# Learning Deep Representation for Face Alignment with Auxiliary Attributes

Zhanpeng Zhang, Ping Luo, Chen Change Loy, *Member, IEEE* and Xiaoou Tang, *Fellow, IEEE*

**Abstract**—In this study, we show that landmark detection or face alignment task is not a single and independent problem. Instead, its robustness can be greatly improved with auxiliary information. Specifically, we jointly optimize landmark detection together with the recognition of heterogeneous but subtly correlated facial attributes, such as gender, expression, and appearance attributes. This is non-trivial since different attribute inference tasks have different learning difficulties and convergence rates. To address this problem, we formulate a novel tasks-constrained deep model, which not only learns the inter-task correlation but also employs dynamic task coefficients to facilitate the optimization convergence when learning multiple complex tasks. Extensive evaluations show that the proposed task-constrained learning (i) outperforms existing face alignment methods, especially in dealing with faces with severe occlusion and pose variation, and (ii) reduces model complexity drastically compared to the state-of-the-art methods based on cascaded deep model.

1、关键点检测或者脸部矫正不是单一或者独立任务。可以使用辅助信息提升鲁棒性

**Index Terms**—Face Alignment, Face Landmark Detection, Deep Learning, Convolutional Network



## 1 Introduction

Face alignment, or detecting semantic facial landmarks (*e.g.*, eyes, nose, mouth corners) is a fundamental component in many face analysis tasks, such as facial attribute inference [1], face verification [2], and face recognition [3]. Though great strides have been made in this field (see Sec. 2), robust facial landmark detection remains a formidable challenge in the presence of partial occlusion and large head pose variations (Fig. 1).

Landmark detection is traditionally approached as a single and independent problem. Popular approaches include template fitting approaches [4], [5], [6], [7] and regression-based methods [8], [9], [10], [11], [12]. More recently, deep models have been applied too. For example, Sun *et al.* [13] propose to detect facial landmarks by coarse-to-fine regression using a cascade of deep convolutional neural networks (CNN). This method shows superior accuracy compared to previous methods [9], [14] and existing commercial systems. Nevertheless, the method requires a complex and unwieldy cascade architecture of deep model.

We believe that facial landmark detection is not a standalone problem, but its estimation can be influenced by a number of heterogeneous and subtly correlated factors. Changes on a face are often governed by the same rules determined by the intrinsic facial structure. For instance, when a kid is smiling, his mouth is widely opened (the second image in Fig. 1(a)). Effectively discovering and exploiting such an intrinsically correlated facial attribute would help in detecting the mouth corners more accurately. Also, the inter-ocular distance is smaller in faces with large yaw rotation (the first image in Fig. 1(a)). Such pose information can be leveraged as an additional source of information to constrain the solution space of landmark estimation. Indeed, the input and solution spaces of face alignment can be effectively divided given auxiliary face attributes. In a small experiment, we average a set of face images according to different attributes, as shown in Fig. 1(b)), where the frontal and smiling faces show the mouth corners, while there are no specific details for the image averaged over the whole dataset. Given the rich auxiliary information, treating facial landmark detection in isolation is counterproductive.

This study aims to investigate the possibility of optimizing facial landmark detection (the main task) by leveraging auxiliary information from attribute inference tasks. Potential auxiliary tasks include head pose estimation, gender classification, age estimation [15], facial expression recognition, or facial attribute inference [16]. Given the multiple tasks, deep convolutional network appears to be a viable model choice since it allows for joint features learning and multi-objective inference. Typically, one can formulate a cost function that encompasses all the tasks and use the cost function in the network back-propagation learning. We show that this conventional multi-task learning scheme is challenging in our problem. There are several reasons. First, the different tasks of face alignment and attribute inference are inherently different in learning difficulties. For instance, learning to identify "wearing glasses" attribute is easier than determining if one is smiling. Second, we rarely have auxiliary tasks with similar number of positive/negative cases. For instance, male/female classification enjoys more balanced samples than facial expression recognition. As a result, different tasks have different convergence

• *The authors are with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong.*
*E-mail: {zz013, pluo, ccloy, xtang}@ie.cuhk.edu.hk*

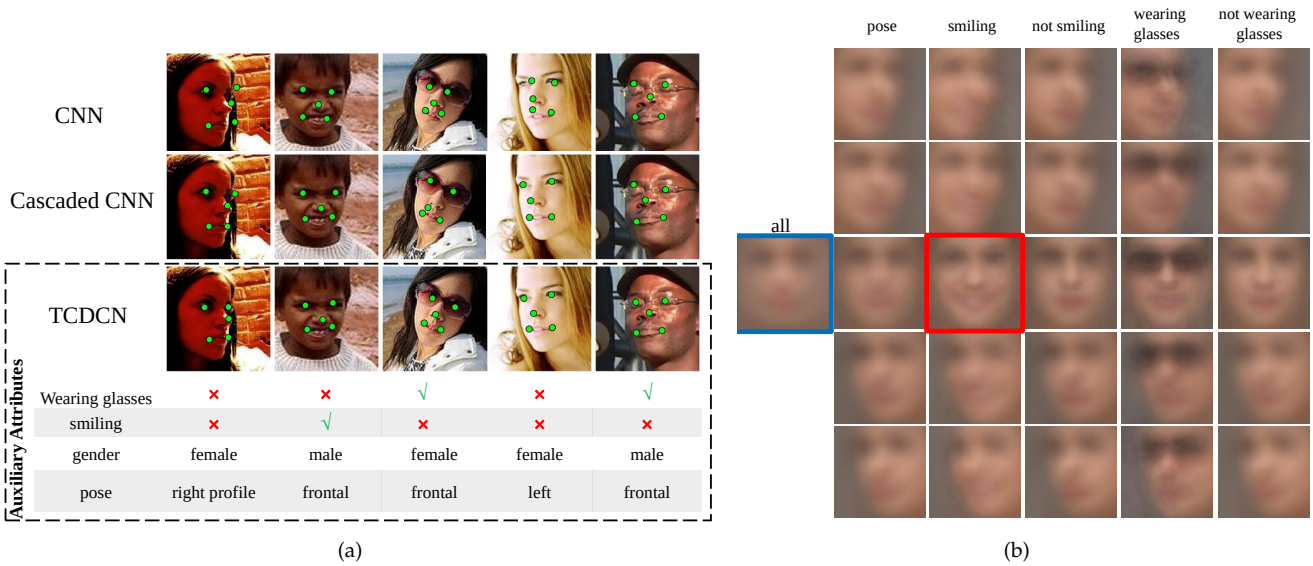



Fig. 1. (a) Examples of facial landmark detection by a single conventional CNN, the cascaded CNN [13], and the proposed Tasks-Constrained Deep Convolutional Network (TCDCN). More accurate detection can be achieved by optimizing the detection task jointly with related/auxiliary tasks. (b) Average face images with different attributes. The image in blue rectangle is averaged among the whole training faces, while the one in red is from the smiling faces with frontal pose. It indicates that the input and solution space can be effectively divided into subsets, which are in different distributions. This lowers the learning difficulty.

rates. In many cases we observe that the joint learning with a specific auxiliary task improves the convergence of landmark detection at the beginning of the training procedure, but become ineffective when the auxiliary task training encounters local minima or over-fitting. Continuing the training with all tasks jeopardizes the network convergence, leading to poor landmark detection performance.

Our study is the first attempt to demonstrate that face alignment can be jointly optimized with the inference of heterogeneous but subtly correlated auxiliary attributes. We show that the supervisory signal of auxiliary tasks can be back-propagated jointly with that of face alignment to learn the underlying regularities of face representation. Nonetheless, the learning is non-trivial due to the different natures and convergence rates of different tasks. Our key contribution is a newly proposed *Tasks-Constrained Deep Convolutional Network* (TCDCN), with new objective function to address the aforementioned challenges. In particular, our model considers the following aspects to make the learning effective:

- *Dynamic task coefficient* - Unlike existing multi-task deep models [17], [18], [19] that treat all tasks as equally important, we assign and weight each auxiliary task with a coefficient, which is adaptively and dynamically adjusted based on training and validation errors achieved so far in the learning process. Thus a task that is deemed not beneficial to the main task is prevented from contributing to the network learning. This approach can be seen as a principled way of achieving "early stopping" on specific task. In the experiments, we show that the dynamic task coefficient is essential to reach the peak performance for face alignment.
- *Inter-task correlation modeling* - We additionally model the relatedness of heterogeneous tasks in a covariance matrix in the objective function. Different from the dynamic task coefficient that concerns on the learning convergence, inter-task correlation modeling helps better exploiting the relation between tasks to achieve better feature learning.

All the network parameters, including the filters, dynamic task coefficients, and inter-task correlation are learned automatically using a newly proposed alternating optimization approach.

Thanks to the effective shared representation learned from multiple auxiliary attributes, the proposed approach outperforms other deep learning based approaches for face alignment, including the cascaded CNN model [13] on five facial point detection. We demonstrate that shared representation learned by a TCDCN for sparse landmarks can be readily transferred to handle an entirely different configuration with more landmarks, *e.g.* 68 points in the 300-W dataset [20]. With the transferred configuration, our method further outperforms other existing methods [5], [6], [8], [9], [12], [21], [22] on the challenging 300-W dataset, as well as the Helen [23] and COFW [8] dataset.

In comparison to our earlier version of this work [24], we introduce the new dynamic task coefficient

to generalize the original idea of task-wise early stopping [24] (discussed in Sec. 3.1). Specifically, we show that the dynamic task coefficient is a relatively more effective mechanism to facilitate the convergence of a heterogeneous task network. In addition, we formulate a new objective function that learns different tasks and their correlation jointly, which further improves the performance and allows us to analyze the usefulness of auxiliary tasks more comprehensively. Apart from the methodology, the paper was also substantially improved by providing more technical details and more extensive experimental evaluations.

## 2 Related Work

**Facial landmark detection**: Conventional facial landmark detection methods can be divided into two categories, namely regression-based method and template fitting method. A regression-based method estimates landmark locations explicitly by regression using image features. For example, Valstar *et al.* [25] predict landmark location from local image patch with support vector regression. Cao *et al.* [9] and Burgos-Artizzu *et al.* [8] employ cascaded fern regression with pixel-difference features. A number of studies [10], [11], [22], [26], [27], [28] use random regression forest to cast votes for landmark location based on local image patch with Haar-like features. Most of these methods refine an initial guess of the landmark location iteratively, the first guess/initialization is thus critical. By contrast, our deep model takes raw pixels as input without the need of any facial landmark initialization. Importantly, our method differs in that we exploit auxiliary tasks to facilitate landmark detection learning.

A template fitting method builds face templates to fit input images [4], [29], [30]. Part-based model has recently been used for face fitting [5], [6], [31]. Zhu and Ramanan [5] show that face detection, facial landmark detection, and pose estimation can be jointly addressed. Our method differs in that we do not limit the learning of specific tasks, *i.e.* the TCDCN is readily expandable to be trained with additional auxiliary tasks. Specifically, apart from pose, we show that other facial attributes such as gender and expression, can be useful for learning a robust landmark detector. Another difference to [5] is that we learn feature representation from raw pixels rather than pre-defined HOG as face descriptor.

**Landmark detection by deep learning**: The methods [12], [13], [32] that use deep learning for face alignment are close to our approach. The methods usually formulate the face alignment as a regression problem and use multiple deep models to locate the landmarks in a coarse-to-fine manner, such as the cascaded CNN by Sun *et al.* [13]. The cascaded CNN requires a pre-partition of faces into different parts, each of which are processed by separate deep CNNs. The resulting outputs are subsequently averaged and channeled to separate cascaded layers to process each facial landmark individually. Similarly, Zhang *et al.* [12] uses successive auto-encoder networks to perform coarse-to-fine alignment. Instead, our model requires neither pre-partition of faces nor cascaded networks, leading to drastic reduction in model complexity, whilst still achieving comparable or even better accuracy. This opens up possibility of application in computational constrained scenario, such as the embedded systems. In addition, the use of auxiliary task can reduce the overfitting problem of deep model because the local minimum for different tasks might be in different places. Another important difference is that our method performs feature extraction in the whole face image automatically, instead of handcraft local regions.

**Learning multiple tasks in neural network**: Multitask learning (MTL) is the process of learning several tasks simultaneously with the aim of mutual benefit. This is an old idea in machine learning. Caruana [33] provides a good overview focusing on neural network. Deep model is well suited for learning multiple tasks since it allows for joint features learning and multi-objective inference. Joint learning of multiple tasks has also proven effective in many computer vision problems [17], [18], [19]. However, existing deep models [17], [34] are not suitable to solve our problem because they assume similar learning difficulties and convergence rates across all tasks. For example, in the work of [19], the algorithm simultaneously learns a human pose regressor and multiple body-part detectors. This algorithm optimizes multiple tasks directly without learning the task correlation. In addition, it uses pre-defined task coefficients in the iterative learning process. Applying this method on our problem leads to difficulty in learning convergence, as shown in Sec. 4. We mitigate this shortcoming by introducing dynamic task coefficients in the deep model. This new formulation generalizes the idea of early stopping. Early stopping of neural network can date back to the work of Caruana [33], but it is heuristic and limited to shallow multilayer perceptrons. The scheme is also not scalable for a large quantity of tasks. Different from the work of [35], which learns the task priority to handle outlier tasks, the dynamic task coefficient in our approach is based on the training and validation error, and aims to coordinate tasks of different convergence rates. We show that dynamic task coefficient is important for joint learning multiple objectives in deep convolutional network.

## 3 Landmark Detection with Auxiliary Attributes

### 3.1 Overview

We cast facial landmark detection as a nonlinear transformation problem, which transforms the raw pixels of a face image to the positions of dense landmarks. The proposed framework is illustrated in Fig. 2, showing that the highly nonlinear function is modeled

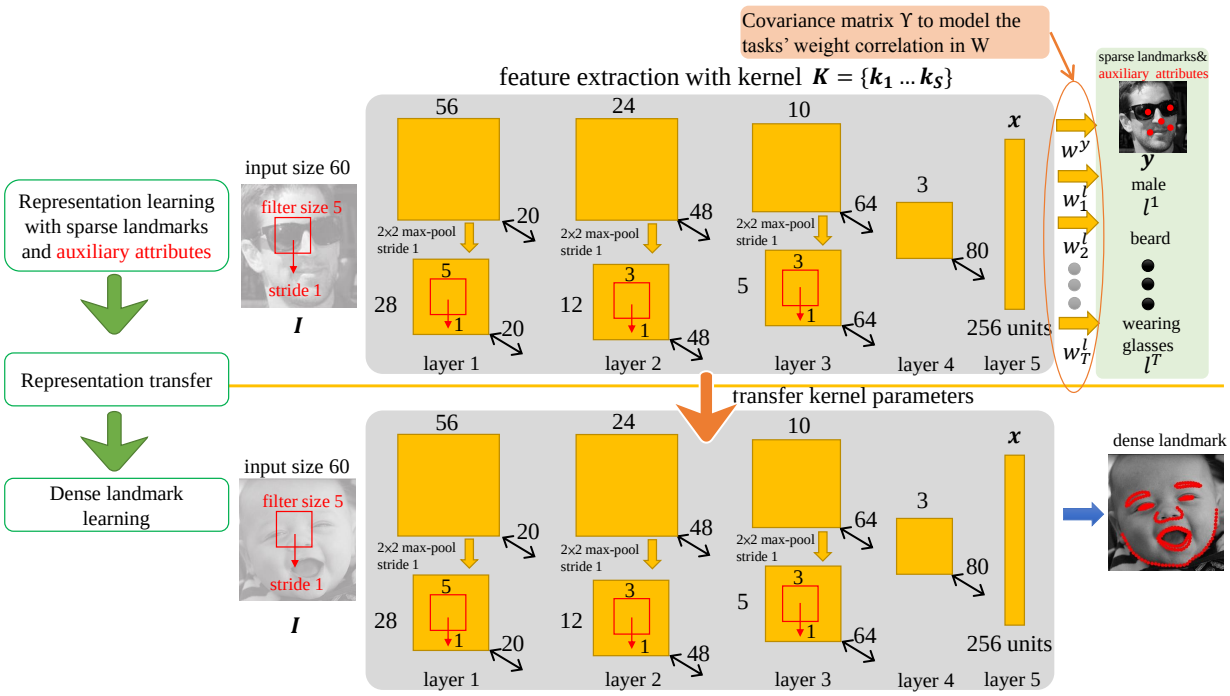



Fig. 2. Structure specification for TCDCN. A $60 \times 60$ image is taken as input. In the first layer, we convolve it with 20 different $5 \times 5$ filters, using a stride of 1. The obtained feature map is $56 \times 56 \times 20$, which is subsampled to $28 \times 28 \times 20$ with a $2 \times 2$ max-pooling operation. Similar operations are repeated in layer 2, 3, 4, as the parameters shown in the figure. The last layer is fully-connected. Then the output is obtained by regression.

as a DCN, which is pre-trained by five landmarks and then fine-tuned to predict the dense landmarks. Since dense landmarks are expensive to label, the pre-training step is essential because it prevents DCN from over-fitting to small dataset. In general, the pre-training and fine-tuning procedures are similar, except that the former step initializes filters by a standard normal distribution, while the latter step initializes filters using the pre-trained network.

As shown in Fig. 2, DCN extracts a high-level representation $\mathbf{x} \in \mathbb{R}^{D\times 1}$ on a face image $I$ using a set of filters $\mathbf{K} = \{\mathbf{k}_s\}_{s=1}^{S}$, $\mathbf{x} = \phi(I|\mathbf{K})$, where $\phi(\cdot)$ is the nonlinear transformation learned by DCN. With the extracted feature $\mathbf{x}$, we jointly estimate landmarks and attributes, where landmark detection is the main task and attribute prediction is the auxiliary task. Let $\{y^m\}_{m=1}^{M}$ denote a set of real values, representing the x-,y-coordinate of the landmarks, and let $\{l^t\}_{t=1}^{T}$ denote a set of binary labels of the face attributes, $\forall l^t \in \{0, 1\}$. Specifically, $M$ equals $5 \times 2 = 10$ in the pre-training step, implying that the landmarks include two centers of the eyes, nose, and two corners of the mouth. $M$ represents the number of dense landmarks in the fine-tuning step, such as $M = 194 \times 2 = 388$ in Helen dataset [23] and $M = 68 \times 2 = 136$ in 300-W dataset [20]. This work investigates the effectiveness of 22 attributes in landmark detection, *i.e.* $T = 22$.

Both landmark detection and attribute prediction can be learned by the generalized linear models [36]. Suppose $\mathbf{W} = [\mathbf{w}_1^y, \mathbf{w}_2^y, ..., \mathbf{w}_M^y, \mathbf{w}_1^l, \mathbf{w}_2^l, ..., \mathbf{w}_T^l]$ be a weight matrix, $\mathbf{W} \in \mathbb{R}^{D\times(M+T)}$, where each column vector corresponds to the parameters of a single task. For example, $\mathbf{w}_2^y \in \mathbb{R}^{D\times 1}$ indicates the parameter vector for the y-coordinate of the first landmark. With these parameters, we have

$$y^m = \mathbf{w}_m^{y\,\mathsf{T}}\mathbf{x} + \epsilon_m^y, \tag{1}$$

where $\epsilon_m^y$ represents an additive random error variable that is distributed according to a normal distribution with mean zero and variance $\sigma_m^2$, *i.e.* $\epsilon_m^y \sim \mathcal{N}(0, \sigma_m^2)$. Similarly, each $\mathbf{w}_t^l$ represents the parameter vector of the $t$-th attribute, which is model as

$$l^t = \mathbf{w}_t^{l\,\mathsf{T}}\mathbf{x} + \epsilon_t^l, \tag{2}$$

where $\epsilon_t^l$ is distributed following a standard logistic distribution, *i.e.* $\epsilon_t^l \sim Logistic(0, 1)$.

If all the tasks are independent, $\mathbf{W}$ can be simply modeled as a product of the multivariate normal distribution, *i.e.* $\forall \mathbf{w}_m^y, \mathbf{w}_t^l \sim \mathcal{N}(\mathbf{0}, \varepsilon^2\mathcal{I})$, where $\mathbf{0}$ is a $D \times 1$ zero vector, $\mathcal{I}$ denote a $D \times D$ identity matrix, and $\varepsilon^2$ is a $D \times 1$ vector representing the diagonal elements of the covariance matrix. However, this work needs to explore which auxiliary attribute is crucial to landmark detection, implying that we need to model the correlations between tasks. Therefore, we assume $\mathbf{W}$ is distributed according to a matrix normal distribution [37], *i.e.* $\mathbf{W} \sim \mathcal{MN}_{D\times(M+T)}(\mathbf{0}, \Upsilon, \varepsilon^2\mathcal{I})$, where $\mathbf{0}$ is a $D \times (M + T)$ zero matrix and $\Upsilon$ is a $(M + T) \times (M + T)$

task covariance matrix. The matrix $\Upsilon$ is learned in the training process and can naturally capture the correlation between the weight of different tasks.

As landmark detection and attribute prediction are heterogenous tasks, different auxiliary attribute behaves differently in the training procedure. They may improve the convergence of landmark detection at the beginning of the training procedure, but may become ineffective as training proceeds when local minima or over-fitting is presented. Thus, each auxiliary attribute is assigned with a dynamic task coefficient $\lambda_t$, $t = 1...T$, which is adjusted adaptively during training. $\lambda_t$ is distributed according to a normal distribution with mean $\mu_t$ and variance $\sigma_t^2$, *i.e.* $\lambda_t \sim \mathcal{N}(\mu_t, \sigma_t^2)$, where we assume $\sigma_t^2 = 1$ and $\mu_t$ is determined based on the training and validation errors (detailed in Sec. 3.3).

It is worth pointing out that in in the early version of this work [24], we introduce a task-wise early stopping scheme to halt a task after it is no longer beneficial to the main task. This method is heuristic and the criterion to determine when to stop learning a task is empirical. In addition, once a task is halted, it will never resume during the training process. In contrast to this earlier proposal, the dynamic task coefficient is dynamically updated. Thus a halted task may be resumed automatically if it is found useful again during the learning process. In particular, the dynamic task coefficient has no single optimal solution across the whole learning process. Instead, its value is updated to fit the current training status.

In summary, given a set of face images and their labels, we jointly estimate the filters $\mathbf{K}$, the weight matrix $\mathbf{W}$, the task covariance matrix $\Upsilon$, and the dynamic coefficients $\Lambda = \{\lambda_t\}_{t=1}^T$.

### 3.2 Problem Formulation

The above problem can be formulated as a probabilistic framework. Given a data set with $N$ training samples, denoted as $\{\mathbf{I}, \mathbf{Y}, \mathbf{L}\}$, where $\mathbf{I} = \{I_i\}_{i=1}^N$, $\mathbf{Y} = \{\{y_i^m\}_{m=1}^M\}_{i=1}^N$, and $\mathbf{L} = \{\{l_i^t\}_{t=1}^T\}_{i=1}^N$, and a set of parameters $\{\mathbf{K}, \mathbf{W}, \Upsilon, \Lambda\}$, we optimize the parameters by maximizing a posteriori probability (MAP)

$$\mathbf{K}^*, \mathbf{W}^*, \Upsilon^*, \Lambda^* = \underset{\mathbf{K},\mathbf{W},\Upsilon,\Lambda}{\operatorname{argmax}}\, p(\mathbf{K}, \mathbf{W}, \Upsilon, \Lambda | \mathbf{I}, \mathbf{Y}, \mathbf{L}). \quad (3)$$

Eqn.(3) is proportional to

$$p(\mathbf{K}, \mathbf{W}, \Upsilon, \Lambda | \mathbf{I}, \mathbf{Y}, \mathbf{L}) \propto p(\mathbf{Y}|\mathbf{I}, \mathbf{K}, \mathbf{W}_M) p(\mathbf{L}|\mathbf{I}, \mathbf{K}, \mathbf{W}_T, \Lambda) \cdot p(\mathbf{W}|\Upsilon) p(\Lambda) p(\mathbf{K}), \quad (4)$$

where the first two terms are the likelihood probabilities and the last three terms are the prior probabilities. Moreover, $\mathbf{W}_M$ and $\mathbf{W}_T$ represent the first $M$ columns and the last $T$ columns of $\mathbf{W}$, respectively. In the following, we will introduce each term of Eqn.(4) in detail.

• The likelihood probability $p(\mathbf{Y}|\mathbf{I}, \mathbf{K}, \mathbf{W}_M)$ measures the accuracy of landmark detection. As discussed in Eqn.(1), each variable of landmark position can be modeled as a linear regression plus a Gaussian noise. The likelihood can be factorized as

$$p(\mathbf{Y}|\mathbf{I}, \mathbf{K}, \mathbf{W}_M) = \prod_{i=1}^N \prod_{m=1}^M \mathcal{N}(\mathbf{w}_m^{y\,\mathsf{T}} \mathbf{x}_i, \sigma_m^2). \quad (5)$$

• The likelihood probability $p(\mathbf{L}|\mathbf{I}, \mathbf{K}, \mathbf{W}_T, \Lambda)$ measures the accuracy of attribute prediction. As introduced in Eqn.(2), each binary attribute is predicted by a linear function plus a logistic distributed random noise, implying that the probability of $l_i^t$ is a sigmoid function, which is $p(l_i^t = 1|\mathbf{x}_i) = f(\mathbf{w}_t^{l\,\mathsf{T}} \mathbf{x}_i)$, where $f(x) = 1/(1 + \exp\{-x\})$. Thus, the likelihood can be defined as product of Bernoulli distributions

$$p(\mathbf{L}|\mathbf{I}, \mathbf{K}, \mathbf{W}_T, \Lambda) \propto \prod_{i=1}^N \prod_{t=1}^T \{p(l_i^t = 1|\mathbf{x}_i)^{l_i^t} \big(1 - p(l_i^t = 1|\mathbf{x}_i)\big)^{1-l_i^t}\}^{\lambda_t}. \quad (6)$$

• The prior probability of the weight matrix, $p(\mathbf{W}|\Upsilon)$, is modeled by a matrix normal distribution with mean zero [37], which is able to capture the correlations between landmark detection and auxiliary attributes. It is written as

$$p(\mathbf{W}|\Upsilon) = \frac{\exp\big\{-\frac{1}{2}\mathrm{tr}[(\varepsilon^2\mathcal{I})^{-1}\mathbf{W}\Upsilon^{-1}\mathbf{W}^{\mathsf{T}}]\big\}}{(2\pi)^{\frac{D(M+T)}{2}} |\varepsilon^2\mathcal{I}|^{\frac{M+T}{2}} |\Upsilon|^{\frac{D}{2}}}, \quad (7)$$

where $\mathrm{tr}(\cdot)$ calculates the trace of a matrix and $\Upsilon$ is a positive semi-definite matrix modeling the task covariance, denoted as $\Upsilon \succeq \mathbf{0}$, $\Upsilon \in \mathbb{R}^{(M+T)\times(M+T)}$. Referring to Eqn.(7), the variance between the $m$-th landmark and the $t$-th attribute is obtained by $\sum_{d=1}^D \mathbf{W}_{(d,m)} \Upsilon_{(m,m+t)}^{-1} \mathbf{W}_{(d,m+t)}$, where $\mathbf{W}_{(d,m)}$ denotes the element in the $d$-th row and $m$-th column, showing that the relation of a pair of tasks is measured by their corresponding weights with respect to each feature dimension $d$. For instance, if two different tasks select or reject the same set of features, they are highly correlated. More clearly, Eqn.(7) is a matrix form of the multivariate normal distribution. They are equivalent if $\mathbf{W}$ is reshaped as a long vector.

• The prior probability of the tasks' dynamic coefficients is defined as a product of the normal distributions, $p(\Lambda) = \prod_{t=1}^T \mathcal{N}(\mu_t, \sigma_t^2)$, where the mean is adjustable based on the training and validation errors. It has significant difference with the task covariance matrix. For example, the auxiliary attribute 'wearing glasses' is probably related to the landmark positions of eyes. Their relation can be measured by $\Upsilon$. However, if 'wearing glasses' converges more quickly than the other tasks, it becomes ineffective because of local minima or over-fitting. Therefore, its dynamic coefficient could be decreased to avoid these side-effects.

• The DCN filters can be initialized as a standard multivariate normal distribution as previous methods [38] did. In particular, we define $p(\mathbf{K}) = \prod_{s=1}^S p(\mathbf{k}_s) = \prod_{s=1}^S \mathcal{N}(\mathbf{0}, \mathcal{I})$.

By taking the negative logarithm of Eqn.(4) and combining Eqn.(5), (6), and (7), we obtain the MAP objective function

$$\begin{aligned}&\underset{\mathbf{K},\mathbf{W},\Lambda,\Upsilon\succeq\mathbf{0}}{\operatorname{argmin}} \sum_{i=1}^{N}\sum_{m=1}^{M}(y_i^m - \mathbf{w}_m^{y\,\mathsf{T}}\mathbf{x}_i)^2 \\ &- \sum_{i=1}^{N}\sum_{t=1}^{T}\lambda_t\Big\{l_i^t \ln f(\mathbf{w}_t^{l\,\mathsf{T}}\mathbf{x}_i) + (1-l_i^t)\ln\big(1-f(\mathbf{w}_t^{l\,\mathsf{T}}\mathbf{x}_i)\big)\Big\} \\ &+ \operatorname{tr}(\mathbf{W}\Upsilon^{-1}\mathbf{W}^{\mathsf{T}}) + D\ln|\Upsilon| + \sum_{s=1}^{S}\mathbf{k}_s^{\mathsf{T}}\mathbf{k}_s + \sum_{t=1}^{T}(\lambda_t-\mu_t)^2.\end{aligned} \tag{8}$$

Eqn.(8) contains six terms. For simplicity of discussion, we remove the terms that are constant. We also assume the variance parameters such as $\sigma_m, \forall m = 1...M$, $\sigma_t, \forall t = 1...T$, and $\varepsilon$ equal one. Thus, the regularization parameters of the above terms are comparable and can be simply ignored.

Eqn.(8) can be minimized by updating one parameter with the remaining parameters fixed. First, although the first three terms are likely to be jointly convex with respect to $\mathbf{W}$, $\mathbf{x}_i$ in the first two terms is a highly nonlinear transformation with respect to $\mathbf{K}$, *i.e.* $\mathbf{x}_i = \Phi(I_i|\mathbf{K})$. In this case, no global optima are guaranteed. Therefore, following the optimization strategies of CNN [39], we apply stochastic gradient descent (SGD) [38] with weight decay [40] to search the suitable local optima for both $\mathbf{W}$ and $\mathbf{K}$. This method has been demonstrated working reasonably well in practice [38]. Here, the fifth term can be considered as the weight decay of the filters. Second, the third term in Eqn.(8) is a convex function regarding $\Upsilon$, but the fourth term is concave since negative logarithm is a convex function. In other words, learning $\Upsilon$ directly is a convex-concave problem [41]. However, with a well-known lemma [42], $\ln|\Upsilon|$ has a convex upper bound, $\ln|\Upsilon| \leq \operatorname{tr}(\Upsilon) - M - T$. Thus, the fourth term can be replaced by $D\operatorname{tr}(\Upsilon)$. Both the third and the fourth terms are now convex regarding $\Upsilon$. Finally, since the dynamic coefficients in Eqn.(8) are linear and independent, finding each $\lambda_t$ has a closed form solution.

### 3.3 Learning Algorithm

We solve the MAP problem in an iterative manner. First, we jointly update the DCN filters $\mathbf{K}$ and the weight matrix $\mathbf{W}$ with the tasks' dynamic coefficients $\Lambda$ and covariance matrix $\Upsilon$ fixed. Second, we update the covariance matrix $\Upsilon$ by fixing all the other parameters with their current values. Third, we update $\Lambda$ in a similar way to the second step.

**In the first step**, we optimize $\mathbf{W}$ and $\mathbf{K}$ in the DCN and fix $\Upsilon$ and $\Lambda$ with their current values. In this case, the fourth and the last terms in Eqn.(8) are constant and thus can be removed. We write the loss function in a matrix form as follows

$$\begin{aligned}E(\mathbf{I}) = \sum_{i=1}^{N}\Big\{&||\mathbf{y}_i - \mathbf{W}_M^{\mathsf{T}}\mathbf{x}_i||^2 - \boldsymbol{\ell}_i^{\mathsf{T}}\operatorname{diag}(\Lambda)\ln f(\mathbf{W}_T^{\mathsf{T}}\mathbf{x}_i) \\ &- (\mathbf{1}-\boldsymbol{\ell}_i)^{\mathsf{T}}\operatorname{diag}(\Lambda)\ln\big(\mathbf{1}-f(\mathbf{W}_T^{\mathsf{T}}\mathbf{x}_i)\big)\Big\} \\ &+ \operatorname{tr}(\mathbf{W}\Upsilon^{-1}\mathbf{W}^{\mathsf{T}}) + \operatorname{tr}(\mathbf{K}\mathbf{K}^{\mathsf{T}}),\end{aligned} \tag{9}$$

where $\mathbf{y}$ is a $M\times 1$ vector, and $\boldsymbol{\ell}$, $\mathbf{1}$ are both $T\times 1$ vectors. $\operatorname{diag}(\Lambda)$ represents a diagonal matrix with $\lambda_1, ..., \lambda_T$ being the values in the diagonal. The fourth term in Eqn.(9) can be considered as the parameterized weight decay of $\mathbf{W}$, while the last term is the weight decay of the filters $\mathbf{K}$, *i.e.* $\operatorname{tr}(\mathbf{K}\mathbf{K}^{\mathsf{T}}) = \sum_{s=1}^{S}\mathbf{k}_s^{\mathsf{T}}\mathbf{k}_s$. Eqn.(9) combines the least square loss and the cross-entropy loss to learn the DCN, which can be optimized by SGD [38], since they are defined over individual sample. Fig. 2 illustrates the architecture of DCN, containing four convolutional layers and one fully-connected layer. This architecture is a tradeoff between accuracy of landmark detection and computational cost, and it works well in practice. Note that the learning method introduced in this work is naturally compatible with any deep network structure, but exploring them is out of the scope of this paper.

Now we introduce the learning procedure. At the very beginning, each column of $\mathbf{W}$ and each filter of $\mathbf{K}$ are initialized according to a multivariate standard normal distribution. To learn the weight matrix $\mathbf{W}$, we calculate its derivative, $\Delta\mathbf{W} = -\eta\frac{\partial E}{\partial \mathbf{W}} = -\eta\frac{\partial E}{\partial \mathbf{o}}\frac{\partial \mathbf{o}}{\partial f}\frac{\partial f}{\partial \mathbf{W}}$, where $\mathbf{o}$ and $\eta$ denote the network outputs (predictions) and the step size of the gradient descent, respectively. By simple derivation, we have

$$\frac{\partial E}{\partial \mathbf{W}_M} = \mathbf{x}_i(\mathbf{y}_i - \mathbf{o}_i)^{\mathsf{T}}, \tag{10}$$

$$\frac{\partial E}{\partial \mathbf{W}_T} = \mathbf{x}_i(\boldsymbol{\ell}_i - \mathbf{o}_i)^{\mathsf{T}}\operatorname{diag}(\Lambda), \tag{11}$$

where $\mathbf{o}_i$ is the corresponding tasks' predictions. For example, $\mathbf{o}_i = \mathbf{W}_M^{\mathsf{T}}\mathbf{x}_i$ in Eqn.(10) indicates the predictions of the landmark positions, while $\mathbf{o}_i = f(\mathbf{W}_T^{\mathsf{T}}\mathbf{x}_i)$ in Eqn.(11) indicates the predictions of auxiliary attributes. In summary, the entire weight matrix in the $(j+1)$-th iteration is updated by $\mathbf{W}_{j+1} = \mathbf{W}_j - \eta_1\frac{\partial E}{\partial \mathbf{W}_j} - \eta_2(2\mathbf{W}_j\Upsilon^{-1})$, where $\eta_1, \eta_2$ are the regularization parameters of the gradient and the weight decay.

To update filters $\mathbf{K}$, we propagate the errors of DCN from top to bottom, following the well-known back-propagation (BP) strategy [43], where the gradient of each filter is computed by the cross-correlation between the corresponding input channel and the error map [39]. In particular, at the fully-connected layer as shown in Fig. 2, the errors are obtained by first summing over the losses of both landmark detection and attribute predictions, and then the sum is multiplied by the transpose of the weight matrix. For each convolutional layer, the errors are achieved by the de-convolution [39] between its filters and the back-propagated errors. Several pairs of face images and their features obtained

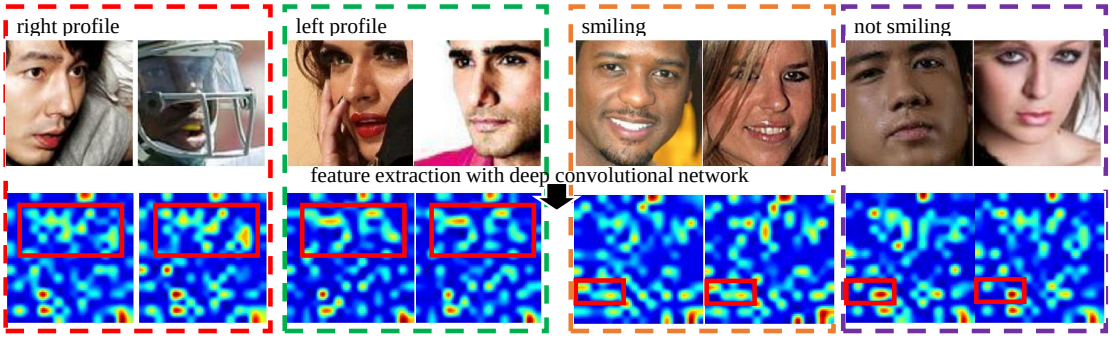



Fig. 3. The TCDCN learns shared features for facial landmark detection and auxiliary tasks. The first row shows the face images and the second row shows the corresponding features in the shared feature space, where the face images with similar poses and attributes are close with each other. This reveals that the learned feature space is robust to pose, expression, and occlusion.

by filters $\mathbf{K}$ are shown in Fig. 3, which shows that the learned features are robust to large poses and expressions. For example, the features of smiling faces or faces have similar poses exhibit similar patterns.

**In the second step**, we optimize the covariance matrix $\Upsilon$ with $\mathbf{W}$, $\mathbf{K}$, and $\Lambda$ fixed. As discussed in Eqn.(8), the logarithm of $\Upsilon$ can be relaxed by its upper bound. The optimization problem for finding $\Upsilon$ then becomes

$$\begin{aligned} &\min_{\Upsilon} \ \mathrm{tr}(\mathbf{W}\Upsilon^{-1}\mathbf{W}^{\mathsf{T}}) \\ &\text{s.t.} \quad \Upsilon \succeq \mathbf{0}, \ \ \mathrm{tr}(\Upsilon) \leq \eta. \end{aligned} \tag{12}$$

For simplicity, we assume $\eta = 1$. Problem (12) with respect to $\Upsilon$ is a naive semi-definite programming problem and has a simple closed form solution, which is $\Upsilon = \frac{(\mathbf{W}^{\mathsf{T}}\mathbf{W})^{\frac{1}{2}}}{\mathrm{tr}\left((\mathbf{W}^{\mathsf{T}}\mathbf{W})^{\frac{1}{2}}\right)}$.

**In the third step**, we update the dynamic coefficients $\Lambda$ with $\mathbf{W}$, $\mathbf{K}$, and $\Upsilon$ fixed. By ignoring the constant terms in Eqn.(8), the optimization problem becomes

$$\begin{aligned} \min_{\Lambda} \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} &-\lambda_t\{l_i^t \ln f(\mathbf{w}_t^{l\,\mathsf{T}}\mathbf{x}_i) \\ &+ (1-l_i^t)\ln\left(1-f(\mathbf{w}_t^{l\,\mathsf{T}}\mathbf{x}_i)\right)\} + \frac{1}{2}\sum_{t=1}^{T}(\lambda_t-\mu_t)^2, \\ s.t. \quad & 1 \geq \lambda_t \geq \epsilon, \ t = 1, 2, ..., T \end{aligned} \tag{13}$$

where $\epsilon$ is a small constant close to zero. Each $\lambda_t$ has a analytical solution, which is $\lambda_t = \min\left\{1, \max\left\{\epsilon, \ \mu_t + \frac{1}{N}\sum_{i=1}^{N} l_i^t \ln f(\mathbf{w}_t^{l\,\mathsf{T}}\mathbf{x}_i) + (1-l_i^t)\ln\left(1-f(\mathbf{w}_t^{l\,\mathsf{T}}\mathbf{x}_i)\right)\right\}\right\}$, implying that each dynamic coefficient is determined by its expected value and the loss value averaged over $N$ training samples. Here, we can define $\mu_t$ similar to the task-wise early stopping [24]. Suppose the current iteration is $j$, let $E_{val}^t(j)$, and $E_{tr}^t(j)$ be the values of the loss function of task $t$ on the validation set and training set, respectively. We can have

$$\mu_t = \rho \times \frac{E_{val}^t(j-\tau) - E_{val}^t(j)}{E_{val}^t(j-\tau)} \times \frac{E_{tr}^t(j-\tau) - E_{tr}^t(j)}{E_{tr}^t(j-\tau)}, \tag{14}$$

where $\rho$ is a constant scale factor, and $\tau$ controls a training strip of length $\tau$. The second term in Eqn.(14) represents the tendency of the validation error. If the validation error drops rapidly within a period of length $\tau$, the value of the first term is large, indicating that training should be emphasized as the task is valuable. Similarly, the third term measures the tendency of the training error. We can see that the task-wise early stopping strategy proposed in [24] can be treated as a special case of the dynamic coefficient $\lambda_t$. In addition, we does not need a tuned threshold to decide whether to stop a task as [24], and we can provide better performance (see Sec. 4.2).

### 3.4 Transferring TCDCN for Dense Landmarks

After training the TCDCN model on sparse landmarks and auxiliary attributes, it can be readily transferred from sparse landmark detection to handle more landmark points, *e.g.* 68 points as in 300-W dataset [20]. In particular, we initialize the network (*i.e.* , the lower part of Fig. 2) with the learned shared representation and fine-tune using a separate training set only labeled with dense landmark points. Since the shared representation of the pre-trained TCDCN model already captures the information from attributes, the auxiliary tasks learning is not necessary in the fine-tuning stage.

## 4 Implementation and Experiments

**Network Structure.** Fig. 2 shows the network structure of TCDCN. The input of the network is $60 \times 60$ gray-scale face image (normalized to zero-mean and unit-variance). The feature extraction stage contains four convolutional layers, three pooling layers, and one fully connected layer. The kernels in each convolutional layer produce multiple feature maps. The commonly used rectified linear unit is selected as the activation function. For the pooling layers, we conduct max-pooling on non-overlap regions of the feature map. The fully connected layer following the fourth convolutional layer produces

a feature vector that is shared by the multiple tasks in the estimation stage.

**Evaluation metrics**: In all cases, we report our results on two popular metrics [8], [9], [13], [27], *i.e.* mean error and failure rate. The mean error is measured by the distances between estimated landmarks and the ground truths, and normalized with respect to the inter-ocular distance. Mean error larger than 10% is reported as a failure.

### 4.1 Datasets

**Multi-Attribute Facial Landmark (MAFL)**[1]: To facilitate the training of TCDCN, we construct a new dataset by annotating 22 facial attributes on 20,000 faces randomly chosen from the Celebrity face dataset [44]. The attributes are listed in Table 1 and all the attributes are binary, indicating the attribute is presented or not. We divide the attributes into four groups to facilitate the following analyses. The grouping criterion is based on the main face region influenced by the associated attributes. In addition, we divide the face into one of five categories according to the degree of yaw rotation. This results in the fifth group named as "head pose". All the faces in the dataset are accompanied with five facial landmarks locations (eyes, nose, and mouth corners), which are used as the target of the face alignment task. We randomly select 1,000 faces for testing and the rest for training. Example images are provided in Fig. 12.

**Annotated Facial Landmarks in the Wild (AFLW) [45]:** AFLW contains 24,386 face images gathered from Flickr. This dataset is selected because it is more challenging than other conventional datasets, such as BioID [46] and LFPW [14]. Specifically, AFLW has larger pose variations (39% of faces are non-frontal in our testing images) and severe partial occlusions. Each face is annotated with 21 landmarks at most. Some landmarks are not annotated due to out-of-plane rotation or occlusion. We randomly select 3,000 faces for testing. Fig. 12 depicts some examples.

**Caltech Occluded Faces in the Wild (COFW) [8]:** This dataset is collected from the web. It is designed to present faces in occlusions due to pose, the use of accessories (*e.g.*, sunglasses), and interaction with objects (*e.g.*, food, hands). This dataset includes 1,007 faces, annotated with 29 landmarks, as shown in Fig. 15.

**Helen [23]:** Helen contains 2,330 faces from the web, annotated densely with 194 landmarks (Fig. 13).

**300-W [20]:** This dataset is well-known as a standard benchmark for face alignment. It is a collection of 3,837 faces from existing datasets: LFPW [14], AFW [5], Helen [23] and XM2VTS [47]. It also contains faces from an additional subset called IBUG, consisting images with difficult poses and expressions for face alignment, as shown in Fig. 16. Each face is densely annotated with 68 landmarks.

1. Data and codes of this work are available at http://mmlab.ie.cuhk.edu.hk/projects/TCDCN.html. (Ping Luo is the corresponding author)

TABLE 1
Annotated Face Attributes in MAFL Dataset

| Group | Attributes |
|---|---|
| eyes | bushy eyebrows, arched eyebrows, narrow eyes, bags under eyes, eyeglasses |
| nose | big nose, pointy nose |
| mouth | mouth slightly open, no beard, smiling, big lips, mustache |
| global | gender, oval face, attractive, heavy makeup, chubby |
| head pose | frontal, left, left profile, right, right profile |

TABLE 2
Comparison of mean error ($\times 10^{-2}$) on MAFL dataset under different network configurations

| | without inter-task correlation learning | with inter-task correlation learning |
|---|---|---|
| task-wise early stopping [24] | 8.35 | 8.21 |
| dynamic task coefficient | 8.07 | 7.95 |

### 4.2 Training with Dynamic Task Coefficient

Dynamic task coefficient is essential in TCDCN to coordinate the learning of different tasks with different convergence rates. To verify its effectiveness, we train the proposed TCDCN with and without this technique. Fig. 4 (a) plots the main task's error of the training and validation sets up to 200,000 iterations. Without dynamic task coefficient, the training error converges slowly and exhibits substantial oscillations. In contrast, convergence rates of both the training and validation sets are fast and stable when using the proposed dynamic task coefficient. In addition, we illustrate the dynamic task coefficients of two attributes in Fig. 4 (b). We observe that the values of their coefficients drop after a few thousand iterations, preventing these auxiliary tasks from over-fitting. The coefficients may increase when these tasks become effective in the learning process, as shown by the sawtooth-like pattern of the coefficient curves. These two behaviours work together, facilitating the smooth convergence of the main task, as shown in Fig. 4 (a).

In addition, we compare the dynamic tasks coefficient with the task-wise early stopping proposed in the earlier version of this work [24]. As shown in Table 2, dynamic task coefficient achieves better performance than the task-wise early stopping scheme. This is because the new method is more dynamic in coordinating the different auxiliary tasks across the whole training process (see Sec. 3.1).

### 4.3 Inter-task Correlation Learning

To investigate how the auxiliary tasks help facial landmark detection, we study the learned correlation between these tasks and the facial landmarks. In particular, as we have learned the task covariance matrix $\Upsilon$, given the relation between correlation matrix and covariance matrix, we can compute the correlation between any two tasks, by normalizing their covariance with the square root of the product of their variances.

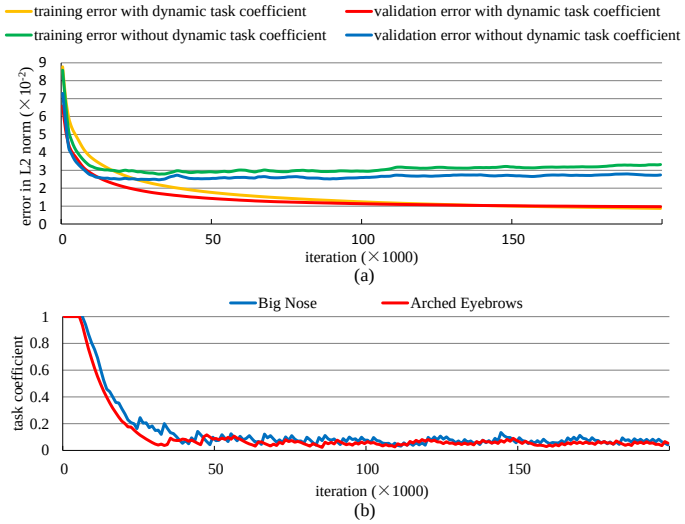



Fig. 4. (a) Facial landmark localization error curve with and without dynamic task coefficient. The error is measured in L2-norm with respect to the ground truth of the 10 coordinates values (normalized to [0,1]) for the 5 landmarks. (b) Task coefficients for the "Big Nose" and "Arched Eyebrows" attributes over the training process.

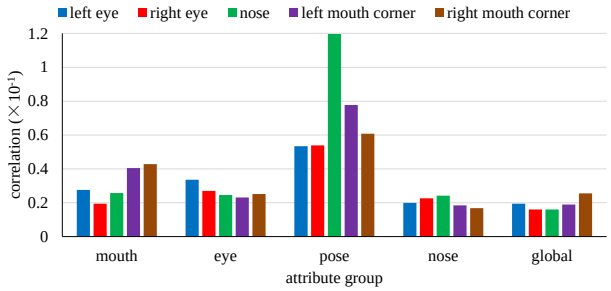



Fig. 5. Correlation of each attribute group with different landmarks.

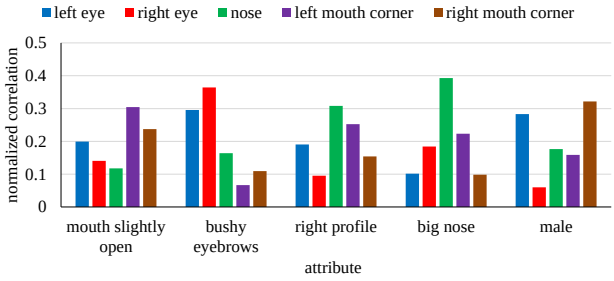



Fig. 6. Normalized correlation of the attributes with different landmarks. The attributes are randomly selected from each attribute group. The correlation is normalized among the five landmarks.

In Fig. 5, we present the learned correlation between the attribute groups and facial landmarks. In particular, for each attribute group, we compute the average absolute value of the correlation with the five facial landmarks, respectively. It is shown that for the group of "mouth", the correlations with the according landmarks (*i.e.* , mouth corners) are higher than the others. Similar trends can be observed in the group of "nose" and "eyes". For the group of "global", the correlations are roughly even for different landmarks because the attributes are determined by the global face structure. The correlation of the "pose" group is much higher than that of the others. This is because the head rotation directly affects the landmark distribution. Moreover, in Fig. 6, we randomly choose one attribute from each attribute group and visualize its correlation to other landmarks. For clarification, for each attribute, we normalize the correlation among the landmarks (*i.e.* , the sum of the correlation on the five landmarks equals one). We can also observe that the attributes are more likely to be correlated to its according landmarks.

In addition, we visualize the learned correlation between the auxiliary tasks in Fig. 7. Because the attributes of "Left Profile", "Left", "Frontal", "Right", "Right Profile" are mutually exclusive (i.e., only one attribute can be positive for a face) and describe the yaw rotation, we aggregate these five attributes as one attribute (*i.e.* "pose"), by computing the average absolute correlation with other attributes. One can observe some intuitive results in this figure. For examples, the head pose is unrelated to other attributes; "Heavy Makeup" has high positive correlation with "Attractive", and high negative correlation with "Male". In Table 2, we show the mean errors of facial landmark localization on MAFL dataset with and without inter-task correlation learning (without correlation learning means that we simply apply multiple tasks as targets and do not use the term of $\Upsilon$ in Eq. (8)). It demonstrates the effectiveness of task correlation learning.

### 4.4 Evaluating the Effectiveness of Auxiliary Task

To further examine the influence of auxiliary tasks more comprehensively, we evaluate different variants of the proposed model. In particular, the first variant is trained only on facial landmark detection. We train another five model variants on facial landmark detection along with the auxiliary tasks in the groups of "eyes", "nose", "mouth", "global", "head pose", respectively. In addition, we synthesize a task with random objective and train it along with the facial landmark detection task, which results in the sixth model variant. The full model is trained using all the attributes. For simplicity, we name each variant by facial landmark detection (FLD) and the auxiliary tasks, such as "FLD only", "FLD+eyes", "FLD+pose", "FLD+all".

It is evident from Fig. 8 that optimizing landmark detection with auxiliary tasks is beneficial. In particular, "FLD+all" outperforms "FLD" by a large margin, with a reduction of over 7% in failure rate. When single auxiliary task group is present, "FLD+pose" and "FLD+global" perform better than the others. This is not surprising since the pose variation affects locations of all landmarks directly and the "global' attribute group influences the whole face region. The other auxiliary tasks such as "eyes" and "mouth" are

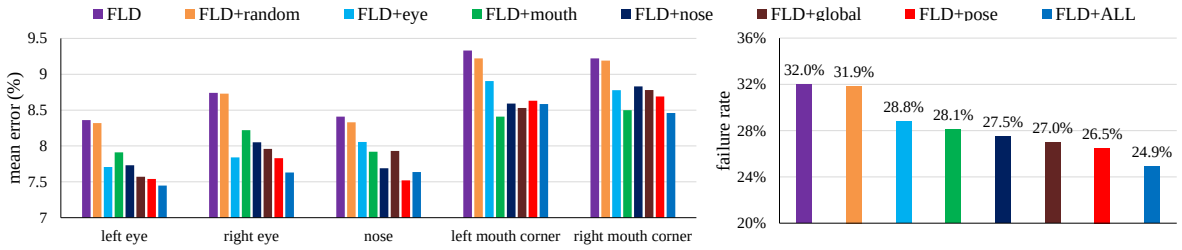



Fig. 8. Comparison of different model variants of TCDCN: the mean error over different landmarks (left), and the overall failure rate (right).

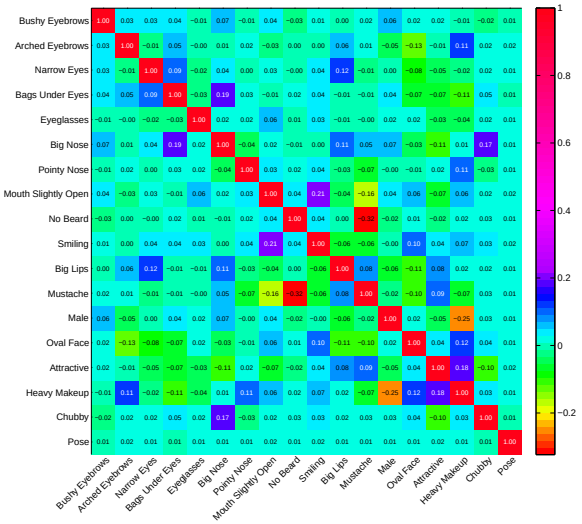



Fig. 7. Pairwise correlation of the auxiliary tasks learned by TCDCN (best viewed in color).

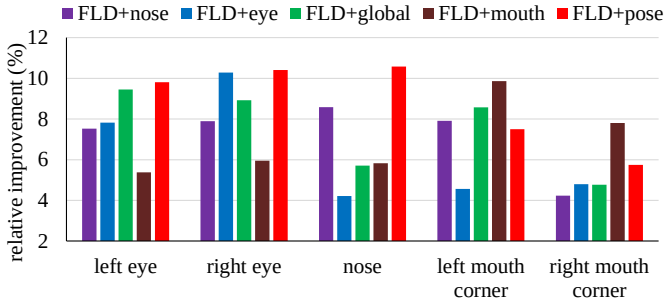



Fig. 9. Improvement over different landmarks by different attribute groups.

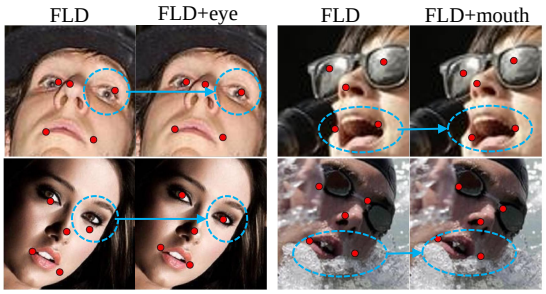



Fig. 10. Examples of improvement by attribute group of "eye" and "mouth".

observed to have comparatively smaller influence to the final performance, since they mainly capture local information of the face. As for "FLD+random" the performance is hardly improved. This result shows that the main task and auxiliary task need to be related for any performance gain in the main task.

In addition, we show the relative improvement caused by different groups of attributes for each landmark in Fig. 9. In particular, we define relative improvement $= \frac{\text{reduced error}}{\text{original error}}$, where original error is produced by the model of "FLD only". We can observe a trend that each group facilitates the landmarks in the according face region. For example, for the group of "mouth", the benefits are mainly observed at the corners of mouth. This observation is intuitive since attributes like smiling drives the lower part of the faces, involving Zygomaticus and levator labii superioris muscles, more than the upper facial region. The learning of these attributes develops a shared representation that describes lower facial region, which in turn facilitates the localization of corners of mouth. Similarly, the improvement of eye location is much more significant than mouth and nose for the attribute group of "eye". However, we observe the group of "nose" improves the eye and mouth localization remarkably. This is mainly because the nose is in the central of the face, there exists constrain between the nose location and other landmarks. The horizontal coordinate of the nose is likely to be the mean of the eyes in frontal face. As for the group of "pose" and "global", the improvement is significant in all landmarks. Fig. 10 depicts improvements led by adding "eye" and "mouth" attributes. Fig. 12 shows more example results, demonstrating the effectiveness on various face appearances of TCDCN.

### 4.5 Comparison with Deep Learning based Methods

Although the TCDCN, cascaded CNN [13] and CFAN [12] are built upon deep model, we show that the proposed model can achieve better detection accuracy with lower computational cost and model complexity. We use the full model "FLD+all", and the publicly available binary code of cascaded CNN [13] and CFAN [12] in this experiment.

**Landmark localization accuracy:** In this experiment, we employ the testing images of MAFL and AFLW [45] for evaluation. It is observed from Fig. 11 that the overall accuracy of the proposed method is superior to that of cascaded CNN and CFAN.

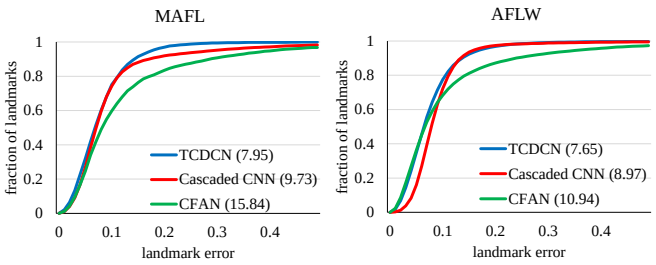



Fig. 11. Cumulative error curves of the proposed method, cascaded CNN [13] and CFAN [12], on the MAFL and AFLW [45] dataset (5 landmarks). The number in the legend indicates the according mean error ($\times 10^{-2}$).

TABLE 3
Comparison of Different Deep Modes for Facial Landmark Detection

| Model | #Parameter | Time (per face) |
|---|---|---|
| Cascaded CNN [13] | ∼990K | 120ms |
| CFAN [12] | ∼18,500K | 30ms |
| TCDCN | ∼100K | 18ms |

**Model complexity**: The proposed method only has one CNN, whereas the cascaded CNN [13] deploys multiple CNNs in different cascaded layers (23 CNNs in its implementation). Also, for each CNN, both our method and cascaded CNN [13] have four convolutional layers and two fully connected layers, with comparable input image size. However, the convolutional layer in [13] uses locally unshared kernels. Hence, TCDCN has much lower computational cost and model complexity. The cascaded CNN requires 0.12s to process an image on an Intel Core i5 CPU, whilst TCDCN only takes 18ms, which is *7 times faster*. Also, the TCDCN costs 1.5ms on a NVIDIA GTX760 GPU. Similarly, the complexity is larger in CFAN [12] due to the use of multiple auto-encoders, each of which contains fully connected structures in all layers. Table 3 shows the details of the running time and network complexity.

### 4.6 Face Representation Transfer and Comparison with State-of-the-art Methods

As we discussed in Section 3.4, we can transfer the trained TCDCN to handle more landmarks beyond the five major facial points. The main idea is to pre-train a TCDCN on sparse landmark annotations and multiple auxiliary tasks, followed by fine-tuning with dense landmark points.

We compare against various state-of-the-arts. The first class of methods use regression methods that directly predict the facial landmarks: (1) Robust Cascaded Pose Regression (RCPR) [8]; (2) Explicit Shape Regression (ESR) [9]; (3) Supervised Descent Method (SDM) [21]; (4) Regression based on Local Binary Features (LBF) [22]; (5) Regression based on Ensembles of Regression Trees [48] (ERT); (6) Coarse-to-Fine Auto-Encoder Networks (CFAN) [12] (as this method can predict dense landmarks, we compare with it again); (7) Coarse-to-Fine Shape Searching (CFSS) [49]. The second class of methods employ a face template: (8) Tree Structured Part Model (TSPM) [5], which jointly estimates the head pose and facial landmarks; (9) A Cascaded Deformable Model (CDM) [6]; (10) STASM [50], which is based on Active Shape Model [51]; (11) Component-based ASM [23]; (12) Robust discriminative response map fitting (DRMF) method [31]; (13) Gauss-newton deformable part models (GN-DPM) [7]; In addition, we compare with the commercial face analysis software: (14) Face++ API [52]. For the methods of RCPR [8], SDM [21], CFAN [12], TSPM [5], CDM [6], STASM [50], DRMF [31], and Face++ [52], we use their publicly available implementation. For the methods which include their own face detector (like TSPM [5] and CDM [6]), we avoid detection errors by cropping the image around the face. For methods that do not release the code, we report their results on the related literatures.

TABLE 4
Mean errors (%) on Helen [23] dataset

| Method | 194 landmarks | 68 landmarks |
|---|---|---|
| STASM [50] | 11.1 | - |
| CompASM [23] | 9.10 | - |
| DRMF [31] | - | 6.70 |
| ESR [9] | 5.70 | - |
| RCPR [8] | 6.50 | 5.93 |
| SDM [21] | 5.85 | 5.50 |
| LBF [22] | 5.41 | - |
| CFAN [12] | - | 5.53 |
| CDM [6] | - | 9.90 |
| GN-DPM [7] | - | 5.69 |
| ERT [48] | 4.90 | - |
| CFSS [49] | 4.74 | 4.63 |
| TCDCN | **4.63** | **4.60** |

Fig. 13. Example alignment results on Helen [23].

**Evaluation on Helen [23]**: Helen consists of 2,000 training and 330 testing images as specified in [23]. In particular, the 194-landmark annotation is from the original dataset and the 68-landmark annotation is from [20]. Table 4 reports the performance of the competitors and the proposed method. Most of the images are in high resolution and the faces are near-frontal. Although our method just uses the input of $60 \times 60$ grey image, it still achieves better result. Fig. 13 visualizes some of our results. It can be observed that driven by rich facial attributes, our model can capture various facial expression accurately.

**Evaluation on 300-W [20]**: We follow the same protocol in [22]: the training set contains 3,148 faces, including

Fig. 12. Example alignment results on MAFL dataset (the first row), and AFLW [45] datasets (the second row). Red rectangles indicate wrong cases.

TABLE 5
Mean errors (%) on 300-W [20] dataset (68 landmarks)

| Method | Common Subset | Challenging Subset | Fullset |
|---|---|---|---|
| CDM [6] | 10.10 | 19.54 | 11.94 |
| DRMF [31] | 6.65 | 19.79 | 9.22 |
| RCPR [8] | 6.18 | 17.26 | 8.35 |
| GN-DPM [7] | 5.78 | - | - |
| CFAN [12] | 5.50 | 16.78 | 7.69 |
| ESR [9] | 5.28 | 17.00 | 7.58 |
| SDM [21] | 5.57 | 15.40 | 7.50 |
| ERT [48] | - | - | 6.40 |
| LBF [22] | 4.95 | 11.98 | 6.32 |
| CFSS [49] | **4.73** | 9.98 | 5.76 |
| TCDCN | 4.80 | **8.60** | **5.54** |

the AFW, the training sets of LFPW, and the training sets of Helen. The test set contains 689 faces, including IBUG, the testing sets of LFPW, and the testing sets of Helen. Table 5 demonstrates the superior of the proposed method. In particular, for the challenging subset (IBUG faces) TCDCN produces a significant error reduction of over **10%** in comparison to the state-of-the-art [49]. As can be seen from Fig. 16, the proposed method exhibits superior capability of handling difficult cases with large head rotation and exaggerated expressions, thanks to the shared representation learning with auxiliary tasks. Fig. 17 shows more results of the proposed method on Helen [23], IBUG [20], and LPFW [14] datasets.

**Evaluation on COFW [8]**: The testing protocol is the same as [8], where the training set includes LFPW [14] and 500 COFW faces, and the testing set includes the remaining 507 COFW faces. This dataset is more challenging as it is collected to emphasize the occlusion cases. The quantitative evaluation is reported in Fig. 14. Example results of our algorithm are depicted in Fig. 15. It is worth pointing out that the proposed method achieves better performance than RCPR [8] even that we do not explicitly learn and detect occlusions as [8].

## 5 Conclusions

Instead of learning facial landmark detection in isolation, we have shown that more robust landmark detection can be achieved through joint learning with heterogeneous but subtly correlated auxiliary tasks, such as appearance attribute, expression, demographic, and head pose. The proposed Tasks-Constrained DCN allows errors of auxiliary tasks to be back-propagated in deep hidden layers for constructing a shared representation to be

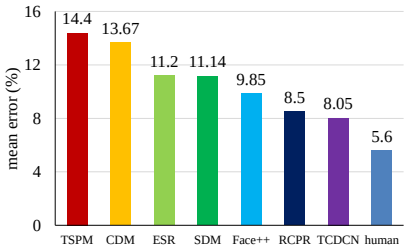



Fig. 14. Mean errors on COFW [8] dataset (29 landmarks) for the method of TSPM [5], CDM [6], ESR [9], SDM [21], Face++ [52], RCPR [8] and the proposed method.

Fig. 15. Example alignment results on COFW [8] dataset.

relevant to the main task. We also show that by learning dynamic task coefficient, we can utilize the auxiliary tasks in a more efficient way. Thanks to learning with the auxiliary attributes, the proposed model is more robust to faces with severe occlusions and large pose variations compared to existing methods. We have observed that a deep model needs not be cascaded [13] to achieve the better performance. The lighter-weight CNN allows real-time performance without the usage of GPU or parallel computing techniques. Future work will explore deep learning with auxiliary information for other vision problems.

## References

[1] N. Kumar, P. Belhumeur, and S. Nayar, "Facetracer: A search engine for large collections of images with faces," in *European Conference on Computer Vision*, 2008, pp. 340–353.
[2] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with GaussianFace," in *AAAI Conference on Artificial Intelligence*, 2015.
[3] Z. Huang, X. Zhao, S. Shan, R. Wang, and X. Chen, "Coupling alignments with recognition for still-to-video face recognition," in *IEEE International Conference on Computer Vision*, 2013, pp. 3296–3303.
[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
[5] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.


Fig. 16. Results of ESR [9], SDM [21], LBF [22] and our method on the IBUG faces [20].

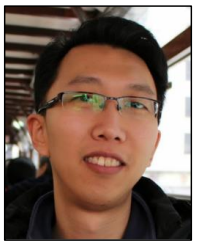
Fig. 17. Example alignment results on Helen [23], IBUG [20], and LPFW [14] datasets (68 landmarks). Red rectangles indicate wrong cases.

[6] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *IEEE International Conference on Computer Vision*, 2013, pp. 1944–1951.
[7] G. Tzimiropoulos and M. Pantic, "Gauss-newton deformable part models for face alignment in-the-wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1851–1858.
[8] X. P. Burgos-Artizzu, P. Perona, and P. Dollar, "Robust face landmark estimation under occlusion," in *IEEE International Conference on Computer Vision*, 2013, pp. 1513–1520.
[9] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2887–2894.
[10] T. F. Cootes, M. C. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting," in *European Conference on Computer Vision*, 2012, pp. 278–291.
[11] H. Yang and I. Patras, "Sieving regression forest votes for facial feature detection in the wild," in *IEEE International Conference on Computer Vision*, 2013, pp. 1936–1943.
[12] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment," in *European Conference on Computer Vision*, 2014, pp. 1–16.
[13] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.
[14] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 545–552.
[15] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2467–2474.
[16] P. Luo, X. Wang, and X. Tang, "A deep sum-product architecture for robust facial attributes analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2864–2871.
[17] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing, "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks," in *European Conference on Computer Vision*. Springer, 2008, pp. 69–82.
[18] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 639–655.
[19] S. Li, Z.-Q. Liu, and A. Chan, "Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network," *International Journal of Computer Vision*, 2014.
[20] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: the first facial landmark localization challenge," in *IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.
[21] X. Xiong and F. De La Torre, "Supervised descent method and its applications to face alignment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 532–539.
[22] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685 – 1692.
[23] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *European Conference on Computer Vision*, 2012, pp. 679–692.
[24] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European Conference on Computer Vision*, 2014, pp. 94–108.

[25] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial point detection using boosted regression and graph models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2729–2736.
[26] P. Dollár, P. Welinder, and P. Perona, "Cascaded pose regression," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1078–1085.
[27] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2578–2585.
[28] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *European Conference on Computer Vision*, 2014, pp. 109–122.
[29] X. Liu, "Generic face alignment using boosted appearance model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
[30] M. Pedersoli, R. Timofte, T. Tuytelaars, and L. V. Gool, "Using a deformation field model for localizing faces and facial points under weak supervision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3694–3701.
[31] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3444–3451.
[32] W. L. Baoguang Shi, Xiang Bai and J. Wang, "Deep regression for face alignment," arXiv:1409.5230, Tech. Rep., 2014.
[33] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.
[34] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *International Conference on Machine Learning*, 2008, pp. 160–167.
[35] Y. Liu, A. Wu, D. Guo, K.-T. Yao, and C. Raghavendra, "Weighted task regularization for multitask learning," in *International Conference on Data Mining Workshops*, 2013, pp. 399–406.
[36] P. McCullagh, J. A. Nelder, and P. McCullagh, *Generalized linear models*. Chapman and Hall London, 1989.
[37] A. K. Gupta and D. K. Nagar, *Matrix variate distributions*. CRC Press, 1999.
[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[40] J. Moody, S. Hanson, A. Krogh, and J. A. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, pp. 950–957, 1995.
[41] A. L. Yuille, A. Rangarajan, and A. Yuille, "The concave-convex procedure (cccp)," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1033–1040, 2002.
[42] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[43] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
[44] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.
[45] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *IEEE International Conference on Computer Vision Workshops*, 2011, pp. 2144–2151.
[46] O. Jesorsky, K. J. Kirchberg, and R. Frischholz, "Robust face detection using the hausdorff distance," in *International Conference on Audio- and Video-Based Biometric Person Authentication*, 2001, pp. 90–95.
[47] K. Messer, J. Matas, J. Kittler, J. Lttin, and G. Maitre, "Xm2vtsdb: the extended m2vts database," in *International Conference on Audio and Video-based Biometric Person Authentication*, 1999, pp. 72–77.
[48] V. Kazemi and S. Josephine, "One millisecond face alignment with an ensemble of regression trees," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.
[49] S. Zhu, C. Li, C. C. Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4998–5006.
[50] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *European Conference on Computer Vision*, 2008, pp. 504–513.
[51] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.
[52] Megvii Inc., "Face++," http://www.faceplusplus.com.

**Zhanpeng Zhang** received the B.E. and M.E. degrees from Sun Yat-sen University, Guangzhou, China, in 2010 and 2013, respectively. He is currently working toward the Ph.D. degree with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include computer vision and machine learning, in particular, face tracking and analysis.

**Ping Luo** is a postdoctoral researcher at the Department of Information Engineering, The Chinese University of Hong Kong, where he received his Ph.D. in 2014. His research interests include deep learning, computer vision, and computer graphics, focusing on face analysis, pedestrian analysis, and large-scale object recognition and detection.



**Chen Change Loy** received the PhD degree in Computer Science from the Queen Mary University of London in 2010. He is currently a Research Assistant Professor in the Department of Information Engineering, Chinese University of Hong Kong. Previously he was a postdoctoral researcher at Vision Semantics Ltd. His research interests include computer vision and pattern recognition, with focus on face analysis, deep learning, and visual surveillance.

**Xiaoou Tang** received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a Professor in the Department of Information Engineering and Associate Dean (Research) of the Faculty of Engineering of the Chinese University of Hong Kong. He worked as the group manager of the Visual Computing Group at the Microsoft Research Asia from 2005 to 2008. His research interests include computer vision, pattern recognition, and video processing. Dr. Tang received the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009. He is a program chair of the IEEE International Conference on Computer Vision (ICCV) 2009 and has served as an Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) and International Journal of Computer Vision (IJCV). He is a Fellow of IEEE.