Robotics and Autonomous Systems 103 (2018) 1–12

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Real-time head pose estimation using multi-task deep neural network

Byungtae Ahn, Dong-Geol Choi *, Jaesik Park [1], In So Kweon

Robotics and Computer Vision Lab, KAIST, Daejeon, Republic of Korea

ARTICLE INFO

Article history:
Available online 7 February 2018

Keywords:
Head pose
Advanced driver assistance system
Deep learning
Convolutional neural network

ABSTRACT

Driver inattention is one of the main causes of traffic accidents. To avoid such accidents, advanced driver assistance system that passively monitors the driver's activities is needed. In this paper, we present a novel method to estimate a head pose from a monocular camera. The proposed algorithm is based on multi-task learning deep neural network that uses a small grayscale image. The network jointly detects multi-view faces and estimates head pose even under poor environment conditions such as illumination change, vibration, large pose change, and occlusion. We also propose a multi-task learning method that does not bias on a specific task with different datasets. Moreover, in order to fertilize training dataset, we establish and release the RCVFace dataset that has accurate head poses. The proposed framework outperforms state-of-the-art approaches quantitatively and qualitatively with an average head pose mean error of less than 4° in real-time. The algorithm applies to driver monitoring system that is crucial for driver safety.

1. Introduction

Driver inattention is a major cause of traffic accidents. According to the National Highway Traffic Safety Administration (NHTSA), many of the traffic accident fatalities and casualties in the United States during the past two years have been caused by driver inattention. In addition, about 3400 of the 35,092 US traffic deaths in 2015 were caused by driver distraction, which is 8.8% more than the 3197 deaths from the same cause in 2014. This is because the probability of a traffic accident is high due to a mistake of a driver rather than a defect in a car or a road. Therefore, if the driver inattention is automatically detected, a traffic accident can be avoided by giving a warning to the driver in advance. Driver inattention occurs mainly when the driver's distracted or tired. If this happens, the driver adopts a different head pose than usual, so driver inattention can be detected before an accident. Therefore, head pose estimation plays an important role in active safety and advanced driver assistance systems (ADAS) in intelligent vehicles.

From the viewpoint of computer vision, head pose estimation is a process of inferring the position (x, y) and direction (*roll*, *pitch*, and *yaw*) from input face images. The existing approaches can be roughly classified into two types: generative methods and discriminative methods. Generative methods use geometric clues or a variable face model. These methods output continuous head pose values rather than individual categories, and they have the advantage of obtaining facial landmarks for various applications. However, since these methods heavily rely on the detection of facial feature points, estimated head pose gets less reliable in environments where facial feature points are difficult to detect, such as large variations in head pose or facial expression, occlusion, noise, blur, and low-resolution images. Discriminative methods use machine learning methods along with visual features of the entire face. These methods are robust to challenging head poses and low-resolution images. However, most methods use facial images divided into specific head pose intervals, and classify input images into corresponding categories. Therefore, the estimates are categorized at large intervals (usually over 10°) rather than continuous values.

Head pose estimation is challenging problem in practice. Lighting changes, severe vibrations, and large pose changes frequently occur, and these affect appearance of a driver's face. In addition, it is necessary to calculate the head pose in real time and give a warning to the driver. This paper addresses these problems using a multi-task deep learning method. Our approach uses low-resolution grayscale images for real-time calculation. The proposed method was found to be superior to existing methods through qualitative and quantitative evaluation.

2. Related work

There have been several approaches to head pose estimation using an image. This section presents the related studies according to approaches, and discusses the advantages and disadvantages of representative algorithms in each category.

* Corresponding author.
E-mail addresses: joyel@kaist.ac.kr (B. Ahn), dgchoi@kaist.ac.kr (D.-G. Choi), jaesik.park@intel.com (J. Park), iskweon@ee.kaist.ac.kr (I.S. Kweon).

[1] This work is done while J. Park was with Robotics and Computer Vision Lab. He is currently with Intel Labs, 2200 Mission College Blvd., Santa Clara, CA 95054-1549, USA.

https://doi.org/10.1016/j.robot.2018.01.005
0921-8890/

2.1. Generative approaches

Generative approaches model faces with two terms of shape and appearance. Since it can generate virtual images of various poses and expressions using 2D or 3D facial models, the accuracy of the head pose is high. It outputs continuous values for head pose, not discrete classes. Hu et al. [1] roughly estimated head pose using asymmetric distribution characteristics of the facial features. The estimated poses were refined using a 3D-to-2D geometric model. Active shape models (ASMs) [2] and active appearance models (AAMs) [3] are very popular statistical models of faces. These models were proposed for facial landmark localization, but they have been extended to head pose estimation tasks [4–7]. Tawari et al. [8] proposed a method to detect facial features robustly even when large head pose changes occur by using multiple cameras. They estimated head pose using the constrained local model (CLM) and mixture of pictorial structures (MPS) as face fitting models. Their approach shows average mean error of more than 6°. Narayanan et al. [9,10] estimated the yaw angle of a driver's face using cylindrical and ellipsoidal face models. Their approach detect faces using the conventional [11] method. It made a theoretical contribution to the ellipsoidal framework, but the yaw angle estimation is inadequate for driver inattention monitoring, and the processing time is several seconds to several tens of seconds, so it is difficult to use in a real-time application. Vicente et al. [12] developed the eye-off-the-road (EOR) system, which detects facial features and poses, and estimates a driver's gaze using a commercial program, IntraFace [5,6]. The IntraFace algorithm handles the nonlinear least squares (NLS) problem of parameterized appearance models by suggesting a supervised descent method (SDM), which showed the best performance among the approaches discussed with an average mean error of 4.6° at 25 fps.

However, these model-based generative methods are very sensitive to pose changes in low-quality images, such as vibration and noise, low resolution, lighting changes, and large pose changes, since the accuracy of head pose heavily depends on facial feature detection.

2.2. Discriminative approaches

These methodologies directly infer the relationship between appearance in 2D images and 3D head pose. This is done by learning a regressor that implicitly specifies the important factors used in head pose estimation. Balasubramanian et al. [13] and Foytik et al. [14] proposed a manifold embedding framework that maps the high-dimensional space of a facial image to a low-dimensional manifold. Foytik et al. proposed a two-step method for estimating head poses in a coarse-to-fine manner for manifold embedding. Gruji et al. [15] proposed a method to find the most similar head orientation with an input face image compared to images already obtained and stored in databases. The estimated initial head orientation value is refined again using candidate images from the database. However, the experimental results of the methods proposed by Foytik et al. and Gruji et al. for the Pointing04 dataset [16] show errors larger than 13°. Huang et al. [17] proposed a discrete head pose label classifier with random forests using Gabor features. They combined random forests with linear discriminative analysis (LDA) to improve discriminative performance. Zhu et al. [11] proposed an integrated algorithm for face detection, head pose estimation, and facial feature localization. They used a mixture of tree-structured part models to find the relationship between shape changes and yaw angles. However, even if it is an integration framework, it only categorizes the input image into several categorized yaw angles at 15° intervals, and it takes a lot of processing time (several to several tens of seconds) to process a 640 × 480 resolution image.

In contrast to these discrete labeling approaches, BenAbdelkader et al. [18], and Ji et al. [19] have treated head pose estimation as a nonlinear regression problem that computes a continuous 3D pose parameters. Several studies have used depth information with machine learning techniques for continuous head pose estimation. Breitenstein et al. [20] estimated a head pose by aligning the range image with the reference pose using their alignment error function and showed an operation speed of 10 fps based on the GPU-based implementation. Fanelli et al. [21,22] introduced a random forest based voting framework for real-time continuous head pose estimation. They applied the framework to 3D facial landmarks localization tasks, and they also provided a head pose dataset, the Biwi Kinect Head Pose Database, consisting of color images, depth maps, and ground truth head pose data. Due to the depth information, these methods achieve relatively strong results at night and have the advantage of obtaining a 3D face model. However, they have the disadvantage that they require a special device, like Kinect, and can be used indoors only.

2.3. Deep convolutional neural networks

As the graphics processing unit (GPU) evolves and access to big data becomes more convenient, deep learning technologies have been actively pursued. Among these deep learning methods, the convolutional neural network (CNN) [23] has been widely used in various fields and has been developed into AlexNet [24], GoogLeNet [25], and ResNet [26]. Also, these deep neural networks (DNNs) have been applied to face related applications. Sun et al. [27] introduced DNN into coarse-to-fine facial feature localization. They proposed a three-stage cascade structure consisting of one DNN and two shallow neural networks, and they analyzed the impact of techniques such as absolute value rectification and local weight sharing of facial feature localization. Li et al. [28] proposed a CNN cascade structure for robust face detection in various environments where pose, facial expression, and lighting vary. Zhang et al. [29] improved the cascaded CNNs [28] to detect face and five facial landmarks. Ahn et al. [30] proposed a method of learning the mapping relationship between visual appearance and 3D head direction using DNN. They experimented and selected various DNN parameters to design a DNN structure for head orientation estimation. To stabilize the estimate, particle filter tracking was applied as a post-processing method. Based on the assumption that a face is correctly detected, the network estimates a relatively accurate head direction with an error of about 3° at over real-time. However, their method lacks the face detection step, which is essential for head pose estimation, and their evaluation method does not guarantee independence between training and test data, so it is difficult to get the reported performance in a real environment.

2.4. Multi-task learning

Multi-task learning has proved effective for many computer vision problems [31–34]. Multi-task learning is based on the idea that the learned features in one task are useful for other tasks as well, so DNNs can be successfully applied to multi-task learning. Previous studies, such as [28,35], have presented relatively simple ways of learning each individual network and cascading the results. Deep learning methods inevitably lead to overfitting problems for given learning data if the learning data cannot represent all cases (in other words, the diversity and quantity of data are not sufficient). Therefore, a scheme like drop out is used to train more general features. However, this problem remains intact for the methodologies of cascading individual networks. For each partial task dataset, each network may be overfitting, which may degrade the performance of the combined system.

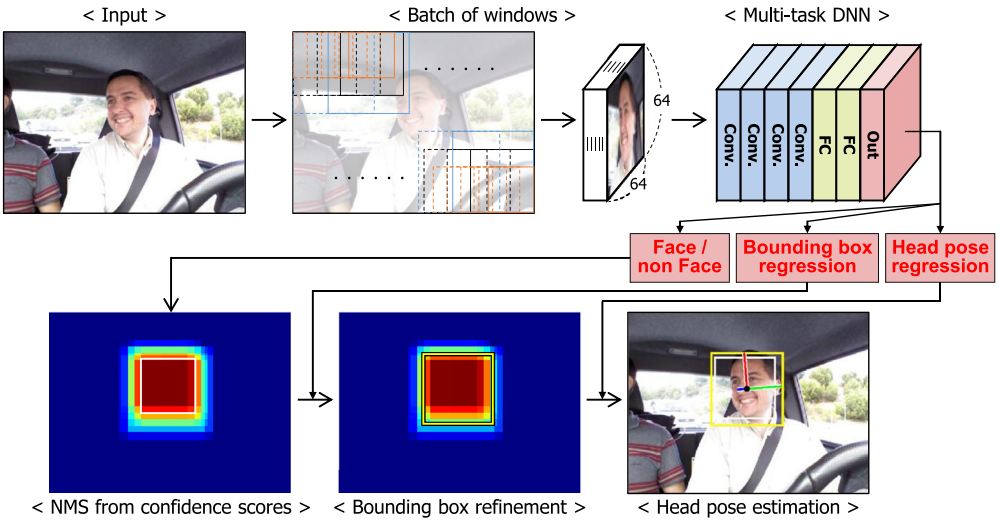


Fig. 1. Overview of our real-time multi-view face detection and head pose estimation system using multi-task DNN. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Multi-task learning simultaneously learns common features for interrelated tasks. Due to the common generic features learned, the features are not biased against the dataset of a particular task, so the performance of the overall system is more synergistic than when the attributes of each task are separately learned. For this purpose, Li et al. [36] jointly learned the images of eight body parts for body pose estimation; Zhang et al. [29] learned facial landmarks, gender, and facial expression together for face alignment; and Jung et al. [37] jointly learned face appearance and facial landmark points for facial expression recognition. The effects of multi-task learning are discussed in detail in 4.1. It is also an additional advantage to use only a series of common features, even if one does not have task-specific features for each task.

To incorporate existing multi-task learning techniques into deep learning, a database with ground truth labels for all tasks is required. However, it is difficult to obtain sufficient databases in practice. To handle this difficulties, we propose an algorithm that can train multi-task DNNs with individual datasets correspond to each task. Our integrated framework detects face and estimates 5D (x, y, *roll*, *pitch*, and yaw) head pose. We also propose an algorithm that can train multi-task DNNs with individual datasets correspond to each task. The proposed system shows excellent performance in real time with low-resolution grayscale images, and it is robust to various environments and large head pose changes. The contributions of this paper are summarized as follows:

1. To propose a multi-task DNN for multi-view face detection and head pose estimation.
2. To propose a multi-task learning method whose datasets' labels are not jointly annotated.
3. To propose a head pose estimation algorithm that is robust to various environments using DNN.

3. Proposed method and datasets

In this section, we first give an overview of the proposed multi-view face detection and head pose estimation algorithm. The next sections discuss the details of the multi-task learning network and datasets.

3.1. Overview of the proposed algorithm

Fig. 1 shows an overview of our real-time multi-view face detection and head pose estimation algorithm. We address the three issues of face detection, bounding box refinement, and head pose estimation by using shared features learned through multi-task learning. The process of the whole algorithm is briefly described as follows: A batch of detection windows is created by splitting the entire input image into many bounding boxes with a stride proportional to size of the bounding box. The bounding box is composed of various sizes for multi-scale detection. It is resized to the input size of the network, $64 \times 64 \times N$. The first task in the network, binary classification for face detection, is performed, resulting in a heat map according to the confidence value of each detection window. To prevent duplication, non-maximum suppression (NMS) is applied and an initial bounding box is obtained. The bounding box regression, the second task, is performed with appearance of the initial bounding box. With these procedure, a bounding box of the most appropriate position and size is obtained. The final task is head pose estimation. We use a traditional right-handed Cartesian coordinate system with roll, pitch, and yaw angles to represent head orientation. By definition, the roll and pitch are angles that rotate clockwise with respect to the x (blue) and y (green) axes, respectively, and yaw represents the angle of rotation about the z axis (red) counterclockwise. Note that the shared features in the multi-task learning network shown in Fig. 2 are used in all three tasks.

3.2. Multi-tasks learning deep neural network

The most relevant research to ours is that by Ahn et al. [30], which was the first study that exploited CNN to estimate head orientation. They designed DNN that learned the mapping relationship between visual appearance and 3D head orientation. However, their algorithm assumes that the face is detected correctly, and estimates only the 3D head orientation, which limits its practical use to various applications. In this paper, by extending previous research, the proposed DNN includes three tasks of face detection, optimal bounding box extraction, and head pose estimation, as shown in Fig. 2. The input to the network is a 64×64 grayscale image. The feature extraction stage has four convolution layers, three pooling layers, one shared fully connected layer, and one fully connected layer for each task. A rectified linear unit (ReLU) [38] was used as the activation function, and max-pooling was used in the pooling layer. The fully connected layer, which follows the four convolution layers, creates a 64-dimensional feature vector that is shared by the multi-task estimation stage.

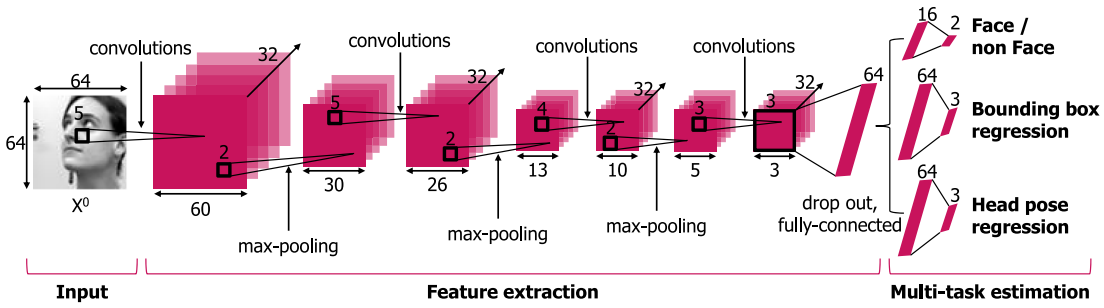


Fig. 2. The proposed multi-task learning DNN and its parameters.

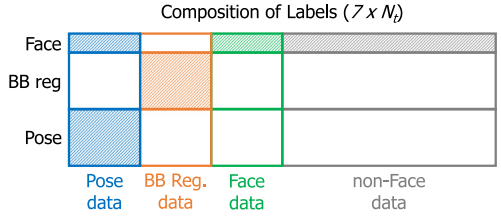


Fig. 3. Composition of labels from different datasets. The shaded area denotes the corresponding dataset has labels.

3.2.1. Deep joint learning

For multi-task learning, ground-truth labels for each task must be annotated in a dataset. If there is a dataset with labels for several tasks, features for those tasks can be jointly learned at a time in the training phase. A unified training model with global cost function estimates all tasks' labels simultaneously, and a number of multi-task learning studies have used this methodology [36,37,29]. However, this method cannot be used when there is no jointly annotated dataset for all the tasks. Furthermore, in the case of the bounding box regression task, samples overlapping with the ground-truth bounding box by 50% is also ambiguous to attach the face or non-face labels for the face detection task. Due to this nature of the bounding box regression task, multi-task learning networks that include the bounding box regression task cannot be jointly trained in one step. In this work, we use a mixed dataset that is partially annotated, as shown in Fig. 3. Since the composition ratio of the number of images for the classification task (binary face classification) and regression tasks (bounding box regression and pose estimation) is different for each training batch, the global cost function to bundle them uniformly is not appropriate.

We built a whole training database by fusing several kinds of datasets which have labels of corresponding task only, and we propose a suitable multi-task learning method. In addition, since the proposed method does not require jointly annotated data which is hard to obtain, it is advantageous to supplement the entire database if there is another dataset for each task. Fig. 3 represents the annotation composition of labels of the database. It has dimensions of $7 \times N_t$. Here, N_t is the total number of image patches in the database, and 1D label for binary classification for face detection, 3D (x, y, and w) labels for bounding box regression, and 3D (*roll*, *pitch*, and *yaw*) labels for head pose estimation. The shaded areas indicate annotated labels, and color means different kinds of datasets. In the training phase, we sequentially update features of each task using data for each task in a training batch: Features are updated for one task, and then the updated features are applied to the next task and are updated again for the task. The update method is based on the traditional backpropagation algorithm. A training batch contains all the data for three tasks: face detection, bounding box regression, and head pose estimation. The training process is described in Algorithm 1. Note that, since each task in the network can train with other datasets for the task, one can freely supplement or subtract datasets for each task.

Algorithm 1 Joint learning multi-task datasets whose labels are not jointly annotated.

```
for <# of training epoch> do
   Shift training sequence circularly
   for <# of batches> do
      1. Select next task
      2. Pick samples in the batch for the selected task
      3. Update the network for this task by stochastic gradient descent w.r.t the selected samples
   end for
end for
```

This cascading training method can bias the entire feature to one specific task based on the training order. To prevent this issue, training sequence is circularly shifted for each new training batch. As a result, the errors of all tasks converge without being biased to a specific task, as shown in Fig. 4(b). Fig. 4(a) shows the loss errors when training is conducted in the order of head pose, bounding box, and face detection, and Fig. 4(b) shows that of applying the circular shifting training sequence.

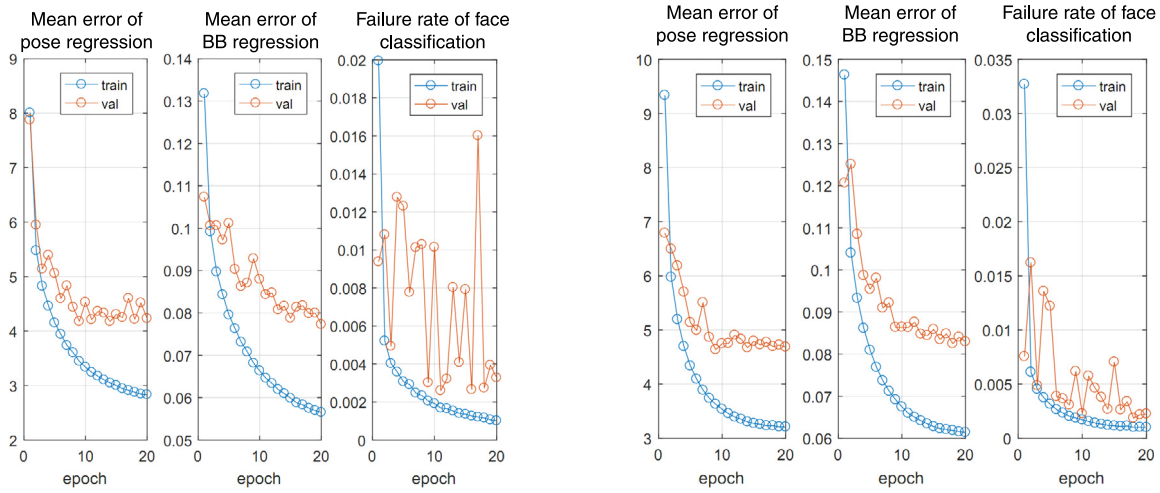
3.2.2. Face detection

A face is detected by binary classifier. The cost function of the output layer uses the softmax function of Eq. (1). To densely scan the input image, we create a batch of detection windows (image array) with multiple scale (three in this study) sliding windows. The batch is resized to the input size of the proposed network, $64 \times 64 \times N$. Here, $N = [(W - s_w)/s_t + 1] \times [(H - s_w)/s_t + 1]$, where s_w indicates the size of detection window, s_t denotes the stride value, and the size of the input image is $W \times H$. After binary classifier applied, N confidence score boxes are produced, which is the input of the non-maximum suppression to eliminate highly overlapped detection windows for the initial bounding box. Fig. 5 shows a heat map composed of the confidence scores, and the initial bounding box is represented as a white box.

$$p(S(\mathbf{X}))_k = \frac{e^{S_k(\mathbf{X})}}{\sum_{j=1}^{K} e^{S_j(\mathbf{X})}} \tag{1}$$

where $S(\mathbf{X})$ is a vector containing the scores of each class for the instance $\mathbf{X}$, $p(S(\mathbf{X}))_k$ is the estimated probability that the instance $\mathbf{X}$ belongs to class given the scores of each class for that instance k, and K is the number of classes. In our experiment, K is 2.

To improve the performance of the final pose estimation, the result of non-maximum suppression is not directly used to estimate the pose. We added a bounding box regression task to the multi-task learning network to estimate a more appropriate bounding box position and size.



(a) Training errors without circular shifting training sequence.

(b) Training errors with circular shifting training sequence.

Fig. 4. Training errors for head pose regression, bounding box regression, and binary face classification.

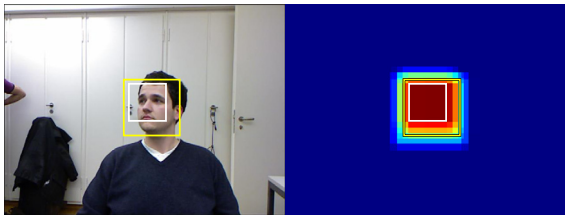


Fig. 5. An input image and corresponding heat map for face detection. White box: initial bounding box obtained from heat map. Yellow box: refined bounding box by bounding box regression task. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. Some examples in bounding box regression training data, which are ambiguous to annotate face or non-face label for face detection task.

3.2.3. Bounding box regression

The traditional methodology of pose estimation proposed by Zhu et al. [11] and Xhing et al. [5,6] first detects faces from an input image and then performs pose estimation with the detected window. However, simply performing the pose estimation with the detected window may degrade performance. This is because the face detector is generally considered positive if the detection window overlaps the ground-truth box by more than just 50%. In the pose estimation step, it is important to detect the major components of the face. Although the face is detected, if the face detector does not produce tight bound for facial region, it may miss important part of the face such as eyes, nose, mouth, and chin, the accuracy of the pose estimation results will be much lower. Fig. 6 shows these examples. Also, the pose estimation result will be very unstable in the temporal domain due to the unstable detection box positions. In other words, even if the face detection success rate is high, the performance of head pose estimation may be very low. Therefore, it is important to extract the bounding box of optimal position and size.

Fanelli et al. [22] solved this partial observation problem inherently by detecting the face region based on depth information using Kinect depth sensors. Our method is an appearance-based approach that uses only gray scale camera. To handle this problem, we introduced the bounding box regression technique using a DNN. The face detection box regression is a face-related task, such as face detection and head pose estimation, so the proposed multi-task learning network was constructed assuming that the same features are shared. The proposed idea is related to R-CNN [39]. It used class-specific bounding box regression with a separate CNN. In the training phase, the input is a set of N training pairs $(\mathbf{i}, \mathbf{g})$ where $\mathbf{i} = (i_x, i_y, i_s)$. i_x, i_y is the pixel coordinates of the upper left corner of the initial bounding box, and i_s is the size of one side of the square bounding box. Each ground-truth bounding box is $\mathbf{g} = (g_x, g_y, g_s)$ in the same way. The goal is to learn the transformation to map the initial bounding box $\mathbf{i}$ to ground-truth box $\mathbf{g}$. The transformation is parameterized to three terms $dx(\mathbf{i})$, $dy(\mathbf{i})$, and $ds(\mathbf{i})$. The first two represent the scale-invariant translation of the upper left corner of the bounding box $\mathbf{i}$, and the third term represents the log-space translation of the size of the square bounding box $\mathbf{i}$. By these functions, an input proposal $\mathbf{i}$ is transformed to a predicted ground-truth box $\hat{\mathbf{g}}$ via the following transformation:

$$\begin{aligned} \hat{g}_x &= i_s d_x(\mathbf{i}) + i_x \\ \hat{g}_y &= i_s d_y(\mathbf{i}) + i_y \\ \hat{g}_s &= i_s e^{d_s(\mathbf{i})} \end{aligned} \tag{2}$$

Transformation terms $d_*(\mathbf{i})$ (where $*$ is one of x, y, s) are modeled as a linear function of the last convolution layer's features of $\mathbf{i}$, denoted by $\phi(\mathbf{i})$. Therefore the transformation terms are modeled by $d_*(\mathbf{i}) = \mathbf{w}_*^T \phi(\mathbf{i})$, where $\mathbf{w}_*$ is a trainable parameter vector. $\mathbf{w}_*$ is learned by optimizing the cost function as follows:

$$\mathbf{w}_* = \underset{\hat{w}_*}{\operatorname{argmin}} \sum_k^N (t_*^k - \hat{\mathbf{w}}_*^T \phi(\mathbf{i}^k))^2 + \lambda \|\hat{\mathbf{w}}_*\|^2 \tag{3}$$

The regression target function t_* for the training pair $(\mathbf{i}, \mathbf{g})$ is defined as follows:

$$\begin{aligned} t_x &= (g_x - i_x)/i_s \\ t_y &= (g_y - i_y)/i_s \\ t_s &= \log(g_s/i_s) \end{aligned} \tag{4}$$

3.2.4. Head pose regression

As shown in Fig. 2, the final task is head pose regression. For this task, the loss function uses the Euclidean distance described

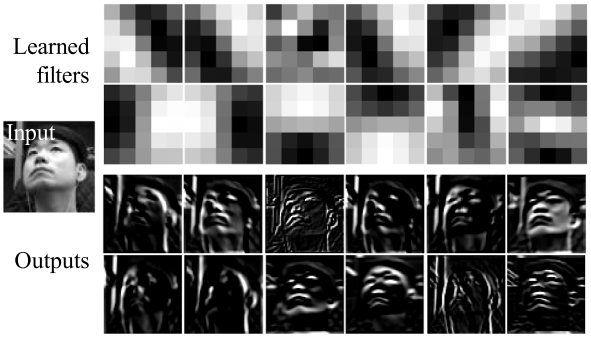


Fig. 7. Several trained features and corresponding outputs in the first convolution layer.

as follows:

$$E(\mathbf{X}^k; \mathbf{W}) = \sum_k \|\mathbf{g}^k - f(\mathbf{X}^k; \mathbf{W})\|_2^2 \quad (5)$$

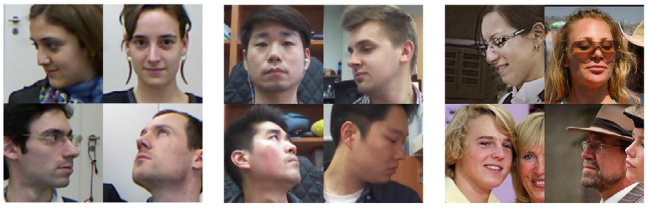
where $\mathbf{X}$ is an input image of 60 × 60 px size. k is an index of a training sample, and f is the estimated head pose (*roll*, *pitch*, and *yaw*) in normalized degree unit. $\mathbf{W}$ is a set of features, weights of the convolution filters, and $\mathbf{g}^k$ is the ground-truth head pose.

Fig. 7 shows some of the learned features and corresponding outputs in the first layer of the proposed DNN. The dimensions of the features and outputs are 5 × 5 and 60 × 60, respectively. The features exhibit structure and are uncorrelated, which proves the features are well trained. The outputs show that the parts that are important to estimate the head pose (eye, nose, head, jaw, and etc.) are highlighted. Note that the emphasized parts of the outputs are high-level features commonly used for face detection, bounding box refinement, and head pose estimation.

3.3. Datasets

3.3.1. Training data

We used three datasets: the Biwi Kinect Head Pose (BKHP) dataset [22], a new RGB image dataset (RCVFace), and the Annotated Facial Landmarks in the Wild (AFLW) dataset [40]. The BKHP dataset consists of 15,678 images of the upper body of 20 people (four subjects appear twice but have different clothes and hairstyles). The ground-truth head pose information was obtained by fitting user-specific 3D templates constructed from a depth map to the user. The user-specific 3D templates were built off-line [41]. The BKHP dataset provides head orientation in the form of a 3D rotation matrix. The head orientation covers [−50°, 50°] for roll, [−60°, 60°] for pitch, and [−75°, 75°] for yaw. In this work, the color images were used by converting them to gray scale, but the color images of the BKHP dataset consist of only two almost homogeneous white backgrounds. If the network is trained with only this dataset, features are overfitted on white homogeneous backgrounds and they may fail to estimate head pose on cluttered backgrounds. The amount of data, only 24 subjects, is too small to train a general face detector. To solve this problem, we added the RCVFace and AFLW face detection datasets. The RCVFace data, in the same format as the BKHP data described above, comprises images obtained from an additional 23 people, about 30,000 on various backgrounds. The sequential images with the head pose were labeled off-line by a state-of-the-art template-based head tracker as the same way with BKHP dataset [22]: when a user moves the head in front of the depth sensor, the scanned point cloud is integrated and fit to the generic template model [41]. The user-specific 3D head model is used for accurate head pose tracking. In this study, we used 37 subjects out of 47 from the BKHP and RCVFace datasets for training, and 10 subjects for validation. We decided to use AFLW dataset for face detection task as it consists of various poses on various backgrounds. It is worth to note that other datasets [42–46] are composed of frontal or near frontal faces. Samples of the three datasets are shown in Fig. 8.



(a) BKHP. (b) RCVFace. (c) AFLW.

Fig. 8. Some samples of the training datasets: (a) Biwi Kinect Head Pose dataset. (b) RCVFace dataset. (c) Annotated Facial Landmarks in the Wild dataset.

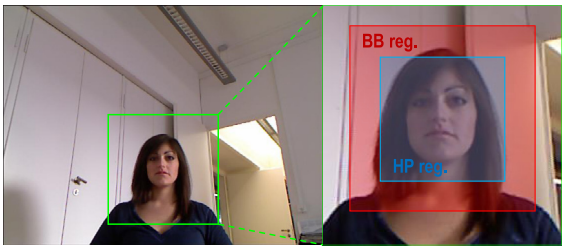


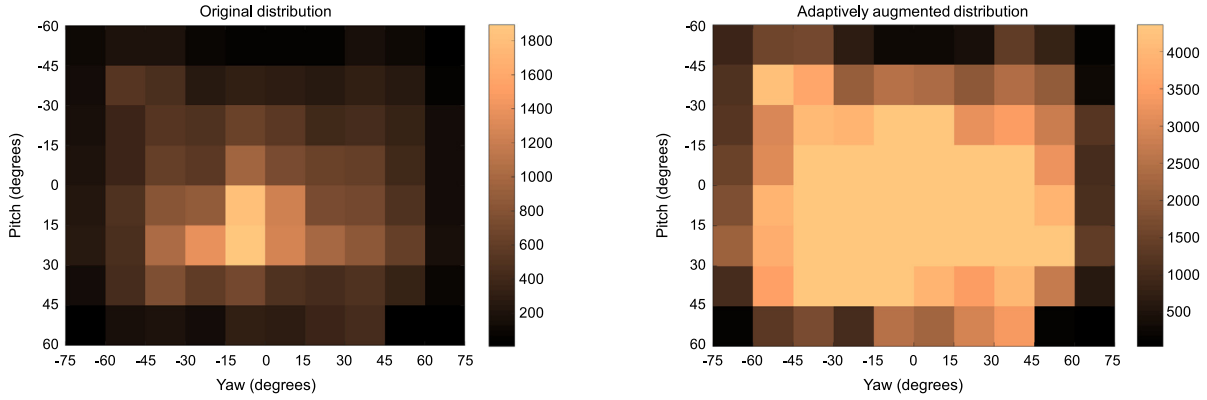
Fig. 9. Ranges for data augmentation: Red region is for bounding box regression. Blue region is for face detection and head pose regression. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3.2. Ranges for data augmentation

In training CNNs, training data are augmented by jittering to ensure that the learned features have translation invariance characteristics. Several data augmentation approaches have been introduced so far, such as flipping and rotating for rotation invariance, and adding noise and blur for robustness [24]. This ensures that the detected bounding box can be correctly detected or classified by the tolerance of features location inside the box, even if it does not exactly match the ground-truth bounding box. In this work, training data were augmented with small jittering, including translation and zooming, to train features robust to bounding boxes of incomplete position and size (about 80% overlapped with the ground-truth box). Fig. 9 shows the data augmentation ranges covered by each task. In the region overlapped with the ground-truth box more than 80% (blue area), training data were augmented for translation and scale invariance. Training data for the bounding box regression task were generated in the region overlapped with the ground-truth box more than 25% (red area). We added negative samples twice as positive face patches to the training patches for the face detection task. Finally, a total of about 300,000 grayscale image patches of 64 × 64 px size were used to train the proposed multi-task learning DNN.

3.3.3. Data sampling for uniform distribution

Most face-related datasets have an overwhelming number of upright front face images compared to other angles. Fig. 10(a) shows the distribution of pitch and yaw angles in the database containing BKHP and RCVFace datasets, and the majority of images are of frontal or near frontal faces. Thus, if the all of the images of this dataset are used without careful sampling, the learned features will be over-fitting to front faces. This will result in similar performance to the conventional front face detector [47], which has



(a) Original distribution.

(b) Adaptively augmented distribution for uniform distribution.

Fig. 10. Distribution of pitch and yaw angles in the database containing BKHP and RCVFace datasets.

difficulty in detection when large pose changes occur. To handle this problem, we split the ground truth data into 80 (10 in yaw and 8 in pitch) bins, and uniformly selected and adaptively augmented the samples from each bin. As a result, the distribution of the training data shows a uniform distribution over a wider range as shown in Fig. 10(b). This allows features to be generalized so that the network does not overfit only on frontal faces.

4. Experimental results

In this section, we demonstrate our real-time head pose estimation algorithm. We report experiments that we conducted on various aspects to quantitatively and qualitatively verify the performance of the proposed algorithm. We verified the validity of the proposed multi-task learning methodology, tested the performance of the proposed algorithm in face detection and head pose estimation, and compared it with those of state-of-the-art algorithms. Finally, we present the results of our algorithm with images under harsh environments.

The datasets used in the evaluation were the BKHP [22], RCVFace, AFLW [40], and Naturalistic Driving Study (NDS) datasets [48]. We decide to use these dataset, because most of the existing head pose datasets annotate the ground truth value in their own way without a uniform reference. The performance of the head pose estimation was evaluated with the validation set in the database used in this work and compared with that of a state-of-the-art algorithm which used the same database. The face detection performance was evaluated on the NDS dataset and compared with that of a state-of-the-art algorithm using the same dataset. The NDS dataset contains 41 driver face images. Each sequence consists of a 360 × 240 relatively low-resolution video clip recorded at 15 fps, and the dataset includes a number of extreme head poses (*e.g.* −90° for yaw and −50° for pitch) and extreme illumination changes (*e.g.* direct sunlight and IR lighting at night), making it a suitable test set for the purposes of this research.

4.1. Comparison with cascading individual networks

To verify the validity of the proposed multi-task learning methodology, we compared the proposed method that trains a common set of features for three tasks and a method that trains each feature set using a single DNN for each individual task. The training data for both DNNs was the same, taken from the BKHP, RCVFace, and AFLW datasets. Fig. 11(a) shows the results of a validation set consisting of 64 × 64 px grayscale image patches for each task. The left side shows the mean error of the head pose,

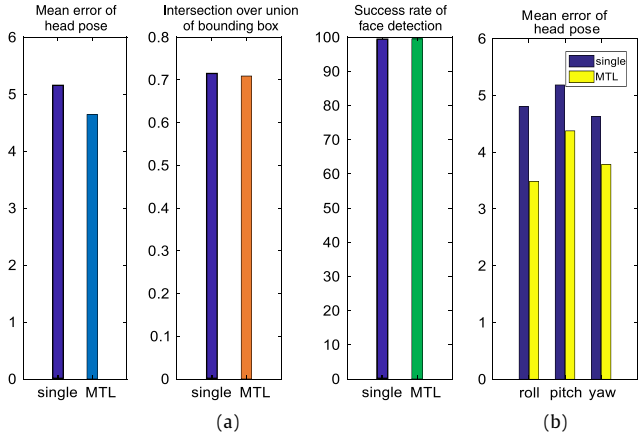


Fig. 11. Comparison of the cascaded three individual DNNs and the multi-task DNN (a) Results on validation set consisting of 64 × 64 px grayscale image patches for each task (b) Results of head pose estimation of the cascaded three single DNNs and the multi-task DNN.

the middle shows the intersection over union (IoU) value of the bounding box (Eq. (5) is used as the cost function in training phase, but the IoU value was used in the evaluation for the intuitive understanding.), and the right side shows the success rate of the face classification task. Fig. 11(a) shows that the results of single DNNs and the multi-task learning DNN were almost the same for the same validation data, which are comprised small image patches. This is the natural result of training networks of the same structure with the same dataset (note that, the individual task modules in the multi-task learning DNN have the same structure as the corresponding single DNN for each task). On the other hand, Fig. 11(b) shows the experimental results of both DNNs with the proposed algorithm (from sliding windows to head pose estimation) shown in Fig. 1. We compared the method that cascade three individual DNNs and the proposed multi-task learning DNN. The performance of the multi-task learning method is improved by about 22.7% compared with the method of cascading individually learned DNNs. Compared to Fig. 11(a), the performance of the overall system is improved because the features are not biased in a specific training dataset of the individual task.

4.2. Face detection

We utilize the NDS dataset for validating face detection. The NDS dataset consists of 41 driver images. Twenty of these are

Table 1
Success rates of face detection of the stopped/parked environment on NDS dataset [48] (unit: %).

Algorithm	Night	Transition	Day
GOTS	79.0	89.1	66.0
GOTS Tracked	84.7	92.9	75.5
IntraFace	62.6	78.7	74.3
MTCNN	84.0	93.0	84.5
Ours	**89.7**	**95.4**	**92.9**

Table 2
Success rates of face detection of the driving environment on NDS dataset [48] (unit: %).

Algorithm	Night	Transition	Day
GOTS	79.1	83.7	75.5
GOTS Tracked	87.9	88.9	83.8
IntraFace	64.7	68.3	64.8
MTCNN	87.4	90.7	91.3
Ours	**92.2**	**92.6**	**93.7**

videos recorded in a stopped/parked environment where the vehicle is not in motion, and 21 are videos recorded in a driving environment that is running for about 30 min. In a stopped/parked environment, the driver performs various tasks such as looking around, taking off glasses, taking a cell phone call, and so on. Data were recorded at day, night, and day and night transition period, and only IR illumination was used at night. The proposed algorithm was compared with the following four commercial programs and open sources.

GOTS and GOTS Tracked [48] This method uses government off-the-shelf (GOTS) software that works well on non-frontal faces. The tool performs face detection, tracking, face landmark detection, head pose estimation, and face recognition tasks, and has been the benchmark for performance evaluation. Experiments were performed in two modes: simple face detection mode and detection with tracking mode (GOTS Tracked) for an improved detection rate. For more information, see [48].

IntraFace [5,6] This is a publicly-available software package for face detection, face landmark detection, and head pose estimation proposed by Xiong et al. They proposed the supervised descent method (SDM) to solve the non-linear least squares (NLS) problem of parameterized appearance models, and it showed the best performance among generative approaches.

MTCNN [29] This method is one of the state-of-the-art methods for face detection based on deep learning. They utilized WIDER FACE [49] and CelebA [50] datasets to train their network to detect face and five facial landmarks.

The existing methods used in our experiments were tested with each pre-trained model. Tables 1 and 2 show that the face detection performance of our method is superior to that of the state-of-the-art methods under all environmental conditions of the NDS dataset. It is noteworthy that IntraFace, a state-of-the-art algorithm in face detection and face landmark localization, showed the worst performance. It is because facial model was unable to fit into the face image, due to NDS dataset's poor resolution, noise, and extreme lighting and pose changes. This is an inherent problem for most of the generative approaches. Deep learning based methods ([29] and ours), on the other hand, showed relatively good performance. Representative examples are shown in Fig. 12.

To consider only environmental variables, the driver was fixed, and the results were compared for five different vehicle environments: lighting change, large pose change, night time, wearing sunglasses, and occlusion. Overall, the proposed method was stable in most cases, whereas the generative approach showed unstable results or face detection failures due to unsatisfactory facial landmark localization in low-resolution images. In our experiment, the size of the sliding window was set to three scales (68, 112, and 192), and the stride size was set to 1/4 of the sliding window. In the case of 320 × 240 resolution images, a total of 211 windows per frame are extracted and processed at a speed of about 33 fps enable real-time application. The test system environment was a 3.6 GHz Intel Core i7 CPU with Nvidia™ GeForce GTX 1080 GPU. Even with relatively large stride values, the appropriate bounding box position and size can be estimated thanks to the bounding box regression step, resulting in excellent performance with relatively little computation. On the other hand, in the case of full search with the stride size of 1, a total of 76,112 windows per frame should be computed at the same scales.

Table 3
Comparison of head pose estimation results of the proposed algorithm with Drouard et al.'s [51] and Fanelli et al.'s method [22].

	Mean error ± Std deviation (°)		
	Roll	Pitch	Yaw
Drouard et al.	4.9 ± 4.1	5.9 ± 4.8	4.7 ± 4.6
Fanelli stride 15	5.5 ± 6.2	3.8 ± 6.4	4.2 ± 7.8
Fanelli stride 10	5.5 ± 6.2	3.6 ± 6.0	4.0 ± 7.1
Fanelli stride 5	5.4 ± 6.0	3.5 ± 5.8	3.8 ± 6.5
Ours w/o BB reg.	4.5 ± 4.4	5.1 ± 4.3	5.4 ± 5.1
Ours with BB reg.	**3.4 ± 3.9**	**4.3 ± 3.7**	**3.6 ± 3.1**

4.3. Head pose estimation

We compared our method with two state-of-the-art methods of the discriminative approaches. To estimate head pose, Drouard et al. [51] used probabilistic high-dimensional regression with a color image, and Fanelli et al. [22] used random forest regression with Kinect depth sensors. Also, they provided and used the Biwi Kinect Head Pose dataset used in our system. Table 3 shows the comparison results on mean error and standard deviation. The accuracy and precision values of our method were better than those obtained with the state-of-the-art algorithms. Drouard et al. [51] considered head pose estimation issue as a high-dimensional to low-dimensional mapping problem. They mapped high-dimensional HOG-based region descriptors onto low-dimensional head poses by learning a mixture of linear regression model. The method proposed by Fanelli et al. [22], votes head pose by comparing the internal depth values of randomly extracted patches, while our DNN-based approach uses features that are automatically learned from a number of training images. As a result, important high-level information (relative positions of eyes, nose, and jaws) is extracted implicitly. In addition, depth sensors using infrared cannot be used outdoors during the day, whereas our method can only operate with gray-scale images during the day and at night. The bounding box regression step refines position and size of a coarsely detected bounding box in the first step, so it improves not only the accuracy but also stability of head pose estimation as shown in Table 3. Furthermore, since the second and third step, bounding box regression and pose estimation, are applied only to the bounding box detected as a face in the first step, the processing time of the bounding box regression step is less than 1 ms per bounding box. Some estimation results are shown in Fig. 14. The white box is the initial bounding box after the non-maximum suppression, and the yellow box is the bounding box regression result. The figure shows that the proposed algorithm works well for various facial expressions and poses.

Fig. 15 shows the normalized success rates of the estimates every 15-by-15 degrees compared to that of [22], Angular error below 15° is considered a success, and the background color reflects the number of images in that area. As shown in Fig. 15(a), estimates are 100%, or close to 100% in most areas, and outperform those of [22] shown in Fig. 15(b).

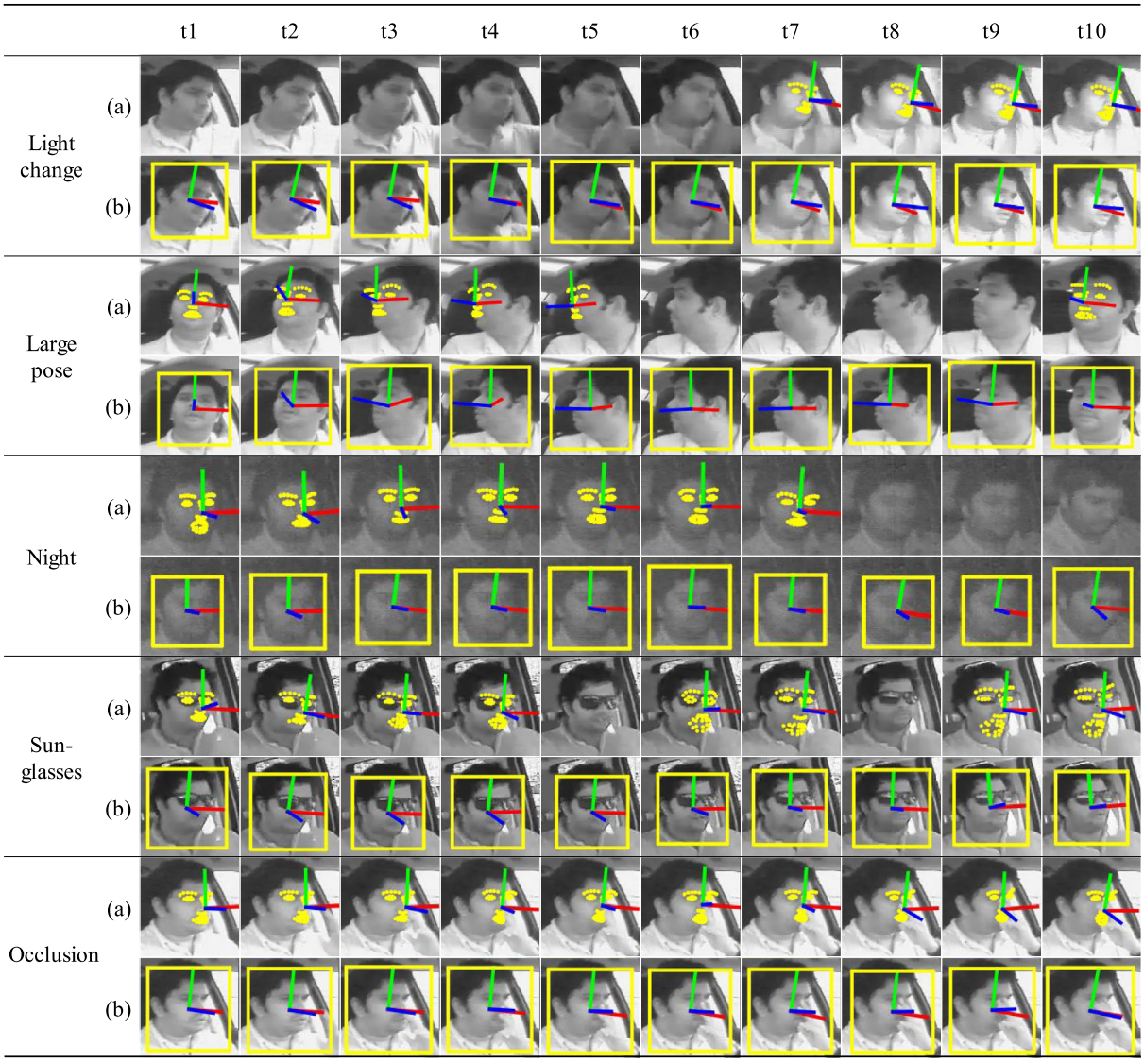


Fig. 12. Results of IntraFace [5] and the proposed method in various environments. (a) Results of Intraface (b) Our results. Here, t1–t10 indicates a sequence of input images.

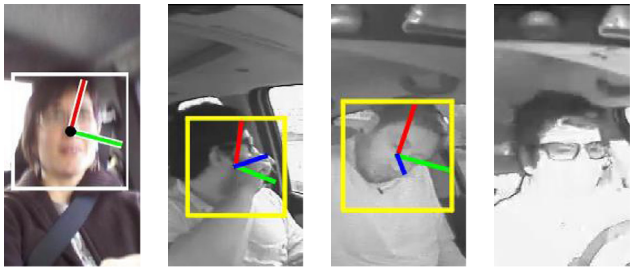


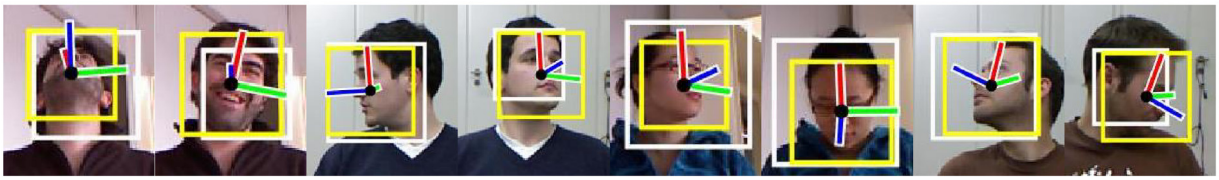
Fig. 13. Failure cases (a) Heavy blur (b) Large occlusion (c) Pose out of range (d) Oversaturation by direct sunlight.

4.4. Failure cases

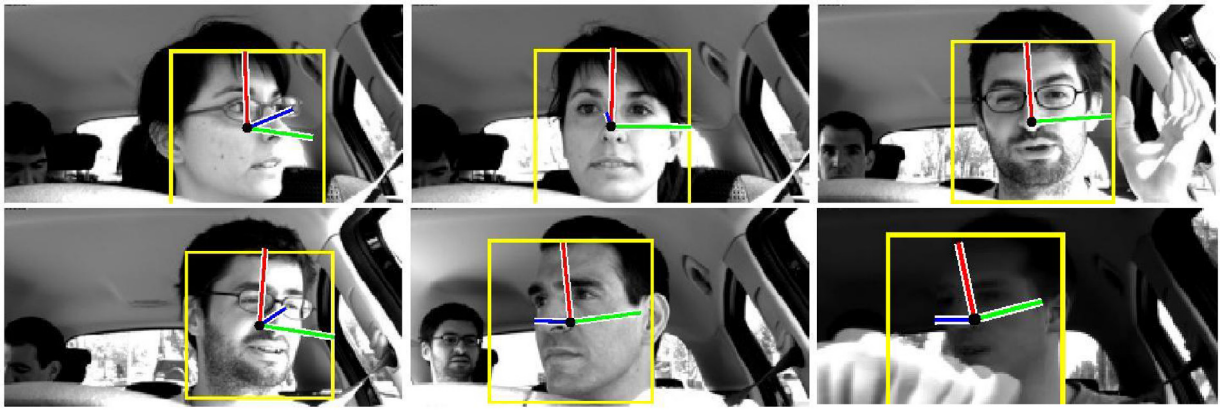
Even the multi-task DNN is trained from various kinds of datasets, they are not for the sake of specific issues such as occlusions, blurs, and light changes. The datasets are just for face detection and head pose estimation in ordinary conditions, so the proposed method cannot cover all environmental conditions. Fig. 13 shows several failure cases including failures to estimate head pose due to heavy blur from vibration (Fig. 13(a)), large occlusion (Fig. 13(b)), pose outside the range of training data (Fig. 13(c)) and a failure to detect a face due to oversaturation by direct sunlight (Fig. 13(d)). We believe that those limitations arise because training data do not contain enough of these cases. However, it is very difficult to capture all the conditions such as vibration, large occlusion, large pose, and direct sunlight while driving, so utilizing synthetic data with real images might address a number of failure cases.

5. Conclusion

In this paper, we proposed a multi-task learning-based real-time deep learning framework that can robustly estimate a driver's head pose using images obtained under poor conditions in various vehicle environments. We also introduced a method that trains multi-task learning DNN with individual datasets, even if there are no jointly annotated datasets. Compared with the single DNN-based learning method, the proposed multi-task learning-based system showed better accuracy without overfitting to specific data.

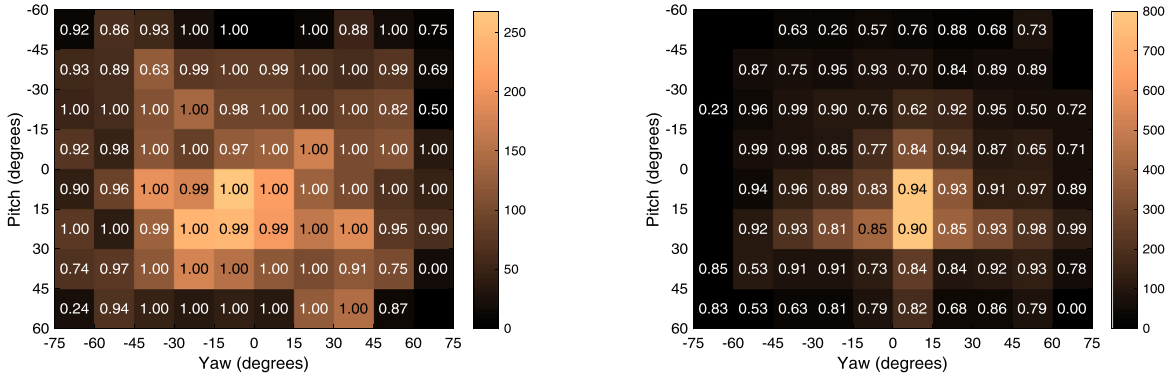


(a) Results on Biwi Kinect Head Pose dataset.



(b) Results on RS-DMV dataset.

Fig. 14. Some examples of head pose estimation results on BKHP and RS-DMV [52] datasets.



(a) Normalized success rates of ours.

(b) Normalized success rates of Fanelli et al.'s method.

Fig. 15. Comparison of normalized success rates of the proposed algorithm and Fanelli et al.'s method [22].

In addition, experimental results show that the proposed algorithm estimates more accurate drivers' head pose than the state-of-the-art technology even under harsh environmental conditions, such as noise, lack of light, large head pose changes, wearing sunglasses, and occlusion. In addition, we have demonstrated the superiority of our method by comparing with the state-of-the-art technologies of generative and discriminative approaches.

There are several ways that the proposed system and algorithm could be improved. In this work, we used only real image data for DNN training, but we will exploit a deep architecture using synthetic data to improve performance and make it easy to apply. Furthermore, by extending the proposed system, we will analyze not only a driver's head pose but also a pedestrian's head pose. A multi-sensor system that can determine whether both a driver and a pedestrian are aware of the vehicle will be developed and improve the vehicle's active safety system.

Acknowledgments

This research was supported by the Ministry of Trade, Industry and Energy and the Korea Evaluation Institute of Industrial Technology (KEIT) with the program number of "10060110".

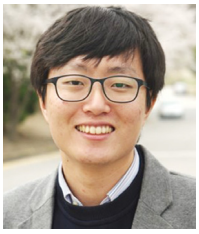
References

[1] Y. Hug, L. Chen, Y. Zhoug, H. Zhang, Estimating face pose by facial asymmetry and geometry, in: IEEE International Conference on Automatic Face and Gesture Recognition, FG, IEEE, 2004, pp. 651–656.
[2] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models-their training and application, Comput. Vis. Image Underst. 61 (1995) 38–59.
[3] T.F. Cootes, G. Edwards, C. Taylor, Active appearance models, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 681–685.
[4] P. Martins, J. Batista, Accurate single view model-based head pose estimation, in: IEEE International Conference on Automatic Face and Gesture Recognition, FG, IEEE, 2008, pp. 1–6.
[5] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2013, pp. 532–539.

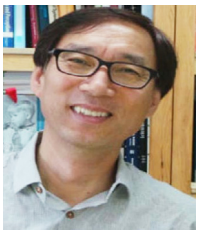
[6] F.D. la Torre, W.S. Chu, X. Xiong, F. Vicente, X. Ding, J. Cohn, Intraface, in: IEEE International Conference on Automatic Face and Gesture Recognition, FG, Vol. 1, 2015, pp. 1–8.
[7] A. Dopfer, H.-H. Wang, C.-C. Wang, 3d active appearance model alignment using intensity and range data, Robot. Auton. Syst. 62 (2) (2014) 168–176.
[8] A. Tawari, S. Martin, M.M. Trivedi, Continuous head movement estimator for driver assistance: issues, algorithms, and on-road evaluations, IEEE Trans. Intell. Transp. Syst. 15 (2) (2014) 818–830.
[9] A. Narayanan, R.M. Kaimal, K. Bijlani, Yaw estimation using cylindrical and ellipsoidal face models, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2308–2320.
[10] A. Narayanan, R.M. Kaimal, K. Bijlani, Estimation of driver head yaw angle using a generic geometric model, IEEE Trans. Intell. Transp. Syst. 17 (12) (2016) 3446–3460.
[11] X. Zhu, D. Ramanan, Face detection, pose estimation and landmark localization in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2012, pp. 2879–2886.
[12] F. Vicente, Z. Huang, X. Xiong, F.D. la Torre, W. Zhang, D. Levi, Driver gaze tracking and eyes off the road detection system, IEEE Trans. Intell. Transp. Syst. 16 (4) (2015) 2014–2027.
[13] V.N. Balasubramanian, J. Ye, S. Panchanathan, Biased manifold embedding: a framework for person-independent head pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2007.
[14] J. Foytik, V.K. Asari, A two-layer framework for piecewise linear manifold-based head pose estimation, Int. J. Comput. Vis. 101 (2013) 270–287.
[15] N. Grujić, S. Ilić, V. Lepetit, P. Fua, 3D facial pose estimation by image retrieval, in: IEEE International Conference on Automatic Face and Gesture Recognition, FG, IEEE, 2008.
[16] N. Gourier, D. Hall, J.L. Crowley, Estimating face orientation from robust detection of salient facial features, in: IEEE International Conference on Pattern Recognition, ICPR, IEEE, 2004.
[17] C. Huang, X. Ding, C. Fang, Head pose estimation based on random forests for multiclass classification, in: IEEE International Conference on Pattern Recognition, ICPR, IEEE, 2010, pp. 934–9376.
[18] C. BenAbdelkader, Robust head pose estimation using supervised manifold learning, in: European Conference on Computer Vision, ECCV, IEEE, 2010, pp. 518–531.
[19] H. Ji, R. Liu, F. Su, Z. Su, Y. Tian, Robust head pose estimation via convex regularized sparse regression, in: IEEE International Conference on Image Processing, ICIP, IEEE, 2011, pp. 3617–3620.
[20] M.D. Breitenstein, D. Kuettel, T. Weise, L. van Gool, Real-time face pose estimation from single range images, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2008.
[21] G. Fanelli, T. Weise, J. Gall, L.V. Gool, Real time head pose estimation from consumer depth cameras, in: IEEE International Conference on Pattern Recognition, ICPR, IEEE, 2011, pp. 101–110.
[22] G. Fanelli, M. Dantone, J. Gall, A. Fossati, L.V. Gool, Random forests for real time 3D face analysis, Int. J. Comput. Vis. 101 (2013) 437–458.
[23] Y. Lecun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012).
[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015.
[26] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.
[27] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2013, pp. 3476–3483.
[28] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 5325–5334.
[29] Z. Zhang, P. Luo, C.C. Loy, X. Tang, Learning deep representation for face alignment with auxiliary attributes, IEEE Trans. Pattern Anal. Mach. Intell. 38 (5) (2016) 918–930.
[30] B. Ahn, J. Park, I.S. Kweon, Real-time head orientation from a monocular camera using deep neural network, in: Asian Conference on Computer Vision, ACCV, 2014, pp. 82–96.
[31] R. Caruana, Multitask learning, Mach. Learn. (1997) 41–75.
[32] A. Vezhnevets, J.M. Buhmann, Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010, pp. 3249–3256.
[33] B. Romera-paredes, A. Argyriou, N. Bianchi-berthouze, M. Pontil, U.I. Centre, Exploiting unrelated tasks in multi-task learning, Adv. Neural Inf. Process. Syst. (2012).
[34] M. Lapin, B. Schiele, M. Hein, Scalable multitask representation learning for scene classification, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 1434–1441.
[35] A. Toshev, C. Szegedy, Deeppose: human pose estimation via deep neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2014.
[36] S. Li, Z.-Q. Liu, A.B. Chan, Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network, Int. J. Comput. Vis. 113 (1) (2015) 19–36.
[37] H. Jung, S. Lee, J. Yim, S. Park, J. Kim, Joint fine-tuning in deep neural networks for facial expression recognition, in: IEEE International Conference on Computer Vision Workshops, ICCVW, 2015, pp. 2983–2991.
[38] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: International Conference on Machine Learning, ICML, Omnipress, 2010, pp. 807–814.
[39] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 580–587.
[40] M. Koestinger, P. Wohlhart, P.M. Roth, H. Bischof, Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization, in: IEEE International Workshop on Benchmarking Facial Image Analysis Technologies, 2011.
[41] T. Weise, S. Bouaziz, H. Li, M. Pauly, Realtime performance-based facial animation, 30 (4) (2011).
[42] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Tech. Rep. 07–49, University of Massachusetts, Amherst, 2007.
[43] P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2930–2940.
[44] V. Le, J. Brandt, Z. Lin, L. Bourdev, T.S. Huang, Interactive facial feature localization, in: European Conference on Computer Vision, ECCV, 2012, pp. 679–692.
[45] X.P. Burgos-Artizzu, P. Perona, P. Dollár, Robust face landmark estimation under occlusion, in: IEEE International Conference on Computer Vision Workshops, ICCVW, 2013, pp. 1513–1520.
[46] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2010, pp. 94–101.
[47] P. Viola, M. Jones, Robust real-time object detection, Int. J. Comput. Vis. (2001).
[48] J. Paone, D. Bolme, R. Ferrell, D. Aykac, T. Karnowski, Baseline face detection, head pose estimation, and coarse direction detection for facial data in the SHRP2 naturalistic driving study, in: IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 174–179.
[49] S. Yang, P. Luo, C.C. Loy, X. Tang, Wider face: A face detection benchmark, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016.
[50] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision, ICCV, 2015.
[51] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, R. Horaud, Head pose estimation via probabilistic high-dimensional regression, in: IEEE International Conference on Image Processing, ICIP, 2015, pp. 4624–4628.
[52] J. Nuevo, L.M. Bergasa, P. Jiménez, RSMAT: Robust simultaneous modeling and tracking, Pattern Recognit. Lett. 31 (2010) 2455–2463.

Byungtae Ahn received a B.S degree in Electronic Engineering from Kumoh national Institute of Technology, Korea, in 2007, and a M.S. degree in Bio-mechatronics from Sungkyunkwan University, Korea, in 2011. He is currently working toward the Ph.D. degree in Robotics Program at KAIST. He received a Qualcomm Innovation Award in 2013, and has been listed in Marquis Who's Who in the World, 2016. His research interests include deep learning, Human–Robot Interaction (HRI), and Advanced Driver Assistance System (ADAS). He is a student member of the IEEE.

Dong-Geol Choi received the B.S and M.S degree in Electric Engineering and Computer Science from Hanyang University in 2005 and 2007, respectively, and the Ph.D degrees in the Robotics Program from KAIST in 2016. He is currently a post-doctoral researcher at the Information & Electronics Research Institute, in KAIST. His research interests include sensor fusion, autonomous robotics, and artificial intelligence issues. Dr. Choi received a fellowship award from Qualcomm Korea R&D Center in 2013. He was a member of 'Team KIAST,' which won the first place in DARPA Robotics Challenge Finals 2015. He is a member of the IEEE.



Jaesik Park received his Bachelor degree (Summa cum laude) in media communication engineering from Hanyang University in 2009. He received his Master degree and Ph.D. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2011 and 2015, respectively. He joined Intel Labs as a research scientist in 2015. His research interests include depth map refinement, image-based 3D reconstruction. He is a member of the IEEE.



In So Kweon received the B.S. and M.S. degrees in Mechanical Design and Production Engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the Ph.D. degree in Robotics from the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1990. He worked for the Toshiba R&D Center, Japan, and joined the Department of Automation and Design Engineering, KAIST, Seoul, Korea, in 1992, where he is now a professor with the Department of Electrical Engineering. His research interests are sensor fusion, color modeling and analysis, visual tracking, and visual SLAM. He was the general chair for the Asian Conference on Computer Vision 2012 and he is on the honorary board of the International Journal of Computer Vision (IJCV). He has been serving as a director for the Personal Plug and Play DigiCar Center which is one of the National Core Research Center since 2010. He was a member of 'Team KAIST' which won the first place in DARPA Robotics Challenge Finals 2015. He is a member of the IEEE.