Merged Multi-CNN with Parameter Reduction for Face Attribute Estimation

Hiroya Kawai, Koichi Ito and Takafumi Aoki
Graduate School of Information Sciences, Tohoku University, Japan.
hiroya@aoki.ecei.tohoku.ac.jp

Abstract

This paper proposes a face attribute estimation method using Merged Multi-CNN (MM-CNN). The proposed method merges single-task CNNs into one CNN by adding merging points and reduces the number of parameters by removing the fully-connected layers. We also propose a new idea of reducing parameters of CNN called Convolutionalization for Parameter Reduction (CPR), which estimates attributes using only convolution layers, in other words, does not need any fully-connected layers to estimate attributes from extracted features. Through a set of experiments using the CelebA and LFW-a datasets, we demonstrated that MM-CNN with CPR exhibits higher efficiency of face attribute estimation than conventional methods.

1. Introduction

Face recognition is the most attractive research topic in the field of biometric recognition, pattern recognition and computer vision [13]. Automated human face recognition is one of the most difficult problems in biometric recognition, since its performance is significantly decreased due to pose changes, expression changes, illumination changes, low-resolution images, human motion, etc. Hence, a large number of face recognition methods have been proposed even now [32, 1]. Recent Convolutional Neural Network (CNN)-based approaches have had a significant impact also on face recognition [27]. The performance of face recognition has been dramatically improved by CNN, while more performance improvement is required from the viewpoint of practical use compared with fingerprint and iris recognition.

One of performance improvement approaches for face recognition is to use face attributes. A face has a lot of attribute information such as age, gender, ethnicity, hair color, nose size, mouth shape, etc., which can be used as low- or mid-level features in face recognition. The performance of face recognition can be improved by screening face images using face attributes before face recognition or by combining face attributes with face features in face recognition, etc. In addition, face attributes are useful in human-computer interface, video surveillance, criminal investigation, etc. [21, 5]. Therefore, face attribute estimation is one of the important topics in face recognition.

978-1-7281-3640-0/19/$31.00 ©2019 IEEE

The general processing pipeline of face attribute estimation consists of face detection, feature extraction and classification [12, 21]. In traditional approaches, handcrafted local features such as Scale-Invariant Feature Transform (SIFT) [15] and Local Binary Patterns (LBPs) [17] are used in feature extraction and Support Vector Machine (SVM) is used in classification. A few types of attributes may be classified using such traditional approaches, while it is difficult to design local features so as to classify many attributes. Addressing this problem, CNN-based methods have been proposed to estimate face attributes from face images [30, 14, 31, 8, 28, 5, 4, 3] because of their significant impact on image recognition and the availability of large-scale face image databases with annotated labels.

This paper proposes a novel CNN architecture specially designed for multi-task processing such as face attribute estimation. The proposed CNN consists of 5-stage convolution blocks for each attribute, where convolution blocks are connected to each other for each stage so as to improve the estimation accuracy. We also propose a new idea of reducing parameters of CNN called Convolutionalization for Parameter Reduction (CPR), which estimates attributes using only convolution layers, in other words, does not need any fully-connected layers to estimate attributes from extracted features. Through a set of experiments using the CelebA and LFW-a datasets, we demonstrate that the proposed method exhibits the efficient performance on face attribute estimation compared with conventional methods in terms of the small size of architecture and the estimation accuracy.

2. Related Work

Table 1 shows a summary of face attribute estimation methods. Kumar et al. [12] proposed one of famous face attribute estimation methods using handcrafted local features. This method extracts pixel values from grayscale,

Table 1: A summary of face attribute estimation methods.

Method	Feature extraction	Classifier
Kumar et al. [12]	Pixel value (gray, RGB and HSV), edge magnitude and orientation	One SVM for each attribute
Zhang et al. [30]	Pose Aligned Networks (4 conv and 1 fc) for Deep Attribute modeling (PANDA)	One linear SVM for each attribute
Liu et al. [14]	LNet (5 conv) for face localization and ANet (4 conv) for face attribute prediction	One linear SVM for each attribute
Zhong et al. [31]	FaceNet [22] and VGG [24]	One linear SVM for each attribute
Huang et al. [8]	DeepID2 [25]	Large Margin Local Embedding (LMLE)-kNN
Wang et al. [28]	Siamese network (2 conv and 7 inception and 1 fc)	Cross entropy
Ehrlich et al. [2]	Multi-Task Restricted Boltzmann Machines (MT-RBMs) with PCA and facial landmark detectors	Multi-task classifier
Hand et al. [5]	Multi-task deep Convolutional Neural Network (MCNN) (3 conv and 2 fc)	Multi-task classifier using an AUXilirary network (AUX)
Gao et al. [3]	ATNet, ATNet_G, ATNet_GT (4 conv and 3 fc)	Softmax
Han et al. [4]	AlexNet-like CNN [11] (5 conv and 4 fc)	Softmax
Proposed	Merged Multi-CNN (MM-CNN) (5 conv and 1 fc)	Softmax
	MM-CNN with CPR (5 conv)	Softmax

RGB and HSV color spaces and edge magnitude and orientation as features and classifies them into face attributes using an SVM for each attribute. After this work, most of methods have employed a CNN-based approach due to its excellent performance on image classification.

Zhang et al. [30] proposed Pose Aligned Networks for Deep Attribute modeling (PANDA), which consists of feature extraction by CNNs with poselet detection and attribute prediction by a linear SVM for each attribute. Liu et al. [14] proposed two CNN architectures: LNet for face localization and ANet for face attribute prediction with a linear SVM for each attribute. Zhong et al. [31] proposed feature extraction by FaceNet [22] and VGG [24] and attribute prediction by a linear SVM for each attribute. Both approaches [14, 31] estimate attributes by inputting features to SVM.

Huang et al. [8] proposed Large Margin Local Embedding (LMLE)-kNN for predicting face attributes, which uses feature extraction by DeepID2 [25]. Wang et al. [28] proposed a GoogLeNet-like CNN architecture, which consists of 3 parts: face recognition, weather prediction and location estimation. Face attributes are estimated from concatenated features in the fully-connected layers. The approach of using a single CNN for feature extraction and classification is used in most CNN-based methods because of its high performance and simple training. Such methods modify the training procedure or the CNN architecture in order to process multiple tasks in a single CNN.

In recent, multi-task learning approaches have been used in face attribute estimation. Ehrlich et al. [2] proposed Multi-Task Restricted Boltzmann Machines (MT-RBMs) with PCA and facial landmark detectors. Hand et al. [5] proposed Multi-task deep Convolutional Neural Network with an AUXilirary network (MCNN-AUX). Gao et al. [3] proposed 3 multi-task CNNs: ATNet, ATNet_G and ATNet_GT, which are designed according to multiple clusters obtained by classifying face attributes using the k-means algorithm. Han et al. [4] proposed a method of multi-task learning of CNNs using labels determined by their own rule in light of correlation among face attributes.

One of performance improvement approaches is to design the CNN architecture dedicated to multi-task process. CNN architectures are designed for multi-task process by making some groups by manually or automatically classifying face attributes sharing parameters for each groups in convolution layers. Another approach of performance improvement is to use one CNN for one attribute, which may obtain the best performance on face attribute estimation, while the drawback is its enormous parameters. The number of parameters can be decreased by eliminating the number of channels of CNNs, while the performance of feature extraction is degraded. Addressing the above problem, the proposed method keeps the performance of feature extraction on multiple CNNs by merging independent CNNs into a single multi-task CNN. The proposed method also introduces a novel parameter reduction approach of eliminating the fully-connected layers, which can dramatically reduce

Table 2: Correspondence between attribute and its index in the CelebA and LFW-a datasets.

Idx.	Attribute	Idx.	Attribute
1	5 O'Clock Shadow	21	Male
2	Arched Eyebrows	22	Mouth Slightly Open
3	Attractive	23	Mustache
4	Bags Under Eyes	24	Narrow Eyes
5	Bald	25	No Beard
6	Bangs	26	Oval Face
7	Big Lips	27	Pale Skin
8	Big Nose	28	Pointy Nose
9	Black Hair	29	Receding Hairline
10	Blond Hair	30	Rosy Cheeks
11	Blurry	31	Sideburns
12	Brown Hair	32	Smiling
13	Bushy Eyebrows	33	Straight Hair
14	Chubby	34	Wavy Hair
15	Double Chin	35	Wearing Earrings
16	Eyeglasses	36	Wearing Hat
17	Goatee	37	Wearing Lipstick
18	Gray Hair	38	Wearing Necklace
19	Heavy Makeup	39	Wearing Necktie
20	High Cheekbones	40	Young

the number of parameters of CNNs with a little degradation of estimation accuracy.

An additional approach is to improve the training procedure like [4], although we do not focus on this topic, since our main topic in this paper is CNN architectures for face attribute estimation. This approach can be applied to our method for further performance improvement.

3. Proposed Method

This section describes the proposed CNN architecture, which is specially designed for face attribute estimation, and the proposed parameter reduction method. In the following, we use 40 attribute indices and descriptions commonly used in the CelebA and LFW-a datasets as shown in Table 2, which was created by Liu et al. [14].

3.1. Merged Multi-CNN (MM-CNN)

We propose a novel CNN specially designed for multi-task processes. Our CNN architecture is based on Multiple CNNs (Multi-CNN), which consists of one CNN for one task, as shown in Fig. 1 (a). Each CNN has 5 convolution blocks and 1 fully-connected layer. The convolution block consists of a convolution layer and a batch normalization layer. A max pooling layer is included in the 1st and 2nd blocks. The output feature map of the 5th block is input

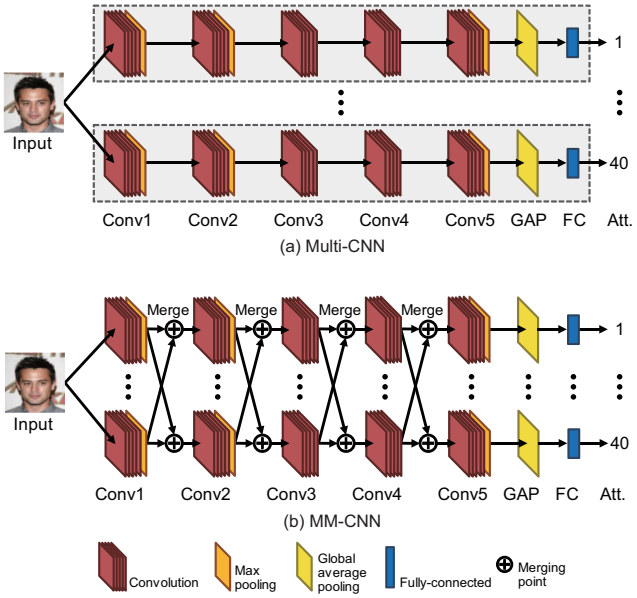

Figure 1: Overview of network architectures: (a) Multi-CNN and (b) MM-CNN.

to the fully-connected layer through global average pooling (GAP).

Fig. 1 (b) shows the proposed CNN called Merged Multi-CNN (MM-CNN) and Table 3 show the detailed architecture of MM-CNN. We add the merging points after the 1st to 4th convolution blocks of Multi-CNN and connect all the convolution blocks and all the merging points in the same stage. All the output feature maps are aggregated into one feature map by the merging function and the aggregated feature map is input to the subsequent convolution block. This paper employs two merging functions: Concatenation (Conc) and addition (Add). Fig. 2 shows examples of two merging functions. The Conc function concatenates feature maps towards channels. If we do not take care of an activation function, normalization and weights, the Conc function is the same as normal convolution. The Add function adds feature maps for each channel. If we do not take care of an activation function, normalization and weights, the Add function is the same as copying, concatenating and convolving feature maps. We also introduce the hyperparameter c in order to control the network size. c is defined by the number of output channels in convolution layers included in each convolution block. Large value of c means that the network has a lot of weight parameters. In this paper, we set c to the integer value from 5 to 60.

3.2. Convolutionalization for Parameter Reduction (CPR)

In the general flow of CNN, feature maps are extracted in convolution layers, pooling layers, etc. and a classifica-

Table 3: Detailed network architecture of MM-CNN.

Type	Kernel Size	Stride	Output ch. (Add)	Output ch. (Conc)
conv1	7×7	2	c	c
norm1			c	c
pool1	3×3	2	c	c
merge1			c	$c \times 40$
conv2	5×5	1	$2 \times c$	$2 \times c$
norm2			$2 \times c$	$2 \times c$
pool2	3×3	1	$2 \times c$	$2 \times c$
merge2			$2 \times c$	$2 \times c \times 40$
conv3	3×3	1	$2 \times c$	$2 \times c$
norm3			$2 \times c$	$2 \times c$
merge3			$2 \times c$	$2 \times c \times 40$
conv4	3×3	1	$2 \times c$	$2 \times c$
norm4			$2 \times c$	$2 \times c$
merge4			$2 \times c$	$2 \times c \times 40$
conv5	3×3	1	$1,000$	$1,000$
norm5			$1,000$	$1,000$
GAP			$1,000$	$1,000$
dropout (30%)			$1,000$	$1,000$
fc			2	2
softmax			2	2

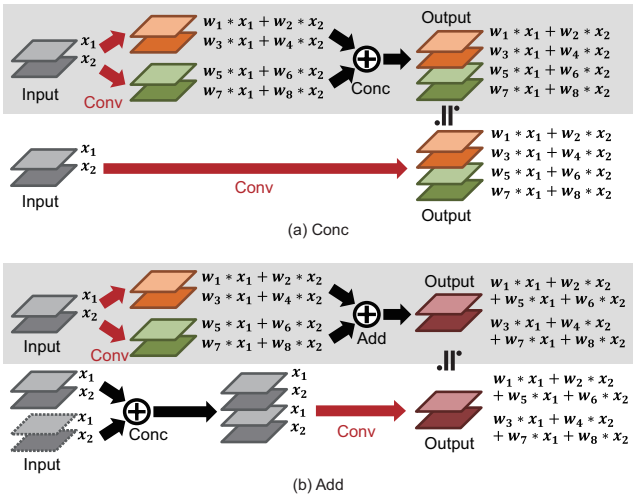

Figure 2: Example of the merging function: (a) Conc and (b) Add.

tion score or a regression value is obtained from the output of fully-connected layers. Early CNNs [11, 24] employ 3 fully-connected layers in classification as shown in Fig. 3 (a). The use of some fully-connected layers significantly increases the number of parameters, since the number of parameters in the fully-connected layers is much more than that in the convolution layers.

The state-of-the-art methods [6, 26, 7, 9] in image recognition employ the approach of aggregating feature maps extracted from convolution layers using GAP and inputting

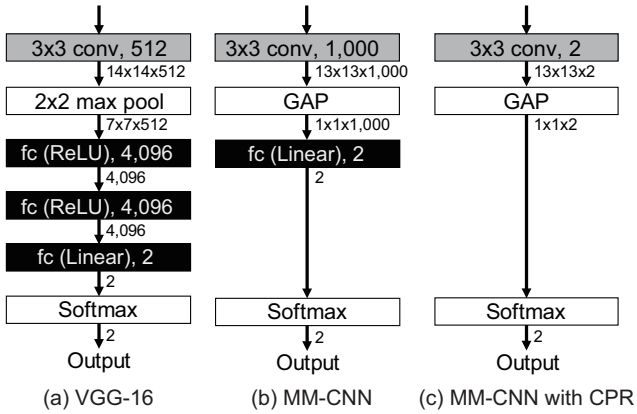

Figure 3: Network architectures for classification: (a) VGG-16 [24], (b) MM-CNN and (c) MM-CNN with CPR.

the aggregated feature map to one fully-connected layer as shown in Fig. 3 (b). Therefore, the number of fully-connected layers in the recent CNNs is decreased. There are some CNNs without fully-connected layers for semantic segmentation and image generation [18, 19, 23]. Sandler et al. [20] removed all the fully-connected layers by replacing them into 1×1 convolution. Their purpose is to speed-up the CNN computation and implement the CNN onto GPUs, and the process and the number of parameters are the same in the CNN with fully-connected layers. There is no CNN without fully-connected layers proposed for the purpose of reducing the number of parameters to the best of our knowledge.

We propose Convolutionalization for Parameter Reduction (CPR), which removes all the fully-connected layers in CNN, for reducing the number of parameters. CPR controls the number of outputs in the final convolution layer so as to correspond to the number of classes and aggregates the output feature maps into the binary score by GAP. The score is output after applying the softmax function as shown in Fig. 3 (c). Table 4 shows comparison of the number of parameters in MM-CNN with and without CPR, where the merging function is the addition and $c = 30$. MM-CNN without CPR reduces the number of parameters in the fully-connected layer by using GAP as shown in Fig. 3 (b). In this case, the number of output channels in the final convolution layer must be increased so as to keep the performance of CNN, resulting in an increasing number of parameters. On the other hand, CPR reduces the number of parameters by reducing the number of output channels in the final convolution layer.

4. Experiments and Discussion

This section describes the performance evaluation of face attribute estimation methods using the CelebA [14] and

Table 4: Comparison of the number of parameters in MM-CNN with and without CPR.

	MM-CNN w/o CPR		MM-CNN w/ CPR	
Type	# of params	Ratio	# of params	Ratio
conv1	177,600	0.7%	177,600	3.9%
conv2	1,802,400	6.9%	1,802,400	39.0%
conv3	1,298,400	4.9%	1,298,400	28.1%
conv4	1,298,400	4.9%	1,298,400	28.1%
conv5	21,640,000	82.3%	43,280	0.9%
fc	80,080	0.3%	—	—
Total	26,296,880	100%	4,620,080	100%

LFW-a [29] datasets.

4.1. Databasets

The detail of both datasets is summarized as follows.
CelebA[*]: The CelebA dataset consists of 202,599 face images of 10,177 persons with 5 landmark locations (left and right eyes, nose, the left and the right of mouse) and 40 binary attributes. We use aligned face images by 5 landmarks in the experiments.
LFW-a[†]: The Labeled Faces in the Wild-a (LFW-a) dataset consists of 13,233 face images of 5,749 persons with 73 binary attributes from the LFW dataset [10]. Note that we use 40 attributes commonly used in the CelebA dataset.

4.2. Experimental Settings

The following is the experimental settings employed in this paper. As for the CelebA dataset, we employ the experimental protocol recommended in the CelebA dataset, where 182,637 images and the remaining 19,962 images are used for training and test, respectively, and 10% of training data is used to verify overfitting. As for the LFW-a dataset, we use 13,143 images having face attribute labels, where 6,263 images and the remaining 6,880 images are used for training and test, respectively, and 10% of training data is used to verify overfitting as well as the CelebA dataset. Note that the initial CNNs for LFW-a are pretrained using the training data of CelebA, since the number of training data for LFW-a is less than that for CelebA.

We describe the implementation details in the following. We use a cross-entropy loss for all the attribute scores in training. We perform Nesterov Accelerated Gradient (NAG) [16] in weight optimization. The number of epochs is set to 30 and the learning rate is 0.002. Note that the learning rate is decreased when the loss is not improved during 2 consecutive epochs, and the training is finished if the loss is not improved during 5 consecutive epochs. The input images are normalized to have 0 mean and 1 variance, are randomly flipped in the horizontal direction, and are resized to 227×227 pixels.

We compare the performance of the proposed method with that of 6 conventional methods: LNets+ANet [14], FaceNet+SVM [31], SiameseNet [28], MT-RBM [2], MCNN-AUX [5] and ATNet_GT [3]. We evaluate the performance of the proposed method for the two merging functions: Conc and Add. As for Add, we evaluate the performance for $c = 5, 10, 20, 30, 60$. Note that we only evaluate the performance of MM-CNN (Conc) with $c = 60$ for the CelebA dataset, since we emprically confirmed that the accuracy of MM-CNN (Conc) is lower than MM-CNN (Add) for all the values of c.

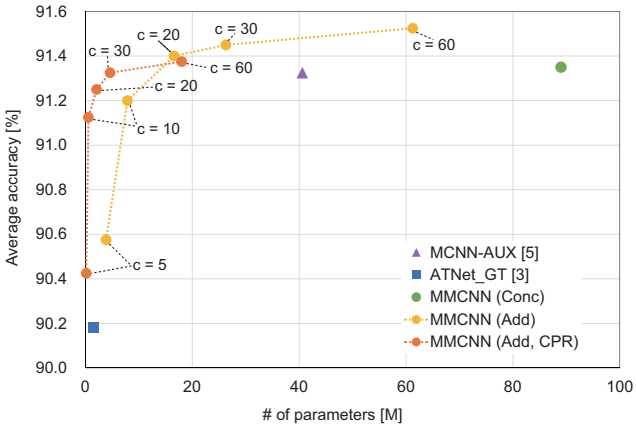

Figure 4: Efficiency of each method in face attribute estimation.

4.3. Experimental Results

Table 5 and 6 show the estimation accuracy for the CelebA and LFW-a datasets, respectively. The accuracy of the proposed methods is higher than that of conventional methods. Among them, MM-CNN (Add, $c = 60$) exhibits the highest estimation accuracy for 24 attributes in the CelebA dataset and 22 attributes in the LFW-a dataset. The most remarkable point about the proposed method is that the parameter efficiency of the proposed method is higher than that of MCNN-AUX [5], which is also the multi-task CNN architecture, although the estimation accuracy of the proposed methods is comparable with MCNN-AUX. Table 7 summarizes the estimation accuracy and the number of parameters for each method and Fig. 4 shows the efficiency of each method, where the horizontal axis is the number of parameters and the vertical axis is the average accuracy of face attribute estimation. The use of CPR exhibits higher efficiency when the number of parameters is less than 10M, comparing the efficiency between MM-CNN (Add) and MM-CNN (Add, CPR). The number of parameters for MM-CNN (Add, $c = 30$, CPR) is 1/10 for MCNN-

[*] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
[†] https://talhassner.github.io/home/projects/lfwa/

Table 5: Accuracy [%] of face attribute estimation for the CelebA dataset.

	Attribute index																				
Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
LNet+ANet [14]	91	79	81	79	98	95	68	78	88	95	84	80	90	91	92	99	95	97	90	87	
FaceNet+SVM [31]	89	83	82	79	96	94	70	79	87	93	87	79	87	88	89	99	94	95	91	87	
SiameseNet [28]	84	**87**	**84**	**87**	92	**96**	**78**	**91**	84	92	91	81	93	89	93	97	92	95	**96**	**95**	
MT-RBMs [2]	90	77	76	81	98	88	69	81	76	91	95	83	88	95	96	96	96	97	85	83	
MCNN-AUX [5]	**95**	83	83	85	**99**	**96**	71	85	**90**	**96**	**96**	**89**	**93**	**96**	96	**100**	**97**	**98**	92	88	
ATNet_GT [3]	92	81	81	84	**99**	**96**	71	83	89	95	**96**	87	92	94	96	99	**97**	**98**	90	86	
MM-CNN (Conc, $c = 60$)	**95**	84	83	85	**99**	**96**	72	84	90	**96**	**96**	**89**	**93**	**96**	96	**100**	**97**	**98**	92	88	
MM-CNN (Add, $c = 60$)	**95**	84	83	86	**99**	**96**	72	85	90	**96**	**96**	**89**	**93**	**96**	96	**100**	**97**	**98**	92	88	
MM-CNN (Add, $c = 60$, CPR)	94	84	83	85	**99**	**96**	72	84	90	**96**	**96**	**89**	**93**	**96**	**97**	**100**	**97**	**98**	92	88	
MM-CNN (Add, $c = 30$, CPR)	94	84	83	85	**99**	**96**	72	84	90	**96**	**96**	**89**	**93**	**96**	96	**100**	**97**	**98**	92	88	
MM-CNN (Add, $c = 30$, CPR)	94	84	83	85	**99**	**96**	72	84	90	**96**	**96**	**89**	**93**	**96**	96	**100**	**97**	**98**	92	88	
MM-CNN (Add, $c = 10$, CPR)	94	84	83	85	**99**	**96**	71	84	90	**96**	**96**	**89**	**93**	95	96	**100**	**97**	**98**	92	87	

	Attribute index																				
Method	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	Ave.
LNet+ANet [14]	98	92	95	81	95	66	91	72	89	90	96	92	73	80	82	**99**	93	71	93	87	87.3
FaceNet+SVM [31]	**99**	92	93	78	94	67	85	73	87	88	95	92	73	79	82	96	93	73	91	86	86.6
SiameseNet [28]	96	**97**	90	79	90	**79**	85	77	84	**96**	92	**98**	75	**85**	**91**	96	92	77	84	86	88.7
MT-RBMs [2]	90	82	**97**	86	90	73	96	73	92	94	96	88	80	72	81	97	89	87	94	81	87.0
MCNN-AUX [5]	98	94	**97**	87	**96**	76	**97**	77	**94**	95	**98**	93	**84**	84	90	**99**	**94**	87	**97**	88	91.3
ATNet_GT [3]	97	93	**97**	86	94	76	**97**	75	93	95	97	92	80	82	89	**99**	93	86	96	88	90.2
MM-CNN (Conc, $c = 60$)	98	94	**97**	87	**96**	76	**97**	77	**94**	95	**98**	93	**84**	84	90	**99**	**94**	87	**97**	88	91.4
MM-CNN (Add, $c = 60$)	98	94	**97**	**88**	**96**	77	**97**	**78**	**94**	95	**98**	93	**84**	84	90	**99**	**94**	**88**	**97**	**89**	**91.5**
MM-CNN (Add, $c = 60$, CPR)	98	94	**97**	**88**	**96**	76	**97**	77	**94**	95	**98**	93	**84**	84	90	**99**	**94**	87	**97**	88	91.4
MM-CNN (Add, $c = 30$, CPR)	98	94	**97**	87	**96**	76	**97**	77	**94**	95	**98**	93	**84**	84	90	**99**	**94**	87	**97**	88	91.3
MM-CNN (Add, $c = 10$, CPR)	98	94	**97**	87	**96**	76	**97**	77	93	95	**98**	93	83	83	89	**99**	**94**	87	**97**	88	91.1

Table 6: Accuracy [%] of face attribute estimation for the LFW-a dataset.

	Attribute index																				
Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
LNet+ANet [14]	**84**	82	**83**	**83**	88	88	75	81	90	97	74	77	82	73	78	**95**	78	84	95	**88**	
FaceNet+SVM [31]	77	**83**	79	**83**	91	**91**	78	83	90	97	**88**	76	83	75	80	91	83	87	95	**88**	
MCNN-AUX [5]	77	82	80	**83**	92	90	79	**85**	**93**	97	85	81	85	**77**	**82**	91	83	**89**	**96**	**88**	
MM-CNN (Add, $c = 60$)	79	82	81	**83**	92	**91**	79	**85**	91	97	**88**	**82**	**86**	75	**82**	93	**84**	**89**	95	**88**	
MM-CNN (Add, $c = 60$, CPR)	78	81	80	82	**93**	90	77	84	90	97	87	81	84	75	81	93	83	**89**	95	**88**	
MM-CNN (Add, $c = 30$, CPR)	78	81	80	**83**	92	**91**	**80**	83	91	97	87	**82**	85	75	81	93	83	**89**	95	**88**	
MM-CNN (Add, $c = 10$, CPR)	78	81	79	82	92	90	78	84	91	97	86	81	84	75	80	92	83	**89**	95	87	

	Attribute index																				
Method	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	Ave.
LNet+ANet [14]	**94**	82	92	81	79	74	84	80	85	78	77	91	76	76	94	88	**95**	88	79	**86**	83.9
FaceNet+SVM [31]	**94**	81	**94**	81	80	75	73	83	86	82	82	90	77	77	94	90	**95**	**90**	81	**86**	84.7
MCNN-AUX [5]	**94**	**84**	93	**83**	**82**	**77**	**93**	84	86	**88**	**83**	**92**	79	**82**	**95**	90	**95**	**90**	81	**86**	86.3
MM-CNN (Add, $c = 60$)	**94**	**84**	**94**	**83**	**82**	76	91	**85**	86	**88**	**83**	91	79	81	94	**91**	**95**	**90**	**83**	85	**86.4**
MM-CNN (Add, $c = 60$, CPR)	93	83	**94**	81	**82**	**77**	90	84	86	87	**83**	91	79	80	94	**91**	**95**	**90**	**83**	85	85.9
MM-CNN (Add, $c = 30$, CPR)	93	83	**94**	82	**82**	75	90	84	**87**	**88**	82	91	**80**	80	94	**91**	94	**90**	82	85	86.0
MM-CNN (Add, $c = 10$, CPR)	93	82	**94**	81	82	73	89	83	85	86	81	90	79	80	94	**91**	**95**	89	82	85	85.5

AUX [5], although the estimation accuracy of both methods is comparable. The number of parameters for MM-CNN (Add, $c = 5$, CPR) is 1/7 for ATNet_GT [3], although the estimation accuracy of both methods is comparable. As observed above, MM-CNN can control the estimation accuracy and the number of parameters by the hyper parameter c and CPR. We expect that the use of the MM-CNN with CPR makes it possible to solve the problems in multi-task learning and respond to the recent demands for implementing CNNs onto embedded devices.

5. Conclusion

This paper proposed a face attribute estimation method using Merged Multi-CNN (MM-CNN) with Convolutionalization for Parameter Reduction (CPR). The proposed method merges single-task CNNs into one CNN by adding merging points and reduces the number of parameters by

Table 7: Estimation accuracy [%] and the number of parameters for the CelebA dataset.

Method	Ave. acc.	# of params
MCNN-AUX [5]	91.33%	40.6M
ATNet_GT [3]	90.18%	1.5M
MM-CNN (Conc, $c = 60$)	91.35%	89.0M
MM-CNN (Add, $c = 60$)	91.53%	61.3M
MM-CNN (Add, $c = 30$)	91.45%	26.3M
MM-CNN (Add, $c = 20$)	91.40%	16.6M
MM-CNN (Add, $c = 10$)	91.20%	7.9M
MM-CNN (Add, $c = 5$)	90.58%	3.9M
MM-CNN (Add, $c = 60$, CPR)	91.38%	18.0M
MM-CNN (Add, $c = 30$, CPR)	91.33%	4.6M
MM-CNN (Add, $c = 20$, CPR)	91.25%	2.1M
MM-CNN (Add, $c = 10$, CPR)	91.13%	0.6M
MM-CNN (Add, $c = 5$, CPR)	90.43%	0.2M

removing the fully-connected layers. Merging improves the accuracy of face attribute estimation and CPR archives the compact CNN models. Through a set of experiments using the CelebA and LFW-a datasets, we demonstrated that MM-CNN with CPR exhibits higher efficiency of face attribute estimation than conventional methods. The performance of MM-CNN can be improved by changing basic CNN architectures, since MM-CNN is designed by merging CNNs of target classes into one CNN using the merging function. In addition, MM-CNN can be applied to other multi-task problems, since MM-CNN can be automatically designed according to the number of tasks inspired by the approach of MM-CNN, we will consider automatic design of CNN architectures for multi-task problem in future.

Acknowledgment

This work was supported, in part, by JSPS KAKENHI Grant Numbers 18H03253.

References

[1] C. Ding and D. Tao. A comprehensive survey on pose-invariant face recognition. *ACM Trans. Intell. Syst. Technol.*, 7(3):37:1–37:42, Feb. 2016. 1

[2] M. Ehrlich, T. Shields, T. Almaev, and M. Amer. Facial attributes classification using multi-task representation learning. *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pages 47–55, June 2016. 2, 5, 6

[3] D. Gao, P. Yuan, N. Sun, X. Wu, and Y. Cai. Face attribute prediction with convolutional neural networks. *IEEE Int'l Conf. Robotics and Biomimetics*, pages 1294–1299, Dec. 2017. 1, 2, 5, 6, 7

[4] H. Han, A. Jain, F. Wang, S. Shan, and X. Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 40(11):2597–2609, Nov. 2018. 1, 2, 3

[5] E. Hand and R. Chellappa. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. *Proc. the Thirty-First AAAI Conf. Artificial Intelligence*, pages 4068–4074, Feb. 2017. 1, 2, 5, 6, 7

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 770–778, June 2016. 4

[7] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861:1–9, Apr. 2017. 4

[8] C. Huang, Y. Li, C. Loy, and X. Tang. Learning deep representations for imbalanced classification. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 5375–5383, June 2016. 1, 2

[9] G. Huang, Z. Liu, L. Maaten, and K. Weinberger. Densely connected convolutional networks. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2261–2269, July 2017. 4

[10] G. Huang, M. Matter, H. Lee, and E. Miller. Learning to align from scratch. *Proc. Annual Conf. Neural Information Processing Systems*, pages 773–781, Dec. 2012. 5

[11] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Proc. Annual Conf. Neural Information Processing Systems*, pages 1–9, 2012. 2, 4

[12] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describable visual attributes for face verification and image search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(10):1962–1977, Oct. 2011. 1, 2

[13] S. Li and A. Jain. *Handbook of Face Recognition*. Springer, 2011. 1

[14] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the Wild. *Proc. Int'l Conf. Computer Vision*, pages 3730–3738, Dec. 2015. 1, 2, 3, 4, 5, 6

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Comput. Vision*, 60(2):91–110, Nov. 2004. 1

[16] Y. Nesterov. A method of solving a convex programming problem with convergence rate O(1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983. 5

[17] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer Vision Using Local Binary Patterns*. Springer, 2011. 1

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434:1–16, Nov. 2015. 4

[19] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention*, pages 234–241, Oct. 2015. 4

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 4510–4520, June 2018. 4

[21] W. Scheirer, N. Kumar, K. Ricanek, P. Belhumeur, and T. Boult. Fusing with context: A Bayesian approach to combining descriptive attributes. *Proc. Int'l Joint Conf. Biometrics*, Dec. 2011. 1

[22] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 815–823, June 2015. 2

[23] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 39(4):640–651, Apr. 2017. 4

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 4

[25] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. *Proc. Int'l Conf. Neural Information Processing Systems*, 2:1988–1996, Dec. 2014. 2

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–9, June 2015. 4

[27] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014. 1

[28] J. Wang, Y. Cheng, and R. Feris. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2295–2304, June 2016. 1, 2, 5, 6

[29] L. Wolf, T. Hassner, and Y. Taigman. Effective face recognition by combining multiple descriptors and learned background statistics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(10):1978–1990, Oct. 2011. 5

[30] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose aligned networks for deep attribute modeling. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1637–1644, June 2014. 1, 2

[31] Y. Zhong, J. Sullivan, and H. Li. Face attribute prediction using off-the-shelf CNN features. *Proc. Int'l Conf. Biometrics*, June 2016. 1, 2, 5, 6

[32] H. Zhou, A. Mian, L. Wei, D. Creighton, M. Hossny, and S. Nahavandi. Recent advances on singlemodal and multimodal face recognition: A survey. *IEEE Trans. Human-Machine Systems*, 44(6):701–716, Dec. 2014. 1