Rotating Your Face Using Multi-task Deep Neural Network

Junho Yim[1] Heechul Jung[1] ByungIn Yoo[1,2] Changkyu Choi[2] Dusik Park[2] Junmo Kim[1]
[1]School of Electrical Engineering, KAIST, South Korea
[2]Samsung Advanced Institute of Technology
{junho.yim, heechul, junmo.kim}@kaist.ac.kr
{byungin.yoo, changkyu_choi, dusikpark}@samsung.com

Abstract

Face recognition under viewpoint and illumination changes is a difficult problem, so many researchers have tried to solve this problem by producing the pose- and illumination- invariant feature. Zhu et al. [26] changed all arbitrary pose and illumination images to the frontal view image to use for the invariant feature. In this scheme, preserving identity while rotating pose image is a crucial issue. This paper proposes a new deep architecture based on a novel type of multitask learning, which can achieve superior performance in rotating to a target-pose face image from an arbitrary pose and illumination image while preserving identity. The target pose can be controlled by the user's intention. This novel type of multi-task model significantly improves identity preservation over the single task model. By using all the synthesized controlled pose images, called Controlled Pose Image (CPI), for the pose-illumination- invariant feature and voting among the multiple face recognition results, we clearly outperform the state-of-the-art algorithms by more than 4∼6% on the MultiPIE dataset.

1. Introduction

Recently, there have been significant advances in face recognition technologies, especially due to deep learning. Zhu et al. [26] proposed a deep model that can convert a face image with an arbitrary pose and illumination to a so-called canonical face image as if it is viewed from the front with a standard illumination. DeepFace [18] achieved a human-level performance in face recognition for the first time with a 97% recognition rate on very challengning LFW dataset [6], and this record has recently been updated by a more powerful deep learning method [17] achieving an impressive 99% recognition rate on LFW dataset.

One important challenge related to face recognition is changing the viewpoint of a face image or synthesizing a novel view while preserving the identity of the face. For instance, DeepFace also relies on a preprocessing stage that rotates the input face images to a canonical view. A recent work [27] that extends [26] can generate not only a canonical view but also many face images with arbitrary poses preserving the identity.

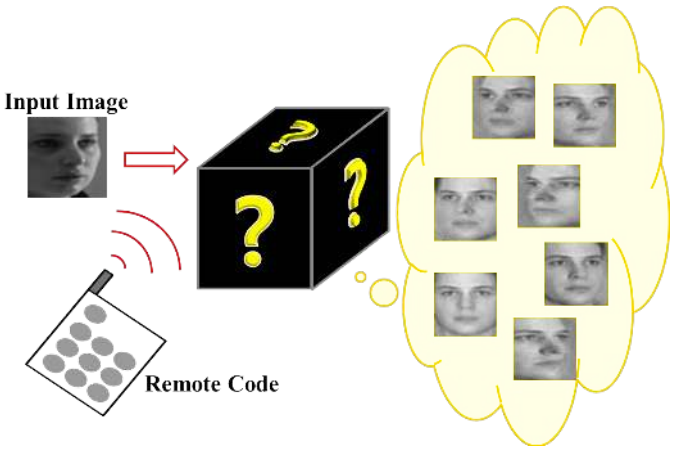

Figure 1. Conceptual diagram of our proposed model. The Input image under an arbitrary pose and illumination is transformed into another pose image. The Remote Code represents the target pose code corresponding to the output image. By interacting between the input image and the Remote Code, our model produces desired pose image.

This paper improves upon these recent achievements by proposing a simple yet powerful way to rotate a two-dimensional face image to a different pose selected by a user. More specifically, an arbitrary pose and illumination are used for input to the network, and a controlled pose under frontal illumination is generated as output. The concept is illustrated in Figure 1. We train a deep neural network (DNN) that takes a face image and a binary code encoding a target pose, which we call Remote Code, and generates a face image with the same identity viewed at the target pose indicated by the Remote Code. It is as if the user has a remote control and a black-box rotator, which can rotate a given face image according to the user's Remote Code. The quality of this rotator can be measured by the degree to

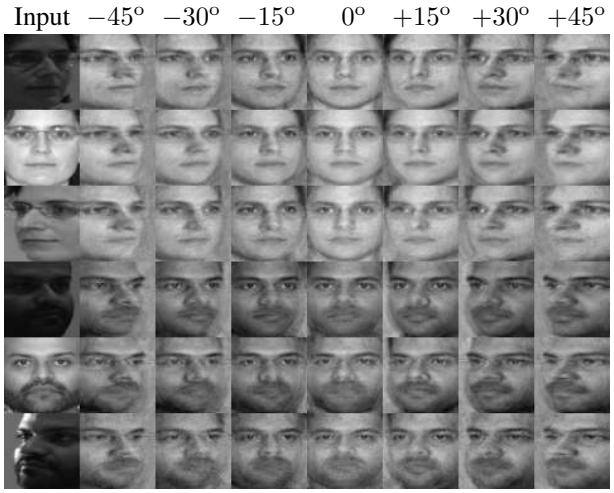

Figure 2. The first column represents the input test images of two individuals from the MultiPIE dataset. The remaining columns are the outputs from the input images with different Remote Codes. For example, the third column represents the -30° pose images resulting from the first column images and the Remote Code that represents -30°. The top three rows have the same identity, and the bottom three rows are the same identity under different illuminations and poses.

which the output face image accords with the desired pose and the degree to which the identity of the face is preserved. Figure 2 shows the final results of our model. From the input images under various illuminations and poses with the same identity, our model can produce almost the same images for each controlled pose under frontal illumination.

To improve the identity-preserving ability of the deep neural network, we introduce an auxiliary DNN and an auxiliary task that requires that the series interconnection of the main DNN, which generates the desired pose image, and the auxiliary DNN reconstructs the original input image, i.e. the auxiliary DNN reconstructs the original input image back from the output image of the main DNN. The idea is that if the series interconnection of the main DNN and auxiliary DNN can reconstruct the original input image, the output of the main DNN should be identity-preserving and contain sufficient information about the identity of the input image. If the identity is not preserved by the main DNN, the output image of the main DNN already takes a different identity and the result of the next auxiliary DNN would deviate even further from the set of valid face images of the original identity.

Another conceptual diagram for this multi-task learning approach is shown in Figure 3. Suppose you would like to rotate a given face image to 30°. A DNN trained with a typical single-task approach would warp the face image along a path that deviates from the ground truth path to some extent, which is depicted by the yellow region. The output image will be somewhere in the intersection of the yellow region and subspace corresponding to pose parameter 30°.

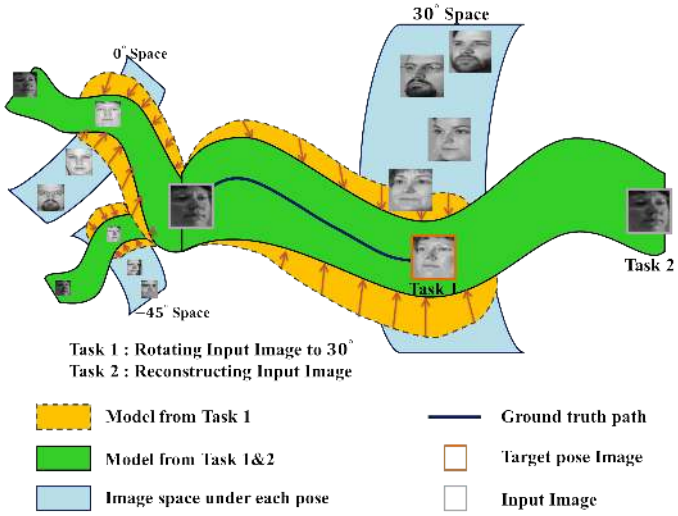

Figure 3. Conceptual diagram for our multi-task learning. By attaching second task, the path from input image to target pose image is closer to the ground truth path than the single task.

With the additional task that restores the original input image back from the output image, the warping path would get closer to the ground truth path as depicted by the green regions due to improved identity preserving ability. Similarly, the target pose can be 0° or -45°, etc., as illustrated in Figure 3.

Previous multitask learning models have shared some layers to determine common features [13]. After the shared layers, the remaining layers are split into the multi-tasks. However, we have designed the multi-task model in a novel way, as described in Figure 4. Our multi-task model shares all the layers involved in the main DNN and attaches auxiliary DNN right after the main DNN to improve the identity-preserving ability. To evaluate the performance of our model, we prepared a face recognition task. We trained and tested on the large MultiPIE face dataset [5], which contains face images taken in various poses and under diverse illuminations. We use several pose-changed images from each test image as the pose- and illumination- invariant features. **Our contributions** are as follows: **1.** We propose the new architecture and Remote Code, which can efficiently change the image into the desired pose. Unlike [27], where several candidate face images for face rotation should be generated and the best fit for controlled pose is selected from among the many candidates, the proposed method can generate the new face image with the desired pose in a single trial. **2.** We introduce a novel type of multi-task learning strategy, which further improves the identity-preserving ability of the DNN. **3.** We achieve a better face recognition rate than [26] and [27] using all the synthesized images at multiple viewpoints and voting among the multiple face recognition results.

The rest of this paper is organized as follows. In Sec. 2, previous research about the face recognition and multitask learning are explained. The description of our model and

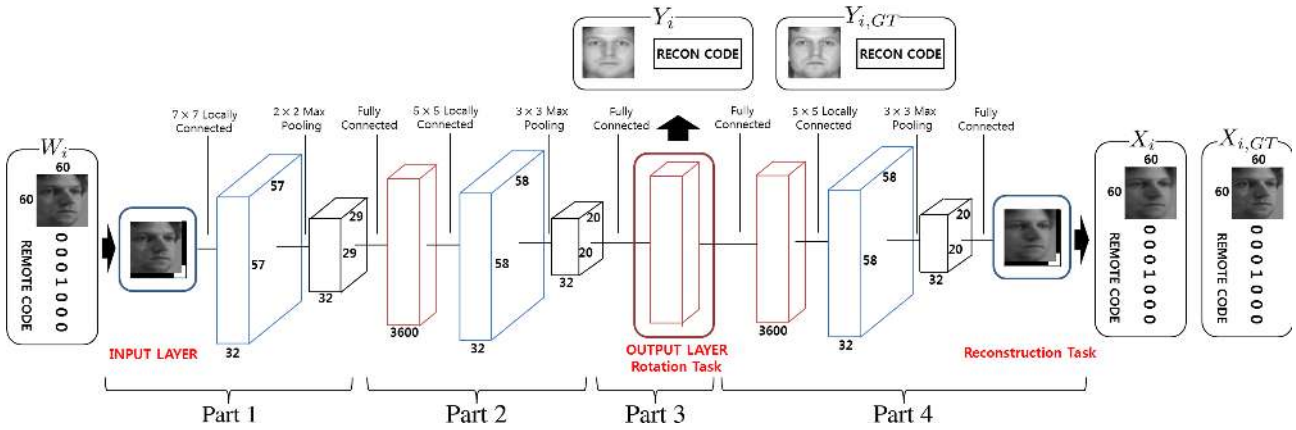

Figure 4. Complete architecture of our DNN model containing four main parts: the feature extraction part, the feature rotation part, the imaging part, and the reconstruction task part which is the auxiliary task. Each part consists of the locally connected layer, the max-pooling layer, and the fully connected layer. In the third part, the red box represents the output layer where the target pose images are generated.

what we focused on designing is explained in Sec. 3, and the parameters of our models are described in Sec. 4. Sec. 5 describes various experiments to demonstrate the strength of our model, followed by the conclusion in Sec. 6.

2. Related Work

Face recognition Typically, for the past twenty years, hand-crafted features such as LBP [1], SIFT [15] or Gabor [14] have been used in the face recognition task. Recently, face recognition and verification across poses have become major issues. These studies are largely separated into studies of 3D methods [2, 12, 24, 20] and 2D methods [11]. For the 3D methods, Asthana et al. [2] rotate non-frontal images to frontal images using the 3D model and landmark points. Li et al. [12] also transform a rotated face image to a frontal image using a morphable displacement field. Yi et al. [20] provide pose-robust features by using transformed filters and the 3D model. On the other hand, 2D methods extract pose-invariant features without 3D information. By representing the test image with a weighted sum of gallery images, Li et al. [11] use these weights as pose-invariant features. The DNN have been used to find pose-robust features without hand-crafted features [26, 7, 27]. Zhu et al. [26] change various pose images into frontal images using CNN and use these ouput images directly as pose-invariant features. This method is also applied in [7], by changing poses step-by-step to minimize the effects of the manifold. Zhu et al. [27] propose a multi-view perceptron (MVP), which can untangle the identity and pose by using random hidden neurons.

Deep learning with multitask learning Recently, many DNN architectures have improved the performance on several computer vision tasks by using multitask learning [21, 23, 4]. To obtain the global weights that can extract features for the various tasks, Collobert et al. [4] iteratively trains the single model on each training set corresponding to each different task. Instead of sharing all weights, a DNN has the shared layers in the front part, followed by separated layers to perform different tasks [25, 13, 22]. However, our model shares all layers of the main task with the second task.

3. Model Description

Two key objectives of our model are creating a new posed image according to what the Remote Code represents, and preserving input image identity even though the pose is changed. Our model is carefully designed to produce superior performance in these objectives. Figure 4 represents the final design of the network. Our model uses an image $M \in \mathbb{R}^{N \times N}$ and the Remote Code $C \in \{0,1\}^{2N+1}$ for the input $W \in \mathbb{R}^{(N+1)\times(N+1)}$, which is defined as:

$$W_{(x,y)} = \begin{cases} M_{(x,y)} & \text{if } 1 \le x, y \le N \\ C^{N+1-x+y} & \text{otherwise} \end{cases}, \quad (1)$$

where (x, y) and C^j represent the pixel coordinate (x, y) and the j-th bit of C, respectively. As shown in Figure 4, the Remote Code surrounds the input image to make a square input set. Experimentally, the way to attach the Remote Code to an image doesn't effect on the performance.

Many previous works have efficiently used CNN to train the DNN model from images [16, 9]. However, CNN shares filters over all images, when it is inappropriate to apply filters, which share weights, to the Remote Code attached image. For that reason, we use the locally connected layer without weight sharing for the first part. For the second part, we use the fully connected layer to change features to contain the target pose information that the Remote Code represents. The locally connected layer and the pooling layer are applied after the fully connected layer to make features more effectively contain pose information and preserve identity. After the second part, the output layer, which consists of the fully connected layer, functions to construct

the new pose image. Furthermore, the novel element of the additional task part is attached after the third part. A detailed explanation of part 4 is contained in Sec. 3.2.

The whole set of parameters is expressed as Input(61×61)-L(7,32)-P(2,2)-FC(3600)-L(5,32)-P(3,3)-FC(3729)-FC(3600)-L(5,32)-P(3,3)-FC(3721). L, P, and FC denote the locally connected layer, pooling layer, and fully connected layer, respectively. L(7,32), P(2,2), and FC(3600) mean that this layer applies 32 filters without weight sharing with size 7, the max-pooling layer whose size is 2 with stride 2, and the fully connected layer with 3600 neurons, respectively. FC(3729) is the output layer that produces a target image and code that represents the informations of input image. In addition, FC(3721) means a second task layer that reconstructs an input image and Remote Code the same as the input layer. The locally connected layer and fully connected layer use an ReLU activation function [9]. The whole locally connected layer has stride 1, and all the strides of pooling layers are set to the same as their filter size. However, the output layer and the last layer contain the linear activation function without rectification. Parameter settings can be varied flexibly depending on the input image size and the number of target poses. The above parameter settings are designed for the experiment with 60×60 input images and 7 poses, described in Sec. 5.2.1.

3.1. Remote Code

We use two special codes at the input and output layer to control the input image to change their pose. The code at the input layer, Remote Code, C_i, $i = 1, \ldots, n$, instructs the input image to change to the i-th pose out of n poses with the same identity. The Remote Code, which is a kind of simple repetition code, $C_i \in \{0,1\}^l$ with total length l is defined as:

$$C_i^j = \begin{cases} 1 & \text{if } (i-1) \times k < j \leq i \times k \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where C_i^j is the j-th bit of code C_i and $k = \lfloor l/n \rfloor$. (l, n) were set equal to $(121, 7)$ and $(65, 9)$ for the experiments described in Sec. 5.2.1 and Sec. 5.2.2, respectively. As the output layer generates the target pose image with frontal illumination from various illumination images, we do not need the illumination information at the input layer code. However, the auxiliary DNN that starts with the output layer of the main DNN needs the information of not only the pose but also the illumination of the input image to reconstruct the input image. We set the output layer code, called Recon Code, $\{Q_i, S_t\}$, $i = 1, \ldots, n, t = 1, \ldots, m$, which represents the i-th pose out of n poses with the t-th illumination condition out of m illumination variations of the input image. Similarly to the Remote Code, we set the pose code, $Q_i \in \{0,1\}^l$ with total length l is defined as:

$$Q_i^j = \begin{cases} 1 & \text{if } (i-1) \times k < j \leq i \times k \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

where Q_i^j is the j-th bit of code Q_i and $k = \lfloor l/n \rfloor$. (l, n) were set equal to $(49, 7)$ and $(72, 9)$ for the experiments described in Sec. 5.2.1 and Sec. 5.2.2, respectively. Furthermore, the illumination code, $S_t \in \{0,1\}^l$ with total length l is defined as:

$$S_t^j = \begin{cases} 1 & \text{if } (t-1) \times k < j \leq t \times k \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

where S_t^j is the j-th bit of code S_t and $k = \lfloor l/m \rfloor$. (l, m) were set equal to $(80, 20)$ and $(60, 20)$ for the experiments described in Sec. 5.2.1 and Sec. 5.2.2, respectively.

Finally, we can define the training dataset. As we can make n Remote Codes for one image, the dataset is n times larger than the original dataset. We can set the training dataset, input and output pairs for the each image M, $L = \{\{M, C_i\}, \{M_i, Q_j, S_t\}\}$, where $i = 1, \ldots, n$ and M_i is the i-th pose image with frontal illumination with the same identity as M. Q_j and S_t are the pose and illumination code of the image M, respectively.

3.2. Multitask Learning

We used a multitask learning model as described in Figure 4. Although the main objective of our model is to construct the new pose image, we additionally attached a second task, reconstructing the input image, after the first task model to preserve the input identity while rotating an input image.

We take the squared L_2 norm as the cost function for both tasks. For the first task, the cost function of the output layer, constructing the new pose image and the Recon Code, is defined as:

$$E_c = \sum_{i=1}^{N} \|Y_{i,GT} - Y_i\|_2^2, \quad (5)$$

where $Y_{i,GT}$ and Y_i are the ground-truth and the generated output that contains the changed pose image, and the pose and illumination information of the input image, respectively. Furthermore, i and N indicates the index of the training input and total batch size, respectively.

The cost function of the second task, reconstructing the input image and the Remote Code, is defined as:

$$E_r = \sum_{i=1}^{N} \|X_{i,GT} - X_i\|_2^2, \quad (6)$$

where $X_{i,GT}$ and X_i are the ground-truth and the constructed output containing the input image and Remote

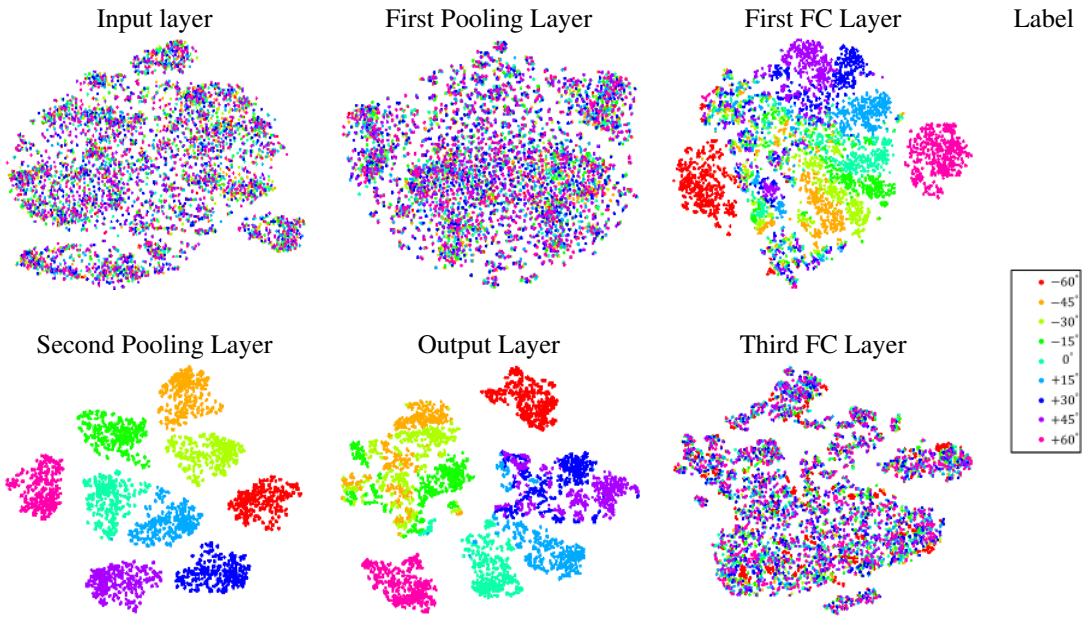

Figure 5. Feature space of 6000 features of the testing images, from MultiPIE dataset, extracted on each layer. Each dot of the same color represents the feature of input set containing the same Remote Code; for example, the red dot is the feature containing the Remote Code representing -60°.

Code, respectively. Our final cost function is the weighted sum of the cost function of the first and the second task,

$$E = \lambda_c E_c + \lambda_r E_r, \tag{7}$$

where λ_c and λ_r are the weights for the first and second task, respectively. We assumed that the two tasks have same importances. λ_c and λ_r were set equal to 1 for all experiments.

4. Training

All our experiments used the cuda-convnet [8], which is one of the popular DNN toolboxes. We can control several parameters including the initial weight (iniW), the learning rate of weight and bias (epsW, epsB), momentum of weight and bias (monW, momB), and L_2 weight decay (wc). For all experiment settings, we use the same parameters. For the locally connected layer and the fully connected layer, we set the iniW equal to 0.001 and 0.01, respectively. In addition, for all layers except the first locally connected layer, we set the epsW, epsB, momW, momB, and wc equal to 0.0001, 0.0002, 0.9, 0.9, and 0.04, respectively. We set epsW to 0.001 and epsB to 0.02 for the first locally connected layer. We trained our model using mini-batch gradient descent with back propagation [10]. The batch size is equal to 100.

To obtain the input and output training set, we carried out two preprocessing steps for the image set, not for the Remote Code. First, in order to be robust to illumination changes, each image is subtracted and divided by the mean and variance of each image, respectively. Second, we also subtracted the per-pixel mean and divided by the per-pixel variance, computed over the training images.

5. Experiments

The experiment section consists of four parts to demonstrate the strength of our model. Sec. 5.1 shows the feature space of each layer to analyze how the input face image rotates along the deep architecture. We use the t-SNE method [19], one of the famous tools to transform high-dimensional space into two-dimensional space. Sec. 5.2 contains the results of face recognition experiments with state-of-the-art procedures to demonstrate the ability to preserve identity. We carefully designed our model to construct a target image and preserve an identity at the same time, performing well on both tasks. Accordingly, we contruct the experiments to demonstrate the effectiveness of our model. In Sec. 5.3, we compare our multitask model to a single task model. Finally, in Sec. 5.4, we construct an experiment to show the advantages of putting a fully connected layer at the beginning.

5.1. Feature Space

As shown in Figure 5, the features from the first pooling layer are mixed together in similar patterns to the input layer. This shows that the first locally connected layer and the pooling layer extract useful features of the input image, rather than the changed pose. Features that have the same Remote Code inserted in the input layer start to merge with

each other from the first fully connected layer. However, some different-color dots mixed with each other show that the one fully connected layer is not enough to change the pose perfectly. The locally connected and pooling layers attached after the fully connected layer clearly performed the objective of changing pose. As shown in Figure 5, at the second pooling layer, features are perfectly separated from the other colors. As the output layer is a fully connected layer, operating to change the features into the target pose image, features are mixed with those of similar poses; for example, the -45^o, -30^o, and -15^o images are similar. Features extracted from the second pooling layer show a better performance than those extracted from other layers in the face recognition task, as described in Table 3.

5.2. Face Recognition

To demonstrate how our model maintains the identity of input images, we take the face recognition task by using the MultiPIE dataset [5]. We prepared two experiment settings: Setting 1, we only used session 1 images in the MultiPIE dataset which includes 249 subjects. 100 subjects (ID 001 to 100) under 7 poses (-45^o to $+45^o$) and 20 illuminations, were used for training the model to analyze a human face. After the training, we chose the remaining 149 subjects (ID 101 to 250 except 213) under 6 poses ($-45^o \sim +45^o$ except 0^o) with 19 illuminations (ID 01 $\sim$ 20 except frontal illumination, ID 07) for the probes to test. For the gallery images, one frontal image with frontal illumination for each subject was used. Therefore, 14000 images were used for training, and 16986 images were used for testing. For Setting 2, we prepared more large scale data in the MultiPIE dataset. We used 200 subjects (ID 001 $\sim$ 200) under 9 poses (-60^o to $+60^o$) with 20 illuminations for training. For the testing, we used remaining 137 subjects under 9 poses with 20 illuminations, 137×9×20 images in total. The selecting procedure of gallery images are same with Setting 1.

For the test step, we extracted features from the output layer, called Controlled Pose Image (CPI), which is marked with a red box in Figure 4. Furthermore, we extracted features from the second pooling layer in front of the output layer, termed as Controlled Pose Feature (CPF). To evaluate, all experiments used L_2 distance norm to compare the test image and gallery images. As our model can create different pose images from one image, we make n, the number of trained poses (7 for Setting 1 and 9 for Setting 2), images P_i ($i = 1, \ldots, n$) per one probe image. Furthermore, we make n images G_i^j ($i = 1, \ldots, n$, j is subject identity) per one gallery image. For each i, the result of the equation

$$\min_j \|P_i - G_i^j\|_2^2, \tag{8}$$

is calculated. The result for each i is voted to produce the final result.

	-45^o	-30^o	-15^o	$+15^o$	$+30^o$	$+45^o$	*Avg*
Li[11]	63.5	69.3	79.7	75.6	71.6	54.6	69.3
Z.Zhu[26]	67.1	74.6	86.1	83.3	75.3	61.8	74.7
CPI	66.6	78.0	87.3	85.5	75.8	62.3	75.9
CPF	**73.0**	**81.7**	**89.4**	**89.5**	**80.4**	**70.3**	**80.7**

Table 1. Recognition rates (%) for the various poses under Setting 1. Best results are written in bold.

	00	01	02	03	04	05	06
Li[11]	51.5	49.2	55.7	62.7	79.5	88.3	**97.5**
Z.Zhu[26]	**72.8**	**75.8**	75.8	75.7	75.7	75.7	75.7
CPI	66.0	62.6	69.6	73.0	79.1	84.5	86.6
CPF	59.7	70.6	**76.3**	**79.1**	**85.1**	**89.4**	91.3
	08	09	10	11	12	13	14
Li[11]	**97.7**	**91.0**	79.0	64.8	54.3	47.7	67.3
Z.Zhu[26]	75.7	75.7	75.7	75.7	75.7	**75.7**	73.4
CPI	86.5	84.2	80.2	76.0	70.8	65.7	76.1
CPF	92.3	90.6	**86.5**	**81.2**	**77.5**	72.8	**82.3**
	15	16	17	18	19	*Avg*	
Li[11]	67.7	75.5	69.5	67.3	50.8	69.3	
Z.Zhu[26]	73.4	73.4	73.4	72.9	**72.9**	74.7	
CPI	78.2	80.7	79.4	77.3	65.4	75.9	
CPF	**84.2**	**86.5**	**85.9**	**82.9**	59.2	**80.7**	

Table 2. Recognition rates (%) for the various illuminations under Setting 1. Best results are written in bold.

5.2.1 Result of Setting 1: Containing 7 Poses

In this setting, we used 60×60-size images for the input as described in Figure 4. We compared our results with the state-of-the-art results [26] and Li et al. [11]. The results of recognition rates for different poses are shown in Table 1. As with human perception, our model found it difficult to imagine the face identity from the greatly rotated images, -45^o and $+45^o$ cases. However, Table 1 shows that not only the CPI but also the CPF outperformed the state-of-the-art for most poses. Table 2 shows the recognition rates for 20 different illuminations. As we tested on 19 illumination settings excluding frontal illumination (ID 07), only 19 results are shown. The CPF outperforms all the other methods for 12 out of 19 parts.

5.2.2 Result of Setting 2: Containing 9 Poses

As the state-of-the-art [27] uses training and test images with size 32×32, we prepared the same setting. The changed input image size requires different parameter settings. The whole set of parameters is expressed as Input(33×33)-L(5,16)-P(2,2)-FC(1600)-L(5,16)-P(2,2)-FC(1156)-FC(1600)-L(7,16)-P(2,2)-FC(1089). We compared our results with several features listed in [27]. All the previous settings used LDA to reduce the dimensions of features. As shown in Table 3, the CPF outperforms all the other methods for all different poses. Extracting features from different layers produced different results. As

	-60^o	-45^o	-30^o	-15^o	0^o	$+15^o$	$+30^o$	$+45^o$	$+60^o$	*Avg*
Landmark LBP[3]	35.5	52.8	71.4	83.9	94.9	82.9	68.2	48.3	32.1	63.2
FIP+LDA[26]	49.3	66.1	78.9	91.4	94.3	90.0	82.5	62.0	42.5	72.9
RL+LDA[26]	44.6	63.6	77.5	90.5	94.3	89.8	80.0	59.5	38.9	70.8
MTL+RL+LDA[27]	51.5	70.4	80.1	91.7	93.8	89.6	83.3	63.8	50.2	74.8
Z.Zhu+LDA[27]	60.2	75.2	83.4	93.3	95.7	92.2	83.9	70.6	60.0	79.3
CPI	55.8	71.8	80.0	90.1	98.4	90.2	82.7	71.0	52.9	77.0
CPF	**63.2**	**80.4**	**88.1**	**94.5**	**99.5**	**95.4**	**88.9**	**79.4**	**60.6**	**83.3**
CPF-FC1600	45.4	72.7	80.8	885	96.8	90.3	79.6	70.22	42.5	74.1
CPF-Pool1	9.7	39.1	51.6	69.9	92.5	70.8	51.1	39.4	9.3	48.1

Table 3. Recognition rates (%) for the various poses under Setting 2. The CPF-FC1600 and the CPF-Pool1 indicates the features extracted from the first FC(1600) layer and the first pooling layer, respectively. Best results are written in bold.

the output layer is converting high level features into target images, some of the discriminative features useful for discerning face identities may be lost at the output layer. This is why the high level feature just before the output layer, CPF, performs the best. Our model achieves remarkable performance on 0^o. Showing a 99.5% recognition rate, our method clearly outperforms the state-of-the-art algorithm, which reports a 95% recognition rate on 0^o. Indeed, the recognition rate of our method amounts to 14 misclassifications out of 2740 images.

Although the above final results are produced by voting among the multiple results which are produced by each CPFs, most of the correct results are generated by large number of votes meaning high confidence as shown in Figure 6. This result indicates that face recognition based on synthesized face images at each target pose gives quite consistent result and that the proposed DNN can generate high quality multi-view face images across pose. In addition, as described in Figure 7, proposed model can preserve identity while changed pose as well.

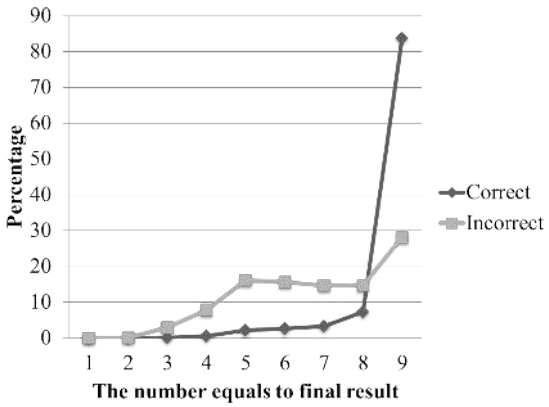

Figure 6. The percentage of the number of CPFs contributing to final result. Most of the incorrect results are generated by low confidence, e.g. 5 out of 9 CPFs are voted. On the other hand, most correct results are produced from high confidence. We can infer that each CPF has an ability to preserve identity.

5.3. Compare to Single Task Learning

We constructed a new experiment to demonstrate the effectiveness of appending the reconstruction task layer after the output layer. We prepared two models, the multi-task model the same as Figure 4 and the single task model as only the first task of the first model. As the CPF outperforms the CPI, we take CPF for both models in the same experiment as Setting 1. The recognition rates for various poses and illuminations are shown in Table 4 and 5, respectively. For all pose and illumination settings, multi-task model is better than single task model. The first task of multitask model is to construct a target pose image that the Remote Code represents, the same as the single task model's objective. However, the second task is to reconstruct the input image and the input Remote Code from the output layer features. Since the output layer of the multi-task model must contain identity preserving features to reconstruct the input layer in the second task, the multitask

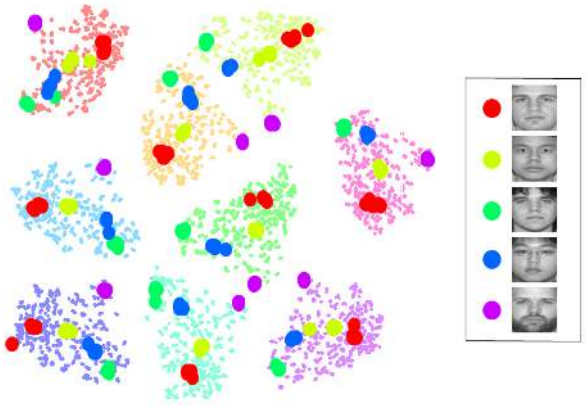

Figure 7. The feature space of 6000 features from the second pooling layer with Setting 2. Each pale dot color represents a different Remote Code. 54 dots with the same deep colors represent the features from a single identity. In the feature space, not only are the features united among the Remote Codes, but also the deep dots are united among the same identity in each pose.

	$-45°$	$-30°$	$-15°$	$+15°$	$+30°$	$+45°$	*Avg*
Single	65.4	76.5	85.9	85.8	76.3	63.2	75.5
Multi	**73.0**	**81.7**	**89.4**	**89.5**	**80.4**	**70.3**	**80.7**

Table 4. Recognition rates (%) for the various poses under Setting 1, comparing with single task model and multitask model. Best results are written in bold.

	00	01	02	03	04	05	06
Single	45.4	64.3	72.9	74.9	82.0	86.9	89.8
Multi	**59.7**	**70.6**	**76.3**	**79.1**	**85.1**	**89.4**	**91.3**
	08	09	10	11	12	13	14
Single	89.7	87.9	81.7	76.5	72.2	66.7	76.9
Multi	**92.3**	**90.6**	**86.5**	**81.2**	**77.5**	**72.8**	**82.3**
	15	16	17	18	19	*Avg*	
Single	80.9	82.7	79.9	76.5	47.1	75.5	
Multi	**84.2**	**86.5**	**85.9**	**82.9**	**59.2**	**80.7**	

Table 5. Recognition rates (%) for the various illuminations under Setting 1, comparing with single task model and multitask model. Best results are written in bold.

model retains identity features more effectively than the single task model.

5.4. Effectiveness of Early FC layer

Most DNN models are composed of two large parts, feature extraction and combination of the features. As a large-scale input size is difficult to handle, convolutional layers or locally connected layers are usually used at the beginning of the network for feature extraction. In addition, fully connected layer is used to combine features in the rear. However, as described in Figure 4, our model uses a fully connected layer at the beginning. To examine the effectiveness of an early fully connected layer model (EFC), we constructed an experiment with Setting 2. Our model is described in Section 5.2.2. We designed another model (LFC) in which the fully connected layer is located just before the output layer, not at the beginning. The whole parameter set is defined as Input(33×33)-L(5,16)-P(2,2)-L(5,16)-P(2,2)-FC(1600)-FC(1156)-FC(1600)-L(7,16)-P(2,2)-FC(1089). All parameters are the same, except the position of the fully connected layer, the first FC(1600) layer. The recognition rates of the two models are noted in Table 6. We also include the results of using the CPI and CPF for each model. CPF-LFC and CPI-LFC are extracted from the FC(1600) and FC(1156), respectively.

As the locally connected layer filter acts on local parts, only the fully connected layer operates globally. Accordingly, at the fully connected layer, the Remote Code starts to change features to contain the target pose that the Remote Code represents. Thus in the early fully connected layer model, the features that contain the target poses appear earlier than in the late fully connected model. As the results show, although CPI-LFC results are better than those

	CPI-EFC	CPF-EFC	CPI-LFC	CPF-LFC
Result	77.0	**83.3**	78.3	79.7

Table 6. Recognition rates (%) compared with our model, which has an FC layer at the beginning, and Late FC model, which has no FC layer at the beginning. Best results are written in bold.

of CPI-EFC, the early fully connected model is better than the late fully connected model in the case of the best performance feature, CPF.

6. Conclusion

In this paper, we proposed a novel type of multi-task network that can synthesize the desired pose and frontal illumination face image by utilizing user's Remote Code represents. By attaching a second task model which reconstructs the input image, after the first task model which rotates an input image to a certain pose, proposed multi-task network produced better performance at preserving identity than the single task model. Activation values of the second pooling layer at the first task model can be used as the pose- and illumination- invariant feature. In the face recognition task under arbitrary poses and illuminations, our model clearly win against the previous state-of-the art model by more than 4∼6%.

Acknowledgement

This work was supported in part by Samsung Advanced Institute of Technology (SAIT) and the Technology Innovation Program, 10045252, Development of robot task intelligence technology, funded by the Ministry of Trade, industry & Energy (MOTIE, Korea). Furthermore, this research was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (2014-003140) and (MSIP)(2010-0028680).

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006. 3

[2] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith. Fully automatic pose-invariant face recognition via 3d pose normalization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 937–944. IEEE, 2011. 3

[3] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3025–3032. IEEE, 2013. 7

[4] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008. 3

[5] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 2, 6

[6] G. B. Huang, M. Mattar, T. Berg, E. Learned-Miller, et al. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008. 1

[7] M. Kan, S. Shan, H. Chang, and X. Chen. Stacked progressive auto-encoders (spae) for face recognition across poses. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1883–1890. IEEE, 2014. 3

[8] A. Krizhevsky. cuda-convnet. `http://code.google.com/p/cuda-convnet/`. July 2012. 5

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 3, 4

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 5

[11] A. Li, S. Shan, and W. Gao. Coupled bias–variance tradeoff for cross-pose face recognition. *Image Processing, IEEE Transactions on*, 21(1):305–315, 2012. 3, 6

[12] 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 *Computer Vision–ECCV 2012*, pages 102–115. Springer, 2012. 3

[13] S. Li, Z.-Q. Liu, and A. B. Chan. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 488–495. IEEE, 2014. 2, 3

[14] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *Image processing, IEEE Transactions on*, 11(4):467–476, 2002. 3

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[17] Y. Sun, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. *CoRR*, abs/1406.4773, 2014. 1

[18] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014. 1

[19] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008. 5

[20] D. Yi, Z. Lei, and S. Z. Li. Towards pose robust face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3539–3545. IEEE, 2013. 3

[21] X.-T. Yuan, X. Liu, and S. Yan. Visual classification with multitask joint sparse representation. *Image Processing, IEEE Transactions on*, 21(10):4349–4360, 2012. 3

[22] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 1036–1041. IEEE, 2014. 3

[23] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *International journal of computer vision*, 101(2):367–383, 2013. 3

[24] X. Zhang, Y. Gao, and M. K. Leung. Recognizing rotated faces from frontal and side views: An approach toward effective use of mugshot databases. *Information Forensics and Security, IEEE Transactions on*, 3(4):684–697, 2008. 3

[25] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014*, pages 94–108. Springer, 2014. 3

[26] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 113–120. IEEE, 2013. 1, 2, 3, 6, 7

[27] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems*, pages 217–225, 2014. 1, 2, 3, 6, 7