Conditional Convolutional Neural Network for Modality-aware Face Recognition

Chao Xiong[1], Xiaowei Zhao[1], Danhang Tang[1], Karlekar Jayashree[3], Shuicheng Yan[2], and Tae-Kyun Kim[1]

[1]Department of Electrical and Electronic Engineering, Imperial College London
[2]Department of Electrical and Computer Engineering, National University of Singapore
[3]Panasonic R&D Center Singapore
{*chao.xiong10, x.zhao, d.tang11*}*@imperial.ac.uk, Karlekar.Jayashree@sg.panasonic.com, eleyans@nus.edu.sg, tk.kim@imperial.ac.uk*

Abstract

Faces in the wild are usually captured with various poses, illuminations and occlusions, and thus inherently multimodally distributed in many tasks. We propose a conditional Convolutional Neural Network, named as c-CNN, to handle multimodal face recognition. Different from traditional CNN that adopts fixed convolution kernels, samples in c-CNN are processed with dynamically activated sets of kernels. In particular, convolution kernels within each layer are only sparsely activated when a sample is passed through the network. For a given sample, the activations of convolution kernels in a certain layer are conditioned on its present intermediate representation and the activation status in the lower layers. The activated kernels across layers define the sample-specific adaptive routes that reveal the distribution of underlying modalities. Consequently, the proposed framework does not rely on any prior knowledge of modalities in contrast with most existing methods. To substantiate the generic framework, we introduce a special case of c-CNN via incorporating the conditional routing of the decision tree, which is evaluated with two problems of multi-modality – multi-view face identification and occluded face verification. Extensive experiments demonstrate consistent improvements over the counterparts unaware of modalities.

1. Introduction

It can be commonly assumed that data may appear in different views or styles in computer vision. For example, objects of the same class may have different types in object recognition, e.g., cars may be of various types and brands; or in human pose estimation, people with the same pose may have different identities. Similarly, many face related tasks deal with images with variations in terms of pose, occlusion and lighting, and thus are inherently multimodal. Such multimodality leads to a large intra-class variation, which poses a great challenge to most existing approaches for face identification or verification.

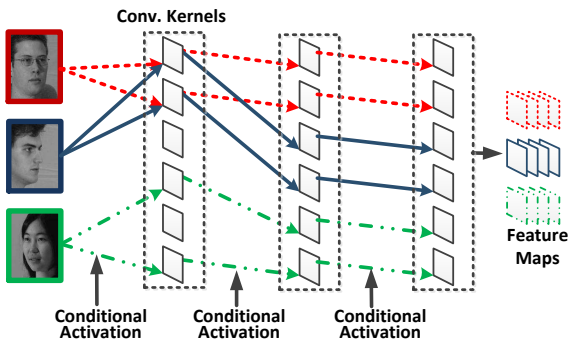

Figure 1. Illustration of c-CNN. Each line type stands for one modality. Each image is passed along with a modality-specific route indicated by the corresponding colored arrows. Only the kernels along the route are activated and utilized to extract features. The passing route defines the splitting w.r.t. inherent modalities in a coarse-to-fine manner: similar modalities, e.g., modality of red dashed line and blue solid line, may share certain kernels at the beginning layers.

A general approach to handle multimodal problems is to find a shared feature space where data of different modalities are directly comparable. Conventional methods, such as Canonical Correlation Analysis (CCA) [9] and Partial Least Squares (PLS) [6], aim at learning modality-specific projection matrices that lead to maximal covariance among instances of the same class in the shared latent space. Many works are specifically designed to deal with two-view data. In particular, data of one view are carefully projected into the subspace of the other modality. This idea has seen popular applications in synthesis based approaches in various problems, such as sketch-photo verification [27], low resolution vs. high resolution face matching [19], etc. Despite excellent work has been done on synthesis, this may in principle be an ill-posed problem that is more difficult than discriminatively comparing images of two different modalities.

Most of the aforementioned approaches are built on hand-crafted features. However, it is difficult to manually design features insensitive to the variations across modalities, since instances of different modalities usually span different feature spaces. In addition, the generic features, such as SIFT [20], HOG [3] and LBP [21], are designed to solve certain problems, and thus may not be optimal for the specific variations in the given problems. Moreover, some characteristic visual information may be lost in extraction (especially the quantization) stage, which usually cannot be recovered in the later stages. Recent deep learning methods [25, 7, 18], on the other hand, are able to learn an effective representation from raw-pixel inputs by directly optimizing with regard to the given objective. Deep learning also witnessed several attempts in handling cross-modality variations [31, 32, 12]. In most aforementioned approaches, the training or even testing instances come along with pre-defined modality information. For example, many approaches for multi-pose face recognition assume that the head pose is known during training. However, the ground-truth modality information is not usually available in practice. Moreover, it is also possible that the modalities of data are vague and difficult to define explicitly when, e.g., faces appear with multiple variations in poses, illumination, expression, occlusion, etc.

In this paper, we introduce a generic deep learning framework, termed as conditional Convolution Neural Network (c-CNN), to address multimodal classification problems with no prior knowledge on data modality. The proposed network automatically learns the inherent modality distribution and the feature representation with regard to a unified objective. In traditional CNN, the convolution kernels for each sample are immutable during training, and all the input samples are processed with the same kernels if no modality information is provided. In contrast, we include conditional computation of the "routes" for samples to propagate through the network. In particular, for each sample in one training epoch, the convolution kernels are sparsely activated within each layer, and the activated kernels across layers define a "route" for the given sample as shown in Figure 1. The activations of kernels in different layers are dependent and jointly optimized in a learnt manner. To be more specific, the activation probability of a kernel for a certain sample is conditioned on the corresponding intermediate representation and the routing status in the lower layers.

Conditional routing brings benefits in two folds: 1) The large intra-class variations across modalities make it very difficult to model the complex problem with a unified representation. The conditional routing gradually projects data of different modalities into several subspaces where the intra-class variations are more easier to be learnt; 2) Conditional routing activates only a limited number of convolution kernels in a learnt and optimized way. As a result, the computation cost is largely reduced, which makes the network more scalable. Decision tree inherently embeds the concept of conditional computation via hierarchical partitions, and thus is incorporated into CNN to substantiate the proposed framework. In particular, each tree node learns the intermediate representation and finds an optimal way to split samples at the same time. The proposed method is evaluated in two recognition problems of multi-modal faces, and proved to be effective with various comparisons.

2. Related Work

Multimodality spans a wide range of research, and has been explored in a large number of prior works. Due to the page limit, we only include works related to cross-modal face analysis in this section.

Common approaches handle the variations across modalities via mapping samples into a shared latent space. Kim and Josef [14] introduced a set of locally linear transformations to address multi-view face recognition. The proposed method maximizes the separability of classes locally while promoting consistency between the multiple local representations of single class objects. Abhishek et al. [23] used Partial Least Squares (PLS) to linearly map images in different modalities to a common linear subspace in which they are highly correlated. The proposed method is evaluated in cross-view, cross-resolution and sketch vs. photos face matching problems, and demonstrates considerable improvements over conventional methods. Abhishek et al. [22] proposed the Discriminant Multiple Coupled Latent Subspace framework to handle cross-view face recognition. It learns a set of pose-specific projection directions such that the projected images of the same subject are maximally correlated in the target latent space. Kan et al. [13] followed a similar approach to handle multi-view object recognition. They jointly learn multiple view-specific linear transformation in a non-pairwise manner. In these papers, the global non-linear data structures are assumed to be linearly separable in the transformed local spaces. Motivated by the recent success of deep features [15, 18], this paper proposes to learn the required nonlinear mappings within the latent local spaces with deep neural network.

Synthesizing faces of a certain modality is also explored in a statistic manner by many previous studies. Liu et al. [19] synthesized high-resolution face images from low-resolution images via integrating a global parametric model and a local non-parametric model. Wang and Tang [27] proposed a face photo retrieval system, which transforms a face image into a sketch. The proposed system conducts transformation on shape and texture of face images respectively. Zhang et al. [29] targeted at face recognition with variations of illumination and pose. They proposed a texture synthesis method by employing a generic 3D face shape. Similarly,

Li et al. [17] transformed faces of multiple poses to their frontal view via 3D face registration. However, the cross-modality transformation is complex and difficult to learn since it usually requires the corresponding samples in the target modalities to be available for each image, which is not always the case in practice. Therefore, the cross-modality synthesis could be an harder problem than the direct discriminative matching of multimodal subjects.

Recent research on deep learning [25, 7, 18] stimulates many applications of deep models in recognition problems with multimodality. Zhu et al. [31] transformed faces under any pose and illumination to their canonical view. The proposed network learns the feature extraction layers and the reconstruction layer jointly. Kan et al. [12] also addressed the cross-pose problem with a reconstruction-based deep model. The model transforms faces of large view gradually to its frontal view layer by layer. Zhu et al. [32] proposed a multi-task learning method to optimize the pose estimation and recognition objective in a joint manner. The results indicate that the pose information provides important clues in matching faces across views. In most aforementioned works (with either manual features or deep learning), the feature extraction or subspace transformation are defined or learnt specifically for each modality. Under such a framework, the modalities of data have to be pre-defined explicitly, or in other words, the modality which the data instance belongs to has to be known. In contrast, our framework defines a generic method in handling multimodality problems without any prior knowledge on modality. Instead, the modality is learnt together with the feature representation in a deep model with conditional computation.

The cascade of sample splitting in decision tree embeds the idea of conditional computation, and is well explored by many tree-structured classifiers [26, 28, 30, 11]. The fusion of decision tree (forest) and feature learning is also mentioned in a few recent works. Bulo and Kontschieder [1] aimed at finding the optimal split function at each node of the tree with MLP. However, the optimal splitting of samples is learnt in the traditional layer-by-layer manner. In other words, the optimization of the split network in a node is isolated from the learning of both its parent node and the existing nodes in other branches. Fanello et al. [5] attempted to learn the optimal filtering kernels and apply them to each data point. However, the filters are adopted as the PCA components learnt from noisy patches of multi-scale. The optimal filters are actually "chosen" from a random pool to minimize the energy functions of the nodes. Similar to [1], the split function is learnt separately for each node. Moreover, there is no joint learning of features and splitting nodes in either approach. In contrast, we jointly optimize the splitting nodes of the tree and the convolution kernels of the neural network with regard to a unified objective function. To the best of our knowledge, this kind of approach has not been tackled in prior works.

3. Conditional Convolutional Neural Network

In this paper, we assume that the given problem is potentially multimodal, and the modality information is not known for either training or testing. This is a more general assumption in practice.

The inherent modality is explored via finding the optimal set of convolution kernels to be activated. For a given sample, only the corresponding activated convolution kernels are utilized to extract features. The activated kernels within each layer define a passing route for a given sample. Intuitively, training samples of the same modality should follow the same route through the network. Traditional CNN activates all the kernels for all the training samples. For c-CNN, the activation of kernels in the layer i is jointly determined by the present input representation $\boldsymbol{X}_n^{(i)}$ to the layer i and the passing route in the lower layers $\{\boldsymbol{\theta}_n^{(j)}, j = 0, ..., i-1\}$.

We denote n as the index of input samples, and the corresponding forward function can be formulated as follows

$$\boldsymbol{X}_{n,k}^{(i+1)} = g_{n,k}^{(i)} \cdot \sigma(\widetilde{\boldsymbol{W}}_k^{(i)} * \boldsymbol{X}_n^{(i)} + b^{(i)}), \tag{1}$$

where $\boldsymbol{X}_{n,k}^{(i+1)}$ is the k-th kernel map of the n-th sample in layer $i+1$, and $g_{n,k}^{(i)}$ denotes the activation indicator of the k-th convolution kernel $\widetilde{\boldsymbol{W}}_k^{(i)}$. $g_{n,k}^{(i)}$ follows a Bernoulli distribution, i.e., $g_{n,k}^{(i)} \sim B\{1, p_{n,k}^{(i)}\}$, where

$$p_{n,k}^{(i)} = Pr(\theta_{n,k}^{(i)} | \boldsymbol{X}_n^{(i)}, \boldsymbol{\theta}_n^{(i-1)}, ..., \boldsymbol{\theta}_n^{(0)}), \tag{2}$$

and $\theta_{n,k}^{(i)}$ is the k-th element in $\boldsymbol{\theta}_n^{(i)}$. It should be noted that c-CNN aims at exploring the underlying modality distribution of data and the corresponding feature representation for each modality in a unified framework. In particular, the feature extraction parameters $\widetilde{\boldsymbol{W}}_k^{(i)}$ and $b^{(i)}$ and the kernel activation parameter $\boldsymbol{\theta}_n^{(i)}$ are learnt with regard to a unified objective function in a joint manner.

The conditional activation of convolution kernels can be defined in various ways. Decision tree embeds the concept of conditional computation in the hierarchy of simple decisions and has seen plentiful applications in various fields. The leaf nodes in one layer are mutually exclusive, and each sample can be passed only to one leaf node. The choice of leaf nodes for certain input is conditioned on the split function of its parent nodes and the existing route in the above layers. The aforementioned characteristics make the decision tree a good option to realize c-CNN. In this paper, the conditional computation of decision tree is incorporated into CNN as a specific instance of c-CNN. In particular, the tree nodes split the convolution kernels in each layer into several mutually exclusive kernel sets. However, there is

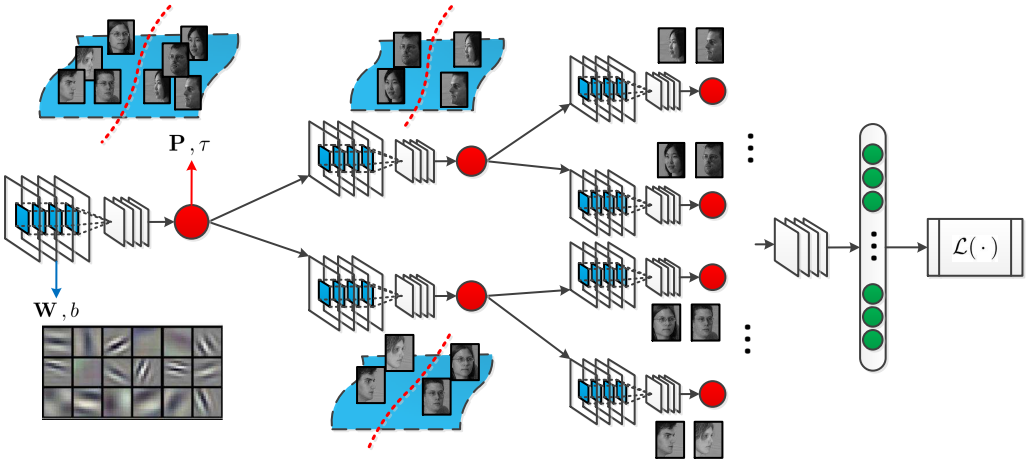

Figure 2. A specific example of c-CNN with Modality-aware Projection Tree (MPT). Each tree node computes the intermediate representation with CNN and the partition of samples in the projected latent space. With the help of MPT, samples of different modalities are gradually separated layer by layer and finally passed into the different leaf nodes. Both the features and the split functions are jointly optimized w.r.t. one unified loss function $\mathcal{L}$.

no such a hard segmentation constraint for generic c-CNN. The assignments of convolution kernels are more flexible for a given input sample in generic c-CNN. Therefore, this decision tree based approach can be regarded as a simplified case. The proposed network includes two components – Modality-aware Projection Tree and Convolutional Neural Network Branch. Detailed explanations are included for both components in the following subsections.

3.1. Modality-aware Projection Tree

Modality-aware Projection Tree (MPT) aims at defining a hard partition in the sample space such that samples of the same modality fall into the same leaf node. The modality is explored via learning of the split function for each node of the tree. To be more specific, we intend to learn the splitting of samples in an unsupervised manner such that the sample space is segmented with regard to the inherent modalities as illustrated in Figure 2.

Let's denote $\mathcal{X}$ and $\mathcal{Y}$ as the input and output space for a given classification problem. To begin with, we define a fully-grown decision tree of depth D. The node of the tree is denoted as $V^{(i,j)}$, where i is the index of the layer in the tree and j is the index of the leaf node in the i-th layer. Correspondingly, $\boldsymbol{X}_n^{(i,j)}$ is the intermediate representation of the sample $\boldsymbol{x}_n \in \mathcal{X}$.

Within the node $V^{(i,j)}$, the passing route of a sample is determined by a split function $\varphi : \mathcal{S} \rightarrow \{\mathcal{S}^{\mathcal{L}}, \mathcal{S}^{\mathcal{R}}\}$, if we denote the whole input set for this node as $\mathcal{S}$, and the subsets of two child nodes as $\mathcal{S}^{\mathcal{L}}$ and $\mathcal{S}^{\mathcal{R}}$ respectively. The split function can be formulated as

$$\boldsymbol{x} = \begin{cases} \mathcal{S}^{\mathcal{L}}, & \varphi(\boldsymbol{x}) \geq 0 \\ \mathcal{S}^{\mathcal{R}}, & \varphi(\boldsymbol{x}) < 0 \end{cases} \tag{3}$$

In this paper, the MPT is constructed in a similar way as Random Projection Tree [4]. The split function is defined with a projection vector $\boldsymbol{P}^{(i,j)}$ and a bias $\tau^{(i,j)}$ as follows,

$$\varphi(\boldsymbol{x}) = \boldsymbol{x}^T \cdot \boldsymbol{P}^{(i,j)} + \tau^{(i,j)}. \tag{4}$$

An unsupervised constraint is imposed for each node such that the distance between the centroids of two sub-clusters is maximized. The corresponding node-wise loss is formulated as

$$\mathcal{L} = \frac{\frac{1}{N} \sum\limits_{\boldsymbol{x} \in \mathcal{S}} \varphi(\boldsymbol{x})^2}{(\frac{1}{N_{\mathcal{L}}} \sum\limits_{\boldsymbol{x} \in \mathcal{S}^{\mathcal{L}}} \varphi(\boldsymbol{x}) - \frac{1}{N_{\mathcal{R}}} \sum\limits_{\boldsymbol{x} \in \mathcal{S}^{\mathcal{R}}} \varphi(\boldsymbol{x}))^2}, \tag{5}$$

where $N_{\mathcal{L}}$ and $N_{\mathcal{R}}$ are the numbers of samples falling into the left and the right child node respectively, and $N = N_{\mathcal{L}} + N_{\mathcal{R}}$.

3.2. Convolutional Neural Branch

Apart from the splitting of the input samples, each tree node also learns an intermediate representation with regard to the given objective directly. In particular, a tree node $V^{(i,j)}$ contains a standard convolutional layer $C^{(i,j)}$ with max-pooling.

When a sample is given at the root node of the tree, it is passed forward along a specific path. Along that path,

the given sample is processed through a complete Convolutional Neural Network, named as Convolutional Neural Branch (CNB), at the same time. MPT is prone to constrain samples with the same inherent modality to follow the same path such that each CNB learns a modality-specific mapping to the shared latent feature space. Different from conventional approaches for learning modality-specific mapping, CNBs of different modalities can share certain intermediate nodes as in Figure 2. Our motivation is that samples of similar modalities should be processed more similarly than those of distant modalities.

We denote $\boldsymbol{W}^{(i,j)}$ and $b^{(i,j)}$ to be the weight and bias of the convolutional layer for the node $V^{(i,j)}$. The corresponding forward function is defined as

$$\widetilde{\boldsymbol{X}_n^{(i,j)}} = \sigma(\boldsymbol{W}^{(i,j)} * \boldsymbol{X}_n^{(i,j)} + b^{(i,j)}), \tag{6}$$

where $*$ represents the convolution operator.

The hierarchical splitting of decision tree inherently takes into account the routing status in the previous layers. Accordingly, the conditional forward function in Eqn.(1) is transformed as follows,

$$\begin{cases} \boldsymbol{X}_n^{(i+1,2j)} & = \mathbb{1}(\varphi(\widetilde{\boldsymbol{X}}_n^{(i,j)}) \geq 0) \cdot \widetilde{\boldsymbol{X}}_n^{(i,j)} \\ \boldsymbol{X}_n^{(i+1,2j+1)} & = \mathbb{1}(\varphi(\widetilde{\boldsymbol{X}}_n^{(i,j)}) < 0) \cdot \widetilde{\boldsymbol{X}}_n^{(i,j)} \end{cases}, \tag{7}$$

where $\boldsymbol{X}_n^{(i+1,2j)}$ and $\boldsymbol{X}_n^{(i+1,2j+1)}$ are the input representations for the two child nodes of $V^{(i,j)}$ respectively, and $\mathbb{1}$ represents an indicator function.

Network Configuration. Throughout the whole paper, we adopt the same network structure as shown in Figure 2. The depth of the decision tree is set as 3. Correspondingly, each CNB is a three-layered neural network – 20 convolution kernels in the first layer, 20 in the second and 40 in the third. The kernel size is set as 5×5 for the 1st and 2nd layer, and 3×3 for the last layer, respectively. The nonlinearity function $\sigma(\cdot)$ in Eqn.(6) is defined as ReLu for all the convolution layers. Each convolutional layer is followed by a max-pooling operator with pooling size 2×2 and pooling stride 2×2. To regulate over-fitting, we adopt momentum, ℓ-2 norm regularization and dropout in the learning process. The momentum is set as 0.5, and linearly increased to 0.9 within 50 iterations. Dropout is adopted at each layer, and the dropout rate is 0.5 for multi-PIE and 0.2 for Occluded LFW respectively. We adopt a smaller drop rate for Occluded LFW since the number of training samples is much larger than that of multi-PIE, and the network suffers less from over-fitting. All the parameters (including those for the tree partitioning) are initialized by uniform sampling within the range [-0.1, 0.1]. The output feature maps of each neural branch are forwarded to a shared fully-connected layer L with 50 hidden units. The output of this layer is the final representation of input faces. An n-class softmax layer is then appended on the top for the given classification problem.

Computation Analysis. As the depth of our decision tree is fixed as 3, we have 4 leaf nodes in the final layer. Compared with the conventional CNN of the same structure as one CNB in Figure 2, the proposed network appears to contain more parameters. However, each input sample is only passed through one possible CNB. Namely, the computation complexity for each input sample is the same as the conventional single-model CNN. For fair comparisons, we increase the width of the single-model CNN so that it can have the same number of parameters as ours C the baseline CNN has 20 filters in the 1st layer, 40 in the 2nd and 160 in the 3rd. The runtime complexity of the i-th layer in c-CNN is $\frac{N^{(i-1)}}{2^{i-2}} \cdot \frac{N^{(i)}}{2^{i-1}} \cdot O(conv.)$, and the complexty of the CNN baseline is $N^{(i-1)} \cdot N^{(i)} \cdot O(conv.)$, where $O(conv.)$ is the complexity of the convolution operation of one kernel over one feature map, and $N^{(i)}$ is the number of kernels in layer i.

3.3. Joint Learning of MPT and CNN Branch

Different from prior works that learn features nodebynode in a decision tree [5, 1], the feature representation and the split function of all nodes are jointly learnt with regard to a unified objective as

$$\mathcal{L} = \sum_n \mathcal{J}(\boldsymbol{x}_n, y_n) + \beta \sum_i \sum_j \mathcal{L}^{(i,j)}, \tag{8}$$

where the first term represents the softmax loss for the nclass classification problem, and the second term is the node-wise loss defined in Eqn.(5), and β is a scaling factor.

The network is optimized via back propagation with Stochastic Gradient Descent method. For a node $V^{(i,j)}$, the network needs to update 4 parameters – $\boldsymbol{P}^{(i,j)}$, $\tau^{(i,j)}$, $\boldsymbol{W}^{(i,j)}$ and $b^{(i,j)}$. The gradient w.r.t. each parameter is given in details in the following. It is noted that the optimization is conducted in a batch-wise manner. To be more specific, we use the partition parameters $\boldsymbol{P}^{(i,j)}$ and $tau^{(i,j)}$ learnt with the previous data batch to split samples in the present batch. In this way, the dynamic routing of samples is determined before updating the parameters in the present batch iteration. To compute the gradient w.r.t. $\boldsymbol{W}^{(i,j)}$ and $b^{(i,j)}$, we need to derive the gradient w.r.t. $\widetilde{\boldsymbol{X}}_n^{(i,j)}$ first,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \widetilde{\boldsymbol{X}}_n^{(i,j)}} = \frac{\partial \mathcal{L}}{\partial \widetilde{\boldsymbol{X}}_n^{(i+1,2j)}} \cdot \mathbb{1}(\varphi(\widetilde{\boldsymbol{X}}_n^{(i,j)}) \geq 0) + \\ \frac{\partial \mathcal{L}}{\partial \widetilde{\boldsymbol{X}}_n^{(i+1,2j+1)}} \cdot \mathbb{1}(\varphi(\widetilde{\boldsymbol{X}}_n^{(i,j)}) < 0) + \beta \frac{\partial \mathcal{L}^{(i,j)}}{\partial \widetilde{\boldsymbol{X}}_n^{(i,j)}}. \end{aligned} \tag{9}$$

Based on Eqn.(9), $\frac{\partial \mathcal{L}}{\partial \boldsymbol{X}_n^{(i,j)}}$, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_n^{(i,j)}}$ and $\frac{\partial \mathcal{L}}{\partial b_n^{(i,j)}}$ can be easily derived similarly as standard CNN with the chain rule.

The splitting parameters can be updated as follows,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}^{(i,j)}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{X}_n^{(i+1,\cdot)}} \cdot \frac{\partial \boldsymbol{X}_n^{(i+1,\cdot)}}{\partial \boldsymbol{P}^{(i,j)}} + \beta \frac{\partial \mathcal{L}^{(i,j)}}{\partial \boldsymbol{P}^{(i,j)}}, \quad (10)$$

$$\frac{\partial \mathcal{L}}{\partial \tau^{(i,j)}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{X}_n^{(i+1,\cdot)}} \cdot \frac{\partial \boldsymbol{X}_n^{(i+1,\cdot)}}{\partial \tau^{(i,j)}} + \beta \frac{\partial \mathcal{L}^{(i,j)}}{\partial \tau^{(i,j)}}. \quad (11)$$

Since $\frac{\partial \boldsymbol{X}_n^{(i+1,\cdot)}}{\partial \boldsymbol{P}^{(i,j)}}$ and $frac \partial \boldsymbol{X}_n^{(i+1,\cdot)} \partial \tau^{(i,j)}$ are all zeros, the gradients are actually determined by the tree node loss $\mathcal{L}^{(i,j)}$, i.e.,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}^{(i,j)}} = \beta \frac{\partial \mathcal{L}^{(i,j)}}{\partial \boldsymbol{P}^{(i,j)}}, \quad (12)$$

$$\frac{\partial \mathcal{L}}{\partial \tau^{(i,j)}} = \beta \frac{\partial \mathcal{L}^{(i,j)}}{\partial \tau^{(i,j)}}. \quad (13)$$

To simplify the problem, τ can be set as the mean value of samples after projections, i.e., $\varphi(\boldsymbol{x}) = \boldsymbol{x}^T \cdot \boldsymbol{P}^{(i,j)}$, such that there are only three parameters to optimize.

4. Experiments

Our method is evaluated with two problems: 1) multiview face identification on Multi-PIE dataset [8] and 2) occluded face verification on Labeled face in the Wild (LFW) [10] with synthetic occlusions. The proposed c-CNN is built on a basic assumption that the modality information is unknown for both training and testing. Therefore, we do not include the comparison with some existing methods using the specific modality information of each sample. Experimental results are analyzed in details in the following subsections.

4.1. Experiment Settings

On both datasets, we use the same network configuration as shown in Sec.3.2. The implementation of c-CNN is based on Theano[1] and Pylearn2[2]. The supervised cost $\mathcal{J}(\cdot)$ in both experiments is the negative likelihood of an n class softmax function, and thus n is set as 150 and 2 for multi-PIE and occluded LFW respectively. With more classes, the initial cost is much larger in scale. To balance the relative effect of the supervised cost and tree node cost, β is set to 5 and 1 accordingly. As most CNNs are optimized with batch-based SGD, the tree node loss in Eqn.(5) is only defined in batches. Thus, larger batch size can lead to better results. In this paper, all the experiments are conducted with GTX TiTan GPU with 3GB memory. Due to the memory limit, we set the batch size as 1,000 in the following experiments.

[1] http://deeplearning.net/software/theano/

[2] http://deeplearning.net/software/pylearn2/

4.2. Multi-View Face Identification

We evaluate the performance of c-CNN in multi-view face identification on Multi-PIE. It contains images of 337 identities with 20 illumination levels and 15 poses ranging from $-90°$ to $+90°$. The database is arranged in four sessions, and we evaluate our method on Session 1 only, which includes faces of 250 subjects. Previous experiments reported on MultiPIE are usually conducted on faces with small poses ($-45°$ to $+45°$). However, our method is testified on faces under all poses. We follow a similar evaluation protocol as in [31]. For training, we utilize all the images (15 poses, 20 illumination levels) of the first 150 identities. For testing, one frontal image with neutral illumination marked as ID 07 is chosen as the gallery image for each of the remaining 100 subjects. The remaining images are used as probes. The average precision is reported with regard to pose in Table 1 for comparison.

Four methods are included for comparison in this subsection. Fisher Vector [24] is built on hand-crafted features, i.e., SIFT and LBP in this experiment. Both FIP [31] and CNN_40 are deep learning based methods. We include the results of FIP with two network configurations. FIP_20 has exactly the same number of convolution kernels as one CNB in c-CNN. FIP_40 is included to show the improvements of c-CNN over the network with the same total number of parameters. In particular, FIP_40 has 20 kernels in layer 1, 40 in layer 2 and 160 in layer 3. FIP is a reconstruction based approach, and thus requires the frontal view and neural illumination for each image during training. In this experiment, we apply PCA on features of the last convolution layer such that the final dimension is the same as c-CNN. CNN_40 is a single CNN network with the same configuration of convolutional layers as FIP_40. Note that although our network has approximately the same number of parameters, the computation cost is much lower as analyzed in Sec. 3.2. Clearly, c-CNN achieves the best performance, especially for large poses, such as $\pm 90°$ and $\pm 90°$. The improvements can be up to nearly 10%. Different from FIP which requires frontal images for each subject, we do not utilize any pose information and our method still reaches higher accuracy. Moreover, c-CNN outperforms both CNN_40 and FIP_40 while maintaining a much lower computation cost. Moreover, we include two extra baselines – Cluster_CNN and Tree_CNN. Cluster_CNN firstly clusters the samples based on LBP features and trains a separate CNN for each cluster. Tree_CNN follows the c-CNN structure, but optimizes the branching parameters w.r.t. the node-wise loss first, and then learns the parameters of CNN while fixing the branching parameters. The improvement brought by c-CNN demonstrates the effectiveness of joint optimization over filters and tree branching. In this subsection, we also explore the possibility of extending tree to forest as shown by c-CNN Forest. For this approach, we include 3 trees with

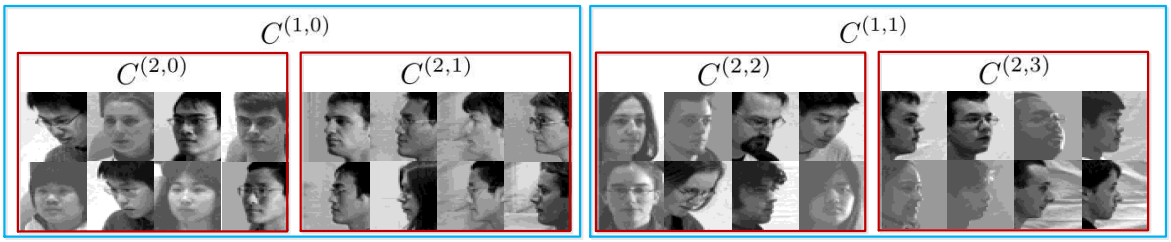

Figure 3. Partitioned samples of multi-PIE in leaf nodes. The blue boxes represent the tree nodes in the second layer, and the red ones stand for those in the third layer. The node notations are given inside the corresponding boxes. Clearly, samples of similar modalities (poses) are prone to be passed into the same nodes.

	Avg.	$\pm 90^\circ$	$\pm 75^\circ$	$\pm 60^\circ$	$\pm 45^\circ$	$\pm 30^\circ$	$\pm 15^\circ$	pose
Fisher Vector [24]	66.60	24.53	45.51	68.71	80.33	87.21	93.30	×
FIP_20 [31]	67.87	34.13	47.32	61.64	78.89	89.23	95.88	√
FIP_40 [31]	70.90	31.37	49.10	69.75	85.54	92.98	96.30	√
CNN_40	70.81	32.08	47.79	69.48	85.99	93.04	96.60	×
Cluster_CNN	69.87	36.80	47.36	68.20	82.43	90.67	93.75	×
Tree_CNN	71.16	39.90	50.29	67.21	83.63	91.31	94.66	×
c-CNN	73.54	41.71	55.64	70.49	85.09	92.66	95.64	×
c-CNN Forest	76.89	47.26	60.66	74.38	89.02	94.05	96.97	×

Table 1. Comparisons of precision (%) with some prior methods on multi-PIE for different poses. The last column indicates the dependency on head pose information.

Figure 5. Examples in Occluded LFW. Six categories of occlusions are synthesized for each image, including hair, hand, mask, mustache, painting and glass.

β = 5, 7 and 10 respectively. In c-CNN Forest, we take the average of the cosine distance matrices of the derived corresponding feature vectors. As can be observed in the table, the performance is further improved by more than 3%. Further randomization on parameters and bagging in the forest are expected to produce better results.

In addition, we illustrate some of the samples in each leaf node in Figure 3. Without any human intervention, the proposed method automatically discovers the inherent modality of the data (pose in this experiment) and clusters samples with similar poses into corresponding leaf nodes. Since the intermediate representation and splitting projections are jointly optimized w.r.t. Eqn.(5), the acquired clusters rarely contain noisy samples.

4.3. Face Verification with Various Occlusions

We evaluate c-CNN with occluded face verification on a synthesized dataset from Labeled Face in the Wild (LFW) [10] occluded LFW. LFW is a standard database collected to evaluate benchmark algorithms for face verification. It contains 13,000 images of 5,749 individuals downloaded from the Internet. We follow the image-restricted protocol of LFW. All the algorithms are evaluated with 6,000 pre-defined image pairs. The data are divided into 10 mutually excluded folds. In each experiment, data of only one fold are used for testing, and the remaining 9 folds are used for training.

In occluded LFW, each face image of LFW is synthesized with 6 kinds of occlusions, including hair, hand, mask, mustache, painting and glass. Each category includes 16 images occluded by the corresponding object. We crop the occlusion objects from a large collection of images from the Internet. Afterward the occlusions of objects are appended on the face images with reference to the detected landmarks. Some examples of the occluded faces are illustrated in Figure 5. Due to the large size of the dataset, we use a subset to evaluate the proposed network. In particular, for each image within a pair in standard protocol, we randomly sample 8 occluded images. The resulting two groups of images

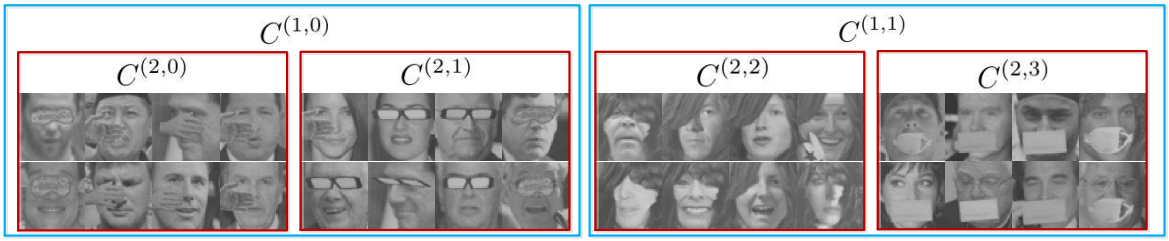

Figure 4. Partitioned samples of occluded LFW in leaf nodes. The blue boxes represent the tree nodes in the second layer, and the red ones stand for those in the third layer. Clearly, samples of similar modalities (occlusion categories and positions) are prone to be passed into the same nodes.

	1	2	3	4	5	6	7	8	9	10	Avg.
HDLBP [2]	69.77	68.79	66.39	69.09	67.45	66.89	67.70	67.26	66.71	69.85	67.99
Fisher Vector [24]	70.83	72.90	73.21	72.83	71.80	73.44	73.33	72.29	72.96	73.29	72.68
PEM [16]	62.87	65.08	65.44	63.17	62.70	65.50	63.08	61.58	64.46	63.81	63.76
CNN_20	74.40	73.12	71.69	72.94	71.38	74.65	72.63	74.63	71.27	72.40	72.91
CNN_40	75.40	73.83	74.12	73.30	72.74	76.20	72.36	76.20	71.43	73.50	73.90
c-CNN	77.63	75.09	75.00	75.03	73.69	76.55	76.16	76.85	74.80	74.43	75.52
c-CNN Forest	77.65	75.16	75.00	76.17	73.71	77.67	77.27	77.81	76.10	75.83	76.24

Table 2. Comparisons of precision (%) with some prior methods on occluded LFW for ten folds.

are then randomly combined to form 8 occluded pairs. This procedure is conducted for each fold.

Five baselines are included for comparison in this set of experiments. The results are reported on each fold in terms of the average precision in Table 2. HDLBP [2], Fisher Vector [24] and PEM [16] are implemented with hand-crafted features. The aforementioned methods follow the same training protocols (with no outside data) for fair comparison. We also include the single CNN based methods with the same network structure as one neural branch, i.e., CNN_20. As shown in the table, c-CNN demonstrates consistent improvements over CNN_20 and CNN_40, up to 3.5%. The significant improvements over CNN_20 can better demonstrate the superiority of the proposed method, since the two methods are of comparable computation cost. The improvements brought by c-CNN are further analyzed by showing some of the examples of corrected image pairs in Figure 6. Compared with modality-unaware CNN, c-CNN is more capable of modeling the intra-class similarities across different modalities. The synthesized data are very challenging due to the large occlusion area on the faces, thus most manually designed features result in low precision. By including deep feature learning, c-CNN outperforms HDLBP, Fisher Vector and PEM on all the folds. As for the extension to the forest structure, we include 3 trees with $\beta = 0.7$, 1.0 and 1.2 respectively. The final score for each sample is computed as the maximum among the scores of each tree. The resulting performance is further improved by around 0.7%.

Some of the examples in each leaf node are illustrated in Figure 4. With the exactly same setting as in multi-view face identification experiment, c-CNN discovers the inherent modality of input samples accordingly. It shows that the modality information is learnt as the occlusion type and position in this experiment.

Figure 6. Exemplars of the corrected image pairs by c-CNN.

5. Conclusions and Future Work

We proposed a conditional Convolutional Neural Network to address cross-modality face recognition. By introducing conditional routing, c-CNN explores the hidden modalities of samples and learns the modality-specific features while maintaining a low computation cost. Both the conditional routing and the feature extraction are learnt optimally with the direct guidance of a unified loss. We evaluate c-CNN with decision tree on two cross-modality classification problems. In both experiments, c-CNN demonstrates consistent improvements. As a generic framework in handling cross-modalities, c-CNN can be easily applied in various research fields and we are expecting similar results as those in this paper. Moreover, the decision tree based approach is a simplified case of c-CNN, which divides the convolutional kernels into mutually exclusive sets. In future, we shall pursue a more generic c-CNN that enables flexible (soft) assignments of convolution kernels in each layer.

References

[1] S. R. Bulo and P. Kontschieder. Neural decision forests for semantic image labelling. In *CVPR*, 2014. 3, 5

[2] 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. 8

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[4] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *STOC*, 2008. 4

[5] S. R. Fanello, C. Keskin, P. Kohli, S. Izadi, J. Shotton, A. Criminisi, U. Pattacini, and T. Paek. Filter forests for learning data-dependent convolutional kernels. In *CVPR*, 2014. 3, 5

[6] P. Geladi and B. R. Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 1986. 1

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2, 3

[8] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 2010. 6

[9] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 2004. 1

[10] 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. 6, 7

[11] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 1994. 3

[12] M. Kan, S. Shan, H. Chang, and X. Chen. Stacked progressive auto-encoders (spae) for face recognition across poses. In *CVPR*, 2014. 2, 3

[13] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen. Multi-view discriminant analysis. In *ECCV*. 2012. 2

[14] T.-K. Kim and J. Kittler. Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *TPAMI*, 2005. 2

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[16] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic matching for pose variant face verification. In *CVPR*, 2013. 8

[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. 3

[18] M. Lin, Q. Chen, and S. Yan. Network in network. In *ArXiv e-prints*, 2013. 2, 3

[19] C. Liu, H.-Y. Shum, and W. T. Freeman. Face hallucination: Theory and practice. *IJCV*, 2007. 1, 2

[20] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2

[21] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 2002. 2

[22] A. Sharma, M. Al Haj, J. Choi, L. S. Davis, and D. W. Jacobs. Robust pose invariant face recognition using coupled latent space discriminant analysis. *CVIU*, 2012. 2

[23] A. Sharma and D. W. Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *CVPR*, 2011. 2

[24] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *BMVC*, 2013. 6, 7, 8

[25] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014. 2, 3

[26] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *ICCV*, 2005. 3

[27] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *TPAMI*, 2009. 1, 2

[28] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *ICCV*, 2007. 3

[29] X. Zhang, Y. Gao, and M. K. Leung. Automatic texture synthesis for face recognition from single views. In *ICPR*, 2006. 2

[30] X. Zhao, T.-K. Kim, and W. Luo. Unified face analysis by iterative multi-output random forests. In *CVPR*, 2014. 3

[31] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *ICCV*, 2013. 2, 3, 6, 7

[32] 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. 2, 3