The Visual Computer
https://doi.org/10.1007/s00371-018-1561-3

ORIGINAL ARTICLE

Unified convolutional neural network for direct facial keypoints detection

Je-Kang Park[1] · Dong-Joong Kang[1]

Abstract
We propose a novel approach to directly estimate the position of the facial keypoints via convolutional neural networks (CNN). Our method estimates the global position and the local positions from a unified CNN and combines them through a simplified optimization process. There are twofolds of advantages for our approach. First, the global geometrical position and the local detailed position of the facial keypoints are combined complementarily to avoid local minimums caused by occlusions and pose variations. Second, unlike the traditional method such as a cascade of multiple CNN, we propose a unified deep and large architecture network consisted by global position network and local position network. Our design shares most of computations for facial features between networks, and this efficient high-level features improves largely to the precise estimate of facial keypoints. We conduct comparative experiments with the state-of-the-art researches and commercial services. In experiments, our approach shows a remarkable performance.

Keywords Facial keypoints detection · Face alignment · Unified convolutional neural networks · Shared network

1 Introduction

Facial keypoints detection is a problem of estimating the position of eyes, nose, and mouth in a facial image. This problem, also known as face alignment, has been widely studied for many years in the field of computer vision because of its relevance to various face analysis applications such as face recognition [26], face attribute recognition [3], head pose estimation [13], and 3D face modeling systems [8]. Aligned face images allow for the face analysis systems to concentrate on the appearance of the face without considering the variations in the pose of the face. Therefore, facial keypoints detection helps to achieve better face recognition performance [28] and it is essential for preprocessing of the face analysis applications.

There are several factors that make facial keypoints detection challenging. The size, shape, and skin color of faces vary from person to person. Besides, there are large variation of head orientations, facial expressions, illuminations, and partial occlusions depending on various environmental conditions. In order to develop a detector that is robust to disturbances and environmental changes, existing studies have proposed a feature extraction algorithm [1,5] or a method that can directly model the shape of the face [6,7,21]. Recently, various CNN-based regression methods [27,30,31] have been proposed.

The limitations of these previous methods are twofold. First, since the recently proposed methods [4,22,27,31] commonly use a cascade scheme, they require lots of regressors and hyperparameters (e.g., number of stage, number of branch in each stage, etc.). These methods estimate initial position of facial keypoints approximately (e.g., mean shape) and progressively improve them to estimate the position accurately. Second, they utilize the local features and require the limited size of input images. In [4,22,32], they extract the local features such as local binary features, Fern and HoG from a small patch. The sizes of receptive field of these features are typically less than 10–20 pixels, because pixel difference or edge-based features cannot well encode a context information at the large region. As a result, the size of the input image is limited because each component of the face should be forced to fit within the small patch.

✉ Dong-Joong Kang
djkang@pusan.ac.kr

Je-Kang Park
jkp@pusan.ac.kr

[1] Pusan National University, Busan, Korea

Published online: 28 May 2018

To overcome the above problems and directly estimate the positions of the facial keypoints from the image, it is required to learn a more discriminative features and the regressor that accommodates large shape variations. In this paper, we propose a carefully designed convolutional network that is deeper and larger than that used in [27,31]. A deeper network can well encode the information such as shape or context on a large receptive region, so we can learn more discriminative features. Furthermore, a larger network can learn a mapping function that directly transforms the face image to the facial keypoints unlike in [4,21,22,32] that transform the previous stage points (or mean points) to the current stage points. However, a large and deep network requires lots of computations. To efficiently utilize such a large network, we design the networks that compute the local and global positions to share some of their bottom layers.

Our contributions are as follows. First, we propose a novel direct method for localizing both global geometrical position and local detailed positions independently (not a cascade) and combining them through a simplified optimization process. Second, unlike the cascade of multiple networks, we propose an efficient unified convolutional network structure (sharing networks) that simultaneously estimates both the global position and local position of the keypoints.

2 Related works

ASM [6,21] and AAM [7,23] optimized a generative model for the holistic shape and appearance of a face. Since geometric correlations between the keypoints are preserved, outliers do not occur. However, this method requires an iterative gradient descent optimization process in test phase and is sensitive to the initial position of the keypoints. In addition, due to the limited expressiveness in terms of complex texture features, the learned model is vulnerable to changes of face position, direction, and lighting condition.

The regression-based method learns a mapping function that directly estimates the position of keypoints from the feature vectors extracted from an image. Xiong and Torre [29] proposed a method that minimizes nonlinear least squares functions with SIFT features. Ren et al. [22] learn a linear regression function with locally mapped LBP features via a Random Forest. Recent CNN-based methods [27,31] learn a regression function that performs feature extraction and position estimation at the same time. This gradually improves the position of the keypoints through a cascade scheme. These methods do not require a gradient descent optimization to fit a current shape to the trained model in the test phase or any extra searching process because they directly estimate the positions of the keypoints.

3 Local and global position regressions

In our approach, we compute both global and local positions of the five facial keypoints [left eye (LE), right eye (RE), nose tip (NT), left mouth corner (LM), right mouth corner (RM)] through CNN regression. These two kinds of positions have different characteristics. The global position is computed once on the entire image and is inferior in accuracy but robust against occlusion, because it contains the shape information of the face. The local position is computed on all the local regions of the image, and it is accurate but vulnerable to false alarm caused by occlusion. We complementarily combine the benefits of these two computed positions through a simple optimization process. Figure 1 shows the overview of our approach.

3.1 Global position regression

The global position regression network (GPN) takes a face image as input and outputs the 2D-positions of the five facial keypoints:

$$\begin{aligned} &g = \Phi_g(I, \theta_g) \\ &I \in \mathbb{R}^{3\times h\times w}, \quad g \in \mathbb{R}^{5\times 2} \end{aligned} \tag{1}$$

where I is a 3-channel RGB image, θ_g is learnable weights of GPN, and g is the global position. Since GPN is a mapping function that transforms the input image into the global position, the regression target $\hat{g}$ for learning θ_g is the ground truth 2D-position of the keypoints. The objective function to be minimized is defined as smooth $L1$ loss [10], as follows.

$$L_{\text{global}} = \frac{1}{5}\sum_{k=1}^{5} \text{smooth}_{L1}\left(\hat{g}^k - g^k\right) \tag{2}$$

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \tag{3}$$

when learning a deep network regression with randomly initialized weights, gradient is unstable at the beginning of learning. In Eq. (3), since the $L1$ loss is less sensitive to outliers than the $L2$ loss, the network can avoid exploding of the gradients.

The first half of GPN is constructed by stacking convolution layers for the purpose of feature extraction. A stacked convolutional network extracts high-level features and eliminates unnecessary features while reducing the resolution of feature map. The extracted feature map transforms into a 2D position by a subsequent dense (fully connected) layer. The detailed structure of GPN is represented in Fig. 2.

The difference in our approach compared to the previous methods [27,30,31] is to project the feature map just before entering the dense layer into both directions of the x-

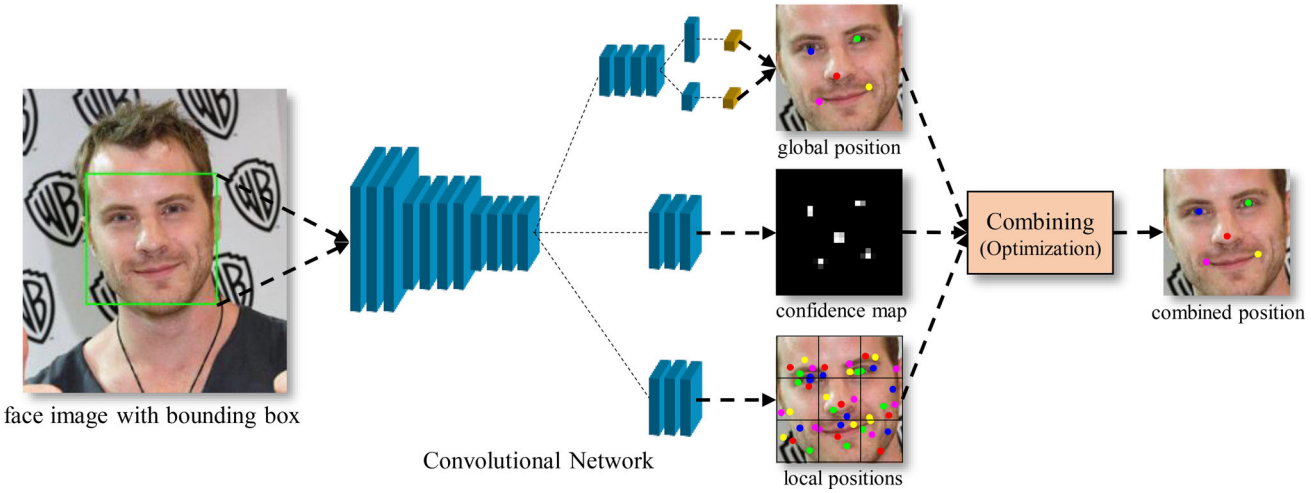

Fig. 1 The overview of our approach. The convolutional network computes a global position, local positions, and confidence map from the face image. Since the global position contains the shape information and the local positions contain the accurate position of the individual components of the face, both positions are complementarily combined through a simple optimization process. Note that the input image in this figure is only divided into 9 patches to form the local regions for clarity, but actually there are more local regions. More details are described in 3.2

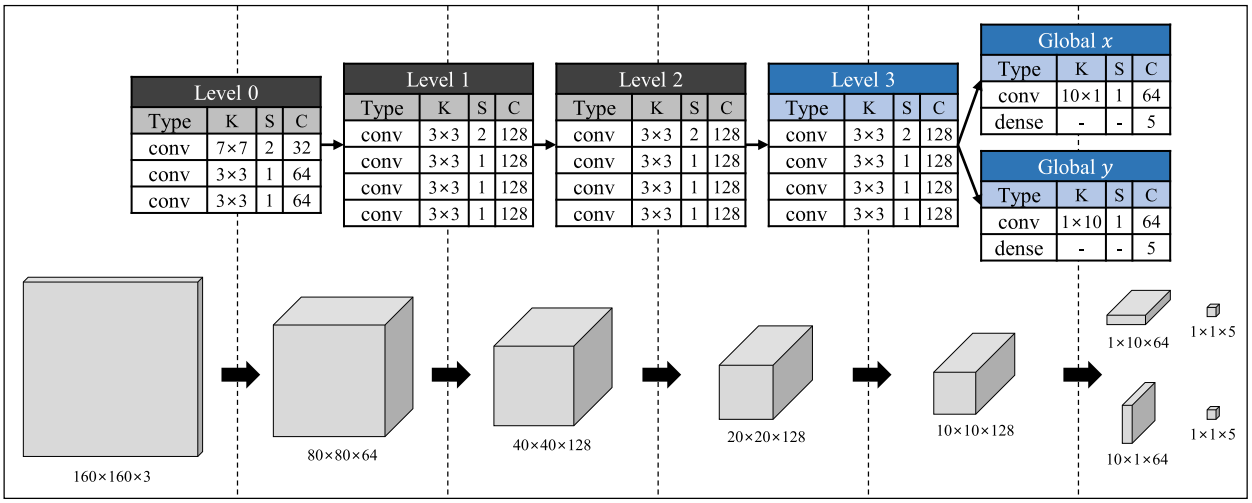

Fig. 2 Detailed structure of the global position regression network (GPN). The feature map is extracted through four-level convolution layers, and the subsequent convolution layers that have an asymmetric kernel project the feature map to compute the global x and y position

and y-axis, respectively. In the previous methods, since the 2D feature map is directly connected to the dense layer, all the unnecessary data in the 2D feature map are utilized for computing each 1D coordinate (i.e., x or y position). However, since 2D convolution preserves the axial direction of the image, when computing the specific dimensional coordinate, the data in the other axial direction are not necessary and thus can be suppressed. As a result, the unnecessary features used for the computation are reduced and the number of learnable weights is reduced, so that the learning of the network becomes easier. This idea was first introduced in [9]. They projected the feature map through max-pooling. On the other hand, we learned the projection matrix as a kernel of the convolution layer. Table 1 shows the comparative validation error and the number of weights for each projection method. Our convolution projection method reduces the learnable weights significantly. Furthermore, the validation error also decreased since the projection matrix is learned.

3.2 Local position regression

The local position regression network (LPN) takes a face image as input and produces the 2D position of the five facial keypoints and a probability distribution for the type of the keypoints (with background) in all local regions:

Table 1 The comparative validation error and the number of learnable weights for the different projection methods

	Projection methods		
	None [27,30,31]	Max-pooling [9]	Convolution (ours)
Validation error (%)	3.63	3.67	3.52
Number of weights	8,316,688	1,770,256	1,927,824

Where 'none' refers to a method that directly feeding the feature map into the dense layer without projection, and 'max-pooling' refers to a method that projecting a feature map with an asymmetric max-pooling kernel and then feeding the projected feature map into a dense layer

$$
\begin{aligned}
&c, l = \Phi_l(I, \theta_l) \\
&I \in \mathbb{R}^{3\times h\times w}, \quad c \in \mathbb{R}^{n\times 6}, \quad l \in \mathbb{R}^{n\times 5\times 2}
\end{aligned} \tag{4}
$$

where I is a 3-channel RGB image, θ_l is learnable weights of LPN, n is the number of local regions, c is the confidence map of the local confidence, and l is the local positions. The local position is the location of the five keypoints detected in the specific local region (single patch of the input image divided into grid shapes). In other words, the local position is the ground truth position of the keypoints on the reference coordinate of the local region. The local position assumes that the local region includes the keypoints. However, most local regions do not include the keypoints. Therefore, it is necessary to select the local regions including the keypoints. The local region may not include the keypoints or may include one of the five keypoints. Therefore, it is possible to select the local region including the keypoints by estimating the probability distribution of the 6 classes (5 keypoints and background). To distinguish the local regions including the keypoints, LPN learns not only the local position but also the confidence map, where the confidence map indicates the probability distribution estimated in all local regions.

The training data of the local position is the position of the ground truth keypoints on the coordinate of the local region. This can be easily computed by moving the ground truth keypoints on the image coordinate to the local region coordinate. The local position assumes that the local region includes the keypoints. Therefore, the only local regions that include actually the ground truth keypoints are used for learning. The training data of the confidence map is a probability distribution of the types of ground truth keypoints included in each local region. Since the local region that does not include the ground truth keypoints belongs to the background class, all local regions are used for learning. The detailed process of generating learning data of the local position and confidence map is shown in Fig. 3. The objective functions to be minimized are defined as smooth $L1$ loss and cross entropy loss, as follows.

$$
L_{\text{local}} = \frac{1}{5M}\sum_{j}^{M}\sum_{k=1}^{5}\text{smooth}_{L1}\left(\hat{l}_j^k - l_j^k\right) \tag{5}
$$

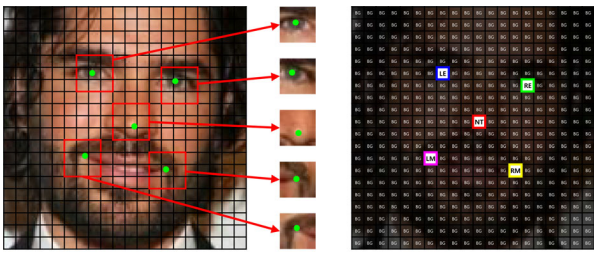

Fig. 3 The training data of the local position (**a**) and the confidence map (**b**)

$$
L_{\text{conf}} = \frac{1}{6N}\sum_{i}^{N}\sum_{k=0}^{5}\hat{c}_i^k \log c_i^k + \left(1-\hat{c}_i^k\right)\log\left(1-c_i^k\right) \tag{6}
$$

where $\hat{l}_j$ and $\hat{c}_i$ are jth ground truth local positions and ith local confidence of a local region, respectively, M is number of local regions including the ground truth keypoints, and N is number of all local regions.

In Fig. 4, the first half of LPN (level 0–2) is constructed same as GPN for feature extraction while the remaining part is composed of the convolution layers that play a role of sliding detector. In other words, we explore the entire image and compute l_i and c_i in all local regions. Since convolutional network does not extract the features of the overlapped local regions repeatedly, it can efficiently search for all regions [24], and no additional computations are required because the first half of LPN is shared with the GPN. We will discuss this shared network in Sect. 4.1.

3.3 Combining local and global positions

After estimating the global and local positions, we combine two positions through a simplified optimization process to compute more refined position. Figure 5 shows the characteristics of the global geometrical position and the local accurate position. Considering combination of these characteristics, our optimization function is as follows.

$$
\begin{aligned}
&\hat{i} = \underset{i}{\text{argmax}}\left(c_i^k + \alpha\cdot\exp\left(-\|l_i^k - g^k\|_2\right)\right) \\
&p^k = c_{\hat{i}}^k\cdot l_{\hat{i}}^k + \left(1-c_{\hat{i}}^k\right)\cdot g^k \\
&(k = 1,\ldots,5)
\end{aligned} \tag{7}
$$

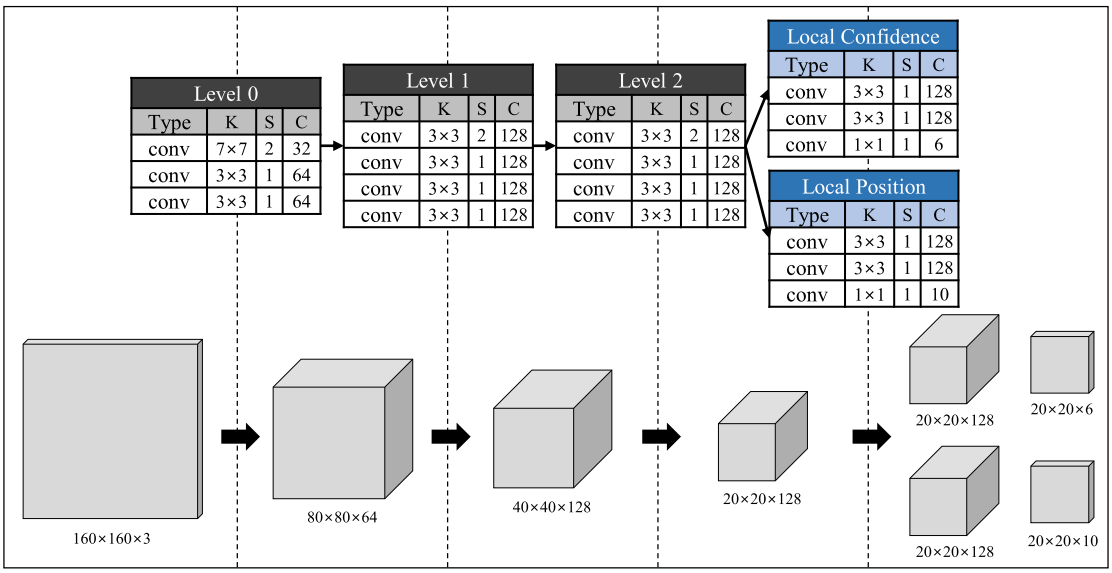

Fig. 4 Detailed structure of the local position regression network (LPN). The feature map is extracted through three-level convolution layers. Then, the following two branch of convolution layers explore the feature map and compute the local position and confidence, respectively. Since the stride for sliding search over the feature map is 1, the number of local regions in this figure is 400 (20×20)

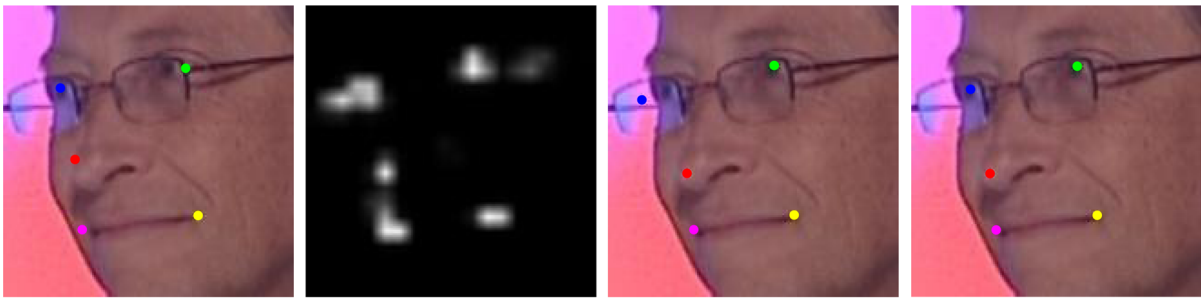

Fig. 5 The complementarily combination of global and local positions. The global position **a** is inaccurate with the right eye and the nose tip. The local position **c** missed the left eye due to the glasses. The combined position **d** computed from Eq. (7) complements the weaknesses of **a**, **c**, so all the feature points are detected well

where α is a weight that controls the influence of the global position, i iterates over all local regions, and p is the combined position. Equation (7) takes the global position if there is no local position with high confidence. Therefore, if the local position is not detected due to occlusion, the global position is highly utilized to estimate the combined position.

4 Implementation details

4.1 Sharing networks

The facial keypoints to be estimated by GPN and LPN are ultimately same, only the regression target region is different. Therefore, both networks can share low-level visual features. We constructed a shared position network by sharing the convolution layers from Level 0 to Level 2 (just before the LPN branched out) in Figs. 2 and 4. The objective function of shared network is as follows, and we uniformly weight each loss.

$$L_{\text{total}} = L_{\text{global}} + L_{\text{local}} + L_{\text{conf}} \tag{8}$$

This sharing network scheme has several advantages. First, the training time and processing time are reduced. About 52% of network weights are shared, so they do not learn or compute redundantly. Second, the error of the global position is reduced. While training the shared network, the spatial

information of the facial image is preserved in the local confidence map. This can be interpreted as the coarse position of the keypoints are learned together, so that the global position of the shared network is more accurate and converges quickly.

4.2 Network structures and training

Our network builds on the design of AlexNet [17] and VGG [25]. Most of the convolution layers, except the first one, have a 3×3 kernel and followed by the ReLU activation. We modify subsampling layers; all max-pooling layers are replaced by strided convolution layers for slightly faster convergence. All weights were initialized by [12] and learned from scratch. We used a standard stochastic gradient descent optimization method with a mini-batch size of 32. We set the momentum as 0.9 and the weight decay as 0.0005, which were same with AlexNet [17]. The initial learning rate was 0.01 and continuously decreased by multiplying it by 0.1 when the current iteration reached 50, 75, and 87.5% of the total 40,000 iterations.

4.3 Data augmentation

To improve generalization performance, we adopt some data augmentation techniques during training. We compute the tight bounding boxes of the keypoints and resize image so that the box size becomes 45 and 65% of the input image size. Next, the image is rotated so that the angle between the vector or eyes and the x-axis is $-35°$ to $35°$. The image is then cropped with an input size at random positions including the face. Additionally, we adopt a color augmentation method that adds -32 to 32 to pixels for the brightness change and multiplies the pixels by 0.5–1.5 for the contrast change.

5 Experiments

We first investigate various sizes of input image and different depth and size of network on the LFW + Web [27] dataset, and then, we conduct a comparative experiment with the state-of-the-art methods and commercial software on two public datasets BioID [15] and LFPW [2]. All experiments are conducted on Caffe [16] with Geforce GTX Titan X (original, not Pascal model). All codes and learned model for the test are released on the (https://github.com/jedol/facial-keypoints-detection) with the reported results.

The error metric we used is the normalized root-mean-square error (RMSE) as used in [4]. The $L2$ distance between the predicted keypoints and the ground truth is normalized by inter-ocular distance.

$$\text{error} = \frac{1}{5}\sum_{k=1}^{5}\frac{\|\hat{g}^k - p^k\|_2}{\|\hat{g}^1 - \hat{g}^2\|_2} \tag{9}$$

where $\hat{g}$ is the ground truth keypoints, p is the predicted keypoints, and $\hat{g}^1$ and $\hat{g}^2$ are the ground truth left and right eye center, respectively.

5.1 Efficiency of shared network

In this section, we will show how our shared network structure is more efficient than the unshared GPN + LPN (GPN and LPN are composed of independent CNN) or single GPN in terms of speed and performance. Prior to observing the efficiency of the shared network, we train and evaluate the network with the LFW + Web [27] dataset to find the optimal network structure and input image size. LFW + Web consists of 13,466 images of arbitrary size. 5590 images come from LFW [14], and 7876 images collected from web. The ground truth position of the five facial keypoints and the face bounding box are labeled on all images. 10,000 images are used for training and 3466 images are used for validation.

In order to generate a fixed size input image from the raw image of arbitrary size, we first crop the bounding box area from the raw image. Then, the cropped image was resized to fit the size of the input image. The results of comparative experiments for various input image sizes and designs of the network are summarized in Table 2. The case 1 shows the experiments about various input image sizes, the case 2 shows the experiments about various capacities of the network, and the case 3 shows the experiments about various depths (the number of convolution layers at level 1 and 2) of the network. Note that hyperparameter α in Eq. (7) is selected through cross-validation. Figure 6 shows the distribution of validation error over various α. Since our network is deep and large, the optimal input image size is 160×160 (case 1-B) which is considerably larger than that used in [4,22,27,30–32]. For the various designs of the network, the large capacity model (case 2-B) reported the lowest validation error, but the FPS was the slowest too. Therefore, we selected case 1-B as a baseline model to trade-off the speed and performance.

We train GPN, LPN and the shared GPN + LPN with the baseline model and compare the performance. The results are summarized in Table 3. First, although the baseline model is sufficiently deep and large, GPN shows a quite large error. However, as GPN combined with LPN, the error is significantly reduced. Also, GPN and LPN preserve their performance even though they share the weights of the level 0 to level 2 layers. As a result, the shared GPN + LPN shows the same performance as the unshared GPN + LPN, while the computation time is 1.6 times faster than the unshared GPN

Table 2 Validation error and FPS for the different input image sizes and various designs of network

Case	Type	Input size	Depth	Weights ($1e^3$)	α	FPS	Error (%)
1	A	96 × 96	8	2157	4	2247	2.36
	B	160 × 160	8	2225	4	840	2.28
	C	256 × 256	8	2327	4	355	2.39
2	A	160 × 160	8	742	4	1074	2.61
	B	160 × 160	8	21,839	4	246	2.21
3	A	160 × 160	4	1635	4	995	2.39
	B	160 × 160	16	3406	4	619	2.39

The case 1 shows various input image sizes, the case 2 shows various capacities of the network and the case 3 shows the various depths (the number of convolution layers at level 1 and 2) of the network. Note that hyperparameter α is selected through cross-validation

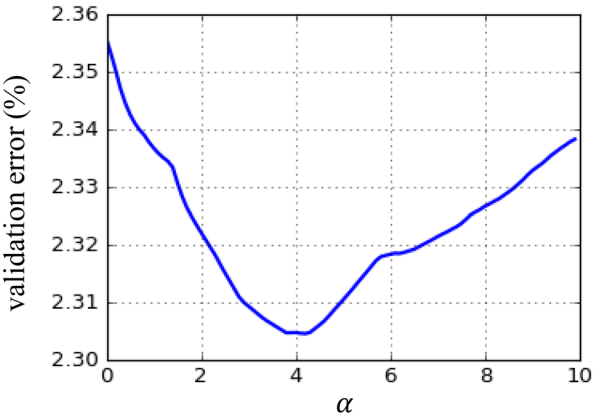

Fig. 6 Distribution of the validation error over various α. α controls the influence of the global position, so it shows poor performance at too high or low value

+ LPN. We also train ResNet [11] with the same capacity as the baseline model to investigate the performance for the expressive design of network and achieve better performance than the baseline model.

5.2 Comparison with other methods

We conduct comparative experiments with the state-of-the-art methods and commercial software to demonstrate the performance of our method. In experiments, we use the LFW + Web [27], BioID [15] and LFPW [2] public datasets. LFPW originally contains URLs for 1132 training images and 300 test images. But some URLs are no longer valid; we only used 781 training images and 249 test images. Since 1521 BioID images and 1030 LFPW images do not overlap with LFW + Web datasets, we train the model with all 13,466 LFW + Web images and test on all BioID and LFPW. Note that this dataset configuration is same with [27] and used identically in all comparative methods for fair comparison.

We compare our method with [4,18–20,22,27]. The results of [18,19,27] are reported in [27], and other results are collected from their released code. Note that Microsoft Cognitive Face [20] does not detect left and right eyes center, so we compare only other three keypoints. Among the comparison methods, data augmentation technique is an important factor because [4,18,22,27] are learning-based method. Therefore, we used the same data augmentation technique in all comparative experiments. [18,27] used data augmentation technique like ours to rotate, translate, and resize the image. [4,22] augment the initial shape of the keypoints, which leads the same effect as our data augmentation technique of transforming the image. Another important consideration for a fair comparison is the network architecture. In comparison with other CNN-based method [27], the network structure (depth, channels, kernel size, etc.) cannot be

Table 3 The efficiency of the shared network

Case	Type	Weights ($1e^3$)	FPS	Error (%)
Baseline	GPN	2157	885	3.52
	Unshared GPN + LPN	3392	534	2.29
	Shared GPN + LPN	2225	840	2.28
ResNet [11]	GPN	1976	613	2.52
	Unshared GPN + LPN	3768	337	2.16
	Shared GPN + LPN	2570	553	2.15

Where 'unshared GPN + LPN' means that GPN and LPN are composed of independent CNN. Although the computation time of the shared GPN + LPN is slightly slower than GPN, the performance is similar to the unshared GPN + LPN

Table 4 Experimental results on the BioID and LFPW dataset

	BioID [15]						LFPW [2]						FPS
	LE	RE	NT	LM	RM	Avg.	LE	RE	NT	LM	RM	Avg.	
Liang et al. [18]	4.11	3.88	7.03	9.06	9.23	6.66	7.09	7.03	10.58	11.52	11.33	9.51	1.67
Luxand. [19]	4.13	3.65	4.95	4.25	5.22	4.44	5.65	4.48	5.67	4.93	5.15	5.18	1953.13
Microsoft. [20]	–	–	5.31	4.33	4.08	4.58	–	–	4.75	5.37	5.26	5.12	–
Ren et al. [22]	2.14	1.89	7.57	3.02	2.74	3.47	2.94	2.95	6.35	3.59	3.67	3.90	**4166.67**
Cao et al. [4]	2.49	2.22	4.91	4.11	4.14	3.57	3.37	3.07	4.43	4.21	4.29	3.88	200.00
Sun et al. [27]	1.73	1.47	2.27	2.18	2.45	2.02	2.10	2.00	**2.16**	2.59	2.45	2.26	8.33
Ours-baseline	1.29	1.14	1.75	**1.91**	2.02	1.62	1.62	1.51	2.17	2.29	2.42	2.00	553.00
Ours-ResNet	**1.18**	**1.06**	**1.71**	1.93	**1.98**	**1.57**	**1.49**	**1.43**	2.35	**2.15**	**2.20**	**1.92**	840.00

The results are reported in normalized RMSE (%) of each component of the face (left eye, right eye, nose tip, left mouse, right mouse) and mean error. Our method shows the lowest average error on both datasets. Note that the lowest error among the comparison methods is shown in bold

exactly same because it is optimized for their own methods. For the most fair comparison as possible, therefore, the baseline model of proposed method had designed similar to VGG [25], without using relatively recent design techniques (batch normalization, residual block, and inception module).

The experimental results are presented in Table 4. We reported the errors for each of the 5 keypoints and also reported their mean errors. The optimization-based method [18] recorded the highest error on both datasets because of limited expressiveness in terms of complex texture features. [19,20] are commercial software, but recorded relatively high errors. The relatively recent methods [4,22,27], which are regression-based methods, show well performance. Particularly, [27] using the CNN's deep and discriminative features is the most competitive method and recorded the lowest error for the nose tip in the experiment on the LFPW dataset. Our method overall recorded the lowest error and achieved the lowest mean error in the both experiments.

The proposed method utilizes a large CNN, so the computational cost is very large. However, since CNN has a high proportion of simple arithmetic operations that can be easily parallelized, GPU accelerator can greatly improve the operation speed. On the other hand, existing methods use CPU because they have a high proportion of sorting or histogram operations that are difficult to parallelize. In most papers, for this reason, the computation speed (FPS) is compared instead of the computation cost of the algorithm which is difficult to measure accurately. We added the results of the computation speed comparison with the other methods in Table 4. The reported computational speed of each method comes from the published paper or the specification of API. The FPS of the proposed method is 553, which is the third fastest among the 6 comparison methods. However, the comparative methods reported higher FPS than the proposed method show about double the error. Therefore, considering the trade-off between computation speed and accuracy, the proposed method has more advantages.

To analyze our method in more detail, we represented the cumulative error distribution of the 5 keypoints in Fig. 7. The eyes show relatively lower error because they have distinctive visual features compared to the other facial components. On the other hand, the mouth corners are sensitive to changes in facial expression. Especially when the mouth is opened, the position of the mouth corners become ambiguous. The edge of nose tip is clear in the side view of the face, but the position is ambiguous in the other views. Therefore, the eyes show lower errors than the nose tip and mouth corner and more consistent (steep slope) results. Furthermore, we represented the cumulative error distribution for comparison experiments with other methods in Fig. 8. Our method not only recorded the lowest error but also showed consistent performance with low error variance. Figure 9 shows the detected keypoints by our method in the LFW + Web, BioID and LFPW datasets. Although the test images are under the various situations of poses, facial expressions, lighting conditions, and occlusions, our method well detected all keypoints.

5.3 Studies for alpha

In Eq. (7), α can be learned through the gradient decent method. In this case, the objective function is defined as minimizing the distance between the final combined position and the ground truth keypoints. α controls the influence of the global position that covers occlusion well. Therefore, α is determined according to the proportion of the occluded images in the training dataset. If the proportion of the non-occluded image in the training dataset is large, α will be close to zero to reduce the influence of the global position

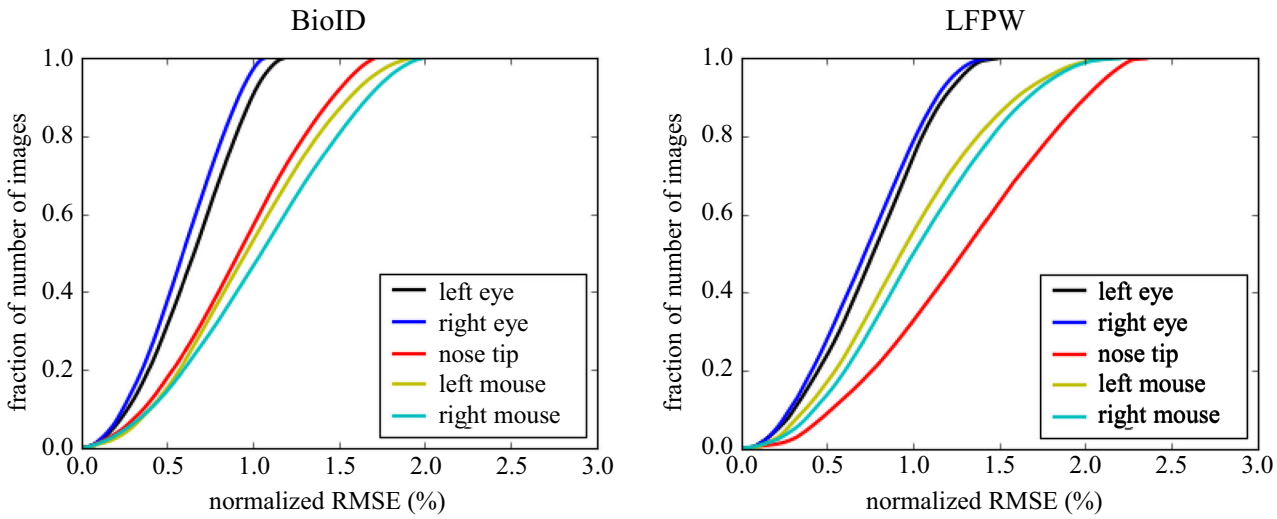

Fig. 7 Cumulative error curves for each face component of our method. The eyes have distinct features, while the nose edges are unclear and the mouth corners show various shapes according to the change of facial expression. Therefore, the eyes show lower errors than the nose tip and mouth corner and more consistent (steep slope) results

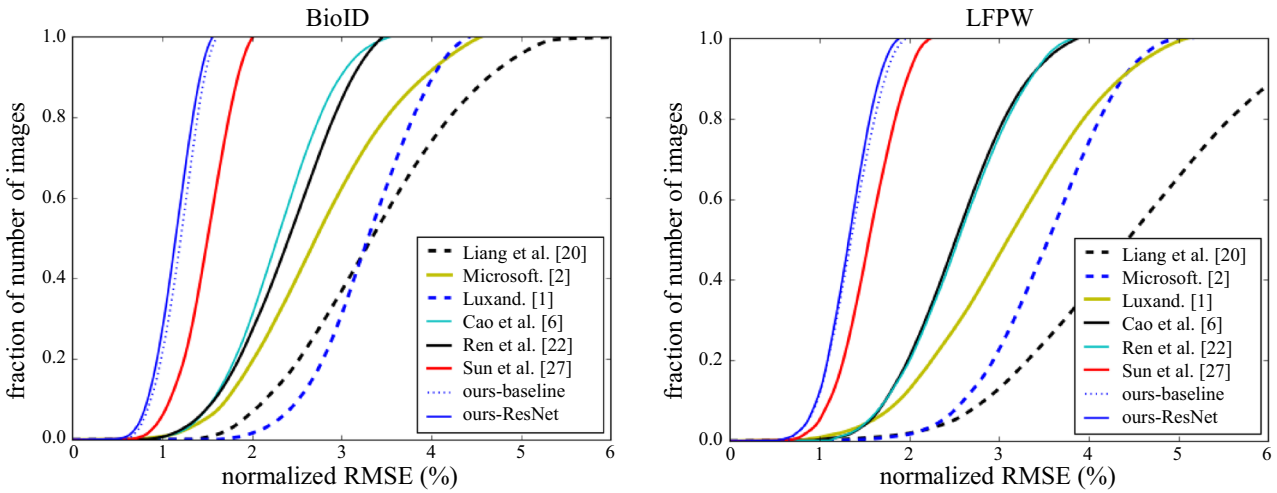

Fig. 8 Cumulative error curves for comparative experiments. Our method shows lowest error and consistent result. Note that all learning-based method [4,18,22,27] used same data augmentation technique and similar network structure for fair comparison

because the accuracy of the local position is high for the visible keypoints. Since the dataset used in the experiment does not provide a ground truth of occlusion, we do not know the proportion of the occluded images in the training dataset. To check the proportion of the occluded images in the training and test dataset indirectly, we set α to zero (the error of the occluded samples will increase because the global position information is not used), and then, counting the samples that have relatively large error. As a result, the proportion of the occluded images in training dataset (LFW + Web) was much smaller than that of the test dataset (LFPW). Therefore, when α is learned with LFW + Web dataset, α will be biased to the local position. For this reason, we did not train α to adjust the α depending on the proportion of the occluded images in the test dataset.

6 Conclusions

We propose a novel facial keypoints detection method that directly estimates and combines the global position and the local positions. Furthermore, two positions are estimated independently, but their feature extraction networks are shared so the computation time is significantly reduced. Our method detects the keypoints well even in highly fluctuat-

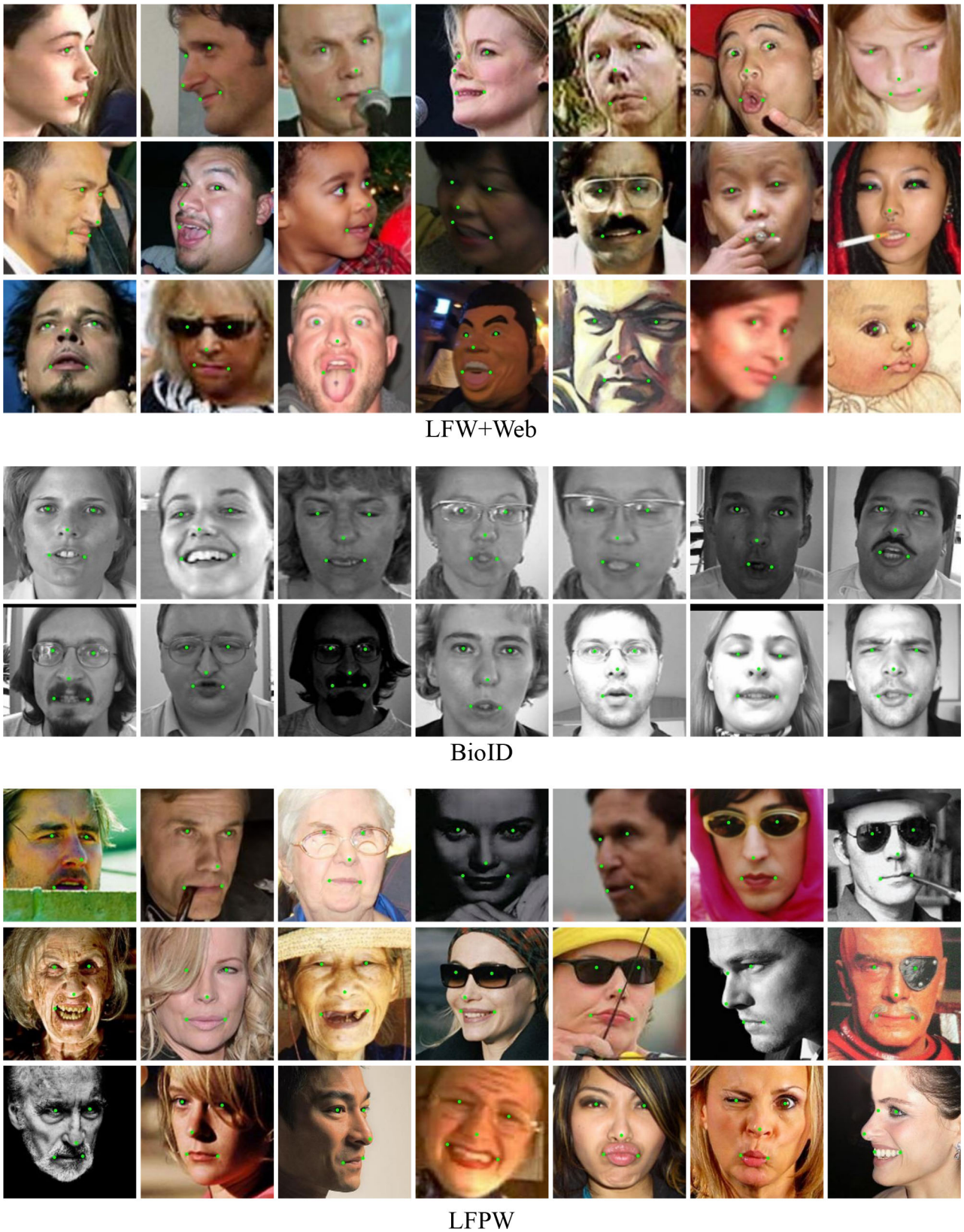

Fig. 9 Some detection results of the our method. The faces vary on head orientations, illuminations, expressions, and have partial occlusions

ing images and shows remarkable performance improvement compared with the state-of-the-art methods.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2016R1A2B4007608), National IT Industry Promotion Agency (NIPA) grant funded by the Korea government (MSIT) (No. S0602-17-1001) and Technology & Information Promotion Agency for SMEs (TIPA) grant funded by the Korea government (MSIT) (No. C0507460).

References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2037–2041 (2006). https://doi.org/10.1109/TPAMI.2006.244
2. Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2930–2940 (2013). https://doi.org/10.1109/TPAMI.2013.23
3. Berretti, S., del Bimbo, A., Pala, P.: Automatic facial expression recognition in real-time from dynamic sequences of 3D face scans. Vis. Comput. **29**(12), 1333–1350 (2013). https://doi.org/10.1007/s00371-013-0869-2
4. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2887–2894 (2012). https://doi.org/10.1109/CVPR.2012.6248015
5. Cao, Z., Yin, Q., Tang, X., Sun, J.: Face recognition with learning-based descriptor. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2707–2714 (2010). https://doi.org/10.1109/CVPR.2010.5539992
6. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. Comput. Vis. Image Underst. **61**(1), 38–59 (1995). https://doi.org/10.1006/cviu.1995.1004
7. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models, pp. 484–498. Springer, Berlin (1998). https://doi.org/10.1007/BFb0054760
8. Ding, L., Ding, X., Fang, C.: 3D face sparse reconstruction based on local linear fitting. Vis. Comput. **30**(2), 189–200 (2014). https://doi.org/10.1007/s00371-013-0795-3
9. Gidaris, S., Komodakis, N.: Locnet: improving localization accuracy for object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 789–798 (2016). https://doi.org/10.1109/CVPR.2016.92
10. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
12. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034 (2015). https://doi.org/10.1109/ICCV.2015.123
13. Hu, J., Hua, J.: Pose analysis using spectral geometry. Vis. Comput. **29**(9), 949–958 (2013). https://doi.org/10.1007/s00371-013-0850-0
14. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst (2007)
15. Jesorsky, O., Kirchberg, K.J., Frischholz, R.W.: Robust Face Detection Using the Hausdorff Distance, pp. 90–95. Springer, Berlin (2001). https://doi.org/10.1007/3-540-45344-X_14
16. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates Inc., Nevada (2012)
18. Liang, L., Xiao, R., Wen, F., Sun, J.: Face Alignment Via Component-Based Discriminative Search, pp. 72–85. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-88688-4_6
19. Luxand facesdk. http://www.luxand.com/facesdk/. Accessed 19 July 2017
20. Microsoft cognitive face. https://azure.microsoft.com/services/cognitive-services/face/. Accessed 19 July 2017
21. Milborrow, S., Nicolls, F.: Locating Facial Features with an Extended Active Shape Model, pp. 504–513. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-88693-8_37
22. Ren, S., Cao, X., Wei, Y., Sun, J.: Face alignment at 3000 FPS via regressing local binary features. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1685–1692 (2014). https://doi.org/10.1109/CVPR.2014.218
23. Saatci, Y., Town, C.: Cascaded classification of gender and facial expression using active appearance models. In: 7th International Conference on Automatic Face and Gesture Recognition (FGR06), pp. 393–398 (2006). https://doi.org/10.1109/FGR.2006.29
24. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. ArXiv e-prints (2013). http://arxiv.org/abs/1312.6229
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ArXiv e-prints (2014)
26. Singh, C., Walia, E., Mittal, N.: Robust two-stage face recognition approach using global and local features. Vis. Comput. **28**(11), 1085–1098 (2012). https://doi.org/10.1007/s00371-011-0659-7
27. Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3476–3483 (2013). https://doi.org/10.1109/CVPR.2013.446
28. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014). https://doi.org/10.1109/CVPR.2014.220
29. Xiong, X., la Torre, F.D.: Supervised descent method and its applications to face alignment. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 532–539 (2013). https://doi.org/10.1109/CVPR.2013.75
30. Zhang, C., Zhang, Z.: Improving multiview face detection with multi-task deep convolutional neural networks. In: IEEE Winter Conference on Applications of Computer Vision, pp. 1036–1041 (2014). https://doi.org/10.1109/WACV.2014.6835990
31. Zhou, E., Fan, H., Cao, Z., Jiang, Y., Yin, Q.: Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In: 2013 IEEE International Conference on Computer Vision Workshops, pp. 386–391 (2013). https://doi.org/10.1109/ICCVW.2013.58
32. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2879–2886 (2012). https://doi.org/10.1109/CVPR.2012.6248014

Je-Kang Park received his B.S. degree in the school of mechanical engineering from Pusan National University, Busan, South Korea. He is currently a Ph.D. student in the same graduate school. His current research interests are computer vision, machine learning, deep neural network, and object detection.

Dong-Joong Kang received a B.S. in precision engineering from Pusan National University in 1988 and a Ph.D. in automation and design engineering at KAIST (Korea Advanced Institute of Science and Technology) in 1998. From 2004 to 2005, he was postdoc research at Cornell university, and from 1997 to 1999, he was a research engineer at Samsung Advanced Institute of Technology (SAIT). He has been a professor at the School of Mechanical Engineering at Pusan National University since 2006 and associate editor of the International Journal of Control, Automation, and Systems since 2007. His current research interests include visual surveillance, intelligent vehicles/robotics, machine vision, and deep learning.