Facial Expression Recognition with Multithreaded Cascade of Rotation-invariant HOG

Jinhui Chen
Graduate School of System Informatics
Kobe University
Kobe, 657-8501, Japan
Email: ianchen@me.cs.scitec.kobe-u.ac.jp

Tetsuya Takiguchi
Graduate School of System Informatics
Kobe University
Kobe, 657-8501, Japan
Email: takigu@kobe-u.ac.jp

Yasuo Ariki
Graduate School of System Informatics
Kobe University
Kobe, 657-8501, Japan
Email: ariki@kobe-u.ac.jp

***Abstract*—We propose a novel and general framework, named the multithreading cascade of rotation-invariant histograms of oriented gradients (McRiHOG) for facial expression recognition (FER). In this paper, we attempt to solve two problems about high-quality local feature descriptors and robust classifying algorithm for FER. The first solution is that we adopt annular spatial bins type HOG (Histograms of Oriented Gradients) descriptors to describe local patches. In this way, it significantly enhances the descriptors in regard to rotation-invariant ability and feature description accuracy; The second one is that we use a novel multithreading cascade to simultaneously learn multiclass data. Multithreading cascade is implemented through non-interfering boosting channels, which are respectively built to train weak classifiers for each expression. The superiority of McRiHOG over current state-of-the-art methods is clearly demonstrated by evaluation experiments based on three popular public databases, CK+, MMI, and AFEW.**

***Keywords*—multithreading cascade; Ri-HOG; FER**

I. Introduction

Facial expression recognition (FER) is a typical multi-class classification problem in Affective Computing. Furthermore, since it is one of the most significant technologies for auto-analyzing human behavior, which can be widely applied to various application domains. Therefore, the need for this kind of technology in various different fields continues to propel related research forward every year.

Nevertheless, there are still many difficulties, because testees in images usually own variable appearances and the wide range of poses that they can adopt, in particular making their heads appear in a invariant orientation, which is an almost impossible task to overcome. Unfortunately, current approaches of FER usually ignore these problems and do not present a robust feature set and its corresponding robust classifying framework that allows the expression to be discriminated cleanly under these situations. Reviewing [1] makes it clear that this situation has not been well improved. Doing a further survey of the experimental reports in these works [2]–[7], we also find that the best precision achieved by any of these state-of-the-art methods is not more than 33.7%, when evaluated by some much challenging databases (*e.g.,* AFEW [8]). Therefore, FER is still an extremely challenging task in Affective Computing. The first need is a robust feature and the matching high-quality training framework.

In this paper, we propose a novel framework that adopts robust feature representation for training the multithreading boosting cascade. We adopt rotation-invariant HOG (Ri-HOG) as features, which is reminiscent of Dalal *et al.*'s HOG [9]. However, in this paper, we noticeably enhance the conventional HOG in rotation-invariant ability and feature extraction speed. We carry out a detailed study of the effects of various implementation choices in descriptor performance. We subdivide the local patch into annular spatial bins to achieve spatial binning invariance. Besides, we apply radial gradient to attain gradient binning invariance, which is inspired by Takacs *et al.*'s RGT (Radial Gradient Transform) [10].

The proposed learning model is derived from AdaBoost [11], but it is a novel, multi-class, simultaneous cascade; *i.e.*, a multithreaded one. There are many precursors who focus on boosting cascade research, such as, BinBoost [12], joint cascade [13] and SURF cascade [14] for facial detection, soft cascade [15] for object detection, and HOG cascade for detecting humans [16] *etc.* These are outstanding methods derived from Viola-Jones (V-J) framework [11], but the same as V-J framework, they only reached maturity, when used as detection applications. Based on these algorithms, there are seldom applications that succeed in FER, because current cascade models lack the robustness that allows the training framework to process simultaneous multi-class classifications smoothly. Therefore, we call these cascade models as the single-threaded boosting cascade, which is binary learning model. This learning model limits the application range of boosting training.

There is also another type of boosting training model (*e.g.*, Multi-class AdaBoost [17], [18] and LUT-AdaBoost [19]–[21]), which focuses on allowing the weak classifier to be trained to fit complex distributions. In other words, these classifiers can achieve multi-class recognition. However, they did not present an effective approach for improving the robustness of their classifiers further. As far as we know, it is still a challenging task using these methods because their algorithms cannot appropriately organize the ensemble of weak classifiers. Therefore, we summarize these approaches as the *"thin"* multi-boosting training, which limits the recognition ability of classifiers based on boosting training.

Differing from the single-threaded boosting cascade and

978-1-4799-9953-8/15/$31.00 ©2015 IEEE

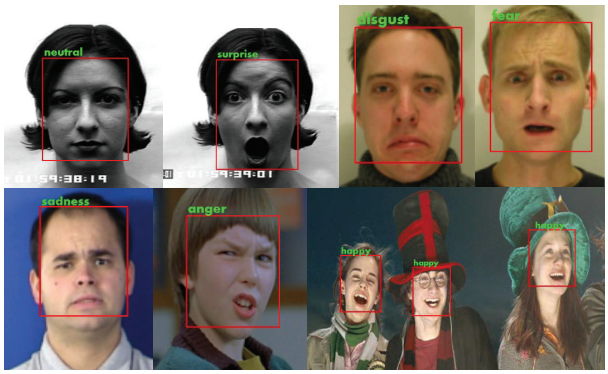


Fig. 1. Examples of facial expression recognition results.

the thin multi-boosting training model, we propose a novel and robust cascade algorithm, called Multithreading Cascade, to learn multiclass cascades with Ri-HOG simultaneously. Multithreading Cascade is implemented through configuring the AUC (Area under ROC curve) [22] of the weak classifier for each data category into a real-valued lookup list. These non-interfering lists are built into thread channels where the related boosting cascade can train each data category classifiers individually. In this way, boosting cascade-based approaches can be trained to fit complex distributions and can simultaneously process multi-class events much robustly. In this paper, the proposed framework is applied to FER. In experiments, the proposed framework is evaluated on three public expression database, covering both of the lab-controlled scenarios and real-world situations. Some examples of expression recognition results are shown in Figure 1. The experiments show that the proposed method can construct a robust FER system whose results outperform the well-known state-of-the-art methods on FER.

Our main contribution is that we develop a framework (McRiHOG) that can simultaneously learn multiclass classifiers for FER. In so doing, we have these contributions: 1) Generally, the boosting classifier is trained as binary classification models. We propose a multithreading cascade learning model, which allows the multiple categories data to be simultaneously trained on cascade learning model; 2) The McRiHOG is excellent method for FER application. Its performance experimentally outperforms many the state-of-the-arts methods; 3) Derived from Takacs *et al.*'s approach [10], we use magnitudes of radial gradients to represent HOG features,in this way, we can enhance the HOG features descriptors in regard to invariant representation ability. These are very important to those with closely related research interests.

In the remainder of this paper, we describe the proposed method in Sect. II. Sect. III gives the detailed stages of process in experiments and conclusions are drawn in Sect. IV.

II. Proposed Method

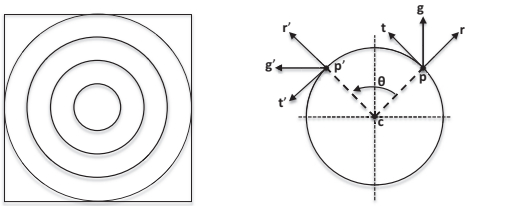
This section describes the proposed framework, which has these ingredients: the Ri-HOG features for local patch description; logistic regression based weak classifiers, which are also combined with AUC as a single criterion for cascade convergence testing; and multithreading cascade for fitting multiplex categories boosting training. We separately discusses these approaches in this section.

A. Feature description

HOG are feature descriptors, which are computed on a dense grid of uniformly-spaced cells and use overlapping local contrast normalization for improved accuracy. This features set based on *cells* and *blocks* representation system is widely used in object detection, especially human detection. The describing ability of HOG features set outperforms many existing features [16], however, its robustness against image rotation does not reach maturity. Therefore, there many researchers have tried to improve the robustness of HOG. Currently, two of the most popular and representative ones are 2D HOG [25] and HOG 3D [7], which are interesting solutions to the rotation problems.

Nevertheless, the bottleneck problems also exist in these approaches: 2D HOG descriptor is inspired by Jhuang *et al.*'s approaches [26] that use 2D Gabor-filter responses combined with optical flow. Such dense representations avoid some of the problems discussed above, but cannot solve these problems completely. Moreover, it brings further more time complexity because 2D HOG requires a region of interest (ROI) around the task region, which is usually obtained by using either a separate detector or background subtraction followed by blob detection; Inspired by SIFT descriptor [27], HOG 3D constructs a platonic solids system using auxiliary coordinate system to achieve the intention of invariant feature representation. It is an interesting solution yet with high computational time and memory cost. Although they further distribute their task images (faces) over the 2D polar coordinates and make all task images be congruent in order to reduce memory cost, the computing speed is still a bottleneck. Furthermore, HOG 3D have to rely on the integral videos [7], which limits HOG 3D in some restricted application areas. Therefore, these approaches cannot be considered as complete solutions to the above problems.

In this paper, we adopt radial gradient to represent the gradient for HOG descriptors, which is derived from Takacs *et al.*'s rotation-invariant image features [10]. But different from Takacs *et al.*'s approach, we only use the radial gradient to replace the Gaussian gradient function of conventional HOG. We subdivide the local patch into annular spatial cells (see Fig. 2(a)). How to calculate these descriptors is shown in Fig. 2. In Fig. 2(b), $\forall$ a point p in the circle c, the task is to compute the radial gradient magnitude of point p (x, y). Decompose vector g into its local coordinate system as $(g^T r, g^T t)$, by projecting g into the r and t orientations as shown in Fig. 2(b). Because the component vectors of g in r and t orientations can be quickly obtained by $r = \frac{p-c}{\|p-c\|}$, $t = R_{\frac{\pi}{2}} r$, where we can obtain the gradient g easily on the gradient filter. In addition, R_θ is the rotation matrix by angle θ. About why the representation system based on radial gradient and annular



(a) Annular spatial cells (b) Radial gradient transform

Fig. 2. Illustration of rotation-invariant HOG descriptors.

spatial cells is rotation-invariant, please refer to Takacs *et al.*'s work [10] for the detailed verification.

Since Takacs *et al.* focus on image tracking applications, the speed is more important, they use Approximate RGT and ROC curve to compute the feature descriptors [10]. However, in so doing, it will decrease the distinctiveness of feature descriptors for recognition applications. In order to keep the distinctiveness of feature descriptors for recognition application, we do not follow Takacs *et al.*'s way to abandon gradient magnitudes, cells, and blocks representation system. Therefore, essentially, the feature (Ri-HOG) that we adopt here is an improved HOG feature, but the approach proposed by Takacs *et al.* is a very excellent and novel feature representation method for image tracking applications, which cannot be considered as a type of HOG feature. Ri-HOG persists and develops the discriminative representation of conventional HOG features. Meanwhile, it also can significantly enhances the descriptors in regard to rotation-invariant ability. Simply, we use the following four steps to extract Ri-HOG descriptors:

1. Subdivide the local patch into annular spatial cells as shown in Fig. 2(a);
2. Calculate the radial gradient $(g^T r, g^T t)$ of each pixel in the cell;
3. Calculate the gradient magnitudes and the orientations of radial gradients using the Eq. 1:

$$\begin{aligned} M_{GRT}(x,y) &= \sqrt{(g^T r)^2 + (g^T t)^2}, \\ \theta(x,y) &= arctan\frac{g^T t}{g^T r}; \end{aligned} \tag{1}$$

4. Accumulating the gradient magnitude of radial gradient for each pixel over the annular spatial cells into 9 bins, which are separated according to the orientation of radial gradient. In this way, we can extract the feature descriptors from a dense annular spatial bin of these uniformly spaced cells.

About the normalization, we tried all of 4 approaches listed by Dalal *et al.* in [9]. In practice, L_2-Hys, L_2 normalization followed by clipping is shown working best. The recognition template is 100×100 with 10 cells, and it allows the patch size ranging from 50×50 pixels to 100×100 pixels. We slide the patch over the recognition template with 5 pixels forward to ensure enough feature-level difference. We further allow different aspect ratio for each patch (the ratio of width and height). The descriptors are extracted according to the order from the inside to the outside of cells. Hence, concatenating descriptors in 10 cells together yield a 90-dimensional feature vector.

B. Weak Classifier Construction

One hand, we build a weak classifier over each local patch described by the rotation-invariant HOG descriptor, and pick optimum patches in each boosting iteration from the patch pool. On the other hand, we construct the weak classifier for each local patch by logistic regression to fit our classifying framework, due to that it is a linear classifier with probability. Given a Ri-HOG feature $\mathbb{F}$ over local patch, logistic regression defines a probability model:

$$P(q|\mathbb{F}, \mathbf{w}) = \frac{1}{1 + \exp(-q(\mathbf{w}^T\mathbb{F} + b))}, \tag{2}$$

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

$$\sum_{k=1}^{K} \log(1 + \exp(-q_k(\mathbf{w}^T\mathbb{F}_k + b))) + \lambda \left\|\mathbf{w}\right\|_p, \tag{3}$$

where λ denotes tunable parameter for the regularzation term, and $\left\|\mathbf{w}\right\|_k$ means L_k norm of the weight vector. Note that it is also applied to L_2-loss and L_1-loss linear SVMs by well known open source code LIBLINEAR [28]. Therefore, this problem can be solved on algorithms in [28].

C. Multithreaded Cascade

1) Multithreaded Cascade Channel Construction: Assuming there are total N boosting iteration rounds, given weak classifiers $h_i^{(n)}$ for category i data, the strong classifier is defined as $H_i^{(N)}(\mathbb{F}) = \frac{1}{N}\sum_{n=1}^{N} h_i^{(n)}(\mathbb{F})$. 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 test model will produce a highest AUC score [22], [29] $J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k))$. *i.e.*,

$$S_i^{(n)} = \max_{k=1,\cdots K} J(H_i^{(n-1)}(\mathbb{F}) + h_i^{(n)}(\mathbb{F}_k)). \tag{4}$$

This procedure is repeated until the AUC score is converged, or the designed 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 will build an AUC score pool for each one class of object.

In order to learn multi-class classifiers simultaneously, we adopt these AUC data to construct independent channels for boosting learning. The details are summarized as follows:
1. Assuming 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 $\mathbf{C} = \{\text{bin}_j = [\frac{(j-1)}{M}, \frac{j}{M}] | \ j = 1, \cdots, M\}$. In each channel, we will build an independent boosting model for training the classifiers which can recognize a corresponding category task;
2. Set $u = S_i(\mathbb{F}, x)$ and define the weak classifier $h_i(x)$ as follows:

$$\begin{aligned} &\textbf{if } u \in \ \mathbf{C} \text{ and } x \in \{\text{category } i \text{ samples}\}, \\ &\textbf{then } h_i(x) = 2P(q|\mathbb{F}, \mathbf{w}) - 1. \end{aligned} \tag{5}$$

These will guarantee the precision of h is more than 0.5;
3. Given the characteristic function

$$B^{(j,i)}(u, \mathbf{Y}) = \begin{cases} 1 & u \wedge \mathbf{Y} = i \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

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

$$h(\mathbb{F}) = \sum_{j=1}^{M} \sum_{i=1}^{M} \left(2P(q|\mathbb{F}, \mathbf{w}) - 1\right) B^{(j,i)}(u, \mathbf{Y}). \tag{7}$$

Using the above approaches, M independent channels can be constructed. Meanwhile, the classifier category is able to be judged and auto-selected into the related channel. In this way, we can learn the classifiers on Algorithm 1 and train multithreaded boosting cascades simultaneously in their training channels via Algorithm 2.

2) Learning Weak Classifiers: Like most existing multiclass classification algorithms, our approach is crucially dependent on the labeled data of sample space to learn the classifiers. In this paper, we adopt this approach to combine 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), define $Y(i)$ as

$$Y(i) = \begin{cases} 1 & \text{if } \ i \in Y \\ -1 & \text{if } \ i \notin Y \end{cases}, \tag{8}$$

where $x \in \mathbf{X}, l \in \mathbf{Y}, Y \subseteq \mathbf{Y}$. In order to avoid overfitting, we restricted the number of used samples during training as in [30]. In practice, we sampled an active subset from the whole training set according to the boosting weight. It is generally good to use about $30 \times p$ samples of each class, where p is multiple coefficient (Algorithm 1 step 3.a).

Algorithm 1 Learning Boosting Classifiers on Ri-HOG.

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 τ 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 Ri-HOG 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 previous stage: $S_i^{(n-1)}$ and each $h_i(\mathbb{F}_k)$;
   d. Choose the best model $S_i^{(n)}$ which contains the best weak classifier $h_i(\mathbb{F}_j)$, according to the Eq. 4;
   f. Update 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, on which it can make the weight follow to $\sum w^+ = 1$ and $\sum w^- = 1$;
   g. 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) Boosting Cascade Training: To the best of our knowledge, almost all existing cascade detection frameworks are trained based on two conflicted criteria, *i.e.* false-positive-rate (FPR) f_j and hit-rate (or recognition rate) r_j for the detection-error tradeoff. The overall FPR of a $T-$stage cascade is $F = \prod_{j=1}^{T} f_j$, while the overall hit-rate is $R = \prod_{j=1}^{T} r_j$. Inspired by [22] and [14], here we introduce AUC as a single criterion for cascade convergence testing, which will realize adaptive FPR among different stages (details about literature description for AUC, please refer to [22]). Hence, combined with logistic regression based weak classifiers to adopt Ri-HOG features, this approach can yield fast convergence speed and cascade model with much shorter stages.

Within one stage, we did not need to give threshold for intermediate weak classifiers. We just need to determine each decision threshold θ_i for $i-$th emotional category in its threading channel. In our case, using ROC curve, FPR of each emotional category is easily determined when given the minimal hit-rate $d_i^{(min)}$. We decreased $d_i^{(j)}$ 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 desire θ_i. After

Algorithm 2 Training Multithreaded 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 II-C1;
    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.

one stage of classifiers learning is converged via Algorithm 2, we continue to train another one with false-positive samples coming from scanning non-target images with partial trained cascade. We repeat this procedure until the overall FPR reach the goal.

III. Experiments

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

A. Databases and Protocols

The proposed framework is evaluated on three public databases, *i.e.*, CK+, MMI and AFEW, covering the lab-controlled scenarios (CK+ and MMI) and the real-world one (AFEW).

CK+ DB It is a set of facial expression samples posed by 123 people. There are 327 sequences, which are found from 593 sequences to meet the criteria for 1 of 7 discrete emotions (Anger, Contempt, Disgust, Fear, Happiness, Sadness, and Surprise) based on FACS [23]. In our experiments, we divided these samples into several groups for each expression by the person-independent rule, and each group included 10 posers. Person-independent 10-fold cross-validation had been done for comparing with some outstanding current methods.

Fig. 3. Top-3 local patches picked by training procedure in the green-red-blue order on AFEW database.

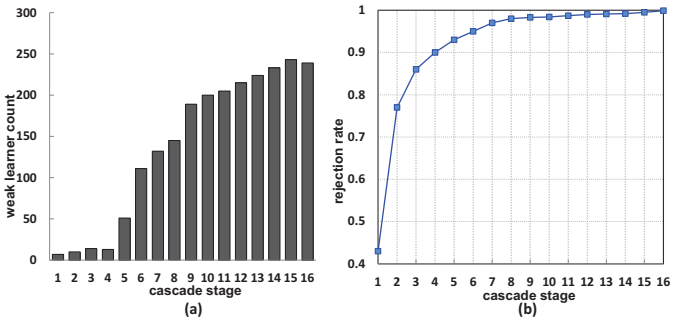


Fig. 4. (a) The number of weak classifiers at each cascade stage; (b) the accumulated rejection rate over all stages.

MMI DB MMI is a public database that includes over 30 subjects in which female-male ratio is near 11:15. The subjects' age from 19 to 62, they are European, Asian or South American *etc*. 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 6 expressions from the MMI dataset.

AFEW DB The evaluation experiments have done using AFEW [8], which is a much challenging task. All of the sets in AFEW have been collected from movies to depict so-call wild scenario. In this paper, we adopted its version of 2013, which was used as the criteria database of *EmotiW 2013* [1], because the evaluation results of many state-of-the-art methods are based on version 2013. We trained version 2013's training set and the results are reported on its validation set, which is the same way as the the latest FER work [2] doing.

We used all training samples in AFEW training set and collected training samples from CK+ and MMI according to the person-independent 10-fold cross-validation rule. In order to reduce the process time of training, the samples from three datasets were trained together. In order to enhance the generalization performance of boosting learning, we dealt with the training samples by some transformations (mirror reflection, rotate the images *etc*.), finally, the original samples were increased by a factor of 64. In the training stages, the training data of current processing expression were adopted as positive sample data; the other expressions' data were used for negative data.

TABLE I
RECOGNITION RESULTS ON CK+ AND MMI.

Method	Accuracy on CK+ (%)								Accuracy on MMI(%)						
	An	Co	Di	Fe	Ha	Sa	Su	Ave.	An	Di	Fe	Ha	Sa	Su	Ave.
CLM [32]	70.1	52.4	92.5	72.1	94.2	45.9	93.6	74.4	–	–	–	–	–	–	–
HOE [5]	76.4	65.4	83.6	73.3	92.1	88.6	92.8	82.3	46.4	58.3	33.2	62.6	60.8	65.1	55.5
LBP-TOP [4]	82.2	77.8	91.5	72.0	98.6	57.1	97.6	82.4	58.1	56.3	53.6	78.6	46.9	50.0	57.2
ITBN [6] (15-flod)	91.1	78.6	94.0	83.3	89.8	76.0	91.3	86.3	46.9	54.8	57.1	71.4	65.6	62.5	59.7
HOG 3D [7]	84.4	77.8	**94.9**	68.0	**100**	75.0	**98.8**	85.6	61.3	53.1	39.3	78.6	43.8	55.0	55.2
LSH-CORF [3]	71.3	–	90.8	79.0	92.6	**90.5**	96.6	86.8	59.6	**71.4**	62.3	68.9	**70.3**	75.1	61.8
3D LUT [21]	76.3	35.1	60.5	73.8	91.0	48.2	92.8	68.2	43.3	55.3	56.8	71.4	28.2	77.5	47.2
3DCNN-DAP [31]	91.1	66.7	96.6	80.0	98.6	85.7	96.4	87.9	64.5	62.5	50.0	**85.7**	53.1	57.5	62.2
STM [2]	–	–	–	–	–	–	–	91.1	–	–	–	–	–	–	65.4
McRiHOG	**94.3**	**82.9**	92.7	**91.5**	93.1	81.6	97.3	90.5	**68.9**	48.0	**80.1**	82.4	52.4	**86.9**	**71.4**

TABLE II
RECOGNITION RESULTS ON AFEW.

Method	Accuracy on AFEW (%)						
	An	Di	Fe	Ha	Sa	Su	ave.
HOE [5]	11.2	16.5	9.0	33.5	15.3	28.3	19.0
LBP-TOP [4]	11.7	**19.6**	17.9	42.3	**23.8**	33.6	24.8
HOG 3D [7]	–	–	–	–	–	–	26.9
LSH-CORF [3]	23.1	12.8	**38.6**	9.7	21.1	10.9	19.4
3D LUT [21]	45.7	0	0	**62.0**	13.2	48.6	28.2
STM [2]	–	–	–	–	–	–	31.7
McRiHOG	**68.2**	0	**48.1**	**83.3**	**32.0**	**91.6**	**53.6**

TABLE III
AVE. PRECISION USING DIFFERENT FEATURES.

Database	Precision of feature (%)				
	SIFT	SURF	Haar	HOG	Ri-HOG
CK+	82.6	72.2	68.6	77.3	**90.5**
MMI	65.4	46.0	42.2	58.8	**71.4**
AFEW	41.5	35.8	17.3	32.4	**53.6**

B. Speed Evaluation Results

Training Speed: We replaced 40 types of the local patches on the 100×100 detection template as described in subsection II-A. The proposed method used 377 minutes to converge at the $16th$ iteration stage. The cascade detector contained $2,394$ classifiers of all categories, and only need to evaluate 1.5 HOG per window. After training, we observed that the top-3 picked local patches for FER laid in the regions of two eyes and mouth. This situation is similar to Haar-based classifiers [21], see the examples in Fig. 3.

More details for cascade of FER are illustrated in Fig. 4(a) and Fig. 4(b), which include the number of weak learners in each stage and the average accumulated rejection rate over the whole cascade stages. It shows that the first 8 stages have rejected 98% of the non-current class samples.

C. Recognition Results Comparison

The comparison methods were selected to represent the state-of-the-art level of this field, which includes proposing for the improvement of local spatiotemporal descriptors: such as LBP-TOP [4], HOE [5], HOG 3D [7], which are very popular for FER, while 3DCNN-DAP [31] and STM [2] are the latest ones; also including those methods that focus on enhancing the robustness of their classifying frameworks or making the frameworks can be encoded robustly, like, ITBN [6], 3D LUT [21] and LSH-CORF [3] *etc*. For fair comparison with them, we used the same databases, which were evaluated via the standardized items what they had done.

Table I and Table II compares our method with these state-of-the-art methods. Furthermore, almost of these meothods were conducted using their released codes and the parameters had been tuned to better-adapt for our experiments. However, about some methods, because we cannot obtain their source codes until now (*e.g.* STM [2] and 3DCNN-DAP [31], *etc.*), thus, we have to cite the reported results from the related works. The precisions of our framework (McRiHOG) were 90.5% and 71.4% using CK+ and MMI, and 48.6% on AFEW. The state-of-the-art levels were improved 6% and 21.9% respectively by the proposed framework on MMI and AFEW. In addition, the recognition speed of the proposed framework reached 38 frames per second (FPS).

To date, all the necessary experiments have been carried out, but we still have a query why we have to adopt Ri-HOG as features. The reason is shown in Table III; *i.e*, it dominates others on the accuracy. Meanwhile, its recognition speed can meet the real-time recognition. However, adopting SIFT as features, the real-time recognition is an impossible task (speed: only 12 FPS), although the performance of the proposed framework with SIFT is also quite excellent.

IV. CONCLUSION

In this paper, we have proposed a novel cascade framework (McRiHOG) for robust FER. The main contribution in this paper is that: we propose a multithreading cascade learning model, which allows the multiple categories data to be simultaneously trained on cascade learning model. The concurrency of multithreaded learning model can extend the application range of cascade, which is significant to the related imaging industries. We have used three very popular and representative public databases in FER research field, to experimentally confirm the validity of the proposed method. About the future work, we will attempt to study the question about how does the feature representation error impact on recognition frameworks.

REFERENCES

[1] J. J. K. S. Abhinav Dhall, Roland Goecke and T. Gedeon, "Emotion recognition in the wild challenge 2014," in *Proc. ACM The Int. Conf. on Multimodal Interaction (ICMI)*, 2014.

[2] M. Liu, S. Shan, R. Wang, and X. Chen, "Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 1749–1756.

[3] O. Rudovic, V. Pavlovic, and M. Pantic, "Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 2634–2641.

[4] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 29, no. 6, pp. 915–928, June 2007.

[5] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2013, pp. 2674–2681.

[6] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. ACM Multimedia Conf. (MM)*, 2007, pp. 357–360.

[7] A. Klaser and M. Marszalek, "A spatio-temporal descriptor based on 3d-gradients," in *Proc. British Machine Vis. Conf. (BMVC)*, 2008, pp. 275:1–10.

[8] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Collecting large, richly annotated facial-expression databases from movies," *MultiMedia, IEEE*, vol. 19, no. 3, pp. 34–41, July 2012.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2005, pp. 886–893 vol. 1.

[10] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Fast computation of rotation-invariant image features by an approximate radial gradient transform," *IEEE Trans. Image Proc.*, vol. 22, no. 8, pp. 2970–2982, Aug. 2013.

[11] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis. (IJCV)*, vol. 57, no. 2, pp. 137–154, 2004.

[12] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning Image Descriptors with Boosting," *IEEE Trans. Patt. Analy. Mach. Intell. (TPAMI)*, vol. 37, no. 3, pp. 597–610, March 2015.

[13] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 109–122.

[14] J. Li, T. Wang, and Y. Zhang, "Face detection using SURF cascade," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV) Workshops*, Nov 2011, pp. 2183–2190.

[15] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, June 2005, pp. 236–243.

[16] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, 2006, pp. 1491–1498.

[17] F. Lv and R. Nevatia, "Recognition and segmentation of 3-D human action using hmm and multi-class adaboost," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 359–372.

[18] Y. Sun, S. Todorovic, and J. Li, "Unifying multi-class adaboost algorithms with binary base learners under the margin framework," *Pattern Recognition Letters*, vol. 28, no. 5, pp. 631 – 643, 2007.

[19] B. Wu, H. Ai, and C. Huang, "LUT-based adaboost for gender classification," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 104–110.

[20] Y. Wang, H. Ai, B. Wu, and C. Huang, "Real time facial expression recognition with adaboost," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2004, pp. 926–929.

[21] J. Chen, Y. Ariki, and T. Takiguchi, "Robust facial expressions recognition using 3 D average face and ameliorated adaboost," in *Proc. ACM Multimedia Conf. (MM)*, 2013, pp. 661–664.

[22] C. Ferri, P. A. Flach, and J. Hernández-Orallo, "Learning decision trees using the area under the roc curve," in *Proc. Int. Conf. Machine Learn. (ICML)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 139–146.

[23] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, 2010, pp. 94–101.

[24] M. F. Valstar and M. Pantic, "Induced disgust, happiness and surprise: an addition to the mmi facial expression database," in *Proceedings of Int'l Conf. Language Resources and Evaluation, Workshop on EMOTION*, May 2010, pp. 65–70.

[25] C. Thurau and V. Hlavac, "Pose primitive based human action recognition in videos or still images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2008, pp. 1–8.

[26] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[27] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *Int. J. Comput. Vis. (IJCV)*, vol. 60, no. 2, 2004, pp. 91–110.

[28] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[29] P. Long and R. Servedio, "Boosting the area under the roc curve," in *Proc. Adv. Neural Inf. Proc. Syst. (NIPS)*, 2008, pp. 945–952.

[30] R. Xiao, H. Zhu, H. Sun, and X. Tang, "Dynamic cascades for face detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct 2007, pp. 1–8.

[31] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, "Deeply learning deformable facial action parts model for dynamic expression analysis," in *Prof. Asia Conf. Comput. Vis. (ACCV)*, Nov. 2014.

[32] S. Chew, P. Lucey, S. Lucey, J. Saragih, J. Cohn, and S. Sridharan, "Person-independent facial expression detection using constrained local models," in *FG*, March 2011, pp. 915–920.