)



Active Appearance Model (AAM)



Bayesian Constrained Local Model (BCLM)

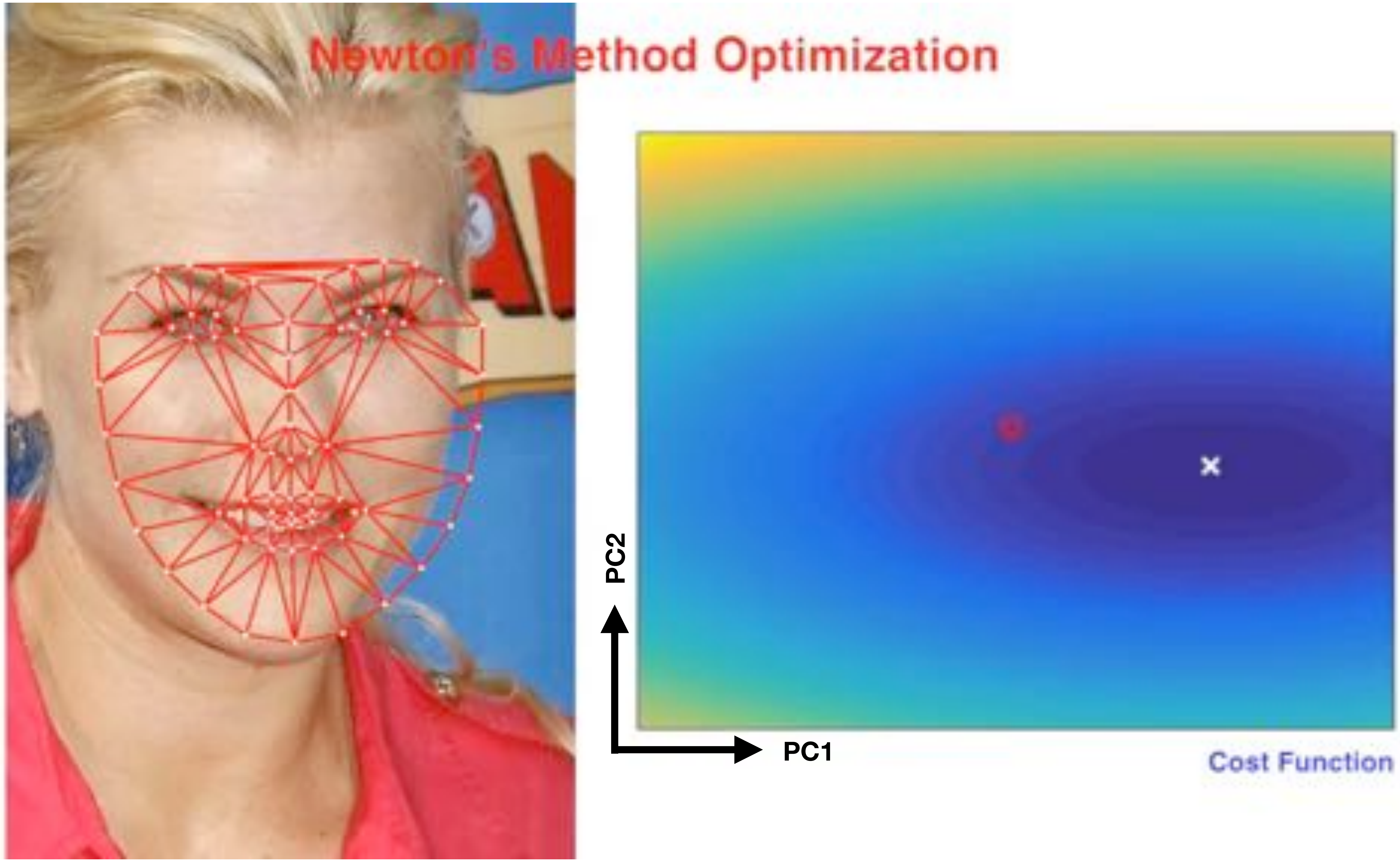


Simultaneous Cascaded Regression (SCR)

Gradient Descent vs. Cascaded Regression

Gradient Descent

- Requires 'good' initialization.
- In general, requires to compute the Jacobian at each iteration.
- Require to compute the Hessian and its inverse (2nd order methods).
- Learning Fast.
- Testing Slow.



Cascaded Regression

- Captures the variance of the initialization.
- Precomputed Regression matrix.
- Learning Slow.
- Testing Fast.

Cascade Regression Framework

Level 1 → Level 2 → Level 3 → • • • Level K

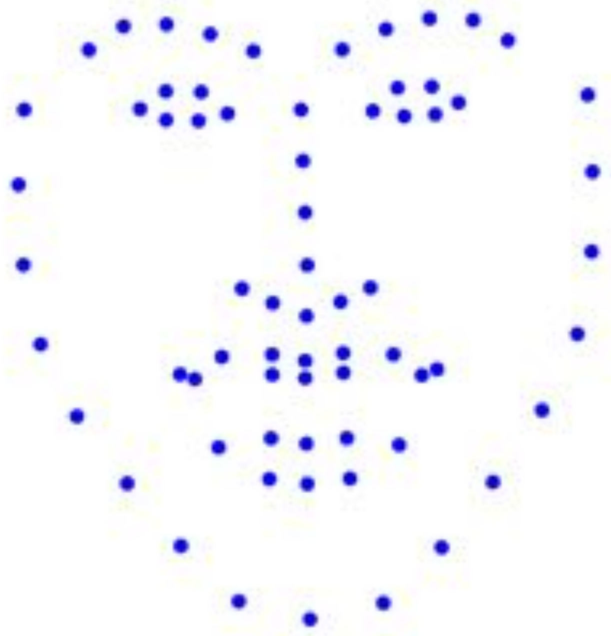
$$\mathbf{s}^k = \mathbf{s}^{k-1} + \mathbf{R}^{k-1}\mathcal{F}(\mathbf{I}, \mathbf{s}^{k-1})$$

k **- cascade level**

$$\mathbf{s} = \begin{pmatrix} x_0 \\ \vdots \\ x_v \\ y_0 \\ \vdots \\ y_v \end{pmatrix}$$

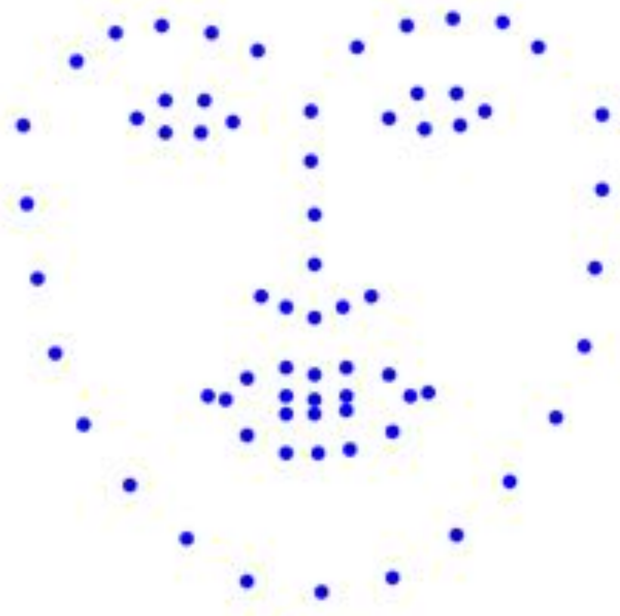
v - landmarks

Updated shape vector



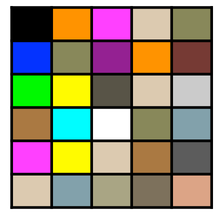
$2v \times 1$

Previous shape vector



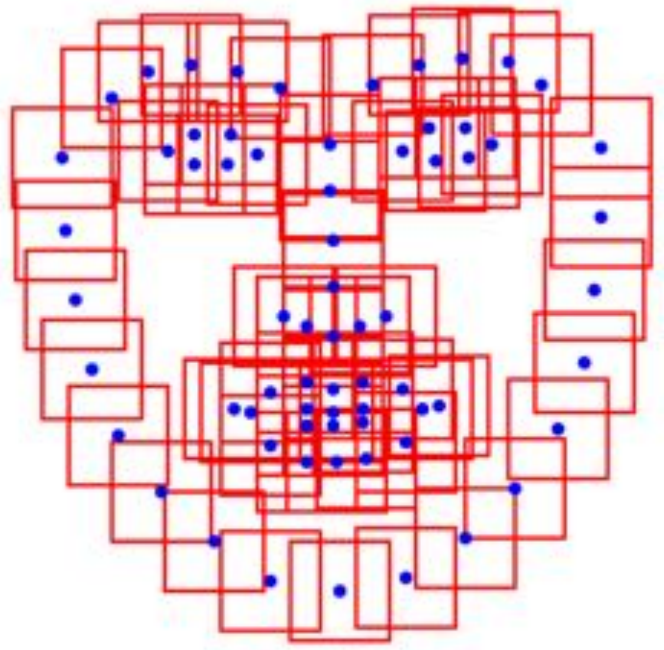
$2v \times 1$

Regression Matrix

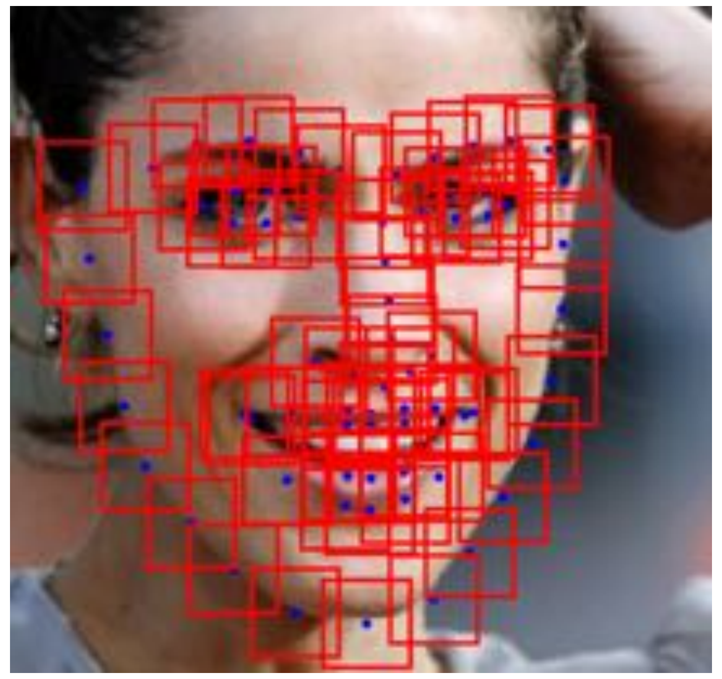


$2v \times d$

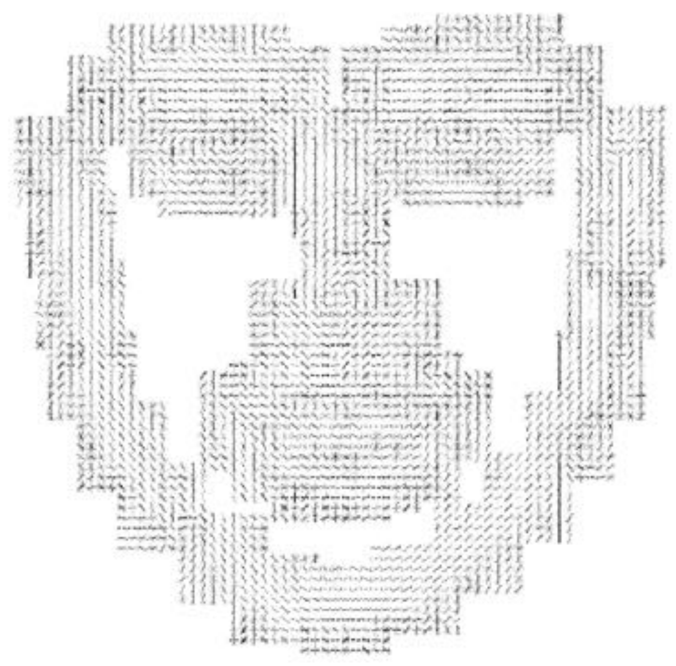
Feature Extraction



$d \times 1$



RGB



HoG

Learning Regression Matrix

Estimate R^k under Multiple Initializations

$$\arg\min_{\mathbf{R}^k} \sum_{i=1}^{N} \sum_{j=1}^{M} \|\Delta \mathbf{s}_j^k - \mathbf{R}^k \mathcal{F}(\mathbf{I}_i, \mathbf{s}_j^k)\|^2$$

k - cascade level
i - training image
j - virtual sample

Estimate noise

$$\Sigma^k = \mathrm{cov}(\mathbf{s}_* - \mathbf{s}_j^k)$$

Deviation from Ground Truth

Regression Labels

$$\Delta \mathbf{s}_j^k = \mathbf{s}_* - \mathbf{s}_j^k$$

Data Matrix (all features)

$$\mathbf{F} = [\ |\ |\ |\ ]$$

N images x M virtual samples

Least Squares Solution

$$\mathbf{R}^k = \Delta \mathbf{S} \left(\mathbf{F}^T \mathbf{F}\right)^{-1} \mathbf{F}^T$$

Linear Regression

Data Collection (F matrix)

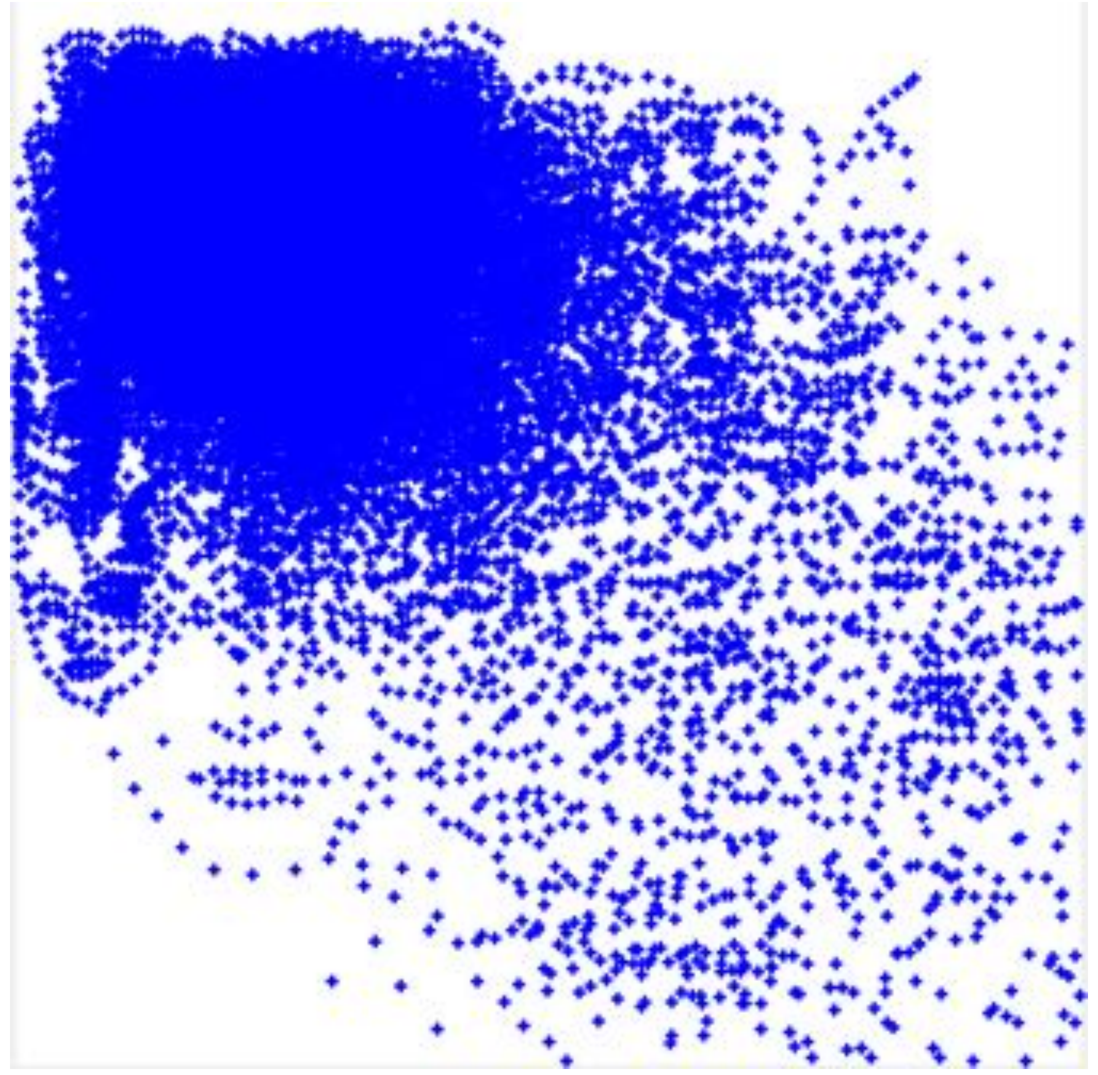
$$\mathbf{s}_j^k \sim \mathcal{N}(\mu^k, \Sigma^k)$$

virtual sample

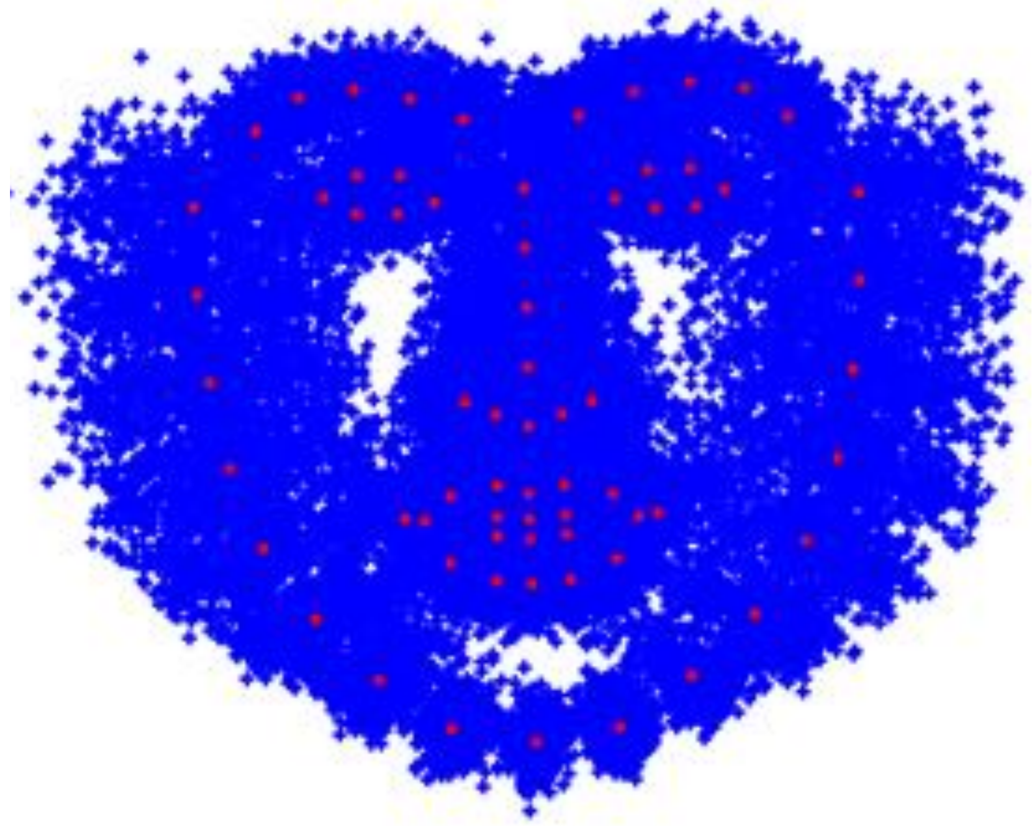
Linear Shape Model

'In the Wild' Image Database

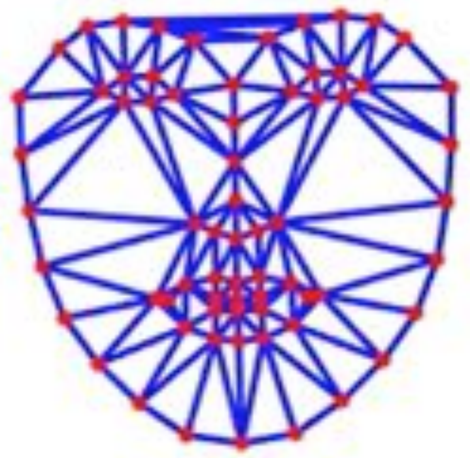
RAW Shape Data



Procrustes Alignment



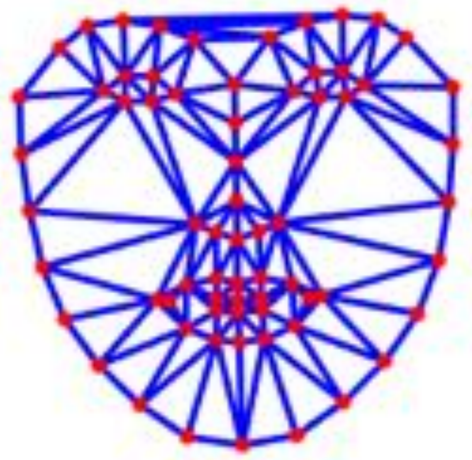
Shape Model



$$\mathcal{B}(\mathbf{s};\mathbf{b}) = \mathbf{s}_0 + \sum_{i=1}^{n} \phi_i b_i$$

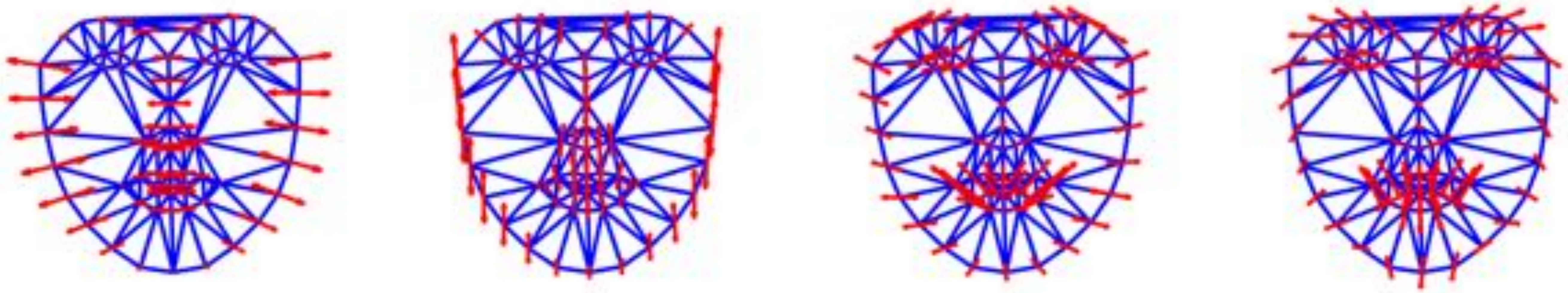
shape parameters

Mean Shape



$\mathbf{s}_0$

Shape Basis



ϕ_1 ϕ_2 ϕ_3 ϕ_4

Similarity Transform

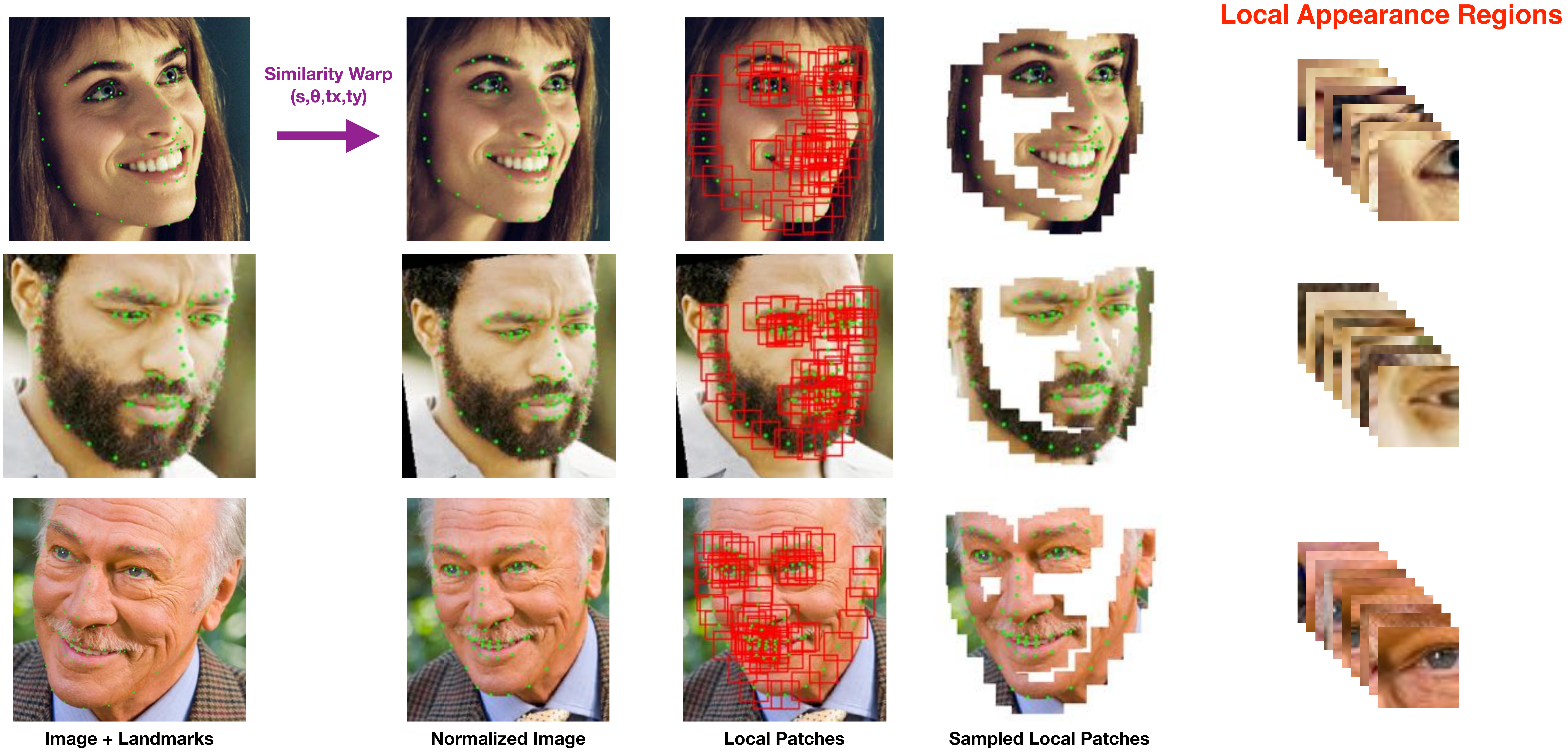
$$\mathcal{S}(\mathbf{s};\mathbf{q}) = \mathbf{s} + \sum_{j=1}^{4} \psi_j q_j$$

pose parameters

Full Shape Model

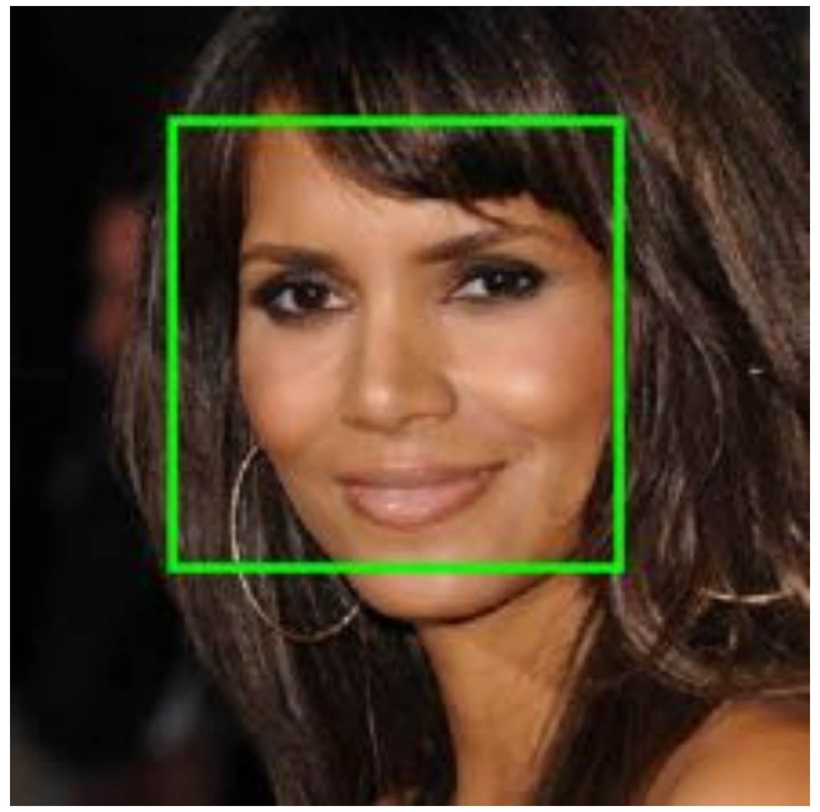
$$\mathbf{s} = \mathcal{S}(\mathcal{B}(\mathbf{b});\mathbf{q})$$

Local Appearance Regions



Initial Shape Estimate ($\mathbf{s}^0$)

Face Detection



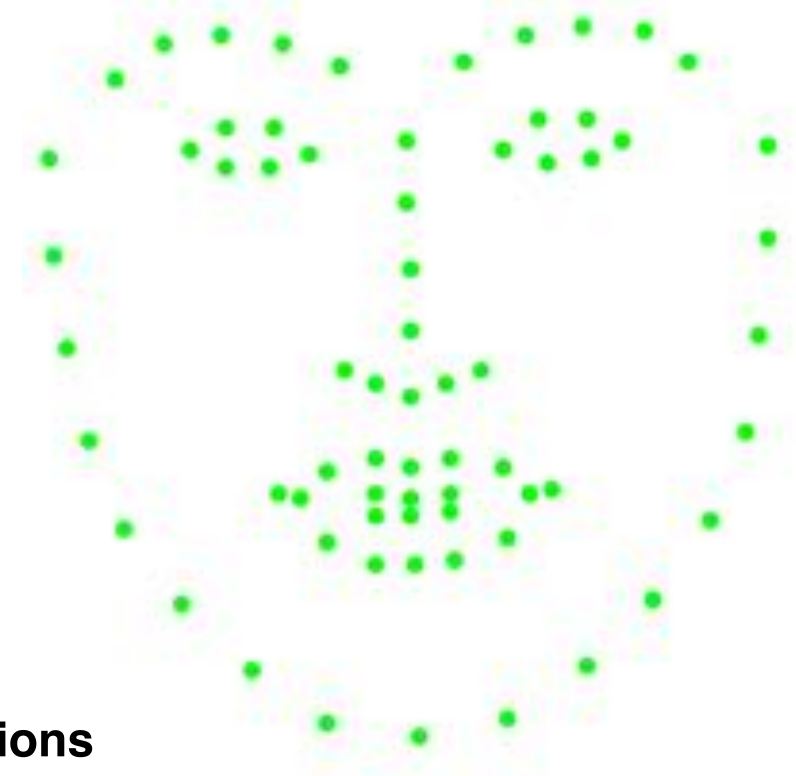
$\mathbf{r} = (x, y, w, h)$

Direct Regression from Rectangles to Shapes

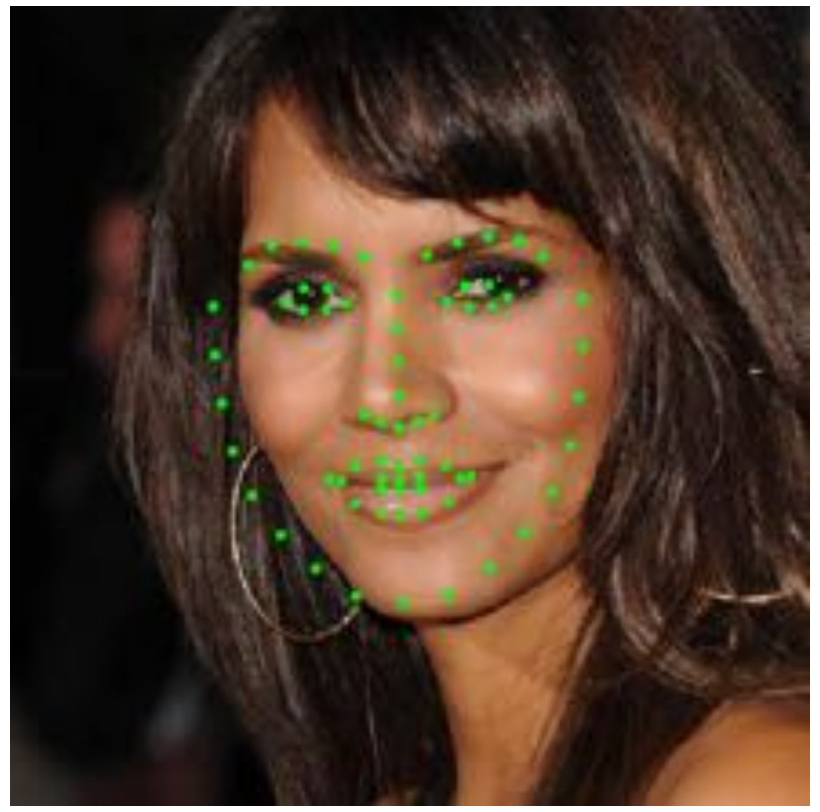
$$\mathbf{R}_d = \mathbf{S}_* \mathrm{Rect}^\dagger$$

all shapes **all detections**

$2v \times 4$

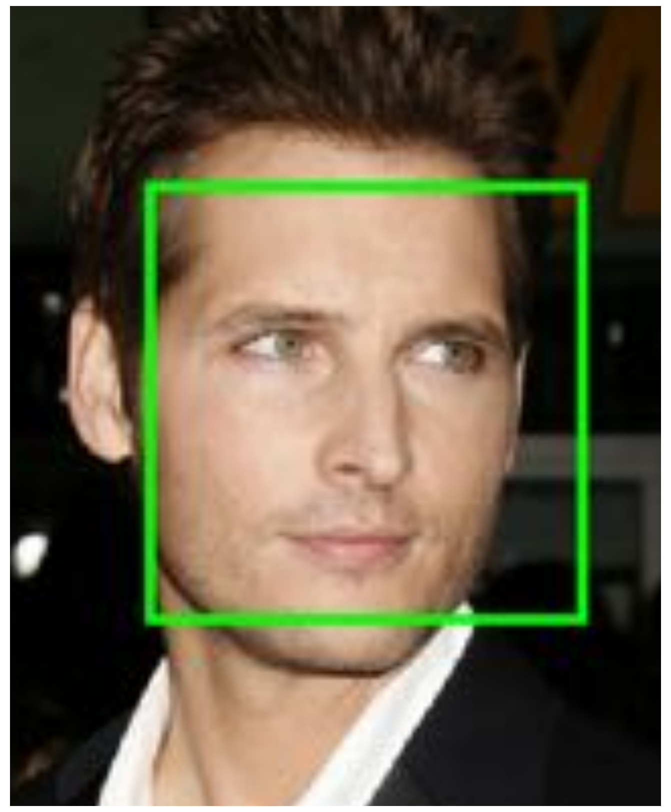


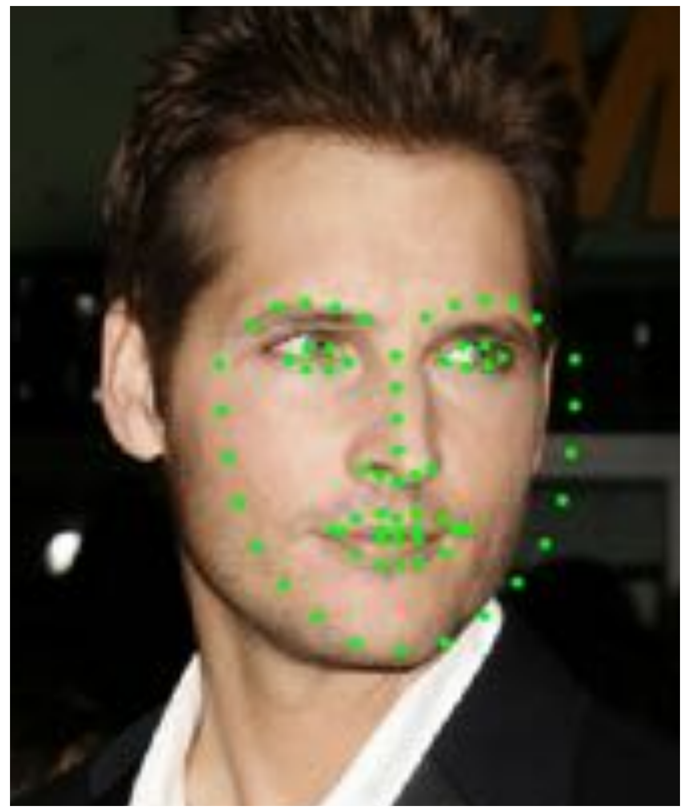
$\mathbf{s}^0 = (x_0, ..., x_v, y_0, ..., y_v)$

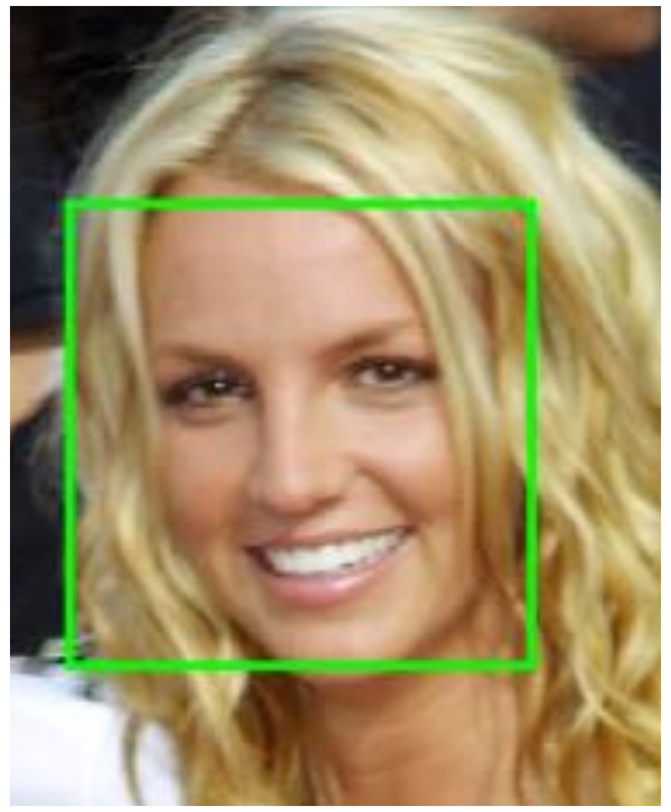


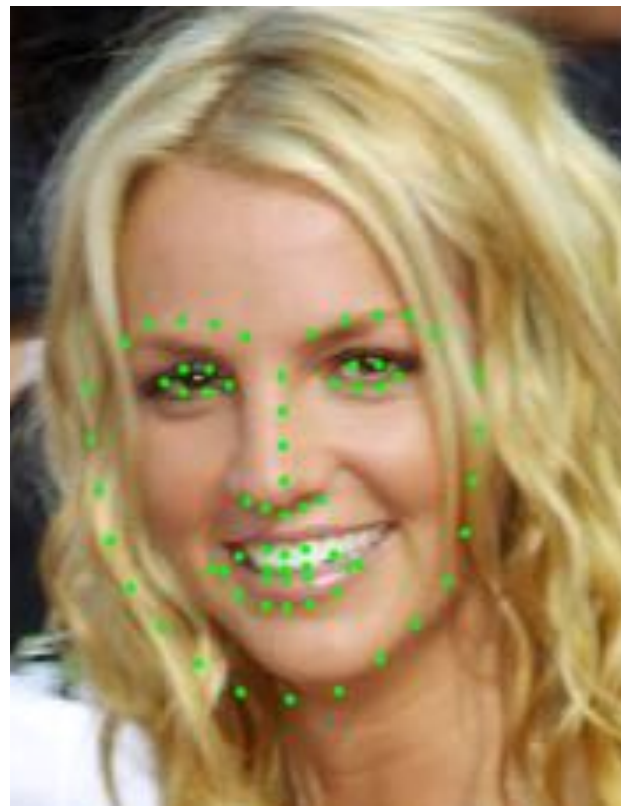
$\mathbf{s}^0 = \mathbf{R}_d \mathbf{r}$

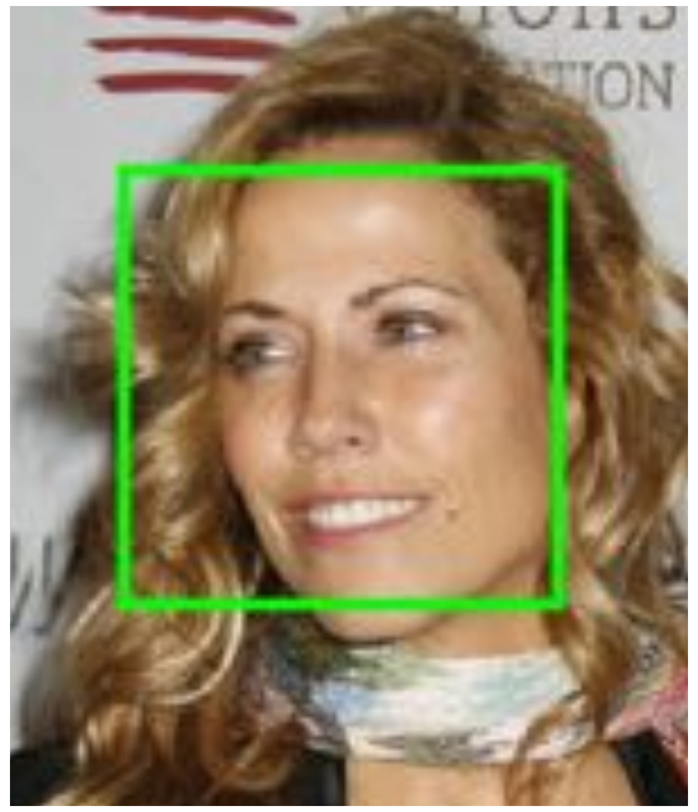
Other Examples:

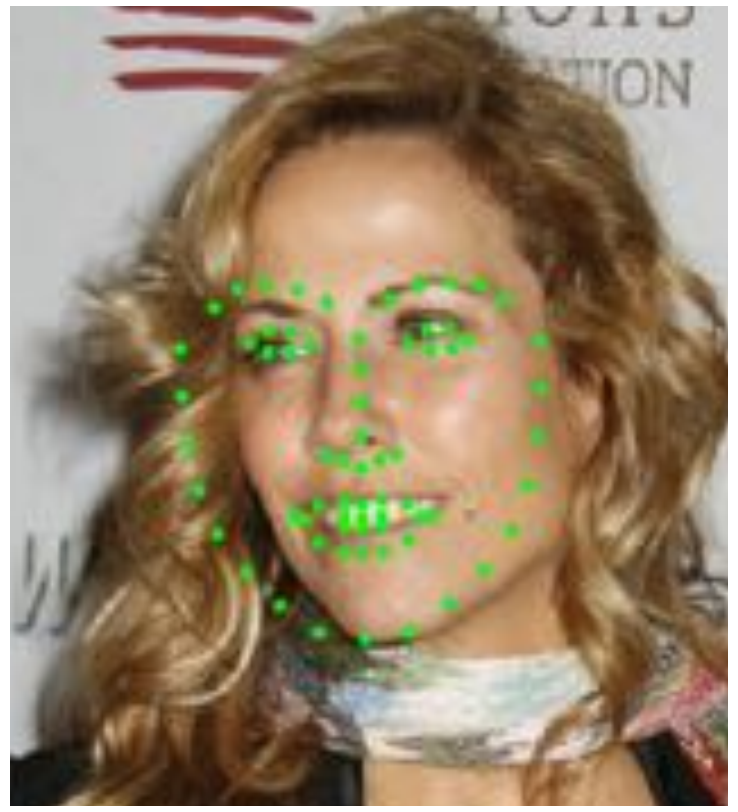




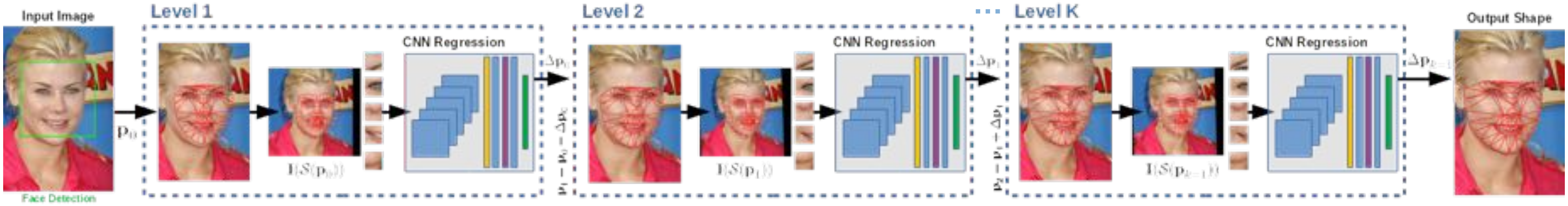








Nonlinear Cascade Regression



Combined shape + pose parameters

$$\mathbf{p} = \begin{bmatrix} \mathbf{b} \\ \mathbf{q} \end{bmatrix} \in \mathbb{R}^{n+4}$$

$$\mathbf{p}^k = \mathbf{p}^{k-1} + \gamma \mathcal{R}^{k-1}\{\mathcal{L}(\mathbf{I}(\mathcal{S}(\mathbf{p}^{k-1})))\}$$

k **- cascade level**

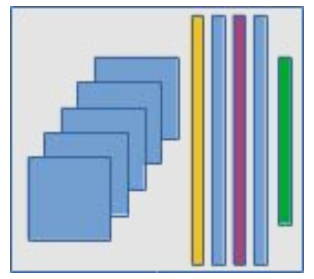
Updated shape instance

$(n+4) \times 1$

Previous shape instance

$(n+4) \times 1$

Nonlinear Mapping



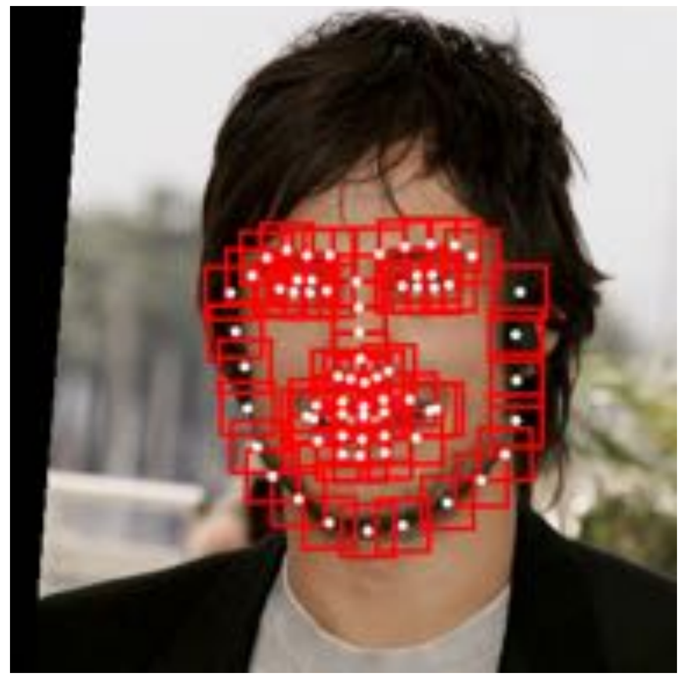
CNN^k

Local Feature Extraction at Normalized Frame

$\mathbf{I}(.)$

Similarity Warp

$\mathbf{p}(n+1 : n+4)$



$\mathbf{I}(\mathcal{S}(\mathbf{p}))$

Sampled 3D Array

$P \times P \times v$

$\mathcal{L}(\mathbf{I}(\mathcal{S}(\mathbf{p})))$

CNN Regression Architecture

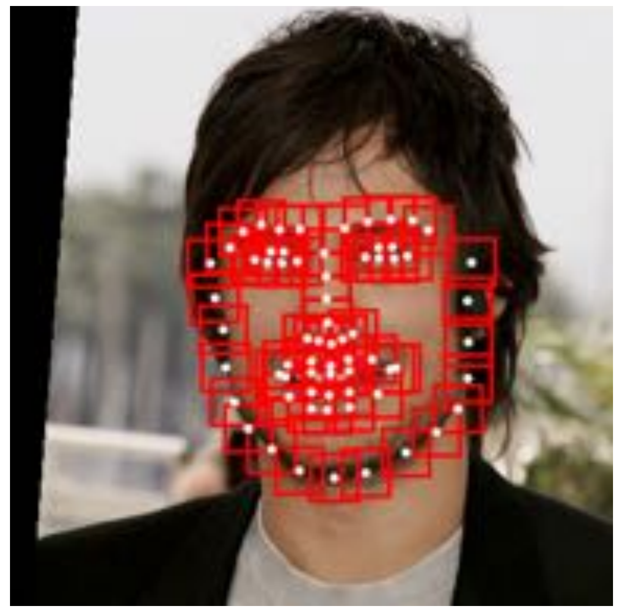
Nonlinear Regression

$$\arg\min_{\mathcal{R}^k} \sum_{i=1}^{N} \sum_{j=1}^{M} \|\Delta \mathbf{p}_{ij}^k - r_L(\ldots r_1(\mathcal{L}(\mathbf{I}_i(\mathcal{S}(\mathbf{p}_j^k)))))\|_{\Sigma^k}^2$$

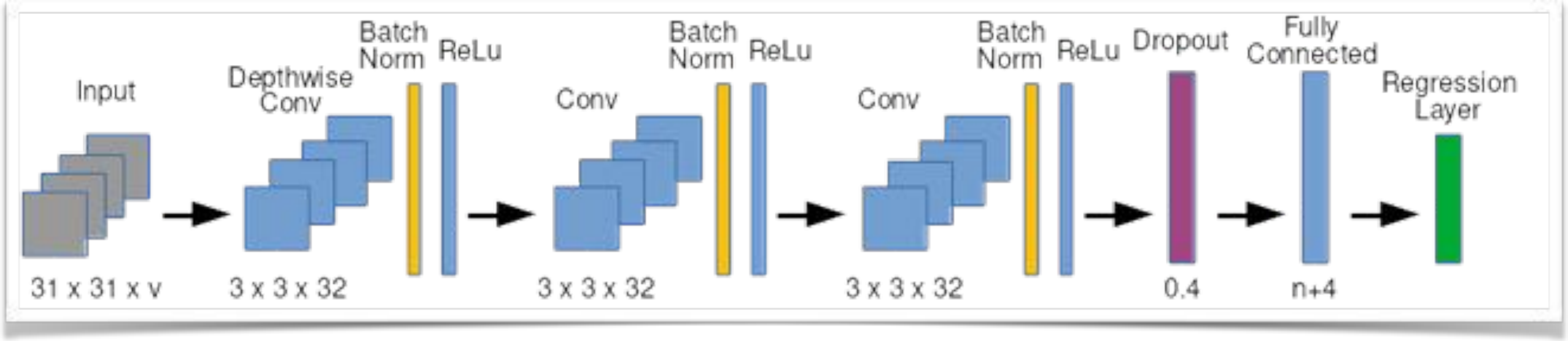
CNN Topology

Input

$\mathcal{L}(\mathbf{I}(\mathcal{S}(\mathbf{p})))$



Pose Normalized Image



Depthwise Convolution 32 filters (3x3) for each local patch

Convolution 32 filters (3x3)

Convolution 32 filters (3x3)

Dropout 0.4

Regression Layer (n+4)

Output

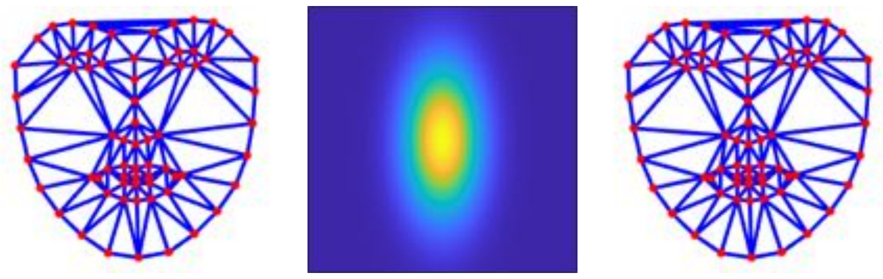
shape parameters update

$\Delta \mathbf{p}$

Loss function

$$L_r = \frac{1}{N} \sum_{j=1}^{N} \Delta \mathbf{p}_j^T \Sigma_{\mathbf{p}}^{-1} \Delta \mathbf{p}_j$$

Mahalanobis Distance



CNN Learning - Data Collection

$$\arg\min_{\mathcal{R}^k} \sum_{i=1}^{N} \sum_{j=1}^{M} \|\Delta \mathbf{p}_{ij}^k - \mathrm{CNN}^k(\mathcal{L}(\mathbf{I}_i(\mathcal{S}(\mathbf{p}_j^k))))\|_{\Sigma^k}^2$$

k - cascade level
i - training image
j - virtual sample

Estimate noise

$$\Sigma^k = \mathrm{cov}(\mathbf{p}_* - \mathbf{p}_{ij})$$

Deviation from Ground Truth

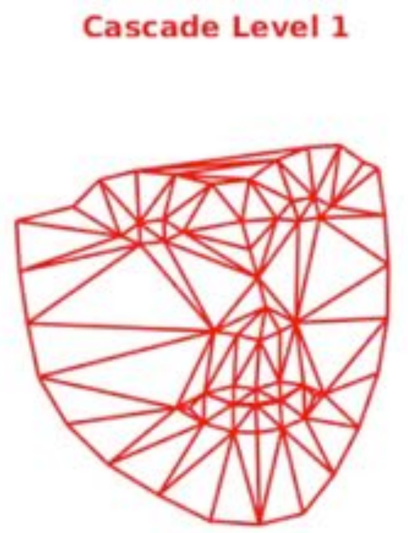
Data Matrix (local normalized patches)

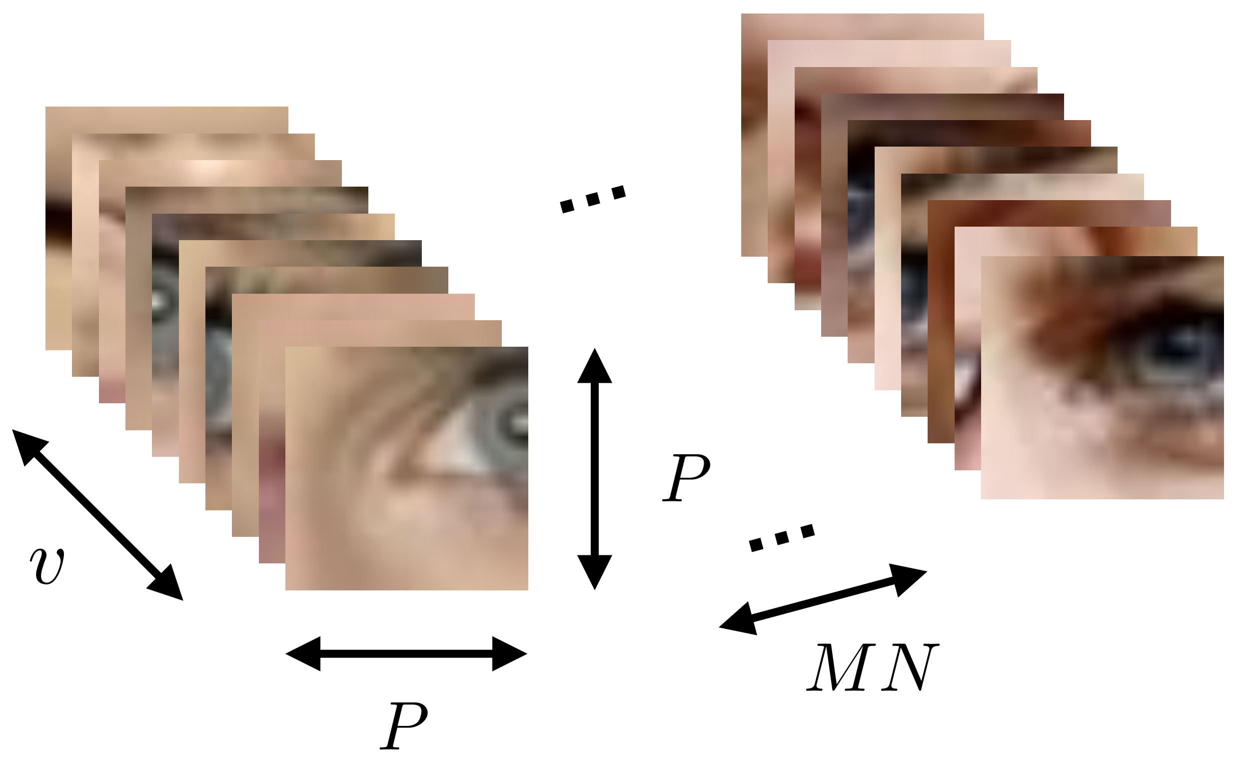
Regression Labels

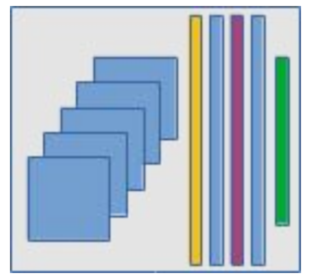
$$\Delta \mathbf{p}_{ij} = \mathbf{p}_* - \mathbf{p}_{ij}$$

$$\mathbf{D}_{ij} = \mathcal{L}(\mathbf{I}_i(\mathcal{S}(\mathbf{p}_j)))$$

Full Data: 4D Array (P x P x v x N.M)

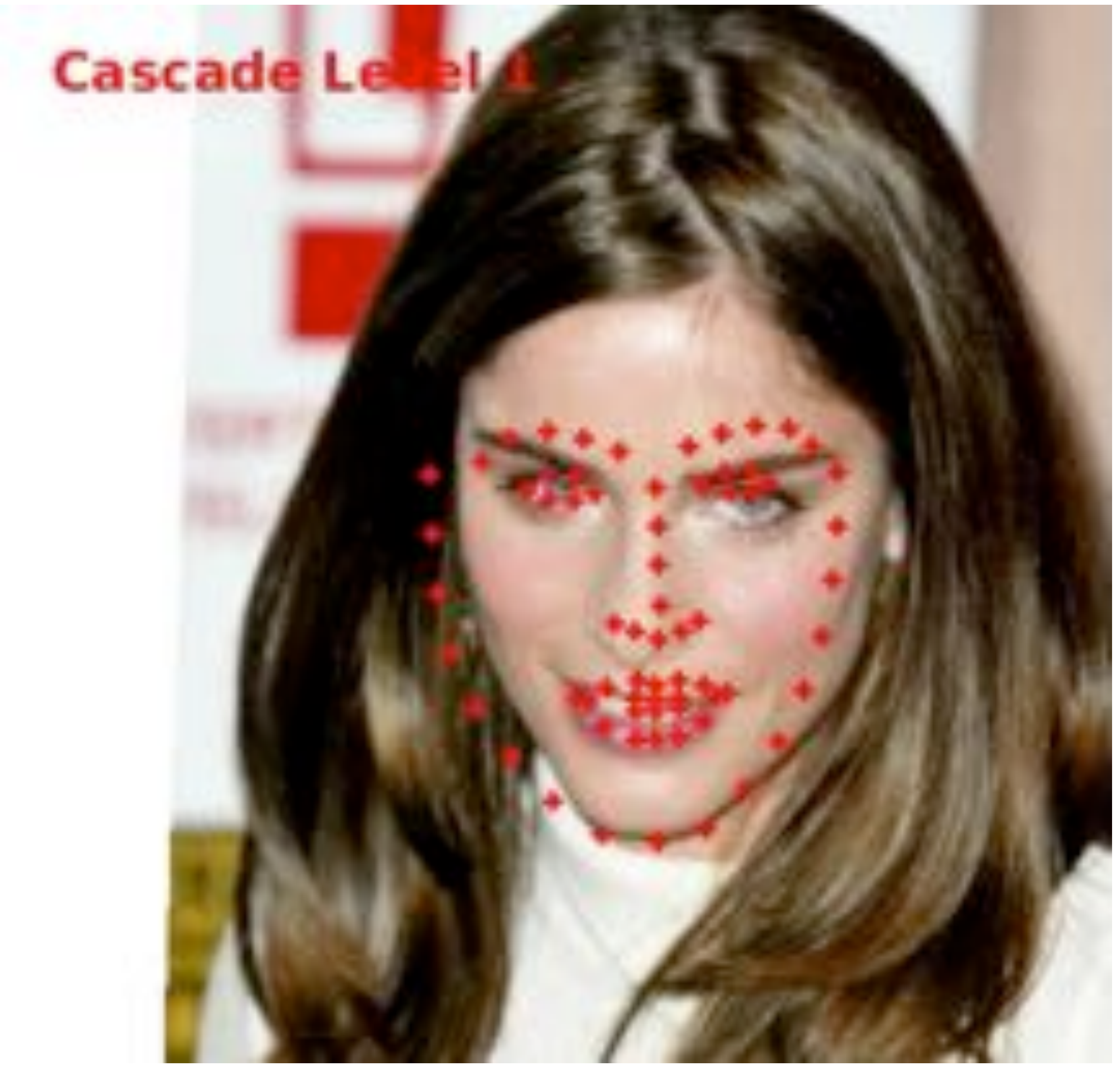






CNN^k

M - augmented examples
N - real examples

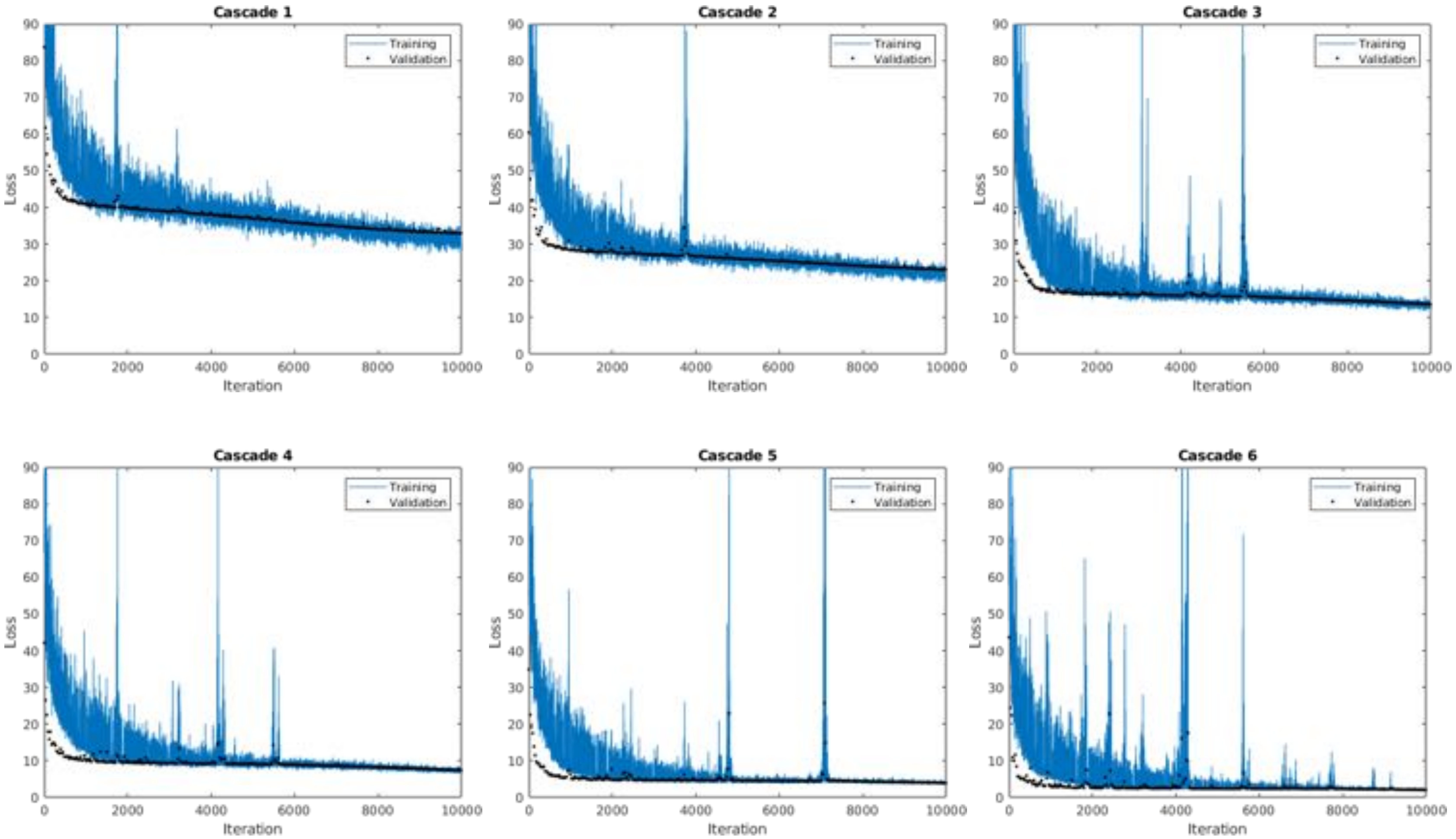


Data Collection (D matrix)

$$\mathbf{p}_{ij} \sim \mathcal{N}(\mathbf{p}_i, \Sigma^k)$$

virtual shape sample

CNN Learning - Optimization Details



	Name	Type	Activations	Learnables
1	imageinput 31x31x68 images	Image Input	31×31×68	-
2	grouped_conv 68 groups of 32 3x3x1 convolutions with stride [1 1] and padding [0 0 0 0]	Grouped Convolution	29×29×2176	Weights 3×3×1×32×68 Bias 1×1×32×68
3	batchnorm_1 Batch normalization with 2176 channels	Batch Normalization	29×29×2176	Offset 1×1×2176 Scale 1×1×2176
4	relu_1 ReLU	ReLU	29×29×2176	-
5	conv_1 32 3x3x2176 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	27×27×32	Weights 3×3×2176×32 Bias 1×1×32
6	batchnorm_2 Batch normalization with 32 channels	Batch Normalization	27×27×32	Offset 1×1×32 Scale 1×1×32
7	relu_2 ReLU	ReLU	27×27×32	-
8	conv_2 32 3x3x32 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	25×25×32	Weights 3×3×32×32 Bias 1×1×32
9	batchnorm_3 Batch normalization with 32 channels	Batch Normalization	25×25×32	Offset 1×1×32 Scale 1×1×32
10	relu_3 ReLU	ReLU	25×25×32	-
11	dropout 40% dropout	Dropout	25×25×32	-
12	fc 31 fully connected layer	Fully Connected	1×1×31	Weights 31×20000 Bias 31×1
13	mahl Mahalanobis sq loss v4	Regression Output	-	-

CNN topology details

CNN Optimization:

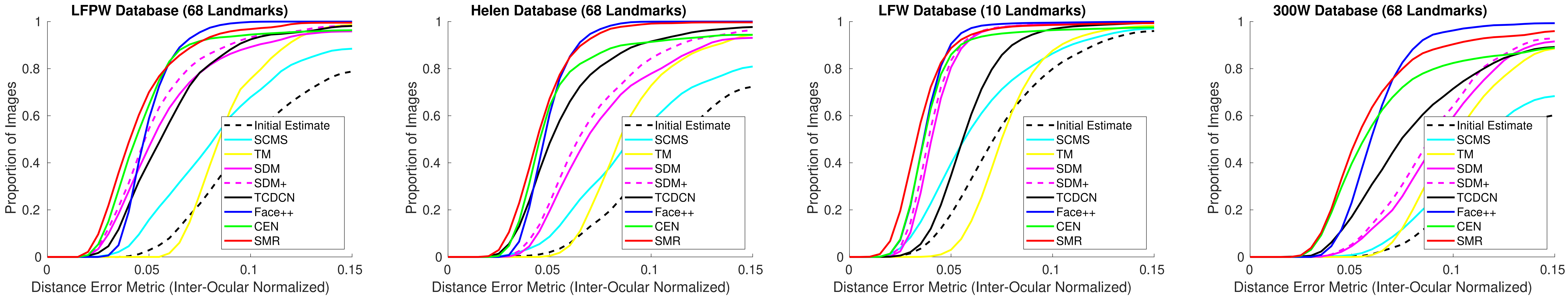
- **Adam solver (default hyperparameters)**
- **Initial learning rate 10^{-4}, exp. decay 0.9 every 5000 iterations**
- **Mini-batch w/ 64 examples**
- **Max cascade levels K = 6**

Demo:

Nonlinear Cascaded SMR Fitting in the LFPW Database

Evaluation Results

Cumulative error distribution function (CDF)

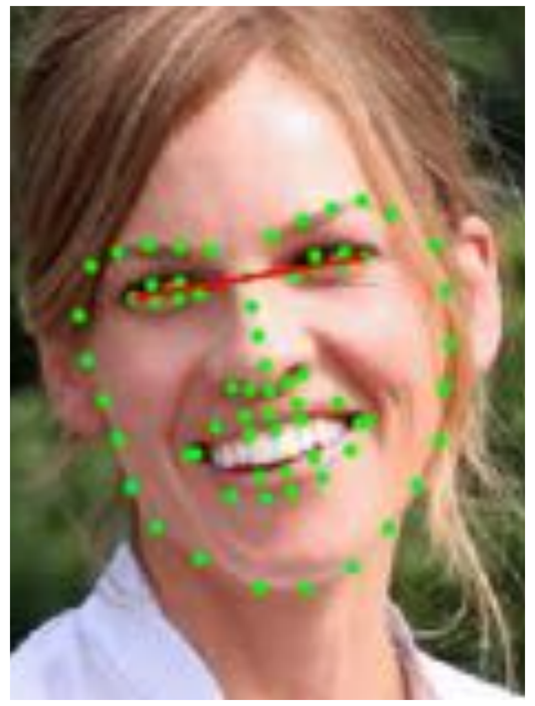


Method / AUC	LFPW	HELEN	LFW	300W
Initial Estimate	30.5	25.4	49	17,6
SCMS	42.4	36.0	57.3	23.2
Tree-Model	40.8	39.7	47.7	30.6
SDM	60.2	48.6	71.5	36.5
SDM+	63.6	52.1	72.4	39.0
TCDCN	59.5	61.1	61.0	44.6
Face++	66.7	67.4	74.8	58.4
CEN	67.2	63.6	72.3	54.0
SMR	69.6	69.1	75.9	59.5

This work → SMR

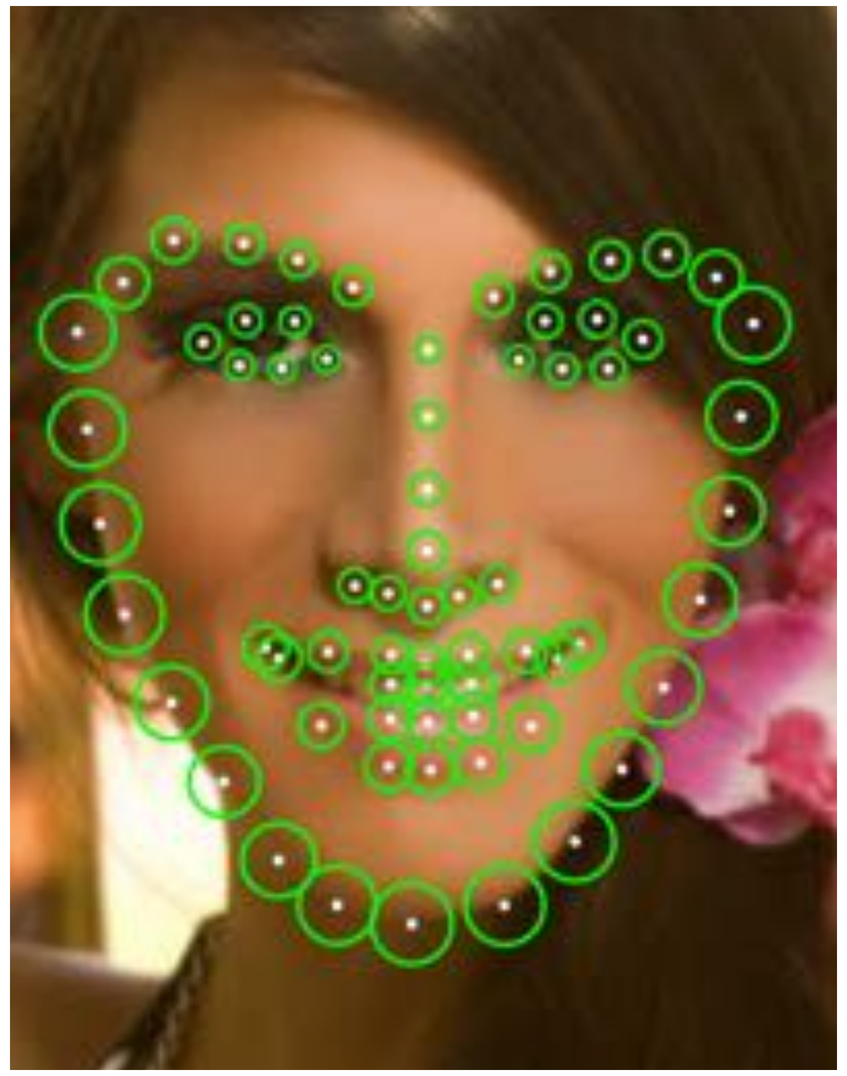
AUC - Area Under Curve

Inter-ocular normalized error

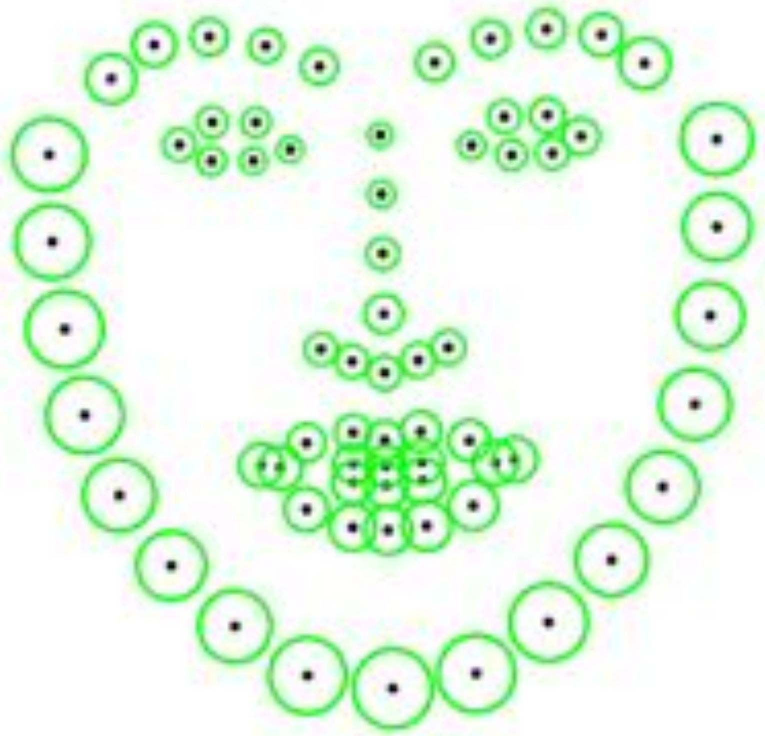


$$e_m(\mathbf{s}) = \frac{1}{v\ d_{\text{eyes}}} \sum_{i=1}^{v} \|\mathbf{s}_i - \mathbf{s}_i^*\|$$

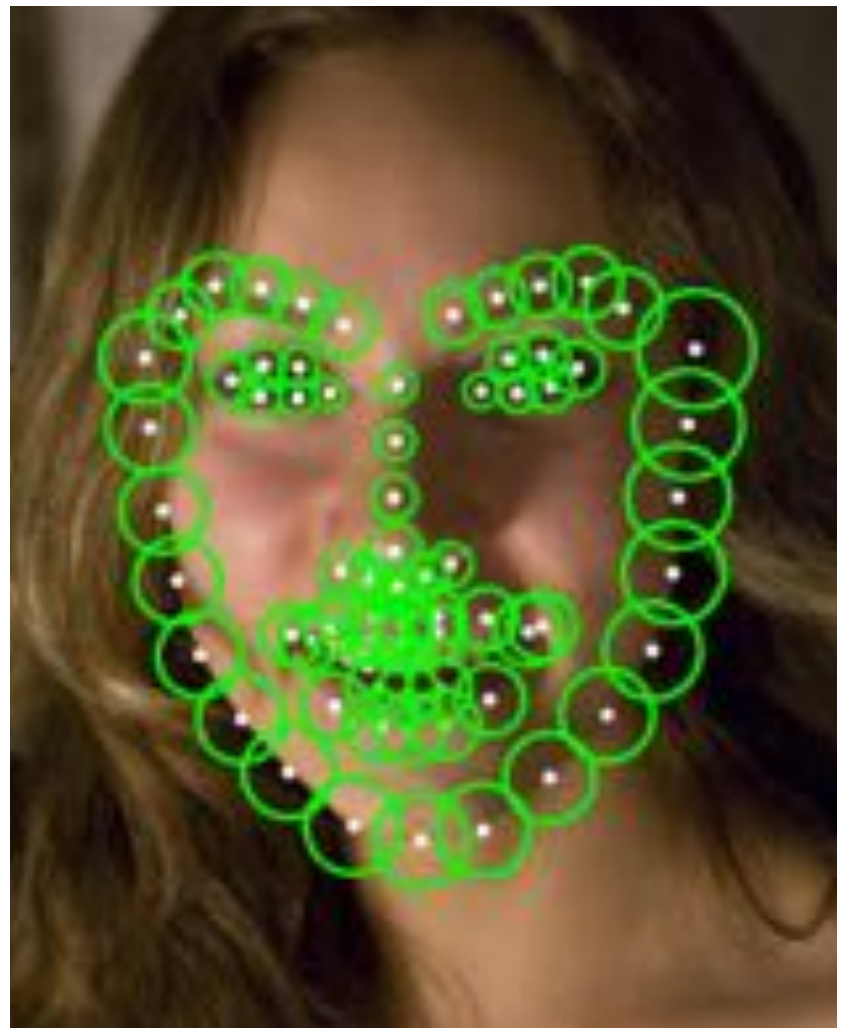
Evaluation Results - Fitting Error Standard Deviation



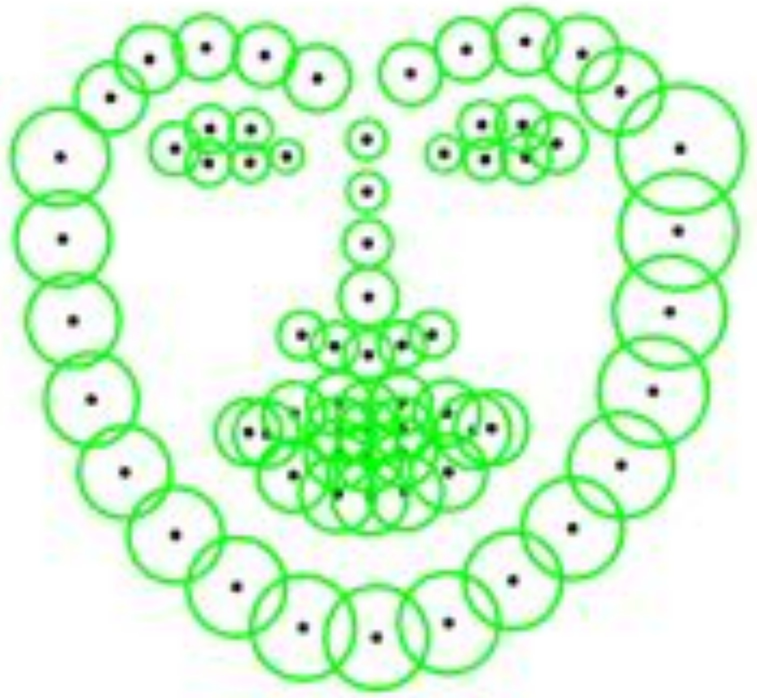
LFPW Database



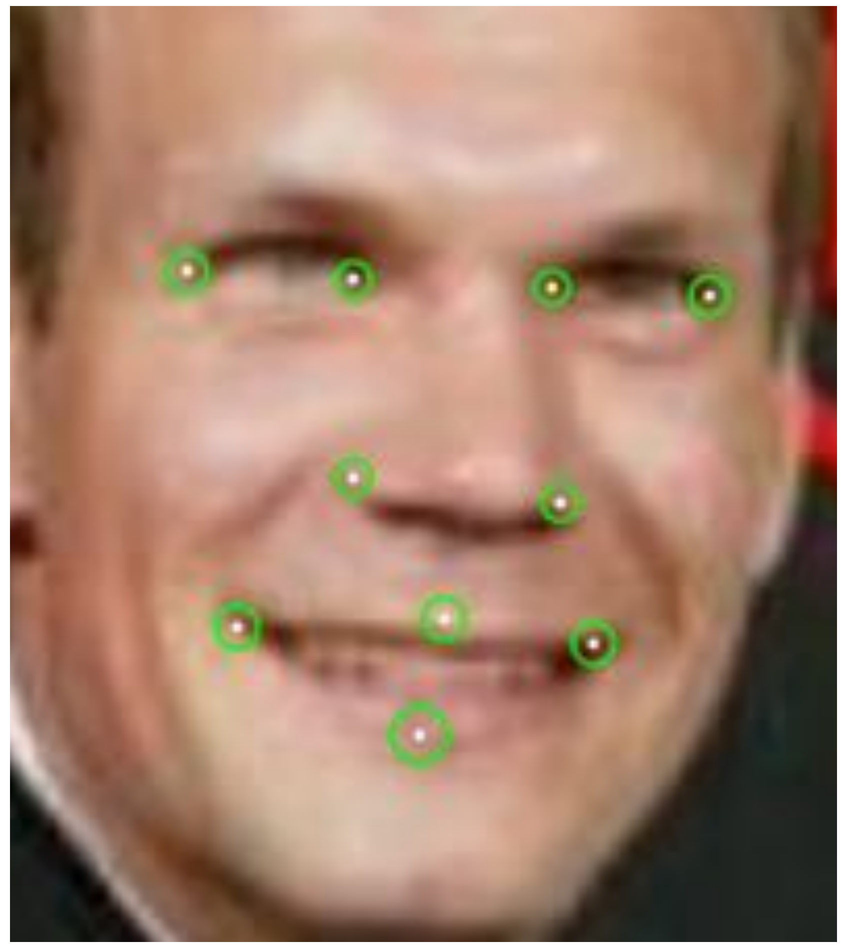
811 train
244 test



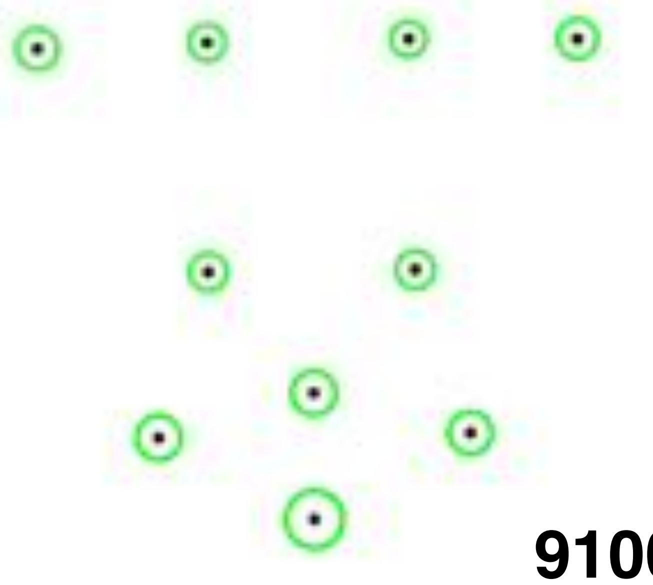
Helen Database



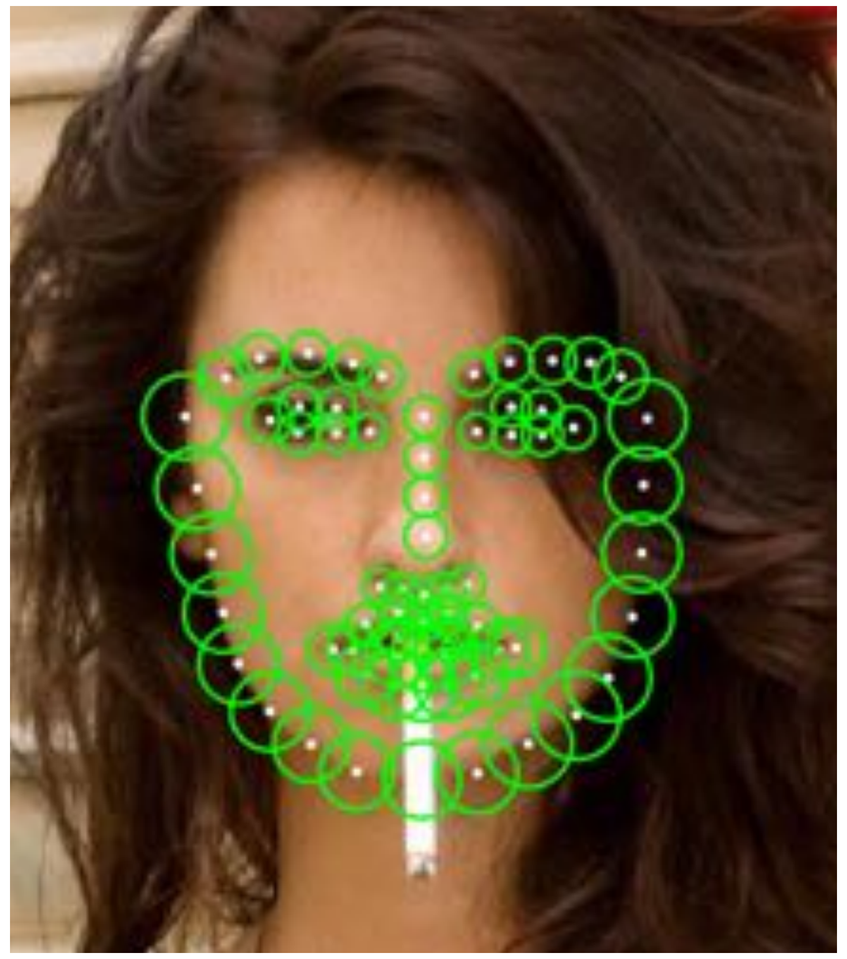
2000 train
300 test



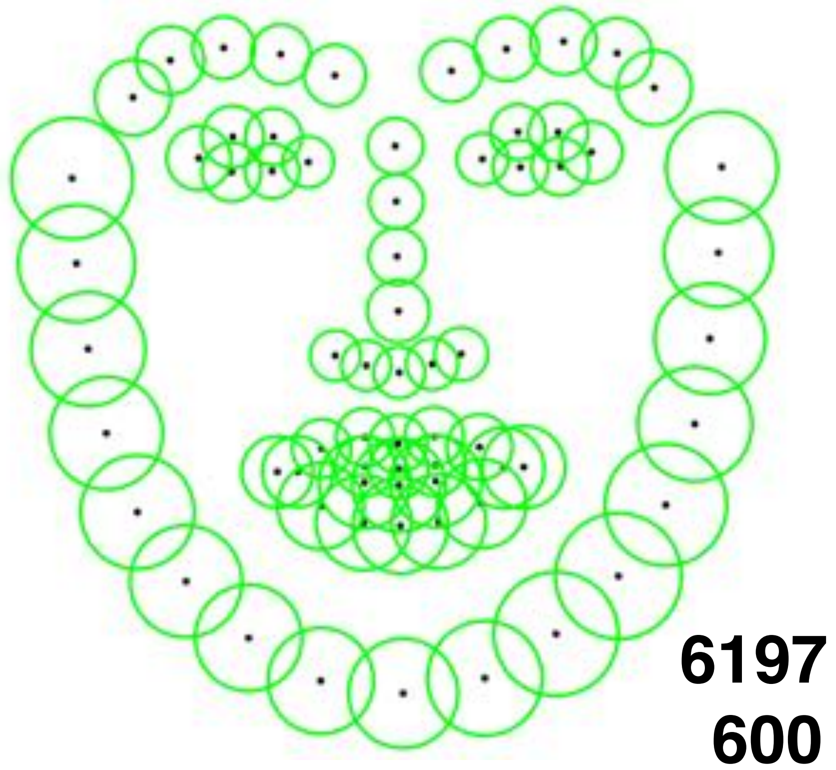
LFW Database



9100 train
3900 test

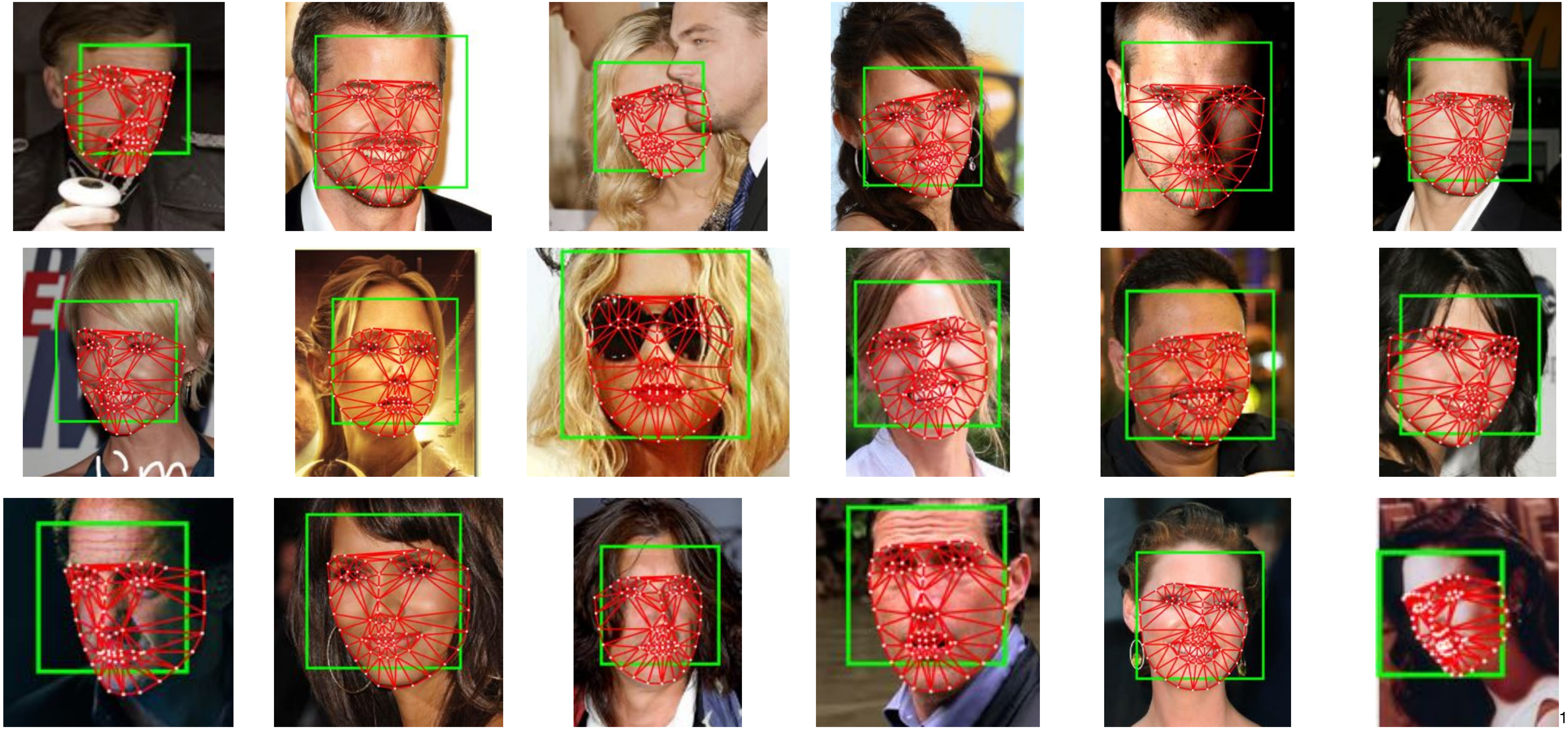


300W Database

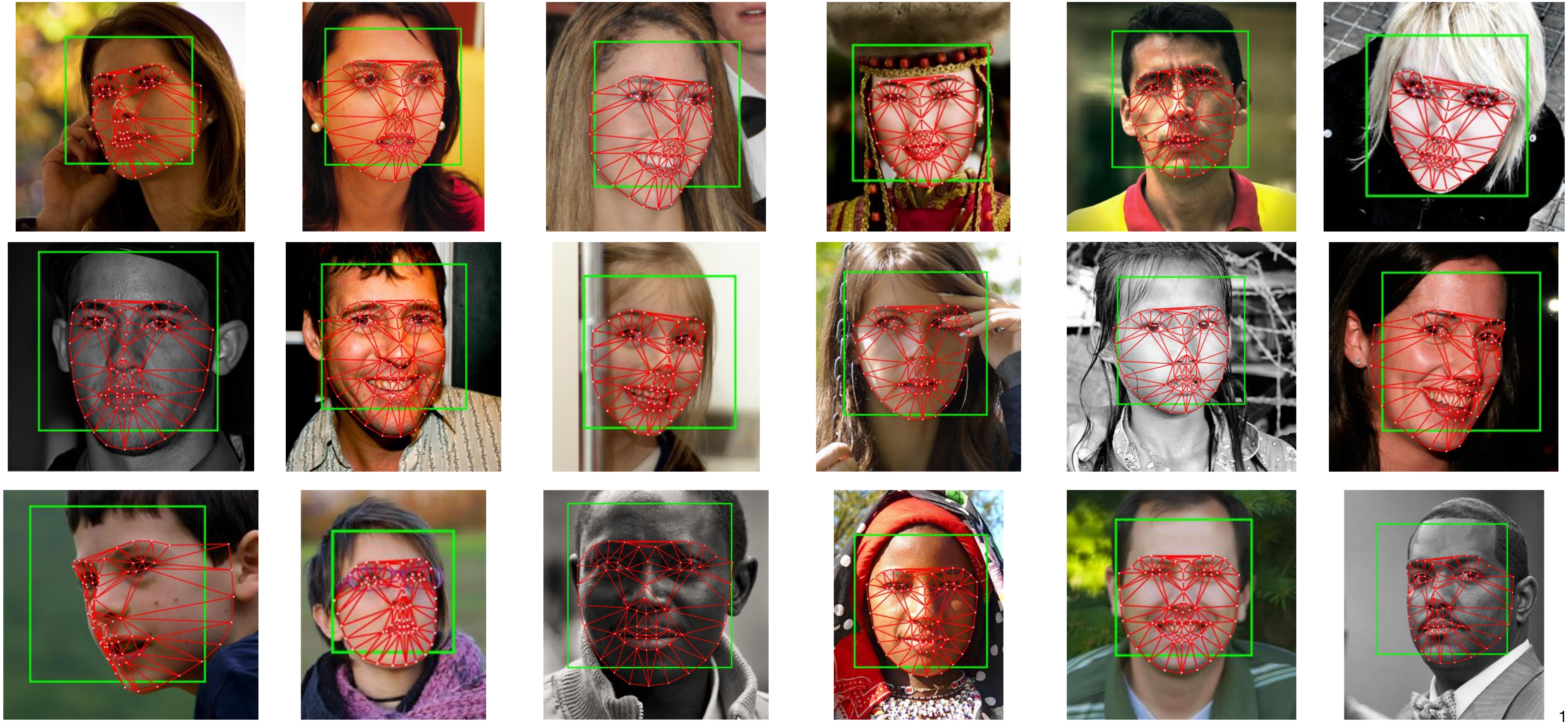


6197 train
600 test

Qualitative Results - LFPW Database



Qualitative Results - HELEN Database



Conclusions

- Proposed an improved face alignment approach (facial landmark localization) w/ deformable face model.
- Nonlinear Cascaded Regression Extension.
  - Compactness of shape model.
  - Enforced shape consistency in Regression (reduced regression effort).
  - Bootstrap model to augment training data (CNN).
  - Loss function, w/ shape aware weighting.
- Demonstrated results in LFPW, HELEN, LFW and 300W datasets.
  - Improvement of ~1.5% AUC (average across all datasets).

Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA
COMPETE
QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL
UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional

Thank you

https://www.isr.uc.pt/~pedromartins
pedromartins@isr.uc.pt