Shape Driven Kernel Adaptation in Convolutional Neural Network for Robust Facial Trait Recognition

Shaoxin Li[1,3], Junliang Xing[2,3], Zhiheng Niu[3], Shiguang Shan[1], Shuicheng Yan[3]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, 100190, China
[3]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

Abstract

One key challenge of facial trait recognition is the large non-rigid appearance variations due to some irrelevant real world factors, such as viewpoint and expression changes. In this paper, we explore how the shape information, i.e. facial landmark positions, can be explicitly deployed into the popular Convolutional Neural Network (CNN) architecture to disentangle such irrelevant non-rigid appearance variations. First, instead of using fixed kernels, we propose a kernel adaptation method to dynamically determine the convolutional kernels according to the spatial distribution of facial landmarks, which helps learning more robust features. Second, motivated by the intuition that different local facial regions may demand different adaptation functions, we further propose a tree-structured convolutional architecture to hierarchically fuse multiple local adaptive CNN subnetworks. Comprehensive experiments on WebFace, Morph II and MultiPIE databases well validate the effectiveness of the proposed kernel adaptation method and tree-structured convolutional architecture for facial trait recognition tasks, including identity, age and gender recognition. For all the tasks, the proposed architecture consistently achieves the state-of-the-art performances.

1. Introduction

In the last decade, great progress has been made in developing deep neural network for various computer vision tasks [11, 31]. Among them, Convolutional Neural Network (CNN) [16] has achieved exciting performance on digit recognition [13, 5], traffic sign recognition [27, 5], object recognition [5, 15, 24, 31] and scene labeling [6].

Despite the great success, CNN based methods learn discriminant features mainly from texture information which may change significantly in real world conditions due to illumination and viewpoints variations. Although the deep CNN model is proven to be very powerful in handling these complex real world factors in image processing [35, 31], additional information that provide more direct and easier way for parsing these mixed factors may further improve the disentangling capability. Shape information, i.e. the configuration of sub-level components, is one of such additional information. As indicated in [25], for humans, both texture and shape information play very important roles in interpreting face images. Because shape information not only provides easier way to disentangle viewpoints and expression variation but also directly holds discriminant power for estimating facial traits, such as gender, age and identity. It is thus expected that artificial neural network, e.g. neurobiologically motivated CNN, can also benefit from additional shape information for robust facial trait recognition. However, up to now, there are only a few works on exploring the shape information for CNN. One previous attempt [21] integrated the shape information (i.e. sub-level components configuration) as a regularization in the feature learning procedure for pedestrian detection. Some other works [29, 36] used shape information obtained from poselet or landmark detection results to extract better aligned image patches for robust feature learning.

In this paper, we also propose to exploit the potential of facial shape information, i.e. a set of facial landmarks, to help CNN based methods learning more powerful and robust face representation. The basic idea is using different convolutional kernel according to the shape information, i.e. distribution of face landmarks, in order that the learned features would become more invariant to appearance variations caused by different viewpoints or expressions. One intuitive example of this idea is shown in Fig. 1. Another potential of this kernel adaptation is that the additional discriminant information contained in the shape may also be directly coded into the learned feature. Specifically, we propose a shape driven kernel adaptation for CNN and use automatically adapted kernels to more efficiently disentan-

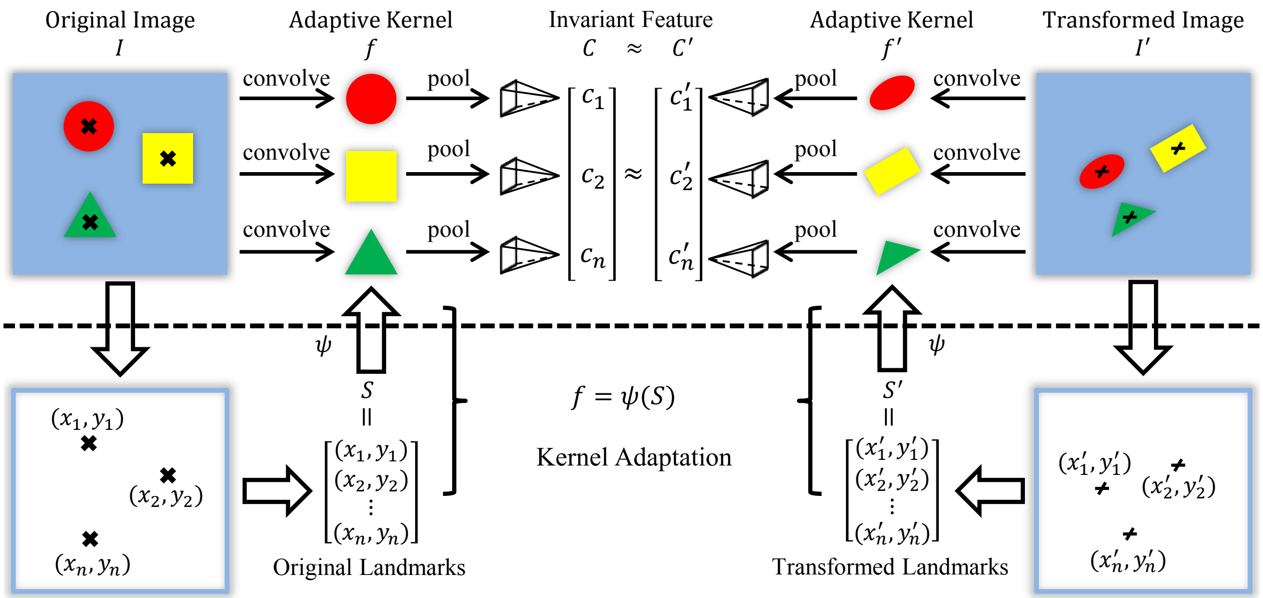


Figure 1. An toy example for clarification of the basic idea of proposed kernel adaptation in a CNN framework. Due to some real world variations, such as different viewpoints, the appearance of an image I may be significantly different to its transformed version of image I'. If proper kernel adaptation function $\psi(\cdot)$ is learned to generate a transformed version of kernel (also called convolutional filter), the feature maps generated can become invariant to these transformations.

gle the mixed factors in each input face image.

Due to the complex geometric structure and muscle movement, the facial appearance variation is often non-rigid. Thus, different facial region may demand different kernel adaptation function. Therefore, we propose a tree-structured convolutional architecture, each leaf of which processes a local facial region and holds a distinctive adaptation function. These adaptively learned local convolutional features are then fed into a tree-structured network for further deep representation learning.

The aforementioned framework is general and the idea of shape driven kernel adaptation may also be helpful in many other computer vision tasks. In this paper, we demonstrate the framework on various facial trait recognition tasks. This is not only because shape information is crucial for facial interpretation, but also because shape information of faces can be obtained more reliably with automatic algorithm, such as [2, 33], compared to other general objects.

In summary, we propose a novel tree-structured kernel adaptive CNN to exploit shape information in the CNN framework for robust facial trait recognition. The contributions of the paper include: 1) propose shape driven kernel adaptation in CNN framework, which helps learning robust face representations that are invariant to non-rigid appearance variations; 2) propose a tree-structured deep convolutional fusion hierarchy, which further enhances the power of kernel adaptation and 3) achieve the state-of-the-art performance in various facial trait recognition tasks, including identity, age and gender recognition.

2. Related Work

Recently, deep learning methods show notable potential in various face related computer vision tasks, such as facial landmark detection [28], facial trait classification [34], face verification [12, 29] and face recognition [38, 37].

One of the main focuses of these methods is designing suitable deep network structure to accomplish some specific tasks [12, 28, 29]. In [12], a hierarchical representation is learned from multiple overlapping local Convolutional Restricted Boltzmann Machine (CRBM). In [28], three levels of deep convolutional network are cascaded to estimate the accurate position of facial landmarks in a coarse to fine manner. In [29, 30], multiple local convolutional subnetworks are gradually fused to obtain robust face verification result. The hybrid structure proposed in [29, 30] is similar to our method, which learn convolutional features from multiple local regions and fuse these local features by a tree-structured network. Although our method also benefits from similar fusion hierarchy, we use such a hierarchy mainly because the kernel adaptation function is not be shared over the whole face image with non-rigid variation. Our method also benefit from the end-to-end optimization of the whole hierarchy, while the hybrid network proposed in [29, 30] only independently learn multiple local ConvNets and an additional metric learning method, e.g. Joint Bayesian [4], is needed for fusing local features.

The other of them try to modify the optimization objective to cope with some special scenarios [34, 38, 37].

In [34], a CNN based method was used to conduct facial age, gender and race classification. The main focus of [34] was to develop an online model adaptation method, which can adapt a pre-trained classification model to new environments in real time after deployment. Both [38] and [37] propose to extract pose-invariant features for across pose face recognition. Our kernel adaptation method also helps achieving pose invariance, but additional shape information is directly exploited in our method to handle general appearance variation in a more effective and intuitive way.

3. Kernel Adaptation for CNN

In real world environment, facial appearance may change significantly due to different poses and expressions, one traditional convolutional layer with fixed kernel functions may generate undesirably different responses for the same face. To achieve feature invariance under these complex variations, a mechanism, which can make the convolution kernel automatically adapt to the specific variations of each face instance, will be beneficial.

To this end, we propose a kernel adaptation mechanism for traditional CNN framework. Suppose the input face image is I and the kernel function is f, we hope the kernel function can be automatically adapted for the input face image according to a latent variable S. Then the convolution with kernel adaptation can be formulated as:

$$f = \psi(S, \Theta), \tag{1}$$

$$C = \varphi(I * f + b), \tag{2}$$

where $\varphi(\cdot)$ is a activation function, and $\psi(\cdot)$ is an adaptation function that can depict the relationship between the latent variable S and the proper kernel f. Θ is the parameter of ψ and b is an additive bias. Generally, the latent variable S can be any data that contains valuable information for disentangling the mixed factors in current input face image I. In this paper, we use facial shape data to serve as the latent variable S, which helps learning pose and expression robust feature for facial trait recognition. The "shape" data S used in this paper is a vector of the normalized coordinates of facial landmarks. In the kernel adaptation, Θ replacing the kernel f becomes the learnable parameter of the network. As long as $\psi(\cdot)$ is differentiable with respect to Θ, Θ can also be trained with the common back-propagation method [16] based on the chain rule without much effort.

Although the ideal adaptation function ψ may be very complex, in this paper, we use a simple linear function to approximate it. Formally, this liner function in our kernel adaptation method can be represented as:

$$f = W \cdot S, \tag{3}$$

where W is the linear matrix used to generate the adaptive kernel f. Note that the form of f in Eqn. (3) is a 1D vector and in Eqn. (2) is a rearranged 2D matrix with the same element values. With kernel adaptation as indicated by Eqn. (3), given an input face image I, the kernel functions f can be adaptively generated according to its shape information S. As a result, the feature learning process can automatically achieve certain complex geometric transformation invariance.

Intuitively, as appearance variation caused by pose and expression is non-rigid, different facial components may demand different kernel adaptation functions. Therefore, instead of using single adaptation function over the whole face, the kernel adaption is separately adopted in multiple local CNN subnetworks, indicated as C_i $(i = 1, 2, ..., N)$, over multiple local facial patches, indicated as P_i $(i = 1, 2, ..., N)$. In this way, each small facial patch P_i has its own adaptation function W_i. Moreover, only landmarks around the patch P_i contain valuable information for modeling the appearance deformations in this local patch. Thus, we only use local shape information S_i to infer the local adaptive kernel f_i of the local patch P_i. Formally, for each local subnetwork C_i, we represent its adaptive kernel f_i as a function of corresponding "shape" information S_i:

$$f_i = W_i \cdot S_i, \tag{4}$$

$$C_i = \varphi(P_i * f_i + b_i). \tag{5}$$

As the variation caused by pose and expression in each small local patch can be assumed as a rigid transformation approximately. This local linear adaptation function is capable in depicting the relation between local shape information and desired kernel function.

4. Tree-structured Kernel Adaptive CNN

In the last section, we propose to learn multiple local kernel adaptive CNNs. Similar to [30], we also attempt to fuse features learned from multiple facial patches. However, instead of ensemble these features in the last few fully connection layers, we propose a end-to-end tree-structured convolutional architecture to deeply integrate the face representation of local subnetworks.

4.1. Architecture

Totally, the proposed tree-structured kernel adaptive CNN consists of three stages, as shown in Fig. 2. Given a normalized face image I and corresponding facial landmarks $S = \{v_i | v_i \in \Re^2, i = 1, 2, ..., N_1\}$, we first construct multiple local kernel adaptive subnetworks C_i^1 $(i = 1, 2, ..., N_1)$ (Fig. 2: Stage 1), each of which learns discriminative features with adaptive kernel f_i^1 within local image patch P_i centered at the facial landmark v_i. Then the convolutional features of multiple local subnetworks are stacked and fed into the part-fusion subnetworks C_i^2 $(i =$

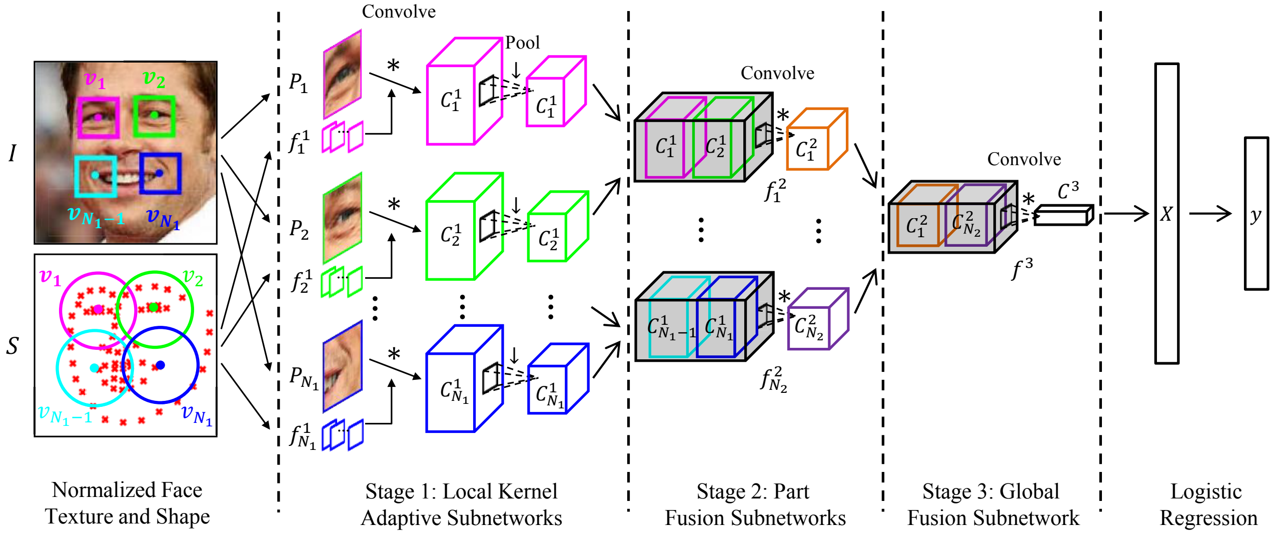

I
S
v1
v2
vN1−1
vN1
Convolve
Pool
P1
f11
C11
P2
f21
C21
PN1
fN11
CN11
Convolve
f12
C12
CN22
fN22
Convolve
C3
f3
X
y
Normalized Face Texture and Shape
Stage 1: Local Kernel Adaptive Subnetworks
Stage 2: Part Fusion Subnetworks
Stage 3: Global Fusion Subnetwork
Logistic Regression

Figure 2. Flowchart of the proposed tree-structured kernel adaptive CNN. Given a normalized face image I and corresponding facial landmarks $S = \{v_i\}_{i=1}^{N_1}$, multiple local kernel adaptive CNN subnetworks $\{C_i^1\}_{i=1}^{N_1}$ are constructed to learn features from multiple local patches $\{P_i\}_{i=1}^{N_1}$. The convolved features learned by multiple local subnetworks are then combined as the middle-level representations to learn high-level features with the fusion subnetworks, i.e. multiple part fusion subnetworks $\{C_i^2\}_{i=1}^{N_2}$ and a global fusion subnetwork C^3. Finally, a logistic regression layer is used to generate the final prediction y from vectorized high level convolved features X.

$1, 2, ..., N_2$) (Fig. 2: Stage 2) to learn the middle level representation. After that, a global-fusion convolutional subnetwork C^3 is used to generate high-level representation from the convolutional features of these parts (Fig. 2: Stage 3). Subnetworks in both fusion stages, i.e. the part-fusion subnetworks and the global-fusion subnetwork, consist of one or more stacked convolution modules. On top of the global-fusion subnetwork, a single logistic regression layer is catenated to generate the final prediction. Note that Fig. 2 only depicts a sketchy structure of the proposed tree-structured kernel adaptive CNN. The detailed configuration of the network which we use in the experiments will be presented in Section 5.

Although the structure is different from the conventional CNN [16], the tree-structure kernel adaptive CNN can also be similarly trained with the back-propagation method. The differences mainly lie in twofold: the kernel adaptation and the convolutional fusion. For kernel adaptation, we need to back-propagate the gradient of the convolutional kernels f_i^1 further to the generation function W_i^1 based on Eqn. (4):

$$\frac{\partial L}{\partial W_i^1} = \frac{\partial L}{\partial f_i^1}\frac{\partial f_i^1}{\partial W_i^1} = \frac{\partial L}{\partial f_i^1} S_i^T, \tag{6}$$

where L is the loss function of the final prediction. As $\frac{\partial L}{\partial f_i^1}$ can be calculated in typical back-propagation optimization of conventional CNN. The gradient of W_i^1 can be calculated with Eqn. (6).

Convolutional fusion is used in both the part-fusion subnetworks and the global-fusion subnetwork. For simplicity, we only present how to forward propagate features and backward propagate gradients in the global-fusion subnetwork. The basic idea is stacking the output feature maps of multiple part-fusion subnetworks as the input of the global-fusion subnetwork. As shown in Stage 3 of Fig. 2, the convolved features generated by the ith part-fusion subnetwork are denoted as C_i^2 ($i = 1, 2, ..., N_2$) and the convolved features generated by the global-fusion subnetwork are denoted as C^3. In the forward propagation step, the kth feature map C_k^3 ($k = 1, 2, ..., K_3$) in C^3 is obtained by convolving all the outputs of the part-fusion subnetworks with kernel function f^3. Suppose C_{ij}^2 ($j = 1, 2, ..., K_2$) is the jth feature map of C_i^2. Then, we have:

$$C_k^3 = \varphi(\sum_{i=1}^{N_2}\sum_{j=1}^{K_2} C_{ij}^2 * f_{ijk}^3 + b_k^3), \tag{7}$$

where f_{ijk}^3 is the corresponding kernel function, b_k^3 is an additive bias and $\varphi(\cdot)$ is the activation function. For better understanding of the subsequent backward propagation procedure, we present the forward propagation, as indicated in Eqn. (7), at element level and the uth row and vth column element $[C_k^3]_{uv}$ of C_k^3 can be expressed as:

$$[C_k^3]_{uv} = \varphi(\sum_{i=1}^{N_2}\sum_{j=1}^{K_2}\sum_{s,t}[C_{ij}^2]_{u+s,v+t} \cdot [g_{ijk}^3]_{st} + b_k^3), \tag{8}$$

where g_{ijk}^3 is central symmetric with respect to the kernel function f_{ijk}^3. In the backward propagation step, based on

Eqn. (8), the derivative of the learnable parameters b_k^3 and g_{ijk}^3 (or f_{ijk}^3) can be calculated with the chain rule by collecting derivatives propagated from each element of C_k^3:

$$\frac{\partial L}{\partial b_k^3} = \sum_{u,v} \frac{\partial L}{\partial [C_k^3]_{uv}} \frac{\partial [C_k^3]_{uv}}{\partial b_k^3}, \tag{9}$$

$$\frac{\partial L}{\partial [g_{ijk}^3]_{st}} = \sum_{u,v} \frac{\partial L}{\partial [C_k^3]_{uv}} \frac{\partial [C_k^3]_{uv}}{\partial [g_{ijk}^3]_{st}}. \tag{10}$$

Suppose the derivative of C_k^3 with respect to the loss function L, i.e. $\frac{\partial L}{\partial C_k^3}$, is already calculated and sigmoid activation function is used, we denote $\delta = \frac{\partial L}{\partial C_k^3} \circ C_k^3 \circ (1 - C_k^3)$, where "$\circ$" indicates the element-wise multiplication operation. Then, Eqn. (9) and (10) can be further simplified as:

$$\frac{\partial L}{\partial b_k^3} = \sum_{u,v} [\delta]_{uv}, \tag{11}$$

$$\frac{\partial L}{\partial [g_{ijk}^3]_{st}} = \sum_{u,v} [\delta]_{uv} \cdot [C_{ij}^2]_{u+s,v+t}, \tag{12}$$

where $[\delta]_{uv}$ is the element of δ, which lies in the uth row and vth column. To further back-propagate the gradient to the local subnetwork, the derivative of C_{ij}^2 is also calculated:

$$\frac{\partial L}{\partial [C_{ij}^2]_{u,v}} = \sum_{s,t} [\delta]_{u-s,v-t} \cdot [g_{ijk}^3]_{st}. \tag{13}$$

With the derivative of C_{ij}^2, the gradients can be further back-propagated to the former stages, i.e. the location adaptive CNN subnetworks. Now the forward and backward propagations through the convolutional fusion networks, i.e. part-fusion and global-fusion subnetworks, can be conducted according to Eqn. (8), (11), (12), and (13).

Note, the proposed tree-structured convolution architecture is equivalent to group convolution proposed in [15] except that the input feature maps of different groups are learned from different image regions.

4.2. Optimization

Typically, CNN is trained in a pure supervised manner without unsupervised pre-training. Although our tree-structured kernel adaptive CNN can also be trained with the back-propagation method as aforementioned, in practice, we adopt a more efficient optimization method. Similar to the strategy used in [38], before globally fine-tuning the whole network, we first conduct a stage-wise pre-training of parameters in each convolutional subnetworks in a pure supervise manner.

More specifically, the optimization of tree-structured kernel adaptive CNN can be divided into four steps. First, we train multiple kernel adaptive CNN models with Eqn. (6). After removing the logistic regression layer of the trained local CNN model, the remaining stacked convolution-pooling modules are used as the first stage of tree-structured CNN. Then, the part-fusion subnetworks and global-fusion subnetwork are supervisely pre-trained in the 2rd and 3rd step using the same strategy. Finally, with the initialized parameters, the whole network is further fine-tuned with the techniques illustrated in the previous subsection. Note, SGD is used to optimize our model with momentum 0.9. base learning rate 0.01 and scaled by 0.95 after each epoch. We run 100 epochs stage-wise pre-training and 60 epochs global end-to-end fine-tuning.

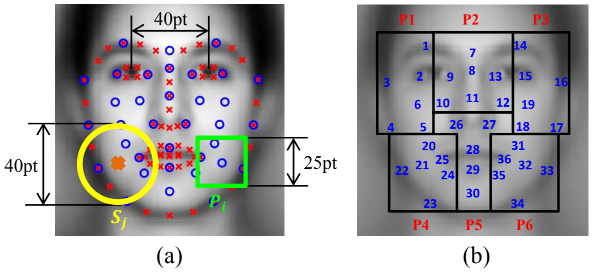


Figure 3. Implementation details of network inputs and topological structure. (a) Local texture and shape information P_i and S_j; (b) topological structure between local adaptive subnetworks and part fusion subnetworks.

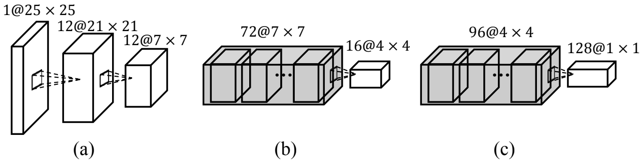


Figure 4. Implementation details of parameter settings. (a) settings of single location adaptive subnetwork; (b) settings of single part-fusion subnetwork and (c) settings of global-fusion subnetwork.

5. Implementation Details

In this section, we elaborate on the structure of the proposed adaptive convolutional neural network which is used to conduct facial trait recognition. First, the face and landmarks are automatically detected. Since our method attempt to extensively exploit shape information, the correctness of the used shape information is an important issue that needs to be considered. To this end, we implement a state-of-the-art face alignment algorithm, i.e. Supervised Descent Method (SDM) introduced in [33], to conduct the facial landmark detection. Trained with large scale wild face image data provided by the i-bug group in [23], the algorithm works well in all three benchmark databases, i.e. WebFace [20] , Morph II [22] and MultiPIE [7].

As shown in Fig. 3 (a), 68 landmarks (red crosses) are detected. With these landmarks, the face can be normal-

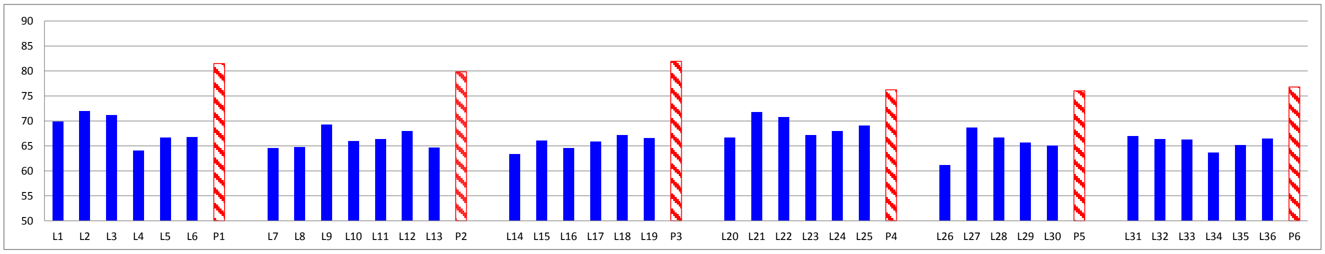


Figure 5. Gender classification of local and part-fusion subnetworks with stage-wise initialization on WebFace database. Note the label ”Li” or ”Pi” under the horizontal axis indicates the ith local or part subnetwork. Please refer to Fig. 3 for definition of the index. As shown, the part fusion subnetwork can effectively fuse multiple local subnetworks.

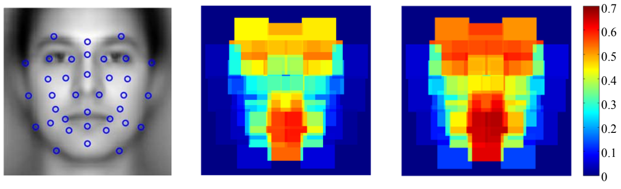
ized with the distance between eye centers, i.e. 40 pixels. Then, 36 landmarks (blue circles in Fig. 3 (a)) are selected to extract local patches P_i (green square in Fig. 3 (a)) with the size of 25×25 pixels. Note, we also select some interpolated landmarks to ensure approximately even sampling of facial regions. With the intensity normalized to $[0, 1]$, these local patches are then fed into the local kernel adaptive subnetworks as texture information. As aforementioned, we conduct kernel adaptation in the local subnetworks. As shown in Fig. 3 (a), the landmarks within a certain distance, i.e. within the yellow circle in Fig. 3 (a), are selected as auxiliary landmarks for the corresponding patch and the coordinates of these nearby auxiliary landmarks are used as the local shape information S_j. In the second stage of tree-structured kernel adaptive CNN, 6 part-fusion subnetworks are used to combine the spatially nearby local adaptive subnetworks. As shown in Fig. 3 (b), each part-fusion subnetwork combines approximately 6 local adaptive subnetworks. Note, the black rectangles in Fig. 3(b) is only used to indicate the connections between local adaptive subnetworks and part-fusion subnetworks, not the real facial regions covered by the part-fusion subnetworks. The parameter settings of the local adaptive subnetworks, the part-fusion subnetworks and the global-fusion subnetwork are shown in Fig. 4 (a), (b) and (c) respectively.

6. Experiments

Comprehensive experiments are conducted on three databases, including WebFace [20] , Morph II [22] and MultiPIE [7]. First, the effectiveness of kernel adaptation and tree-structured architecture are evaluated in Section 6.1 and Section 6.2. Then the comprehensive comparisons of the state-of-art methods in various facial trait recognition tasks are provide in Section 6.3.

6.1. Evaluation on Kernel Adaptation

In order to clarify the effect of kernel adaptation with respect to facial poses, we compare CNN with or without kernel adaptation on MultiPIE database [7], indicated as a-CNN and CNN respectively. There are 337 persons in MultiPIE database, we use the face images that have yaw pose



(a) Reference Face (b) Results of Local CNN (c) Results of Local a-CNN

Figure 6. Face Identification on MultiPIE database. (a) A reference face to clarify local patch positions; (b) Result of CNN without kernel adaptation; (c) Result of CNN with kernel adaptation. Note the colorbar maps accuracy to colors. **Best viewed in color.**

within $[-45^o, +45^o]$ with neutral expression and normal illumination of all four sessions to conduct experiments. Following the evaluation protocol used in [37], we use face images of the last 88 subjects as the training set with the remaining 249 subjects' images as the test set. And similar to [37], we also use random “faces” as labels for optimization, which means we use the same random “face” label for the face images with different poses but the same identity.

The performances of 36 local subnetworks with and without kernel adaptation are shown in Fig. 6(c) and Fig. 6(b) respectively. As shown, in all the local patches, the network models trained with kernel adaptation consistently outperform that trained without kernel adaptation in the multi-pose face recognition task. As shown, for the face regions that change more dramatically in different viewpoints, such as nose, the performance gains are more significant, which further validate the propose kernel adaptation method in handling non-rigid appearance variation.

To better understand the effect of kernel adaptation, we present the differences between the mean frontal adaptive kernels and the corresponding mean profile adaptive kernels in Fig. 7. As shown, the differences between frontal and profile adaptive kernels are symmetric with respect to different yaw poses. This adaptation results may help learning the pose-invariance features.

6.2. Evaluation on Tree-structured Architecture

To evaluate the tree-structured convolutional architecture, we compare our tree-structured CNN, denoted as tree-CNN, with the conventional CNN for gender estimation on

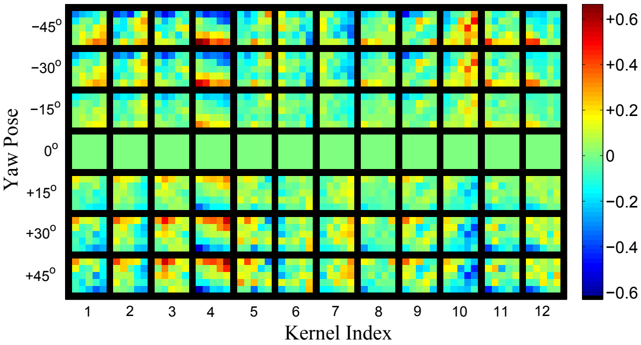


Figure 7. Differences between frontal and profile adaptive kernels. Since the appearance variation caused by yaw pose are approximately symmetric, as is shown, the kernel adaption method can automatically learn such rule and generate symmetrically different kernels to cope with such variation. **Best viewed in color.**

WebFace database [20, 26]. WebFace database contains 59930 face images, which is the largest database in the literature of age and gender estimation study. As the images are collected from the web, the database contains large variations in pose, facial expression, illumination, etc, which make the database one of the most difficult databases for age and gender estimation. For gender classification, we use 51330 images of humans with ages above 5 to conduct 4-fold cross-validation. As we adopt stage-wise supervised initialization, we first show the performance of 36 location adaptive subnetworks and 6 part fusion subnetworks in the first and second stages of the tree-CNN. As shown in Fig. 5, the convolutional fusion of local CNN can consistently improve the classification accuracy.

After stage-wise pre-training and global fine-tuning, as presented in Fig. 8, the proposed tree-structured CNN outperforms conventional CNN with comparable filter number and the same network depth. Note we denote the number of convolutional layers as the depth of a CNN. With the same tree structure, we also evaluate our whole framework, i.e. tree-CNN with kernel adaptation, denoted as tree-a-CNN. As shown in Fig. 8, two layers tree-a-CNN outperforms three layers tree-CNN, which may indicate that proposed kernel adaptation helps learning more compact model to achieve feature invariance. This is very helpful for an end-to-end learning system.

6.3. Evaluation on Facial Trait Recognition

Finally, we present comprehensive results on WebFace [26], Morph II [22] and MultiPIE [7] in Table 1 to evaluate the effectiveness of propose shape driven kernel adaptive CNN for facial trait recognition. For clarification, we highlight with underline the previous state-of-the-art methods, which have achieved convincing results on these databases. The best result in each database are highlighted with bold typeface. For comprehensive comparison, we also show the performance of AlexNet [15] on WebFace and Morph II and DeepFace [32] on MultiPIE. Note, we use the structure configuration files provided by Caffe [14] community to train the AlexNet from scratch on the corresponding database. We realize the local connected layer of DeepFace in Caffe and train DeepFace model from scratch on MultiPIE database. For both AlexNet and DeepFace, we run totally 50,000 iterations, with base learning rate 0.01. Stepsize is set as 20,000 and after each step the learning rate is divided by 10. Note, for DeepFace, 2D image warping instead of the 3D alignment is used to align face images to the mean shape.

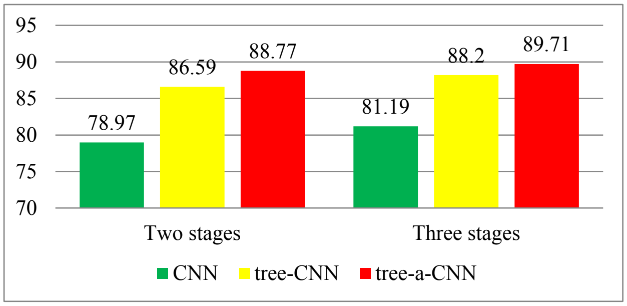


Figure 8. Gender Classification evaluation of CNN, tree-CNN and tree-a-CNN on WebFace Database.

WebFace Database: We conduct age and gender classification on the WebFace database. We use all 59930 images to conduct age estimation and 51330 images with ages above 5 to conduct gender estimation with a 4-fold cross validation protocol as in [26]. The comparison results are shown in Table 1(a). As WebFace is captured in wild environment, images contains large pose and expression variation. Thus our proposed tree-a-CNN, which help disentangling these irrelevant variations, achieves significant improvements over other comparisons. Our 3 layer tree-a-CNN also outperforms Krizhevsky's 8 layer AlexNet, more specifically tree-a-CNN/Alexnet has 0.34M/60M parameters and 23K/650K neurons, which further validate the effectiveness of the shape driven kernel adaptation and tree-structured convolutional fusion architecture.

Morph II Database: Age and gender classification experiments are also conducted on the Morph II database, which contains about 55,000 face images. With the same evaluation protocol used in [8, 9], the experimental results are shown in Table 1(b). AlexNet achieves slightly better performance in gender classification tasks. However, our tree-a-CNN method still achieves best reported results on age estimation task. The relatively small improvements over other comparison method is mainly because that the images of Morph2 are collected under controlled environment and less challenging pose variations are included. The proposed kernel adaptation is more effective in challenging real world environments, in which the captured face images may hold a massive of relevant or irrelevant factors to be disentangled.

Table 1. Facial trait recognition experiments on WebFace, Morph II and MultiPIE databases.

(a) WebFace

Methods	Gender Accuracy	Age MAE
BIF [10]	79.32	10.65
RF [19]	-	9.38
Ridge [26]	86.99	9.75
AlexNet [15]	88.26	9.43
tree-CNN	88.20	8.60
tree-a-CNN	**89.71**	**7.27**

(b) Morph II

Methods	Gender Accuracy	Age MAE
BIF [10]	96.58	5.09
KPLS [8]	98.35	4.04
KCCA [9]	98.45	3.98
Ridge [26]	97.74	4.80
AlexNet [15]	**98.53**	4.39
tree-CNN	98.38	3.90
tree-a-CNN	98.48	**3.61**

(c) MultiPIE

Protocol	Methods	Identification Accuracy per Pose						Avg
		-45^o	-30^o	-15^o	$+15^o$	$+30^o$	$+45^o$	
Setting 1	RFSME-1[37]	81.5	93.2	98.4	96.8	92.4	88.8	91.8
	RFSME-20[37]	96.8	**100**	**100**	**100**	**100**	96.4	98.8
	tree-CNN-1	82.8	99.0	**100**	**100**	98.7	83.6	94.0
	tree-a-CNN-1	95.3	99.9	**100**	**100**	**100**	95.4	98.4
	tree-a-CNN-20	**97.3**	99.9	**100**	**100**	**100**	**97.2**	**99.1**
Setting 2	VAAM[1]	74.1	91	95.7	95.7	89.5	74.8	86.9
	MLCE[18]	90.0	94.3	95.3	94.7	93.7	87.7	92.6
	MDF-SA[17]	93.0	98.7	99.7	99.7	98.3	93.6	97.2
	LE [3]	86.9	95.5	99.9	99.7	95.5	81.8	93.2
	CRBM [12]	80.3	90.5	94.9	96.4	88.3	75.2	87.6
	DeepFace [32]	79.3	96.2	99.2	**100**	97.1	83.5	92.6
	RL [38]	95.6	98.5	**100**	99.3	98.5	**97.8**	98.3
	tree-CNN-1	85.3	99.3	**100**	**100**	99.6	86.2	95.1
	tree-a-CNN-1	**97.1**	**100**	**100**	**100**	**100**	**97.8**	**99.2**

MultiPIE Database: We conduct face identification experiments on MultiPIE database to evaluate capability of proposed method in handling pose variation. The results are presented with two commonly used protocols, i.e. setting 1 (with 88/249 subjects used for training/testing) and setting 2 (with 200/137 subjects used for training/testing). In Table 1(c), the first block evaluation use setting 1 and the second block evaluation use setting 2. Similar to [37], we also use random face as optimization target. Different from age and gender recognition which only use the final output of the network, for face identification task we concatenate the output of local, part and global CNN to constitute a comprehensive representation of the face images. As subjects in the test set are not included in the training set, an additional distance metric is needed to find the nearest neighbor for face identification. In this paper, cosine distance metric is used. With this simple cosine distance over the comprehensive representation learned by tree-a-CNN, we achieve state-of-the-art face identification accuracy (this method is denoted as tree-a-CNN-1). Note the results of tree-a-CNN-20 is obtained by averaging 20 models learned with different set of random face labels.

7. Conclusions and Future Work

In this paper, we propose a kernel adaptation method in CNN to exploit shape information for disentangling irrelevant non-rigid facial appearance variations. Since different facial regions have different deformations, to better exert its function, we adopt kernel adaptation in multiple local regions respectively and further propose a tree-structured convolutional architecture to jointly learn features in an end-to-end manner. Evaluations on facial trait recognition tasks demonstrate the state-of-the-art performances of the proposed tree-a-CNN model.

Although our network has relatively shallow structure comparing to the state-of-the-art deep convolutional neural networks [15, 24, 31], our method achieves comparable or better performance than AlexNet [15] and DeepFace [32]. These results suggest that kernel adaptation method provides a more compact and effective way to disentangle complex factors in facial images. This is very helpful for an large end-to-end system like deep networks. In future, we will try to deploy our kernel adaptation method into larger and deeper networks to fully explore the potential of shape information for robust feature learning.

8. Acknowledgement

This work is partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. (61303178, 61173065, 61222211, 61402443), Microsoft Research Asia grant R-263-000-B13-597.

References

[1] A. Asthana, T. Marks, M. Jones, K. Tieu, and R. MV. Fully automatic pose-invariant face recognition via 3d pose normalization. In *ICCV*, 2011.

[2] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *IJCV*, 2014.

[3] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *CVPR*, 2010.

[4] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *ECCV*. 2012.

[5] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012.

[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013.

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

[8] G. Guo and G. Mu. Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression. In *CVPR*, 2011.

[9] G. Guo and G. Mu. Joint estimation of age, gender and ethnicity: Cca vs. pls. In *FG*, 2013.

[10] G. Guo, G. Mu, Y. Fu, and T. Huang. Human age estimation using bio-inspired features. In *CVPR*, 2009.

[11] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[12] G. Huang, H. Lee, and E. Learned. Learning hierarchical representations for face verification with convolutional deep belief networks. In *CVPR*, 2012.

[13] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, 2009.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998.

[17] 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.

[18] S. Li, X. Liu, X. Chai, H. Zhang, S. Lao, and S. Shan. Maximal likelihood correspondence estimation for face recognition across pose. *TIP*, 2014.

[19] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *ACCV*. 2012.

[20] B. Ni, Z. Song, and S. Yan. Web image mining towards universal age estimator. In *ACM MM*, 2009.

[21] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *ICCV*, 2013.

[22] K. Ricanek and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *FG*, 2006.

[23] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshops*, 2013.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proc. IEEE*, 2006.

[26] Z. Song. Visual image recognition system with object-level image representation. In *PhD thesis, National University of Singapore*, 2012.

[27] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011.

[28] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013.

[29] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *ICCV*, 2013.

[30] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

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

[33] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[34] M. Yang, S. Zhu, F. Lv, and K. Yu. Correspondence driven adaptation for human profile recognition. In *CVPR*, 2011.

[35] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*. 2014.

[36] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *CVPR*, 2014.

[37] Y. Zhang, M. Shao, E. K. Wong, and Y. Fu. Random faces guided sparse many-to-one encoder for pose-invariant face recognition. In *ICCV*, 2013.

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