1. Introduction

Facial expression analysis plays a significant role in understanding human emotions and behaviors. The varying facial expressions are considered to be the most important cues in the psychology of emotion [1]. Analyzing facial expressions accurately has broad applications in areas such as human behavior analysis, human–human interaction, and human–computer interaction. Automatic recognition of emotion from images or videos of human facial expression has been a challenging and actively studied problem for the past few decades [2].

Six basic facial expressions and emotions that are commonly referred include anger, surprise, disgust, sadness, happiness and fear. To quantitatively study various facial expressions, facial action coding system (FACS) has been developed [3]. In total 44 Action Units (AUs) are defined to describe all possible and visually detectable facial changes.

One of the most common expressions in a person’s daily life, smile, conveys the emotion indicating joy, happiness, satisfaction, etc. Smile expression involves two parts of facial muscle movements, namely Cheek Raiser (AU 6) and Lip Corner Puller (AU 12), as shown in Fig. 1. Detecting smile finds its applications mainly in the Human–Computer Interaction systems, such as interactive gaming, customer satisfaction rating, and patient monitoring. As a consequence, some research has been dedicated to smile detection [4–6]. Although accurate smile detection can be achieved in a laboratory controlled database, smile detection for real-world face images is still challenging due to the variations in pose, illumination and image quality. Fig. 2 shows some smile and non-smile real-world face images.

In this paper, we focus on detecting smiles from face images that contain either a smile or a non-smile facial expression. The discriminative classifier is trained in an efficient manner using Extreme Learning Machine (ELM) [7]. ELM is a recently proposed learning framework which has very low computational cost without the hassle of parameter tuning in contrast to the other learning methods. ELM has been previously applied to different tasks such as image super-resolution [8], genome analysis [9], human action recognition [10], face recognition [11], and face emotion recognition [12]. Before training the classifier, we use a standard face detector [13] to detect faces from the images. The faces are registered using a flow-based affine transformation automatically without any manual labeling or key-point detection. The trained model by ELM is then used to predict the smile status of a given face. Experiments, on a collection of laboratory controlled database and a real-world database, using various feature representations show that the proposed approach achieves high accuracy, efficiency, and better generalization performance compared to the benchmark classifiers. Compared to the state-of-the-art method, our method is competitive in detection accuracy, yet it is more efficient and does not require manual image registration.
Apart from abundant literature for facial expression recognition, a few papers are dedicated to smile detection. In [22], smile detection is achieved by using Fisher weight map (FWM) and higher-order local auto-correlation (HLAC). In [6] a real-world image collection of thousands of subjects is introduced. Comprehensive studies are conducted by examining different aspects such as size of database, effect of image registration, image representation, and classifier. Results show that human-level expression recognition accuracy can be achieved using current machine learning methods. Recently, Shan [5] proposes the usage of pixel intensity difference as features for smile detection. AdaBoost is used to choose the weak classifiers and a strong classifier is formed by combining the chosen weak classifiers. This approach achieves state-of-the-art results on a real-world smile detection database.

Besides image-based approach, some methods integrate multimodal information for smile detection. Ito et al. [23] detect smile using an image-based facial expression recognition combined with an audio-based laughter sound recognition method. In [4] audio features from the spectogram and the video features extracted by estimating the mouth movement are used in stacked sequential learning for smile detection.

For commercial applications, some software tools have been developed for smile detection, such as VDFaceSDK [24] and Omron’s smile measurement and analysis software [25].

3. Technical approach

The system pipeline of the proposed approach is shown in Fig. 3. Before face registration, the faces are extracted from the original images using a standard face detector. The detected faces are registered using a fully automated flow-based registration method without the manual labeling of key points. The facial features are then extracted from the registered faces. The ELM is used to train the smile/non-smile binary classifier and the learned model is used to predict the smile status of a given face for testing. In the following each step will be explained in detail.

3.1. Face detection

The faces from original images are extracted using Viola-Jones face detector [13] implemented in OpenCV which is suitable for real-time processing. The Viola-Jones face detector performs well and the detection accuracy is very high at low false alarm rate. The detected faces are normalized to a certain image size using the bicubic interpolation.

3.2. Face registration

Face registration/alignment normally requires the knowledge of the facial key points such as the locations of the eyes, nose, and mouth. Facial points’ localization is either performed through manual labeling or by an automated detector (e.g., [26]).
the image is noisy or of low-resolution, the key points’ detection is prone to error. In our approach, we apply a holistic flow-based face registration method to automatically align the detected faces. No facial points need to be located in this way and the registration produces the smooth output that is robust to minor out-of-plane head rotations. The face registration involves two steps: SIFT flow computation and flow-based affine transformation.

3.2.1. SIFT flow computation

SIFT flow [27] was originally designed for image alignment at the semantic level. This higher level alignment is achieved by matching local, salient and transform-invariant image structures. The SIFT flow algorithm robustly matches dense SIFT features between two images, while maintaining spatial discontinuities. The local gradient descriptor, SIFT [28], is used to extract a pixel-wise feature component. For every pixel in an image, its neighbor-based af

s. Then the features are extracted from the registered faces as input to the ELM. With the learned model, the ELM classifies the input face to be as small as possible when no other information is available. The smoothness constraint in (3) takes care of the similarity of flow vectors for adjacent pixels. In this objective function, the truncated L1 norm is used in both the data term and the smoothness term with t and d as the thresholds for matching outliers and flow discontinuities, respectively. η and α are scale factors for the small displacement and the smoothness constraint, respectively. The dual-layer loopy belief propagation is used as the base algorithm to optimize the objective function. Then, a coarse-to-fine SIFT flow matching scheme is adopted to improve the speed and the matching result.

3.2.2. Flow-based affine transformation

After the SIFT flow is computed, instead of detecting key points on the input face (e.g., eye corners, mouth center) and inferring affine transformation from them, we use the SIFT flow information to estimate an affine transformation from the input face image to the reference face. In homogeneous coordinates, we represent the pixel location of a target frame and a reference frame by \( \mathbf{p} = (x, y, 1) \) and \( \mathbf{p} = (x', y', 1) \), respectively. Given the target frame pixel location and its corresponding flow vectors, the reference frame pixel location can be written as \( x' = x + u(\mathbf{p}), y' = y + v(\mathbf{p}) \).

Thus, we can model the affine transformation for all N pixels in an image as follows:

\[
\begin{pmatrix}
    x_1 & \cdots & x_N \\
    y_1 & \cdots & y_N
\end{pmatrix} =
\begin{pmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{21} & a_{22} & a_{23} \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    x_1 & \cdots & x_N \\
    y_1 & \cdots & y_N \\
    1 & \cdots & 1
\end{pmatrix}
\]

To suppress the outliers from the flow vectors, we solve for the maximum likelihood estimates of this overdetermined system robustly by iteratively reweighted least squares (IRLS) [29]. The reference face is obtained by averaging all of the frontal faces in the CK+ database [30]. The SIFT flow based registration from an input face to the reference face is illustrated in Fig. 4. Note

![Image 1](https://via.placeholder.com/150)

**Fig. 3.** The pipeline of the proposed method for smile detection. After faces are detected from the original images, a flow-based face registration is performed to align the faces. Then the features are extracted from the registered faces as input to the ELM. With the learned model, the ELM classifier is able to predict the smile status of a given face.

![Image 2](https://via.placeholder.com/150)

**Fig. 4.** SIFT flow based face registration. The SIFT flow is first computed between the input face and the reference face. Affine estimation is used to transform the original discontinuous flow vector field to a smooth flow vector field. The aligned face using smoothed flow vector field contains much less artifact compared to the aligned face using the original flow vector field.
that the original flow vectors are discontinuous, and the corresponding warping result using the discontinuous flow vectors has strong artifact. The affine transformation makes the flow vector field continuous and the warping result using the affine transformed flow vector field is smooth without artifact. Fig. 5 shows some sample results of face registration.

3.3. Feature extraction

To extract feature representations for faces, various feature descriptors have been proposed. In the experiments, we adopt three most popular feature descriptors: Local Binary Pattern (LBP) [31], Local Phase Quantization (LPQ) [32], and Histogram of Oriented Gradients (HOG) [33]. The operation of each feature descriptor is explained in the following sections. In addition, the raw pixel intensity values are also used as a baseline feature descriptor.

3.3.1. Local Binary Pattern

Local Binary Pattern (LBP) [31] and its derivatives are among the most popular choices for facial feature representations. The basic LBP feature for a pixel is obtained by comparing its intensity value to the eight immediate neighbors and the generated LBP value is converted to an integer value between 0 and 255. The LBP descriptor for a pixel is obtained by concatenating all the histograms from different blocks. Fig. 6 shows the basic LBP descriptor.

\[ F(u, x) = \sum_{y} f(y) w(y - x)e^{-i2\pi\alpha y} \]

At each pixel position we obtain a vector

\[ F(x) = [F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)] \]

where \( u_1 = [a, 0]^T, u_2 = [0, a]^T, u_3 = [a, a]^T, \) and \( u_4 = [a, -a]^T, \) where \( a \) is a small scalar to ensure that \( H(u) > 0 \) and \( w(x) \) defines the neighborhood \( N_4 \) of size \( M \times M \). We use \( a = 1/7 \) and \( M = 7 \) in our experiments. The local Fourier coefficients are computed at four frequency points and the phase information is recorded by a binary quantizer that observes the signs of each component in \( G_x = [Re(F_x), Im(F_x)] \) where \( Re \) denotes the real part of \( F_x \) and \( Im \) denotes the imaginary part of \( F_x \). The resulting eight binary coefficients for pixel at location \( x \) are then represented as integer values between 0 and 255 using binary coding as \( f_{LPQ}(x) = \sum_{j=0}^{7} q_j(x)2^{j-1} \), where \( q_j \) is the \( j \)-th elements of \( G_x \) after binarization. In addition, a decorrelation process is added to the original LPQ implementation to eliminate the dependency among the neighboring pixels. We use the decorrelation parameter \( \rho = 0.9 \) in our experiments. The image is divided into 10 blocks and for each image block, a histogram is generated from the calculated integer values. The final feature descriptor for the entire image is obtained by concatenating all the histograms from different blocks.

3.3.3. Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) was originally proposed for human detection [36]. Recently it has been applied to face recognition [33]. HOG operates by first computing the image gradients in horizontal and vertical directions of an image. The image is then divided into blocks and for each block, the orientation for each pixel is binned into evenly divided orientation channels spreading from 0 to 180 or 360°. Similar to LBP and LPQ, the final descriptor is the concatenations of HOG histograms from each block in the image. Since the HOG is a local-region based descriptor, it tolerates some geometric and photometric variations. Fig. 7 illustrates the basic idea of HOG feature computing.

3.4. Extreme Learning Machine for smile detection

Extreme Learning Machine (ELM) was initially developed for a single-hidden-layer feed-forward neural networks (SLFNs) [37].
According to ELM theory, the hidden node parameters can be generated by learning the training data, which is contrary to the conventional learning methods. Given an input sample \((x_j, y_j)\) where \(x_j \in \mathbb{R}^d\) and \(y_j \in \mathbb{R}\). The output of ELM with \(L\) hidden nodes is designed as

\[
\sum_{i=1}^{L} \beta_j (w(x_j) + b_j) = h(x_j)\beta, \quad j \in [1, N]
\]

(7)

where \(\beta = [\beta_1, \beta_2, ..., \beta_L]^T\) is a vector consisting of the output weights between the hidden layer of \(L\) nodes and the output node. \(f\) is the activation function, \(w_i\) is the input weights to the \(i\)-th neuron, and \(b_j\) is the hidden layer bias. \(h(x_j) = [h_1(x_j), h_2(x_j), ..., h_L(x_j)]^T\) is the output of the hidden layer given input sample \(x_j\). Function \(h(x_j)\) maps the original input data space to the \(L\)-dimensional feature space.

In case the output of ELM approximates the data perfectly for all \(N\) samples, we have \(H\beta = Y\), where \(Y = (y_1, ..., y_N)^T\) and \(H\) is the hidden-layer output matrix:

\[
H = \begin{bmatrix}
    h_1(x_1) & ... & h_L(x_1) \\
    : & : & : \\
    h_1(x_N) & ... & h_L(x_N)
\end{bmatrix}
\]

(8)

According to [7], ELM aims at achieving not only the minimum training error but also the smallest norm of the output weights. Based on the theory in [38], for feed-forward neural networks reaching a smaller training error, the smaller the norms of the weights are, the better is the generalization performance that the networks tend to have. For generalized SLFNs, this may also hold [39]. Thus, in ELM the following quantities are minimized:

\[
\begin{cases}
    \text{minimize} & \|H\beta - Y\|^2 \\
    \|\beta\|
\end{cases}
\]

(9)

In the implementation, the minimal norm least square method is used instead of the standard optimization method in which only the training error is minimized and the solution is unique [37]. The input weights and bias are randomly assigned in ELM algorithm [37]. Compared to other classifiers, ELM has the following advantages [7]:

1. ELM can be applied to both multi-classification and regression problems with various types of feature mappings.
2. Compared to other classifiers such as Support Vector Machine (SVM), ELM has milder optimization constraints.
3. ELM has less computational complexity.
4. In theory ELM can approximate any continuous target function and classify any disjoint regions. Further, ELM tends to have better scalability and generalization performance.
5. The hidden-node parameters (which are supposed to be tuned in conventional learning algorithms) are not required in ELM.

The features extracted from both smile and non-smile faces are used as training data. The training data and their corresponding labels (smile and non-smile) are fed to ELM to learn the discriminative binary classifier. We use the basic version of ELM with randomly generated hidden nodes. Sigmoid function is chosen as the activation function. The only parameter to be defined is the number of hidden neurons, which is determined through cross-validation. The training is performed off-line and after training the ELM learning model is saved. For smile detection, features from a given face are extracted and ELM classifier is used to predict its smile status.

4. Experiments

In the experiments two databases are used. One database, referred as the MIX database, is generated as a collection of smile and non-smile images from several publicly available databases. A more challenging database, the GENKI-4K database [40] containing real-world smile and non-smile images, is also used for validating the proposed approach. The experimental setup and the results are reported in the following subsections.

4.1. Databases

4.1.1. MIX database

We collect a set of images called MIX database from four publicly available databases: FEI [41], Multi-PIE [42], CAS-PEAL [43], and CK+ [30]. The FEI database is a Brazilian face database containing images of participants who are between 19 and 40 years old. The Multi-PIE database contains images of 337 subjects with different poses and facial expressions. The CAS-PEAL database is a Chinese face database that involves 1042 subjects. The CK+ database is an extensive database for action unit and emotion specified facial expression. In total we select 1534 smile faces and 2035 non-smile faces from these four databases to construct the MIX database. The detected faces are normalized to 200 x 200 and converted to grayscale images. The diversity of the MIX database is addressed by the wide range of age, ethnicity, imaging condition, occlusions (e.g., glasses), etc. Some sample images from the MIX databases are shown in Fig. 8. In the experiments four-fold cross-validation is performed on this database, meaning that each time 75% images are randomly selected as training images and the rest 25% images are used for testing.

4.1.2. GENKI-4K database

The GENKI-4K database [40] contains 4000 face images of a wide range of subjects with different ages and races in real-world pictures with varying pose, illumination and imaging conditions. Among these 4000 images, 2162 images are labeled as smile and the rest 1828 images are labeled as non-smile. This database represents the real-world scenarios for smile detection which is more challenging than detecting smiles in laboratory-controlled databases. Fig. 9 shows some sample images from the GENKI-4K database.

In our experiments the detected faces from the original images are converted to grayscale images with the size normalized to 100 x 100. Four-fold cross-validation is performed on this database, meaning that each time 3000 images are randomly selected as training images and the rest 1000 images are used for testing.

Table 1 provides a summary of the aforementioned two databases.
used in the experiments. The reported results on the two databases are the averaged results of four-fold cross-validation.

4.2. Parameter settings

For LBP, LPQ and HOG, an image is divided into $10 \times 10$ blocks. In LBP, $\text{LBP}^{8,2}$ is used as suggested in [31] for face representation. Similar to [31] and [44], the parameters for LPQ are set to $M=7$, $\alpha=1/7$ and a decorrelation parameter $\rho=0.9$. For HOG, the number of orientation bins is set to 15. Besides these three feature descriptors, the raw pixel intensity values are also used as a baseline feature descriptor. The dimensionality of the extracted feature vectors is reduced to 500 using PCA such that 99.9% of the data variance is retained. For ELM, the number of hidden neurons is set to 600 by cross-validation. For SVM, the soft margin parameter $C$ is set to 10 by cross-validation.

4.3. Benchmark classifiers

The proposed method using ELM is compared with two benchmark classifiers: Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM). These classifiers are commonly used in classification tasks. For the SVM classifier, its linear version is used.

4.4. Smile detection performance

The detection accuracies for the MIX and GENKI-4K databases are shown in Tables 2 and 3, respectively. Compared to the results of LDA and SVM, ELM achieves better or similar performance. By using different features we observe the same performance trend. LPQ combined with ELM achieves the best detection accuracy of 94.6% for the MIX database and HOG+ELM gives the best result of 88.2% for the GENKI-4K database. For the MIX database, even with pixel intensity values as features, the detection rates using different classifiers are above 90% and using more advanced features brings the detection rate to nearly 95%. The high detection accuracy on the MIX database indicates that for the lab-controlled databases, the current framework works very well. For the real-world scenario database, using advanced features help improve the detection rate significantly yet the best accuracy is still lower than the laboratory-controlled database. Both SVM and ELM outperform LDA on the two databases.

The ROC curves for the MIX and GENKI-4K databases with different feature descriptors are shown in Figs. 10 and 11, respectively. The performance of ELM and SVM is quite similar at different false positive rates and both of them outperform LDA.

The Area Under Curve (AUC) for the ROC curves in Figs. 10 and 11 is given in Tables 4 and 5, respectively. As a performance measure for classifiers, AUC indicates the probability that a classifier will give a higher rank to a randomly chosen positive sample than a randomly chosen negative sample. Although overall the AUC for SVM is slightly higher than that of ELM with different feature descriptors, the AUC values for both SVM and ELM are very close. This indicates that as a binary classifier, ELM is very competitive compared to SVM.

Table 6 shows the comparison between the best result by the proposed method and the state-of-the-art result by [5] on the GENKI-4K database. In [5], pixel difference is used as the feature descriptor, and AdaBoost is used for both feature selection and classification. With the same four-fold cross-validation protocol, the detection rate by our method is very competitive compared to the result in [5]. Note that our fully automated method does not involve manual image registration, illumination normalization, or feature selection as in [5]. To classify an image, our method takes only 0.09 ms which is more than two orders of magnitude faster than the required time of 10.9 ms in [5] (both methods are in Matlab implementation).

4.5. Computational cost

The computational cost for the smile detection using ELM is compared against SVM and LDA classifiers. The required time for

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Two databases used in the experiments.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier</td>
<td>Pixel values</td>
</tr>
<tr>
<td>LDA</td>
<td>90.8</td>
</tr>
<tr>
<td>SVM</td>
<td>92.9</td>
</tr>
<tr>
<td>ELM</td>
<td>93.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Smile detection accuracy for the MIX database (in %).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier</td>
<td>Pixel values</td>
</tr>
<tr>
<td>LDA</td>
<td>74.7</td>
</tr>
<tr>
<td>SVM</td>
<td>79.4</td>
</tr>
<tr>
<td>ELM</td>
<td>79.3</td>
</tr>
</tbody>
</table>
training and prediction is shown in Table 7 for the MIX and GENKI-4K databases. The reported time is the averaged time of different runs in cross-validation. The non-optimized codes are implemented in MATLAB on a desktop with 3 GHz CPU and 8 GB of RAM. For SVM, the LIBSVM\(^1\) package is used. Compared to SVM, ELM is significantly faster with time for training and prediction reduced by a factor over 30. Note that in this case the ELM is fully implemented in MATLAB and the SVM is implemented in C++ with MATLAB interface. The time cost for LDA is comparable to ELM. However, the performance of LDA is inferior to ELM. The efficiency and the accuracy of ELM make the smile detection using ELM reliable and possible for large-scale or real-time practical applications.

4.6. Impact of number of hidden neurons

In the experiments, the only parameter to be determined for ELM is the number of hidden neurons. Figs. 12 and 13 show the detection rates with different numbers of neurons for the MIX and GENKI-4K databases. As can be seen, with a small number of neurons, the detection rate is very low for both databases. The detection rate keeps improving as the number of neurons increases from 100 to 600. Adding more neurons does not help to further boost the performance as the number of neurons goes beyond 600. This supports our choice of 600 neurons for ELM in the experiments.

4.7. Discussion on model generalizability

To evaluate model generalizability, we further perform leave-one-out (LOO) cross validation. LOO is a special case of k-fold cross-validation in which k=N and N is equal to the number of samples in the training data. Since maximum use is made of the training set, the LOO cross-validation gives a reliable estimate of the generalization error. As noted in [45] for LOO validation for the ELM classifier, PRESS statistics [46] can be used to calculate the LOO accuracy such that the LOO accuracy can be obtained in one go instead of training the model N times. Tables 8 and 9 show the LOO smile detection accuracy.

On both databases, in general for different classifiers, the detection accuracy using LOO cross-validation is higher than four-fold cross validation as compared to the results in Tables 2 and 3. This is due to the availability of more training data in the LOO scenario. Compared to the baseline LDA and SVM classifiers, the detection accuracy of the ELM classifier is higher. The superiority of the ELM classifier using LOO cross-validation complies with the performance results in Tables 2 and 3.

The previous experiments are conducted within the scope of individual databases. To evaluate how the trained model generalizes on unseen data, here we report smile detection results by cross-database validation. Specifically, we train the models on entire MIX database and validate on GENKI-4K database and vice versa. For cross-database validation, all images from both databases are normalized to 100 × 100. Tables 10 and 11 show the cross-database smile detection accuracy. The results suggest that

---

\(^1\) www.csie.ntu.edu.tw/~cjlin/libsvm/.
compared to LDA and SVM classifiers, the trained ELM models are more robust in cross-database evaluation. The average performance on MIX database as the testing data is higher than the average performance on GENKI-4K database as the testing data. This is because GENKI-4K database is more challenging with real-world images under different imaging conditions.

### 5. Conclusions and future work

Smile detection has many potential applications in practice. In this paper, a fully automated smile detection approach using Extreme Learning Machine (ELM) is proposed. The face registration in our approach is performed automatically in a holistic...
The proposed approach is compatible with any feature descriptors. Thus, the future work will include analysis and selection of more advanced feature descriptors to further improve the smile detection accuracy. In addition, additional preprocessing steps such as image illumination normalization will be incorporated to offset the lighting effects on real-world images for more robust smile detection.

Fig. 13. Detection rate V$\nu$ number of neurons for the GENKI-4K database.

Table 8
Smile detection accuracy (in %) for the MIX database using leave-one-out (LOO) validation strategy.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Pixel values</th>
<th>HOG</th>
<th>LBP</th>
<th>LPQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>91.5</td>
<td>94.0</td>
<td>93.1</td>
<td>93.5</td>
</tr>
<tr>
<td>SVM</td>
<td>93.2</td>
<td>94.5</td>
<td>94.3</td>
<td>94.9</td>
</tr>
<tr>
<td>ELM</td>
<td>93.4</td>
<td>94.6</td>
<td>94.5</td>
<td>95.2</td>
</tr>
</tbody>
</table>

Table 9
Smile detection accuracy (in %) for the GENKI-4K database using leave-one-out (LOO) validation strategy.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Pixel values</th>
<th>HOG</th>
<th>LBP</th>
<th>LPQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>74.5</td>
<td>87.1</td>
<td>77.7</td>
<td>78.3</td>
</tr>
<tr>
<td>SVM</td>
<td>80.6</td>
<td>87.3</td>
<td>85.9</td>
<td>84.5</td>
</tr>
<tr>
<td>ELM</td>
<td>81.1</td>
<td>88.7</td>
<td>86.2</td>
<td>85.5</td>
</tr>
</tbody>
</table>

Table 10
Smile detection accuracy (in %) for the MIX database using GENKI-4K database for training.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Pixel values</th>
<th>HOG</th>
<th>LBP</th>
<th>LPQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>78.6</td>
<td>90.2</td>
<td>83.9</td>
<td>86.4</td>
</tr>
<tr>
<td>SVM</td>
<td>86.7</td>
<td>91.4</td>
<td>88.7</td>
<td>87.8</td>
</tr>
<tr>
<td>ELM</td>
<td>87.1</td>
<td>92.3</td>
<td>90.3</td>
<td>88.9</td>
</tr>
</tbody>
</table>

Table 11
Smile detection accuracy (in %) for the GENKI-4K database using MIX database for training.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Pixel values</th>
<th>HOG</th>
<th>LBP</th>
<th>LPQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>71.8</td>
<td>77.6</td>
<td>73.1</td>
<td>77.1</td>
</tr>
<tr>
<td>SVM</td>
<td>75.3</td>
<td>79.1</td>
<td>75.8</td>
<td>77.2</td>
</tr>
<tr>
<td>ELM</td>
<td>76.4</td>
<td>81.0</td>
<td>78.9</td>
<td>81.4</td>
</tr>
</tbody>
</table>

manner and no manual labeling or key-points’ detection is required. In terms of computational cost, the ELM based classifier is very efficient. Experiments on both lab-controlled database and real-world database show that the proposed ELM based smile detection is very competitive in terms of detection accuracy and its efficiency enables the potential for real-time applications. In addition, the cross-database validation suggests that compared to benchmark classifiers, ELM has better generalization performance.

The proposed approach is compatible with any feature descriptors. Thus, the future work will include analysis and selection of more advanced feature descriptors to further improve the smile detection accuracy. In addition, additional preprocessing steps such as image illumination normalization will be incorporated to offset the lighting effects on real-world images for more robust smile detection.

Acknowledgments
This work was supported in part by National Science Foundation Grants 0905671 and 0915270.

References
Songfan Yang received the B.S. degree in Electrical Engineering from Sichuan University, Chengdu, China and the B.S. degrees in both Computer Science and Electronic Engineering Technology from Eastern New Mexico University, Portales, NM, USA, in 2009. Since then, he has been a Ph.D. candidate in Electrical Engineering at the Center for Research in Intelligent Systems (CRIS) at the University of California Riverside. His research interests include computer vision, pattern recognition, machine learning, human behavior understanding, and facial expression/emotion understanding. He holds the Best Paper Award of the FG 2011 Facial Expression Recognition and Analysis Emotion sub-challenge (FERA) competition.

Bir Bhanu received the S.M. and E.E. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, MA, the Ph.D. degree in Electrical Engineering, from the Image Processing Institute at University of Southern California and the M.B.A. degree from the University of California, Irvine. He is the Distinguished Professor of Electrical Engineering and Cooperative Professor of Computer Science and Engineering, Mechanical Engineering and Bioengineering, and the Director of the Center for Research in Intelligent Systems (CRIS) and the Visualization and Intelligent Systems Laboratory (VSLab) at the University of California, Riverside (UCR). In addition, he serves as the director of NSF IGERT on Video Bioinformatics at UCR. He has been the Principal Investigator of various programs for NSF, DARPA, NASA, AFOSR, ONR, ARO, and other agencies and industries in the areas of video networks, video understanding, video bioinformatics, learning and vision, image understanding, pattern recognition, target recognition, biometrics, autonomous navigation, image databases, and machine-vision applications. He has published seven coauthored and three edited books. He is the holder of 18 (5 pending) patents. He has published more than 450 reviewed technical publications, including over 100 journal papers and 40 book chapters. His current research interests are Computer Vision, Pattern Recognition and Data Mining, Machine Learning, Artificial Intelligence, Image Processing, Image and Video Database, Graphics and Visualization, Robotics, Human–Computer Interactions, Biological, Medical, Military and Intelligence applications. He is a Fellow of IEEE, AAAS, IAPR, and SPIE.

Le An received the B.Eng. degree in Telecommunications Engineering from Zhejiang University, Hangzhou, China, in 2006 and the M.Sc. degree in Electrical Engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2008. He is currently a Ph.D. candidate in Department of Electrical Engineering at the Center for Research in Intelligent Systems (CRIS) at the University of California, Riverside. His research interests include image processing, computer vision, pattern recognition, and machine learning. His current research focuses on face recognition and facial expression analysis. He holds the best paper award of the 2013 IEEE International Conference on Advanced Video and Signal-Based Surveillance.

[40] The MPLAB GENKI Database, GENKI-4K Subset 〈http://mplab.ucsd.edu〉.