Expert Systems With Applications 87 (2017) 304–315

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Sensitive deep convolutional neural network for face recognition at large standoffs with small dataset

Amin Jalali, Rammohan Mallipeddi, Minho Lee*

School of Electronics Engineering, Kyungpook National University, 1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Korea

ARTICLE INFO

Article history:
Received 1 December 2016
Revised 14 June 2017
Accepted 15 June 2017
Available online 16 June 2017

Keywords:
Convolutional neural network
Gradient descent
Input-output mapping sensitivity error back propagation
Face recognition at long distances with small dataset
Sensitivity in cost function
Deep neural structures

ABSTRACT

In this paper, we propose a sensitive convolutional neural network which incorporates sensitivity term in the cost function of Convolutional Neural Network (CNN) to emphasize on the slight variations and high frequency components in highly blurred input image samples. The proposed cost function in CNN has a sensitivity part in which the conventional error is divided by the derivative of the activation function, and subsequently the total error is minimized by the gradient descent method during the learning process. Due to the proposed sensitivity term, the data samples at the decision boundaries appear more on the middle band or the high gradient part of the activation function. This highlights the slight changes in the highly blurred input images enabling better feature extraction resulting in better generalization and improved classification performance in the highly blurred images. To study the effect of the proposed sensitivity term, experiments were performed for the face recognition task on small dataset of facial images at different long standoffs in both night-time and day-time modalities.

1. Introduction

Recently, deep neural networks such as Convolutional Neural Networks (CNN) have been successfully applied in many recognition problems (Ciresan, Meier, & Schmidhuber, 2012). The conceptual architecture of CNN is inspired by Hubel and Wiesel (1959)'s seminal work on the cat's striate cortex called receptive field. Later, Fukushima (1980) explained the Neocognitron, which defines the layer wise structure of neural networks and explains the spatial invariance characteristic of simple cells and complex cells of visual primary cortex. LeCun introduced the structure of CNN for face and digit recognition (LeCun & Bengio, 1995; LeCun, Bottou, Bengio, & Haffner, 1998; LeCun, Huang, & Bottou, 2004), which demonstrated better recognition results than probability density function methodologies (e.g., Gaussian Bayesian approaches and Gaussian Mixture models) or nonparametric clustering approaches (e.g., K-nearest neighbor classifiers). Rowley, Baluja, and Kanade (1998) utilized CNN for face recognition with three layers and three receptive fields in the first layer. Simard, Steinkraus, and Platt (2003) proposed the implementation of a more efficient subsampling approach in the operation of the convolutional layers instead of a separate subsampling layer leading to a faster algorithm in terms of training.

Simonyan and Zisserman (2015) proved that very deep convolutional networks are effective for large scale image classification. As the depth increases, the performance of the network improves on complex recognition tasks. Szegedy et al. (2015) presented a deep CNN structure called "Inception" which has salient features regarding computing resources of the network. It is accomplished by going deeper with convolutions and increasing the depth and width of the network as the computational cost is kept constant. In addition, it utilizes the Hebbian principle and multi-scale processing in its network architecture. He, Zhang, Ren, and Sun (2016) proposed a deep residual learning framework to simplify the training process of the networks that are deep. It is done by introducing the learning residual functions with reference to the input layers. The accuracy of residual network enhances as the depth increases while having lower complexity. Schroff, Kalenichenko, and Philbin (2015) presented FaceNet which learns a mapping from face samples to a Euclidean space. Those distances represent the similarity and form the feature space. They utilized deep convolutional network trained by triplets of roughly aligned face patches using online mining method. This method could achieve the state-of-the-art recognition performance on the Labeled Faces in the Wild (LFW) and YouTube faces datasets. The studies discussed above prove the effectiveness of deep networks in the recognition tasks.

* Corresponding author.
E-mail addresses: max.jalali@gmail.com (A. Jalali), mallipeddi.ram@gmail.com (R. Mallipeddi), mholee@gmail.com, mholee@knu.ac.kr (M. Lee).

http://dx.doi.org/10.1016/j.eswa.2017.06.025
0957-4174/

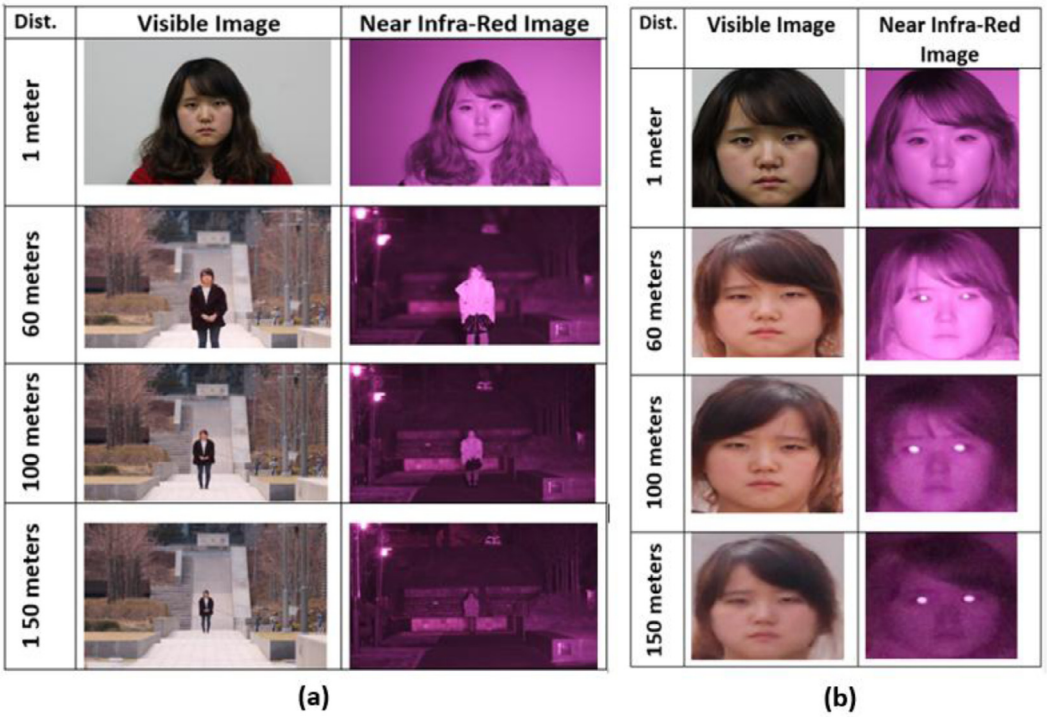


Fig. 1. Examples of long distance and near infrared facial images. (a) Eight images in LDHF database (four near infrared images and four visible images at 1 m, 60 m, 100 m, and 150 m, respectively) for each subject. (b) Cropped sample faces of a subject in LDHF dataset are shown.

In some practical applications such as satellite imaging, long distance object recognition, and public safety control, images are generally captured in unconstrained illumination and environments with complexities such as fog, cloudy, long distance and blurry conditions. Therefore, the quality of the images obtained in such systems is of degraded quality and hence the recognition becomes a challenging task. Generally, major manual processing combined with automatic image recognition methods are applied on the captured images to get acceptable accuracy and efficiency. However, the recognition task becomes further complicated when the number of samples are small. Therefore, for a given dataset with small number of samples and problem complexity, the design of the learning algorithm to have appropriate generalization is the issue of this study.

The objective of this paper is to propose an improved feature extraction and generalization for highly blurred images by considering the sensitive cost function for training process of deep-CNN. The sensitivity regularization term in training algorithm considers the small changes of input images to make the CNN more sensitive to images with high intra-class variance and low inter-class variance. To evaluate the efficiency of the proposed approach the Long Distance Heterogeneous Face dataset (LDHF) (Kang, Han, Jain, & Lee, 2014) is used. This dataset contain few sample images with different face sizes, different modalities (night-time and day-time), and different quality degradations (e.g., blur and noise) at different distances. A sample subject of the LDHF dataset with its corresponding cropped faces is shown in Fig. 1. The incorporated sensitivity term is expected to highlight the small variations in the pixels of the blurred low illumination images during the learning process by better feature extraction. In other words, by incorporating the sensitivity term in the CNN cost function, the neural activations of the hidden layers are located at high gradient region of the activation function for areas with small variations. This results in the modifications of the corresponding weights of the structure during training. To implement the sensitivity in CNN structure, a high pass filter is inherently incorporated in cost function of the error back propagation learning algorithm to highlight the edges and high frequency variations in the blurred images. Throughout the training process, as the error flows back to the first layer, the weights updating takes into consideration the small changes in the input images due to the presence of high pass filter resulting in better classification.

To demonstrate the effectiveness of the proposed sensitivity term, experiments were performed using a face dataset consisting different scenarios. First, the recognition performance in images that are degraded, blurred and with low illumination obtained from a specific distance during day-time (Table 4) and night-time (Table 5) is considered. The proposed method introduces a regularization method to make the deep structure more sensitive to images with high intra-class variance and low inter-class variance in datasets consisting very few data samples with high complexities. It is noticed that the augmented dataset is too small including 6 samples per distance. The intention of proposed sensitivity approach is to highlight the small variations and emphasize the high frequency components for better internal representation of features and further generalization while the dataset has just a few samples. The second scenario investigates the recognition task of images of one modality (either day-time (Table 6) or night-time (Table 7)) for a particular distance compared with a pool of other distances in which the complexity of the dataset increases. Images from one distance are selected for test process and images from other standoffs are integrated during the training process. Due to very small number of samples in the dataset, we integrate the images from other distances during the training process to make the dataset larger. Furthermore, integration of samples makes it more complicated due to high difference between the samples as the distance increases. This experiment illustrates the impact of the proposed method on the performance of deep-CNN by increasing both complexity and data samples. Third experimental setup is the recognition task in the pool of integrated images of night-time and day-time at different standoffs (Table 8) in which images of different modalities and distances are all mixed together to evaluate the

performance of proposed sensitivity term in training of the deep-CNN.

The outline of the paper is as follow; Section 2 introduces the literature review on regularization and generalization methods applied to deep CNNs containing small datasets. Section 3 includes the proposed sensitive CNN with improved input-output mapping sensitivity. Section 4 shows the experimental results, and finally Section 5 presents the discussion and conclusion with a future work.

2. Literature review

CNN is usually trained by supervised gradient descent. When the amount of data is limited, CNNs reveal their strong dependence on large amounts of training data. A proper starting point could lead to valid convergence and generalization. This starting point was found using unsupervised feature learning techniques such as sparse coding or transfer learning. Wagner, Thom, Schweiger, Palm, and Rothermel (2013) compared these methods against simple initialization scheme and showed that pre-training helps to train CNNs from few samples which can push the network to better performance. Hafemann, Oliveira, Cavalin, and Sabourin (2015) investigated transfer learning using CNNs, to take advantage of this type of architecture to problems with smaller datasets. The trained CNN with lots of data projects the target dataset onto another feature space, and then train a classifier on top of this new representation. This method utilizes knowledge learned from tasks with larger dataset in the tasks including small datasets.

Another approach to deal with small dataset is using geometry features to enhance the internal representations of features. Jung, Lee, Yim, Park, & Kim (2015) presented a deep network based on two different models. The first network extracts temporal appearance features from images, while the other network extracts geometry features from facial landmark points. These two models are combined to boost the performance of facial expression recognition.

Annotated web images are great resource of data to augment training dataset while having limited number of samples. Xu, Huang, Zhang, and Tao (2015) proposed a method for fine-grained object recognition that uses part-level annotations and deep CNNs in a unified framework. This method improves classification accuracy in two ways: more discriminative features are generated using a training set augmented by collection a large number of part patches from web images; and more robust object classifiers are learned using a multi-instance learning algorithm jointly on the strong and weak dataset.

In order to enhance the network accuracy, Jalali, Jang, Kang, and Lee (2015) presented regularization method in the training process of CNN. They proposed a novel activation function to improve robustness and further regularization to the outliers in the input data samples. Data samples on the decision boundaries are given more weight by adding the derivatives of the Sigmoid function outputs to avoid abrupt update of the network weights. Therefore, the CNN becomes more robust and generalized to outliers and noisy patterns.

Song, Gao, Ding, and Wang (2016) presented several dataset expansions techniques to expand the scale of available samples. They increased the performance of the system by augmenting the training set containing insufficient samples in both scale and quality. These techniques include random elastic deformation, shear transformation and rotation with a small range, etc. Kolář, Hradiš, and Zemčík (2016) proposed a semi-supervised self-training bootstrap to deep learning which retrieves and utilizes additional images from internet image search. This method tackles the problem of small amounts of dataset to enhance the number of samples from internet resources.

Despite all these augmentations techniques and regularization approaches in deep neural structures, they did not consider the implementation of a training algorithm that can consider the small variations in the image pixels for a better internal representation. As such, they did not introduce any regularization method to make the deep structure more sensitive to images with high intra-class variance and low inter-class variance in datasets consisting very few data samples. In this study, we incorporated the proposed sensitive cost function in deep-CNN to change the neural activations of hidden layers to locate at high gradient region of activation function. This method modifies the neural weights to highlight the edges and high frequency variation in the images.

The presented method is evaluated on highly degraded long distance images with limited data samples at night-time and day-time modalities to be sensitive to high intra-class variances. The proposed method enhances the sensitivity of CNN structure so that the edges and small changes of the input images are highlighted for better feature extraction and improve recognition accuracy. It is done by using a penalty term in the training algorithm which comes from the definition of high pass filter in error measurement function. In order to test the performance of the presented method, we used the LDHF dataset. Kang et al. (2014) collected the LDHF dataset and they performed face verification using an image restoration based on Locally Linear Embedding (LLE) called Augmented Heterogeneous Face Recognition (AHFR).

Since the LDHF dataset contains blurred and low-luminance images at large standoffs with few data samples at both modalities, it is a proper choice for evaluating of our approach. This dataset does not have sufficient samples to train deep structures like CNN. The LDHF dataset is limited to one sample per distance for night-time and day-time and the number of dataset samples is very small. To deal with this matter, we augmented the dataset. So, in addition to its inherent perturbations at long distances and difference modalities, variant types of disturbances and deformations are implemented to dataset to make it large and more perturbed. In spite of other studies which perform image matching and image restoration, we evaluate the performance of the proposed added sensitivity term in deep structure to distinguish the high frequency components and small changes in highly disturbed input samples in classification problem. Therefore, the proposed approach is expected to provide better feature extraction and generalization resulting in improved classification.

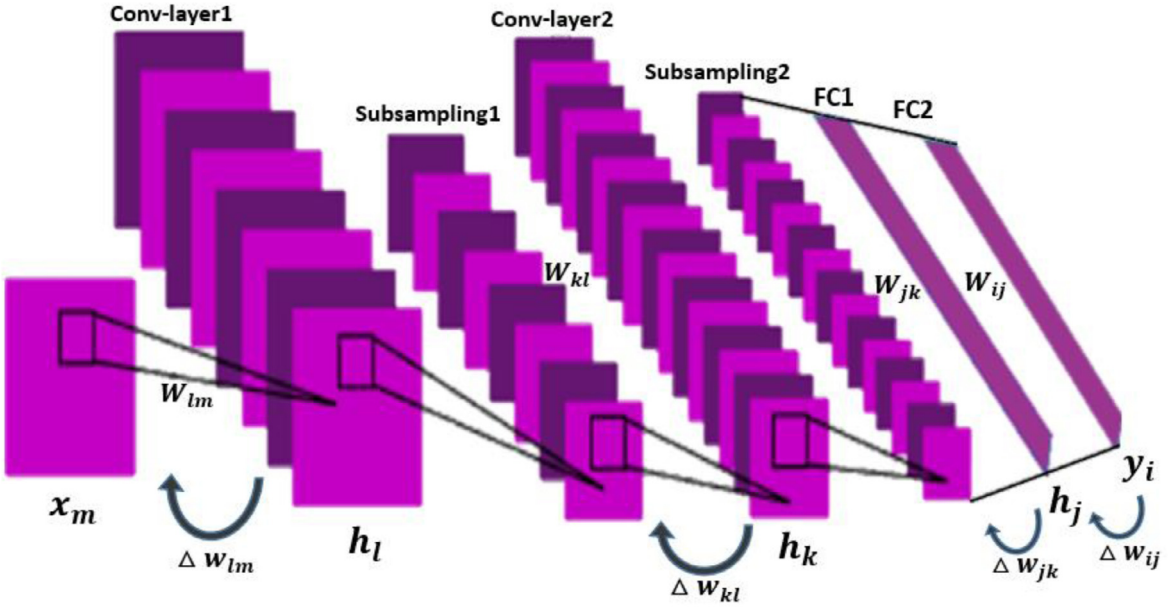
3. Input-output mapping sensitivity error back propagation learning algorithm

Convolutional Neural Networks (CNN) (LeCun et al., 2004) simply mimic the roles of simple and complex cells, and fuses feature extraction and classification phases. The significant characteristic of CNN is its invariance and stability to scaling, distortion, disturbance, and shift. In CNN, back propagation is used for training the structure by adjusting the weights of kernels. It propagates the output error in the last layer back to the previous layers and tunes the weights of the kernels to minimize the error. The weights are updated using (1):

$$\left(w_{np}^{(l)}\right)_{new} = \left(w_{np}^{(l)}\right)_{old} - \eta_1 \Delta w_{np}^{(l)} \tag{1}$$

where $w_{np}^{(l)}$ is the weight of the interconnection between the nth element in lth hidden-layer and the pth element in $(l-1)$th hidden-layer, $\Delta w_{np}^{(l)}$, is the weight update corresponding to $w_{np}^{(l)}$ and η_1 is the learning rate. $\Delta w_{np}^{(l)}$, is calculated for each layer (l) as:

$$\Delta w_{np}^{(l)} = \frac{\partial E}{\partial w_{np}^{(l)}} = \delta_n^{(l)} h_p^{(l-1)} \tag{2}$$



$$\textbf{Conventional CNN:} \quad \Delta w_{np}^{(l)} = \frac{\partial E}{\partial w_{np}^{(l)}} = \delta_n^{(l)} h_p^{(l-1)}$$

$$\textbf{Sensitivity CNN:} \quad \Delta w_{np}^{(l)} = \frac{\partial \tilde{E}}{\partial w_{np}^{(l)}} = \left\{\delta_n^{(l)} + \frac{\eta_2 E \ddot{f}\left(\sum_p w_{np}^{(l)} h_p^{(l-1)}\right)}{\dot{f}\left(\sum_p w_{np}^{(l)} h_p^{(l-1)}\right)^2}\right\} h_p^{(l-1)}$$

Fig. 2. CNN represented by added sensitivity term in cost function of the training algorithm.

where,

$$E = \frac{1}{2MN_i} \sum_{s=1}^{M} \sum_i \left(t_i^s - y_i^s\right)^2 \tag{3}$$

where $\delta_n^{(l)}$ is the error coming from last layers to the lth layer, $h_p^{(l-1)}$ is the input feature maps for lth layer. E, represents the standard output error and M is the number of training samples. N_i indicates the number of output neurons at the output layer. M and N_i normalize the error rate. t_i^s, y_i^s are target and actual output values of the ith output neuron for the s^{th} stored pattern, respectively.

In conventional CNN, the objective function used to tune the weights of kernels during the learning process is based on (3). It must be noted that the derivative of error of last layer ($\triangle E$) with regard to weights of that specific layer needs to be considered for calculating the error back propagation for each layer other than the last layer.

To distinguish the input samples and generalize the problem, the input-output mapping sensitivity is considered to generate more precise response to input patterns containing slight changes. The sensitivity of input feature variations mapping into the output is incorporated in the typical error back propagation-learning algorithm by adding a high pass filter to the structure Jeong, Jung, Kim, Shim, & Lee, 2011; Jung, Kim, Ban, Hwang, & Lee, 2012).

The mapping sensitivity of the structure shown in Fig. 2 is represented by ((4). In Fig. 2, x_m represents the mth neuron of the input layer (x), y_i indicated the ith neuron of the output layer (y), and h_j represents the jth neuron of the h-hidden layer. $FC2$, is the 2nd fully-connected Multi-Layer Perceptron layer. *layer*2 represents 2nd convolutional layer. η_2 is the sensitivity coefficient. f,$\dot{f}$ and $\ddot{f}$ are the activation function and its first and second derivatives, respectively. Based on (4) by increasing the values of $\dot{f}$ for each layer the gradients increase and subsequently the sensitivity is enhanced highlighting the small changes in the input patterns.

$$\begin{aligned}\frac{\partial y_i}{\partial x_m} &= \frac{\partial y_i}{\partial h_j}\frac{\partial h_j}{\partial h_k}\frac{\partial h_k}{\partial h_l}\frac{\partial h_l}{\partial x_m} \\ &= w_{ij}{}^{(FC2)} \dot{f}\left(\sum_j w_{ij}^{(FC2)} h_j\right) w_{jk}{}^{(FC1)} \\ &\times \dot{f}\left(\sum_k w_{jk}^{(FC1)} h_k\right) w_{kl}{}^{(Layer2)} \dot{f}\left(\sum_l w_{kl}^{(Layer2)} h_l\right) w_{lm}{}^{(Layer1)}\end{aligned} \tag{4}$$

Since the interconnection weights (i.e. $w_{ij}{}^{(FC2)}$, $w_{jk}{}^{(FC1)}$ and so on) in (4) can become negative during training, they cannot be embodied in the sensitivity term because the objective function output value should be positive semi-definite. Therefore, only $\dot{f}$ component in (4) is considered in proposed cost function shown in (5) in which the constraint term in denominator adds the sensitivity. The weight changes in each layer using the proposed cost function are calculated in (6) to (9) and specifically shows the impact of sensitivity approach on each layer empirically. The generic form of the output error including sensitivity is (5).

$$\tilde{E} = E/(\dot{f}(\sum_p w_{np} h_p) + \varepsilon) \tag{5}$$

$\tilde{E}$ is the new cost function, E is the conventional cost function, $\dot{f}(\sum_p w_{np} h_p)$ is a generic term representing the derivative of the output of each hidden node with regard to their weights. The constant value of ε is added to prevent the term in (5) from becoming infinite when the first derivative is zero and it is considered as $\varepsilon = 10^{-8}$.

Considering the new cost function to update weights of the kernels based on (1), calculation of derivative of error with regard to corresponding weights of each layer is needed. To consider the sensitivity for each layer according to (5), the corresponding derivative of the output of each hidden node should be considered. Eqs. (6)–(9) show the terms in each layer, which needed to be considered while updating the weights of the layers using the error values from last layers to the first input layer in the chain rule. The generic definition of tuning the weights considering added sensitivity term for any layer defined in (6)–(9) is illustrated in (10). Since the CNN is a hierarchical structure, the error is obtained from last layer and is distributed to the inner layers following the chain rule. While taking derivative of error with respect to a particular layer, the weights of other layers are considered to be constant so the sensitivity term, i.e. $(\frac{E\,\ddot{f}}{(\dot{f})^2})$ has specific added value for each layer.

$$\Delta w_{ij}^{(FC2)} = \frac{\partial \tilde{E}}{\partial w_{ij}^{(FC2)}} \approx -\left\{ \sum_i (t_i - y_i) - \frac{\eta_2 E\ \ddot{f}\left(\sum_j w_{ij}^{(FC2)} h_j\right)}{\dot{f}\left(\sum_j w_{ij}^{(FC2)} h_j\right)^2} \right\} h_j \tag{6}$$

$$\Delta w_{jk}^{(FC1)} = \frac{\partial \tilde{E}}{\partial w_{jk}^{(FC1)}} \approx -\left\{ \sum_i (t_i - y_i)\ w_{ij}{}^{(FC2)} \dot{f}\left(\sum_j w_{ij}^{(FC2)} h_j\right) - \frac{\eta_2 E \ddot{f}\left(\sum_k w_{jk}^{(FC1)} h_k\right)}{\dot{f}\left(\sum_k w_{jk}^{(FC1)} h_k\right)^2} \right\} h_k \tag{7}$$

$$\Delta w_{kl}^{(layer2)} = \frac{\partial \tilde{E}}{\partial w_{kl}^{(layer2)}} \approx -\left\{ \sum_i (t_i - y_i) w_{ij}{}^{(FC2)} \dot{f}\left(\sum_j w_{ij}^{(FC2)} h_j\right)\ w_{jk}{}^{(FC1)} \dot{f}\left(\sum_k w_{jk}^{(FC1)} h_k\right) - \frac{\eta_2 E \ddot{f}\left(\sum_l w_{kl}^{(layer2)} h_l\right)}{\dot{f}\left(\sum_l w_{kl}^{(layer2)} h_l\right)^2} \right\} h_l \tag{8}$$

$$\Delta w_{lm}^{(layer1)} = \frac{\partial \tilde{E}}{\partial w_{lm}^{(layer1)}} \approx -\left\{ \sum_i (t_i - y_i) w_{ij}{}^{(FC2)} \dot{f}\left(\sum_j w_{ij}^{(FC2)} h_j\right)\ w_{jk}{}^{(FC1)} \dot{f}\left(\sum_k w_{jk}^{(FC1)} h_k\right) w_{kl}{}^{(layer2)} \dot{f}\left(\sum_l w_{kl}^{(layer2)} h_l\right) - \frac{\eta_2 E \ddot{f}\left(\sum_l w_{lm}^{(layer2)} x_m\right)}{\dot{f}\left(\sum_m w_{lm}^{(layer2)} x_m\right)^2} \right\} x_m \tag{9}$$

$$\Delta w_{np}^{(l)} = \frac{\partial \tilde{E}}{\partial w_{np}^{(l)}} = \left\{ \delta_n^{(l)} + \frac{\eta_2 E \ddot{f}\left(\sum_p w_{np}^{(l)}\ h_p^{(l-1)}\right)}{\dot{f}\left(\sum_p w_{np}^{(l)}\ h_p^{(l-1)}\right)^2} \right\} h_p^{(l-1)} \tag{10}$$

where $\triangle w_{np}^{(l)}$ is the general definition of the derivative of error with regard to weights between nth neuron in layer lth and pth neuron in layer $(l-1)$th layer, $f(\sum_p w_{np}^{(l)} h_p)$ represents the output of each hidden node. η_2 is the coefficient of sensitivity term determining the significance of sensitivity value in changing the gradients. Based on (1), the new weight tuning equation is (11).

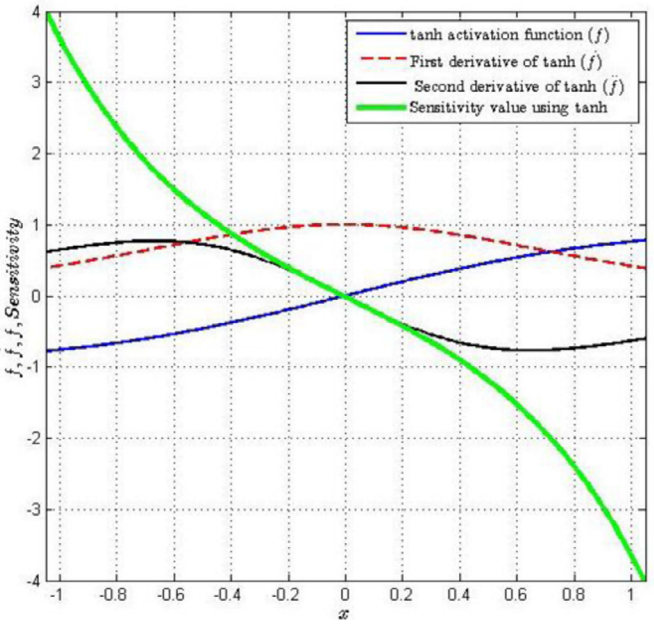


Fig. 3. The bipolar hyperbolic-tangent Sigmoid function f(x) = tanh(x) (f represented by blue line), its first derivative ($\dot{f}$ represented by red line), its second derivative ($\ddot{f}$ represented by black line), and the sensitivity value (represented by green line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\left(w_{np}^{(l)}\right)_{new} = \left(w_{np}^{(l)}\right)_{old} - \eta_1 \left\{ \delta_n^{(l)} + \frac{\eta_2 E \ddot{f}\left(\sum_p w_{np}^{(l)}\ h_p^{(l-1)}\right)}{\dot{f}\left(\sum_p w_{np}^{(l)}\ h_p^{(l-1)}\right)^2} \right\} h_p^{(l-1)} \tag{11}$$

As it is shown in (6)–(9), the error value $(t_i - y_i)$ is obtained from the last layer and to obtain $\triangle w$ in each layer it is multiplied with the W and $\dot{f}$ of the previous layers. This procedure is the same as the conventional error back-propagation. But, there is an added term in each equation, i.e. $(\frac{E\,\ddot{f}}{(\dot{f})^2})$ which represents the sensitivity term. The sensitivity term for each layer comes from the corresponding first and second derivatives of that output hidden layer. The location of the weights in each layer of the CNN structure is depicted in Fig. 2.

In Fig. 3, at the linear band of activation function where the decision boundary is located, the gradient rate is high. By forcing hidden-neural activations to reside nearby decision boundaries, high gradients are assigned to change the structure weights. The process of moving activations to the high gradient area to enhance sensitivity and to further highlight small changes in the input samples is performed by the proposed approach. Due to proportionate relationship of the gradient with sensitivity based on (4), the sensitivity term is utilized in (11). The added sensitivity term to the output error in (11) changes the weights and it may contribute to the internal representation and further generalization ability. There are two term in (11) that contribute to weight changes. One is the output error and the other one is the sensitivity term. The sum of these terms according to Fig. 3 could push the hidden-neural activations far from the saturated area of the activation function. The boundary area covers slight changes of input data that makes difference in the classification performance.

By adding the sensitivity value, i.e.,$(\frac{E\ \ddot{f}}{(\dot{f})^2})h_n$ the data samples are appear more on the middle band of the activation function which has higher gradient. When the activations are located on the saturated part of the activation function, with the aid of the sensitivity term they could be moved to the middle high gradient part. This causes high changes in the output values for a small variation in the input values. This highlights the edges and high frequency parts better. In this way, the small variations in the input images are giving more priority during the training process. In addition, the incorporation of the sensitivity term considers a range of values around a particular weight vector resulting in better generalization compared to the conventional method which only considers the particular weight vector.

The cost function that measures the loss value is another factor in training algorithm. For example, in orderto calculate $\triangle W_{kl}$ using Mean Squared Error (MSE) based on (3), the value of $\triangle W_{kl}$ is (11).

$$\Delta W_{kl} = \eta(t_i - y_i)\dot{f}\left(\sum_j w_{ij}h_j\right)w_{ij}\dot{f}\left(\sum_k w_{jk}h_k\right)w_{jk}\dot{f}\left(\sum_l w_{kl}h_l\right)h_l \tag{11a}$$

By considering the Cross Entropy error value according to (12), the $\triangle W_{kl}$ value is (13).

$$E_o = -\frac{1}{N_o}\sum_i t_i \log y_i \tag{12}$$

$$\Delta W_{kl} = \eta\frac{t_i}{y_i}\ \dot{f}\left(\sum_j w_{ij}h_j\right)w_{ij}\dot{f}\left(\sum_k w_{jk}h_k\right)w_{jk}\dot{f}\left(\sum_l w_{kl}h_l\right)h_l \tag{13}$$

So by using different cost functions, the way of assigning values to weights is different.

In literature, it is common to use rectified linear unit (ReLU) activation function instead of Sigmoid activation function in (14). Since the derivative term $\dot{f}()$ can be converted into identity function which transfers input to output with less computational effort. With ReLU activation function in CNN, we cannot employ the sensitivity term because the second derivative of ReLU function is zero. Instead, we apply the approximate function of ReLU called Softplus in (15) which has second derivative. The sensitivity term by utilizing Sigmoid and Softplus activation functions gets different values. In other words, by using different activation functions, the sensitivity term, i.e.,$(\frac{E\ \ddot{f}}{(\dot{f})^2})h_n$ might get simpler. In (16) and (17) the sensitivity terms are shown with Sigmoid and Softplus activation functions, respectively.

$$\text{Sigmoid function}: f_1(x) = \frac{1}{1+e^{-x}},\ \dot{f}_1 = f_1(1-f_1),\quad \ddot{f}_1 = f_1(1-3f_1+2f_1^2) \tag{14}$$

$$\text{Softplus function}: f_2(x) = \ln(1+e^x),\ \dot{f}_2(x) = \frac{1}{1+e^{-x}},\quad \ddot{f}_2 = \dot{f}_2\left(1-\dot{f}_2\right) \tag{15}$$

$$\text{Sensitivity term with Sigmoid function}: S = \frac{\mathrm{E}(1-2f_1)}{f_1(1-f_1)+\varepsilon} \tag{16}$$

$$\text{Sensitivity term with Softplus function}: s = \frac{\mathrm{E}\left(1-\dot{f}_2\right)}{\dot{f}_2+\varepsilon} \tag{17}$$

Fig. 4 illustrates the Softplus activation function, its first derivative, second derivative, and its corresponding sensitivity term in (17). In Fig. 4, when the activations reside on the low gradient part of the Softplus activation function, they could be forced to the higher gradient locations with the aid of added sensitivity term in (11). In this way, more weights are given to low gradient locations resulting in higher output changes and improving the internal representations. The added term not only incorporates robustness but also highlights the small variations of the input images. It is noticed that the amplitude values of the sensitivity term are multiple times bigger than the activation function values and their derivatives. So, the scaling factor is used to normalize and balance the effect of the sensitivity term on the output error and further tuning of the weights. On the other hand, when the gradient is high, the sensitivity value is proportionately low. Then by considering the scaling factor, a small sensitivity value is assigned to the output error for its weight tuning based on (11) and Fig. 4.

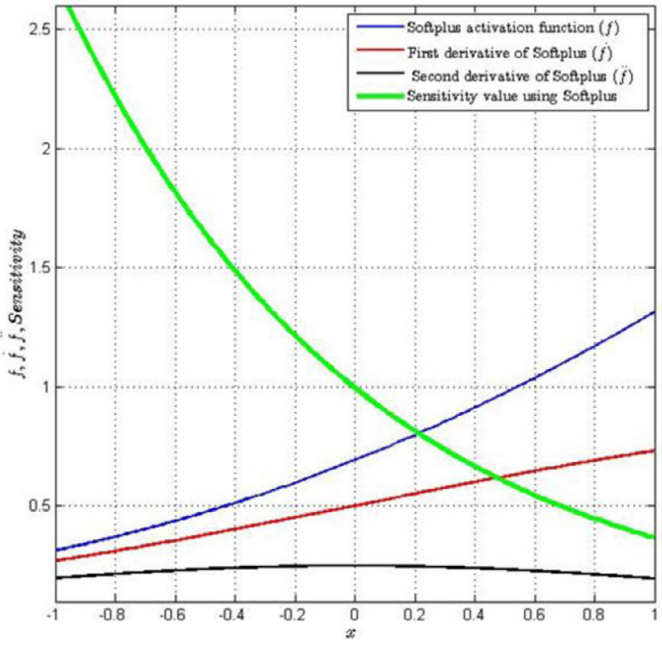


Fig. 4. Softplus activation function (f represented by blue line), its first derivative ($\dot{f}$ represented by red line), its second derivative ($\ddot{f}$ represented by black line), and the sensitivity value using Softplus (illustrated by green color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The modification that we are proposing is the incorporation of sensitivity term in cost function of different CNN structures to highlight the small intra-class and inter-class variations for better feature representation. The reason for employing Softplus instead of ReLU is because second derivative of ReLU is zero and the second derivative value of activation function is required in calculation of sensitivity term. Softplus function is an approximate of ReLU function which has second derivative values.

4. Experimental results

The experiments were performed to compare the performance of the Convolutional Neural Networks with the proposed sensitivity with selected algorithms such as linear PCA accompanied with SVM classifier (Chang & Lin, 2011), nonlinear PCA (NLPCA) introduced by Scholz (2012) with NN classifier, and LeNet (LeCun et al., 1998). It is also compared with AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) and VggNet (Simonyan & Zisserman, 2015). The open-source MatConvNet (Vedaldi & Lenc, 2015) implementation in MATLAB is used to incorporate the sensitivity term into the deep AlexNet and VggNet structures. It supports both CPU and GPU computations. MatConvNet provides a wrapper that invokes CNN

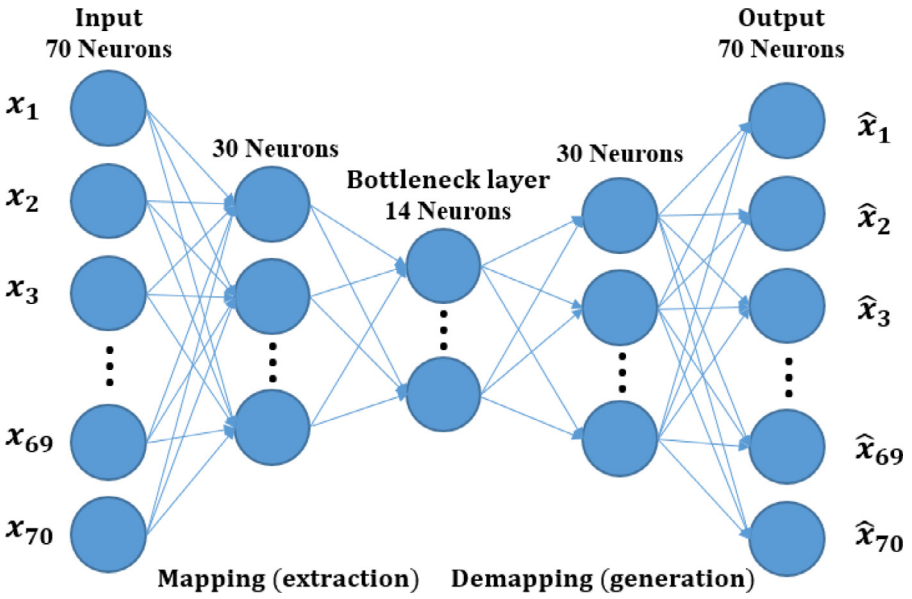


Fig. 5. The structure of Nonlinear PCA (NLPCA).

with linear topology (linear chain of operation blocks) and nonlinear Directed Acyclic Graph (DAG) topology. The DAG structure has functions to handle any arbitrary number of inputs and outputs; variables and network's parameters can have one or several outputs.

Nonlinear Principal Component Analysis (NLPCA) is a nonlinear generalization of standard principal component analysis (PCA). Thus, the subspace in the original data space, which is described by all nonlinear components, is also curved. NLPCA can be obtained by using a neural network with an auto encoder architecture also known as bottleneck type network (Scholz, 2012). NLPCA performs an identity mapping. The output has to be equal to the input, by minimizing the square error. However, the layer in the middle of the network acts as a bottleneck in which a reduction of the dimension of the data is enforced. This bottleneck-layer provides the desired component values. The training of NLPCA is conducted by Conjugate Gradient Descend (CGD). Fig. 5 shows the structure of NLPCA.

To evaluate, we considered the LDHF database that comprises of total eight hundred near infrared and visible image samples at 1 m, 60 m, 100 m, and 150 m indoor and outdoor images of 100 different subjects (Kang et al., 2014). In LDHF database, the cropped face samples are of different sizes in both width and length. By resizing samples to unique size, the system can be trained better with less number of parameters. Thus the samples were converted into the same input size and all the images were imported to proposed methods. The error cost function is cross-entropy. The activation function is ReLU. When the sensitivity term is added the ReLU is changed into Softplus because ReLU function has zero second derivative and in sensitivity definition, the nonzero second derivative is needed. The performance of LeNet, AlexNet, and VggNet without using sensitivity term considering either ReLU or Softplus activation functions is the same.

4.1. Insufficient data issue and data augmentation

In order to use CNN, the number of samples should be sufficient enough to guarantee the performance of the structure. So, in the LDHF dataset each single sample at distance 1 m, 60 m, 100 m, and 150 m are augmented to generate 5 samples. Therefore, we have 6 samples at each distance. In other words, total 24 samples

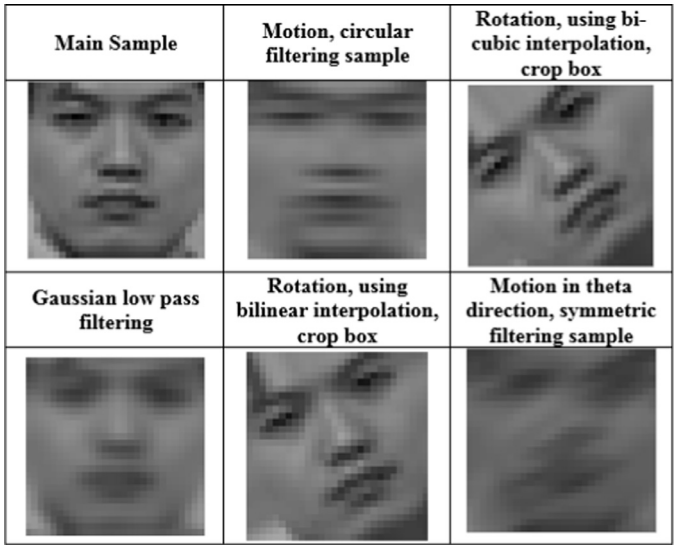


Fig. 6. The sample images for NIR at distance of 60 m for one subject generated by different operation and extracted from the main sample for augmentation.

for each subject for each modality (night-time or day-time) are prepared. The augmentation process employed to generate samples is summarized in Table 1. The sample images for NIR at distance of 60 m are shown in Fig. 6. In LDHF dataset, there are 100 subjects and 24 samples per subject for each visible (VIS) and near infrared (NIR) summed up to 4800 samples.

The experiments were performed by using 6 samples per each image for each modality. The comparison studies were conducted by PCA-SVM, NLPCA-NN, LeNet, AlexNet, and VggNet with and without sensitivity term. Data augmentation contributes in proper recognition of blurred dataset at long distances for both night-time and day-time. To evaluate the performance of each methods, the test part includes one-third (33%) of the total samples, while training includes two-third (67%) of the samples. All the samples were chosen randomly. Table 2 shows the number of samples for the experiment setting.

Table 1
The augmentation operations used to generate new samples.

Augmented samples from the original sample	
Motion exposure sample with circular filtering	The linear motion of a camera by (9×9) pixels with circular filtering. Circular filtering: Input array values outside the bounds of the array are computed by implicitly assuming the input array is periodic.
Rotate image using the bi-cubic interpolation and crop box	Rotate image by angle of 40° in a counterclockwise direction around its center point. Setting the values of output pixels that are outside the rotated image to zero using bi-cubic interpolation.
Gaussian low pass filtering	Gaussian low pass filter of size (7×7) with standard deviation SIGMA = 1
Rotate image using the bilinear interpolation and crop box	Rotate image by angle of 25° in a counterclockwise direction around its center point. Setting the values of output pixels that are outside the rotated image to zero using the bilinear interpolation.
Motion exposure sample with symmetric filtering in theta direction	The linear motion of a camera by (9×9) pixels with symmetric filtering and angle of 30° (theta) in a counterclockwise direction by assigning the darkness on the intensities with coefficient of 0.85.

Table 2
The number of samples for each experimental setup.

Distance	One distance	All four distances
Number of samples per class	Total: 6	Total:24
Number of visible images (VIS)	Total: 600	Total: 2400
	Train: 400	Train: 1600
	Test: 200	Test: 800
Number of near infrared images (NIR)		
VIS and NIR together	Total: 1200	Total: 4800
	Train: 800	Train: 3200
	Test: 400	Test: 1600

4.2. Comparison studies

The experiments were accomplished using PCA-SVM for each scenario of VIS/VIS and NIR/NIR, and their combinations. The average and standard deviation values obtained for each experiment using 3-fold cross validation is shown in Tables 4–8. Using the first 70 principle components obtained using linear PCA, the experiment is repeated multiple times with different combinations of SVM parameter values P_G (Gamma value of kernel function) and P_C (cost parameter of C-SVC) to find the suitable SVM parameters (Chang & Lin, 2011). From the experimentation, we found out the suitable range of parameters to be in the range of $[5 \times 10^{-7}$ to $10^{-6}]$ for P_G and [200 to 500] for P_C as reported in Table 3.

In the next phase to train the NLPCA/NN, we employed 3000 iterations. The experiments were performed by pre-PCA followed by NLPCA to get the nonlinear components and classified by NN. To get the first component from pre-PCA, the images of size 32×32 are converted into a vector of size 1024 and the most dominant Eigen values were extracted. In NLPCA, the number of neurons in the input layer and output layer is set to be 70 after experimenting with different values in the range [20-200]. The number of neurons in the 3 hidden layers is set to be [30, 14, 30] after trial and error search within the range [5-50] (Scholz, 2012). Similarly, the number of hidden neurons in NN classifier is set to be [150, 50] by trial and error search. The training time of NLPCA is longer than the linear PCA. Table 3 summarizes the different parameter values used in different algorithms reported in the current work.

LeNet, AlexNet, and VggNet use GPU; employ DAG structure with either ReLU or Softplus activation functions; use cross-entropy loss function; use Xavier initialization method; max-pooling is used for subsampling; test procedure was performed in 3-fold cross-validation to get the average and standard deviation for each experiment. The input size to the AlexNet and VggNet structures are modified by padding the input layer. Therefore, the images of 32x32 can be imported instead of normal input sizes of 227×227 for AlexNet and 224×224 for VggNet. The zero padding size for AlexNet is 20 pixels for each side and for VggNet structure is 10 pixels. The comparison studies are performed for AlexNet and VggNet with and without sensitivity term considering Dropout regularization method. Dropout is used for AlexNet in second FC layer and for VggNet in first and second FC layers.

In literature, researchers did not introduce any regularization method to make the deep structure more sensitive to images with high intra-class variance and low inter-class variance in datasets consisting very few data samples with high complexities. In the first scenario, we aim to observe the performance improvement obtained using the sensitivity term in the cost function for each specific distances (1 m, 60 m, 100 m, 150 m) at daytime (Table 4) and at night-time (Table 5). The proposed sensitive training algorithm considers the small variations in the image pixels leading to better internal representations. The incorporated sensitivity term changes the neural activations of hidden layers to locate at high gradient region of activation function. Therefore, it modifies the neural weights to highlight the edges and high frequency variations in the images. Table 4 shows the accuracy results for different methods when both train and test samples are visible images (VIS) at different distances. The recognition performance of images that are degraded, blurred, and with low illumination obtained from a specific distance. While Table 5 represents the results when both the train and test samples are near infrared image (NIR). The structure of the VggNet is the 16-layers weights denoted by ConvNet Configuration 'D' in corresponding paper. The experiments denote the superiority of the proposed method comparing with other methods. The improved sensitivity term in the training algorithm of CNN structures perform better internal feature representation and highlight the small variations in images.

Secondly, we aim to observe the recognition performance of images of one modality (either VIS or NIR) for a particular distance compared with a pool of other distances in which the complexity of the dataset increases. Table 6 represents the day-time distance recognition and Table 7 demonstrates the night-time distance recognition and interpolation. In this scenario, images from one distance are selected for test process and images from other standoffs are integrated for training process. We integrated the images from other distances for training process to make the dataset bigger as the dataset has very small number of samples. Moreover, integrating samples makes it more complicated due to high difference between the samples as the distance increases. This scenario illustrates how the proposed approach may influence on the performance by increasing both complexity and data samples.

Table 6 demonstrates an experiment in which both train and test samples are visible images. Therefore, the test was held for visible images in training dataset versus the visible images in test dataset (VIS vs. VIS). In this experiment, images from a particular distance are compared to an integration of the other distances to perform the interpolation and distance recognition task. For example, the third column in Table 6 represents that the CNN structures

Table 3
The parameters used in the experimental setup.

Methods	Parameters
PCA/SVM	SVM type: C-SVC Kernel type: Radial Basis Function Gamma value of kernel function: P_G= P_G=[5×10^{-7},8×10^{-7}, 10^{-6}] Cost parameter of C-SVC:P_C= [200,300,400,500] Employed 70 principle components out of 1024 image input vector using PCA Other parameters are taken from Chang and Lin (2011)
NLPCA/NN	Number of Neurons in Auto encoder:[70, 30, 14, 30, 70] Number of extracted nonlinear components: 14 NN classifier neurons: [150, 50] Iteration: 3000 Pre-PCA: yes Weight-decay: 10^{-3}
LeNet AlexNet VggNet	Learning rate: $\eta_1=10^{-4}$ Sensitivity coefficient: $\eta_2=3\times10^{-3}$ Sensitivity constant value:$\varepsilon=10^{-8}$ Weight-decay: 3×10^{-4} Number of epochs: 200 Momentum: 0.9 Batch-size: 50 samples

Table 4
Intra-distance VIS vs. VIS test accuracy performance of different methods in different standoffs on LDHF dataset.

Methods for VIS vs. VIS	Number of samples	1 m distance (%)	60 m distance (%)	100 m distance (%)	150 m distance (%)
PCA-SVM	Total samples:	57 ± 3	57 ± 3	54 ± 3	54 ± 3
NLPCA-NN	600 Train:400	55 ± 2	55 ± 2	53 ± 2	53 ± 2
LeNet	Test:200	85 ± 1	82 ± 1	92 ± 2	86 ± 2
LeNet + Sensitivity		88 ± 1	83 ± 1	93 ± 2	87 ± 2
AlexNet		90 ± 3	78 ± 2	91 ± 1	81 ± 2
AlexNet + Sensitivity		91 ± 3	81 ± 2	92 ± 1	82 ± 2
AlexNet + Dropout		91 ± 2	85 ± 2	95 ± 1	89 ± 2
AlexNet + Sensitivity + Dropout		93 ± 2	89 ± 2	96 ± 1	91 ± 2
VggNet+ Dropout		94 ± 1	89 ± 1	95 ± 1	89 ± 1
VggNet + Dropout +Sensitivity		96 ± 1	91 ± 1	96 ± 1	92 ± 1

Table 5
Intra-distance NIR vs. NIR test accuracy performance of different methods in different standoffs on LDHF dataset.

Methods for NIR vs. NIR	Number of samples	1 m distance (%)	60 m distance (%)	100 m distance (%)	150 m distance (%)
PCA-SVM	Total samples:	55 ± 3	55 ± 3	53 ± 3	53 ± 3
NLPCA-NN	600 Train:400	53 ± 2	52 ± 3	50 ± 2	50 ± 3
LeNet	Test:200	76 ± 2	76 ± 2	78 ± 1	90 ± 2
LeNet + Sensitivity		78 ± 2	79 ± 1	80 ± 1	92 ± 1
AlexNet		89 ± 1	90 ± 2	84 ± 1	79 ± 1
AlexNet + Sensitivity		90 ± 1	91 ± 2	86 ± 2	81 ± 2
AlexNet + Dropout		95 ± 1	94 ± 1	85 ± 1	85 ± 2
AlexNet + Sensitivity + Dropout		97 ± 1	95 ± 1	87 ± 1	88 ± 1
VggNet + Dropout		95 ± 1	94 ± 1	90 ± 1	92 ± 1
VggNet + Dropout + Sensitivity		97 ± 1	96 ± 2	91 ± 1	93 ± 1

are trained with images from 1 m, 100 m, and 150 m and later is tested on images of 60 m distance. Table 7 is the same process as Table 6 but the performance of the proposed method is evaluated on NIR vs. NIR in which the train and test samples are all near infrared images with different distances.

Finally, the recognition performance by combining images from all distances and both modalities is investigated Table 8) in which images of different modalities and distances are all mixed together to evaluate the performance of proposed sensitivity term in training of the deep-CNN. We performed the evaluation on low-illumination and highly-blurred face images at large standoffs with very few number of samples in both night-time and day-time modalities. The test and train images were randomly chosen from both modalities (VIS and NIR) at different distances. This experiment shows the performance of methods when train data are from both VIS and NIR and the test data also are from VIS and NIR dataset at different distances. This experiment tests the recognition ability of the system regardless of image modality and image distance. From the results the superiority of the proposed sensitivity regularization for better generalization is apparent. The sensitivity term regularizes the neural activations of hidden layers in the structures. It considers the small intra-class and inter-class variations in the image pixels and highlights the edges for better generalization.

During every run, the test and training data obtained by the random split is given to each algorithm employed for comparison. This allows a fair comparison between the algorithms as all the algorithms are given the same training data. Therefore, it would be easy to test the feature extraction capability of different algorithms.

As described in ((11), the incorporation of the sensitivity term introduces the parameter η_2 referred to as sensitivity coeffi-

Table 6
Distance interpolation performance for visible images.

Methods for VIS vs. VIS	Number of samples	Train:1 m, 100 m, 150 m Test:60 m	Train:1 m, 60 m, 150 m Test:100 m	Train:1 m, 60 m, 100 m Test:150 m
LeNet	Total samples:	84 ± 2	91 ± 2	70 ± 2
LeNet + Sensitivity	2400 Train:	86 ± 2	92 ± 2	71 ± 2
AlexNet	1800 Test:600	77 ± 2	87 ± 2	74 ± 2
AlexNet + Sensitivity		79 ± 2	89 ± 2	76 ± 2
VggNet + Dropout		85 ± 2	93 ± 2	83 ± 2
VggNet + Dropout + Sensitivity		88 ± 2	95 ± 2	85 ± 2

Table 7
Distance interpolation performance for near infrared images.

Methods for NIR vs. NIR	Number of samples	Train:1 m, 100 m, 150 m Test:60 m	Train:1 m, 60 m, 150 m Test:100 m	Train:1 m, 60 m, 100 m Test:150 m
LeNet	Total samples:	46 ± 3	69 ± 2	37 ± 3
LeNet + Sensitivity	2400 Train:	48 ± 3	70 ± 2	39 ± 3
AlexNet	1800 Test:600	44 ± 3	70 ± 3	42 ± 3
AlexNet + Sensitivity		47 ± 3	74 ± 3	43 ± 3
VggNet + Dropout		48 ± 3	69 ± 3	42 ± 3
VggNet + Dropout + Sensitivity		50 ± 3	75 ± 3	45 ± 3

Table 8
Cross modality test error of different methods on LDHF dataset.

Methods	Number of samples	VIS_NIR vs. VIS_NIR (%)
PCA-SVM	Total samples:	67.5 ± 0.75
NLPCA-NN	4800	60.5 ± 0.75
LeNet	Train:3200	89.5 ± 0.50
LeNet + Sensitivity	Test: 1600	90.0 ± 0.50
AlexNet		95.5 ± 0.75
AlexNet + Sensitivity		96.0 ± 0.50
AlexNet + Dropout		93.0 ± 0.75
AlexNet + Sensitivity + Dropout		95.0 ± 0.50
VggNet+ Dropout		92.0 ± 0.50
VggNet + Dropout + Sensitivity		93.0 ± 0.50

cient. The sensitivity term η_2 has significant effect on the weight changes and consequently on the performance. The change in the performance of VGGNet and AlexNet for different values of η_2 in the range (0.007, 0.08) is summarized in Table 9. The range of the values is obtained by trial and error. We tried to find the η_2 for the best performance by a heuristic search method in a certain range. From the results, it can be observed that the best performance can be achieved by setting the η_2 to be 0.003. The values outside the range (0.007, 0.08) lead to divergence or giving the same accuracy as the original (VggNet or AlexNet) without sensitivity with the pre-defined parameters in Table 3.

The addition of the sensitivity term into the cost function of LeNet, AlexNet, and VggNet does not introduce any extra weights or biases that need to be trained in the network structure. Therefore, the proposed method does not change the order of system. The computational complexities of the structures (LeNet, AlexNet, and VggNet) remain the same even after adding the sensitivity term. The computational time is slightly higher since the added sensitivity term has some computations. The computational time to train different networks with and without sensitivity is presented in Table 10.

The hardware specifications to perform the experiments are in Table 11.

The improved performance can be attributed to the sensitivity term that can highlight the slight changes in the input which are very important for night vision. In all experiments by adding sensitivity, the performance increases. To illustrate this we compared each of the baseline LeNet, AlexNet, AlexNet-Dropout, and VggNet-Dropout methods with those including sensitivity.

Fig. 7 illustrates the night-time images at a distance of 100 m that are misclassified using AlexNet + Dropout but by incorporating the sensitivity term they are well-recognized (AlexNet + Dropout + Sensitivity). The first row contains the test samples and their corresponding labels are shown in the second row. In all depicted cases, the incorporation of sensitivity improves the classification accuracy as it considers the slight variations at the edges in the feature space.

Fig. 8 presents the night-time test samples at distance of 150 m that are misclassified using AlexNet-Dropout but well-recognized by "AlexNet + Dropout + Sensitivity". The applied sensitivity term to highly blurred, low illuminance, and noisy images highlights the high frequency and edges in the feature maps of the AlexNet structure resulting in better feature extraction and generalization. Moreover, the test samples are augmented with different motions and rotation disturbances to validate the better performance by incorporating the sensitivity term in the training part.

The AlexNet and VggNet weights are also trained with pre-trained ImageNet networks. In the first experiment, the pre-trained fully connected layers were removed and the weight of the rest layers were kept fixed as a feature extractor for new dataset and the whole network retrained using LDHF dataset. In the second experiment, all the pre-trained ImageNet weights were kept as initial weights and fine-tuned by new dataset. While using the pre-trained network, the crucial factor is the size of the new dataset. Since the LDHF dataset size is very small and its samples are different from ImageNet dataset, the best approach is to keep the feature extraction layers fixed and train the classifier at the top layer. Moreover, the smaller learning rate should be applied for classifier weights that are being fine-tuned. After all in either way, the performance is (10–15)% lower by using pre-trained network. The main reason of getting lower performance than the initialization from scratch using Xavier initialization method is that the dataset size is very small and the samples are also completely different from ImageNet dataset.

5. Concluding remarks

In this paper, a new cost function including added sensitivity term was proposed in the training structure of Convolutional

Table 9
The impact of sensitivity coefficient on performance of AlexNet and vggnet.

Value of η_2	0.0007	0.003	0.01	0.08
VggNet + Dropout + Sensitivity for VIS vs. VIS (%)	96.0 ± 0.5	97.0 ± 0.5	95.5 ± 0.5	94.0 ± 0.5
VggNet + Dropout + Sensitivity for NIR vs. NIR (%)	93.5 ± 0.5	95.0 ± 0.5	94.0 ± 0.5	93.0 ± 0.5
AlexNet + Sensitivity + Dropout for VIS_NIR vs. VIS_NIR (%)	94.5 ± 0.5	95.0 ± 0.5	94.0 ± 0.5	93.5 ± 0.5

Table 10
The computational time to train different networks with and without sensitivity term.

Method	Dataset	Without sensitivity	With sensitivity
LeNet	NIR vs NIR/ 150 m distance	253.90 s	264.77 s
LeNet	VISNIR vs. VISNIR/ All distances	395.96 s	471.94 s
AlexNet	NIR vs NIR/ 100 m distance	570.80 s	588.95 s
AlexNet	VIS vs. VIS/ All distances	790.88 s	840.02 s
VggNet	VISNIR vs. VISNIR/ All distances	2533.40 s	2818.40 s
VggNet	VIS vs. VIS/ 60 m distance	888.05 s	969.01 s

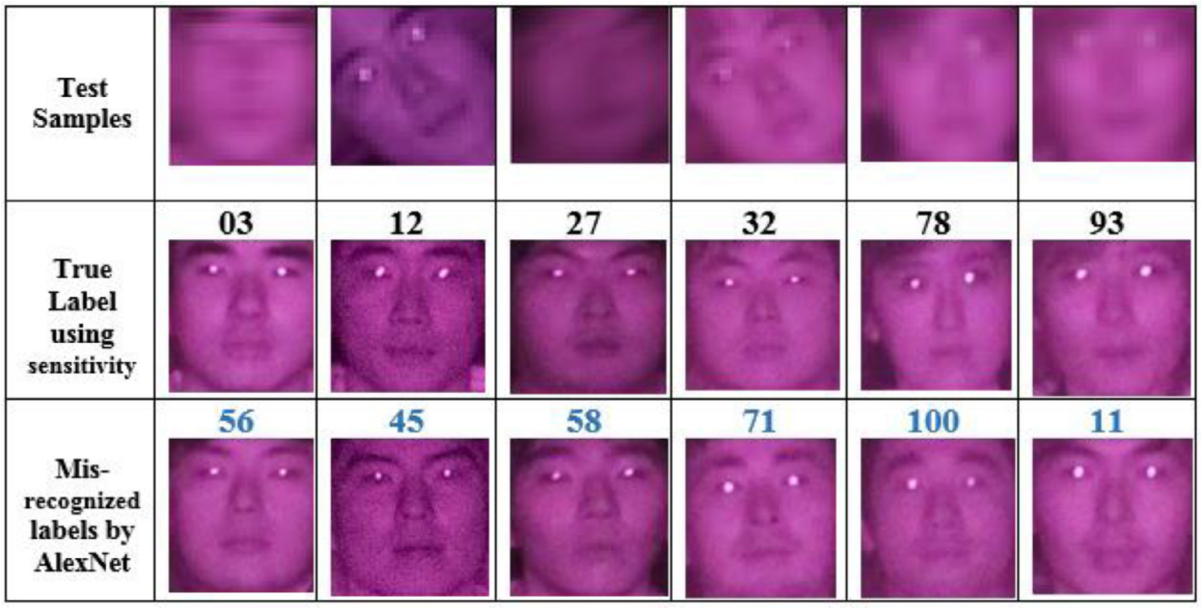


Fig. 7. The misrecognized images by AlexNet + Dropout which are well-recognized by added sensitivity term for night-time images at distance of 100 m.

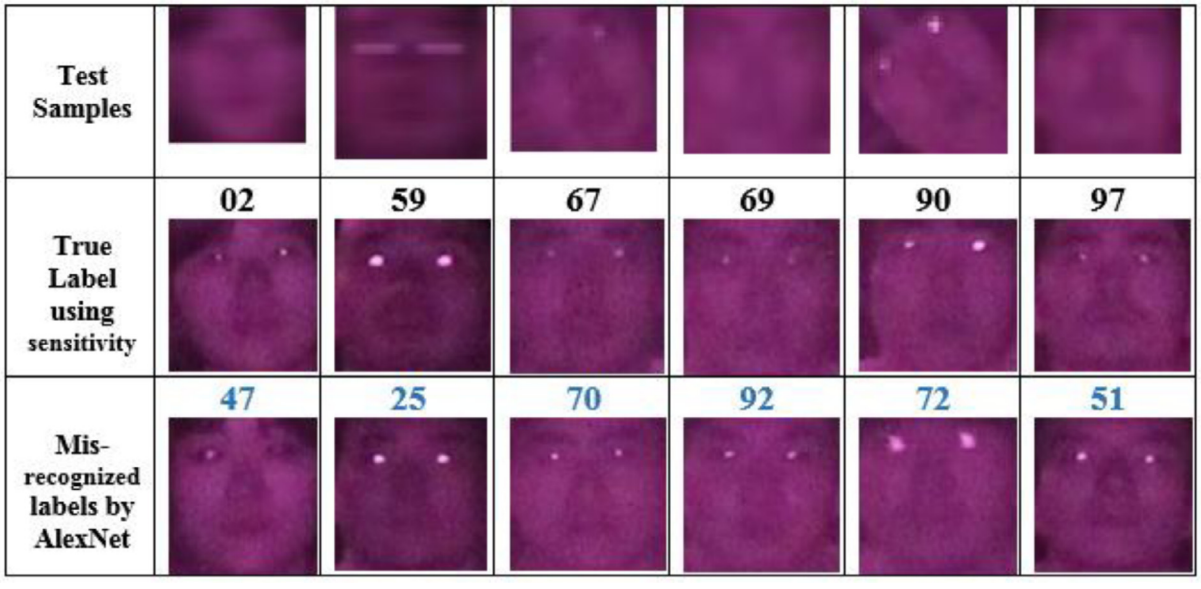


Fig. 8. The misrecognized images by AlexNet + Dropout which are well-recognized by added sensitivity term for night-time images at distance of 150 m.

Table 11
The hardware specifications of the experimental setup.

Memory: 15.6 GiB	Processor: Intel Core i7-6700k CPU @ 4.00 GHz × 8
OS type: 64-bit	Graphics: GeForce GTX TITAN X/PCIe/SSE2

Neural structures to highlight the high frequency components and small variations of the input samples. This is done by pushing the feature map activations to appear on the middle high gradient band of the activation function. To check the validity of the proposed approach, it was applied to low illuminance, quality degraded LDHF images captured in long standoffs. The presented cost function, in the context of training algorithm of Convolutional Neural structures, performs better feature extraction and further improvements in classification outcome. It creates more distinguishable characteristics, which leads to general exploitation, internal representation, and better face recognition. To evaluate the proposed method, we address the highly blurred bi-modal face recognition in long different standoffs by making the algorithm more sensitive to small intensity changes of the input sample edges.

As a further work, we intend to implement the sensitivity term accompanied by a robustness term (Jalali et al., 2015) in the training algorithm of deep structure. It is inspired from the biologically visual structure system in which the sensitivity is exploited on convolutional layer training algorithm and the robustness term is applied on subsampling layer to perform better training and subsequently better feature extraction and recognition.

Acknowledgments

This research was supported by ICT R&D program of MSIP/IITP. [R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding] (50%) and was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2016M3C1B6929647) (50%).

References

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST, 2*, 27.

Ciresan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on* (pp. 3642–3649). IEEE.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics, 36*, 193–202.

Hafemann, L. G., Oliveira, L. S., Cavalin, P. R., & Sabourin, R. (2015). Transfer learning between texture classification tasks using Convolutional Neural Networks. In *Neural networks (IJCNN), 2015 international joint conference on* (pp. 1–7). IEEE.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology, 148*, 574–591.

Jalali, A., Jang, G., Kang, J.-S., & Lee, M. (2015). Convolutional neural networks considering robustness improvement and its application to face recognition. In *Neural information processing* (pp. 240–245). Springer.

Jeong, S., Jung, C., Kim, C.-S., Shim, J. H., & Lee, M. (2011). Laser spot detection-based computer interface system using autoassociative multilayer perceptron with input-to-output mapping-sensitive error back propagation learning algorithm. *Optical Engineering, 50* 084302-084302-084311.

Jung, C., Kim, C.-S., Ban, S.-W., Hwang, I.-K., & Lee, M. (2012). Novel input and output mapping-sensitive error back propagation learning algorithm for detecting small input feature variations. *Neural Computing and Applications, 21*, 705–713.

Jung, H., Lee, S., Yim, J., Park, S., & Kim, J. (2015). Joint fine-tuning in deep neural networks for facial expression recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 2983–2991).

Kang, D., Han, H., Jain, A. K., & Lee, S.-W. (2014). Nighttime face recognition at large standoff: Cross-distance and cross-spectral matching. *Pattern Recognition, 47*, 3750–3766.

Kolář, M., Hradiš, M., & Zemčík, P. (2016). Deep learning on small datasets using online image search. In *Proceedings of the 32nd spring conference on computer graphics* (pp. 87–93). ACM.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks: Vol. 3361* (p. 1995).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*, 2278–2324.

LeCun, Y., Huang, F. J., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on: Vol. 2*. IEEE II–97–104.

Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20*, 23–38.

Scholz, M. (2012). Validation of nonlinear PCA. *Neural Processing Letters, 36*, 21–30.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815–823).

Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR: Vol. 3* (pp. 958–962).

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations*.

Song, X., Gao, X., Ding, Y., & Wang, Z. (2016). A handwritten Chinese characters recognition method based on sample set expansion and CNN. In *Systems and informatics (ICSAI), 2016 3rd international conference on* (pp. 843–849). IEEE.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Vedaldi, A., & Lenc, K. (2015). MatConvNet: Convolutional neural networks for matlab. In *Proceedings of the 23rd annual ACM conference on multimedia conference* (pp. 689–692). ACM.

Wagner, R., Thom, M., Schweiger, R., Palm, G., & Rothermel, A. (2013). Learning convolutional neural networks from few samples. In *Neural networks (IJCNN), the 2013 international joint conference on* (pp. 1–7). IEEE.

Xu, Z., Huang, S., Zhang, Y., & Tao, D. (2015). Augmenting strong supervision using web data for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision* (pp. 2524–2532).