Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition

Xi Yin and Xiaoming Liu *Member, IEEE*

Abstract—**This paper explores Multi-Task Learning (MTL) for face recognition. First, we propose a multi-task Convolutional Neural Network (CNN) for face recognition where identity classification is the main task and Pose, Illumination, and Expression (PIE) estimations are the side tasks. Second, we develop a dynamic-weighting scheme to automatically assign the loss weights to each side task, which solves the crucial problem of balancing between different tasks in MTL. Third, we propose a pose-directed multi-task CNN by grouping different poses to learn pose-specific identity features, simultaneously across all poses in a joint framework. Last but not least, we propose an energy-based weight analysis method to explore how CNN-based MTL works. We observe that the side tasks serve as regularizations to disentangle the PIE variations from the learnt identity features. Extensive experiments on the entire Multi-PIE dataset demonstrate the effectiveness of the proposed approach. To the best of our knowledge, this is the first work using all data in Multi-PIE for face recognition. Our approach is also applicable to in-the-wild datasets for pose-invariant face recognition and achieves comparable or better performance than state of the art on LFW, CFP, and IJB-A datasets.**

Index Terms—**multi-task learning, pose-invariant face recognition, CNN, disentangled representation**

I. Introduction

FACE recognition is a challenging problem that has been studied for decades in computer vision. The large intra-person variations in Pose, Illumination, Expression (PIE), and etc. will challenge any state-of-the-art face recognition algorithms. Recent CNN-based approaches mainly focus on exploring the effects of 3D model-based face alignment [49], larger datasets [49], [41], or new metric learning algorithms [45], [41], [32], [53] on face recognition performance. Most existing methods consider face recognition as a single task problem. We believe that face recognition is not an isolated problem — often tangled with other tasks. This motivates us to explore multi-task learning for face recognition.

Multi-Task Learning (MTL) aims to learn several tasks *simultaneously* to boost the performance of the main task or all tasks. It has been successfully applied to face detection [7], [59], face alignment [62], pedestrian detection [50], attribute estimation [1], and so on. Despite the success of MTL in various vision problems, there is a lack of study on MTL for face recognition. In this paper, we study face recognition as a multi-task problem where identity classification is the main task with PIE estimations being the side tasks. The goal is to leverage the side tasks to improve the performance of the main task, i.e., face recognition.

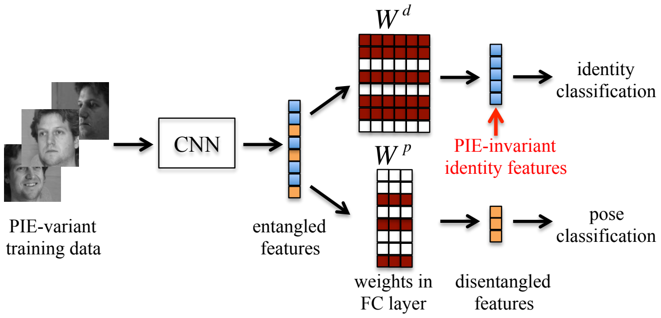


Fig. 1. We propose MTL for face recognition with identity classification as the main task and PIE classifications as the side tasks (only pose is illustrated in this figure for simplicity). A CNN framework learns entangled features from the data. The weight matrix in the fully connected layer of the main task is learnt to have close-to-zero values for PIE features in order to exclude PIE variations, which results in PIE-invariant identity features for face recognition.

We answer the questions of how and why PIE estimations can help face recognition. Regarding how, we propose a multi-task CNN (m-CNN) framework for joint identity classification and PIE classifications, which learns a shared embedding. Regarding why, we conduct an energy-based weight analysis and observe that the side tasks serve as regularizations to inject PIE variations into the shared embedding, which is further disentangled into PIE-invariant identity features for face recognition. As shown in Figure 1, a shared embedding is learnt from PIE-variant training images through a CNN framework. A fully connected layer is connected to the shared features to perform classification of each task. The shared features entangles both identity and PIE variations, and the weight matrix in the fully connected layer performs feature selection for disentanglement.

One crucial problem in MTL is how to determine the importance of each task. Prior work either treats each task equally [57] or obtains the loss weights by greedy search [50]. It may not be fair to assume that each task contributes equally. However, it will be very time consuming or practically impossible to find the optimal weights for all side tasks via brute-force search. Instead, we propose a dynamic-weighting scheme where we only need to determine the overall loss weight for the PIE estimations, and the CNN can learn to dynamically assign a loss weight to each side task during training. This is effective and efficient as will shown in Section IV.

Since pose variation is the most challenging one among other non-identity variations, and the proposed m-CNN already classifies all images into different pose groups, we propose to apply divide-and-conquer to the CNN learning. Specifically, we develop a novel pose-directed multi-task CNN (p-CNN) where the pose labels can categorize the training data

Xi Yin and Xiaoming Liu are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824. Corresponding author: Xiaoming Liu, liuxm@cse.msu.edu

TABLE I
Comparison of the experimental settings that are commonly used in prior work on Multi-PIE. (* The 20 images consist of 2 duplicates of non-flash images and 18 flash images. In total there are 19 different illuminations.)

setting	session	pose	illum	exp	train subjects / images	gallery / probe images	total	references
I	4	7	1	1	200 / 5,383	137 / 2,600	8,120	[4], [28]
II	1	7	20	1	100 / 14,000	149 / 20,711	34,860	[65], [57]
III	1	15	20	1	150 / 45,000	99 / 29,601	74,700	[54]
IV	4	9	20	1	200 / 138,420	137 / 70,243	208,800	[66], [51]
V	4	13	20	1	200 / 199,940	137 / 101,523	301,600	[58]
ours	4	15	20*	6	200 / 498,900	137 / 255,163	754,200	

into three different pose groups, direct them through different routes in the network to learn pose-specific identity features in addition to the generic identity features. Similarly, the loss weights for extracting these two types of features are learnt dynamically in the CNN framework. During the testing stage, we propose a stochastic routing scheme to fuse the generic identity features and the pose-specific identity features for face recognition, which is more robust to pose estimation errors. We find this technique to be very effective for pose-invariant face recognition especially for in-the-wild faces, where pose classification error is more likely to happen compared to controlled datasets with discrete pose angles.

This work utilizes *all* data in the Multi-PIE dataset [16], i.e., faces with the full range of PIE variations, as the main experimental dataset — ideal for studying MTL for PIE-invariant face recognition. To the best of our knowledge, there is no prior face recognition work that studies the full range of variations on Multi-PIE. We also apply our method to in-the-wild datasets for pose-invariant face recognition. Since the ground truth label of the side task is unavailable, we use the estimated poses as labels for training.

In summary, we make four contributions.

- We formulate face recognition as an MTL problem and explore how it works via an energy-based weight analysis.
- We propose a dynamic-weighting scheme to learn the loss weights for each side task automatically in the CNN.
- We develop a pose-directed multi-task CNN to learn pose-specific identity features and a stochastic routing scheme for feature fusion during the testing stage.
- We perform a comprehensive and the first face recognition study on the entire Multi-PIE. We achieve comparable or superior performance to state-of-the-art methods on Multi-PIE, LFW [20], CFP [42], and IJB-A [26].

II. Related Work

A. Face Recognition

Recent progress in face recognition has been mainly focused on developing metric learning algorithms including center loss [53], A-softmax [33], N-pair [44], etc. In this work, we study Pose-Invariant Face Recognition (PIFR) via CNN-based multi-task learning. Therefore, we focus our review on PIFR and MTL-based approaches.

Pose-Invariant Face Recognition According to [11], existing PIFR methods can be classified into four categories including: multi-view subspace learning [27], [2], pose-invariant feature extraction [6], [41], [39], face synthesis [64], [19], [58], and a hybrid approach of the above three [51], [57]. For example, FF-GAN [58] incorporates a 3D Morphable Model (3DMM) [5] into a CNN framework for face frontalization with various loss functions. The frontalized faces can be used to improve the face recognition performance especially for large-pose faces. By modeling the face rotation process, DR-GAN [51] learns both a generative and discriminative representation from one or multiple face images of the same subject. This representation can be used for PIFR and face image synthesis.

Our work belongs to the second category, i.e., pose-invariant feature extraction. Previous work in this category treats each pose separately. For example, Masi et al. [34] propose a pose-aware face recognition method by learning a specific model for each type of face alignment and pose group. And the results are fused together during the testing stage. The idea of divide-and-conquer is similar to our work. Differently, we learn the pose-invariant identity features for all poses jointly in one CNN framework and propose a stochastic routing scheme during the testing stage for feature fusion, which is more efficient and robust. Xiong et al. [54] propose a conditional CNN for PIFR, which discovers the modality information automatically during training. In contrast, we utilize the pose labels as a side task to better disentangle pose variation from the learnt identity features. Peng et al. [39] use the pose classification as a side task to learn a rich embedding, which is further disentangled via pair-wise reconstruction. However, it need to be trained on datasets with non-frontal and frontal face pairs, which is not widely available especially for in-the-wild scenario.

MTL for Face Recognition For MTL-based face recognition method, Ding et al. [12] propose to transform the features of different poses into a discriminative subspace, and the transformations are learnt jointly for all poses with one task for each pose. [57] develops a deep neural network to rotate a face image. The reconstruction of the face is considered as a side task. This multi-task framework is more effective than the single task model without the reconstruction part. Similar work [66], [51] have developed along this direction to extract robust identity features and synthesize face images simultaneously. In this work, we treat face recognition as a multi-task problem with PIE estimations as the side tasks. It sounds intuitive to have PIE as side tasks for face recognition, but we are actually the first to consider this and we have explored how it works.

B. Multi-Task Learning

Multi-task learning has been widely studied in machine learning [15], [3] and computer vision [50], [62]. We focus our review on different regularizations in MTL, CNN-based MTL, and the importance of each task in MTL.

Regularizations The underlying assumption for most MTL algorithms is that different tasks are related to each other. Thus, a key problem is how to determine the task relatedness. One common way is to learn shared features for different tasks. The generalized linear model parameterizes each task with a weight vector. The weight vectors of all tasks form a weight matrix, which is regularized by $l_{2,1}$ norm [3], [37] or trace norm [22] to encourage a low-rank matrix. For example, Obozinski et al. [37] propose to penalize the sum of l_2 norm of the blocks of weights associated with each feature across different tasks to encourage similar sparsity patterns. Lin et al. [30] propose to learn higher order feature interaction without limiting to a linear model for MTL. Other work [14], [61], [31] propose to learn the task relationship from a task covariance matrix computed from the data.

CNN-based MTL It is natural to fuse MTL with CNN to learn the shared features and the task-specific models. For example, [62] proposes a deep CNN for joint face detection, pose estimation, and landmark localization. Misra et. al. [35] propose a cross-stitch network for MTL to learn the sharing strategy, which is difficult to scale to multiple tasks. This requires training one model for each task and introduces additional parameters in combining them. In [59], a task-constrained deep network is developed for landmark detection with facial attribute classifications as the side tasks. However, unlike the regularizations used in the MTL formulation in the machine learning community, there is no principled method to analysis how MTL works in the CNN framework. In this paper, we propose an energy-based weight analysis method to explore how MTL works. We discover that the side tasks of PIE estimations serve as regularizations to learn more discriminative identity features that are robust to PIE variations.

Importance of Each Task It is important to determine the loss weight for each task in MTL. The work of [57] uses equal loss weights for face recognition and face frontalization. Tian et al. [50] propose to obtain the loss weights of all side tasks via greedy search within 0 and 1. Let t and k be the number of side tasks and searched values respectively. This approach has two drawbacks. First, it is very inefficient as the computation scales to the number of tasks (complexity tk). Second, the optimal loss weight obtained for each task may not be jointly optimal. Further, the complexity would be k^t if searching all combinations in a brute-force way. Zhang et al. [62] propose a task-wise early stopping to halt a task during training when the loss no longer reduces. However, a stopped task will never resume so its effect may disappear. In contrast, we propose a dynamic-weighting scheme where we only determine the overall loss weight for all side tasks (complexity k) and let CNN learn to automatically distribute the weights to each side task. In this case when one task is saturated, we have observed the dynamic weights will reduce without the need of early stopping.

III. The Proposed Approach

In this section, we present the proposed approach by using Multi-PIE dataset as an example and extend it to in-the-wild datasets in the experiments. First, we propose a multi-task CNN (m-CNN) with dynamic weights for face recognition (the main task) and PIE estimations (the side tasks). Second, we propose a pose-directed multi-task CNN (p-CNN) to tackle pose variation by separating all poses into different groups and jointly learning pose-specific identity features for each group.

A. Multi-Task CNN

We combine MTL with CNN framework by sharing some layers between different tasks. In this work, we adapt CASIA-Net [56] with three modifications. First, Batch Normalization (BN) [21] is applied to accelerate the training process. Second, the contrastive loss is excluded for simplicity. Third, the output dimension of the fully connected layer is changed according to different tasks. Details of the layer parameters are shown in Figure 2. The network consists of five blocks each including two convolutional layers and a pooling layer. BN and ReLU [36] are used after each convolutional layer. Similar to [56], no ReLU is used after conv52 layer to learn a compact feature representation, and a dropout layer with a ratio of 0.4 is applied after pool5 layer.

Given a training set $\mathbf{T}$ with N images and their labels: $\mathbf{T} = \{\mathbf{I}_i, \mathbf{y}_i\}_{i=1}^N$, where $\mathbf{I}_i$ is the image and $\mathbf{y}_i$ is a vector consisting of the identity label y_i^d (main task) and the side task labels. In our work, we consider three side tasks including pose (y_i^p), illumination (y_i^l), and expression (y_i^e). We eliminate the sample index i for clarity. As shown in Figure 2, the proposed m-CNN extracts a high-level feature embedding $\mathbf{x} \in \mathbb{R}^{D\times 1}$:

$$\mathbf{x} = f(\mathbf{I}; \mathbf{k}, \mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\beta}), \tag{1}$$

where $f(\cdot)$ represents the non-linear mapping from the input image to the shared features. $\mathbf{k}$ and $\mathbf{b}$ are the sets of filters and bias of all the convolutional layers. $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are the sets of scales and shifts in the BN layers [21]. Let $\boldsymbol{\Theta} = \{\mathbf{k}, \mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\beta}\}$ denote all parameters to be learnt to extract the features $\mathbf{x}$.

The extracted features $\mathbf{x}$, which is pool5 in our model, are shared among all tasks. Suppose $\mathbf{W}^d \in \mathbb{R}^{D\times D_d}$ and $\mathbf{b}^d \in \mathbb{R}^{D_d\times 1}$ are the weight matrix and bias vector in the fully connected layer for identity classification, where D_d is the number of different identities in $\mathbf{T}$. The generalized linear model can be applied:

$$\mathbf{y}^d = \mathbf{W}^{d\intercal}\mathbf{x} + \mathbf{b}^d. \tag{2}$$

$\mathbf{y}^d$ is fed to a softmax layer to compute the probability of $\mathbf{x}$ belonging to each subject in the training set:

$$softmax(\mathbf{y}^d)_n = p(\hat{y}^d = n|\mathbf{x}) = \frac{\exp(\mathbf{y}_n^d)}{\sum_j \exp(\mathbf{y}_j^d)}, \tag{3}$$

where $\mathbf{y}_j^d$ is the jth element in $\mathbf{y}^d$. The $softmax(\cdot)$ function converts the output $\mathbf{y}^d$ to a probability distribution over all subjects and the subscript selects the nth element. Finally, the estimated identity $\hat{y}^d$ is obtained via:

$$\hat{y}^d = \underset{n}{\operatorname{argmax}} \quad softmax(\mathbf{y}^d)_n. \tag{4}$$

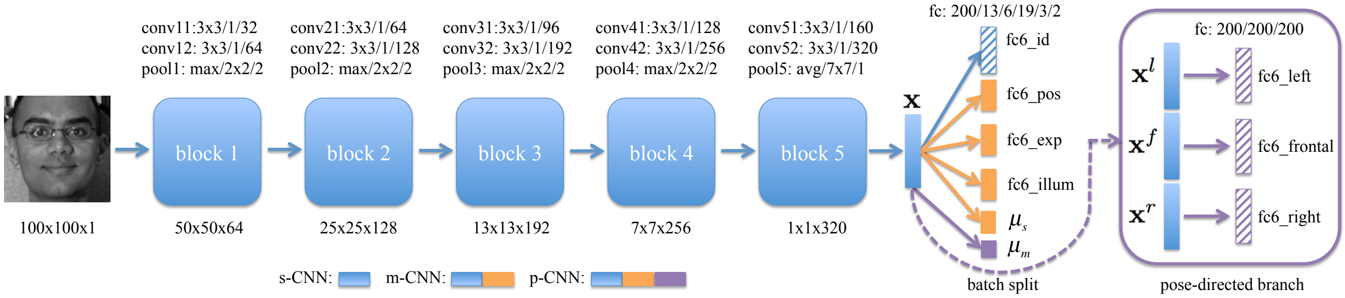


Fig. 2. The proposed m-CNN and p-CNN for face recognition. Each block reduces the spatial dimensions and increases the feature channels. The parameter format for the convolutional layer is: filter size / stride / filter number. The parameter format for the pooling layer is: method / filter size / stride. The feature dimensions after each block are shown on the bottom. The color indicates the component for each model. The dashed line represents the batch split operation as shown in Figure 3. The layers with the stripe pattern are the identity features used in the testing stage for face recognition.

The cross-entropy loss is employed:

$$L(\mathbf{I}, y^d) = -\log(p(\hat{y}^d = y^d | \mathbf{I}, \mathbf{\Theta}, \mathbf{W}^d, \mathbf{b}^d)). \tag{5}$$

Similarly, we formulate the losses for the side tasks. Let $\mathbf{W} = \{\mathbf{W}^d, \mathbf{W}^p, \mathbf{W}^l, \mathbf{W}^e\}$ represent the weight matrices for identity and PIE classifications. The bias terms are eliminated for simplicity. Given the training set $\mathbf{T}$, our m-CNN aims to minimize the combined loss of all tasks:

$$\begin{aligned} \underset{\mathbf{\Theta}, \mathbf{W}}{\operatorname{argmin}} \quad & \alpha_d \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^d) + \alpha_p \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^p) + \\ & \alpha_l \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^l) + \alpha_e \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^e), \end{aligned} \tag{6}$$

where α_d, α_p, α_l, α_e control the importance of each task. It becomes a single-task model (s-CNN) when $\alpha_{p,l,e} = 0$. The loss drives the model to learn both the parameters $\mathbf{\Theta}$ for extracting the shared features and $\mathbf{W}$ for the classification tasks. In the testing stage, the features before the softmax layer ($\mathbf{y}^d$) are used for face recognition by applying a face matching procedure based on cosine similarity.

B. Dynamic-Weighting Scheme

In MTL, it is an open question on how to set the loss weight for each task. Prior work either treats all tasks equally [57] or obtains the loss weights via brute-force search [50], which is very time-consuming especially considering the training time for CNN models. To solve this problem, we propose a dynamic-weighting scheme to automatically assign the loss weights to each side task during training.

First, we set the weight for the main task to 1, i.e. $\alpha_d = 1$. Second, instead of finding the loss weight for each task, we find the summed loss weight for all side tasks, i.e. $\varphi_s = \alpha_p + \alpha_l + \alpha_e$, via brute-force search on a validation set. Our m-CNN learns to allocate φ_s to three side tasks. As shown in Figure 2, we add a fully connected layer and a softmax layer to the shared features $\mathbf{x}$ to learn the dynamic weights. Let $\boldsymbol{\omega}_s \in \mathbb{R}^{D \times 3}$ and $\boldsymbol{\epsilon}_s \in \mathbb{R}^{3 \times 1}$ denote the weight matrix and bias vector in this fully connected layer,

$$\boldsymbol{\mu}_s = softmax({\boldsymbol{\omega}_s}^\intercal \mathbf{x} + \boldsymbol{\epsilon}_s), \tag{7}$$

where $\boldsymbol{\mu}_s = [\mu_p, \mu_l, \mu_e]^\intercal$ are the dynamic weight percentages for the side tasks with $\mu_p + \mu_l + \mu_e = 1$ and $\mu_{p,l,e} \geq 0$. So Equation 6 becomes:

$$\begin{aligned} \underset{\mathbf{\Theta}, \mathbf{W}, \boldsymbol{\omega}_s}{\operatorname{argmin}} \quad & \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^d) + \varphi_s \Big[\mu_p \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^p) + \\ & \mu_l \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^l) + \mu_e \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^e) \Big] \\ s.t. \quad & \mu_p + \mu_l + \mu_e = 1, \quad \mu_{p,l,e} \geq 0 \end{aligned} \tag{8}$$

The multiplications of the overall loss weight φ_s with the learnt dynamic percentage $\mu_{p,l,e}$ are the dynamic loss weights for each side task, i.e., $\alpha_{p,l,e} = \varphi_s \cdot \mu_{p,l,e}$.

We use mini-batch Stochastic Gradient Descent (SGD) to solve the above optimization problem where the dynamic weights are averaged over a batch of samples. Intuitively, we expect the dynamic-weighting scheme to behave in two different aspects in order to minimize the loss in Equation 8. First, since our main task contribute mostly to the final loss ($\varphi_s < 1$), the side task with the largest contribution to the main task should have the highest weight in order to reduce the loss of the main task. Second, our m-CNN should assign a higher weight for an easier task with a lower loss so as to reduce the overall loss. We have observed these effects as will shown in the experiments.

C. Pose-Directed Multi-Task CNN

It is very challenging to learn a non-linear mapping to estimate the correct identity from a face image with arbitrary PIE, given the diverse variations in the data. This challenge has been encountered in classic pattern recognition work. For example, in order to handle pose variation, [29] proposes to construct several face detectors where each of them is in charge of one specific view. Such a divide-and-conquer scheme can be applied to CNN learning because the side tasks can "divide" the data and allow the CNN to better "conquer" them by learning tailored mapping functions.

Therefore, we propose a novel task-directed multi-task CNN where the side task labels categorize the training data into multiple groups, and direct them to different routes in the network. Since pose is considered as the primary challenge in face recognition [54], [60], [66], we propose pose-directed

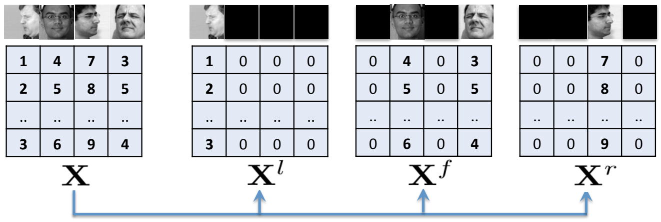


Fig. 3. Illustration of the batch split operation in p-CNN. The first row shows the input images and the second row shows a matrix representing the features $\mathbf{x}$ for each sample. After batch split, one batch of samples is separated into three batches where each of them only consists of the samples belonging to the specific pose group.

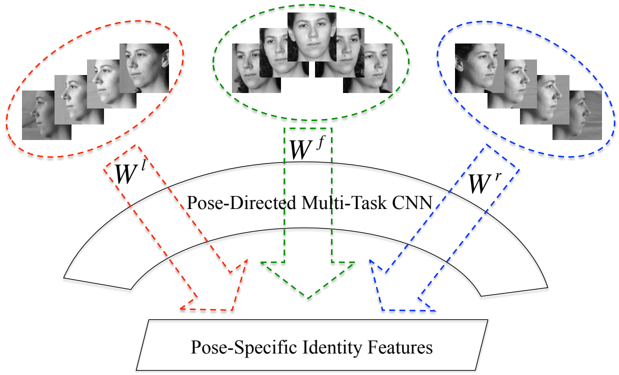


Fig. 4. The proposed pose-directed multi-task CNN aims to learn pose-specific identity features jointly for all pose groups.

multi-task CNN (p-CNN) to handle pose variation. However, it is applicable to other variation.

As shown in Figure 2, p-CNN is built on top of m-CNN by adding the pose-directed branch (PDB). The PDB groups face images with similar poses to learn pose-specific identity features via a batch split operation. We separate the training set into three groups according to the pose labels: left profile (G^l), frontal (G^f), and right profile (G^r). As shown in Figure 3, the goal of the batch split is to separate a batch of N_0 samples ($\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N_0}$) into three batches $\mathbf{X}^l$, $\mathbf{X}^f$, and $\mathbf{X}^r$, which are of the same size as $\mathbf{X}$. During training, the ground truth pose is used to assign a face image into the correct group. Let us take the frontal group as an example to illustrate the batch split operation:

$$\mathbf{X}_i^f = \begin{cases} \mathbf{x}_i, & \text{if } y_i^p \in G^f \\ \mathbf{0}, & \text{otherwise,} \end{cases} \tag{9}$$

where $\mathbf{0}$ denotes a vector of all zeros with the same dimension as $\mathbf{x}_i$. The assignment of $\mathbf{0}$ is to avoid the case when no sample is passed into one group, the next layer will still have valid input. As a result, $\mathbf{X}$ is separated into three batches where each batch consists of only the samples belonging to the corresponding pose group. Each group learns a pose-specific mapping to a joint space, resulting in three different sets of weights: $\{\mathbf{W}^l, \mathbf{W}^f, \mathbf{W}^r\}$, as illustrated in Figure 4. Finally the features from all groups are merged as the input to a softmax layer to perform robust identity classification jointly.

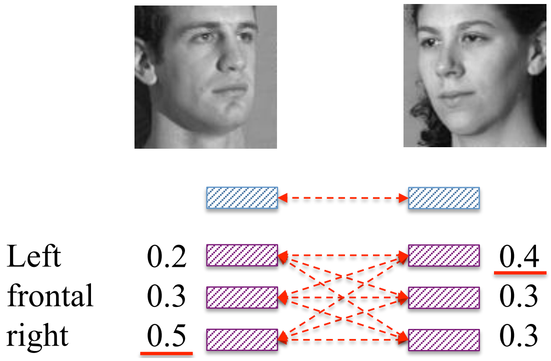


Fig. 5. Blue bars are the generic identity features and purple bars are the pose-specific features. The numbers are the probabilities of each input image belonging to each pose group. The proposed stochastic routing in the testing stage taking account of all pair comparisons so that it is more robust to pose estimation errors.

Our p-CNN aims to learn two types of identity features: $\mathbf{W}^d$ is the weight matrix to extract the generic identity features that is robust to all poses; $\mathbf{W}^{l,f,r}$ are the weight matrices to extract the pose-specific identity features that are robust within a small pose range. Both tasks are considered as our main tasks. Similar to the dynamic-weighting scheme in m-CNN, we use dynamic weights to combine our main tasks as well. The summed loss weight for these two tasks is $\varphi_m = \alpha_d + \alpha_g$. Let $\boldsymbol{\omega}_m \in \mathbb{R}^{D\times 2}$ and $\boldsymbol{\epsilon}_m \in \mathbb{R}^{2\times 1}$ denote the weight matrix and bias vector for learning the dynamic weights,

$$\boldsymbol{\mu}_m = softmax(\boldsymbol{\omega}_m{}^{\intercal}\mathbf{x} + \boldsymbol{\epsilon}_m). \tag{10}$$

We have $\boldsymbol{\mu}_m = [\mu_d, \mu_g]^{\intercal}$ as the dynamic weights for generic identity classification and pose-specific identity classification. Finally, the loss of p-CNN is formulated as:

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{\Theta},\mathbf{W},\boldsymbol{\omega}} \ \varphi_m \Big[\mu_d \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^d) + \mu_g \sum_{g=1}^{G}\sum_{i=1}^{N_g} L(\mathbf{I}_i, y_i^d)\Big] + \\ &\varphi_s \Big[\mu_p \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^p) + \mu_l \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^l) + \mu_e \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^e)\Big] \\ &s.t. \quad \mu_d + \mu_g = 1, \mu_p + \mu_l + \mu_e = 1, \mu_{d,g} \geq 0, \mu_{p,l,e} \geq 0 \end{aligned} \tag{11}$$

where $G = 3$ is the number of pose groups and N_g is the number of training images in the g-th group. $\boldsymbol{\omega} = \{\boldsymbol{\omega}_m, \boldsymbol{\omega}_s\}$ is the set of parameters to learn the dynamic weights for both the main and side tasks. We set $\varphi_m = 1$.

Stochastic Routing Given a face image in the testing stage, we can extract the generic identity features ($\mathbf{y}^d$), the pose-specific identity features ($\{\mathbf{y}^g\}_{g=1}^3$), as well as estimate the probabilities ($\{p^g\}_{g=1}^3$) of the input image belonging to each pose group by aggregating the probabilities from the pose classification side task. As shown in Figure 5, for face matching, we can compute the distance of the generic identity features and the distance of the pose-specific identity features by selecting the pose group with the largest probability (red underline). However, the pose estimation error may cause inferior feature extraction results, which is inevitable especially for unconstrained faces.

To solve this problem, we propose a stochastic routing scheme by taking into account of all comparisons, which is

TABLE II
PERFORMANCE COMPARISON (%) OF SINGLE-TASK LEARNING (S-CNN), MULTI-TASK LEARNING (M-CNN) WITH ITS VARIANTS, AND POSE-DIRECTED MULTI-TASK LEARNING (P-CNN) ON THE ENTIRE MULTI-PIE DATASET.

model	loss weights	rank-1 (all / left / frontal /right)	pose	illum	exp
s-CNN: id	$\alpha_d = 1$	75.67 / 71.51 / 82.21 / 73.29	–	–	–
s-CNN: pos	$\alpha_p = 1$	–	99.87	–	–
s-CNN: exp	$\alpha_l = 1$	–	–	96.43	–
s-CNN: illum	$\alpha_e = 1$	–	–	–	92.44
s-CNN: id+L2	$\alpha_d = 1$	76.43 / 73.31 / 81.98 / 73.99	–	–	–
m-CNN: id+pos	$\alpha_d = 1, \alpha_p = 0.1$	78.06 / 75.06 / 82.91 / 76.21	99.78	–	–
m-CNN: id+illum	$\alpha_d = 1, \alpha_l = 0.1$	77.30 / 74.87 / 82.83 / 74.21	–	93.57	–
m-CNN: id+exp	$\alpha_d = 1, \alpha_e = 0.1$	77.76 / 75.48 / 82.32 / 75.48	–	–	90.93
m-CNN: id+all	$\alpha_d = 1, \alpha_{p,l,e} = 0.033$	77.59 / 74.75 / 82.99 / 75.04	99.75	88.46	79.97
m-CNN: id+all (dynamic)	$\alpha_d = 1, \varphi_s = 0.1$	79.35 / 76.60 / 84.65 / 76.82	99.81	93.40	91.47
p-CNN	$\varphi_m = 1, \varphi_s = 0.1$	79.55 / 76.14 / 84.87 / 77.65	99.80	90.58	90.02

more robust to pose estimation errors. Specifically, the distance c between a pair of face images ($\mathbf{I}_1$ and $\mathbf{I}_2$) is computed as the average between the distance of the generic identity features ($\mathbf{y}_1^d$, $\mathbf{y}_2^d$) and weighted distance of the pose-specific identity features ($\{\mathbf{y}_1^g\}$, $\{\mathbf{y}_2^g\}$):

$$c = \frac{1}{2} h(\mathbf{y}_1^d, \mathbf{y}_2^d) + \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} h(\mathbf{y}_1^i, \mathbf{y}_2^j) \cdot p_1^i \cdot p_2^j, \qquad (12)$$

where $h(\cdot)$ is the cosine distance metric used to measure the distance between two feature vectors. The proposed stochastic routing accounts for all combinations of the pose-specific identity features weighted by the probabilities of each combination. We treat the generic features and pose-specific features equally, and fuse them for face recognition.

IV. EXPERIMENTS

We evaluate the proposed m-CNN and p-CNN under two settings: (1) face identification on Multi-PIE with PIE estimations being the side tasks; (2) face verification/identification on in-the-wild datasets including LFW, CFP, and IJB-A, where pose estimation is the only side task. Further, we analyze the effect of MTL on Multi-PIE and discover that the side tasks regularize the network to learn a disentangled identity representation for PIE-invariant face recognition.

A. Face Identification on Multi-PIE

Experimental Settings The Multi-PIE dataset consists of $754,200$ images of 337 subjects recorded in 4 sessions. Each subject was recorded with 15 different cameras where 13 are at the head height spaced at $15°$ interval and 2 are above the head. For each camera, a subject was imaged under 19 different illuminations. In each session, a subject was captured with 2 or 3 expressions, resulting in a total of 6 different expressions across all sessions.

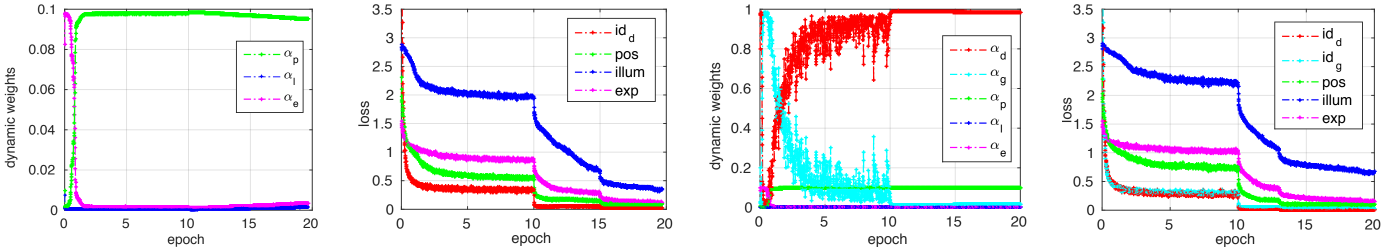
All previous work [10], [18], [12], [17], [25], [28], [54], [55], [60], [64], [66] studies face recognition on a subset of PIE variations on Multi-PIE. In our work, we use the entire dataset including all PIE variations. For the two cameras above the head, their poses are labeled as $\pm 45°$. The first 200 subjects are used for training. The remaining 137 subjects are used for testing, where one image with frontal pose, neutral illumination, and neutral expression for each subject is selected as the gallery set and the remaining as the probe set.

We use the landmark annotations [13] to align each face to a canonical view of size 100×100. The images are normalized by subtracting 127.5 and dividing by 128, similar to [53]. We use Caffe [23] with our modifications. The momentum is set to 0.9 and the weight decay to 0.0005. All models are trained for 20 epochs from scratch with a batch size of 4 unless specified. The learning rate starts at 0.01 and reduces at 10th, 15th, and 19th epochs with a factor of 0.1. The rank-1 identification rate is reported as the face recognition performance. For the side tasks, the mean accuracy over all classes is reported.

We randomly select 20 subjects from the training set to form a validation set to find the optimal overall loss weight for all side tasks. We obtain $\varphi_s = 0.1$ via brute-force search. For p-CNN model training, we split the training set into three groups based on the yaw angle of the image: right profile ($-90°, -75°, -60°, -45°$), frontal ($-30°, -15°, 0°, 15°, 30°$), and left profile ($45°, 60°, 75°, 90°$).

Effects of MTL Table II shows the performance comparison of single-task learning (s-CNN), multi-task learning (m-CNN), and pose-directed multi-task learning (p-CNN) on the entire Multi-PIE. First, we train four single-task models for identity (id), pose (pos), illumination (illum), and expression (exp) classification respectively. As shown in Table II, the rank-1 identification rate of s-CNN is only 75.67%. The performance of the frontal pose group is much higher than those of the profile pose groups, indicating that pose variation is indeed a big challenge for face recognition. Among all side tasks, pose estimation is the easiest task, followed by illumination, and expression as the most difficult one. This is caused by two potential reasons: 1) discriminating expression is more challenging due to the non-rigid face deformation; 2) the data distribution over different expressions is unbalanced with insufficient training data for some expressions.

Second, we train multiple m-CNN models by adding only one side task at a time in order to evaluate the influence of each side task. We use "id+pos", "id+illum", and "id+exp" to represent these variants and compare them to the performance of adding all side tasks denoted as "id+all". To evaluate the effects of the dynamic-weighting scheme, we train a model



(a) m-CNN: dynamic weights (b) m-CNN: losses (c) p-CNN: dynamic weights (d) p-CNN: losses

Fig. 6. The learnt dynamic weights and the losses of each task for m-CNN and p-CNN models during the training process.

TABLE III
MULTI-PIE PERFORMANCE COMPARISON ON SETTING III OF TABLE I.

	±15°	±30°	±45°	±60°	±75°	±90°	avg.
Fisher Vector [43]	93.30	87.21	80.33	68.71	45.51	24.53	66.60
FIP_20 [65]	95.88	89.23	78.89	61.64	47.32	34.13	67.87
FIP_40 [65]	96.30	92.98	85.54	69.75	49.10	31.37	70.90
c-CNN [54]	95.64	92.66	85.09	70.49	55.64	41.71	73.54
c-CNN Forest [54]	96.97	94.05	89.02	74.38	60.66	47.26	76.89
s-CNN (ours)	98.41	96.89	85.18	88.71	82.80	76.72	88.45
m-CNN (ours)	99.02	97.40	89.15	89.75	84.97	76.72	90.08
p-CNN (ours)	**99.19**	**98.01**	**90.34**	**92.07**	**87.83**	**76.96**	**91.27**

TABLE IV
MULTI-PIE PERFORMANCE COMPARISON ON SETTING V OF TABLE I.

	0°	±15°	±30°	±45°	±60°	±75°	±90°	avg.[−60°, 60°]	avg.[−90°, 90°]
FIP [65]	94.3	90.7	80.7	64.1	45.9	–	–	72.9	–
Zhu et al. [66]	95.7	92.8	83.7	72.9	60.1	–	–	79.3	–
Yim et al. [57]	**99.5**	95.0	88.5	79.9	61.9	–	–	83.3	–
DR-GAN [51]	97.0	94.0	90.1	86.2	83.2	–	–	89.2	–
FF-GAN [58]	95.7	94.6	92.5	89.7	85.2	77.2	61.2	91.6	85.2
s-CNN (ours)	95.9	95.1	92.8	91.6	88.9	84.9	78.6	92.5	89.2
m-CNN (ours)	95.4	94.5	92.6	91.8	88.4	85.3	82.2	92.2	89.6
p-CNN (ours)	95.4	**95.2**	**94.3**	**93.0**	**90.3**	**87.5**	**83.9**	**93.5**	**91.1**

with fixed loss weights for the side tasks as: $\alpha_p = \alpha_l = \alpha_e = \varphi_s/3 = 0.033$. The summation of the loss weights for all side tasks are equal to φ_s for all m-CNN variants in Table II for a fair comparison.

Comparing the rank-1 identification rates of s-CNN and m-CNNs, it is obvious that adding the side tasks is always helpful for the main task. The improvement of face recognition is mostly on the profile faces with MTL. The m-CNN "id+all" with dynamic weights shows superior performance to others not only in rank-1 identification rate, but also in the side task estimations. Further, the lower rank-1 identification rate of "id+all" w.r.t "id+pos" indicates that more side tasks do not necessarily lead to better performance without properly setting the loss weights. In contrast, the proposed dynamic-weighting scheme effectively improves the performance to 79.35% from the fixed weighting of 77.59%. As will be shown in Section IV-B, the side tasks in m-CNN help to inject PIE variations into the shared representation, similar to a regularization term. For example, an L2 regularization will encourage small weights. We add L2 regularization on the shared representation to s-CNN ("id+L2"), which improves over s-CNN without regularization. However it is still much worse than the proposed m-CNN.

Third, we train p-CNN by adding the PDB to m-CNN "id+all" with dynamic weights. The loss weights are $\varphi_m = 1$ for the main tasks and $\varphi_s = 0.1$ for the side tasks. The proposed dynamic-weighting scheme allocates the loss weights to both two main tasks and three side tasks. P-CNN further improves the rank-1 identification rate to 79.55%.

Dynamic-Weighting Scheme Figure 6 shows the dynamic weights and losses during training for m-CNN and p-CNN. For m-CNN, the expression classification task has the largest weight in the first epoch because it has the highest chance to be correct with random guess with the least number of classes. As training goes on, pose classification takes over because it is the easiest task (highest accuracy in s-CNN) and also the most helpful for face recognition (compare "id+pos" to "id+exp" and "id+illum"). α_p starts to decrease at the 11th epoch when pose classification is saturated. The increased α_l and α_e lead to a reduction in the losses of expression and illumination classifications. As we expected, the dynamic-weighting scheme assigns a higher loss weight for the easiest and/or the most helpful side task.

For p-CNN, the loss weights and losses for the side tasks behave similarly to those of m-CNN. For the two main tasks, the dynamic-weighting scheme assigns a higher loss weight to

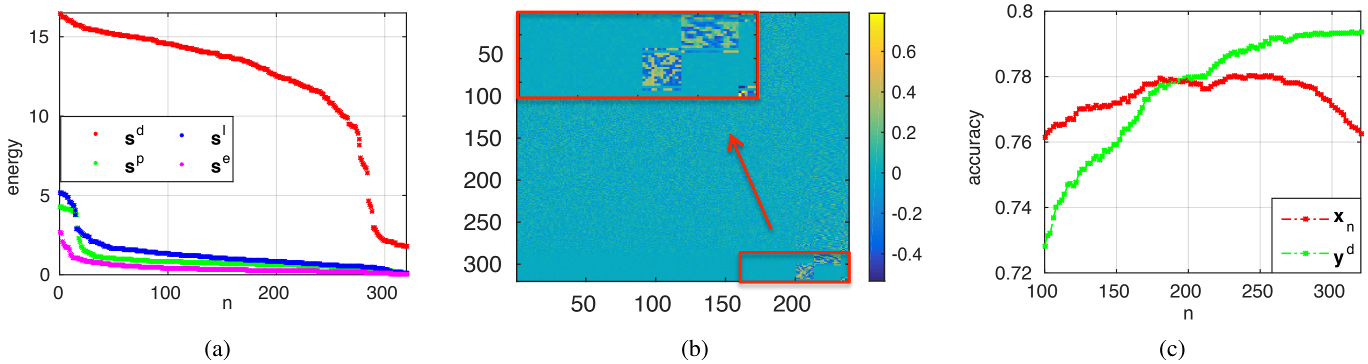


Fig. 7. Analysis on the effects of MTL: (a) the sorted energy vectors for all tasks; (b) visualization of the weight matrix $\mathbf{W}^{all}$ where the red box in the top-left is a zoom-in view of the bottom-right; (c) the face recognition performance with varying feature dimensions.

the easier task at the moment. At the beginning, learning the pose-specific identity features is an easier task than learning the generic identity features. Therefore the loss weight α_g is higher than α_d. As training goes on, α_d increases as it has a lower loss. Their losses reduce in a similar way, i.e., the error reduction in one task will also contribute to the other.

Compare to Other Methods As shown in Table I, no prior work uses the entire Multi-PIE for face recognition. To compare with state of the art, we choose to use setting III and V to evaluate our method since these are the most challenging settings with more pose variation. The network structures and parameter settings are kept the same as those of the full set except that the outputs of the last fully connected layers are changed according to the number of classes for each task. Only pose and illumination are used as the side tasks.

The performance on setting III is shown in Table III. Our s-CNN already outperforms c-CNN forest [54], which is an ensemble of three c-CNN models. This is attributed to the deep structure of CASIA-Net [56]. Moreover, m-CNN and p-CNN further outperform s-CNN with significant margins, especially for non-frontal faces. We want to stress the improvement margin between our method 91.27% and the prior work of 76.89% — a relative error reduction of 62%.

The performance on setting V is shown in Table IV. For fair comparison with FF-GAN [58], where the models are finetuned from pre-trained in-the-wild models, we also finetune s-CNN, m-CNN, p-CNN models from the pre-trained models on CASIA-Webface for 10 epochs. Our performance is much better than previous work with a relative error reduction of 60%, especially on large-pose faces. The performance gap between Table III / IV and II indicates the challenge of face recognition under various expressions, which is less studied than pose and illumination variations on Multi-PIE.

B. How does m-CNN work?

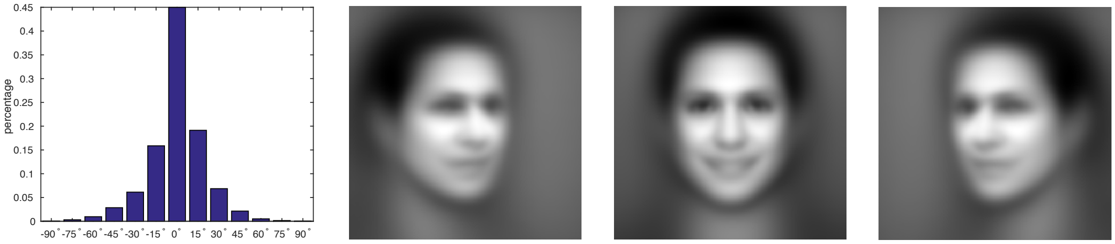
It is well known in both the computer vision and the machine learning communities that learning multiple tasks together allows each task to leverage each other and improves the generalization ability of the model. For CNN-based MTL, previous work [63] has found that CNN learns shared features for facial landmark localization and attribute classifications, e.g. smiling. This is understandable because the smiling attribute is related to landmark localization as it involves the change of the mouth region. However in our case, it is not obvious how the PIE estimations can share features with the main task. On the contrary, it is more desirable if the learnt identity features are disentangled from the PIE variations. Indeed, as we will show later, the PIE estimations regularize the CNN to learn PIE-invariant identity features.

We investigate why PIE estimations are helpful for face recognition. The analysis is done on m-CNN model ("id+all" with dynamic weights) in Table II. Recall that m-CNN learns a shared embeding $\mathbf{x} \in \mathbb{R}^{320\times1}$. Four fully connected layers with weight matrices $\mathbf{W}^d_{320\times200}$, $\mathbf{W}^p_{320\times13}$, $\mathbf{W}^l_{320\times19}$, $\mathbf{W}^e_{320\times6}$ are connected to $\mathbf{x}$ to perform classification of each task (200 subjects, 13 poses, 19 illuminations, and 6 expressions). We analyze the importance of each dimension in $\mathbf{x}$ to each task. Taking the main task as an example, we calculate an energy vector $\mathbf{s}^d \in \mathbb{R}^{320\times1}$ whose element is computed as:

$$\mathbf{s}^d_i = \sum_{j=1}^{200} \mid \mathbf{W}^d_{ij} \mid . \tag{13}$$

A higher value of $\mathbf{s}^d_i$ indicates that the ith feature in $\mathbf{x}$ is more important to the identity classification task. The energy vectors $\mathbf{s}^p$, $\mathbf{s}^l$, $\mathbf{s}^e$ for all side tasks are computed similarly. Each energy vector is sorted and shown in Figure 7 (a). For each curve, we observe that the energy distributes unevenly among all feature dimensions in $\mathbf{x}$. Note that the indexes of the feature dimension do not correspond among them since each energy vector is sorted independently.

To compare how each feature in $\mathbf{x}$ contributes to different tasks, we concatenate the weight matrix of all tasks as $\mathbf{W}^{all}_{320\times238} = [\mathbf{W}^d, \mathbf{W}^p, \mathbf{W}^l, \mathbf{W}^e]$ and compute its energy vector as $\mathbf{s}^{all}$. We sort the rows in $\mathbf{W}^{all}$ based on the descending order in energy and visualize the sorted $\mathbf{W}^{all}$ in Figure 7 (b). The first 200 columns represent the sorted $\mathbf{W}^d$ where most energy is distributed in the first $\sim$ 280 feature dimensions (rows), which are more crucial for face recognition and less important for PIE classifications. We observe that $\mathbf{x}$ are learnt to allocate a separate set of dimensions/features for each task, as shown in the block-wise effect in the zoom-in view. Each block shows the most essential features with high energy for PIE classifications respectively.

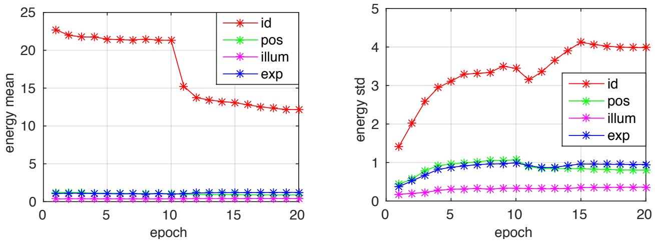


(a) (b) (c) (d)

Fig. 10. (a) yaw angle distribution on CASIA-Webface; (b) average image of the right profile group; (c) average image of the frontal group; (d) average image of the left profile group.

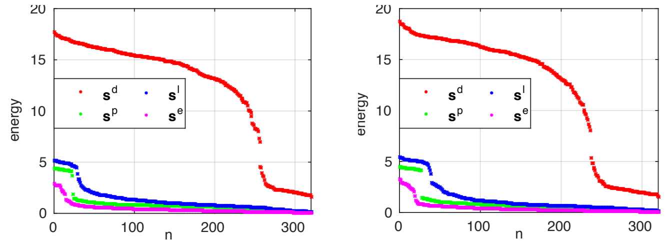
TABLE V
PERFORMANCE COMPARISON ON LFW DATASET.

Method	#Net	Training Set	Metric	Accuracy ± Std (%)
DeepID2 [45]	1	$202,599$ images of $10,177$ subjects, private	Joint-Bayes	95.43
DeepFace [49]	1	4.4M images of $4,030$ subjects, private	cosine	95.92 ± 0.29
CASIANet [56]	1	$494,414$ images of $10,575$ subjects, public	cosine	96.13 ± 0.30
Wang et al. [52]	1	$404,992$ images of $10,553$ subjects, public	Joint-Bayes	96.2 ± 0.9
Littwin and Wolf [32]	1	$404,992$ images of $10,553$ subjects, public	Joint-Bayes	98.14 ± 0.19
MultiBatch [48]	1	2.6M images of 12K subjects, private	Euclidean	98.20
VGG-DeepFace [38]	1	2.6M images of $2,622$ subjects, public	Euclidean	98.95
Wen et al. [53]	1	0.7M images of $17,189$ subjects, public	cosine	99.28
FaceNet [41]	1	260M images of 8M subjects, private	L2	99.63 ± 0.09
s-CNN (ours)	1	$494,414$ images of $10,575$ subjects, public	cosine	97.87 ± 0.70
m-CNN (ours)	1	$494,414$ images of $10,575$ subjects, public	cosine	98.07 ± 0.57
p-CNN (ours)	1	$494,414$ images of $10,575$ subjects, public	cosine	98.27 ± 0.64



(a) mean (b) std.

Fig. 8. The mean and standard deviation of each energy vector during the training process.



(a) $\varphi_s = 0.2$ (b) $\varphi_s = 0.3$

Fig. 9. Energy vectors of m-CNN models with different overall loss weights.

Based on the above observation, we conclude that the PIE classification side tasks help to inject PIE variations into the shared features $\mathbf{x}$. The weight matrix in the fully connected layer learns to select identity features and ignore the PIE features for PIE-invariant face recognition. To validate this observation quantitatively, we compare two types of features for face recognition: 1) $\mathbf{x}_n$: a subset of $\mathbf{x}$ with n largest energies in $\mathbf{s}^d$, which are more crucial in modeling identity variation; 2) $\mathbf{y}^d_{200\times 1} = {\mathbf{W}^d_{n\times 200}}^\intercal \mathbf{x}_{n\times 1} + \mathbf{b}^d$, which is the multiplication of the corresponding subset of $\mathbf{W}^d$ and $\mathbf{x}_n$. We vary n from 100 to 320 and compute the rank-1 face identification rate on the entire Multi-PIE testing set. The performance is shown in Figure 7 (c). When $\mathbf{x}_n$ is used, the performance improves with increasing dimensions and drops when additional dimensions are included, which are learnt to model the PIE variations. In contrary, the identity features $\mathbf{y}^d$ can eliminate the dimensions that are not helpful for identity classification through the weight matrix $\mathbf{W}^d$, resulting in continuously improved performance w.r.t. n.

We further analyze how the energy vectors evolve over time during training. Specifically, at each epoch, we compute the energy vectors for each task. Then we compute the mean and standard deviation of each energy vector, as shown in Figure 8. Despite some local fluctuations, the overall trend is that the mean is decreasing and standard deviation is increasing as training goes on. This is because in the early stage of training, the energy vectors are more evenly distributed among all feature dimensions, which leads to the higher mean values and lower standard deviations. In the later stage of training, the energy vectors are shaped in a way to focus on some key dimensions for each task, which leads to the lower mean values and higher standard deviations.

The CNN learns to allocate a separate set of dimensions in the shared features to each task. The total number of dimensions assigned to each task depends on the loss weights. Recall that we obtain the overall loss weight for the side tasks as $\varphi_s = 0.1$ via brute-force search. Figure 9 shows the energy distributions with $\varphi_s = 0.2$ and $\varphi_s = 0.3$, which are compared to Figure 7 (a) where $\varphi_s = 0.1$. We have two observations. First, a larger loss weight for the side tasks leads to more dimensions being assigned to the side tasks. Second, the energies in $\mathbf{s}^d$ increase in order to compensate the fact that the dimensions assigned to the main task decrease. Therefore, we conclude that the loss weights control the energy distribution between different tasks.

C. Unconstrained Face Recognition

Experimental Settings We use CASIA-Webface [56] as our training set and evaluate on LFW [20], CFP [42], and IJB-A [26] datasets. CASIA-Webface consists of $494,414$ images of $10,575$ subjects. LFW consists of 10 folders each with 300 same-person pairs and 300 different-person pairs. Given the saturated performance of LFW mainly due to its mostly frontal view faces, CFP and IJB-A are introduced for large-pose face recognition. CFP is composed of 500 subjects with 10 frontal and 4 profile images for each subject. Similar to LFW, CFP includes 10 folders, each with 350 same-person pairs and 350 different-person pairs, for both frontal-frontal (FF) and frontal-profile (FP) verification protocols. IJB-A dataset includes $5,396$ images and $20,412$ video frames of 500 subjects. It defines template-to-template matching for both face verification and identification.

In order to apply the proposed m-CNN and p-CNN, we need to have the labels for the side tasks. However, it is not easy to manually label our training set. Instead, we only consider pose estimation as the side task and use the estimated pose as the label for training. We use PIFA [24] to estimate 34 landmarks and the yaw angle, which defines three groups: right profile $[-90^\circ, -30^\circ)$, frontal $[-30^\circ, 30^\circ]$, and left profile $(30^\circ, 90^\circ]$. Figure 10 shows the distribution of the yaw angle estimation and the average image of each pose group. CASIA-Webface is biased towards frontal faces with 88% faces belonging to the frontal pose group based on our pose estimation.

The network structures are similar to those experiments on Multi-PIE. All models are trained from scratch for 15 epochs with a batch size of 8. The initial learning rate is set to 0.01 and reduced at the 10th and 14th epoch with a factor of 0.1. The other parameter settings and training process are the same as those on Multi-PIE. We use the same pre-processing as in [56] to align a face image. Each image is horizontally flipped for data augmentation in the training set. We also generate the mirror image of an input face in the testing stage. We use the average cosine distance of all four comparisons between the image pair and its mirror images for face recognition.

Performance on LFW Table V compares our face verification performance with state-of-the-art methods on LFW dataset. We follow the unrestricted with labeled outside data protocol. Although it is well-known that an ensemble of multiple networks can improve the performance [46], [47], we only compare CNN-based methods with one network for fair comparison. Our implementation of the CASIA-Net (s-CNN) with BN achieves much better results compared to the original performance [56]. Even with such a high baseline, m-CNN and p-CNN can still improve, achieving comparable results with state of the art, or better results if comparing to those methods trained with the same amount of data. Since LFW is biased towards frontal faces, we expect the improvement of our proposed m-CNN and p-CNN to the baseline s-CNN to be larger if they are tested on cross-pose face verification.

Performance on CFP Table VI shows our face verification performance comparison with state-of-the-art methods on CFP dataset. For FF setting, m-CNN and p-CNN improve the verification rate of s-CNN slightly. This is expected, as there is little pose variation. For FP setting, p-CNN substantially outperforms s-CNN and prior work, reaching close-to-human performance (94.57%). Note our accuracy of 94.39% is 9% relative error reduction of the previous state of the art [39] with 93.76%. Therefore, the proposed divide-and-conquer scheme is very effective for in-the-wild face verification with large pose variation. And the proposed stochastic routing scheme improves the robustness of the algorithm. Even with the estimated pose serving as the ground truth pose label for MTL, the models can still disentangle the pose variation from the learnt identity features for pose-invariant face verification.

Performance on IJB-A We conduct close-set face identification and face verification on IJB-A dataset. First, we retrain our models after removing 26 overlapped subjects between CASIA-Webface and IJB-A. Second, we fine-tune the retrained models on the IJB-A training set of each fold for 50 epochs. Similar to [52], we separate all images into "well-aligned" and "poorly-aligned" faces based on the face alignment results and the provided annotations. In the testing stage, we only select images from the "well-aligned" faces for recognition. If all images in a template are "poorly-aligned" faces, we select the best aligned face among them. Table VII shows the performance comparison on IJB-A. Similarly, we only compare to the methods with a single model. The proposed p-CNN achieves comparable performance in both face verification and identification.

V. Conclusions

This paper explores multi-task learning for face recognition with PIE estimations as the side tasks. To solve the problem of balancing each task in MTL, we propose a dynamic-weighting scheme to automatically assign the loss weights to each side task during the training process. This scheme is shown to assign a larger loss weight to an easier side task and/or the most helpful side task. We also propose a pose-directed multi-task CNN to learn pose-specific identity features during training and a stochastic routing scheme for feature fusion in the testing stage. A comprehensive study on the entire Multi-PIE dataset has shown the effectiveness of the proposed approach for PIE-invariant face recognition. An in-depth weight matrix analysis has shown why PIE estimations can help face recognition to learn a disentangled representation.

The proposed method is applicable to in-the-wild datasets with estimated pose serving as the label for training. However, we do not see large improvement on LFW and IJB-A as that on Multi-PIE. This may due to several factors. First, both m-CNN and p-CNN rely on the pose estimation, which is limited

TABLE VI
PERFORMANCE COMPARISON ON CFP DATASET. RESULTS REPORTED ARE THE AVERAGE ± STANDARD DEVIATION OVER THE 10 FOLDS.

Method ↓	Frontal-Frontal			Frontal-Profile		
Metric (%) →	Accuracy	EER	AUC	Accuracy	EER	AUC
Sengupta et al. [42]	96.40 ± 0.69	3.48 ± 0.67	99.43 ± 0.31	84.91 ± 1.82	14.97 ± 1.98	93.00 ± 1.55
Sankarana. et al. [40]	96.93 ± 0.61	2.51 ± 0.81	99.68 ± 0.16	89.17 ± 2.35	8.85 ± 0.99	97.00 ± 0.53
Chen, et al. [9]	**98.67** ± 0.36	**1.40** ± 0.37	**99.90** ± 0.09	91.97 ± 1.70	8.00 ± 1.68	97.70 ± 0.82
DR-GAN [51]	97.84 ± 0.79	2.22 ± 0.09	99.72 ± 0.02	93.41 ± 1.17	6.45 ± 0.16	97.96 ± 0.06
Peng, et al. [39]	**98.67**	–	–	93.76	–	–
Human	96.24 ± 0.67	5.34 ± 1.79	98.19 ± 1.13	94.57 ± 1.10	5.02 ± 1.07	98.92 ± 0.46
s-CNN (ours)	97.34 ± 0.99	2.49 ± 0.09	99.69 ± 0.02	90.96 ± 1.31	8.79 ± 0.17	96.90 ± 0.08
m-CNN (ours)	97.77 ± 0.39	2.31 ± 0.06	99.69 ± 0.02	91.39 ± 1.28	8.80 ± 0.17	97.04 ± 0.08
p-CNN (ours)	97.79 ± 0.40	2.48 ± 0.07	99.71 ± 0.02	**94.39** ± 1.17	**5.94** ± 0.11	**98.36** ± 0.05

TABLE VII
PERFORMANCE COMPARISON ON IJB-A.

Method ↓	Verification		Identification	
Metric (%) →	@FAR=0.01	@FAR=0.001	@Rank-1	@Rank-5
OpenBR [26]	23.6 ± 0.9	10.4 ± 1.4	24.6 ± 1.1	37.5 ± 0.8
GOTS [26]	40.6 ± 1.4	19.8 ± 0.8	44.3 ± 2.1	59.5 ± 2.0
Wang et al. [52]	72.9 ± 3.5	51.0 ± 6.1	82.2 ± 2.3	93.1 ± 1.4
PAM [34]	73.3 ± 1.8	**55.2** ± 3.2	77.1 ± 1.6	88.7 ± 0.9
DR-GAN [51]	77.4 ± 2.7	53.9 ± 4.3	85.5 ± 1.5	**94.7** ± 1.1
DCNN [8]	**78.7** ± 4.3	–	85.2 ± 1.8	93.7 ± 1.0
s-CNN (ours)	75.6 ± 3.5	52.0 ± 7.0	84.3 ± 1.3	93.0 ± 0.9
m-CNN (ours)	75.6 ± 2.8	51.6 ± 4.5	84.7 ± 1.0	93.4 ± 0.7
p-CNN (ours)	77.5 ± 2.5	53.9 ± 4.2	**85.8** ± 1.4	93.8 ± 0.9

by the state-of-the-art pose estimation methods. Second, a large training set might diminishes the benefits of multi-task learning for unconstrained face recognition. Third, both LFW and IJB-A have large variations other than pose such as expression, blurring, etc. that cannot be well handled by the proposed method. Nevertheless, for dataset like CFP where pose variation is the major variation, we achieve state-of-the-art performance on the frontal-to-profile verification protocol.

REFERENCES

[1] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia. Multi-task CNN model for attribute prediction. *TMM*, 2015.

[2] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *ICML*, 2013.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 2008.

[4] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith. Fully automatic pose-invariant face recognition via 3D pose normalization. In *ICCV*, 2011.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. *SIGGRAPH*, 1999.

[6] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *CVPR*, 2013.

[7] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *ECCV*, 2014.

[8] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *WACV*, 2016.

[9] J.-C. Chen, J. Zheng, V. M. Patel, and R. Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. In *ICIP*, 2016.

[10] B. Chu, S. Romdhani, and L. Chen. 3D-aided face recognition robust to expression and pose variations. In *CVPR*, 2014.

[11] C. Ding and D. Tao. A comprehensive survey on pose-invariant face recognition. *ACM Transactions on intelligent systems and technology (TIST)*, 2016.

[12] C. Ding, C. Xu, and D. Tao. Multi-task pose-invariant face recognition. *TIP*, 2015.

[13] L. El Shafey, C. McCool, R. Wallace, and S. Marcel. A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition. *TPAMI*, 2013.

[14] H. Fei and J. Huan. Structured feature selection and task relationship inference for multi-task learning. *Knowledge and information systems*, 2013.

[15] P. Gong, J. Zhou, W. Fan, and J. Ye. Efficient multi-task feature learning with calibration. In *ICKDDM*, 2014.

[16] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *IVC*, 2010.

[17] H. Han, S. Shan, X. Chen, S. Lao, and W. Gao. Separability oriented preprocessing for illumination-insensitive face recognition. In *ECCV*, 2012.

[18] H. Han, S. Shan, L. Qing, X. Chen, and W. Gao. Lighting aware preprocessing for face recognition across varying illumination. In *ECCV*, 2010.

[19] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *CVPR*, 2015.

[20] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint:1502.03167*, 2015.

[22] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, 2009.

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMMM*, 2014.

[24] A. Jourabloo and X. Liu. Large-pose face alignment via CNN-based dense 3D model fitting. In *CVPR*, 2016.

[25] M. Kan, S. Shan, H. Chang, and X. Chen. Stacked progressive auto-encoders (SPAE) for face recognition across poses. In *CVPR*, 2014.

[26] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *CVPR*, 2015.

[27] A. Li, S. Shan, X. Chen, and W. Gao. Maximizing intra-individual correlations for face recognition across pose differences. In *CVPR*, 2009.

[28] S. Li, X. Liu, X. Chai, H. Zhang, S. Lao, and S. Shan. Morphable displacement field based image matching for face recognition across

pose. In *ECCV*, 2012.

[29] Y. Li, B. Zhang, S. Shan, X. Chen, and W. Gao. Bagging based efficient kernel fisher discriminant analysis for face recognition. In *ICPR*, 2006.

[30] K. Lin, J. Xu, I. M. Baytas, S. Ji, and J. Zhou. Multi-task feature interaction learning. In *ICKDDM*, 2016.

[31] K. Lin and J. Zhou. Interactive multi-task relationship learning. In *ICDM*, 2016.

[32] E. Littwin and L. Wolf. The multiverse loss for robust transfer learning. In *CVPR*, 2016.

[33] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[34] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *CVPR*, 2016.

[35] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

[36] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[37] G. Obozinski, B. Taskar, and M. Jordan. Multi-task feature selection. *Technical Report, Statistics Department, UC Berkeley*, 2006.

[38] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[39] X. Peng, X. Yu, K. Sohn, D. N. Metaxas, and M. Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. In *ICCV*, 2017.

[40] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. *arXiv preprint:1604.05417*, 2016.

[41] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[42] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016.

[43] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *BMVC*, 2013.

[44] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016.

[45] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.

[46] Y. Sun, D. Liang, X. Wang, and X. Tang. DeepID3: Face recognition with very deep neural networks. *arXiv preprint:1502.00873*, 2015.

[47] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2015.

[48] O. Tadmor, Y. Wexler, T. Rosenwein, S. Shalev-Shwartz, and A. Shashua. Learning a metric embedding for face recognition using the multibatch method. In *NIPS*, 2016.

[49] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[50] Y. Tian, P. Luo, X. Wang, and X. Tang. Pedestrian detection aided by deep learning semantic tasks. In *CVPR*, 2015.

[51] L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *CVPR*, 2017.

[52] D. Wang, C. Otto, and A. K. Jain. Face search at scale. *TPAMI*, 2016.

[53] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[54] C. Xiong, X. Zhao, D. Tang, K. Jayashree, S. Yan, and T.-K. Kim. Conditional convolutional neural network for modality-aware face recognition. In *ICCV*, 2015.

[55] M. Yang, L. Gool, and L. Zhang. Sparse variation dictionary learning for face recognition with a single training sample per person. In *ICCV*, 2013.

[56] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint:1411.7923*, 2014.

[57] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim. Rotating your face using multi-task deep neural network. In *CVPR*, 2015.

[58] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In *ICCV*, 2017.

[59] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *WACV*, 2014.

[60] H. Zhang, Y. Zhang, and T. S. Huang. Pose-robust face recognition via sparse representation. *Pattern Recognition*, 2013.

[61] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint:1203.3536*, 2012.

[62] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014.

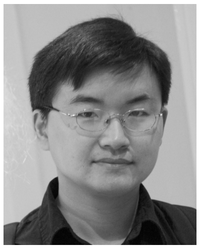
[63] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *TPAMI*, 2016.

[64] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *CVPR*, 2015.

[65] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *ICCV*, 2013.

[66] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: A deep model for learning face identity and view representations. In *NIPS*, 2014.

Xi Yin received the B.S. degree in Electronic and Information Science from Wuhan University, China, in 2013. Since August 2013, she has been working toward her Ph.D. degree in the Department of Computer Science and Engineering, Michigan State University, USA. Her research area are computer vision, deep learning, and image processing. Her paper on multi-leaf segmentation won the Best Student Paper Award at Winter Conference on Application of Computer Vision (WACV) 2014.



Xiaoming Liu is an Assistant Professor at the Department of Computer Science and Engineering of Michigan State University. He received the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2004. Before joining MSU in Fall 2012, he was a research scientist at General Electric (GE) Global Research. His research interests include computer vision, machine learning, and biometrics. As a co-author, he is a recipient of Best Industry Related Paper Award runner-up at ICPR 2014, Best Student Paper Award at WACV 2012 and 2014, and Best Poster Award at BMVC 2015. He has been the Area Chair for numerous conferences, including FG, ICPR, WACV, ICIP, and CVPR. He is the program chair of WACV 2018. He is an Associate Editor of Neurocomputing journal. He has authored more than 100 scientific publications, and has filed 22 U.S. patents.