This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

Head and Body Orientation Estimation Using Convolutional Random Projection Forests

Donghoon Lee, Ming-Hsuan Yang, and Songhwai Oh*

Abstract—In this paper, we consider the problem of estimating the head pose and body orientation of a person from a low-resolution image. Under this setting, it is difficult to reliably extract facial features or detect body parts. We propose a convolutional random projection forest (CRPforest) algorithm for these tasks. A convolutional random projection network (CRPnet) is used at each node of the forest. It maps an input image to a high-dimensional feature space using a rich filter bank. The filter bank is designed to generate sparse responses so that they can be efficiently computed by compressive sensing. A sparse random projection matrix can capture most essential information contained in the filter bank without using all the filters in it. Therefore, the CRPnet is fast, e.g., it requires $0.04ms$ to process an image of 50×50 pixels, due to the small number of convolutions (e.g., 0.01% of a layer of a neural network) at the expense of less than 2% accuracy. The overall forest estimates head and body pose well on benchmark datasets, e.g., over 98% on the HIIT dataset, while requiring at $3.8ms$ without using a GPU. Extensive experiments on challenging datasets show that the proposed algorithm performs favorably against the state-of-the-art methods in low-resolution images with noise, occlusion, and motion blur.

Index Terms—Head pose estimation, body orientation estimation, random forests, convolutional neural network, compressive sensing

1 Introduction

Head and body orientations are important visual cues of a person, which are closely related to a number of applications such as surveillance, social signal processing, and human-computer interaction. In a surveillance system, eye gaze plays an important role in the inference of visual focus and attention [1]. The gaze and body posture can be combined to estimate social signals, e.g., aggressiveness or disagreement [2], and to control robots or smart devices [3].

In recent years there has been a growing interest in vision applications for autonomous driving, where an important component is the detection of pedestrians. In addition, it is critical to infer their moving directions and whether they are aware of the traffic conditions. Such tasks can be aided by estimating eye gazes and body orientations of pedestrians. For example, Figure 1 shows an image from the KITTI dataset acquired by a vehicle on the road. Based on the body orientation of person B, it can be inferred that she intends to cross the road, but recognizes a car and stops. On the other hand, person A is about to cross the road without knowing that a vehicle is approaching. In this case, it is of great interest to develop a system that understands the scene and the potential danger based on head and body poses. This is a challenging problem as it involves the development of a system with high precision and real-time performance. Furthermore, the size of pedestrians may be small which makes the problem more complicated.

In this paper, we propose an efficient algorithm for estimating head poses and body orientations of pedestrians at a distance. We estimate the head pose of a person instead of the exact gaze due to several reasons. First, estimating exact gaze is only possible with face images in near frontal pose when the pupils are visible. It is not feasible for practical scenarios since pedestrians in all directions should be considered. Second, existing methods operate on high-resolution face images in close-up views to infer the gaze. However, the proposed algorithm aims to infer visual cues of pedestrians at a distance to consider the high speed of vehicle. Finally, head pose and eye gaze are highly correlated in terms of visual attention.

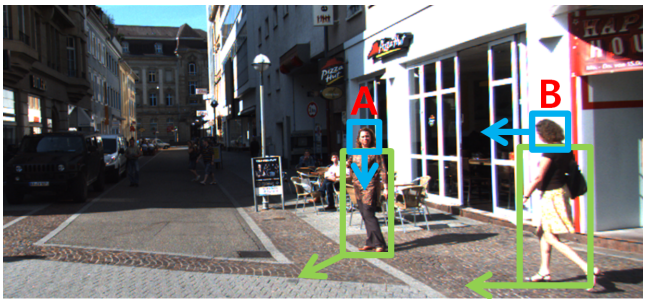


Fig. 1. A sample image acquired from a vehicle. Head poses and body orientations are important cues to predict pedestrian movements. For intelligent vehicles, an estimation algorithm needs to be fast, accurate, and robust to low-resolution images degraded by motion blur and noise.

As low-resolution images are considered in this work, (e.g., 50×50 pixels or smaller for a head region), it is more difficult to estimate orientations using conventional methods. The problem is complicated since useful facial features such as eyes cannot be reliably extracted from low-resolution images. A wide range of variations in skin color, glasses, hair style, and head shape exacerbate the problem [4]. Estimating body orientation is also a challenging

- *D. Lee is with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826 Korea. E-mail: donghoon.lee@cpslab.snu.ac.kr*
- *M.-H. Yang is with the School of Engineering, University of California, Merced, CA 95344 USA. E-mail: mhyang@ucmerced.edu*
- *S. Oh is with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826 Korea. E-mail: songhwai@snu.ac.kr. *S. Oh is the corresponding author of this work.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

problem due to the articulated pose, different clothing, and partial occlusion.

The aforementioned challenging issues are addressed by exploiting the expressive representation of convolutional compressive features and effective estimation of the convolutional random projection forest in the *CRPforest* algorithm. The convolutional compressive features describe an input image by compressing responses of convolutional filters. To generate effective and diverse responses, a network is constructed to learn a rich filter bank that contains multi-channel and multi-scale filters. We insert a layer in the network to handle high-dimensional features from the filter bank. The operation of this layer is based on the compressive sensing which performs compression and sensing at the same time. Thus, the compressed signal can be obtained without computing all responses from the filter bank by using a sparse random projection matrix. As such, it is possible to extract, compress, and classify convolutional filter responses using a single network, which is referred to as the *CRPnet*.

The convolutional random projection forest is based on the random forest [5] and CRPnet. We train a CRPnet as a split function of each node and choose the best random projection matrix based on the impurity measure (e.g., information gain). Consequently, the whole forest is more discriminative as the CRPnet is based on the generative framework of compressive sensing. We use a sparse form of a random projection matrix which induces low generalization errors by strengthening each tree and weakening the correlation between trees [6]. In contrast to the prior work [6], the CRPforest learns more discriminative filters than that using fixed box filters.

Experiments on four challenging benchmark datasets are carried out to evaluate the proposed algorithm against the state-of-the-art methods for head and body pose estimation. The proposed algorithm achieves leading estimation results for all datasets, e.g., over 98% classification accuracy on the HIIT dataset, while each image is processed within a few milliseconds without using a GPU. Furthermore, the proposed approach performs well against other algorithms on low resolution images and degraded images with noise, occlusion, and blurring. We also demonstrate that the proposed CRPforest is more accurate and robust than alternative approaches.

2 Related Work

We discuss the related methods on head pose and body orientation estimation based on templates, detector arrays, regression, manifold embedding, and mid-level visual elements.

Template-based methods. Given an input image, a template method matches against to a set of exemplars with corresponding pose labels for estimation. Orozco et al. construct the template of each class based on the corresponding mean image [7]. The Kullback-Leibler divergence between every pixel of the input image and templates is computed. The similarity map is a feature descriptor and classified by a support vector machine (SVM). Other metrics such as Euclidean, Bhattacharyya, and Mahalanobis distance are also evaluated. However, the pairwise distance between images of the same person in different pose is usually smaller than the distance of different persons in the same pose [8]. Therefore, the estimation accuracy of template-based methods is limited.

Detector-based methods. As more accurate detectors have been developed in the last decade, numerous methods that estimate orientations by training multiple detectors for different poses have been proposed. Detectors in the literature are typically based on the combination of features such as histogram of oriented gradients (HOG) and Haar-like features [9], [10], [11], and classifiers such as SVMs and Adaboost algorithms. These approaches perform well for discrete and coarse estimation of head and body poses. However, it is difficult to resolve the case when two or more detectors predict different poses for the same input image. In addition, training of multiple classifiers for a dense orientation estimation is not straightforward because of unbalanced positive and negative training samples.

Regression-based methods. Orientation estimation can be considered as a regression problem that maps high-dimensional features from an input image to low-dimensional pose parameters [12]. In [13], [14], a random regression forest is used to learn a mapping from depth features to the corresponding head and body orientation. At each node of the forest, the depths of two randomly chosen points are compared. Based on the comparison, each sample is split to the left or right child nodes. Similarly, a method that uses dense SIFT descriptors from input images rather than depth features is developed [15]. These methods require depth images or high-resolution images since SIFT features need to be reliably extracted. In addition, hand-crafted features may not be sufficiently discriminative since, with these features, two images of the same person with different poses may be mapped closer than images of different people with the same pose [4].

Manifold-based methods. Numerous algorithms assume that high-dimensional images (observations) can be modeled well by the corresponding points on a low-dimensional manifold. Thus, the distance on the manifold between two points can be used for the pose estimation. In [16], a weighted array of descriptors computed from overlapping patches is used for head and body pose estimation, where each is described by a covariance matrix of image features. However, this method is computationally expensive and sensitive to distortions, noise, or occlusions since the holistic representation of the image is adopted [17].

Mid-level visual elements. Numerous mid-level elements or patches have been used as representations for vision problems. Typically, a large number of patches are extracted by a random cropping or selective search and clusters are formed based on HOG descriptors or CNN features. Mid-level representations have been shown to be effective for detection [18] and image classification [19], [20], [21]. While mid-level patches are localized in an unsupervised way in general, in this work we determine important patches and learn corresponding convolutional filters based on supervisory signals.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

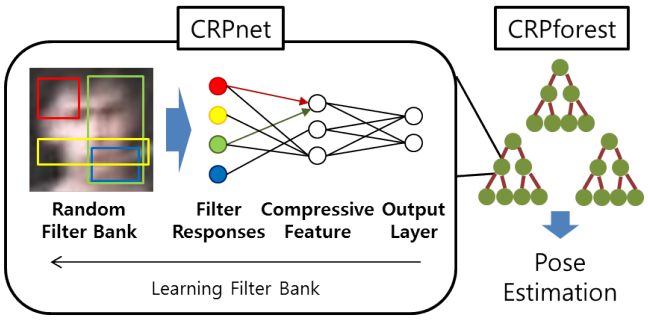


Fig. 2. Proposed CRPforest algorithm. It is equipped with a CRPnet at each node of a tree as a split function. The network is capable of learning discriminative filters at each rectangular region.

3 Algorithmic Overview

The proposed algorithm is based on the CRPnet which learns an effective filter bank, compressive features, and corresponding classifiers as shown in Figure 2. We note that discriminative features (e.g., HOG or SIFT) are not used in this work in order to deal with low-resolution images and reduce computational cost. Instead, we use a rich filter bank that captures information from all possible rectangular regions inside an image. Although responses of the filter bank are in a high-dimensional feature space, we can handle them efficiently based on compressive sensing which compresses signals without losing essential information. The compression is expressed by combinations of a few random filter responses as explained in Section 4.

Figure 2 shows an illustrative example in which four regions are marked in different colors. For each region, a filter that has the same size as the region is convolved and generates a response. Then, the responses are linearly combined to obtain a compressive feature. Finally, the compressive features are fed to a fully-connected layer. As such, the whole network can be trained using an error backpropagation algorithm. Section 5 describes the configurations to learn effective filters using the CRPnet.

The ensuing issue is how to select the regions to convolve, such as the four boxes in the above example. It is desirable to select informative regions in input images. However, the number of possible regions and combinations are too large to perform an exhaustive search. To cope with this problem, the CRPforest is proposed in Section 6. It trains the CRPnet at each node based on random regions and hierarchically selects the most effective setting. It may seem straightforward to combine the network and the random forests since the CRPnet is a randomized algorithm. However, there is a risk of overfitting the CRPnet due to a decrease in the number of training samples after a few splits. To address this problem, we propose a method to train the CRPforest based on stochastic splits. Extensive experiments in Section 7 show that the CRPforest is more effective and robust compared to the CRPnet.

4 Rich Filter Bank

It has been shown that the object classification algorithm based on the features learned from a convolutional neural network (CNN) outperforms the state-of-the-art methods based on hand-crafted features [22]. In this work, we propose an algorithm to learn a rich filter bank which contains a large number of rectangular filters. A rectangular filter $F_{w,h} \in \mathbb{R}^{wh}$ is characterized by its shape, i.e., width w, height h, and values of elements. Since each channel of a filter is applied to the corresponding channel of an input image, we omit the notation for a channel in this paper. The proposed filter bank contains all possible combinations of filter shapes. In other words, for a $\underline{w} \times \underline{h}$ input image, the value at (x, y) of a filter, $F_{w,h}(x, y)$, is a real number where w and h represent all possible widths and heights of the filter, i.e., $1 \leq w \leq \underline{w}$ and $1 \leq h \leq \underline{h}$. As a result, there are $\mathrm{O}((\underline{wh})^2)$ of different filter sizes in the filter bank. For a 100×100 pixels image, there are about 10^8 possible filter sizes. We use compressive sensing to deal with the computational issues.

4.1 Compressed Filter Bank

Within the compressive sensing framework, an original data point x is compressed as follows:

$$y = Ax, \tag{1}$$

where A is an $m \times n$ random projection matrix with $m \ll n$ and y is a compressed signal. It is shown that y captures most essential information contained in the x when A satisfies the restricted isometry property (RIP) condition [23]. To compress the feature space of the filter bank, we first choose an adequate matrix A.

The sparse random projection [24] in (2) is one of the most efficient forms for the matrix A and the element is described by

$$a_{ij} = \sqrt{s} \times \begin{cases} 1, & \text{with probability } \frac{1}{2s}, \\ 0, & \text{with probability } 1 - \frac{1}{s}, \\ -1, & \text{with probability } \frac{1}{2s}, \end{cases} \tag{2}$$

where $s \in \mathrm{o}(n)$[1] and $A = [a_{ij}]$. By setting $s = n/\log(n) \in \mathrm{o}(n)$, the expected number of nonzero elements per row of the matrix A is $\log(n)$. This enables us to preserve the essential information of the filter bank by computing only a few filter responses. The random matrix A is computed once off-line and remains fixed while testing a new image. As a result, an element of the compressed vector, y, is a linear combination of randomly chosen rectangular filter responses.

The remaining issue is to show that filter responses are sparse. Most computer vision tasks that apply compressive sensing rely on the fact that an image can be represented by sparse coefficients in the wavelet domain [23]. However, filter responses of an image are not necessarily a sparse signal. Therefore, we enforce the sparsity by applying a rectified linear unit (ReLU) to filter responses.

1. The little-o notation. $f(n) \in \mathrm{o}(g(n))$ as $n \to \infty$ means that for every positive constant ϵ there exists a constant N such that $|f(n)| \leq \epsilon |g(n)|$ for all $n \geq N$ [25].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

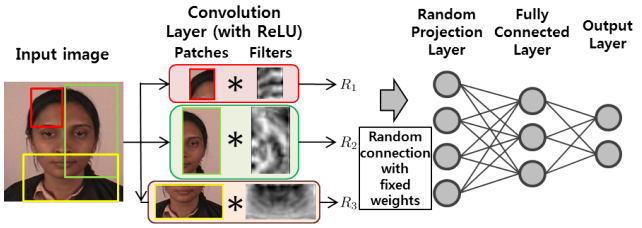


Fig. 3. Proposed CRPnet algorithm for head and body orientation estimation. Three filters and a network with a small number of nodes are shown as an illustrative example.

4.2 Box Filter Bank

A filter bank that uses box filters [6] is a special case of the proposed algorithm. A box filter computes an average of pixel values inside a rectangular region as follows:

$$F_{w,h}(x,y) = \frac{1}{wh} \times \begin{cases} 1, & \text{if } 1 \le x \le w, 1 \le y \le h, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where w and h represent the width and height of the rectangular region. In this case, the filter simply reduces the resolution of the input image which makes the resulting image still sparse in the wavelet domain. Therefore, we apply the random projection matrix in (2) without ReLU step. The box filter resembles to the generalized Haar-like features [26]. In order to compensate limited filter shapes, an input image is decomposed into several channels including different color spaces and magnitudes as well as orientations of gradients.

Although box filters may be less discriminative or robust than learned filters, they are useful when the number of training samples is not sufficient for training a rich filter bank. In the next section, we discuss how to train filters using a neural network.

5 Convolutional Random Projection Net

The convolutional random projection network is proposed to learn a rich filter bank. The structure of the network is based on a directed acyclic graph as shown in Figure 3. It consists of an input layer, a convolution layer, a ReLU layer, a random projection layer, a fully-connected layer, and an output layer. Two major differences between the proposed neural network and a typical CNN are filter locations and the random projection layer.

5.1 Input Layer

For each training image, we first apply mean image subtraction and contrast normalization. We then augment training data for learning the proposed network. The augmentation is important for learning an effective network since the number of image samples in the existing head pose or body orientation databases is much smaller than that for image recognition such as the ImageNet [27] or MSCOCO [28] datasets.

For each mini batch, we apply five augmentations: flipping, rotating, cropping, and adding noise and blurs. The random left-right flip is performed with probability of 0.5 and the corresponding labels are also flipped. We perform 2D rotation of an image with a random angle between -15 degrees and $+15$ degrees based on the center of the image, and empty pixels after the rotation are filled with zero. For random cropping, a rectangular region inside the image is cropped and resized to the original size. The width and height of the rectangle are randomly chosen between 90% and 100% of the width and height of the input image, respectively. The augmentations enrich training data by applying translation and rotation offsets. In addition, noise and blurs are added to make the network more robust to degraded images. We apply the mean zero Gaussian noise at each pixel and the variance is randomly chosen between zero and the 30% of the maximum value of pixels. Finally, an image is blurred by resizing it to a smaller size and then restoring it to its original size using a bilinear interpolation method. We randomly choose the scale of the resized image between 20% and 100% of the original size.

5.2 Convolutional and ReLU Layers

In a typical convolution layer, a filter is applied extensively at image locations, which is computationally expensive. In the proposed algorithm, the filter responses are efficiently computed through the compressive sensing with a sparse random projection matrix in (2). We note that the proposed algorithm convolves multi-scale filters into random locations, while a CNN uses a single-scale filter at all locations.

Figure 4 shows examples of trained filters at random rectangular regions based on the HIIT dataset. We plot mean images of all head orientation classes and show the learned filter on each of them. The filters are trained with different sizes and locations and encode different visual information from face images in various poses. Small filters covering different regions such as eye-nose, nose-cheek, and forehead-hair, resemble the generalized Haar-like features. Some filters learn the shape of a facial feature, for example, eye, nose, mouth, and chin. For larger filters, the general shapes or appearances of face images are learned. Although the CRPnet is not a deep network, the learned filters are fairly diverse and informative. Figure 5 shows that these learned filters are discriminative. The responses are based on the filters in the red box in Figure 4. The shape of this filter is reminiscent of a front-left face. Consequently, the filter is particularly sensitive to images in front-left and left classes. On the other hand, it generates small responses, mostly zero, for most of the right and front-right classes.

As discussed above, the ReLU layer is required to make a sparse signal. In average, more than 50% of the trained filter responses are zero as shown in Figure 5.

5.3 Random Projection Layer

The random projection operator behaves similarly to the pooling operator in conventional CNNs. While both layers have the same purpose as reducing the feature space without significant loss of information, there are three notable differences. First, a pooling layer compresses the input data based on a sliding window. It requires an entire scan of the feature space which is computationally expensive. This property also leads to the second difference that the input of a pooling layer should be a dense feature map. In order to obtain a dense feature map, a dense convolution needs to be used, which is the computational bottleneck as discussed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

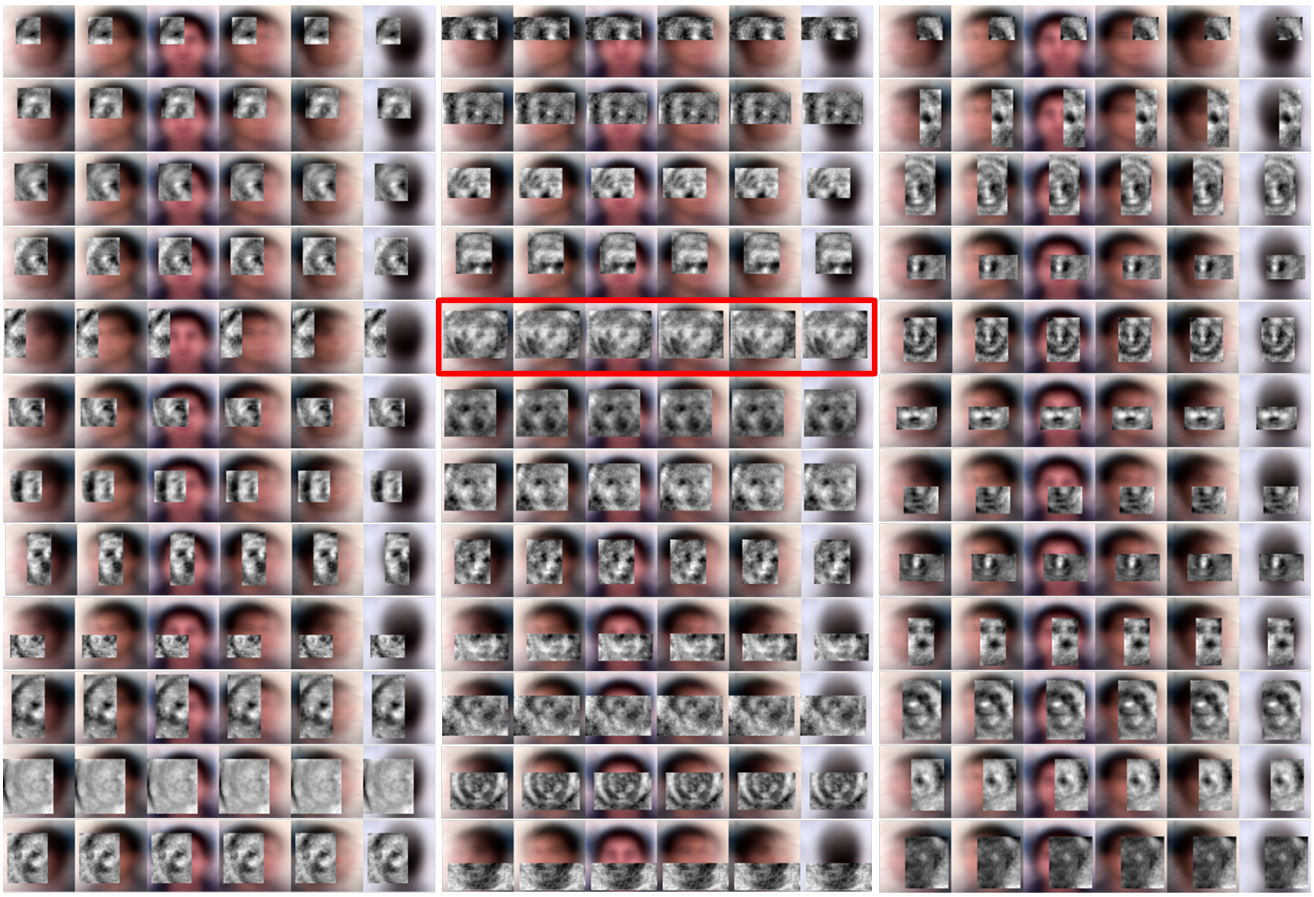


Fig. 4. Examples of trained filters on the HIIT dataset. Each filter is displayed repeatedly in the average image of each head orientation class to help understand its role for classification. The order of orientation classes is left, front-left, front, front-right, right, and rear. The filter in the red box, which resembles a face looking at front left, is used in Figure 5 to compare responses of each orientation class as an example.

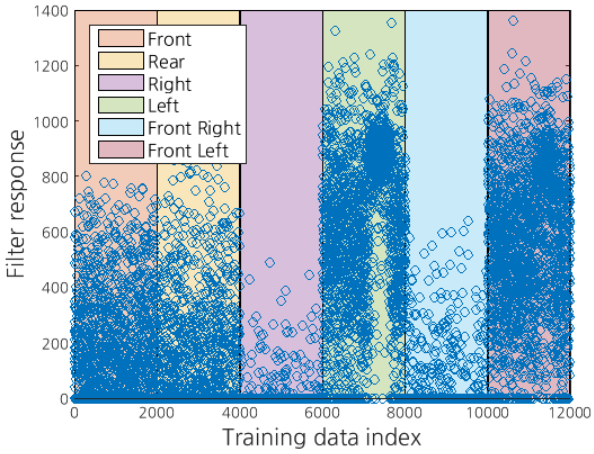


Fig. 5. Responses of a filter on the HIIT test dataset. The filter used in this experiment is the one in the red box of Figure 4. It shows that this filter is able to distinguish most of the right and front-right orientations from other classes.

above. Third, the input feature map of a pooling layer contains single-scale filter responses.

The proposed random projection layer operates as the random projection matrix in (2) that satisfies the RIP condition of compressive sensing and preserves essential information [24]. It randomly selects some responses from the ReLU layer and applies linear combinations. For example, let R_i be the output of the i-th filter from the ReLU layer as shown in Figure 3, and the first row of (2) has nonzero values $\sqrt{s}, \sqrt{s}, -\sqrt{s}$ at $1, 4, 7$-th elements. Then, the output of the first node of the random projection layer becomes $\sqrt{s}R_1 + \sqrt{s}R_4 - \sqrt{s}R_7$. Note that the learning rate of this layer is zero, since elements of the matrix must remain fixed after the random initialization.

5.4 Fully-Connected and Output Layers

In order to regularize the network, we apply the batch normalization. The loss function used in this paper is the multi-class structured hinge loss, i.e., the Crammer-Singer loss [29], as follows:

$$L(X, c) = \max(0, 1 - M(c)), M(c) = X(c) - \max_{q \neq c} X(q), \quad (4)$$

where X is the prediction score and c is the label. It is used for the fair comparison with [6] while the conventional softmax loss gives similar results in our preliminary experiments.

An example of a training curve of the network is shown in Figure 6. This experiment is based on the HIIT training dataset where 20% are randomly selected as the validation set. The error and objective value decrease smoothly and the gap between the training and validation curve is small.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

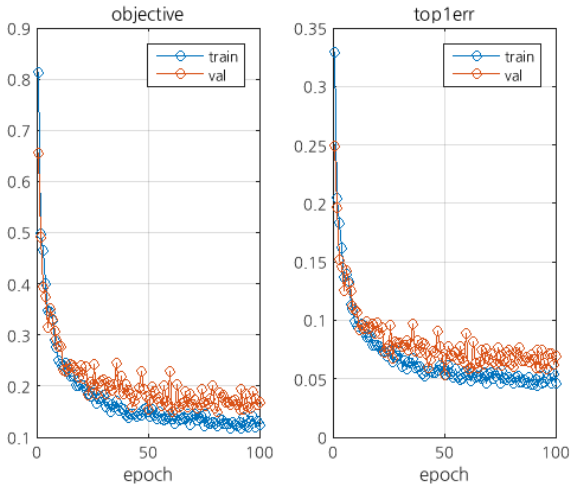


Fig. 6. An example of the objective and error curves of the CRPnet.

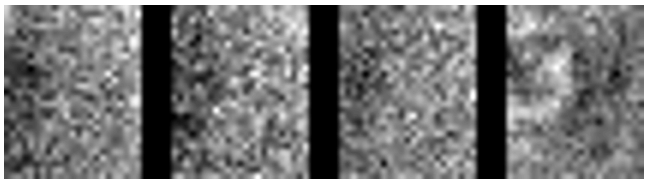


Fig. 7. Examples of poorly trained filters after a few deterministic splits in a tree.

Both Figure 4 and Figure 6 demonstrate that the proposed network is trained properly.

While the proposed network captures the essential information of an image, it relies on a single random projection. In order to increase the generalizability of the network, we propose an algorithm, the convolutional random projection forest, that hierarchically selects random projection matrices which maximize the impurity measure.

6 Convolutional Random Projection Forest

A random forest $\mathcal{F}$ is an ensemble of randomized decision trees. Trees are grown independently using a split function at each non-leaf node. Each split aims to maximize the impurity measure such as information gain between the parent and child nodes. A node stops growing and becomes a leaf node when it satisfies one or more pre-defined conditions. In this work, we set the conditions using the maximum depth, amount of impurity gain, and number of samples in a node. For pose estimation, the distribution of training data stored in leaf nodes is used.

The proposed CRPforest is equipped with a CRPnet at each node. The CRPnet operates as a weak classifier trained using the objective in (4). One of the most important issues for combining neural networks and tree-based algorithms is to maintain a sufficient number of training data for proper learning. It is a challenging problem since the number of samples in each node decreases by the hierarchical splits. Therefore, after only a few splits, a node may lack of training data even if the number of training samples at the root node is large. Without enough data, the network can be easily overfitted. In such cases, trained filters become noisy and less meaningful as shown in Figure 7.

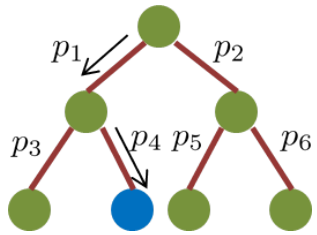


Fig. 8. Illustration of a tree that has a stochastic split function. The sum of probabilities to reach child nodes is one, e.g., $p_1 + p_2 = 1$.

To address this problem, we use a stochastic split [30] instead of the deterministic split. A stochastic split computes the probability that each sample reach a node rather than the actual split of the data. The probability of reaching a node is a product of probabilities at every split along the path from the root node. For example, the probability that reaching the blue node in Figure 8 is $p_1 p_4$.

During the training phase, the reaching probability is used as a weight of each training sample to learn the network at a node:

$$w_i^n = \prod_{j \in E_n} p_j(i) \tag{5}$$

where w_i^n is the weight of an i-th training sample to learn the network at node n, $p_j(i)$ is the probability of the i-th sample passes an edge j, and E_n is a set of edges from the root node to the node n. For the root node, the reaching probability is set to one. The trained network is used as the split function of the node. Based on the network, we compute the probability that each sample will reach a child node.

For each test image, each tree makes a decision by

$$\mathrm{P}_t(c|I) \propto \sum_{m \in L} \sum_{l(i)=c} w_i^m, \tag{6}$$

where $\mathrm{P}_t(c|I)$ is a probability that tree t classifies input image I as a class c, L is a set of leaf node indices, and $l(i)$ is the label of the training sample i. Next, all decisions are merged by

$$\mathrm{P}_{\mathcal{F}}(c|I) = \frac{1}{T} \sum_{t=1}^{T} \mathrm{P}_t(c|I), \tag{7}$$

where $\mathrm{P}_{\mathcal{F}}(c|I)$ is the final estimation.

The combination of a CRPnet and a forest is efficient and effective for the following reasons. First, the proposed network is based on a small number of random regions and cheap calculations, i.e., linear combinations and ReLUs. This allows each non-leaf node as simple as a weak classifier. Second, a network can concentrate on highly weighted samples and learn better. Finally, applying compressive sensing to each node helps reduce the generalization error of the random forest [6].

7 Experimental Results

We evaluate the proposed head and body pose estimation algorithm against the state-of-the-art methods using

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

(a) HIIT dataset

(b) QMUL dataset and background images

(c) HOC dataset

(d) CMU Multi-PIE dataset

(e) FacePix dataset

Fig. 9. Sample images of the evaluation datasets.

numerous benchmark datasets with images of coarse and fine pose information. We carry out head pose classification experiments on the HIIT [16], QMUL, and QMUL with background datasets [7]. For body orientation classification, we use the HOC dataset [16]. In addition, we use the CMU-MultiPIE [31] and FacePix [32] datasets for head pose regression evaluations. Figure 9 shows sample images from the evaluated benchmark datasets. The source code and datasets will be made available to the public.

7.1 Evaluation Datasets

The HIIT dataset contains 24,000 images with 6 head poses in relatively static backgrounds and few occlusions. This dataset is challenging because images are acquired in a wide range of lighting conditions with various facial expressions, as shown in Figure 9(a). Furthermore, it consists of images from different datasets (e.g., the QMUL [7] and CMU Multi-PIE [31]) with large variations in appearance.

The QMUL dataset contains 15,660 images with 4 head poses acquired in different illuminations with occlusions. The images are collected using a head detector with a significant amount of motion blurs, misalignments, and occlusions. It is challenging since subjects often wore caps or sunglasses. This dataset additionally contains 3,099 background images and we refer to the entire dataset as the QMULB database in this paper. The background images

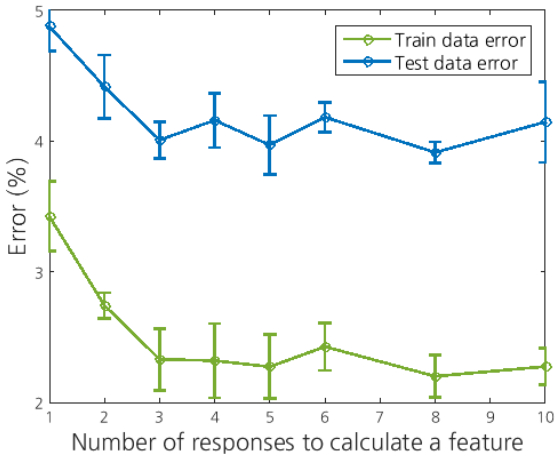


Fig. 10. Accuracy of the CRPnet on the HIIT dataset with different compression layer setting. For this experiment, we use 100 filters and the compressed dimension is fixed to 50. The variance of the result is obtained after ten independent runs.

vary from simple floors to complex scenes as shown in Figure 9(b).

The HOC dataset is derived from the ETHZ database [33] which contains pedestrian images in urban scenes. There are 8,555 images of 132×62 pixels with four classes of body pose. As shown in Figure 9(c), these images contain large variations in appearance caused by clothing, articulated poses, occlusions, different scales, motion blurs, and accessories.

The images of the the CMU-MultiPIE database are collected from 337 subjects with different poses from $-90°$ to $90°$ with $15°$ intervals and 13 yaw directions. For the experiments, we use all images of 6 expressions under the frontal light sources. Existing head pose estimation methods use aligned images based on manually annotated facial features of this dataset. In this work, we consider more realistic scenarios. We crop 360×360 center pixels of the head images and downsample it to 50×50 pixels. The cropped images are not aligned, which are closer to real world applications. We use images of 50% of randomly selected subjects for training and the others for tests. This dataset is challenging since the images are acquired from a large number of subjects with different expressions.

The FacePix dataset contains 30 subjects and 181 images for each person (one image per yaw degree from $-90°$ to $90°$). There are a total of 5,430 images of aligned head with static backgrounds. We perform the leave-one-out evaluations on this dataset. The dataset is challenging because of fine intervals in the yaw orientation.

7.2 CRPnet Characteristics

There are two parameters that specify the CRPnet structure: number of filters and random projection matrix in (2). We analyze the effect of each parameter in this section.

The random projection matrix in (2) maps filter responses from $\mathbb{R}^n$ to $\mathbb{R}^m$ where n is the number of all possible filters and m is the dimension of the compressive feature. Although n is large, the entire matrix can be stored efficiently in memory by the virtue of sparsity. When m

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

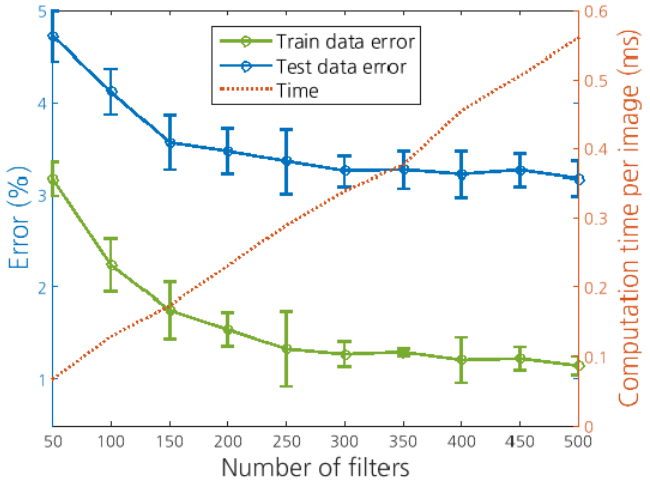


Fig. 11. Accuracy of the CRPnet on the HIIT dataset with different number of filters. We choose two random filters to compute the convolutional compressive feature. The variance of the result is obtained from ten executions, and the execution time is measured based on a single CPU core.

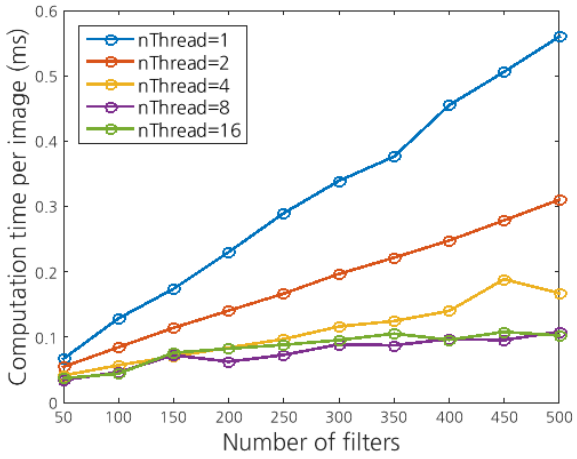


Fig. 12. Computation time of the CRPnet with different number of filters. The dimension of compressive feature is set to one half of the number of filters. We compute the average of 1,000,000 runs for each result.

is too small or the matrix is too sparse, the RIP condition does not hold. On the other hand, the computational cost is increased significantly when a large feature space (large m) or a dense matrix is used. Figure 10 shows the effect of the random projection matrix. In the experiments, we use 100 filters and m is fixed to 50. Overall, the CRPnet performs well with a sparse random projection matrix.

The number of filters is the same as the number of convolutions in the CRPnet. Thus, it directly affects the discriminative strength and computational complexity of the network. Figure 11 shows the effect of the number of filters. By using more filters, the CRPnet performs more accurately at the expense of computational cost. Note that, unlike conventional CNNs, the number of convolutions is independent of the size of an input image. When we use the same number of filters, the computational cost of the CRPnet is reduced significantly compared to conventional CNNs. For example, given an image of 227×227 pixels and 200 different filters of 11×11 pixels, the CRPnet performs 200 convolutions while a CNN performs $(227-11+1)\times(227-11+1)\times 200 = 9,417,800$ convolutions at a single convolution layer. If we use a stride of 4 pixels, there are still 605,000 convolutions. Therefore, the number of convolutions in the CRPnet is only 0.002% to 0.03% of the convolution layer in the CNN depending on the size of the stride. The run time shown in Figure 11 and Figure 12 is measured without a GPU demonstrating the efficiency of the CRPnet. The CRPnet takes less than $0.6ms$ to compute the forward pass even when we use a single CPU core and a large number of filters. Furthermore, the accuracy of the proposed method does not significantly change when more filters are used. However, the computational cost increases linearly as the number of filters increases. This is due to the number of convolutions is proportional to the number of filters and no sliding-window based scheme is involved. To maintain low computational complexity, we use 100 filters for the CRPforest in this work.

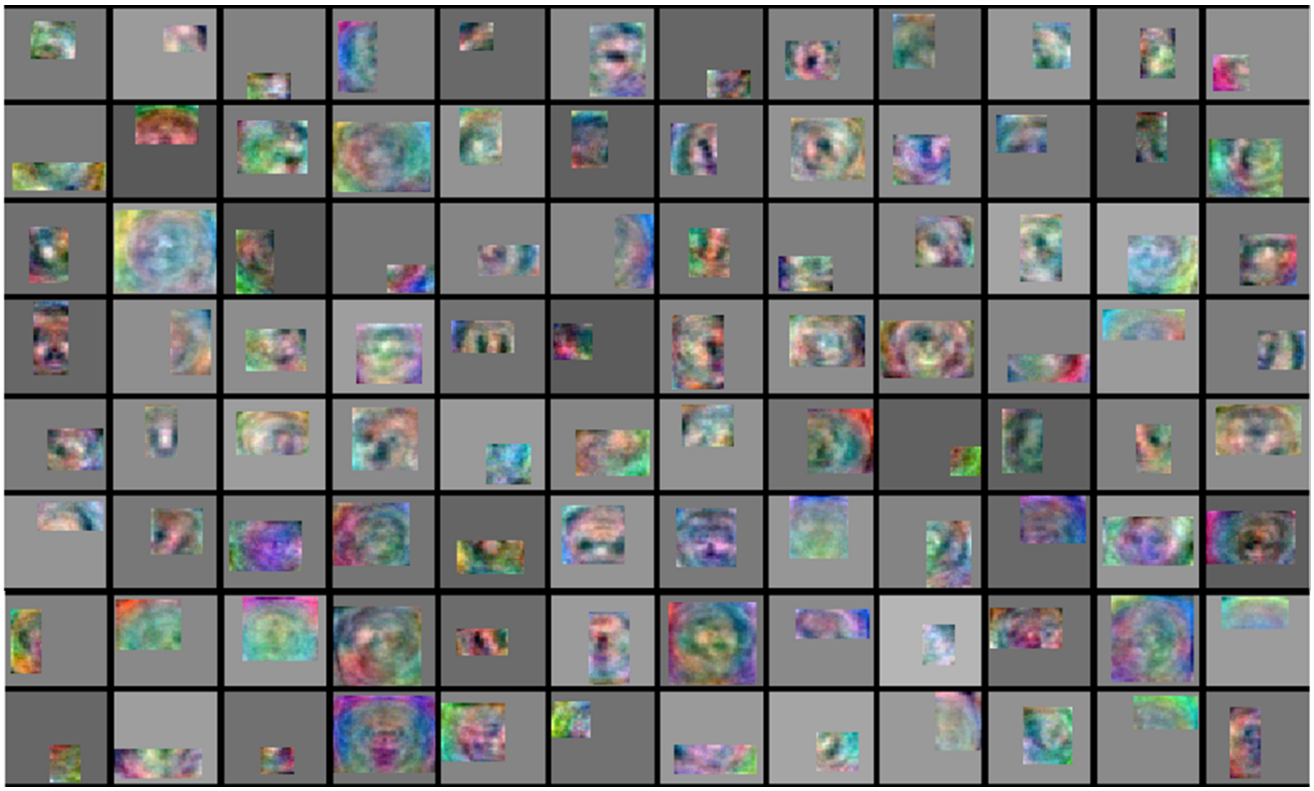
Examples of learned filters for head and body orientation estimation using the CRPnet are shown in Figure 13. Small filters usually extract local visual information such as edges, and medium-scale filters capture partial shapes. On the other hand, larger filters learn holistic shapes and visual appearance under varying lighting conditions. For example, small filters learn to represent edges of a shoulder, an arm, or a gap between legs for the HOC dataset. Other filters learn to describe head-shoulder shapes or body silhouettes. Fewer learned filters represent hands or feet since these body parts appear in diverse locations and poses as shown in the last column.

7.3 Head and Body Orientation Estimation

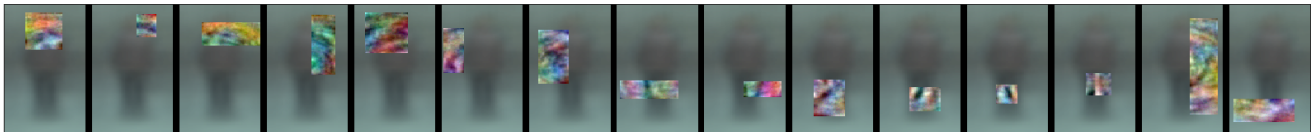
7.3.1 Orientation Classification

We evaluate the proposed algorithm against the state-of-the-art orientation classification methods [6], [7], [16], [34] in terms of the image scale variation, noise, occlusion, blurring, and computational time. In addition, we also report results using convolutional neural network structures that perform well in numerous computer vision tasks, such as image recognition, over the last few years. Since the the size of the input image and the number of output classes are different from conventional CNNs, we train CNNs in two different ways: fine-tuning an existing CNN model [35] or designing problem-specific CNNs. The network [35] is trained based on the VGG-Very-Deep-16 CNN architecture and applied to the face recognition problem. Thus, this network is more relevant to our task than other CNNs tuned for generic object recognition. However, our experiments show that the fine-tuned network does not performs well, e.g., 85% for the QMUL dataset after 1,000 epochs, than other approaches designed for head pose estimation. This can be attributed to the fact that the number of training samples is not sufficient to learn such a deep network. In addition, it takes more than 200 ms to process an image of 224×224 pixels. Thus, we focus on the design of CNNs as shown in Figure 14. Note that we apply the same data augmentation techniques (described in Section 5) to train CNNs, CRPnet, and CRPforest for fair comparisons.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) HIIT dataset



(b) HOC dataset

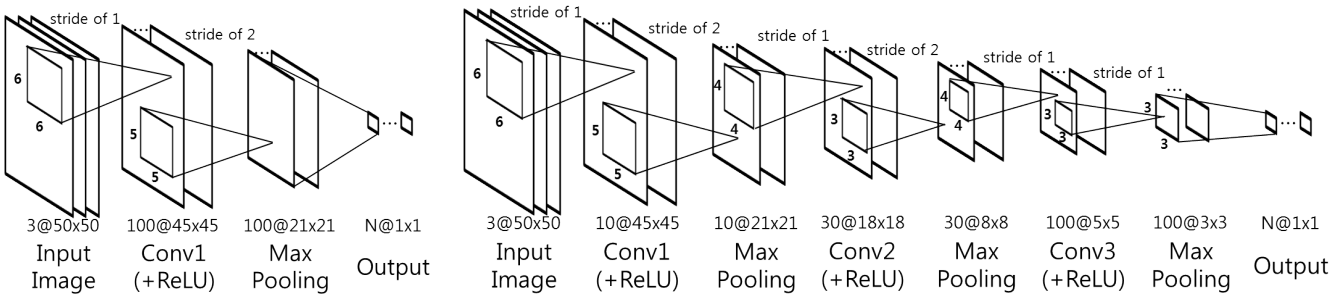
Fig. 13. Examples of trained filters for head pose estimation on the HIIT dataset and the HOC dataset using the CRPnet. Small filters extract simple local information such as edges, and medium-scale filters capture partial shapes. Larger filters learn holistic shapes and lighting conditions. Best viewed in color.

TABLE 1
Classification accuracy on the HIIT, QMUL, and QMULB datasets at different image sizes. The results of [7] and [34] are obtained from the papers. [16]a and [16]b are methods proposed by [16] based on the Frobenius distance and the CBH distance, respectively.

Dataset	Size	[7]	[34]	[16]a	[16]b	RPF [6]	Shallow CNN	3-layer CNN	CRPnet	CRPforest
HIIT	15×15	-	-	82.4%	84.6%	89.7%	92.6%	91.6%	95.8%	**97.9**%
	25×25	-	-	89.6%	90.4%	95.5%	97.3%	97.1%	96.1%	**98.2**%
	50×50	-	-	95.3%	95.7%	97.6%	98.2%	97.8%	96.3%	**98.3**%
QMUL	15×15	-	-	59.5%	59.8%	92.8%	92.8%	94.4%	92.4%	**95.0**%
	25×25	-	-	82.6%	83.2%	94.3%	93.9%	95.3%	92.4%	**95.3**%
	50×50	82.3%	93.5%	94.3%	94.9%	95.2%	95.0%	95.2%	92.4%	**95.3**%
QMULB	15×15	-	-	54.5%	57%	87.1%	90.1%	91.4%	90.4%	**92.0**%
	25×25	-	-	76.5%	76.9%	91.0%	90.9%	92.2%	90.7%	**92.6**%
	50×50	64.2%	89%	91.8%	92.0%	92.2%	92.2%	92.3%	90.8%	**92.4**%
Time	50×50	-	-	$550ms$	$1,689ms$	$295ms$	$5.64ms$	$5.27ms$	**0.04ms**	$3.83ms$

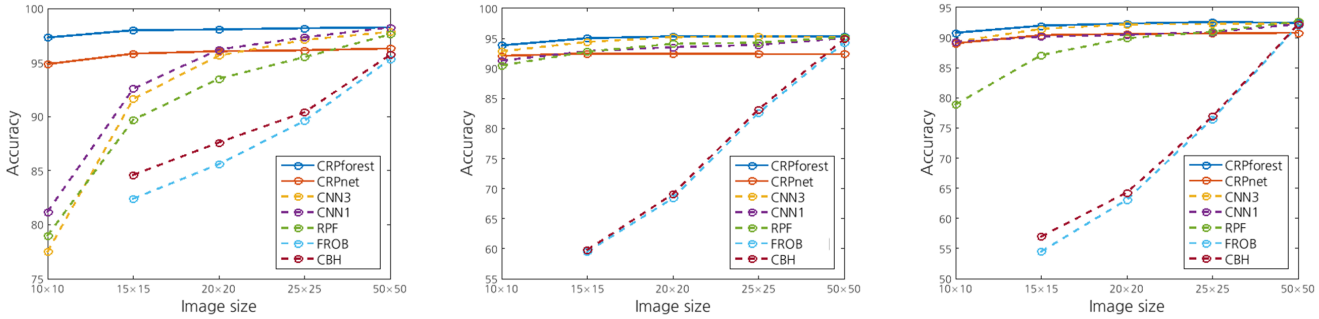
Table 1 summarizes the results by evaluated head pose classification methods on three datasets with different image sizes. We observe that the pose estimation method [6] trains a model for each image size. On the contrary, we train a single model based on images of 50×50 pixels for pose estimation. Test images of lower resolution are resized to the size of 50×50 pixels using bilinear interpolation. Figure 15 shows results with respect to image sizes. Overall, the proposed algorithm performs robustly with respect to size variations against the other methods. The proposed algorithm achieves almost the same estimation accuracy, for example, about 98% on the HIIT dataset until the image size is reduced to 10×10 pixels. We note that accuracy of the method in [16] decreases rapidly when the image size is reduced to below 50×50 pixels, and does not operate when the image size is smaller than 15×15 pixels (using the code

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence



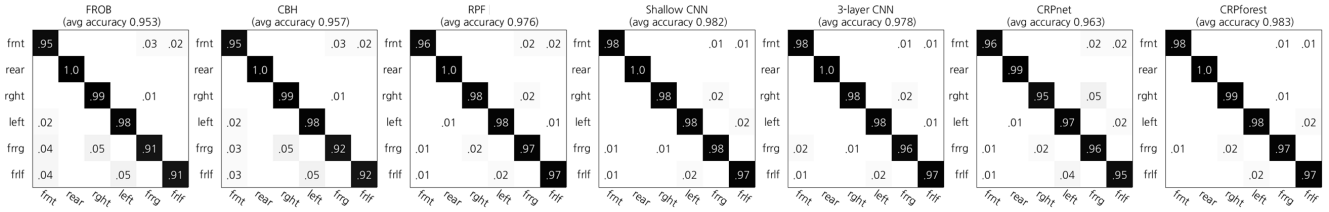
(a) A shallow convolutional neural network (b) CNN with 3 convolutional layers

Fig. 14. Structures of designed CNNs used for comparison in this work (N is the number of output classes).

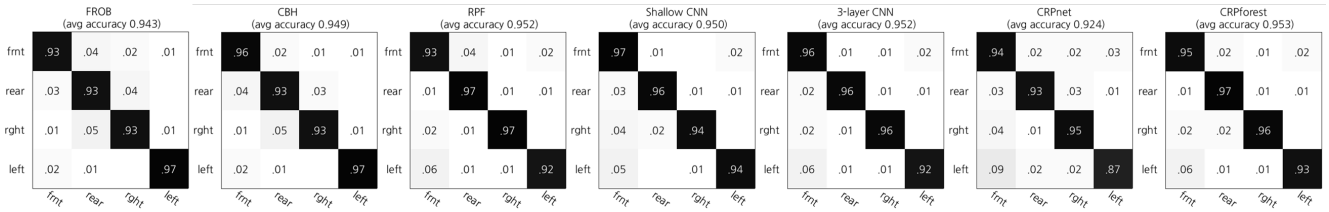


(a) HIIT dataset (b) QMUL dataset (c) QMULB dataset

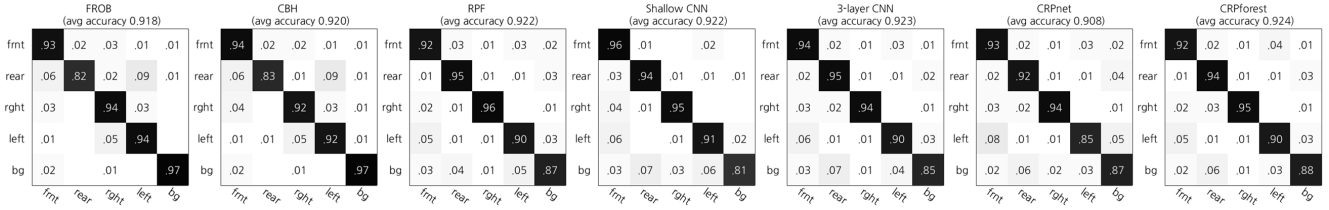
Fig. 15. Accuracy of head pose estimation algorithms on different sizes of test data. All algorithms are tested using a single estimation model trained based on 50×50 pixels training data. CNN1 and CNN3 refer the shallow CNN and 3-layer CNN in 14(a) and 14(b), respectively. RPF is the random projection forest algorithm in [6]. FROB and CBH stand for methods in [16].



(a) HIIT dataset



(b) QMUL dataset

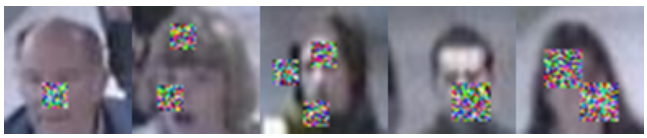


(c) QMULB dataset

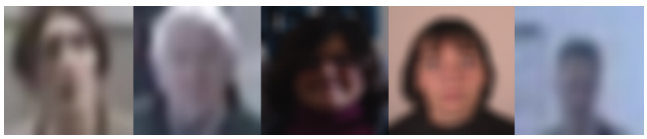
Fig. 16. Confusion matrices of orientation estimation algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence

(a) Add a Gaussian noise at each pixel



(b) Generate random rectangles to synthesize occlusions



(c) Use Gaussian kernel to blur images

Fig. 17. Examples of degraded images on the QMUL dataset.

provided by the authors).

Figure 16 shows the confusion matrices of existing pose estimation methods [6], [16], CNNs, CRPnet, and CRPforest using images of 50×50 pixels from three datasets. Overall, the proposed algorithm is able to estimate head orientations well in all poses. Furthermore, the results on the QMULB dataset indicate that the proposed algorithm is capable of filtering out 90% of background images while estimating head poses.

Head pose classification methods are evaluated and analyzed when input images are degraded by noise, occlusions, and blurs as shown in Figure 17. We add Gaussian noise with kernel width σ_n at each pixel to generate noisy test images. For blurry images, we filters an input image with a Gaussian smoothing kernel with the kernel width σ_b and the size of 5×5 pixels. The occluded images are generated using noisy rectangles randomly located in five settings: (1) one 10×10 rectangle, (2) two 10×10 rectangles, (3) three 10×10 rectangles, (4) one 15×15 rectangle, and (5) two 15×15 rectangles. Note that the intensity value of each pixel in a degraded image is truncated to be within a range of 0 to 255.

Figure 18 shows the performance of evaluated pose estimation methods on degraded images. Overall, the proposed algorithm performs well against other methods for images with different types of degradation. For noisy images, the accuracy of CNNs drops faster than the CRPnet or CRPforest algorithms. This can be attributed to the fact that the max-pooling layer is sensitive to large noise. For blurred images, as the accuracy of [16] drops significantly with a small amount of blur, we only plot the results by the other methods for better visualization. The proposed algorithm performs robustly against different types of blur. For occluded images, two existing methods [6], [16] perform as well as the CRPforest algorithm.

The results of body orientation estimation on the HOC dataset with different image sizes are shown in Table 2. Similar to experiments on head poses, the proposed CRPforest algorithm is effective and robust to variations of input images for estimating body orientations.

TABLE 2
Classification accuracy on the HOC dataset at different image sizes. The architectures of CNN1 and CNN3 are similar to the shallow CNN and 3-layer CNN in 14(a) and 14(b), respectively. For this dataset, CNN1 has a stride of 2 for the convolutional layer and stride of 4 for the pooling layer. CNN3 has a stride of 2 for the last pooling layer.

Size	[16]a	[16]b	CNN1	CNN3	CRPnet	CRPforest
66×31	71%	73%	78.2%	78.3%	76.6%	**81.2%**
99×47	77%	78%	78.4%	80.8%	76.7%	**81.3%**
132×62	78%	78%	79.3%	81.3%	76.7%	**81.3%**

TABLE 3
Regression accuracy on the Multi-PIE dataset. MAE: Mean absolute error in degrees.

	[36]	[37]	[38]	[39]	[40]	Proposed
MAE	5.31°	4.33°	4.12°	2.99°	1.25°	**1.12°**

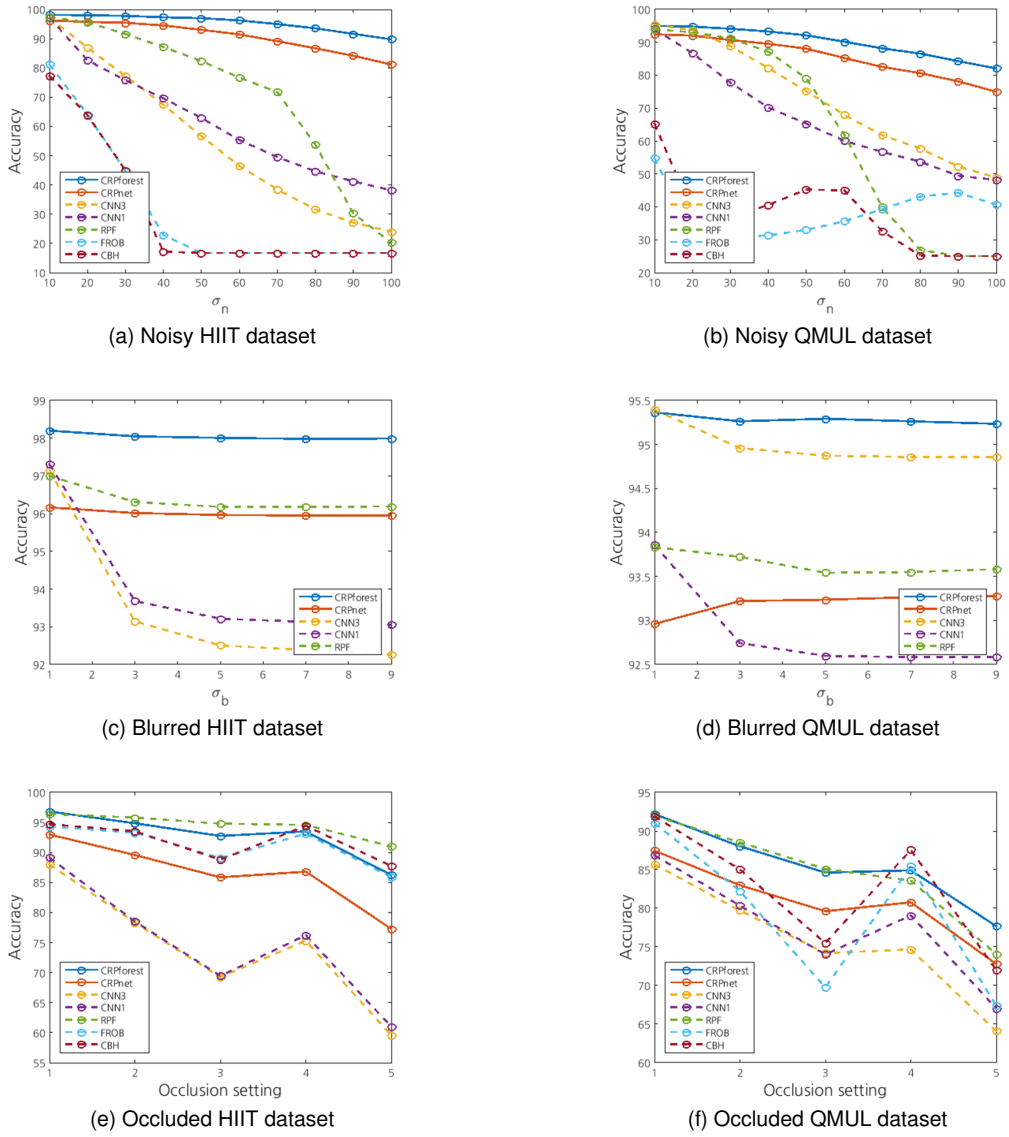
TABLE 4
Regression accuracy on the FacePix dataset. MAE: Mean absolute error in degrees.

	[41]	[42]	[43]	[44]	[45]	Proposed
MAE	6.1°	3.96°	2.75°	2.74°	2.71°	**2.38°**

The aforementioned results show the effectiveness and robustness of the proposed algorithm for estimation of head poses and body orientations. The CRPforest algorithm performs favorably against other methods for all datasets and all types of degraded images. In particular, the CRPforest algorithm performs robustly on low resolution images degraded by motion blur and noise. It is important to develop such methods for applications such as autonomous driving when low resolution images obtained at a distance with different types of degradations. It is worth noticing that the number of convolutions in the CRPnet is independent of the image size. Thus, this method can be applied to some applications as a trade-off between speed and accuracy. Despite the simple network architecture, it performs well with a small number of filters and robust to corruptions as shown in Figure 18. The importance of combining random forests with the CRPnet is demonstrated in the experiments as the CRPforest algorithm performs better than the CRPnet in all cases. In addition, each split in trees is more effective as we use discriminatively learned filters instead of box filters [6]. This allows us to decrease the number of filters computed at each node from 1,000 [6] to 100 while the CRPforest algorithm performs better.

We report the run-time performance of the proposed algorithm on a computer with 3.3 GHz CPUs with an image of 50×50 pixels. The manifold embedding based approach [16] takes 550 ms (using Frobenius norm) and 1,689 ms (using CBH norm). It takes 295 ms for the method based on random projection forests [6]. Note that it is different from the reported time in [6] since they fed the entire dataset to the algorithm and measured the average time. In contrast, the CRPnet takes only 0.04 ms to process one image. The proposed CRPforest algorithm takes 3.83 ms per image for pose estimation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Noisy HIIT dataset (b) Noisy QMUL dataset

(c) Blurred HIIT dataset (d) Blurred QMUL dataset

(e) Occluded HIIT dataset (f) Occluded QMUL dataset

Fig. 18. Classification accuracy on degraded datasets by different algorithms. CNN1 and CNN3 refer the shallow CNN and 3-layer CNN in 14(a) and 14(b), respectively. RPF is the random projection forest algorithm in [6]. FROB and CBH stand for methods in [16]. Examples of degrade samples are provided in Figure 17.

7.3.2 *Orientation Regression*

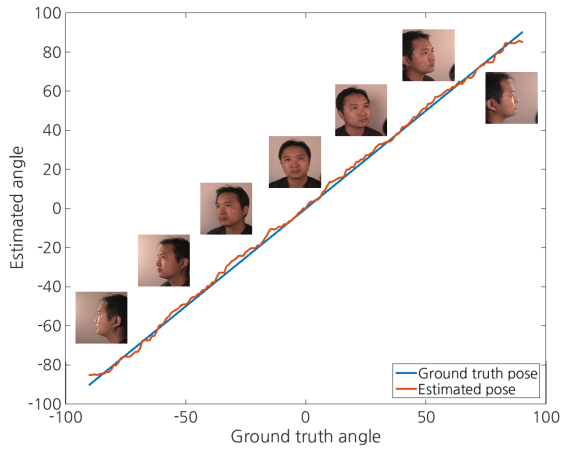
We use the CMU-MultiPIE and FacePix datasets to evaluate head pose regression results. As the number of training samples is relatively small for training the CRPnet, we report the results using box filters described in Section 4.2 for head pose regression. Table 3 and 4 summarize the performance of head pose regressors on the CMU Multi-PIE and FacePix datasets, respectively. We measure the mean absolute error (MAE) between the estimated head pose and ground truth head pose in degree. Examples of head pose regression results on the FacePix dataset are shown in Figure 19. The results show that head poses in all ranges of the yaw degrees can be regressed well. Overall, the proposed algorithm performs favorably against the other methods for head pose regression.

We note that existing methods in the literature use different configurations. For the CMU Multi-PIE dataset, the methods [36], [37], [38], [39] use only a subset for evaluation. On the other hand, the proposed algorithm is evaluated on the entire dataset. For the FacePix dataset, the yaw interval is changed to 2 degrees instead of 1 degree in [41], or the estimation range is narrowed down to $-45°$ to $45°$ [42], [45]. In contrast, the proposed algorithm performs favorably with respect to the other methods based on evaluation of the entire dataset (i.e., 5,430 images, yaw degree from $-90°$ to $90°$ with $1°$ interval and leave-one-out cross validation). As the source code for the above methods are not available to the public, we are not able to carry out evaluations using the entire dataset.

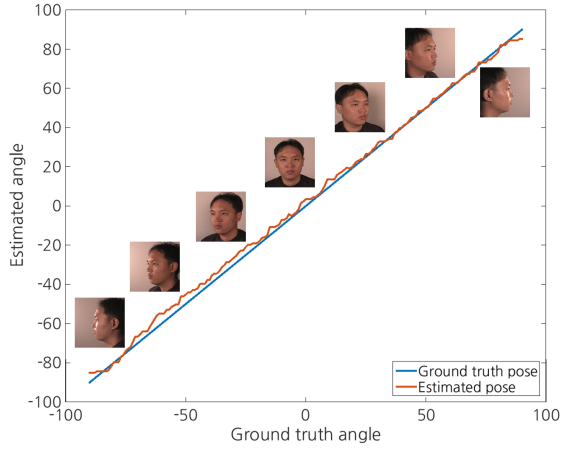
8 Conclusions

In this paper, we propose a fast and accurate orientation estimation algorithm based on the convolutional random projection forest. It is equipped with the convolutional

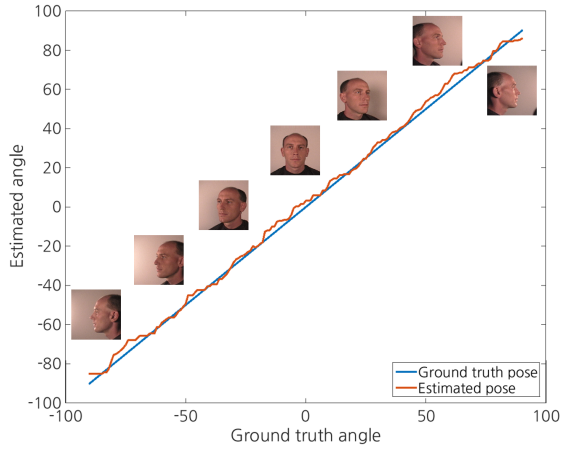
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2784424, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a)



(b)



(c)

Fig. 19. Head pose regression results for three different subjects of the FacePix dataset. Images in the dataset are plotted on the graph at corresponding angles.

random projection network as a split function at each node. The network learns a rich filter bank while compressing and classifying its responses based on the compressive sensing technique. By using a very sparse random projection matrix for the compression, we can keep light computational costs. Based on the filters trained on sub-regions of the input image using the CRPnet, the CRPforest can achieve high accuracy with a fraction of the running time. Extensive experiments on challenging benchmark datasets show that the proposed algorithm performs favorably against the state-of-the-art methods on low-resolution images degraded by noise, occlusions, and blurs.

REFERENCES

[1] K. Smith, S. O. Ba, J.-M. Odobez, and D. Gatica-Perez, "Tracking the visual focus of attention for a varying number of wandering people," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1212–1229, 2008. 1

[2] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image and Vision Computing*, vol. 27, no. 12, pp. 1743–1759, 2009. 1

[3] K. Sakita, K. Ogawara, S. Murakami, K. Kawamura, and K. Ikeuchi, "Flexible cooperation between human and robot by interpreting human intention from gaze information," in *Proc. of the IEEE Intelligent Robots and Systems*, 2004. 1

[4] T. Siriteerakul, "Advance in head pose estimation from low resolution images: A review," *International Journal of Computer Science Issues*, vol. 9, no. 2, 2012. 1, 2

[5] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. 2

[6] D. Lee, M.-H. Yang, and S. Oh, "Fast and accurate head pose estimation via random projection forests," in *In Proc. of the IEEE International Conference on Computer Vision*, 2015. 2, 4, 5, 6, 8, 9, 10, 11, 12

[7] J. Orozco, S. Gong, and T. Xiang, "Head pose classification in crowded scenes." in *Proc. of the British Machine Vision Conference*, 2009. 2, 7, 8, 9

[8] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, 2009. 2

[9] Z. Zhang, Y. Hu, M. Liu, and T. Huang, "Head pose estimation in seminar room using multi view face detectors," in *International Evaluation Workshop on Classification of Events, Activities and Relationships*, 2007. 2

[10] M. Enzweiler and D. M. Gavrila, "Integrated pedestrian classification and orientation estimation," in *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2010. 2

[11] J. Tao and R. Klette, "Integrated pedestrian and direction classification using a random decision forest," in *Proc. of the IEEE International Conference on Computer Vision Workshops*, 2013. 2

[12] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, and R. Horaud, "Head pose estimation via probabilistic high-dimensional regression," in *Proc. of the IEEE International Conference on Image Processing*, 2015. 2

[13] G. Fanelli, J. Gall, and L. V. Gool, "Real time head pose estimation with random regression forests," in *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2011. 2

[14] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013. 2

[15] H. T. Ho and R. Chellappa, "Automatic head pose estimation using randomly projected dense SIFT descriptors," in *Proc. of the IEEE International Conference on Image Processing*, 2012. 2

[16] D. Tosato, M. Spera, M. Cristani, and V. Murino, "Characterizing humans on Riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1972–1984, 2013. 2, 7, 8, 9, 10, 11, 12

[17] K. Sundararajan and D. L. Woodard, "Head pose estimation in the wild using approximate view manifolds," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015. 2

[18] A. Bansal, A. Shrivastava, C. Doersch, and A. Gupta, "Mid-level elements for object detection," *arXiv preprint arXiv:1504.07284*, 2015. 2

[19] Y. Li, L. Liu, C. Shen, and A. van den Hengel, "Mid-level deep pattern mining," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[20] C. Doersch, A. Gupta, and A. A. Efros, "Context as supervisory signal: Discovering objects with predictable context," in *Proc. of the European Conference on Computer Vision*, 2014. 2

[21] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. of the European Conference on Computer Vision*, 2012. 2

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of the Advances in neural information processing systems*, 2012. 3

[23] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008. 3

[24] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proc. of the ACM International Conference on Knowledge Discovery and Data mining*, 2006. 3, 5

[25] R. L. Graham, *Concrete mathematics: a foundation for computer science*. Pearson Education India, 1994. 3

[26] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2002–2015, 2014. 4

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 4

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. of the European Conference on Computer Vision*, 2014. 4

[29] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of machine learning research*, vol. 2, no. Dec, pp. 265–292, 2001. 5

[30] P. Kontschieder, M. Fiterau, A. Criminisi, and S. Rota Bulo, "Deep neural decision forests," in *Proc. of the IEEE International Conference on Computer Vision*, 2015. 6

[31] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010. 7

[32] D. Little, S. Krishna, J. Black, and S. Panchanathan, "A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose angle and illumination angle." in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005. 7

[33] W. R. Schwartz and L. S. Davis, "Learning discriminative appearance-based models using partial least squares," in *Brazilian Symposium on Computer Graphics and Image Processing*, 2009. 7

[34] D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani, "Multi-class classification on Riemannian manifolds for video surveillance," in *Proc. of the European Conference on Computer Vision*, 2010. 8, 9

[35] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. of the British Machine Vision Conference*, 2015. 8

[36] M. A. Haj, J. Gonzalez, and L. S. Davis, "On partial least squares in head pose estimation: How to simultaneously deal with misalignment," in *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2012. 11, 12

[37] D. Huang, M. Storer, F. D. la Torre, and H. Bischof, "Supervised local subspace learning for continuous head pose estimation," in *Proc. of the IEEE Computer Vision and Pattern Recognition*, 2011. 11, 12

[38] X. Peng, J. Huang, Q. Hu, S. Zhang, and D. Metaxas, "Head pose estimation by instance parameterization," in *Proc. of the IEEE International Conference on Pattern Recognition*, 2014. 11, 12

[39] F. Jiang, H. K. Ekenel, and B. E. Shi, "Efficient and robust integration of face detection and head pose estimation," in *Proc. of the IEEE International Conference on Pattern Recognition*, 2012. 11, 12

[40] B. Han, S. Lee, and H. S. Yang, "Head pose estimation using image abstraction and local directional quaternary patterns for multiclass classification," *Pattern Recognition Letters*, vol. 45, pp. 145–153, 2014. 11

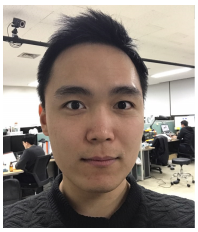
[41] H. Ji, R. Liu, F. Su, Z. Su, and Y. Tian, "Robust head pose estimation via convex regularized sparse regression," in *Proc. of the IEEE International Conference on Image Processing*, 2011. 11, 12

[42] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and R. MV, "Fully automatic pose-invariant face recognition via 3D pose normalization," in *Proc. of the IEEE International Conference on Computer Vision*, 2011. 11, 12

[43] J. Foytik and V. K. Asari, "A two-layer framework for piecewise linear manifold-based head pose estimation," *International journal of computer vision*, vol. 101, no. 2, pp. 270–287, 2013. 11

[44] B. Ma, R. Huang, and L. Qin, "Vod: A novel image representation for head yaw estimation," *Neurocomputing*, vol. 148, pp. 455–466, 2015. 11

[45] A. Dahmane, S. Larabi, I. M. Bilasco, and C. Djeraba, "Head pose estimation based on face symmetry analysis," *Signal, Image and Video Processing*, pp. 1–10, 2014. 11, 12



Donghoon Lee is a PhD student in Electrical and Computer Engineering at Seoul National University, Seoul, Korea. He received the B.S. and M.S. degrees in electrical and computer engineering in the same school in 2011 and 2013, respectively. His research interests include machine learning and computer vision.

Ming-Hsuan Yang is a professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute working on vision problems related to humanoid robots. He coauthored the book Face Detection and Gesture Recognition for Human-Computer Interaction (Kluwer Academic 2001) and edited special issue on face recognition for Computer Vision and Image Understanding in 2003, and a special issue on real world face recognition for IEEE Transactions on Pattern Analysis and Machine Intelligence. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011, and is an associate editor of the International Journal of Computer Vision, Image and Vision Computing and Journal of Artificial Intelligence Research. He received the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.

Songhwai Oh received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1995, 2003, and 2006, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea. Before his Ph.D. studies, he was a Senior Software Engineer at Synopsys, Inc., Mountain View, CA, USA, and a Microprocessor Design Engineer at Intel Corporation, Santa Clara, CA, USA. In 2007, he was a Post-Doctoral Researcher with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA. From 2007 to 2009, he was an Assistant Professor of Electrical Engineering and Computer Science in the School of Engineering, University of California, Merced, CA, USA. His current research interests include robotics, computer vision, cyber-physical systems, and machine learning.