
# INTELLIGENT DETECTION OF FACIAL EXPRESSION BASED ON IMAGE

[a] Shaoping Zhu, [a,b] and Yongliang Xiao
[a] Department of Information Management Hunan University of Finance and Economics, 410205 Changsha, China
[b] School of Information Science and Engineering, Central South University, 410083, China
Emails: zhushaoping_cz@163.com

***Submitted: Oct. 2, 2014*** ***Accepted: Jan. 24, 2015*** ***Published: Mar. 1, 2015***

***Abstract- Human facial expressions detection plays a central role in pervasive health care and it is an active research field in computer vision. In this paper, a novel method for facial expression detection from dynamic facial images is proposed, which includes two stages of feature extraction and facial expression detection. Firstly, Active Shape Model (ASM) is used to extract the local texture feature, and optical flow technique is determined facial velocity information, which is used to characterize facial expression. Then, fusing the local texture feature and facial velocity information get the hybrid characteristics using Bag of Words. Finally, Multi-Instance Boosting model is used to recognize facial expression from video sequences. In order to be learned quickly and complete the detection, the class label information is used for the learning of the Multi-Instance Boosting model. Experiments were performed on a facial expression dataset built by ourselves and on the JAFFE database to evaluate the proposed method. The proposed method shows substantially higher accuracy at facial expression detection than has been previously achieved and gets a detection accuracy of 95.3%, which validates its effectiveness and meets the requirements of stable, reliable, high precision and anti-interference ability etc.***

**Index terms*:* Facial expression, ASM model, Optical flow model, Bag of Words, *Multi-Instance Boosting model*.**

# I. INTRODUCTION

Facial expression is the most expressive way humans display emotions. It delivers rich information about human emotion and provides an important behavioral measure for studies of emotion and social interaction et al. Human facial expression plays an important role in human communications. Facial expressions detection based on vision closely relates to the study of psychological phenomena and the development of human-computer interaction. It is an important addition to computer vision research at present, and has numerous significant theoretic and practical values. It has been widely applied in human-computer interfaces, human emotion analysis, medical care and cure, public security, and financial security, such as real-time video surveillance, bank cryptography and so on.

Automatically recognizing facial expression has recently become a promising research area. There are many researches already carried out to recognize facial expressions from video sequence. Cohen et al. [1] proposed a new architecture of hidden Markov models (HMMs) for automatically segmenting and recognizing human facial expression from video sequences. Morishima and Harashima [2] used emotion space to recognize facial expression. Shan et al. [3] put forward robust facial expression detection using local binary patterns. Aleksic et al. [4] used facial animation parameters and multistream HMMs for automatic facial expression detection. Neggaz et al. [5] proposed the improved Active Appearance Model (AAM) to recognize facial expressions of an image frame sequence. SVM has become one of the most popular research directions in the field of machine learning, and achieved widely research and application successfully in many fields [6]. Zhao and Pietikäinen [7] extended the LBP-TOP features to multi-resolution spatiotemporal space for describing facial expressions and used support vector machine (SVM) classifier to select features for facial expressions detection. Shih et al. [8] combined 2D-LDA and SVM to recognize facial expressions. Siyao Fu et al. [9] proposed a Spiking Neural Network based Cortex-Like Mechanism and Application for facial expression detection. Shan et al. [10] empirically evaluated facial representation based on local binary patterns(LBP) for facial expression detection in 2009.

Facial expression detection based on vision is a challenging research problem. However, these approaches have been fraught with difficulty because they are often inconsistent with other evidence of facial expressions [11]. It is essential for intelligent and natural human computer

interaction to recognize facial expression automatically. In the past several years, significant efforts have been made to identify reliable and valid facial indicators of expressions. L Wang et al. [12] used Active Appearance Models (AAM) to decouple shape and appearance parameters from face images, and used SVM to classify facial expressions. In [13,14], Prkachin and Solomon validated a Facial Action Coding System (FACS) based measure of pain that could be applied on a frame-by-frame basis. Most must be performed offline, which is both timely and costly, and makes them ill-suited for real-time applications. In [15], Zhang et al. combined the advantages of Active Shape Model (ASM) with Gabor Wavelet to extract efficient facial expressions feature and proposed ASM+GW model for facial expression detection. Zhang [16] used supervised locality preserving projections (SLPP) to extract facial expression features, and multiple kernels support vector machines (MKSVM) is used for facial expression detection. Methods described above use static features to characterize facial expression, but these static features cannot fully represent facial expressions.

However, evaluation results and practical experience have shown that facial expression automatically technologies are currently far from mature. Many challenges are to be solved before it can implement a robust practical application. In this paper, we propose a method for automatically recognizing facial expressions from video sequences. This approach includes extracting facial expression features and classifying facial expressions. In the extracting feature, we use Active Shape Model (ASM) for facial local texture feature and motion descriptor based on optical flow for facial velocity features. Then, these two features are integrated and converted to visual words using “bag-of-words” models, and facial expression is represented by a number of visual words. Final, the Multi-Instance boosting model is used for facial expression detection. In addition, in order to improve the detection accuracy, the class label information is used for the learning of the Multi-Instance boosting model.

Given unlabeled facial video sequence, our goal is to automatically learn different classes of facial expressions present in the data, and apply the Multi-Instance boosting model for facial expressions categorization and detection in the new video sequences. Our approach is illustrated in figure 1.

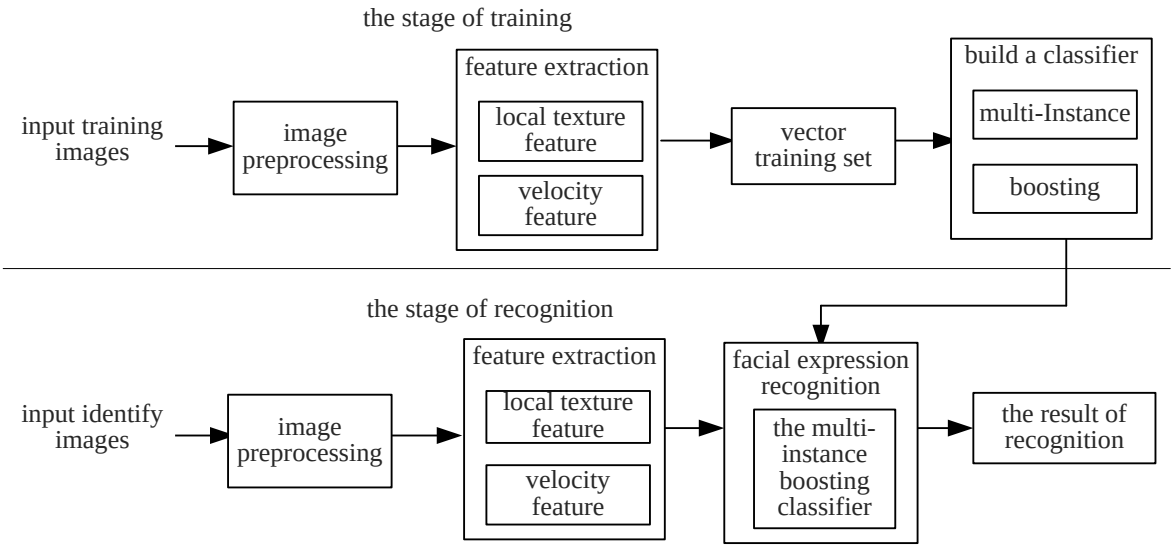


Figure 1. Flowchart of facial expression detection based on ASM and Multi-Instance Boosting

The paper is structured as follows. After reviewing related work in this section, we describe the facial expression feature representation base on ASM model, optical flow technique and "bag-of-words" models in section 2. Section 3 gives details of Multi-Instance boosting model for recognize facial expression. Section 4 shows experiment result, also comparing our approach with three state-of-the-art methods, and the conclusions are given in the final section.

## II. FACIAL EXPRESSION REPRESENTATION

Deriving an effective facial representation from original face images is a vital step for successful facial expression detection. Due to great changes in the dynamic characteristics and lack of constraints, this paper proposes ASM model for facial local texture feature extraction and optical flow model for facial velocity features, which can describe facial expression effectively.

a. Local texture feature extraction

Active Shape Model (ASM) [17] can iteratively adapt to refine estimates of the pose, scale and shape of models of image objects, and is a powerful method for refining estimates of object shape and location. It uses Point Distribution Models (PDM) [18] to derive from sets of training examples. This model represents objects as sets of labeled points. An initial estimate of the location of the model points in an image is improved by attempting to move each point to a better position nearby. Adjustments to the pose variables and shape parameters are calculated. Limits

are placed on the shape parameters ensuring that the example can only deform into shapes conforming to global constraints imposed by the training set. An iterative procedure deforms the model example to find the best fit to the image object. The steps can be briefly described as follows:

First, given a calibration facial key feature point training set $X$.

$$X = \{(I_i, s_i) \mid i = 1, 2, \cdots, \gamma; s_i = (x_i^1, y_i^1, \cdots, x_i^\beta, y_i^\beta)^T\}, \quad (1)$$

where $\gamma$ is the number of training sample, $\beta$ is the number of predefined key feature points, $s_i$ is Shape vector in training set $X$, which is concatenated by predefined and manual calibration $\beta$ key feature points of horizontal ordinate on the training images $I_i$.

Then, all shapes in training set are aligned to the same coordinate system by shape alignment algorithm, and get feature set $X'$ after alignment.

$$X' = \{s_i' \mid i = 1, 2, \cdots, \gamma\}, \quad (2)$$

These alignment shape are analysized by PCA, get the active shape model as follow:

$$X = \bar{X} + p_s b_s, \quad (3)$$

Where $\bar{X}$ is the average shape, $b_s$ for the shape parameter, $p_s$ is eigenvector of main component of transformation matrix, which is obtained by the training set of the eigenvalues of the covariance matrix decomposition. Eigenvector of main component reflects the main mode of shape change. Any shape can be approximately expressed by the average shape deformation, which is modeled by the shape parameter $b_s$ to sum several model weighted. $-3\sqrt{\lambda_i} < b_i < 3\sqrt{\lambda_i}$, $i = 1, 2, \cdots, \gamma$, where $\lambda_k$ is the eigenvalues of the covariance matrix, $\lambda_k \geq \lambda_{k+1}, \lambda_k \neq 0, k = 1, 2, \cdots, 2l$.

Finally, we calculate markov distance to determine the best matching position by analyzing the gray information of neighborhood.

Given local texture model:

$$\bar{G}_{ij} = \frac{1}{\gamma} \sum_{i=1}^{\gamma} G_{ij}, \quad (4)$$

$$T_{G_{ij}} = \frac{1}{\gamma - 1} \sum_{i=1}^{\gamma} (G_{ij} - \bar{G}_{ij})(G_{ij} - \bar{G}_{ij})^T, \quad (5)$$

Where $\bar{G}_{ij}$ is the average texture, $G_{ij}$ is the texture vector after the gray level information of *j*-th fixed point normalizes in the *i*-th training image.

$$G_{ij} = \frac{1}{\sum_{j=1}^{2k+1} \left| d_{g_{ij}} \right|} d_{g_{ij}}, \qquad (6)$$

where $d_{g_{ij}} = [g_{ij,2} - g_{ij,1}, \cdots, g_{ij,2k+1} - g_{ij,2k}]$, $g_{ij}$ is the gray information of the *i*-th feature points, which is the gray level of *k* points from each up and down along the normal direction with feature points as the center, $T_{G_{ij}}$ is the covariance matrix.

Calculate markov distance as follow:

$$d(G'_{ij}) = (G'_{ij} - \bar{G}_{ij})^T (T_{G_{ij}})^{-1} (G'_{ij} - \bar{G}_{ij}), \qquad (7)$$

where $G'_{ij}$ is the normalized vector texture by sampling near the j-th point of target search images.When $d(G'_{ij})$ takes minimum value, the corresponding point is the best candidate.

b. Facial velocity feature extraction

Optical flow-based face representation has attracted much attention [19]. According to the physiology, the expression is a dynamic event; it must be represented by the motion information of a face. So we use facial velocity features to characterize facial expression. The facial velocity features (optical flow vector) are estimated by optical flow model, and each facial expression is coded on a seven level intensity dimension (A–G): “anger”, “disgust”, “fear”, “happiness”, “neutral”, “sadness” and “surprise”.

Given a stabilized video sequence in which the human face appears in the center of the field of view, we compute the facial velocity (optical flow vector) $\upsilon = (\upsilon_x, \upsilon_y)$ at each frame using optical flow equation, which is expressed as:

$$I_x \upsilon_x + I_y \upsilon_y + I_t = 0, \qquad (8)$$

where $I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}, \quad \upsilon_x = \frac{dx}{dt}, \upsilon_y = \frac{dy}{dt}$,

$(x, y, t)$ is the image in pixel $(x, y)$ at time *t*, where $I(x, y, t)$ is the intensity at pixel $(x, y)$ at time $t$, $\upsilon_x$, $\upsilon_y$ is the horizontal and vertical velocities in pixel $(x, y)$. We can obtain $\upsilon = (\upsilon_x, \upsilon_y)$ by minimizing the objective function:

$$C = \int_D \left[ \lambda^2 \left\| \nabla \upsilon \right\|^2 + (\nabla I \cdot \upsilon + I_t)^2 \right] dxdy, \qquad (9)$$

where there are many methods to solve the optical flow equation. We use the iterative algorithm [20] to compute the optical flow velocity：

$$\begin{aligned} \upsilon_x^{k+1} &= \overline{\upsilon}_x^k - \frac{I_x\left[I_x\overline{\upsilon}_x^k + I_y\overline{\upsilon}_y^k + I_t\right]}{\lambda + I_x^2 + I_y^2} \\ \upsilon_y^{k+1} &= \overline{\upsilon}_y^k - \frac{I_y\left[I_x\overline{\upsilon}_x^k + I_y\overline{\upsilon}_y^k + I_t\right]}{\lambda + I_x^2 + I_y^2} \end{aligned}, \qquad (10)$$

where $k$ is the number of iterations, initial value of velocity $\upsilon_x^0 = \upsilon_y^0 = 0$, $\overline{\upsilon}_x^k, \overline{\upsilon}_y^k$ is the average velocity of the neighborhood of point $(x, y)$.

The optical flow vector field $\upsilon$ is then split into two scalar fields $\upsilon_x$ and $\upsilon_y$ corresponding to the $x$ and $y$ components of $\upsilon$. $\upsilon_x$ and $\upsilon_y$ are further half-wave rectified into four none-gative channels $\upsilon_x^+$, $\upsilon_x^-$, $\upsilon_y^+$, $\upsilon_y^-$ so that $\upsilon_x = \upsilon_x^+ - \upsilon_x^-$ and $\upsilon_y = \upsilon_y^+ - \upsilon_y^-$ These four nonnegative channels are then blurred with a Gaussian kernel and normalized to obtain the final four channels $\upsilon b_x^+$, $\upsilon b_x^-$, $\upsilon b_y^+$, $\upsilon b_y^-$.

Facial expression is represented by facial velocity features that are composed of the channels $\upsilon b_x^+$, $\upsilon b_x^-$, $\upsilon b_y^+$, $\upsilon b_y^-$ of all pixels in facial image. Facial expression can be regard as facial motion which are important characteristic features of facial expression, in addition to, the velocity features have been shown to perform reliably with noisy image sequences, and has been applied in various tasks, such as action classification, motion synthesis, etc.

c. Visual words for characterizing facial expression

The human facial expression is a dynamic event, which must be represented by the motion information of the human face. To improve the accuracy of facial expression detection, fusing the facial local texture feature vector and optical flow vector forms a hybrid feature vector by using the method of BoW (Bag of Words) [21], which can be better more effective for facial expression representation.

We divide each facial image into $L \times L$ blocks, and each image block is represented by hybrid feature vector of all pixels in the block. On this basis，Facial expressions are represented by visual words using the method of BoW. We randomly select a subset from all image blocks to construct the codebook. Then, we use k-means clustering algorithms to obtain clusters.

Codewords are then defined as the centers of the obtained clusters, namely visual words. In the end, each face image is converted to the "bag-of-words" representation by appearance times of each codeword in the image is used to represent the image.

$$X_{ij} = \{n(I, w_1), \cdots, n(I, w_j), \cdots, n(I, w_M)\}, \qquad (11)$$

where $n(I, w_j)$ is the number of visual word $w_j$ included in image, $M$ is the number of vision words in word sets.

## III. MULTI-INSTANCE BOOSTING FOR FACIAL EXPRESSION

After characterizing human facial expression, there are many methods to recognize human facial expression. Because human facial expression detection can regard as a Multiple Instance problem, we use the Multi-Instance boosting algorithm to learn and recognize human facial expression. The Multi-Instance boosting model has been applied to various computer vision applications, such as object detection, action detection, human, etc. The Multi-Instance boosting framework is used to learn a unified classifier instead of individual classifiers for all classes in order to increase detection efficiency without compromising accuracy. Our approach is directly inspired by a body of work on using generative Multi-Instance boosting models for visual detection based on the "bag-of-words" paradigm. We propose a novel Multi-Instance boosting framework, which learns a unified classifier instead of individual classifiers for all classes, so that the detection efficiency can be increased without compromising accuracy.

a. Definition of multi-instance problem

Keeler, et. al [22] proposed originally the idea for the multi-instance learning for handwritten digit detection in 1990. It was called Integrated Segmentation and Detection (ISR), and it is the key idea to provide a different way in constituting training samples. Dietterich et al. [23, 24] proposed Multiple-Instance framework for the prediction of drug molecule activity. Multiple-Instance Learning has widely used in image classification [25, 26], human face [27], etc. Definition of multi-instance is as follows:

Assume the set of class labels $C_i \in \{0,1\}$ , $\chi$ is the instance space, multi-instance data set $D = \{X_i, C_i\}_{i=1}^{N}$. The instances are defined as $\{x_k \,|\, k = 1, 2, \cdots, \tau\}$ in multi-instance data set $D$.

All the instances in the positive bags and negative bags is defined as $D^+$ and $D^-$ respectively, where $D^+ = \{x_i^+ \,|\, i = 1, 2, \cdots, \tau\}$, $D^- = \{x_j^- \,|\, j = 1, 2, \cdots, \varsigma\}$.

Given a bag $X_i$, $X_i$ is a positive bag if at least one of its instances is positive; otherwise, $X_i$ is a negative bag.

The multi-instance problem is a function. Its goal of multi-instance is to learn a classifier based on instance.

$f(x_i): x_i \to h$, $x_i \in \chi$ or a classifier based on bags.

$F(X_i): X_i \to h$, that correctly predicts the unlabeled bag.

Given two bags $X_i$ and $X_k$ $(i \neq k)$, $X_i \bigcap X_k \neq \phi$.

In the multi-instance problem, each instance only belongs to one specific bag. Namely, two different bags cannot share the same instance. The framework of multi-instance is shown in figure 2.

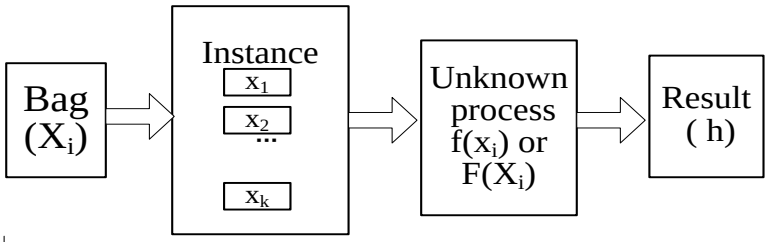


Figure 2. The framework of multi-instance

b. Boosting algorithm analysis

Boosting algorithm is a mathematical model based on a fuzzy rule system. A fuzzy rule system is defined as follows by using the classical case at the beginning. In the classical case, a rule is a function formulated with arguments coupled by logical operators, yielding a logical expression and a corresponding response. It is a semi-supervised learning method [28]. The steps of boosting algorithm can be briefly described as follows.

Assuming the training sample set $\{(x^1, c_1), (x^2, c_2), \cdots, (x^n, c_n)\}$, $c_n \in \{c_1, \cdots, c_l\}$

Give equal initial weights of each sample: $\omega^i = 1/n$, the training sample set trains for $\kappa$ rounds of training and obtains $\kappa$ fuzzy classification rules.

For $t = 1, 2, \cdots, \kappa$ Do

Under the current sample distribution, calculated the corresponding weights of fuzzy rules as follows:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-E(R_t)}{E(R_{t)}}\right), \tag{12}$$

Update the sample weight as follows:

$$\omega^i(t+1) = \frac{\omega^i(t)}{z_t} \times \begin{cases} e^{-\alpha_t \mu_{rt}(x^i)} & c_i = c_t \\ e^{\alpha_t \mu_{R_t}(x^i)} & c_i \neq c_j \end{cases}, \tag{13}$$

The category is obtained by the fuzzy classifier as follows:

$$C_{\max}(x^k) = \arg\max_{C_m} \sum_t \alpha_t \sum_{R_i / c_t = C_k} \mu_{R_t}(x^k), \tag{14}$$

where $x^k$ is unknown sample, $x^k = \left\{x_1^k, x_2^k, \cdots, x_N^k\right\}$.

c. Multi-Instance boosting for expression detection

To improve the detection efficiency, we combine Multiple-Instance and boosting to build Multi-Instance boosting model. Multi-Instance boosting is one of the most efficient machine learning algorithms. In Multi-Instance boosting, training samples are not singletons, at the same time they are in “bags”, where all of the samples in a bag share a label [29]; Samples are organized into positive bags of instances and negative bags of instances, where each bag may contain a number of instances [30]. At least one instance is positive (i.e. object) in a positive bag, while all instances are negative (i.e. non-object) in a negative bag. In Multi-Instance boosting [31], learning must simultaneously learn that samples in the positive bags are positive along with the parameters of the classifier. To obtain training samples, each image is divided into $L \times L$ blocks. We treat each block in a image as a single word $w_j$ and a image as a bag. Each block is used as an example for the purposes of training. It is suitable to represent the object by a bag of multiple instances (non-aligned human face images). Then, Multi-Instance boosting can learn that instances in the positive bags are positive, along with a binary classifier [32]. In this paper, Multi-Instance boosting is used for facial expression with non-aligned training samples. The Multi-Instance boosting for facial expression detection proceeds as follows:

Input: Given data set $\{X_i, C_i\}_{i=1}^{N}$, $X_i$ is training bags, where $X_i = \{x_{i1}, x_{i2}, \cdots, x_{ij}, \cdots, x_{iN}\}$, $C_i$ is the score of the sample, and $C_i \in \{0,1\}$. $n$ is the number of all weak classifiers. A positive bag contains at least one positive sample, and $C_i = \max_j(C_{ij})$.

Pick out $K$ weak classifiers and consist of strong classifier.

*Step 1*: Update all weak classifiers in the pool with data $\{x_{ij}, C_i\}$.

*Step* 2: Initialize all strong classifier: $H_{ij} = 0$ for all $i$ , $j$ .

*Step 3:* Calculate the probability that the *j*-th sample is positive in the *i*-th bag as follows:

For $k = 1, 2, \cdots, K$ do

For $m = 1, 2, \cdots, N$ do

$$P_{ij}^{m} = \sigma(H_{ij} + h_m(x_{ij})), \tag{15}$$

where $P_{ij}^{m} = p(C_i \mid x_{ij}) = \dfrac{1}{1+\exp(-c_{ij})}$.

We calculate the probability that the bag is positive as follow:

$$P_i^m = 1 - \prod_j (1 - p_{ij}^m), \tag{16}$$

where $P_i^m = \mathrm{p}(C_i \mid X_i)$.

The likelihood assigned to a set of training bags is:

$$C^m = \sum_i (C_i \log p_i^m + (1 - C_i)\log(1 - p_i^m)). \tag{17}$$

End for

Finding the maximum *m** from *N* as the current optimal weak classifier as follow:

$$m^* = \arg\min_m C^m, \tag{18}$$

The *m** come into the strong classifier:

$$h_k(x) \leftarrow h_{m^*}(x), \tag{19}$$

$$H_{ij} = H_{ij} + h_k(x), \tag{20}$$

End for

*Step 4:* Output: Strong classifier which consist of weak classifiers as follow:

$$H(x) = \sum_k h_k(x), \tag{21}$$

where $h_k$ is a weak classifier and can make binary predictions using $sign(H_K(x))$.

In Multi-Instance boosting, samples come into positive bags of instances and negative bags of instances. Each instance $x_{ij}$ is indexed with two indices, where $i$ for the bag and $j$ for the instance within the bag. All instances in a bag share a bag label $C_i$. Weight of each sample composes of the weight of the bag and the weight of the sample in the bag. The quantity of the samples can be interpreted as a likelihood ratio, where some (at least one) instance is positive in a bag. $P_{ij}^m$ is the probability that some instance is positive. So the weight of samples in the bags is $P_{ij}^m$. We calculate: $w_{ij} = \frac{\partial \log C^m}{\partial y_{ij}}$, and get weight of the bags $w_{ij}$.

Training in the initial stages is the key to a fast and effective classifier. The result of the Multi-Instance boosting learning process is not only a sample classifier but also weights of the samples. The samples have high score in positive bags which are assigned high weight. The final classifier labels these samples to be positive. The remaining samples have a low score in the positive bags, which are assigned a low weight. The final classifier classifies these samples as negative samples as they should be. We train a complete Multi-Instance boosting classifier to achieve the desired false positive rates and false negative rates. Retrain the initial weak classifier so that a zero false negative rate is obtained on the samples, which label positive by the full classifier. This results in a significant increase in many samples to be pruned by the classifier. Repeat the process so that the second classifier is trained to yield a zero false negative rate on the remaining samples.

For the task of facial expression detection, our goal is to classify a new face image to a specific facial expression class. During the inference stage, given a testing face image, we can treat each aspect in the Multi-Instance boosting model as one class of facial expression. For facial expression detection with large amount of training data, this will result in long training time. In this paper, we adopt a supervised Algorithm to train Multi-Instance boosting model. The supervised training algorithm not only makes the training more efficient, but also improves the overall detection accuracy significantly. Each image has a class labeled information in the training images, which is important for the classification task. Here, we make use of this class label information in the training images for the learning of the Multi-Instance boosting model, since each image directly corresponds to a certain facial expression class on train sets.

## IV. EXPERIMENTAL RESULTS AND ANALYSIS

We studied facial expression feature representation and facial expression classification schemes to recognize seven different facial expressions, such as “anger”, “disgust”, “fear”, “happiness”, “neutral”, “sadness” and “surprise” in the JAFFE database. We verified the effectiveness of our proposed algorithm using C++ and Matlab7.0 hybrid implementation on a PC with Intel CORE i5 3.2 GHz processor and 4G RAM.

JAFFE data set is the most available video sequence dataset of human facial expression. In this database, there are seven groups of images by 10 Japanese women and a total of 213 images, which are “anger”, “disgust”, “fear”, “happiness”, “neutral”, “sadness” and “surprise” respectively [33]. The size of each image is 256×256 pixels in the JAFFE database. Each face image was normalized to a size of 8×8. Some sample images are shown in figure 3.

(a) (b) (c) (d) (e) (f) (g)

Figure 3. Example of seven facial expression images on JAFFE (a) “anger”, (b) “disgust”, (c) “fear”,(d) “happiness”, (e) “neutral”, (f) “sadness”, (g) “surprise”

In Experiments, we chose 30 face images per class randomly for training, 20 face images for testing in JAFFE. These images were pre-processed by aligning and scaling, thus the distances between the eyes were the same for all images, and ensured that the eyes occurred in the same coordinates of the image. The system was run seven times, and we obtained seven different training and testing sample sets. The detection rates were obtained by average detection rate of each run.

We divided each face image into $L*L$ blocks. In order to determine the size of image block, we studied the effect of the size of image block $L*L$ on the detection accuracy. The detection

accuracy curve with different block sizes was shown in Figure 4. We can conclude that the accuracy peaked when the block sizes L is 8. Thus, *L* is set as 8.

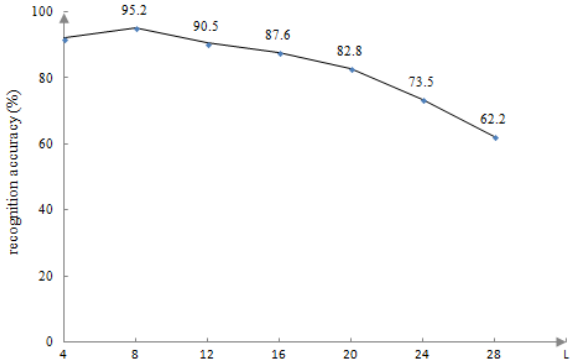


Figure 4. Detection accuracy curve with different block sizes

We researched the value of *M* that is the number of the visual word set so that we could determine the value of *M*. The relation between *M* and detection accuracy was observed in Figure 5. We can see in Figure 5 that the facial expressions detection accuracy is rise up with the increasing of *M* at the beginning, and the detection accuracy is stabled to 95.3% if *M* is larger than or equal to 60. Therefore *M* is set as 60.

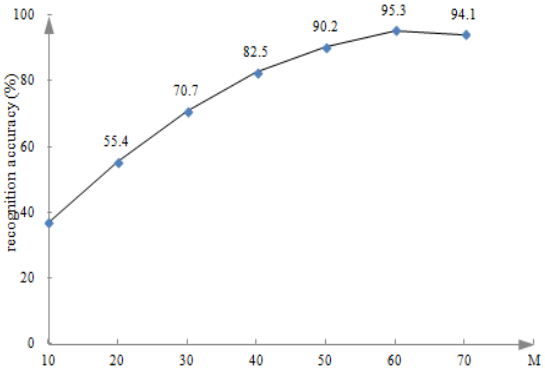


Figure 5. Relation curve between M and accuracy

In order to examine the accuracy of our proposed facial expressions detection approach, we used 500 different face images for this experiment. Some images contained the same person but in different expressions. The detection results are presented in the confusion matrices. The confusion matrix for per-video classification is shown in Figure 6, which used 60 codewords. Where A,B,C,D,E,F,G indicate “anger”, “disgust”, “fear”, “happiness”, “neutral”, “sadness” and “surprise” respectively.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| A | 0.96 | 0 | 0 | 0 | 0 | 0.04 | 0 |
| B | 0.01 | 0.98 | 0.01 | 0 | 0 | 0 | 0 |
| C | 0.04 | 0.02 | 0.94 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 0.97 | 0.02 | 0.01 | 0 |
| E | 0 | 0 | 0 | 0.09 | 0.90 | 0.01 | 0 |
| F | 0 | 0 | 0.03 | 0.01 | 0.02 | 0.93 | 0 |
| G | 0 | 0 | 0.01 | 0 | 0 | 0 | 0.99 |

Figure 6. Confusion matrix for facial expression detection

Each cell in the confusion matrix is the average result of facial expression respectively. We can see that the algorithm correctly classifies most facial expressions. Average detection rate gets to 95.3%. Most of the mistakes are confusions between “anger” and “sadness”,
between “happiness” and “neutral”, between “fear” and “surprise”. It is intuitively reasonable that they are similar facial expressions.

To examine the accuracy of our proposed facial expression detection approach, we compared our method with three state-of-the-art approaches for facial expression detection using the same data. The first method is “AAM+SVM”, which used Active Appearance Models (AAM) to extract face features, and SVM to classify facial expression. The second method is “ASM+GW”, which used Active Shape Model and Gabor Wavelet (ASM+GW) for facial expression detection. The third method is “SLPP+ MKSVM”, which used SLPP to extract facial expression feature, and multiple kernels support vector machines (MKSVM) was used to recognize. 300 different expression images were used for this experiment, where some images contained the same person but in different facial expression. The results of detection accuracy comparison are shown in Table 1.

Table 1. Detection accuracy comparison of different method

| method | Error rate | accuracy |
|---|---|---|
| AAM+SVM | 0.177 | 0.823 |
| ASM+GW | 0.108 | 0.892 |
| SLPP+ MKSVM | 0.114 | 0.886 |
| Our method | 0.047 | 0.953 |

In Table 1, we can see that AAM+SVM obtain average detection accuracy of 82.3%. The average detection rate of ASM+GW is to 89.2%. The average detection accuracy of SLPP+ KSVM is to 88.6%. Our method is stabled to average detection accuracy of 95.3%. Our method has higher detection accuracy, lower error rate and performs significantly better than the above three state-of-the-art approaches for facial expression detection. Because we improved the detection

accuracy in the two stages of facial expression features extraction and facial expression detection. In the stage of facial expression feature extraction, we used local texture feature and motion features that were reliably with noisy image sequences and bag-of-words framework to describe facial expression effectively. In the stage of expression detection, we used Multi-Instance boosting algorithm to classify facial expression images. Our method performs the best, its detection accuracies and speeds are satisfactory.

In order to verify the effectiveness of the proposed algorithm, we have built a database of face images about seven facial expressions and performed experiments on it. In this database, there are seven groups of images (“surprise”, “fear”, “disgust”, “anger”, “happiness”, and “sadness” respectively), and each group includes 25 males and 20 females. The face images were taken under various laboratories controlled lighting conditions, and each face image was normalized to a size of 64×64. Some sample images are shown in figure 7.

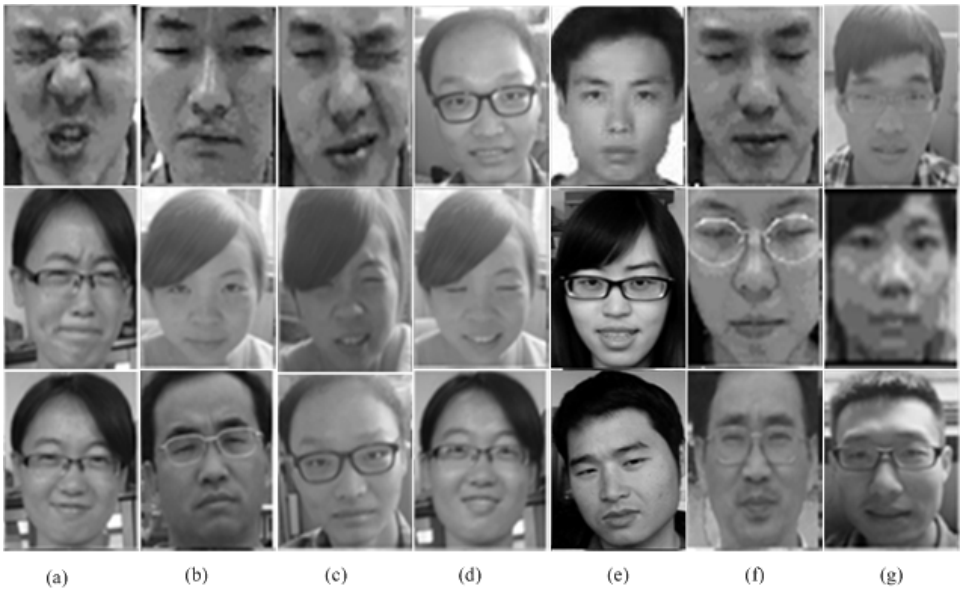


Figure 7 Example of seven facial expression images from facial videos

(a) “anger”, (b) “disgust”, (c) “fear”,(d) “happiness”, (e) “neutral”, (f) “sadness”, (g) “surprise”

In Experiments, 20 face images per class are randomly chosen for training while the remaining images are used for testing. These images were pre-processed by aligning and scaling them so that the distances between the eyes were the same for all images and also ensuring that the eyes occurred in the same coordinates of the image. We run the system 6 times and obtain 6 different training and testing sample sets. The detection rates were found by averaging the detection rate of

each run. We compare our method to three state-of-the-art approaches ("AAM+SVM", "ASM+GW", "SLPP+ MKSVM") for facial expression detection using the same data. The results of detection accuracy comparison are shown in figure 8.

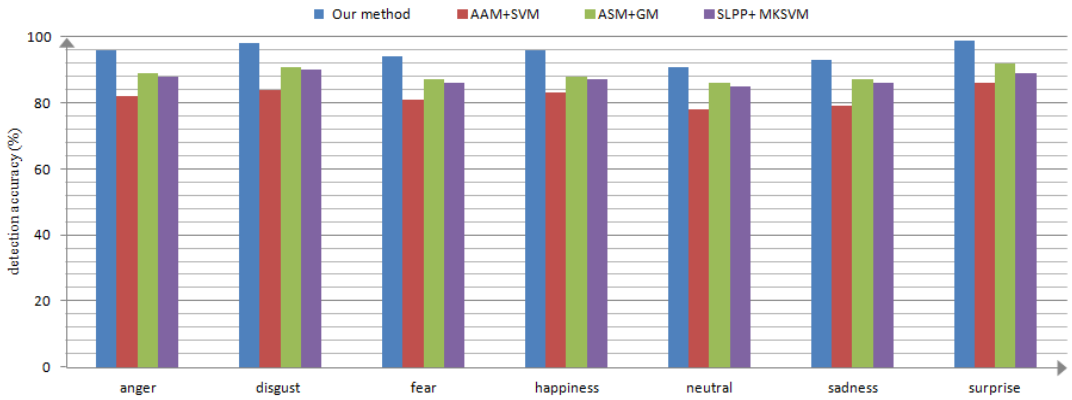


Figure 8 Detection accuracy comparison of different method

As figure 8 shows, our method improves the detection accuracies and achieves 95.2% average detection rate, whereas "AAM+ SVM" obtains 81.9%, "ASM+GW" gets 88.5%, and "SLPP+ MKSVM" attains 87.3%. We can see that our method performs significantly better than the above three state-of-the-art approaches for facial expression detection.

## V. CONCLUSIONS

Facial expression detection can provide significant advantage in public security, financial security, drug-activity prediction, image retrieval, face, etc. In this paper, we have presented a novel method to recognize the facial expression and given the seven facial expression levels at the same time. The main contribution can be concluded as follows:

(1) ASM model was used for local texture features extraction. Optical flow model was used to extract facial velocity features, then after fusing local texture features and facial velocity features, we got hybrid features and converted them into visual words using "bag-of-words" models. Visual words were used for facial expression representation

(2) Multi-Instance boosting model was used for facial expression detection. In our models, Multi-Instance and boosting were used to create Multi-Instance boosting. We proposed a new Multi-Instance boosting framework, which recognized different facial expression categories. In addition,

in order to improve the detection accuracy, the class label information was used for the learning of the Multi-Instance boosting model.

(3) Experiments evaluated our proposed method, which were performed on a facial expression dataset built by ourselves and on JAFFE. Experimental results reveal that our proposed method significantly improves the detection accuracy and performs better than previous ones.

## ACKNOWLEDGMENTS

This work is supported by Research Foundation for Science & Technology Office of Hunan Province under Grant (No.2014FJ3057), by Hunan Provincial Education Science and "Twelve Five" planning issues (No. XJK012CGD022), by Teaching Reform Foundation of Hunan Province Ordinary College under Grant (No. 2012401544), by the Postdoctoral Science Foundation of Central South University, and by the Construct Program of the Key Discipline in Hunan Province.

## REFERENCES

[1] I. Cohen, N. Sebe, A. Garg, et al, "Facial expression detection from video sequences: temporal and static modeling", Computer Vision and Image Understanding, vol.1, No.91，2003, pp.160-187.

[2] S. Morishima and H. Harashima, "Emotion space for analysis and synthesis of facial expression", Proc. 2nd IEEE Int. Workshop on Robot and Human Communication, 1993, pp. 188-193.

[3] C. Shan, S. Gong and P. W. McOwan, "Robust facial expression detection using local binary patterns", Proc. Int. Conf. on Image Processing, ICIP 2005, vol.2, No.2, 2005, pp.370-373.

[4] P. S. Aleksic and A. K. Katsaggelos, "Automatic facial expression detection using facial animation parameters and multistream HMMs", IEEE Transactions on Information Forensics and Security, vol.1, No.1, 2006, pp. 3-11.

[5] N. Neggaz, M. Besnassi and A. Benyettou, "Facial expression detection", Journal of Applied Sciences, vol.15, No.10, 2010, pp. 1572-1579.

[6] Y. Q. Wang and , L. Liu, “New intelligent classification method based on improved meb algorithm”, International Journal on Smart Sensing and Intelligent Systems, vol. 07, No. 1, 2014, pp. 72-95.

[7] G. Zhao and M. Pietikäinen, “Boosted multi-resolution spatiotemporal descriptors for facial expression detection”, Pattern detection letters, vol. 12, No. 30, 2009, pp. 1117-1127.

[8] F. Y. Shih, C. F. Chuang and P. S. P. Wang, “Performance comparisons of facial expression detection in JAFFE database”, Int. J. Pattern Detection and Artificial Intelligence, vol.03, No.22, 2008, pp.445-459.

[9] S. Y. Fu, G. S. Yang and X. K. Kuai, “A spiking neural network based cortex-like mechanism and application to facial expression detection”, Computational Intelligence and Neuroscience, 2012, pp.1-13. Online publication date: 1-Jan-2012.

[10] C. Shan, S. Gong and P. W. McOwan, “Facial expression detection based on local binary patterns: A comprehensive study”, Image and Vision Computing, vol.06, No.27, 2009, pp. 803-816.

[11]D. C. Turk, C. Dennis and R. Melzack,” The measurement of pain and the assessment of people experiencing pain”, Handbook of Pain Assessment, ed D. C. Turk and R. Melzack, New York: Guilford, 2nd edition, 2001, pp. 1-11.

[12] L. Wang, R. F. Li, and K. Wang, ”A novel automatic facial expression detection method based on AAM”, Journal of Computers, vol.03, No.9, 2014, pp.608-617.

[13]K. M. Prkachin, ”The consistency of facial expressions of pain: a comparison across modalities”, Pain, vol. 05, No.3, 1992, pp.297-306.

[14]K. M. Prkachin and P. E. Solomon, “The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain”, Pain, vol.139, No.2, 2008, pp.267-274.

[15] S. J. Zhang, B. Jiang and T. Wang, “Facial expression detection algorithm based on active shape model and gabor wavelet”, Journal of Henan University (Natural Science), vol.05, No.40, 2010, pp.521-524.

[16] W. Zhang and L. M. Xia, “Pain expression detection based on SLPP and MKSVM”, Int. J. Engineering and Manufacturing, No.3, 2011, pp. 69-74.

[17] K. W. Wan, K. M. Lam and K. C. Ng, “An accurate active shape model for facial feature extraction”, Pattern Detection Letters, vol.15, No.26, 2005, pp. 2409-2423.

[18] J. M. Lobo and M. F. Tognelli, “Exploring the effects of quantity and location of pseudo-absences and sampling biases on the performance of distribution models with limited point occurrence data”, Journal for Nature Conservation, vol. 19, No.1, 2011, pp.1-7.

[19]S. M. Bhandarkar and X. Luo, “Integrated and tracking of multiple faces using particle filtering and optical flow-based elastic matching”, Computer Vision and Image Understanding, vol. 06, No.113, 2009, pp. 708-725.

[20] B. K. Horn and B. G. Schunck, “Determining optical flow”, Artificial Intelligence，No.17，1981, pp.185- 204.

[21] G. J. Burghouts and K. Schutte, “Spatio-temporal layout of human actions for improved bag-of-words action ”, Pattern Detection Letters, vol.15, No.34, 2013, pp.1861-1869.

[22] J. D. Keeler, D. E. Rumelhart and W. K. Leow, “Integrated segmentation and detection of hand-printed numerals”, 1990 NIPS-3: Proc. Conf. on Advances in neural information processing systems 3, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1990, pp. 557–563.

[23] T. G. Dietterich, R. H. Lathrop and T. Lozano-Perez, “Solving the multiple instance problem with axis-parallel rectangles”, Artificial Intelligence, No.89, 1997, pp. 31-71.

[24]A. Zafra, M. Pechenizkiy, and S. Ventura, “Relief-MI: an extension of relief to multiple instance learning”, Neurocomputing, No.75, 2012, pp.210-218.

[25] Y. X. Chen, J. B. Bi and J. Z. Wang, “MILES: multiple-instance learning via embedded instance selection”, IEEE Transaction Pattern Analysis and Machine Intelligence, No.28, 2006, pp. 1931-47.

[26] X. F. Song, L. C. Jiao, S. Y. Yang, X. R. Zhang, and F. H. Shang, “Sparse coding and classifier ensemble based multi-instance learning for image categorization”, Signal Processing, No.93, 2013, pp.1-11.

[27] P. Viola, J. Platt, and C. Zhang, “Multiple instance boosting for object ”, Advance in Neutral Information Processing System, No.18, 2006, pp.1419-1426.

[28] M. Nakamura, H. Nomiya and K. Uehara, “Improvement of boosting algorithm by modifying the weighting rule”, Annals of Mathematics and Artificial Intelligence, vol.1, No.41, 2004, pp. 95-109.

[29] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles”, Artificial Intelligence, vol.89, No.1, 1997, pp.31–71.

[30]S. Andrews and T. Hofmann, "Multiple instance learning via disjunctive programming boosting", Advances in Neural Information Processing Systems, No.16, 2004, pp. 65-72.

[31] T. Quazi, S.C. Mukhopadhyay, N. Suryadevara and Y. M. Huang, Towards the Smart Sensors Based Human Emotion Recognition, Proceedings of IEEE I2MTC 2012 conference, IEEE Catalog number CFP12MT-CDR, ISBN 978-1-4577-1771-0, May 13-16, 2012, Graz, Austria, pp. 2365-2370.

[32] Y. T. Chen, C. S. Chen, Y. P. Hung ,et al, "Multi-class multi-instance boosting for part-based human ", IEEE 12th Int. Conf. on. Computer Vision Workshops (ICCV Workshops), 2009, pp.1177-1184.

[33] G.Sengupta, T.A.Win, C.Messom, S.Demidenko and S.C.Mukhopadhyay, "Defect analysis of grit-blasted or spray printed surface using vision sensing technique", Proceedings of Image and Vision Computing NZ, Nov. 26-28, 2003, Palmerston North, pp. 18-23.

[34] O. Yakhnenko and V. Honavar, "Multi-Instance multi-label learning for image classification with large vocabularies", BMVC, 2011, pp.1-12.

[35] G. Sen Gupta, S.C. Mukhopadhyay and M Finnie, Wi-Fi Based Control of a Robotic Arm with Remote Vision, Proceedings of 2009 IEEE I2MTC Conference, Singapore, May 5-7, 2009, pp. 557-562.

[36] F. Cheng, J. Yu, H. Xiong, "Facial expression detection in JAFFE dataset based on Gaussian process classification", IEEE Transactions on Neural Networks, vol.10, No.21, 2010, pp.1685-1690.