EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Multithreading cascade of SURF for facial expression recognition

Jinhui Chen[1*], Zhaojie Luo[2], Tetsuya Takiguchi[3] and Yasuo Ariki[3]

Abstract

We propose a novel and general framework called the multithreading cascade of Speeded Up Robust Features (McSURF), which is capable of processing multiple classifications simultaneously and accurately. The proposed framework adopts SURF features, but the framework is a multi-class and simultaneous cascade, i.e., a multithreading cascade. McSURF is implemented by configuring an area under the receiver operating characteristic (ROC) curve (AUC) of the weak SURF classifier for each data category into a real-value lookup list. These non-interfering lists are built into thread channels to train the boosting cascade for each data category. This boosting cascade-based approach can be trained to fit complex distributions and can simultaneously and robustly process multi-class events. The proposed method takes facial expression recognition as a test case and validates its use on three popular and representative public databases: the Extended Cohn-Kanade, MMI Facial Expression Database, and Annotated Facial Landmarks in the Wild database. Overall results show that this framework outperforms other state-of-the-art methods.

Keywords: AdaBoost, Multithreading cascade, SURF, AUC, Facial expression recognition

1 Introduction

Robustly and simultaneously learning highly discriminative multiclass classifiers with local image features is one of the most significant challenges to computer vision researchers, because they are critical infrastructures for recognition engines; consequently, these researches appear of great importance. Our study focuses on feature descriptors and learning classifiers to develop a novel learning framework for multiclass recognition applications.

In this study, we propose a framework called the multithreading cascade of Speeded Up Robust Features (McSURF), which adopts SURF for training a multithreading boosting cascade. The proposed learning model is applied to facial expression recognition (FER), and while it is derived from AdaBoost [1], it is a novel, multi-class, simultaneous cascade, i.e., a multithreading cascade. In contrast to the conventional boosting cascade models (e.g., BinBoost [2], joint cascade [3], and LUT-AdaBoost [4–6]), we propose a novel and robust cascade algorithm (McSURF) that can simultaneously learn multi-task cascades using the local feature detector and descriptor SURF [7]. The proposed boosting cascade-based approaches can be trained to fit complex distributions and can simultaneously process multi-class events much more robustly.

We experimentally evaluated the proposed method in three public expression databases, i.e., the Extended Cohn-Kanade (CK+) [8], MMI Facial Expression Database [9, 10], and Annotated Facial Landmarks in the Wild (AFEW) database [11], that together represent lab-controlled and real-world scenarios. Some examples of expression recognition results are shown in Fig. 1. The experimental results show that the proposed method can construct a robust FER system whose results outperform well-known state-of-the-art FER methods.

The main contribution of our study is the development of a novel framework (McSURF) that can simultaneously build a cascade learning model while robustly processing a multiclass recognition application. By so doing, we are making the following original contributions: (1) Typically, a boosting classifier is trained as a binary classification model. Our proposed multithreading cascade learning model allows multiple categories to be simultaneously trained on a cascade learning model. (2) The McSURF is an excellent FER application method. Its performance

*Correspondence: ianchen@me.cs.scitec.kobe-u.ac.jp
[1] RIEB, Kobe University, 2-1 Rokkodai, 657-8501 Kobe, Hyogo, Japan
Full list of author information is available at the end of the article

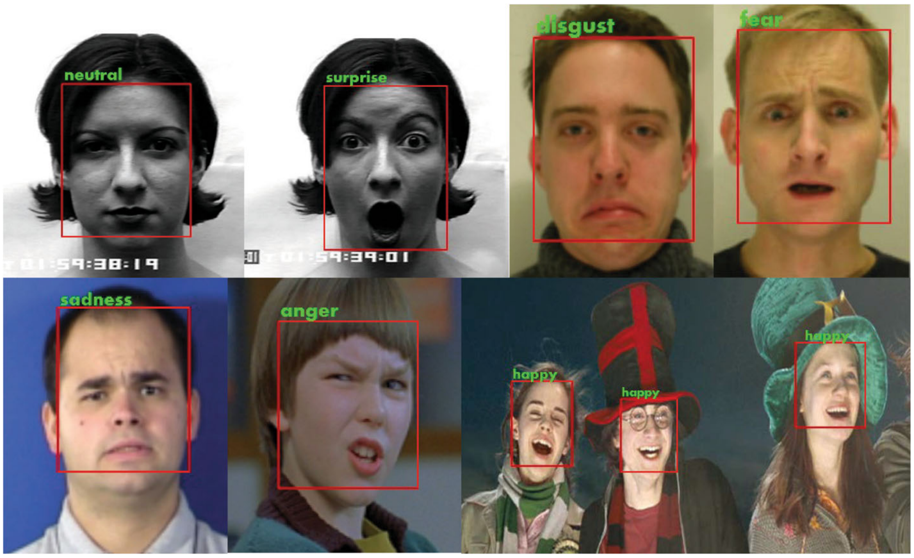

Fig. 1 Examples of facial expression recognition results

experimentally outperforms many state-of-the-art methods. (3) We experimentally evaluated the impact of face registration at both learning and recognition stages and determined how face registration works on a boosting classifier during these stages. This represents an important breakthrough that is relevant to related industries and those with related research interests.

The remainder of this paper is organized as follows: We review the related works in Section 2. We describe the proposed framework in Section 3. In Section 4, we describe our experiments, and we draw our conclusions in Section 5.

2 Related work

Recently, mainstream FER approaches are based on effective local descriptors or facial action units. Local descriptors such as local binary pattern on three orthogonal planes (LBP-TOP) [12], HOE [13], and histograms of oriented gradients (HOG) 3D [14] are extracted from the local facial cuboid to obtain a representation of a certain length independent of time resolution. In other words, these approaches try to describe the spatiotemporal property of facial expressions using descriptors. These feature descriptor approaches present effective and robust FER representations, because they can avoid intra-class variation and face deformation. However, rigid cuboids can only capture low-level feature information and these low-level features often fail to describe high-level facial concepts, i.e., there is a “semantic gap” between low-level features and high-level concepts. Therefore, the effective use of local descriptors to represent complex expressions has been an ongoing problem.

Another approach is adopted for processing facial action areas. Although these approaches are not more popular than those based on local descriptors, this method category is also important to consider. Methods based on facial action areas use a series of facial landmarks, as discussed in [8] and [15] and use the active appearance model (AAM) [16] and the constrained local model (CLM) [15, 17] to encode shape and texture information, respectively. These approaches do not have the semantic gap problem, because they focus solely on the detection of mid-level facial action areas, which contain sufficient semantic cues. However, it is difficult to accurately detect landmarks (or defined action units) when facial expression varies, because these defined landmarks cannot completely address the many varied and complex expressions.

This study aims to present a more ideal solution for FER. It have been proved that local features trained by classifiers can effectively cancel out the problems caused by semantic gap, which leads to an overall significant improvement of the classification performance. Therefore, we propose a novel and general learning framework that contains robust classifiers as well as high-quality local feature descriptors, and the technical details are discussed in the following section.

3 The proposed method

Our proposed framework has these components: SURF features for local patch description; logistic regression-based weak classifiers, which are combined with the area under the receiver operating characteristic (ROC) curve (AUC) [18] as a single criterion for cascade convergence testing; and a multithreading cascade for boosting training that can process multiple categories.

Figure 2 shows a schematic of the implementation process of the proposed framework. First, the facial region is detected based on the V-J framework. Then, the detected facial region is parallel processed by multiple classifiers to estimate the expression. The parallel classification, i.e., the multithreading aspect, is implemented by configuring the AUC of the weak classifier for each data category into a real-value lookup list. As shown in Fig. 2, these non-interfering lists are built into thread channels in which the algorithm can appropriately organize the ensemble of weak classifiers into related classes. In the proposed framework, SURF represents the expressional features of the detected facial regions for weak classifiers. We describe SURF in Section 3.1 and explain how to use SURF features to construct logistic regression-based weak classifiers in Section 3.2. To start the parallel aspect as shown in Fig. 2, we design the multithreading cascade channel in Section 3.3. We describe how to learn weak classifiers in each channel in Section 3.4. Finally, in Section 3.5, we describe the boosting cascade training. These approaches are formulated in the following section.

3.1 Feature description

SURF is a scale- and rotation-invariant interest point detector and descriptor. It is faster than scale-invariant feature transform (SIFT) [7, 19], and AdaBoost-based algorithms that have adopted SURF have been shown to obtain the best accuracy and speed [20]. In this study, we adopt an 8-bin T2 SURF descriptor to describe the local features, inspired by the approach of Li et al. [20]. However, in contrast to Li et al.'s [20] approach, we allow different aspect ratios for each patch (the ratio of width and height) because this can improve the speed of image traversal. We also imported diagonal and anti-diagonal filters to improve the description capability of the SURF descriptors.

Given a recognition window, we define rectangular local patches within it, each patch having four spatial cells and with the patch size ranging from 12 × 12 to 40 × 40 pixels. Each patch is represented by a 32-dimensional SURF descriptor, which can be computed quickly based on the sums of two-dimensional Haar wavelet responses, and we can make efficient use of the integral images [1]. d_x is defined as the horizontal gradient image, which can be obtained using the filter kernel $[-1, 0, 1]$, and d_y is

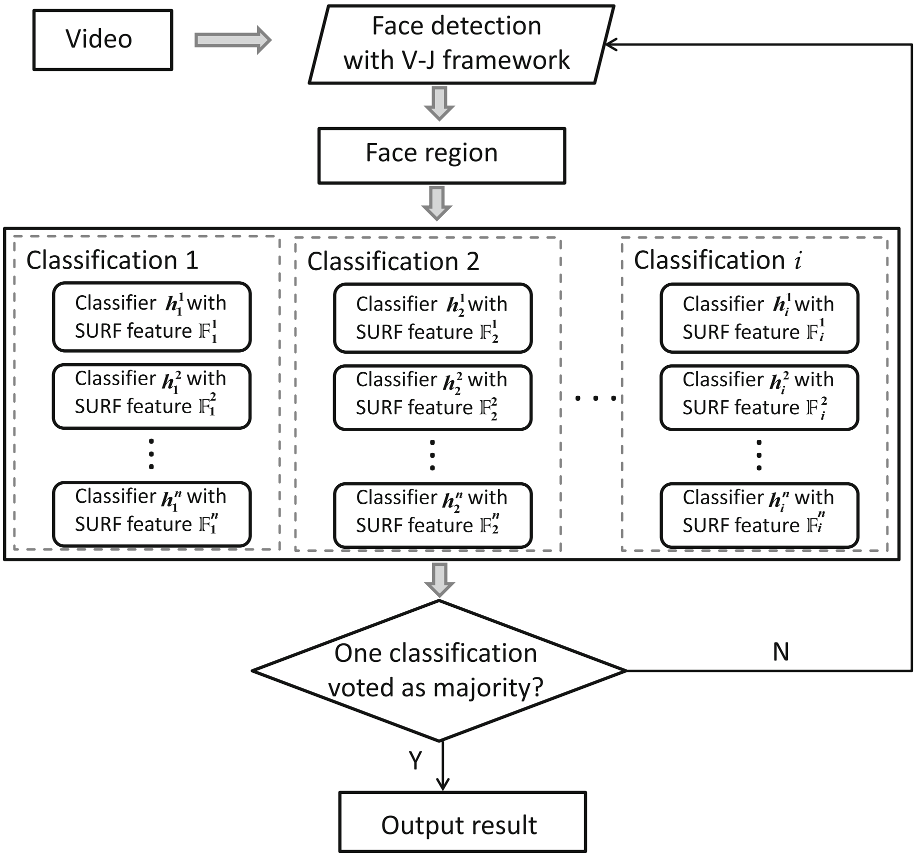
Video
Face detection with V-J framework
Face region
Classification 1
Classifier h_1^1 with SURF feature $\mathbb{F}_1^1$
Classifier h_1^2 with SURF feature $\mathbb{F}_1^2$
Classifier h_1^n with SURF feature $\mathbb{F}_1^n$
Classification 2
Classifier h_2^1 with SURF feature $\mathbb{F}_2^1$
Classifier h_2^2 with SURF feature $\mathbb{F}_2^2$
Classifier h_2^n with SURF feature $\mathbb{F}_2^n$
Classification i
Classifier h_i^1 with SURF feature $\mathbb{F}_i^1$
Classifier h_i^2 with SURF feature $\mathbb{F}_i^2$
Classifier h_i^n with SURF feature $\mathbb{F}_i^n$
One classification voted as majority?
N
Y
Output result

Fig. 2 Processing flow of the proposed framework

the vertical gradient image, which can be obtained using the filter kernel $[-1, 0, 1]^T$; define d_{D} as the diagonal image and d_{AD} as the anti-diagonal image, both of which can be computed using two-dimensional filter kernels diag $(-1, 0, 1)$ and antidiag $(-1, 0, 1)$. Therefore, 8-bin T2 is able to be defined as $v = (\sum(|d_x| + d_x), \sum(|d_x| - d_x), \sum(\left|d_y\right| + d_y), \sum(\left|d_y\right| - d_y), \sum(|d_{\mathrm{D}}| + d_{\mathrm{D}}), \sum(|d_{\mathrm{D}}| - d_{\mathrm{D}}), \sum(|d_{\mathrm{AD}}| + d_{\mathrm{AD}}), \sum(|d_{\mathrm{AD}}| - d_{\mathrm{AD}}))$. Here, d_x, d_y, d_{D}, and d_{AD} can be computed individually, using integral images, by the filters shown in Fig. 3a(1), a(2), b(1), and b(2) , respectively. For details on how to compute the two-dimensional Haar responses with integral images, please refer to [1].

The recognition template for SURF is 40 × 40 pixels with four spatial cells, again with the patch size ranging from 12 × 12 to 40 × 40 pixels. We slide the patch over the recognition template with four pixels forward to ensure a sufficient feature-level difference. In addition, we allow a different aspect ratio for each patch. The local candidate region of the features is also divided into four cells, and the descriptor is extracted from each cell. Hence, concatenating the features in all four cells yields a 32-dimensional feature vector. In practical feature normalization, an L_2 normalization followed by clipping and renormalization (L_2Hys) [21] has been shown to work best.

3.2 Weak classifier construction

In this study, we build a weak classifier over each local patch described by the SURF descriptor and select the optimum patches in each boosting iteration from the patch pool. Meanwhile, we construct the weak classifier for each local patch by logistic regression to fit our classifying framework, due to it being a probabilistic linear classifier.

On one hand, we build a weak classifier over each local patch, as described by the SURF descriptor, and select optimum patches in each boosting iteration from the patch pool. On the other hand, we construct a weak classifier for each local patch by logistic regression to fit our classification framework, since it is a probabilistic linear classifier. Given a SURF feature $\mathbb{F}$ over a local patch, logistic regression defines the probability model

$$P(q|\mathbb{F}, \mathbf{w}) = \frac{1}{1 + \exp\left(-q(\mathbf{w}^T\mathbb{F} + b)\right)}, \qquad (1)$$

where $q = 1$ means that the trained sample is a positive sample of the current class, $q = -1$ indicates negative samples, $\mathbf{w}$ is a weight vector for the model, and b is a bias term. We train classifiers on local patches from a large-scale dataset. Assuming, in each boosting iteration stage, that there are K possible local patches, which are represented by SURF feature $\mathbb{F}$, each stage is a boosting training procedure with logistic regression as weak classifiers. In this way, the parameters can be identified by minimizing the objective

$$\sum_{k=1}^{K} \log\left(1 + \exp\left(-q_k\left(\mathbf{w}^T\mathbb{F}_k + b\right)\right)\right) + \lambda \left\|\mathbf{w}\right\|_p, \qquad (2)$$

where λ denotes a tunable parameter for the regularization term and $\|\mathbf{w}\|_p$ is the L_p norm of the weight vector. Note that it is also applied to L_2-loss and L_1-loss linear support vector machines (SVMs) by the well-known open source code LIBLINEAR [22]. Therefore, this question can be solved using algorithms in [22].

3.3 Multithreading cascade channel construction

In this subsection, we introduce how to implement the parallel aspect. Assuming there are M expression categories in the training sample set, given the weak classifiers $h_i^{(n)}$ for category i data, the strong classifier is defined as $H_i^{(N)}(\mathbb{F}) = \frac{1}{N}\sum_{n=1}^{N} h_i^{(n)}(\mathbb{F})$.

Assuming there are a total of N boosting iteration rounds, in the round n, we will build K weak classifiers $[h_i^{(n)}(\mathbb{F}_k)]_{k=1}^{K}$ for each local patch in parallel from the boosting sample subset. Meanwhile, we also test each model $h_i^{(n)}(\mathbb{F}_k)$ in combination with previous $n-1$ boosting rounds. In other words, we test $H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k)$ for $H_i^{(n)}(\mathbb{F})$ on the all training samples, and each

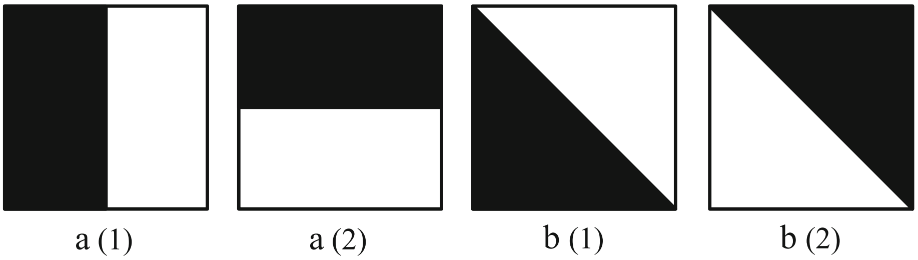

Fig. 3 Filters used for computing SURF descriptors. **a(1)** for d_x, **a(2)** for d_y, **b(1)** for d_{D}, and **b(2)** for d_{AD}

test model will produce a highest AUC score [18, 23] $J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k))$, i.e.,

$$S_i^{(n)} = \max_{k=1,\cdots K} J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k)). \tag{3}$$

This procedure is repeated until the AUC scores converge or the designated number of iterations N is reached. Then, the selected S_i is set as a threshold to generate an AUC score pool, which contains the values of $J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k)) \geq 0.8 \times S_i$. In this way, it builds an AUC score pool for each class of object.

To learn multi-class classifiers simultaneously, we adopt these AUC data to construct independent channels for boosting learning. The details of the procedure are summarized as follows:

1. Assuming the AUC score pools have been normalized to $[0, 1]$, we divide the range into M sub-range bins. Each bin corresponds to a channel ID. In this way, we can obtain a channel ID set $\mathbf{C} = \{\text{bin}_l = [\frac{(l-1)}{M}, \frac{l}{M}] \mid l = 1, \cdots, M\}$. In each channel, we build an independent boosting model for training classifiers that can recognize a corresponding category task.
2. We set $u = S_i(\mathbb{F}, x)$ and define the weak classifier $h_i(x)$ as follows:

   $$\textbf{if } u \in \mathbf{C} \text{ and } x \in \{\text{the samples of expression } i\}, \\ \textbf{then } h_i(x) = 2P(q|\mathbb{F}, \mathbf{w}) - 1. \tag{4}$$

   These guarantee that the precision of h is greater than 0.5.
3. Given the characteristic function

   $$B^{(l,i)}(u, \mathbf{Y}) = \begin{cases} 1 & u \wedge \mathbf{Y} = i \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

   where $i \in \mathbf{Y}$ and $\mathbf{Y}$ is defined as the label set of those expression categories that can be recognized by the classifier h. This function is used to check and ensure that the expression categories of the channel, classifier, and sample are consistent.
4. Lastly, to cover the characteristic function, we formally express the weak classifier as

   $$h(\mathbb{F}) = \sum_{l=1}^{M}\sum_{i=1}^{M} (2P(q|\mathbb{F}, \mathbf{w}) - 1)\, B^{(l,i)}(u, \mathbf{Y}). \tag{6}$$

   As shown in Fig. 2, by using the above approaches, we can construct the parallel aspect for training. Meanwhile, the classifier category is able to be judged and auto-selected into the related channel. In this way, we can learn the classifiers with Algorithm 1 and train multithreading boosting cascades simultaneously in their training channels via Algorithm 2.

Algorithm 1 Learning Boosting Classifiers on SURF.

```
Require:
  1. Given: the number of label categories $M$ and the overall sample set $\mathbf{S} = \{(x_1, y_1), \cdots, (x_\tau, y_\tau)\}$, where $\tau$ is the number of the samples;
  2. Initialize the weight parameter $w_0$ for positive (labeled as "+") samples and negative (labeled as "-") samples:
    a. $w_0^+ = 1/(M \times \tau_+)$ for those $q = 1$;
    b. $w_0^- = 1/(M \times \tau_-)$ for those $q = 1$;
  3.
  for ($j = 0; j < N; j = j + 1$) do
    a. Sampled $30 \times p$ (in this paper, $p = 3$) positive samples and $30 \times p$ negative samples from training set;
    b. Parallel replace each SURF template to train a series of logistic regression models $[h_i(\mathbb{F}_k)]_{k=1}^{K}$;
    c. In order to obtain the AUC score, calculate $H_i^{(n-1)}(\mathbb{F}) + h_i(\mathbb{F}_k)$ on the best model of the previous stage: $S_i^{(n-1)}$ and each $h_i(\mathbb{F}_k)$;
    d. Choose the best model $S_i^{(n)}$ that contains the best weak classifier $h_i(\mathbb{F}_j)$, according to the Eq. 3;
    e. Update the weight
         $w_{j+1} = \frac{w_j \exp(-q_j Y_i h_i(\mathbb{F}_j))}{Z_j}$,
    where $Z_j$ is a normalization factor, which makes the weight follow $\sum w^+ = 1$ and $\sum w^- = 1$;
    f. If AUC value $S_i^{(n)}$ is converged, break the loop;
  end for
  4. In order to ensure the overall AUC score to be the highest one, test all learned models during the current iteration process:
  for ($j = 0; j < K; j = j + 1$) do
    if $H_i^{(n-1)}(\mathbb{F}) + h_i(\mathbb{F}_i) > S_i^{(n)}$ then
      a. $S_i^{(n)} = H_i^{(n-1)}(\mathbb{F}) + h_i(\mathbb{F}_j)$;
      b. Empty those unnecessary data to free the memory;
    end if
  end for
  5. Output final strong model $H_i^{(n)}$ for this stage.
```

3.4 Learning weak classifiers

In this subsection, we describe how to learn weak classifiers in each threading channel as shown in Fig. 2. Like most existing multiclass classification algorithms,

```
Algorithm 2 Training multithreading boosting cascade
Require:
    1. Over all FPR: $F_i^{(n)}$ for $i$-th category data;
    2. Minimum hit-rate per stage $d_i^{(min)}$;
    3. Current class samples: $\mathbf{X}_i^+$;
    4. Non-current class samples: $\mathbf{X}_i^-$;
    5. The number of sample/label categories: $M$;
    Initialize: $j = 0, F_i^{(j)} = 1, D_i^{(j)} = 1$;
    for $(i = 0; i < M; i = i + 1)$ do
        while $(F_i^{(j)} > F_i^{(n)})$ do
            1. j=j+1;
            2. Train a stage classifier $H_i^{(j)}(\mathbb{F})$ by samples of $\mathbf{X}^+$ and $\mathbf{X}^-$ via approaches of subsection 3.3;
            3. Evaluate the model $H_i^{(j)}(\mathbb{F})$ on the whole training set to obtain ROC curve;
            4. Determine the threshold $\theta_i^{(j)}$ by searching on the ROC curve to find the point $(d_i^{(j)},f_i^{(j)})$ such that $d_i^j = d_i^{(min)}$, but when existing the mimimum one $d_i^{(j)}$ that follows to the condition: $d_i^{(j)} < d_i^{(min)}$, set $d_i^{(min)} = d_i^{(j)}$ to update the minimal hit-rate;
            5. Update: $F_i^{(j)} = F_i^{(j-1)} \times f_i^{(j)}$,
                       $D_i^{(j)} = D_i^{(j-1)} \times d_i^{(j)}$;
            6. Empty the set $\mathbf{X}_i^-$;
            7. while $(F_i^{(j)} > F_i^{(j-1)}$ and size $|\mathbf{X}_i^+| \neq |\mathbf{X}_i^-|$ ) do
            Adopt current cascade detector to scan non-target images with sliding window and put false-positive samples into $\mathbf{X}_i^-$;
            end while
        end while
    end for
    8. Output the boosting cascade detector $\{H_i^{(j)} > \theta_i^{(j)}\}$ and overall training accuracy $F$ and $D$.
```

our approach is crucially dependent on the labeled data of sample space to learn the classifiers. In this study, we combine this approach with the above constructed cascade channels to implement multiclass classification. In our case, we denote the sample space as $\mathbf{X}$ and the label set as $\mathbf{Y}$. A sample of a multiclass and multilabel problem is a pair (x, Y), where $Y(i)$ is defined as

$$Y(i) = \begin{cases} 1 \text{ if } i \in Y \\ -1 \text{ if } i \notin Y \end{cases}, \tag{7}$$

where $x \in \mathbf{X}, l \in \mathbf{Y}$, and $Y \subseteq \mathbf{Y}$.

The whole procedure involves a forward selection and inclusion of a weak classifier over possible local patch temples that can be adjusted using different temple configurations, according to the processing images. To enhance both the speed of the learning convergence and robustness, our algorithm further introduces a backward removal approach. For more details on including backward removal or even a floating searching capability into the boosting framework, please refer to [24]. In this study, we implement backward removal on Algorithm 1 step 4 to extend the procedure with the capability to backward remove redundant weak classifiers. In so doing, it is not only able to reduce the number of weak classifiers in each stage but also improve the generalization capability of the strong classifiers.

3.5 Boosting cascade training

Inspired by [18] and [20], here, we introduce AUC as a single criterion for cascade convergence testing, which realizes an adaptive False Positive Rate (FPR) among the different stages (for a more detailed description of AUC, refer to [18]). Hence, combined with logistic regression-based weak classifiers to adopt SURF features, this approach can yield a fast convergence speed and a cascade model with much shorter stages.

Within one stage, no threshold for intermediate weak classifiers is required. We need only determine each decision threshold θ_i for ith emotional category in its threading channel. In our case, using the ROC curve, the FPR of each emotional category is easily determined when given the minimal hit rate $d_i^{(\min)}$. We decrease $d_i^{(j)}$ from 1 on the ROC curve, until reaching the transit point $d_i^j = d_i^{(\min)}$. The corresponding threshold at that point is the desired θ_i, i.e., the FPR is adaptive to different stage, and it is usually much smaller than 0.5. Therefore, its convergence speed is much quicker than the conventional methods.

To avoid overfitting, we restricted the number of samples used during training, as in [25]. In practice, we sampled an active subset from the whole training set according to the boosting weight. It is generally good practice to use about 30 × p samples of each class, where p is a multiple coefficient (Algorithm 1 step 3.a).

After one stage of classifiers learning is converged via Algorithm 2, we continue to train another one with false-positive samples coming from the scanning of non-target images with the partially trained cascade . We repeat this procedure until the overall FPR reaches the stated goal. As with many current methods [3, 20, 26, 27], this measure was also inspired by the V-J framework [1], and although we indicated in Section 3.3 that we had adopted this approach, our approach is able to process binary cascades, as well as multi-class cascades. In every independent threading channel, respective cascade recognition sub-frameworks can be trained simultaneously for each data category. Furthermore, we propose an algorithm (Algorithm 2) to implement the boosting ensemble of classifiers for multiclass cascades, which is an original contribution to boost learning research. Equally important is that in our approach, the cascade training process is based on

AUC analysis, and the FPR is usually much smaller than 0.5. In addition, it is adaptive for different stages. Therefore, this approach can result in a model size that is much smaller and has a recognition speed that is dramatically increased.

4 Experiments

In this section, we provide details of the dataset and evaluation results for our proposed method, as applied to FER. We implemented all training and recognition programs in C++ on Red Hat Enterprise Linux (RHEL) 6.5 OS, processed with a PC with a Core i7-2600 3.40 GHz CPU and 8 GB RAM.

4.1 Databases and protocols

We evaluated the proposed method on three public databases, i.e., CK+, MMI, and AFEW, which include two lab-controlled databases (CK+ and MMI) and one with real-world scenarios (AFEW).

4.1.1 CK+ DB

The CK+ database (DB) is a set of facial expression samples posed by 123 people. There are 327 sequences, taken from 593 sequences that meet the criteria for one of seven discrete emotions of the Facial Action Coding System (FACS) [8] (anger (An), contempt (Co), disgust (Di), fear (Fe), happiness (Ha), sadness (Sa), and surprise (Su)). In our experiments, we divided these samples into several groups for each expression by the person-independent rule, and each group included ten posers. A person-independent tenfold cross-validation had been conducted for this DB to compare the results of a number of the outstanding current methods. For the recognition experiments, we put these images into 10-min-length videos, 640 × 480 in size, and with a frame rate of 60 frames per second (FPS), based on the person-independent rule.

4.1.2 MMI DB

The MMI DB is a public database that includes more than 30 subjects, in which the female-to-male ratio is roughly 11:15. The subjects' ages range from 19 to 62, and they are of European, Asian, or South American descent. This database is considered to be more challenging than CK+ because some posers have worn accessories such as glasses. In the experiments, we used all 205 effective image sequences of the six expressions in the MMI dataset. As with the CK+ DB, a person-independent tenfold cross-validation had been completed to compare results from the state-of-the-art methods. For the recognition stage, these images were also made into 10-min-length videos, 640× 480 in size, and with a frame rate of 60 FPS based on the person-independent rule.

4.1.3 AFEW DB

For the AFEW DB, which is a much more challenging database, evaluation experiments also have been done [11]. All of the AFEW sets were collected from movies to depict the so-called wild scenarios. For this study, we adopted the 2013 AFEW version [28], because the evaluation results of many state-of-the-art methods have been based on this version. We trained the training set, and the results are reported for its validation set, in the same way as for the latest FER work [29].

4.2 Face registration

Like most facial research, our recognition performance is assessed based on faces normalized by the position of the eyes [30–32], i.e., eye centers are used to register faces, whereas we utilized elastic bunch graph matching (EBGM) training images for the the rest [31, 32]. Some examples of randomly selected faces on eye perturbation are shown in Fig. 4.

At first, to determine the impact of face registration on the boosting convergence speed, we considered eye perturbation in the training sets only. Our results showed that the proposed method used only 261 min to converge at the 11th stage. In contrast, our proposed method used 422 min to converge at the 16th cascade stage when not using any face registration approach, and there is no corresponding increase in performance.

We expect that classifier testing on a dataset with similar registration may improve the recognition results. Therefore, we next considered eye perturbation in both the recognition and training stages. As shown in Fig. 5b, the performance improved by 3–6 % , compared to the results in Fig. 5a.

The experiments show that in the FER case, the registration of the face is very important, because if it was necessary to craft features for every permutation, this would require more data. However, this problem is solved by using a good face registration approach, which requires less data and a reduced number of boosting convergence stages. Face registration in both the testing and training sets can improve the robustness of these algorithms in FER applications. Because the classifiers are trained on face images with similar eye perturbations, they can therefore better cope with face images containing registration errors. This also gives us some insight into why V-J face detection [1] is followed by the use of eye detectors.

4.3 Computational cost evaluation

We used all the training samples in the AFEW training set and collected training samples from the CK+ and MMI DBs, according to the person-independent tenfold cross-validation rule. To reduce the training process time, we trained the samples from the three datasets together. All of the training samples were normalized to 100 × 100 pixel

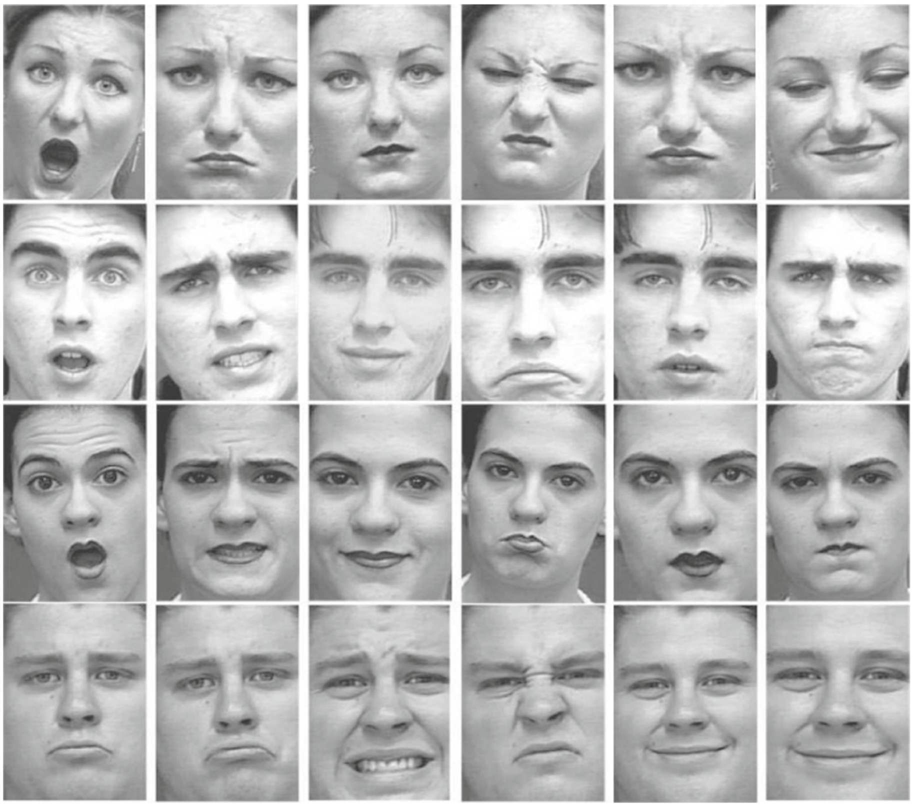

Fig. 4 Randomly selected faces from samples on eye perturbation

facial patches and processed by histogram equalization, and no color information was used. To enhance the generalization performance of boosting learning, we used some transformations in the training samples (mirror reflection, rotate the images, etc.), and finally increased the original number of samples by a factor of 64. Normalization was not performed on any of the testing sample sequences. In the training stages, we adopted the training data of the current processing expression as positive sample data, and data from other expressions as negative data.

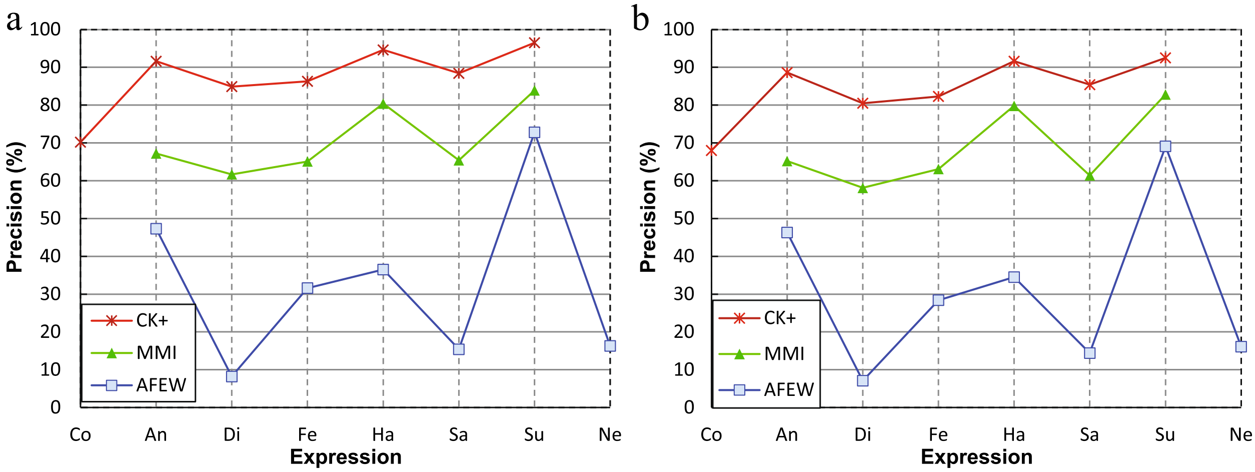

Fig. 5 Evaluation of face registration: the *red line* is the recognition precision on CK+ dataset; the *green line* is the recognition precision on MMI dataset; the *blue line* is the recognition precision on AFEW dataset. **a** Recognition precision introduced face registration into both the training and recognizing stages. **b** Recognition precision introduced face registration into training stages

For every expressional category, we set the maximum number of weaker classifiers in each stage as 128. The proposed method took 281 min to converge at the $11th$ iteration stage. The cascade detector contained 2963 classifiers for all expressions and needed to evaluate only 3.5 SURF per window. Details of the FER cascade, as illustrated in Fig. 6a, b, include the number of weak classifiers in each stage, and the average accumulated rejection rate for all the cascade stages. The results indicate that the first seven stages rejected 98 % of the non-current class samples. After training, we observed that the top three picked local patches for FER laid in the regions of two eyes and mouth. This situation is similar to Haar-based classifiers [6], see the examples in Fig. 7.

In order to evaluate the convergence speed of the AUC model, we determined the FPR at each boosting stage. The results show that, in the AUC model, the FPR f_j at each cascade stage is adaptive among the different stages, ranging from 0.04486 to 0.26837, and is much smaller than the conventional model FPR of 0.5. In almost all existing cascade frameworks, FPR $\prod_{j=1}^{T} f_j$ (T denotes the total cascade stages) reaches the goal (it is usually set as 10^{-6}). This means that conventional models require more iterations and that the AUC model cascade can converge much faster. These relate directly to training efficiency and recognition speed. Therefore, these experimental results confirm that the AUC cascade model is much more efficient than the conventional cascade models. However, since the proposed framework makes the classifiers parallel recognize the multiclass expressions, the peak of memory cost is nearly six times more than the conventional one.

In addition, the average recognition speed of the proposed method is 54.6 FPS for the three datasets. We tried almost all existing local features, such as HOG [21] and SIFT. At first, we thought that SIFT and HOG features would be more discriminating than SURF, but the results show that HOG descriptors lack robustness with regard to head rotation, which has also been pointed out by Klaeser et al. [14]. The accuracy of the SIFT-based version is similar to the results of the 8-bin T2 SURF descriptor, but its memory requirement is four times than that of the 8-bin T2 SURF descriptors. Moreover, the speed was only 15.4 FPS, which cannot process real-time scenes smoothly. In addition, we adopted Haar's version, which contains more than 26 boosting stages and 27,396 classifiers of all categories. It also requires more than 37 Haar-like features per window and has the slowest convergence speed of all. Consequently, we concluded that SURF is more ideal for the proposed framework.

4.4 Recognition result evaluations

In this study, we used the same labels for the expression categories as those in the original databases. All of the recognition experiments are based on videos, and we evaluated their accuracies frame by frame. Here, we present the recognition results for the three representative public databases (CK+, MMI, and AFEW), because we needed to evaluate both the lab-controlled (CK+ and MMI) and real-world (AFEW) scenarios.

We also selected a number of methods for comparison to represent the state-of-the-art of this field, including the methods that have been proposed for improving spatiotemporal descriptors: LBP-TOP [12], HOE [13], PLBP [33], and HOG 3D [14]. CLM [15] is a typical approach that is used to process facial action units. These methods are very popular for FER, while 3DCNN-DAP [34] and STM-ExpLet [29] are the latest methods. We also compared methods that focus on enhancing the robustness of classification approaches for their classifying frameworks, such as ITBN [35], 3D LUT [6], and LSH-CORF [36]. For a fair comparison, we used the same databases, which were evaluated via standardized items. Tables 1, 2, and 3 compare our method (McSURF) with these

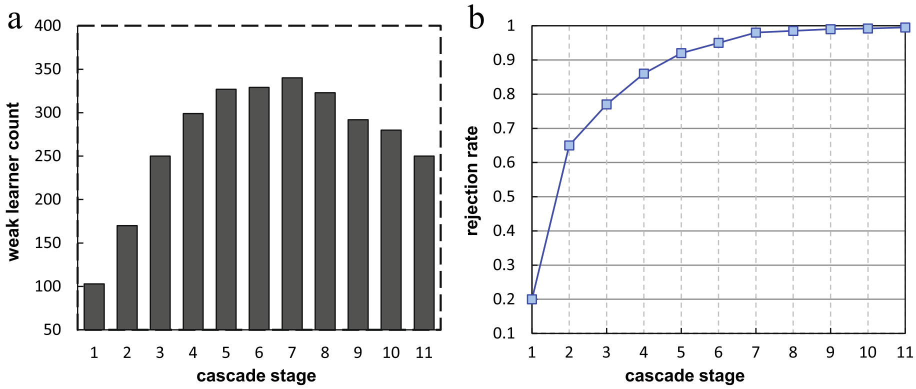

Fig. 6 The number of weak classifiers in each stage and the accumulated rejection rate over cascade stages. **a** The average number of weak classifiers at each stage of FER. **b** The accumulated rejection rate over all the cascade stages

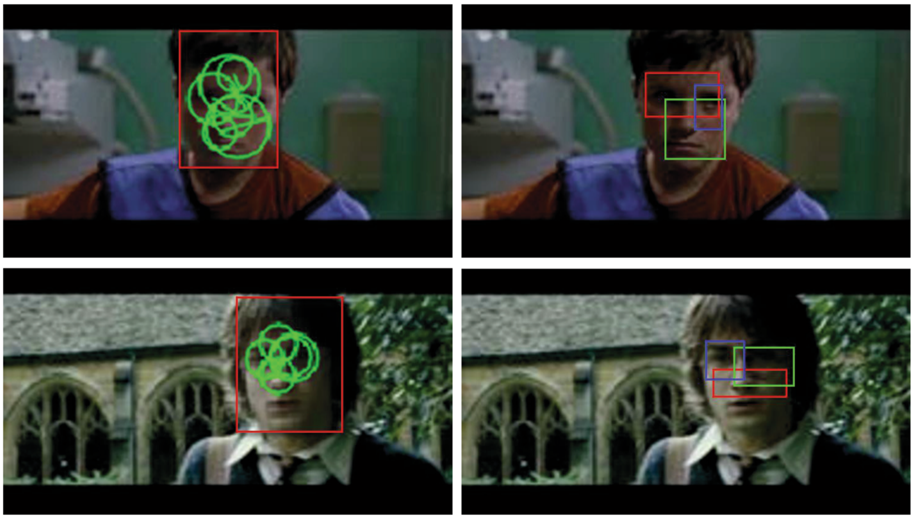

Fig. 7 The extracted SURF features for expressions in facial regions and the top thee local patches picked by training procedure in the *green-red-blue* order on the AFEW database

state-of-the-art methods, most of which were conducted using their released codes and with their parameters tuned to better adapt to our experiments. However, for some methods, because we could not obtain their source codes (e.g., STM-ExpLet [29] and 3DCNN-DAP [34]), it was necessary to simply cite the results reported from related studies. In addition, McSURF (3T) is the item of recognition results (person-independent tenfold) by using the data from the three databases together, yet McSURF (OD) denotes the performance (tenfold) with the original data from each database.

In Table 1, the experimental results, for the CK+ database, compare our approach (McSURF) with eight state-of-the-art methods (CLM [15], HOE [13], LBP-TOP [12], ITBN [35], HOG 3D [14], LSH-CORF [36], 3D LUT [6], and 3DCNN-DAP [34]). The mean average precision (mAP) of our method is highly competitive with state-of-the-art methods.

Table 2 lists the evaluation experiment results for the MMI DB. The proposed method outperformed those state-of-the-art methods. In addition, unlike many existing methods that only evaluate some selected samples, in our experiments, we used all 205 effective image sequences of the six expressions (anger (An), disgust (Di), fear (Fe), happiness (Ha), sadness (Sa), and surprise (Su)).

Table 3 shows the evaluation results for the AFEW database (Ne means neutral), which is designed as a real-world scenario dataset and where the faces have sharp rotations. Since CK+ and MMI are lab-controlled datasets, they have some shortcomings with respect to being evaluated in real-world scenarios. Therefore, we once again compared our proposed method with

Table 1 Recognition results for the CK+ database

Method	Accuracy on CK+ (%)							
	An	Co	Di	Fe	Ha	Sa	Su	mAP.
CLM [15]	70.1	52.4	92.5	72.1	94.2	45.9	93.6	74.4
LBP-TOP [12]	82.2	77.8	91.5	72.0	98.6	57.1	97.6	82.4
HOE [13]	76.4	65.4	83.6	73.3	92.1	88.6	92.8	82.3
HOG 3D [14]	84.4	77.8	**94.9**	68.0	**100**	75.0	**98.8**	85.6
ITBN [35]	91.1	*78.6*	94.0	83.3	89.8	76.0	91.3	86.3
LSH-CORF [36]	71.3	–	90.8	79.0	92.6	90.5	96.6	86.8
3D LUT [6]	76.3	35.1	60.5	73.8	91.0	48.2	92.8	68.2
3DCNN-DAP [34]	91.1	66.7	96.6	80.0	98.6	85.7	96.4	87.9
PLBP [33]	–	–	–	–	–	–	–	**96.7**
McSURF (3T)	91.6	70.2	84.9	86.3	94.6	88.4	96.5	87.5
McSURF (OD)	**96.1**	74.6	86.2	**92.5**	98.2	**94.1**	96.4	91.2

The boldface data are the best results in their items

Table 2 Recognition results for the MMI database

Method	Accuracy on MMI (%)						
	An	Di	Fe	Ha	Sa	Su	mAP.
LBP-TOP [12]	58.1	56.3	53.6	78.6	46.9	50.0	57.2
HOE [13]	46.4	58.3	33.2	62.6	60.8	65.1	55.5
HOG 3D [14]	61.3	53.1	39.3	78.6	43.8	55.0	55.2
3D LUT [6]	43.3	55.3	56.8	71.4	28.2	77.5	47.2
ITBN [35]	46.9	54.8	57.1	71.4	65.6	62.5	59.7
LSH-CORF [36]	59.6	**71.4**	62.3	68.9	**70.3**	75.1	61.8
3DCNN-DAP [34]	64.5	62.5	50.0	**85.7**	53.1	57.5	62.2
STM-ExpLet [29]	–	–	–	–	–	–	65.4
McSURF (3T)	67.2	61.7	65.1	80.4	65.4	**83.9**	70.6
McSURF (OD)	**69.5**	65.3	**68.4**	83.9	68.2	82.6	**73.0**

The boldface data are the best results in their items

the state-of-the-art methods. Our results show that our method can achieve 32.6 %, a performance better than the following state-of-the-art methods: HOE [13] 19.5 %, LBP-TOP [12] 25.1 %, HOG 3D [14] 26.9 %, LSH-CORF [36] 21.8 %, 3D LUT [6] 25.2 %, and STM-ExpLet [29] 31.7 %.

To date, we have performed all the necessary experiments and covered all items relating to the latest works in FER. The proposed method, 3DCNN-DAP, and STM outperform the other methods. This means general learning frameworks lead to greater robustness with respect to intra-class variation and face deformation. Because the local descriptor-based methods, such as LBP-TOP, HOE, and HOG 3D, lack semantic meanings, they can hardly represent complex variations over mid-level facial action areas, so accuracy is difficult to achieve in methods based on facial action areas. However, to obtain the spatiotemporal property of expressions, 3DCNN-DAP and STM treat the time of the video as the third dimension, which limits the possible number of subject-independent applications. They can obtain good performance only in dynamic images. Hence, although the mean average precision of 3DCNN-DAP is almost the same as the average accuracy of our proposed method, its results sharply decline in the MMI database. In contrast, the proposed framework treats feature learning separately by dopting the subject-independent classifier to the final objective of classification. Since local features trained by classifiers can effectively cancel out the problems caused by semantic gap, which leads to an overall significant improvement of the classification performance [37–39]. Thus, the learned feature and classifier have specificity and discriminative capability. Therefore, the performance of the proposed framework is distinctive.

5 Conclusions

In this study, we proposed a novel cascade framework called the multithreading cascade of SURF (McSURF) for robust FER. The main contribution is our proposed multithreading cascade learning model, which allows multiple categories of data to be simultaneously trained. The concurrency of this multithreading learning model can extend the application range of cascades and represents a significant advance in related imaging industries.

Table 3 Recognition results for the AFEW database

Method	Accuracy on AFEW (%)							
	Ne	An	Di	Fe	Ha	Sa	Su	mAP.
LBP-TOP [12]	9.0	11.7	**19.6**	17.9	42.3	**23.8**	33.6	25.1
HOE [13]	6.1	11.2	16.5	9.0	33.5	15.3	28.3	19.5
HOG 3D [14]	–	–	–	–	–	–	–	26.9
3D LUT [6]	6.8	45.7	0	0	**62.0**	13.2	48.6	25.2
LSH-CORF [36]	–	23.1	12.8	**38.6**	9.7	21.1	10.9	21.8
STM-ExpLet [29]	–	–	–	–	–	–	–	31.7
McSURF (3T)	**16.3**	**47.3**	8.2	31.6	36.5	15.4	72.8	**32.6**
McSURF (OD)	15.3	46.5	7.6	31.2	34.7	14.6	**75.2**	32.2

The boldface data are the best results in their items

We used three popular and representative public databases in the FER research field to experimentally confirm the validity of the proposed method. Based on our experimental results, we analyzed the impact of face registration on both the learning and recognition stages, obtaining detailed answers on how face registration works on AdaBoost-based algorithms and why it can improve the robustness of these algorithms in FER applications. These issues are important to those with related research interests.

In future work, we will first attempt to improve the discriminative power of the multiple classification framework and investigate how feature representation errors impact recognition frameworks.

Authors' contributions

JC designed the core methodology of the study and carried out the implementation and experiments, and he drafted the manuscript. ZL assisted in doing the experiments. TT and YA participated in the study and helped to draft the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]RIEB, Kobe University, 2-1 Rokkodai, 657-8501 Kobe, Hyogo, Japan. [2]Graduate School of System Informatics, Kobe University, 1-1 Rokkodai, 657-8501 Kobe, Hyogo, Japan. [3]Organization of Advanced Science and Technology, Kobe University, 1-1 Rokkodai, 657-8501 Kobe, Hyogo, Japan.

Received: 26 October 2015 Accepted: 7 October 2016
Published online: 22 October 2016

References

1. P Viola, M Jones, Robust real-time face detection. Int. J. Comput. Vis. (IJCV). **57**(2), 137–154 (2004)
2. T Trzcinski, M Christoudias, V Lepetit, Learning image descriptors with boosting. IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI). **37**(3), 597–610 (2015)
3. D Chen, S Ren, Y Wei, X Cao, J Sun, in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Joint cascade face detection and alignment, (2014), pp. 109–122
4. B Wu, H Ai, C Huang, in *Proc. Audio-and Video-Based Biometric Person Authentication*. LUT-based AdaBoost for gender classification, (2003), pp. 104–110
5. Y Wang, H Ai, B Wu, C Huang, in *Proc. Int. Conf. Pattern Recognit. (ICPR)*. Real time facial expression recognition with AdaBoost, vol. 3, (2004), pp. 926–929
6. J Chen, Y Ariki, T Takiguchi, in *Proc. ACM Multimedia Conf. (MM)*. Robust facial expressions recognition using 3D average face and ameliorated AdaBoost, (2013), pp. 661–664
7. H Bay, A Ess, T Tuytelaars, LV Gool, Speeded-up robust features (SURF). Comput. Vis. Image Underst. (CVIU). **110**(3), 346–359 (2008)
8. P Lucey, JF Cohn, T Kanade, J Saragih, Z Ambadar, I Matthews, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*. The Extended Cohn-Kanade Dataset (CK+): a complete dataset for action unit and emotion-specified expression, (2010), pp. 94–101
9. MF Valstar, M Pantic, in *Proc. of Int. Conf. Language Resources and Evaluation, Workshop on EMOTION*. Induced disgust, happiness and surprise: an addition to the MMI Facial Expression Database, (2010), pp. 65–70
10. M Pantic, MF Valstar, R Rademaker, L Maat, in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*. Web-based database for facial expression analysis, (2005), pp. 317–321
11. A Dhall, R Goecke, S Lucey, T Gedeon, Collecting large, richly annotated facial-expression databases from movies. MultiMedia, IEEE. **19**(3), 34–41 (2012)
12. G Zhao, M Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI). **29**(6), 915–928 (2007)
13. L Wang, Y Qiao, X Tang, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Motionlets: Mid-level 3D parts for human motion recognition, (2013), pp. 2674–2681
14. A Klaeser, M Marszalek, C Schmid, in *Proc. British Machine Vis. Conf. (BMVC)*. A spatio-temporal descriptor based on 3D-gradients, (2008), pp. 99–19910
15. SW Chew, P Lucey, S Lucey, J Saragih, JF Cohn, S Sridharan, in *FG*. Person-independent facial expression detection using constrained local models, (2011), pp. 915–920
16. TF Cootes, GJ Edwards, CJ Taylor, Active appearance models. IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI). **23**(6), 681–685 (2001)
17. C David, C Tim, in *Proc. British Machine Vis. Conf. (BMVC)*. Feature detection and tracking with constrained local models, (2006), pp. 95–19510
18. C Ferri, PA Flach, J Hernández-Orallo, in *Proc. Int. Conf. Machine Learn. (ICML)*. Learning decision trees using the area under the ROC curve, (2002), pp. 139–146
19. DG Lowe, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Object recognition from local scale-invariant features, vol. 2, (1999), pp. 1150–1157
20. J Li, T Wang, Y Zhang, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*. Face detection using SURF cascade, (2011), pp. 2183–2190
21. N Dalal, B Triggs, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Histograms of oriented gradients for human detection, (2005), pp. 886–8931
22. R-E Fan, K-W Chang, C-J Hsieh, X-R Wang, C-J Lin, LIBLINEAR: A Library for Large Linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
23. P Long, R Servedio, in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*. Boosting the area under the ROC curve, (2007), pp. 945–952
24. SZ Li, Z Zhang, H-Y Shum, H Zhang, in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*. FloatBoost learning for classification, (2002), pp. 993–1000
25. R Xiao, H Zhu, H Sun, X Tang, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Dynamic cascades for face detection, (2007), pp. 1–8
26. L Bourdev, J Brandt, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Robust object detection via soft cascade, vol. 2, (2005), pp. 236–243
27. Q Zhu, M-C Yeh, K-T Cheng, S Avidan, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Fast human detection using a cascade of histograms of oriented gradients, vol. 2, (2006), pp. 1491–1498
28. A Dhall, R Goecke, J Joshi, K Sikka, T Gedeon, in *Proc. ACM The Int. Conf. on Multimodal Interaction (ICMI)*. Emotion recognition in the wild challenge 2014, (2014), pp. 461–466
29. M Liu, S Shan, R Wang, X Chen, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Learning Expressionlets on spatio-temporal manifold for dynamic facial expression recognition, (2014), pp. 1749–1756
30. E Rentzeperis, A Stergiou, A Pnevmatikakis, L Polymenakos, in *Proc. Artificial Intelligence Applications and Innovations (AIAI). Int. Federation for Inf. Proc. (IFIP)*. Impact of face registration errors on recognition, vol. 204, (2006), pp. 187–194
31. L Wiskott, J-M Fellous, N Kuiger, C von der Malsburg, Face recognition by elastic bunch graph matching. IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI). **19**(7), 775–779 (1997)
32. A Albiol, D Monzo, A Martin, J Sastre, A Albiol, Face recognition using HOG-EBGM. Pattern Recognition Letters. **29**(10), 1537–1543 (2008)
33. RA Khan, A Meyer, H Konik, S Bouakaz, Framework for reliable, real-time facial expression recognition for low resolution images. Pattern Recogn. Lett. **34"**(10), 1159–1168 (2013)
34. M Liu, S Li, S Shan, R Wang, X Chen, in *Proc. Asia Conf. Comput. Vis. (ACCV)*. Deeply learning deformable facial action parts model for dynamic expression analysis, vol. 9006, (2014), pp. 143–157
35. P Scovanner, S Ali, M Shah, in *Proc. ACM Multimedia Conf. (MM)*. A 3-dimensional SIFT descriptor and its application to action recognition, (2007), pp. 357–360
36. O Rudovic, V Pavlovic, M Pantic, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Multi-output Laplacian dynamic ordinal regression for facial expression recognition and intensity estimation, (2012), pp. 2634–2641
37. L Xie, Q Tian, M Wang, B Zhang, Spatial pooling of heterogeneous features for image classification. IEEE Trans. Image Proc.(TIP). **23**, 1994–2008 (2014)

38. J Chen, T Takiguchi, Y Ariki, A robust SVM classification framework using PSM for multi-class recognition. EURASIP J. Image Video Process. **2015**(1), 1–12 (2015)
39. J Sánchez, F Perronnin, T Mensink, J Verbeek, Image classification with the Fisher vector: theory and practice. Int. J. Comput. Vis. (IJCV). **105**(3), 222–245 (2013)