Robust Face Alignment
Using a Mixture of Invariant Experts

Oncel Tuzel[1], Tim K. Marks[1], and Salil Tambe[2]

[1] Mitsubishi Electric Research Labs (MERL)
onceltuzel@gmail.com, tmarks@merl.com
[2] Intel Corporation
salil.tambe@intel.com

Abstract. Face alignment, which is the task of finding the locations of a set of facial landmark points in an image of a face, is useful in widespread application areas. Face alignment is particularly challenging when there are large variations in pose (in-plane and out-of-plane rotations) and facial expression. To address this issue, we propose a cascade in which each stage consists of a mixture of regression experts. Each expert learns a customized regression model that is specialized to a different subset of the joint space of pose and expressions. The system is invariant to a predefined class of transformations (e.g., affine), because the input is transformed to match each expert's prototype shape before the regression is applied. We also present a method to include deformation constraints within the discriminative alignment framework, which makes our algorithm more robust. Our algorithm significantly outperforms previous methods on publicly available face alignment datasets.

1 Introduction

Face alignment refers to finding the pixel locations of a set of predefined facial landmark points (e.g., eye and mouth corners) in an input face image. It is important for many applications such as human-machine interaction, videoconferencing, gaming, and animation, as well as numerous computer vision tasks including face recognition, face tracking, pose estimation, and expression synthesis. Face alignment is difficult due to large variations in factors such as pose, expression, illumination, and occlusion.

1.1 Previous work

Great strides have been made in the field of face alignment since the Active Shape Model (ASM) [1] and Active Appearance Model (AAM) [2] were first proposed. AAM-based face alignment methods proposed since then include [3,4,5]. To handle wider variations in pose, multi-view AAM and ASM models [6,7,8] explicitly model and predict the head pose, e.g., by learning a different deformable model for each of several specific pose ranges [7,8]. Another line of research involves multi-camera AAMs, in which an AAM is simultaneously fitted to images of a

face captured by multiple cameras [9,10]. Like ASMs and AAMs, Constrained Local Models (CLMs) [11,12,13,14] have explicit joint constraints on the landmark point locations (e.g., a subspace shape model) that constrain the positions of the landmarks with respect to each other. Building on CLMs, [15] propose the Gauss-Newton Deformable Part Model (GN-DPM), which uses Gauss-Newton optimization to jointly fit an appearance model and a global shape model.

Recently, much of the focus in face alignment research has shifted toward discriminative methods [16,17,18,19,20,21,22]. These methods learn an explicit regression that directly maps the features extracted at the facial landmark locations to the face shape (e.g., the locations of the landmarks) [17,18,23,24,25,26]. In Project-Out Cascaded Regression (PO-CR) [26], the regression is performed in a subspace orthogonal to facial appearance variation. To cope with inaccurate initialization, [27] begin a regression cascade at multiple initial locations and combine the results. Tree-based regression methods [23,24,25,28,16] are also gathering interest due to their speed. In [23], a set of local binary features are learned using a random forest regression to jointly learn a linear regression function for the final estimation, while [24] utilize a gradient boosting tree algorithm to learn an ensemble of regression trees. Software libraries such as [29] implement a wide range of face alignment methods.

In the Supervised Descent Method (SDM) [17], a cascade of regression functions operate on extracted SIFT features to iteratively estimate facial landmark locations. An extension of SDM, called Global SDM (GSDM) [30], partitions the parameter space into regions of similar gradient direction, and uses the result from the previous frame of video to determine which region's model to use in the current frame. Unlike our method, which takes individual test images as input, GSDM is a tracking method that requires a video sequence. Other methods that report results only on video input include [31].

A variety of recent face alignment methods incorporate deep neural networks, including deep regression networks [32] and coarse-to-fine neural network approaches [33,34]. A different coarse-to-fine approach is taken by [35]. Other recent variations on face alignment research include methods that are specially designed to handle partially occluded faces [36,37].

1.2   Our approach

Our method is related to SDM [17] in that we also perform a cascade of regressions on SIFT features that are computed at the currently estimated landmark locations. However, our method improves upon SDM in a number of ways. In SDM, the same linear regression function must work across all possible variations in facial expressions and pose, including both in-plane and out-of-plane head rotations. In addition to requiring a large and varied training dataset, this forces the learned regression functions to be too generic, thereby limiting accuracy. We address this shortcoming in two ways. First, we propose a transformation invariance step at each level of the cascade, prior to the regression step, which makes our method invariant to an entire class of transformations. (Here, we choose the class of 2D affine transformations.) As a result, our regression functions do not
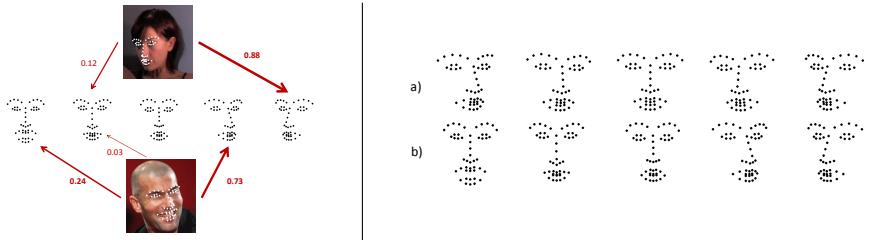
Fig. 1: *Left:* Each expert specializes in a subset of the possible poses and facial expressions. Arrows show the assignment weights of each image's landmark point configuration to the 5 experts. *Right:* Cluster evaluation. a) Euclidean cluster centers. b) Affine-invariant cluster centers. Affine-invariant clustering accounts for both the pose variations and the facial expressions.

need to correct for such global changes in pose and face shape, enabling them to be fine tuned to handle the remaining, smaller variations in landmark locations.

To further improve robustness to variations in pose and expression, at each stage of the cascade we replace the linear regression from SDM by a mixture of experts [38]. In our cascade, each stage is a mixture of experts, where each expert is a regression specialized to handle a subset of the possible face shapes (e.g., a particular region of the joint space of face poses and expressions). As illustrated in Fig. 1 (left), each expert corresponds to a different prototype face shape. This improves alignment significantly, especially when the training dataset is biased towards a certain pose (e.g., frontal).

Unlike alignment methods based on parametric shape models (such as AAM, ASM, and CLM), SDM has no explicit global constraints to jointly limit the locations of multiple landmark points. Our method addresses this limitation simply, by penalizing deviations of landmark locations from each expert's prototype face shape. We accomplish this in the regression framework by extending the feature vector to include the difference between the prototype landmark locations and the currently estimated landmark locations, weighted by a scalar that determines the rigidity of the model. This global regularization of the face shape prevents feature points from drifting apart.

Contributions In summary, we propose a robust method for real-time face alignment which we call **M**ixture of **I**nvariant E**x**perts (MIX). Novel elements include:

- A transformation invariance step, before each stage of regression, which makes our method invariant to a specified class of transformations. (In this study, we choose the class of 2D affine transformations.)
- A simple extension to the feature vectors that enables our regressions to penalize deviations of feature locations from a prototypical face shape.
- A mixture-of-experts regression at each stage of the cascade, in which each expert regression function is specialized to align a different subset of the input data (e.g., a particular range of expressions and poses).

- A novel affine-invariant clustering algorithm to learn the prototype shapes used in the mixture model.

These novel elements enable our method to achieve precise face alignment on a wide variety of images. We perform exhaustive tests on the 300W [39,40] and AFW [41] datasets, comparing with eight recent methods: Coarse-to-Fine Auto-encoder Networks (CFAN) [34], ensemble of regression trees (TREES) [24], Coarse-to-Fine Shape Searching (CFSS) [35], SDM [17], its incrementally learned adaptation CHEHRA [18], GN-DPM [15], Fast-SIC (an AAM method trained on "in-the-wild" images) [5], and PO-CR [26]. We demonstrate that the proposed method significantly outperforms these previous state-of-the-art approaches.

2   Supervised Descent Method

We now describe the Supervised Descent Method (SDM) [17], which is related to our method, while introducing notation that we will use throughout the paper. Let I be an input face image, and let $\mathbf{x}$ be the $2p \times 1$ vector of p facial landmark locations in image coordinates. At each of the p landmark locations in $\mathbf{x}$, we extract a d-dimensional feature vector. In this paper, we use SIFT features [42] with $d = 128$. Let $\phi(I, \mathbf{x})$ be the $pd \times 1$ consolidated feature vector, which is a concatenation of the p feature descriptors extracted from image I at the landmark locations $\mathbf{x}$.

Given a current estimate, $\mathbf{x}_k$, of the landmark locations in image I, SDM formulates the alignment problem as finding an update vector $\Delta\mathbf{x}$ such that the features computed at the new landmark locations $\mathbf{x}_k + \Delta\mathbf{x}$ better match the features computed at the ground-truth landmark locations $\hat{\mathbf{x}}$ in the face image. The corresponding error can be written as a function of the update vector $\Delta\mathbf{x}$:

$$f(\mathbf{x}_k + \Delta\mathbf{x}) = \left\| \phi(I, \mathbf{x}_k + \Delta\mathbf{x}) - \hat{\phi} \right\|^2, \tag{1}$$

where we define $\hat{\phi} = \phi(I, \hat{\mathbf{x}})$. This function f could be minimized by Newton's method. The Newton step is given by

$$\Delta\mathbf{x} = -\mathbf{H}^{-1}\mathbf{J}_f = -2\mathbf{H}^{-1}\mathbf{J}_\phi \left[\phi_k - \hat{\phi} \right], \tag{2}$$

where $\mathbf{H}$ is the Hessian matrix of f, $\mathbf{J}_f$ and $\mathbf{J}_\phi$ represent the Jacobian with respect to $\mathbf{x}$ of f and ϕ, respectively, and we define $\phi_k = \phi(I, \mathbf{x}_k)$. The Hessian and Jacobian in (2) are evaluated at $\mathbf{x}_k$, but we have omitted the argument $\mathbf{x}_k$ to emphasize the dependence on ϕ_k. In SDM, (2) is approximated by the multivariate linear regression

$$\Delta\mathbf{x} = \mathbf{W}_k \phi_k + \mathbf{b}_k, \tag{3}$$

in which coefficients $\mathbf{W}_k$ and bias $\mathbf{b}_k$ do not depend on $\mathbf{x}_k$.

In SDM [17], a cascade of K linear regressions $\{\mathbf{W}_k, \mathbf{b}_k\}$, where $k = 1, \ldots, K$, are learned using training data. Face alignment is achieved by sequentially applying the learned regressions to features computed at the landmark locations

output by the previous stage of the cascade:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{W}_k\boldsymbol{\phi}_k + \mathbf{b}_k. \tag{4}$$

To learn the regressions $\{\mathbf{W}_k, \mathbf{b}_k\}$, the N face images in the training data are augmented by repeating every training image M times, each time perturbing the ground-truth landmark locations by a different random displacement. For each image I_i in this augmented training set $(i = 1, \ldots, MN)$, with ground-truth landmark locations $\hat{\mathbf{x}}_i$, we displace the landmarks by random displacement $\Delta\hat{\mathbf{x}}_i$. The first regression function $(k = 1)$ is learned by minimizing the L2-loss function

$$\{\mathbf{W}_k, \mathbf{b}_k\} = \arg\min_{\mathbf{W}, \mathbf{b}} \sum_{i=1}^{MN} \|\Delta\hat{\mathbf{x}}_i - \mathbf{W}\boldsymbol{\phi}(I_i, \hat{\mathbf{x}}_i - \Delta\hat{\mathbf{x}}_i) - \mathbf{b}\|^2. \tag{5}$$

For training the later regressions $\{\mathbf{W}_k, \mathbf{b}_k\}_{k=2,\ldots,K}$, rather than using a random perturbation, the target $\Delta\hat{\mathbf{x}}_i$ is the residual after the previous stages of the regression cascade.

3 Mixture of invariant experts

In this section, we present our model. Our model significantly improves upon the alignment accuracy and robustness of SDM by introducing three new procedures: a transformation invariance step before each stage of regression, learned deformation constraints on the regressions, and the use of a mixture of expert regressions rather than a single linear regression at each stage of the cascade.

3.1 Transformation invariance

In order for the regression functions in SDM [17] to learn to align facial landmarks for any face pose and expression, the training data must contain sufficiently many examples of faces covering the entire space of possible variations. Although being able to align faces at any pose is a desired property, learning such a function requires collecting (or synthesizing) training data containing all possible face poses. In addition, the learning is a more difficult task when there are large variations in the training set, and hence either a sufficiently complex regression model (functional form and number of features) is required, or the alignment method will compromise accuracy in order to align all these poses. As a general rule, increased model complexity leads to poorer generalization performance. This suggests that a simpler or more regularized model, which learns to align faces for a limited range of poses, would perform better for those poses than would a general alignment model that has been trained on all poses. As a simple example, consider a regression function that is trained using a single upright face image versus one trained using multiple in-plane rotations of that face image. In the former case, the regression function must have a root for the upright pose, whereas in the latter case, the regression function must have a root for every in-plane rotation.

Algorithm 1 Stage k of Transformation-Invariant SDM (TI-SDM)

Inputs: Prototype shape $\bar{\mathbf{x}}$, Regression $\{\mathbf{W}_k, \mathbf{b}_k\}$,
Image I, Initial landmark estimates $\mathbf{x}_k$

1: Use (6) to find transformation $\mathbf{A}_k$ that warps $\mathbf{x}_k$ to $\bar{\mathbf{x}}$
2: Warp to prototype coords: $I' = \mathbf{A}_k(I)$, $\mathbf{x}'_k = \mathbf{A}_k(\mathbf{x}_k)$
3: Extract features: $\boldsymbol{\phi}'_k = \boldsymbol{\phi}(I', \mathbf{x}'_k)$
4: Linear regression: $\mathbf{x}'_{k+1} = \mathbf{x}'_k + \mathbf{W}_k \boldsymbol{\phi}'_k + \mathbf{b}_k$
5: Warp back to image coords: $\mathbf{x}_{k+1} = \mathbf{A}_k^{-1}(\mathbf{x}'_{k+1})$
Output: Landmark locations $\mathbf{x}_{k+1}$

Our goal with transformation invariance is to train each regression on a smaller set of poses, while still being able to align faces in an arbitrary pose. To do so, we apply a transformation invariance step prior to each stage's regression function. We first construct a prototype shape, $\bar{\mathbf{x}}$, which contains the mean location of each landmark point across all of the training data (after uniform scaling and translation transformations have been applied to each training image to make them all share a canonical location and scale).

In this paper, we choose affine transformations as our class of transformations for invariance, although one could also use our method with a different class of transformations. At each stage k of regression, we find the affine transformation $\mathbf{A}_k$ that transforms the landmark locations $\mathbf{x}_k$ that were estimated by the previous stage of regression so as to minimize their sum of squared distances to the prototype landmark locations, $\bar{\mathbf{x}}$:

$$\mathbf{A}_k = \arg\min_{\mathbf{A} \in \mathcal{A}} \|\mathbf{A}(\mathbf{x}_k) - \bar{\mathbf{x}}\|^2, \tag{6}$$

where $\mathcal{A}$ denotes the set of all affine transformations. Next, we use the transformation $\mathbf{A}_k$ to warp the input image I and the landmark locations into the prototype coordinate frame: $I' = \mathbf{A}_k(I)$, and $\mathbf{x}'_k = \mathbf{A}_k(\mathbf{x}_k)$. Note that we slightly abuse notation here by using the same affine transformation operator $\mathbf{A}_k$ to both transform a vector of landmark locations, $\mathbf{A}_k(\mathbf{x}_k)$, and warp an image, $\mathbf{A}_k(I)$. The regression is then performed in the prototype coordinate frame:

$$\mathbf{x}'_{k+1} = \mathbf{x}'_k + \mathbf{W}_k \boldsymbol{\phi}(I', \mathbf{x}'_k) + \mathbf{b}_k. \tag{7}$$

The estimated landmark locations in image coordinates are given by the inverse transformation, $\mathbf{x}_{k+1} = \mathbf{A}_k^{-1}(\mathbf{x}'_{k+1})$.

The resulting algorithm, which we call Transformation-Invariant SDM (TI-SDM), consists of K stages of alignment and regression, illustrated in Fig. 2. Algorithm 1 summarizes what happens at each stage of TI-SDM.

3.2   Learning deformation constraints

One of the problems associated with using SDM for tracking landmark locations is that it puts no explicit constraint on the regression behavior of neighboring points, which makes it possible for the points to drift apart. This would be a straightforward problem to deal with in an optimization setting by introducing
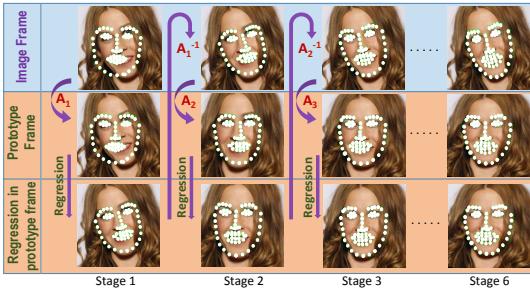
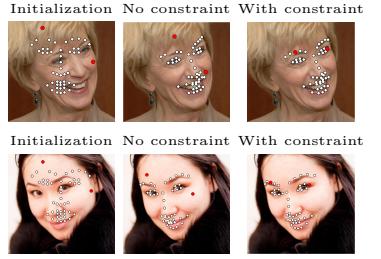Fig. 2: The Transformation-Invariant SDM (TI-SDM) algorithm. See Section 3.1 and Algorithm 1 for details.

Fig. 3: Effect of initial outliers (red) on results *without* vs. *with* deformation constraint.

explicit constraints or penalties on the free-form deformation of the landmark points. However, rather than utilizing an optimization procedure, which can be slow, we want to maintain the speed advantages of forward prediction using a regression function. To achieve the effect of constraints within a regression framework, we introduce additional features that allow the regression model to learn to constrain landmark points from drifting.

We introduce a soft constraint in the form of an additional cost term $||\mathbf{x} - \bar{\mathbf{x}}||^2$ in equation (1):

$$f_c(\mathbf{x}_k + \Delta \mathbf{x}) = \left\| \phi(I, \mathbf{x}_k + \Delta \mathbf{x}) - \hat{\phi} \right\|^2 + \lambda \left\| \mathbf{x}_k + \Delta \mathbf{x} - \bar{\mathbf{x}} \right\|^2. \qquad (8)$$

This enforces a quadratic penalty when the landmark locations drift away from the prototype shape $\bar{\mathbf{x}}$. The weight λ controls the tradeoff between data and the constraint. The Newton step for this constrained f is given by

$$\Delta \mathbf{x} = -2\mathbf{H}^{-1} \left(\mathbf{J}_\phi \ \mathbf{I} \right) \begin{pmatrix} \phi_k - \hat{\phi} \\ \lambda \left(\mathbf{x}_k - \bar{\mathbf{x}} \right) \end{pmatrix}, \qquad (9)$$

where $\mathbf{H}$ is the Hessian matrix of f_c with respect to $\mathbf{x}$, and $\mathbf{J}_\phi$ is the Jacobian of ϕ with respect to $\mathbf{x}$. Just as we approximated (2) by (3), we can approximate this constrained Newton step (9) by a linear regression function of a constrained feature vector, ϕ_k^*:

$$\Delta \mathbf{x} = \mathbf{W}_k \phi_k^* + \mathbf{b}_k, \quad \text{where} \quad \phi_k^* = \begin{pmatrix} \phi_k \\ \lambda \left(\mathbf{x}_k - \bar{\mathbf{x}} \right) \end{pmatrix}. \qquad (10)$$

As in unconstrained SDM, we can learn the regression coefficients $\mathbf{W}_k$ and bias $\mathbf{b}_k$ using training data. The only difference between the constrained (10) and unconstrained (3) regression models is that in the constrained version, we extend the feature vector to include additional features, $\lambda \left(\mathbf{x}_k - \bar{\mathbf{x}} \right)$, encoding the deviation of the landmark locations from the prototype landmark locations. In general,

during our experiments, the constrained regression learns to move landmark locations towards the mean shape by learning negative values for the associated regression coefficients. The learned coefficients' norms are larger for the initial regression stage of the cascade, but smaller in the later stages, which enforces weaker constraints on deformation as the landmark locations approach convergence. Note that it would be possible to incorporate λ into $\mathbf{W}_k$ and $\bar{\mathbf{x}}$ into $\mathbf{b}_k$, and just expand the feature vector $\boldsymbol{\phi}^*$ with $\mathbf{x}_k$ rather than $\lambda(\mathbf{x}_k - \bar{\mathbf{x}})$. However, we choose to keep the difference vector form as in (10), which becomes important for the regularized training described in Section 3.4.

To unify notation, in the rest of this paper we will refer to the expanded feature vector $\boldsymbol{\phi}^*$ as simply $\boldsymbol{\phi}$. That way, Equations (3–7) and Algorithm 1 apply to the constrained model without modification. In Fig. 3, we analyze the effect of the deformation constraint. See Section 4 for details.

3.3   Mixture-of-experts regression

The transformation invariance step described in Section 3.1 lets our model learn regression functions that are invariant to affine transformations of the faces. Still, the remaining variations in the data (e.g., due to out-of-plane rotations and facial expressions) are large enough that it is challenging for a single regression function to accurately align all faces. In particular, the training set in our experiments includes many more frontal faces with mild facial expressions than faces with large out-of-plane rotations or extreme expressions. Thus, the prototype (mean) face is close to a frontal face with neutral expression, and the regression function tends to work less well for more extreme poses and expressions.

We propose to use a mixture-of-experts regression model, in which each expert is a regression function that is specialized for a different subset of the possible poses and expressions. Each expert's subset is determined by the expert's prototype shape. We construct L prototype shapes, $\{\bar{\mathbf{x}}^l\}_{l=1,\ldots,L}$, such that the set of ground-truth landmark locations $\hat{\mathbf{x}}_n$ of each of the N faces in the dataset is well aligned with one of the prototype shapes. We write the determination of the prototype shapes as an optimization problem:

$$\{\bar{\mathbf{x}}^l\}_{l=1,\ldots,L} = \operatorname*{arg\,min}_{\{\dot{\mathbf{x}}^l\}_{l=1,\ldots,L}} \sum_{n=1}^{N} \min_{\substack{\mathbf{A}\in\mathcal{A}, \\ l\in\{1,\ldots,L\}}} \left\|\mathbf{A}(\hat{\mathbf{x}}_n) - \dot{\mathbf{x}}^l\right\|^2, \qquad (11)$$

where each $\dot{\mathbf{x}}^l$ is a $2p\times1$ vector representing a possible prototype face shape (i.e., the locations of p landmarks). If the class of transformations, $\mathcal{A}$, only contains the identity transformation, then this problem reduces to Euclidean clustering of training samples based on landmark locations (see Fig. 1a).

When $\mathcal{A}$ is the class of affine transformations, we call this affine-invariant clustering. In this case, (11) is a homogenous optimization problem in which additional constraints on the prototype shapes or the transformations are necessary to avoid the zero solution (which assigns zero to all of the transformations and prototype shapes). Moreover, the objective function is non-convex due to

the joint optimization of the shapes and the assignment of training samples to shapes. We decouple this problem into two convex sub-problems, which we solve iteratively. The first sub-problem assigns every training face image n to one of the prototype shapes via the equation

$$l_n = \arg\min_l \left[\min_{\mathbf{A} \in \mathcal{A}} \left\| \mathbf{A}(\hat{\mathbf{x}}_n) - \bar{\mathbf{x}}^l \right\|^2 \right] \tag{12}$$

assuming that the prototype shapes $\bar{\mathbf{x}}^l$ are fixed. This problem can be solved independently for each training face: The optimal assignment is the prototype to which the face's ground-truth landmark locations can be affine-aligned with minimum alignment error. The second sub-problem solves for the prototype shapes. Each prototype shape consists of the landmark locations that minimize the sum of the squared affine alignment errors of the ground-truth locations $\hat{\mathbf{x}}_n$ of the training faces that were assigned to that prototype shape:

$$\bar{\mathbf{x}}^l = \arg\min_{\dot{\mathbf{x}}^l} \sum_{n \text{ s.t. } l_n = l} \min_{\mathbf{A} \in \mathcal{A}} \left\| \mathbf{A}(\hat{\mathbf{x}}_n) - \dot{\mathbf{x}}^l \right\|^2 \quad s.t. \quad \mathbf{C}\dot{\mathbf{x}}^l = \mathbf{m}, \tag{13}$$

where to avoid degeneracy, the matrix $\mathbf{C}$ and vector $\mathbf{m}$ impose linear constraints on the prototype shape such that the mean location of the 5 landmark points of the left eyebrow is fixed, as are the mean location of the 5 right eyebrow points and the mean vertical location of the 16 mouth points. This optimization problem is quadratic with linear constraints, and the optimal solution is computed by solving a linear system. The two optimization sub-problems are alternately solved until the assignments do not change. In our experiments, 20–30 iterations suffice for convergence.

In Fig. 1 (right), we compare Euclidean clustering (a) with the proposed affine-invariant clustering (b). Euclidean clustering only accounts for the pose variations in the dataset. However, some of the out-of-plane poses can be approximately aligned to each other with an affine alignment, enabling the affine-invariant clustering to account for variations in both pose and facial expressions.

Each expert E^l corresponds to one of the L prototype shapes. At each stage of the regression cascade, we learn a separate regression for each expert. Hence, in addition to its prototype shape $\{\bar{\mathbf{x}}^l\}$, each regression expert E^l has a regression function $\{\mathbf{W}_k^l, \mathbf{b}_k^l\}$ for each of the K levels of the cascade:

$$E^l = \left\{ \bar{\mathbf{x}}^l, \; \left\{ \mathbf{W}_k^l, \mathbf{b}_k^l \right\}_{k=1,\ldots,K} \right\}. \tag{14}$$

At each stage, k, of the cascade, each expert E^l performs Algorithm 1 using prototype $\bar{\mathbf{x}}^l$ and regression function $\{\mathbf{W}_k^l, \mathbf{b}_k^l\}$:

$$\mathbf{x}_{k+1}^l = \text{Algorithm 1}\left(\bar{\mathbf{x}}^l, \left\{ \mathbf{W}_k^l, \mathbf{b}_k^l \right\}, I, \mathbf{x}_k \right). \tag{15}$$

The gating function for each regression expert E^l is a soft assignment $\alpha^l(\mathbf{x}_k)$ given by the softmax transformation of the transformation invariance error $\epsilon^l(\mathbf{x}_k)$

Algorithm 2 Mixture of Invariant Experts (MIX)

Inputs: Image I, Initial landmark estimates $\mathbf{x}_1$,
Experts $E^l = \left\{ \bar{\mathbf{x}}^l, \left\{ \mathbf{W}_k^l, \mathbf{b}_k^l \right\}_{k=1,\ldots,K} \right\}_{l=1,\ldots,L}$

1: **for** $k = 1$ to K **do**
2:    **for** $l = 1$ to L **do**
3:       Compute soft assignment $\alpha^l(\mathbf{x}_k)$ using (16)
4:       Apply one stage of TI-SDM:
          $\mathbf{x}_{k+1}^l = \text{Algorithm 1}\left(\bar{\mathbf{x}}^l, \left\{ \mathbf{W}_k^l, \mathbf{b}_k^l \right\}, I, \mathbf{x}_k \right)$
5:    **end for**
6:    Average over L experts:
       $\mathbf{x}_{k+1} = \sum_{l=1}^{L} \alpha^l(\mathbf{x}_k) \mathbf{x}_{k+1}^l$
7: **end for**
Output: Final landmark locations $\mathbf{x}_{K+1}$

---

between the starting landmark locations $\mathbf{x}_k$ and each prototype shape $\bar{\mathbf{x}}^l$. The soft assignments are computed using

$$\alpha^l(\mathbf{x}) = \frac{e^{-\epsilon^l(\mathbf{x})}}{\sum_{l=1}^{L} e^{-\epsilon^l(\mathbf{x})}}, \quad \text{where} \quad \epsilon^l(\mathbf{x}) = \min_{\mathbf{A} \in \mathcal{A}} \left\| \mathbf{A}(\mathbf{x}) - \bar{\mathbf{x}}^l \right\|^2. \quad (16)$$

Here, as in (6), $\mathcal{A}$ denotes the set of all affine transformations. A high score $\alpha^l(\mathbf{x}_k)$ indicates that the current estimate $\mathbf{x}_k$ is close to the prototype shape of the lth expert, and hence the regression results obtained from E^l would be given a high weight. In Fig. 1 (left), we show the assignment weights of two faces to experts in the model.

At each stage, k, of the cascade, our alignment algorithm applies every expert's regression function to the starting estimate of landmark locations $\mathbf{x}_k$, then averages the outputs according to the gating function $\alpha^l(\mathbf{x}_k)$ to obtain the updated estimate of landmark locations, $\mathbf{x}_{k+1}$:

$$\mathbf{x}_{k+1} = \sum_{l=1}^{L} \alpha^l(\mathbf{x}_k) \mathbf{x}_{k+1}^l. \quad (17)$$

Algorithm 2 summarizes our alignment method, which we call **M**ixture of **I**nvariant **Ex**perts (MIX).

Note that our mixture-of-experts model is quite different from multi-view models [6,7,8,43], which explicitly model and predict the head pose (e.g., by learning a different deformable model for each of several specific pose ranges). In contrast, MIX is a discriminative mixture model that discovers a data-dependent partitioning of the shape space (see Fig. 1b) based on facial expressions and other affine-invariant shape variations (including affine-invariant variations due to pose), and learns a different optimization for each partition.

3.4   Training the experts model

To learn the regression experts E^l, the N face images in the training data are augmented by repeating every training image M times, each time perturbing the
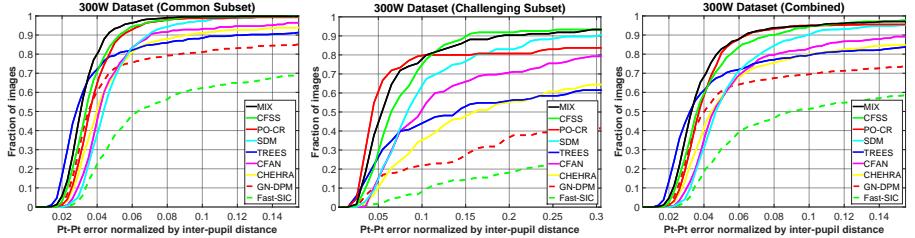
Fig. 4: Comparison of MIX with other state-of-the-art methods on the 300W dataset.
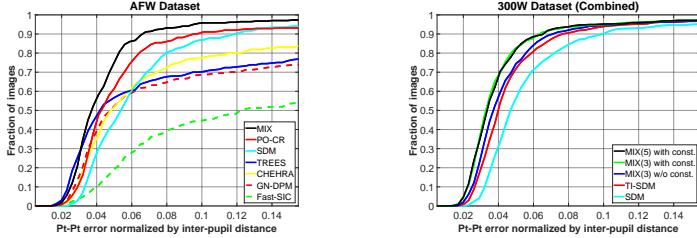

Fig. 5: *Left:* Comparison of MIX with other state-of-the-art methods on the AFW dataset. CFAN and CFSS are not compared, because both included AFW in their training set. *Right:* Comparing variations of proposed method on 300W (Combined).

ground-truth landmark locations by a different random displacement. For each image I_i in this augmented training set $(i = 1, \ldots, MN)$, with ground-truth landmark locations $\hat{\mathbf{x}}_i$, we displace the landmarks by a random displacement $\Delta\hat{\mathbf{x}}_i$. For every expert E^l, we use (16) to compute the soft assignment α_i^l of the ith sample's perturbed landmark locations to the prototype shape $\bar{\mathbf{x}}^l$:

$$\alpha_i^l = \alpha^l \left(\hat{\mathbf{x}}_i + \Delta\hat{\mathbf{x}}_i \right). \tag{18}$$

While computing this soft assignment, let $\mathbf{A}_i^l$ denote the global (affine) transformation from (16) that best aligns the ith sample's perturbed landmark locations to prototype shape $\bar{\mathbf{x}}^l$. Use $\mathbf{A}_i^l$ to transform the ground-truth landmark locations and displacement vectors into the prototype coordinate frame of expert E^l:

$$\hat{\mathbf{x}}_i^l = \mathbf{A}_i^l(\hat{\mathbf{x}}_i), \qquad \Delta\hat{\mathbf{x}}_i^l = \mathbf{A}_i^l(\Delta\hat{\mathbf{x}}_i). \tag{19}$$

The first regression function $(k = 1)$ is then learned by minimizing a Tikhonov regularized L2-loss function:

$$\{\mathbf{W}_k^l, \mathbf{b}_k^l\} = \underset{\mathbf{W}, \mathbf{b}}{\arg\min} \sum_{i=1}^{MN} \alpha_i^l \|\Delta\hat{\mathbf{x}}_i^l - \mathbf{W}\boldsymbol{\phi}(I_i, \hat{\mathbf{x}}_i^l - \Delta\hat{\mathbf{x}}_i^l) - \mathbf{b}\|^2 + \gamma \left[\|\mathbf{W}\|_F^2 + \|\mathbf{b}\|_F^2 \right].$$

$$\tag{20}$$

For each l and k, the regularizer weight γ is selected via grid search in log space using 2-fold cross validation.

Table 1: Numerical comparison of all tested methods on the 300W (Combined) dataset.

Method	$\text{NAUC}_{0.1}$	$\text{NAUC}_{0.2}$	$\text{NAUC}_{0.3}$	$\text{NAUC}_{0.4}$	$\text{NAUC}_{0.5}$
MIX	**0.5945**	**0.7810**	**0.8482**	**0.8828**	**0.9043**
CFSS [35]	0.5528	0.7613	0.8354	0.8733	0.8967
PO-CR [26]	0.5610	0.7568	0.8242	0.8588	0.8798
SDM [17]	0.4475	0.6957	0.7880	0.8362	0.8662
TREES [24]	0.5187	0.6746	0.7406	0.7792	0.8063
CFAN [34]	0.4357	0.6594	0.7487	0.7985	0.8317
CHEHRA [18]	0.4376	0.6390	0.7217	0.7703	0.8038
GN-DPM [15]	0.4274	0.5796	0.6531	0.7000	0.7354
Fast-SIC [5]	0.2490	0.4122	0.4984	0.5644	0.6179

For training the later regressions $\{\mathbf{W}_k, \mathbf{b}_k\}_{k=2,\dots,K}$, rather than using a random perturbation, the target $\Delta\hat{\mathbf{x}}_i$ is the residual of the previous stages of the cascade. In training, the regression function may diverge for a few samples, producing large residuals. To avoid fitting these outliers, at each stage k, we remove 5% of the samples with the largest residuals from the training set. We choose the number of regression stages K by training until the cross-validation error cannot be reduced further.

The training samples are generated by randomly perturbing the ground-truth facial landmark locations along the major deformation directions of the training set, which are determined via principal component analysis. In addition, we apply random rotation, translation, and anisotropic scaling to the landmark locations, and add i.i.d. Gaussian noise. After learning the cascade model for this training set (usually $K = 3$–4 stages), we learn a second cascade model using a training set consisting of only small amount of i.i.d. Gaussian noise, and append this model to the original model. The second model has 1–2 stages and improves fine alignment.

During testing, the initial landmark locations for regression are given by the mean landmark locations from all of the training data, translated and scaled to fit the given face detector bounding box.

4   Experiments

In the first experiment, we compare our proposed algorithm (MIX) to eight state-of-the-art algorithms: CFAN [34], TREES [24], CFSS [35], SDM [17], CHEHRA [18], GN-DPM (using SIFT features) [15], Fast-SIC [5], and PO-CR [26]. We evaluate performance on the 300W [39,40] and AFW [41] datasets.

We train our MIX algorithm on the training sets of two standard datasets: LFPW [44] (811 training faces) and Helen [45] (2000 training faces). We augment the training data by horizontally flipping each image, yielding $N = 5,622$ training images. From each image, we sample $M = 15$ training initializations (see Section 3.4). We use 3 experts ($L = 3$), because using more ($L = 5$) did not significantly improve performance (see the second experiment, below). For SDM, we use our own implementation, trained on the same training data as MIX.

For the other seven methods, we use their authors' publicly available code. Note that the training set for our algorithm is a smaller subset of the training sets used by CFSS [35] and CFAN [34]. Both CFSS and CFAN include AFW (337 faces) in the training set, but we do not, opting instead to test on AFW.

We test all methods on 300W using the same test set as [35], which comprises the test sets of LFPW [44] (224 test faces) and Helen [45] (330 test faces)[3] as well as the IBUG dataset (135 test faces). For all test images, we used the bounding box initializations provided on the 300W website (face detector bounding boxes). To compute errors of results, for all datasets we used the ground-truth locations of 49 landmarks from [39,40]. As in [35], the 300W *common subset* contains the test samples from LFPW (224) and HELEN (330), the *challenging subset* is IBUG (135), and *combined* refers to all 689 test images. Fig. 4 plots the cumulative distribution of the fraction of images, as a function of error normalized by the inter-pupil distance.

Table 1 presents a numerical comparison of our MIX algorithm with the previous eight methods on the entire (combined) 300W test set. Rather than measuring mean error, which is extremely sensitive to outliers with large alignment error [46], we instead use a normalized variation of the AUC_α error metric proposed by [46]. The error metric we use, Normalized AUC_α ($NAUC_\alpha$), measures the area under each cumulative distribution curve (the curves in Fig. 4) up to a threshold normalized error value α, then divides by α (the maximum possible area for that threshold). The resulting $NAUC_\alpha$ error measure, indexed by α, is always between 0 and 1 (where 1 is a perfect score): $NAUC_\alpha = \frac{1}{\alpha} \int_0^\alpha f(e)de$, where e is the normalized error, $f(e)$ is the cumulative error distribution function, and α is the upper bound that is used to calculate the definite integral.

The results in Fig. 4 and Table 1 show that our method outperforms all of the other recent methods on 300W. Note that for the next best method, CFSS, we used the code provided by the authors (the more accurate, but slower, version described in [35]), which is not practical for real-time use: the CFSS code required 1.7 s per face on our machine.

The evaluation results on the AFW dataset (Fig. 5, left) show a similar trend, in which our algorithm outperforms the other methods. CFAN and CFSS are not compared on AFW, because both included AFW in their training set.

In the second experiment, we compare several variants of our algorithm and analyze the contribution of each of the novel components described in Section 3. The baseline algorithm for this experiment is SDM [17]. MIX(L) refers to our Mixture of Invariant Experts with L experts (Section 3.3), and *with* or *without const.* refers to whether or not we use our extended deformation-constraint features (Section 3.2). TI-SDM is our Transformation-Invariant SDM (Section 3.1), which could also be called MIX(1) w/out const. Fig. 5 (right) shows that each element of our algorithm improves its performance. Performance is significantly improved by adding transformation invariance (SDM → TI-SDM), by including the mixture of experts at each stage of the cascade (TI-SDM → MIX(3) w/out

---

[3] The CFAN [34] algorithm included the 330 test faces from Helen in its training data. Thus when testing CFAN, we had to omit these 330 faces from the 300W test set.
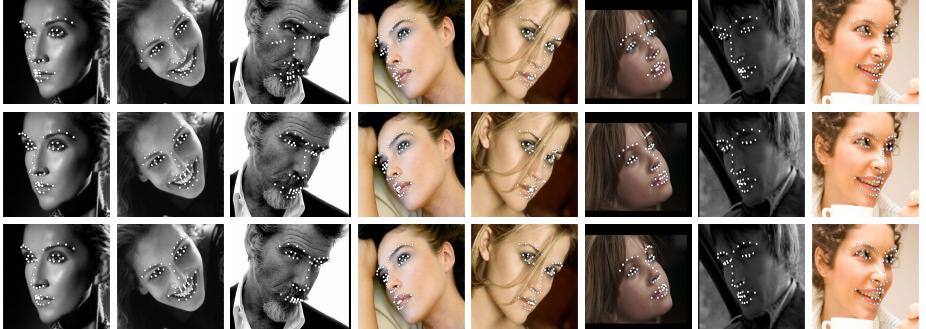
Fig. 6: Visual results on the challenging subset of 300W dataset. *First row:* SDM. *Second row:* TI-SDM. *Third row:* Our MIX algorithm. Transformation invariance (TI) significantly improves the accuracy of SDM. Improvement from single models (SDM and TI-SDM) to mixture models (MIX) is apparent particularly for large out-of-plane rotations and unusual facial expressions.

const.), and by using the extended deformation constraint features (MIX(3) w/out const. → MIX(3) with const.). Using mixtures of more than 3 regression experts (MIX(3) → MIX(5)) yields very minor improvement. This is because of the limited number of training images, particularly with extreme expressions or large out-of-plane rotations, which leads experts specializing in these less common face shapes to overfit the data (as we observed during cross-validation). In Fig. 6, we visually compare sample results on the challenging subset of 300W dataset. The improvement from a cascade of single models (SDM and TI-SDM, rows 1–2) to a cascade of mixture models (MIX, row 3) is greatest for large out-of-plane rotations and unusual facial expressions. As shown in Fig. 1, each expert specializes for particular poses and expressions, yielding more precise alignment.

In the third experiment, we illustrate the behavior of deformation constraint features by simulating a case in which a few points are poorly initialized or drift away during any regression stage. As shown in column 1 of Fig. 3, we initialize the alignment algorithm within the detection bounding box as usual, but to simulate drifting points we manually displace the two points shown in red (on the left eyebrow and on the outer corner of the right eye) to outside of the detection box. We then align using two models, one without deformation constraint (column 2) and the other with our extended deformation-constraint features (column 3). The model without deformation constraint fails to correct the outlier points, whereas the deformation constraint features move outlier points towards the prototype shape of the expert, enabling it to obtain the correct landmark locations.

On a single core of an Intel Core i5-6600 3.30GHz processor, MIX with 3 experts runs at 65 ms, of which SIFT feature computation takes 54 ms and the rest of the algorithm takes 11 ms. With multi-core implementation (3 experts run in parallel), run time is reduced to 30 ms (including SIFT feature computation).

5   Conclusion

We proposed a novel face alignment algorithm based on a cascade in which each stage consists of a mixture of transformation-invariant (e.g., affine-invariant) regression experts. Each expert specializes in a different part of the joint space of pose and expressions by (affine) transforming the landmark locations to its prototype shape and learning a customized regression model. We also present a method to include deformation constraints within the discriminative alignment framework. Extensive evaluation on benchmark datasets shows that the proposed method significantly improves upon the state of the art.

References

1. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. Computer vision and image understanding **61**(1) (1995) 38–59
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on pattern analysis and machine intelligence **23**(6) (2001) 681–685
3. Sauer, P., Cootes, T.F., Taylor, C.J.: Accurate regression procedures for active appearance models. In: BMVC. (2011) 1–11
4. Sung, J., Kim, D.: Adaptive active appearance model with incremental learning. Pattern recognition letters **30**(4) (2009) 359–367
5. Tzimiropoulos, G., Pantic, M.: Optimization problems for fast aam fitting in-the-wild. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). (2013)
6. Romdhani, S., Gong, S., Psarrou, A., et al.: A multi-view nonlinear active shape model using kernel pca. In: BMVC. Volume 10. (1999) 483–492
7. Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J.: View-based active appearance models. Image and Vision Computing **20**(9) (2002) 657–664
8. Asthana, A., Marks, T., Jones, M., Tieu, K., M.V., R.: Fully automatic pose-invariant face recognition via 3d pose normalization. In: IEEE International Conference on Computer Vision (ICCV). (November 2011) 937–944
9. Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J.: Coupled-view active appearance models. In: BMVC. (2000) 52–61
10. Hu, C., Xiao, J., Matthews, I., Baker, S., Cohn, J.F., Kanade, T.: Fitting a single active appearance model simultaneously to multiple images. In: BMVC. (2004) 1–10
11. Cristinacce, D., Cootes, T.F.: Feature detection and tracking with constrained local models. In: BMVC. (2006)
12. Cristinacce, D., Cootes, T.F.: Boosted regression active shape models. In: BMVC. (2007) 1–10
13. Zhou, F., Brandt, J., Lin, Z.: Exemplar-based graph matching for robust facial landmark localization. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 1025–1032
14. Smith, B.M., Zhang, L.: Joint face alignment with non-parametric shape models. In: Computer Vision–ECCV 2012. Springer (2012) 43–56
15. Tzimiropoulos, G., Pantic, M.: Gauss-newton deformable part models for face alignment in-the-wild. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 1851–1858

16. Tuzel, O., Porikli, F., Meer, P.: Learning on lie groups for invariant detection and tracking. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
17. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 532–539
18. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Incremental face alignment in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 1859–1866
19. Liu, X.: Discriminative face alignment. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(11) (2009) 1941–1954
20. Kazemi, V., Sullivan, J.: Face alignment with part-based modeling. In: Proceedings of the British Machine Vision Conference, BMVA Press (2011) 27–1
21. Saragih, J.M., Lucey, S., Cohn, J.F.: Deformable model fitting by regularized landmark mean-shift. International Journal of Computer Vision **91**(2) (2011) 200–215
22. Dollár, P., Welinder, P., Perona, P.: Cascaded pose regression. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 1078–1085
23. Ren, S., Cao, X., Wei, Y., Sun, J.: Face alignment at 3000 fps via regressing local binary features. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 1685–1692
24. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 1867–1874
25. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. International Journal of Computer Vision **107**(2) (2014) 177–190
26. Tzimiropoulos, G.: Project-out cascaded regression with an application to face alignment. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2015)
27. Yan, J., Lei, Z., Yi, D., Li, S.Z.: Learn to combine multiple hypotheses for accurate face alignment. In: Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, IEEE (2013) 392–396
28. Cootes, T.F., Ionita, M.C., Lindner, C., Sauer, P.: Robust and accurate shape model fitting using random forest regression voting. In: Computer Vision–ECCV 2012. Springer (2012) 278–291
29. Alabort-i Medina, J., Antonakos, E., Booth, J., Snape, P., Zafeiriou, S.: Menpo: A comprehensive platform for parametric image alignment and visual deformable models. In: Proceedings of the ACM International Conference on Multimedia. MM '14, New York, NY, USA, ACM (2014) 679–682
30. Xiong, X., De la Torre, F.: Global supervised descent method. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2015)
31. Xiao, S., Yan, S., Kassim, A.A.: Facial landmark detection via progressive initialization. In: The IEEE International Conference on Computer Vision (ICCV) Workshops. (December 2015)
32. Zhang, J., Kan, M., Shan, S., Chen, X.: Leveraging datasets with varying annotations for face alignment via deep regression network. In: The IEEE International Conference on Computer Vision (ICCV). (December 2015)
33. Zhou, E., Fan, H., Cao, Z., Jiang, Y., Yin, Q.: Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In: ICCV Workshop. (2013)

34. Zhang, J., Shan, S., Kan, M., Chen, X.: Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In: ECCV. (2014)
35. Zhu, S., Li, C., Change Loy, C., Tang, X.: Face alignment by coarse-to-fine shape searching. In: CVPR. (2015)
36. Burgos-Artizzu, X.P., Perona, P., Dollar, P.: Robust face landmark estimation under occlusion. In: The IEEE International Conference on Computer Vision (ICCV). (December 2013)
37. Yu, X., Lin, Z., Brandt, J., Metaxas, D.N.: Consensus of regression for occlusion-robust facial feature localization. In: Computer Vision–ECCV 2014. Springer (2014) 105–118
38. Rao, A., Miller, D., Rose, K., Gersho, A.: Mixture of experts regression modeling by deterministic annealing. Signal Processing, IEEE Transactions on **45**(11) (Nov 1997) 2811–2820
39. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: A semi-automatic methodology for facial landmark annotation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR-Workshops), 5th Workshop on Analysis and Modeling of Faces and Gestures (AMFG2013), Portland Oregon, USA (June 2013)
40. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: Proceedings of IEEE International Conference on Computer Vision (ICCV-Workshops), 300 Faces in-the-Wild Challenge (300-W), Sydney, Australia (December 2013)
41. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2879–2886
42. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2) (2004) 91–110
43. Čech, J., Franc, V., Uřičář, M., Matas, J.: Multi-view facial landmark detection by using a 3D shape model. Image and Vision Computing **47** (2016) 60–70
44. Belhumeur, P.N., Jacobs, D.W., Kriegman, D., Kumar, N.: Localizing parts of faces using a consensus of exemplars. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 545–552
45. Le, V., Brandt, J., Lin, Z., Bourdev, L., Huang, T.S.: Interactive facial feature localization. In: Computer Vision–ECCV 2012. Springer (2012) 679–692
46. Yang, H., Jia, X., Loy, C.C., Robinson, P.: An empirical study of recent face alignment methods. CoRR **abs/1511.05049** (2015)