



Video mining for facial action unit classification using statistical spatial–temporal feature image and LoG deep convolutional neural network

Masoud Z. Lifkooee¹ · Ömer M. Soysal² · Kazim Sekeroglu²

Received: 26 July 2017 / Revised: 6 May 2018 / Accepted: 24 July 2018 / Published online: 13 August 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

In this paper, we aim to improve the convolutional deep learning by means of proposed statistical feature image descriptor and Laplacian of Gaussian filtering. We propose a statistical feature image descriptor (SFID) that is composed of spatial and temporal parts for video content classification using convolutional deep neural network. We apply the proposed descriptor to multi-view action unit classification. The SFID is a statistical representation of the raw image based upon K -abstraction levels. It is capable of addressing the fixing input array size of the deep learning model. Further, it eliminates redundancy in representation of the content; hence, it reduces computation cost. The proposed SFID can be in spatial and/or temporal form. The temporal form is particularly important in video content classification. We added a new layer of Laplacian of Gaussian filter (LoG) right before fully connected layer into the regular deep convolutional neural network (DCNN) structure. The parameters of the LoG are adaptively calculated using the Gaussian mixture models. The classification results are compared with regular DCNN, SVM models, and KNN together with feature descriptors of SIFT and SURF. The results show that the proposed feature descriptor and introducing a LoG filter layer give promising performance for deep learning.

Keywords Action unit classification · Facial expression · Image descriptor · Deep learning · Video mining

1 Introduction

Face is the primary nonverbal human's tool to transfer emotions and communicate with others. Studying the human's facial expressions has been an important topic in different cultures and countries [1]. Facial expression recognitions are an important aspect of human–machine interaction [2]. Similar facial expressions such as happiness follow the same structured patterns of the relaxation and contraction of muscle in the specific regions of face. Based on the regional structure of facial expressions, some local units can be defined. The action unit (AU) structure is the most common system used for describing facial expressions [3]. The action

units can be defined as the variation of the muscles in different locations of face such as mouth and eyebrow that follows a general set of patterns such as relaxation and contraction in different facial expression [4]. The variation of the muscles can be defined as the temporal information which is an important part of the action units as the action units are built based on the dynamic of the face [5]. A total of 46 action units have been defined which are able to describe more than 7000 facial expressions [6]. Figure 1 has some sample images of the action units that are used in this paper. AU 1, AU 4, AU 6, and AU 7 are associated with upper part of the face, such as eyes and eyebrow. AU 10, AU 12, AU 14, AU 15, AU 17, and AU 23 are associated with the lower part of face, such as mouth.

Factually, the main advantage of using action units is that a large number of facial expressions can be described by a limited number of action units. There have been some applications for facial and action recognition including biology, neuroscience and psychology [1] disease detection, analysis of human emotions, security, learning systems, and surveillance [7].

Masoud Z. Lifkooee and Ömer M. Soysal have contributed equally to this article.

✉ Ömer M. Soysal
omermssoysal@gmail.com

¹ Louisiana State University, Baton Rouge, LA 70803, USA

² Southeastern Louisiana University, Hammond, LA 70402, USA

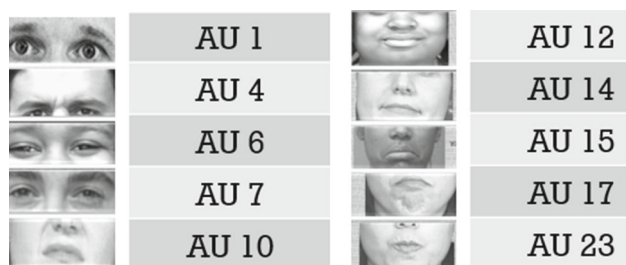


Fig. 1 Ten action units existing in FERA 2017 dataset [4]

Facial action unit recognition can be a challenging task as many variations such as head-pose, illumination [1], and scale may affect the accuracy of the results. Moreover, the spatial relevance of the action units to the class label may not be so clear for many classifiers since each action unit belongs to a specific region of face. Furthermore, big size of the input image can be a problem for many deep learning architectures as the computational cost of the deep learning architecture is highly dependent on the size of the inputs.

In this paper, we propose a spatial–temporal feature image and LoG filter operation to enhance the deep convolutional neural network (DCNN)-based learning to classify action units in multi-view facial expression videos. DCNN is a state-of-the-art method that is suitable for complex image processing problems [8, 9]. We employed spatial–temporal feature image instead of raw images that is common in DCNN-based classification. Furthermore, the trained feature maps are processed by LoG filter layer whose parameters are adaptively computed by mixtures of Gaussian models.

We compared the performance of SIFT and SURF descriptors from which the proposed statistical feature image is formed. Our experimental results show that the SURF-based SFID is better than that of SIFT. Among the feature extractors, SURF descriptors [10] provide a good representation under rotation and scale variation [11]. However, there are three issues when SURF descriptors are used. Firstly, SURF descriptors can handle a slight amount of variation in rotation while in the multi-view case, face images may have a significant amount of rotation. Secondly, the number of key points extracted by SURF descriptors is not fixed. However, in many types of classifiers such as neural network-based classifiers, the number of inputs must be fixed. Thirdly, the number of key points can be high that makes the computational cost expensive. Figure 2 shows an example of SURF key points detected in faces at different views and scales.

In summary, a new content-based image representation SFID is proposed in this paper. In addition, we showed that the proposed adaptive LoG-based DCNN structure improves classification performance of facial action units.

1.1 Previous work

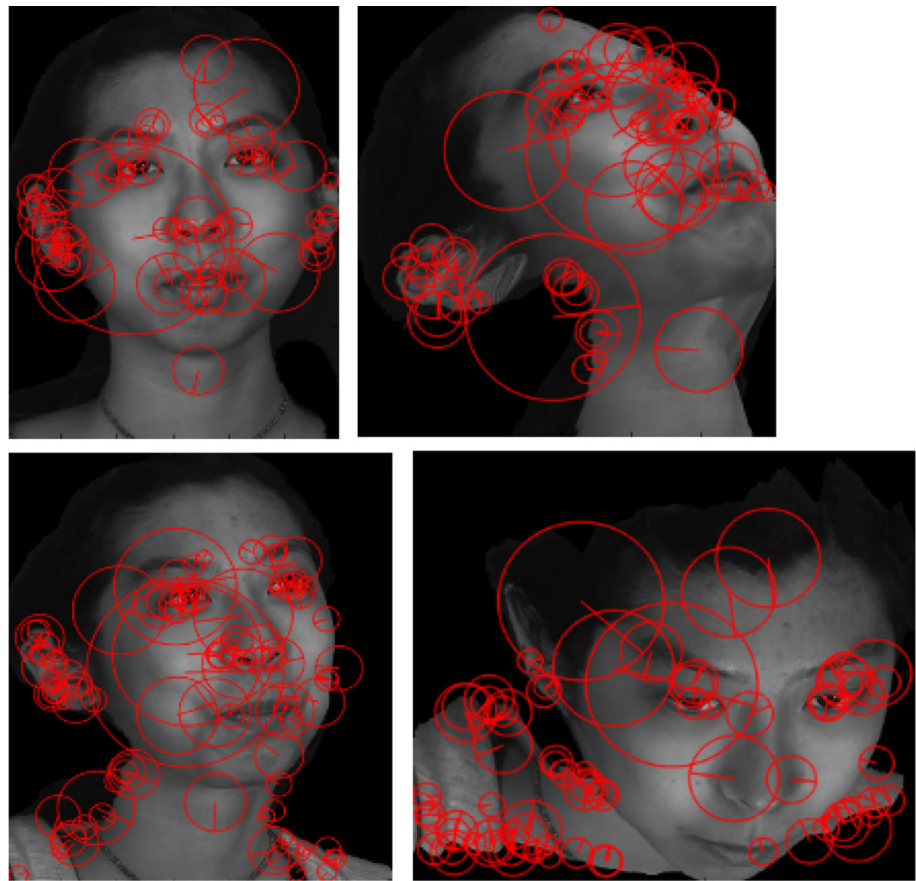
The study of facial expression recognition can be categorized into two groups: The first group utilizes only spatial appearance information and the second group exploits temporal dynamic features in addition to the spatial one.

The most of the researches utilized non-deep learning methods. Using spatial images with a non-deep architecture is a common method for facial expression and action unit recognition. In [2], the researchers proposed a real-time algorithm for facial expression recognition that is invariant to pose and scale change. They utilized patch-based histogram of oriented gradients (HOG) features. Several key points in the face are detected to determine the most informative patches. A support vector machine (SVM) is used to classify the extracted features.

Non-deep methods also have been used with temporal images. In [12], the cumulative different gabor features were used to extract temporal features. Independent component analysis (ICA) was conducted to solve the class separation problem. Multiple classifiers are created to recognize different action units. Each testing sample is compared to the manifold vectors in the dictionary for classification. However, creating separated classifiers for a binary problem may cause overfitting, especially for some action units that the number of positive samples is low and the training process cannot fit an appropriate model to the data.

Deep learning method with spatial images has been considered by several researchers for facial expression and action unit recognition. Using Restricted Boltzmann Machine (RBM) to compute the relevancy of the actions units and facial expression in a low-level representation is suggested in [13]. The gradient descent algorithm using log likelihood cost function is exploited for the training part and some probability measurements are considered to link the facial expressions and action units in a high-level representation. Nevertheless, it was not defined that how temporal variation can be considered while the action units and facial expression relationship is captured by the deep neural network. After a preprocessing stage including eye detection and image cropping in [14], convolutional neural network (CNN) was used for facial expression recognition. Six types of expressions were recognized and among them happy, surprise and disgust received the highest classification rate. Still, the proposed facial expression recognition system is highly dependent on the preprocessing stage such as eye detection that may lead to some errors if there is any fault in the preprocessing stage. Moreover, no temporal information is added to the proposed system and also the input image size is too low. [15], designed a deep and multi-label learning (DRML) for action unit classification of raw images. They used region layers to capture the local facial features (action units) as well as regular convolutional layers. Still, the image size is 170×170

Fig. 2 The SURF key points detected in face images with different view and scale (face image is courtesy of [29])



that may not be sufficient. Furthermore, the weights assigned to classify different action units are equal. However, different action units are localized differently and may need different assigned weights. Using a hierarchical deep model with raw facial images was proposed in [16]. Firstly, image preprocessing is done for the rigid transformation normalization. After dividing imaged to patches, DCNN is used for feature extraction and two classifiers, Softmax and SVM are used to classify the facial action units. Nevertheless, the proposed structure of the network is too simple for handling many variations such as pose change. In [17], after importing facial images and segmenting the image to patches using sliding window, face component detector is applied using HOG features and RBM. Using the Gabor feature of eyes and mouth, facial expressions are recognized using Autoencoder. However, the preprocessing steps are too complex that may result to the different outputs in different runs of the algorithm.

Using deep learning approaches and spatial and temporal information is a powerful method that has been suggested by a few researchers. In [18], the researchers exploited a deep approach that uses two parallel dynamic and appearance modules. Raw images are used as the input of the facial appearance module and the images obtained by the optical flow approach are considered as the input of the dynamic

module. Training the both of the modules is done by using two DCNNs. The final classification results are obtained based on the final score calculated from fully connecting layers with joint outputs. However, both of temporal and spatial networks use the same frames which may not be efficient as they may capture the same information. Similar to [18], in [19] they used two networks for classifying spatial and temporal features but as opposed [18], two spatial and temporal processes are sequential. After importing the image sequences, they manually segment the action units based on a reference image. They used both raw units and binary units in the first layer of the DCNN and then they join two raw units and the binary units in the second layer. The output of the DCNN is exported to a temporal network, short-term memory neural networks, for the final decision. Nevertheless, the obtained results are not so promising. Moreover, the preprocessing algorithms to obtain image patches seem to be manual.

In [20], model-based and feature-based approaches are described. For feature-based methods for instance, shape evaluation of face images is done in [21] and curvelet variation investigation is done in [22]. As examples of model-based methods, 3D models of facial expression are created in

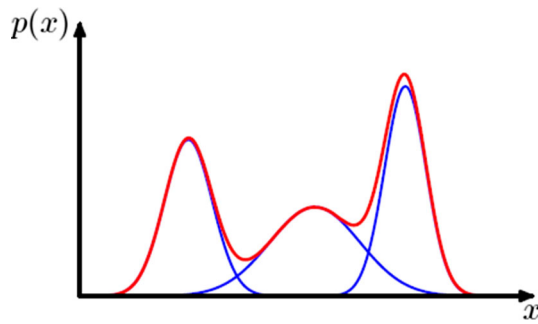


Fig. 3 An example of one-dimensional Gaussian mixture models with three Gaussian densities [25]

[23] and augmented Lagrangian multiplier approach (ALM) was used in [24].

2 Method

In this section, we explain the proposed statistical feature image descriptor and utilization of Laplacian of Gaussian filter whose parameters are adaptively calculated by means of Gaussian mixture models.

2.1 Gaussian mixture models

Gaussian mixture models (GMM) were introduced to handle some limitations of single Gaussian model [25]. Figure 3 shows a GMM with three Gaussian densities. For K Gaussian densities, the mixture of Gaussian models can be calculated by the linear combination of the densities as follows:

$$p(x) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k) \quad (1)$$

where the weight π_k is mixing coefficient to form linear combinations of Gaussian densities $N(x|\mu_k, \Sigma_k)$ with mean μ_k and covariance Σ_k [25].

A dataset of $\{x_1, x_2, \dots, x_N\}$ can be modeled with GMM by maximizing the likelihood function based on parameters including mean, covariance, and mixing coefficients. The step of the modeling can be described as follows [25].

1. Initialize mean μ_k , covariance Σ_k , and mixing coefficients π_k
2. Calculate the posterior probability for the random variable z_{nk} , $P(z_{nk})$ using the following equation:

$$P(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n|\mu_j, \Sigma_j)} \quad (2)$$

where K is the number of Gaussian densities.

3. Re-estimate the parameters as follows:

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N P(z_{nk}) x_n \quad (3)$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N P(z_{nk}) (x_n - \mu_k^{\text{new}})(x_n - \mu_k^{\text{new}})^T \quad (4)$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad (5)$$

$$\text{where } N_k = \sum_{n=1}^N P(z_{nk}) \quad (6)$$

4. Calculate the log likelihood:

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k) \right\} \quad (7)$$

5. Repeat from step 2 until convergence

2.2 Laplacian of Gaussian

Laplacian of Gaussian as can be formulated as the following equation is a filter that can be used to find rapid change in pixels values in image data. The σ can actually control the amount of filtering [26]:

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^4} \left(2 - \left(\frac{x^2 + y^2}{\sigma^2} \right) \right) e^{-(x^2 + y^2)/2\sigma^2} \quad (8)$$

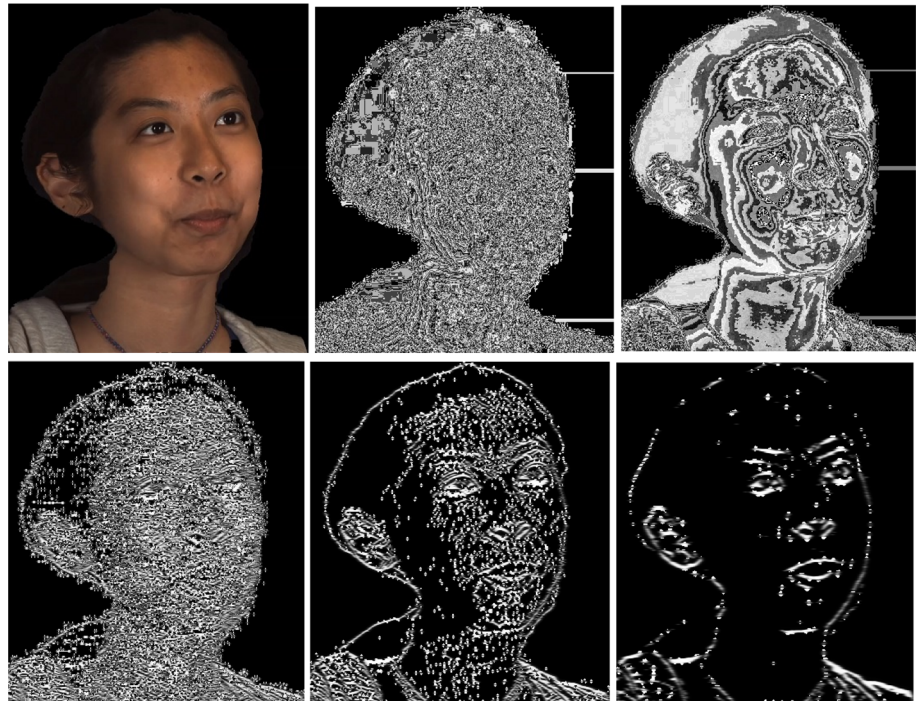
Laplacian of Gaussian can be used as appropriate features of images by finding the significant variation in the pixels instead of using all the pixel values that can contain meaningless data [27]. Figure 4 shows the outputs of the LoG filters with different variances.

2.3 Statistical feature image descriptor

The DCNN models require predefined input size. This requirement imposes difficulty when raw images are used as an input to a DCNN model. On the other hand, the raw images inherently explode representation space making the content recognition more challenging. As solution, we propose transforming features obtained from the raw content into the form of an image.

The main steps of obtaining the SFID: (1) Compute feature descriptors, such as SURF descriptors, from the raw image, (2) obtain *sampling signature* by applying a discretization to each dimension of the feature descriptors. The discrete levels are obtained from K -percentiles of feature matrix. Let Pr_K denote the sequence of K -percentiles. Let $Per_k(S)$ denote $Pr_K(k)$ th percentile of the set S ; e.g., $Pr_4 = (20, 40, 60, 80)$, then $Per_2(S)$ will be 40-percentile of the set S . In the formal

Fig. 4 Five examples of outputs of LoG filters with different variances (face image is courtesy of [29])



form, given a key descriptor matrix $D = [F_1 F_2 \dots F_N]$ with N feature vectors $F_j = (f_{j,1}, f_{j,2}, \dots, f_{j,L})^T$ of size L , where $j = 1, 2, \dots, N$, then

$$\text{SFID}_K \triangleq \{k : \text{Per}_k(S_i); k = 1, 2, \dots, K - 1; \\ S_i = (f_{1,i}, f_{2,i}, \dots, f_{N,i}), i = 1, 2, \dots, L\} \quad (9)$$

Intuitively, the SFID-K is a statistical representation of the raw image based upon K -abstraction levels. Note that the SFID-K is a 2D-array of size $K \times L$ and $D = [S_1; S_2; \dots; S_L]$ where it is the stack of row vectors. Figure 5 shows the flow of the system for SFID and LoG layers. As an example, a SURF descriptor of size 64 produces an image with the size of $K \times 64$. Figure 6 shows an example of SFID for AU1 and AU4. More examples and an illustrative example of obtaining SFID are provided in Appendix.

2.4 LoG-DCNN

Figure 7 shows the block diagram of the LoG-DCNN. At each epoch, sub-images are obtained through convolution and pooling for all statistical images. Each sub-image, then, is transformed to a 1D feature vector to be used by a Gaussian mixture models. The parameter of the LoG filters is calculated utilizing this GMM. Subsequently, the LoG-filtered images are utilized to train a regular feedforward ANN. At the final stage, the classification error is back-propagated to update each filter.

2.5 Adaptive computation of LoG parameter by means of GMMs

We propose to use the mean values of the trace of each covariance matrix (MoTC) as the variances of the LoG filters. Note that the diagonal elements of the covariance matrix of a GMM give the variation at the corresponding dimension. The rationale behind use of GMMs is to learn the variation from the data that is obtained from the multi-view multi-category of the same class objects.

During the training stage at each epoch, the GMMs are computed from the feature vector of size $m \times p \times k$ that are obtained by concatenating m sub-images with size of $p \times k$. Hence each sample image is represented by GMMs. During the testing stage, the variance of the LoG filters is predetermined experimentally.

We iteratively calculated the LoG filter variance that gives the minimum testing error. Initially, a minimum–maximum range is defined and the classification performance is measured for five different variances within this interval. The variance that gives the minimum error is used to update the range. As an example, the initial range is set to $[0.001, 10]$; the minimum errors are obtained with variance = 0.01. Then, the range is set to $[0.005, 0.05]$ around 0.01. This procedure continues until the error does not change significantly.

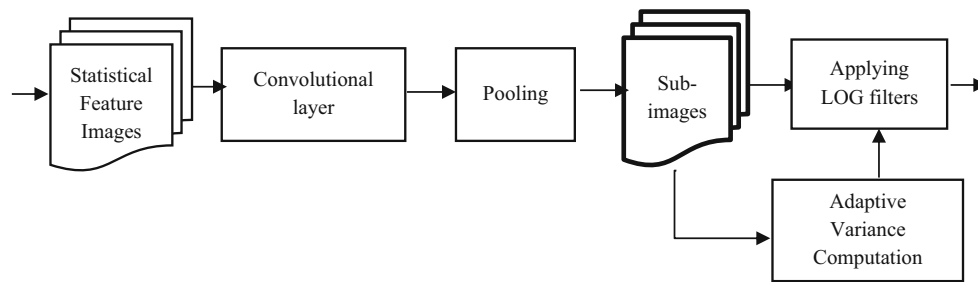


Fig. 5 A DCNN model with proposed statistical feature image descriptor and LoG layer

Fig. 6 Examples of SFIDs for AU1 and AU4 (SFID images are transposed)

	AU1	AU4
Raw Image		
SIFT-SFID		
SURF-SFID		

2.6 Statistical spatial–temporal feature images

Action units can be defined as four types of actions in face muscle: No activity, increasing the contraction of the muscle, being stable and decreasing the contraction of the muscle [1]. To capture these actions in different regions of the face, a patched temporal feature image is exploited. An example of patch-based SURF features can be seen in Fig. 8. While the spatial feature image is the original SFID, temporal image can be derived from the spatial image in different frames and different patches. In this study, the temporal image is obtained by the accumulative difference of the pixels of the patches of SFID in 10 frames. Let's consider a temporal feature image TI with the size of $K \times L$.

The pixel of (p, l) of the temporal feature image can be calculated as:

$$TI_{p,l}^c = \sum_{f=1}^{10} SFID_{p,l}^{f,c} - SFID_{p,l}^{(f-1),c} \quad (10)$$

where $p \in \{2, 3, \dots, P-1\}$, $c \in \{1, 2, \dots, C\}$, $l \in \{1, 2, \dots, L\}$ and P is the number of percentiles used in the temporal image, C is the number of patches, L is the size of SURF feature vector and f is the frame number in a sequence of facial expression video.

We explored two architectures that utilize temporal/spatial feature images and DCNN. In the first architecture, a single DCNN is designated separately for each spatial and feature images; the final decision is obtained by averaging. In the second architecture (named DCNN-2), both spatial and temporal feature images are merged to form a single feature image as seen in Fig. 9.

We also utilized Hu moments [28] as a temporal descriptor. The image is divided into 64 patches and Hu-feature

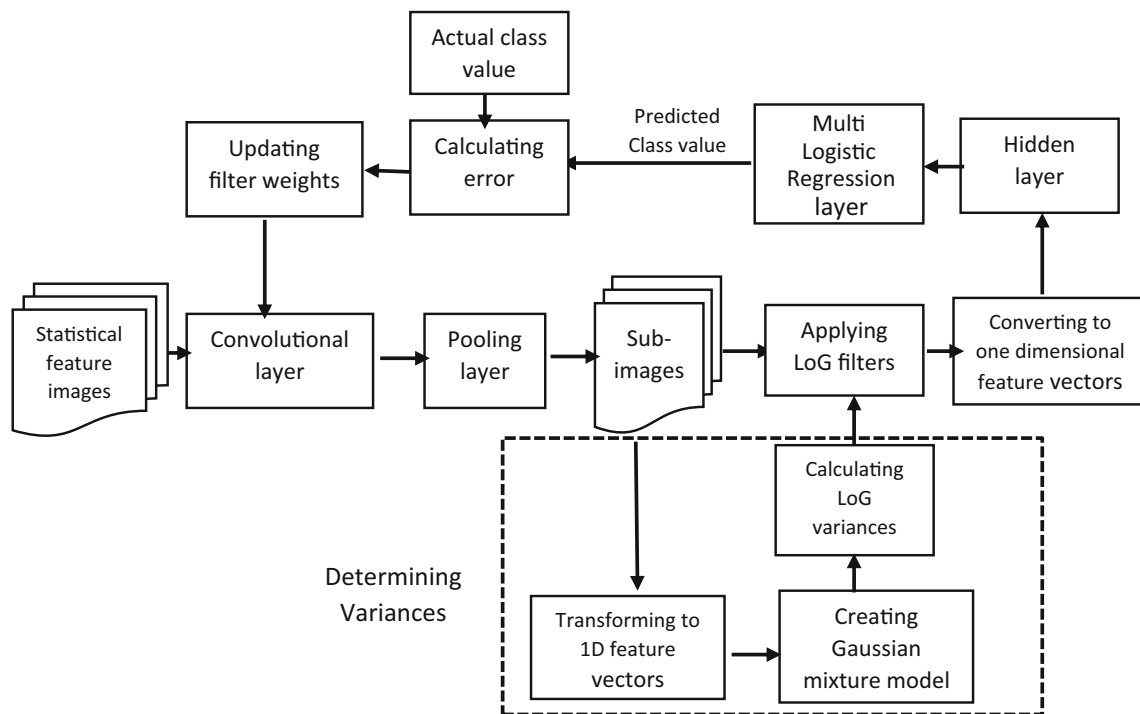


Fig. 7 Block diagram of the LoG-CNN

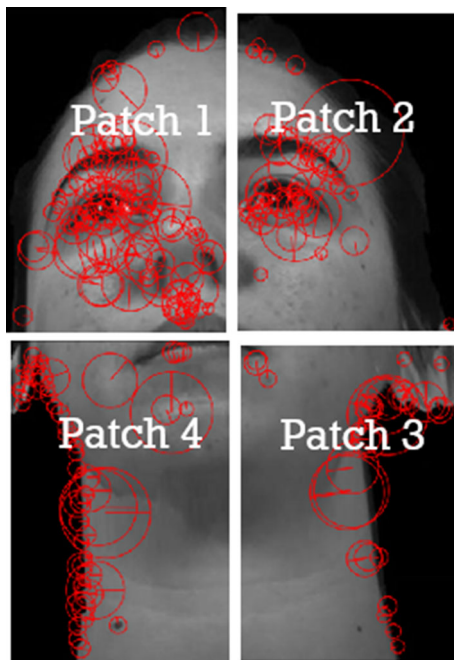


Fig. 8 An example of patch-based SURF feature with four patches (face image is courtesy of [29])

vector is calculated for each patch. These feature vectors are merged to form the temporal feature image. Figure 10 shows a face image with 64 patches. Our Mahalanobis distance-based analysis shows that the Hu-feature vectors provide

good separation among different classes of the action units. Figure 11 supports this observation as the Mahalanobis distances among each sets are greater than zero.

3 Experiments and results

We used the FERA 2017 [29] dataset which includes several facial expressions of different individuals under different views. The proposed spatial-temporal statistical feature image with LoG-DCNN is compared with raw image, traditional SIFT, and SURF as image descriptor together with the classifiers including DCNN, SVM, and KNN. Figure 3 illustrates the flow of experiments for the comparison. The parameters of the classifiers compared are provided in Table 1. The comparison is conducted for ten action units listed in Fig. 12.

We exploited a multi-label classification approach in all DCNN models; as a cost function we used binary cross-entropy. Note that each action unit is considered as a class. After testing with different parameters, the best model is selected. The number of patches used for statistical SURF feature image is 4 and that of Hu is 64. The number of percentile used in spatial feature image is 38 (skipping the first and last ones given the increment of 2.5 percentile). A total of 100 detected key points are selected from each sample image to obtain the SIFT and SURF feature vectors (Fig. 13).

Fig. 9 Two architectures that utilize spatial/temporal feature image with DCNN

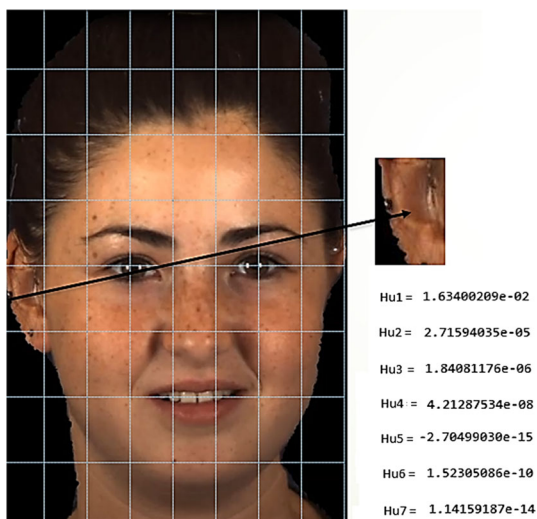
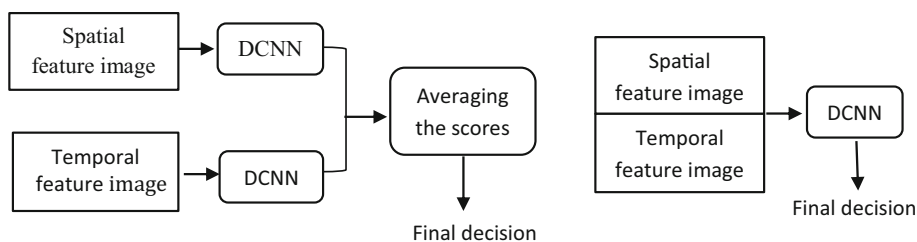


Fig. 10 An image of 64 patches; a Hu-feature vector is extracted per patch (face image is courtesy of [29])

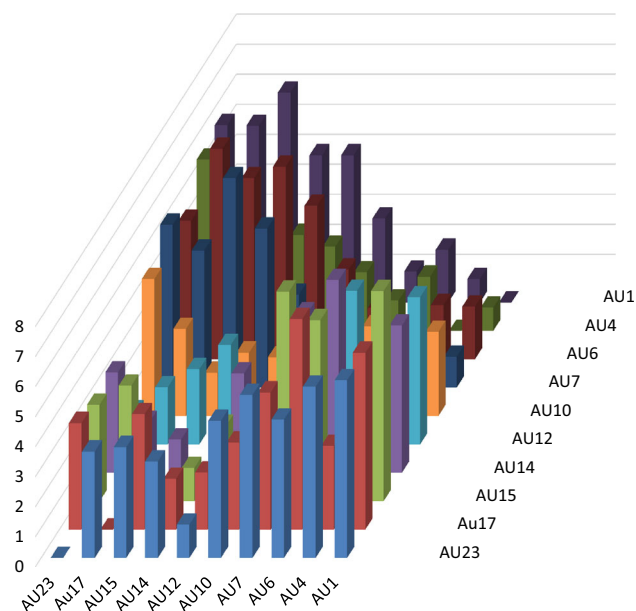


Fig. 11 Mahalanobis distance among the sets of HU-feature vectors from different action units

Table 1 Parameters of the classifiers used in comparison

Classifier Attribute	KNN	DCNN	SVM (three kernels)
K	5	–	–
Training method	–	Stochastic gradient descent method with mini-batches	–
Number of hidden units	–	50	–
Number of convolutional layers	–	2	–
Number of hidden layers	–	1	–
Learning rate	–	0.1	–
Batch size	–	100	–
Penalty parameter C	–	–	1.0
Degree	–	–	3
Gamma	–	–	0.01
Tolerance	–	–	0.001

3.1 Results and discussion

In this section, we compare the proposed LoG–DCNN and the feature image descriptor SFID with other classifiers including regular DCNN, SVM models, KNN and the descriptors including SIFT, SURF, raw image in terms of the testing error and the area under ROC curve (AURC) [30].

Table 2 shows the abbreviations for classifiers and features used in this paper with their explanation. Thereafter, these abbreviations will be used alone or in combination with a classifier and/or a feature. As an example, LoG–DCNN–SURF–S-SFID means that LoG–DCNN uses SURF as a spatial SFID.

Figure 14 shows the testing and training average errors out of 10 action units some methods used in this paper. As can be seen in the figure SFID gave a smoother training and testing phase compared to RI.

Figure 15 compares the proposed feature images and classifiers with different deep and non-learning approaches based upon average testing error out of 10 action units. Among

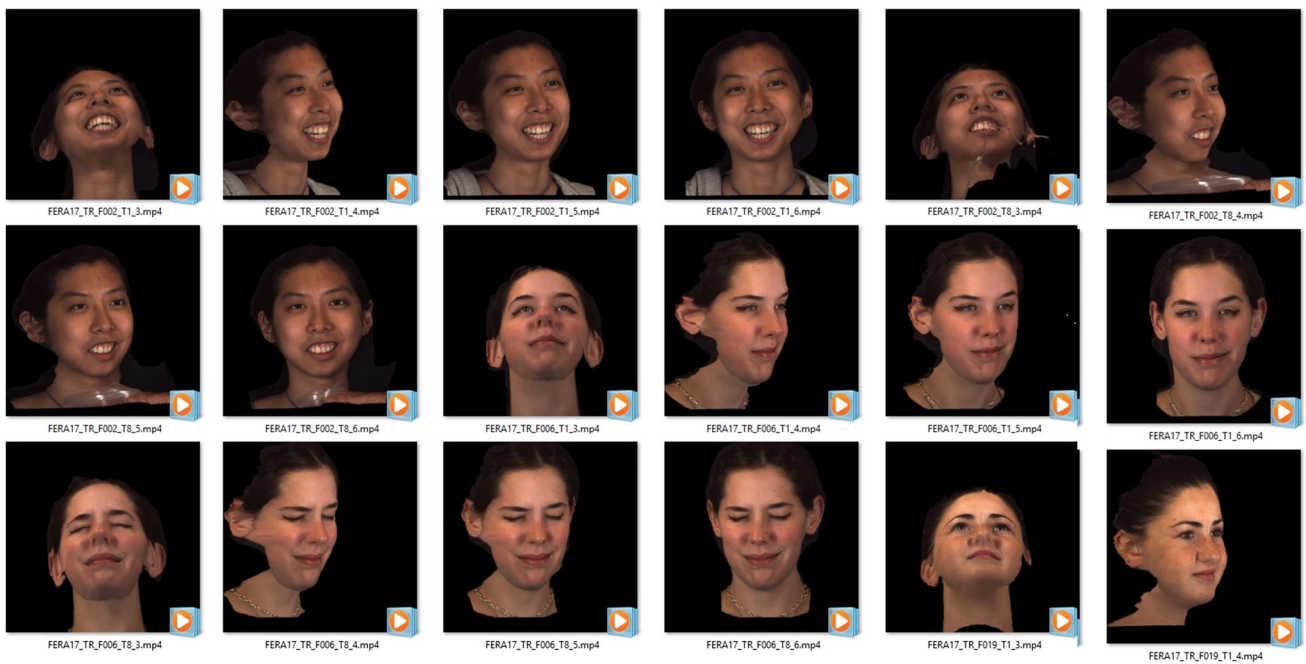


Fig. 12 Some videos used from FER2017 datasets [29]

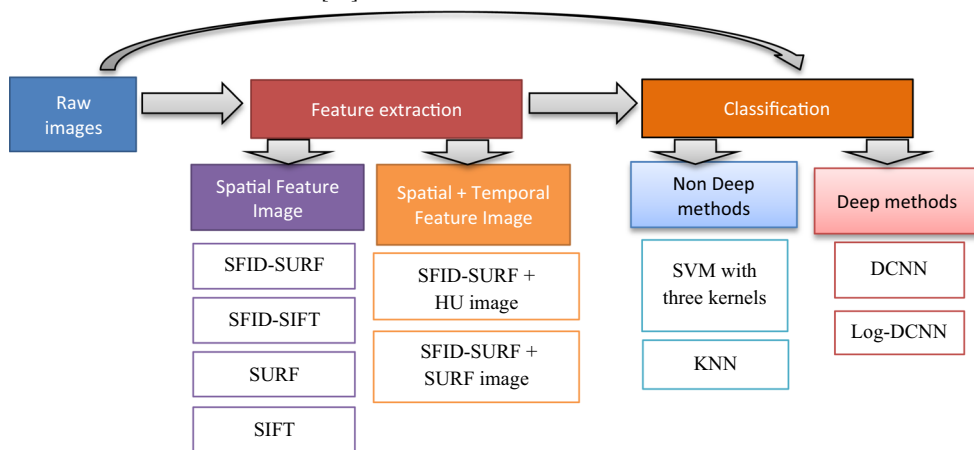


Fig. 13 Organization of experiments conducted to compare different descriptors and methods

Table 2 Abbreviations and their definitions

Abbreviation	Definition
LoG-DCNN	The proposed LoG-DCNN
LoG-DCNN-2	Two separate LoG-DCNNs are used one for spatial SFID and the other for temporal feature image
SVM-P	SVM with polynomial kernel
SVM-L	SVM with linear kernel
SVM-R	SVM with RBF kernel
SURF-S-SFID	SURF is used to generate spatial SFID
SIFT-S-SFID	SIFT is used to generate spatial SFID
SURF-ST-SFID	SURF is used to generate spatial SFID and temporal feature image
SURF-HU-ST-SFID	SURF is used to generate spatial SFID and HU moments are used to generate temporal feature image
SURF-I	SURF image
SIFT-I	SIFT image
RI	Raw image

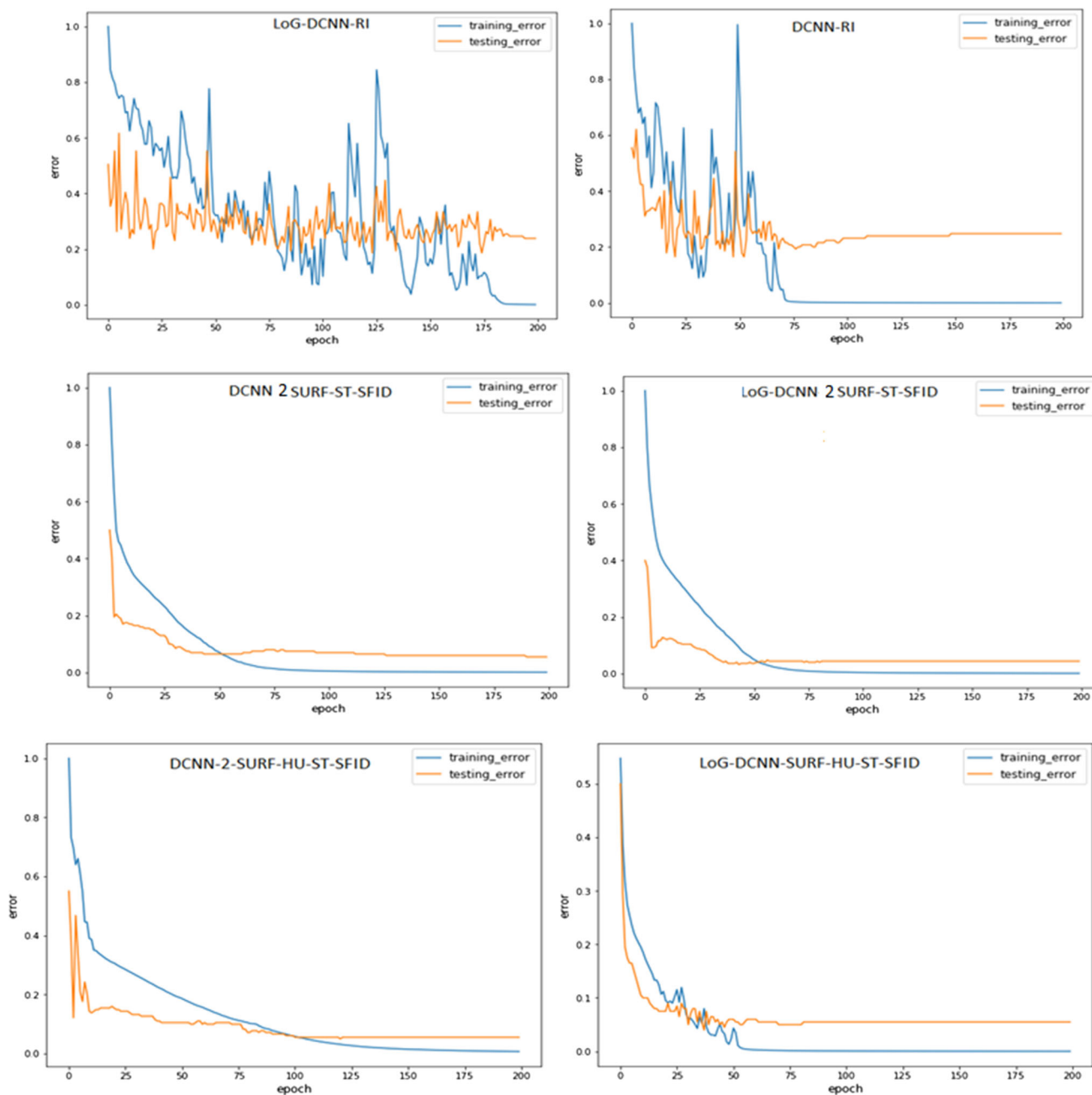


Fig. 14 Training versus testing average error curves of some methods used in this research

the deep learning methods, the best performance is achieved by using LoG-DCNN-SURF-ST-SFID (5.3% testing error) and for the non-deep approaches, SVM-L-SURF-S-SFID (9.54% testing error) produced the best accuracy. In general, LoG-DCNN gave the best accuracy when any feature or raw image is used. Moreover, using the SFID resulted in a remarkable better performance compared to SURF, SIFT, or Raw image when a deep learning method is used.

In comparison with our proposed statistical feature images with the LoG-DCNN, the SURF-ST-SFID gives the highest performance of 5.37% in aver-

age testing error. LoG-DCNN-2-SURF-ST-SFID and LoG-DCNN-SURF-S-SFID follow the highest performance with 6.03 and 6.13%, respectively. SURF-HU-ST-SFID produces closer results using LoG-DCNN (8.37%) or LoG-DCNN-2 (9.84%). SIFT-S-SFID resulted in the testing error of 13.74%. In contrast, LoG-DCNN-RI (26.18%), LoG-DCNN-SIFT-I (25.96%), and LoG-DCNN-SURF-I (26.45%) achieve significantly lower performance compared to the statistical feature images.

As with the proposed LoG-DCNN, the proposed SURF-ST-SFID with regular DCNN achieved the highest per-

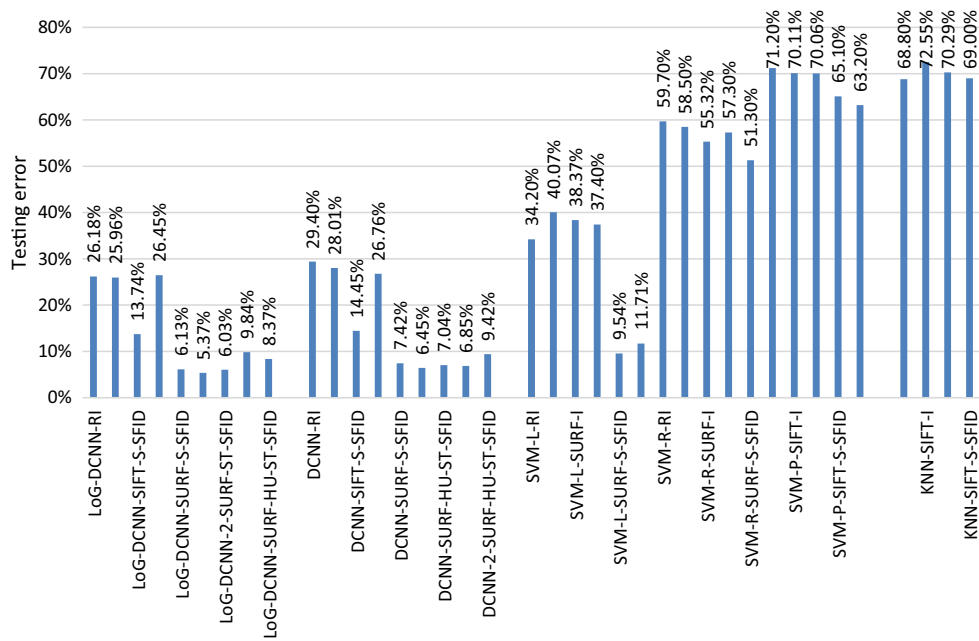


Fig. 15 The average testing error of action units for different methods

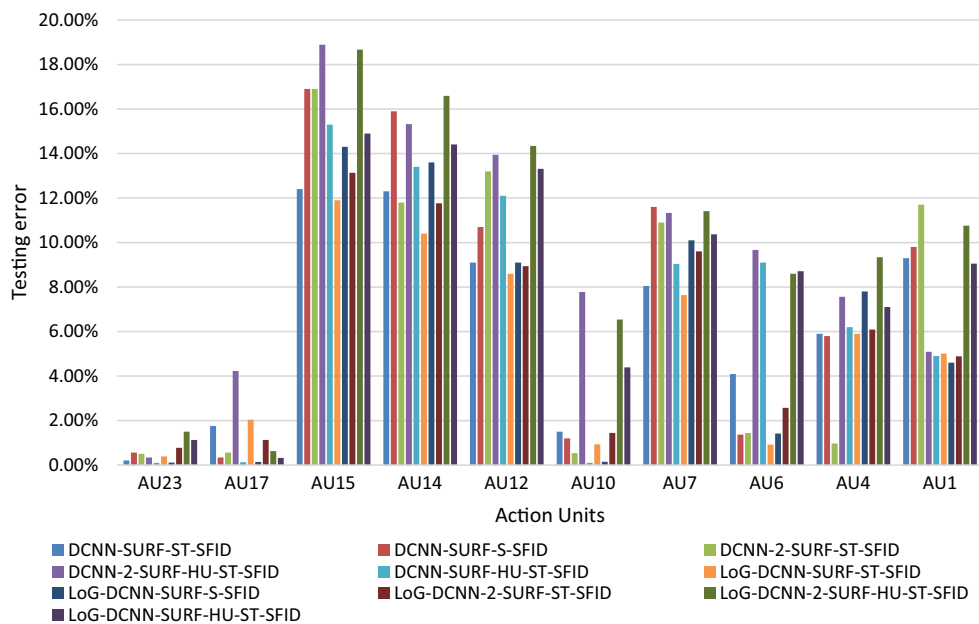


Fig. 16 The testing error of 10 action units using different deep learning and feature image approaches

formance of 6.45% testing error followed by the DCNN-2-SURF-ST-SFID of 6.85%. DCNN-SURF-HU-ST-SFID caused a testing error of 7.04% which is lower than that of DCNN-2-SURF-HU-ST-SFID (testing error of 9.42%). DCNN-SURF-S-SFID resulted in a better performance with testing error of 14.45% compared to that of DCNN-SIFT-SFID (testing error of 14.45%). Using SURF-I, -SIFT-I, and RI with DCNN gave the testing error of 26.76, 28.01, and 29.40%, respectively, which are remark-

ably higher than the testing errors of the statistical feature images.

For the methods based on the SVM classifier, SVM-L-SURF-S-SFID resulted in the highest accuracy with testing error of 9.54%. SVM-L-SURF-ST-SFID was in the second place with testing error of 11.71% that shows that using temporal features with non-deep methods might not be as good as with deep methods. SIFT-S-SFID yielded the low accuracy with SVM classifiers of testing error 37.40, 57.30,

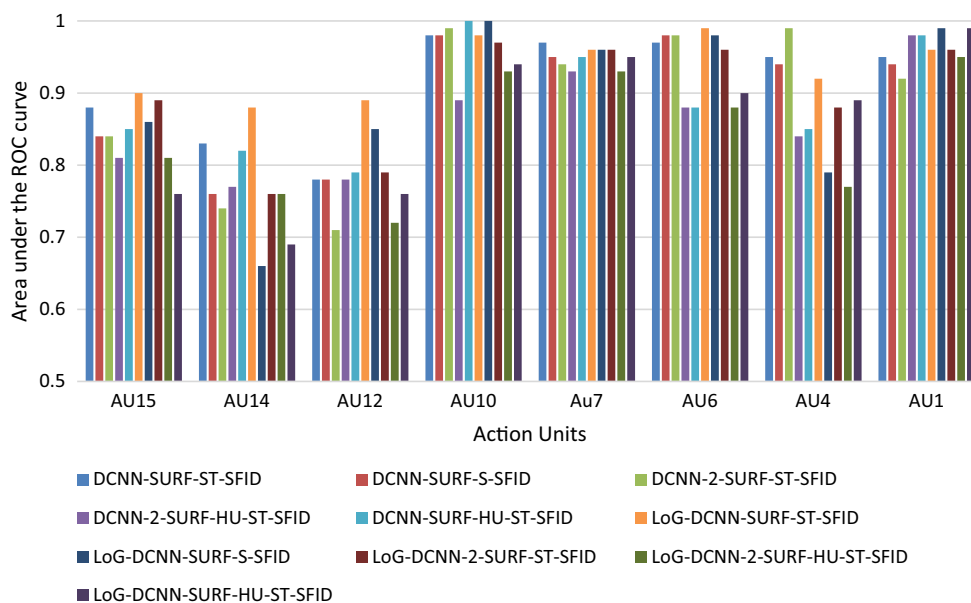


Fig. 17 AURC values for different methods and action units

and 65.10% for SVM-L, SVM-R, and SVM-P, respectively. SURF-I and SIFT-I gave close results; the testing errors for SIFT-I (and SURF-I) are 40.07% (38.37%) with SVM-L, 58.50% (55.32%) with SVM-R, and 70.11% (70.06%) with SVM-P. Using RI caused the worst performance for SVM-P and SVM-R (testing errors of 71.20 and 59.70%) and an average performance for SVM-L (testing error of 34.20%). The performance of the KNN is shown to be the lowest with the testing errors for SIFT-S-SFID (69.00%), SURF-I (70.29%), and SIFT-I (72.55%). Generally, the results show that SVM-L was the best classifier among the non-deep methods. Comparing the SVM methods, the linear SVM performed significantly better than its other variants. This would be due to linearly distributed nature of the sample population. Inherently, the results would be an example of SVM's theoretical optimality criterion.

Figure 16 compares the different deep learning approaches for 10 action units based on the testing error. The results show that some action units can be recognized significantly better than others; as an example, AU15, AU14, AU12, and AU7 were recognized with the highest error than others. In most of the cases, the proposed LoG-DCNN and feature image outperform other classifier and descriptors. In general, the accuracy was high when the proposed spatial-temporal feature images were used. AU23 had a lowest error of classification among all the action units. DCNN-SURF-HU-ST-SFID led to best classification accuracy with the error 0.10% that is followed by 0.11% achieved by LoG-DCNN-SURF-S-SFID. For AU23, the worst classifier was LoG-DCNN-2-SURF-HU-ST-SFID with the error of 1.50%. Similar to AU23, AU17 and AU10 were also

classified with high accuracy. While, the lowest errors for AU17 and AU10 are achieved by DCNN-SURF-HU-ST-SFID with errors of 0.13 and 0.10%, respectively, the highest errors are obtained by DCNN-2-SURF-HU-ST-SFID with the error of 4.23 and 7.78%, respectively. In contrast, AU15 had the highest error of classification. While, the best accuracy for AU15 is achieved by LoG-DCNN-SURF-ST-SFID with the error of 11.90% followed by DCNN-SURF-ST-SFID with the error of 12.40%, the worst classifier was DCNN-2-SURF-HU-ST-SFID with the error of 18.89%. Like as AU23, AU14 and AU12 had high errors of classification. The highest errors are obtained using LoG-DCNN-2-SURF-HU-ST-SFID that were 16.59 and 14.34% for AU14 and AU12, respectively.

Figure 17 compares the different methods for 10 action units based on the area under ROC curves (AURC). For AU1, all the methods resulted in acceptable performance with the range between the minimum AURC of 0.92 achieved by DCNN-2-SURF-ST-SFID and the maximum AURC of 0.99 achieved by LoG-DCNN-SURF-HU-ST-SFID and LoG-DCNN-SURF-S-SFID. For AU4, DCNN-2-SURF-ST-SFID resulted in the highest performance (AURC of 0.99). The second and the third were DCNN-SURF-ST-SFID (AURC of 0.95) and DCNN-SURF-S-SFID (AURC of 0.94). LoG-DCNN-2-SURF-HU-ST-SFID (AURC of 0.77) and LoG-DCNN-SURF-S-SFID (AURC of 0.79) were the worst methods for classifying AU4. For AU 6, LoG-DCNN-SURF-ST-SFID resulted in the highest performance (AURC of 0.99) while DCNN-2-SURF-HU-ST-SFID, DCNN-SURF-HU-ST-SFID and LoG-DCNN-2-SURF-HU-ST-SFID gave the lowest one

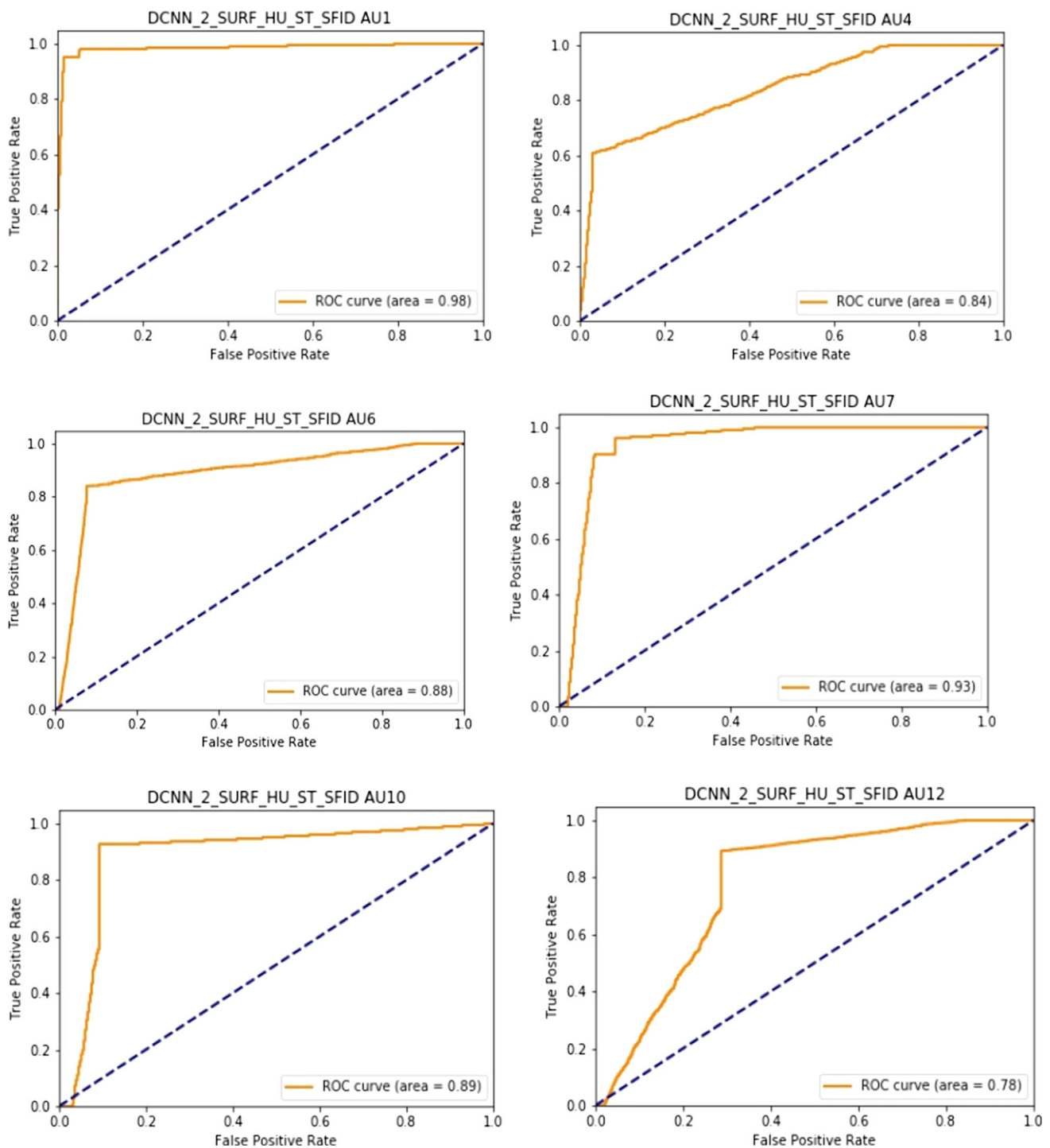


Fig. 18 ROC curves and AURC of different action units for the method DCNN_2_SURF_HU_ST_SFID

(AURC of 0.88). For AU7, just as AU1, the performances were high using any method with the range between the minimum AURC of 0.93 and the maximum AURC of 0.97 achieved by DCNN-2-SURF-HU-ST-SFID and DCNN-SURF-ST-SFID, respectively. For AU12, only LoG-DCNN-SURF-ST-SFID (AURC of 0.89) and

LoG-DCNN-SURF-S-SFID (AURC of 0.85) resulted in the acceptable performance. AU14 was the most challenging action units since only LoG-DCNN-SURF-ST-SFID resulted in a fair performance (AURC of 0.88), while LoG-DCNN-SURF-S-SFID was the worst classifier (AURC of 0.69). Figure 18 shows the ROC curves

and AURC values of different action units for the method DCNN_2_SURF_HU_ST_SFID.

4 Conclusion

In this paper, we proposed an image content representation SFID and adaptive LoG-based DCNN structure to classify multi-view facial action units. The proposed representation utilizes the spatial-temporal statistical content of the video stream. We compared SIFT and SURF descriptors from which the SFID is constructed. Our results show that SURF-SFID outperforms SIFT-SFID. Secondly, to handle the variation of full pose change of face images, we proposed to add an adaptive Laplacian of Gaussian layer into the traditional DCNN structure. The parameters of the LoG layer are determined by the mixture of Gaussian models. The results show promising performance compared to the raw data, SIFT, SURF as well as regular DCNN, SVM models, and KNN.

We aim to improve data collection scheme for better description of action units. An automatic detection of action unit areas would improve classification performance when un-related regions of the face are masked out in the preprocessing stage.

Appendix

An example of obtaining SFID (Figs. 19, 20, 21, 22).

Assume that $K = 4$, $L = 5$, $N = 10$, and D is given as

$$D = \begin{bmatrix} 0.55 & 0.48 & 0.07 & 0.94 & 0.73 & 0.19 & 0.69 & 0.78 & 0.43 & 0.06 \\ 0.43 & 0.17 & 0.76 & 0.09 & 0.81 & 0.17 & 0.43 & 0.56 & 0.14 & 0.09 \\ 0.01 & 0.04 & 0.38 & 0.86 & 0.03 & 0.40 & 0.18 & 0.03 & 0.56 & 0.02 \\ 0.69 & 0.97 & 0.02 & 0.32 & 0.99 & 0.76 & 0.54 & 0.09 & 0.07 & 0.65 \\ 0.31 & 0.02 & 0.01 & 0.19 & 0.42 & 0.81 & 0.19 & 0.21 & 0.87 & 0.33 \end{bmatrix}$$

Then, we get $Pr_4 = (20, 40, 60, 80)$. The percentile values of the first row $S = (0.55, 0.48, 0.07, 0.94, 0.73, 0.19, 0.69, 0.78, 0.43, 0.06)$, values of the first dimension, in D is calculated as $Per_1(S) = 0.07$, $Per_2(S) = 0.43$, and so on. The final SFID will be

$$SFID_4 = \begin{bmatrix} 0.07 & 0.09 & 0.02 & 0.07 & 0.02 \\ 0.43 & 0.17 & 0.03 & 0.32 & 0.19 \\ 0.55 & 0.43 & 0.18 & 0.65 & 0.31 \\ 0.73 & 0.56 & 0.4 & 0.76 & 0.42 \end{bmatrix}$$

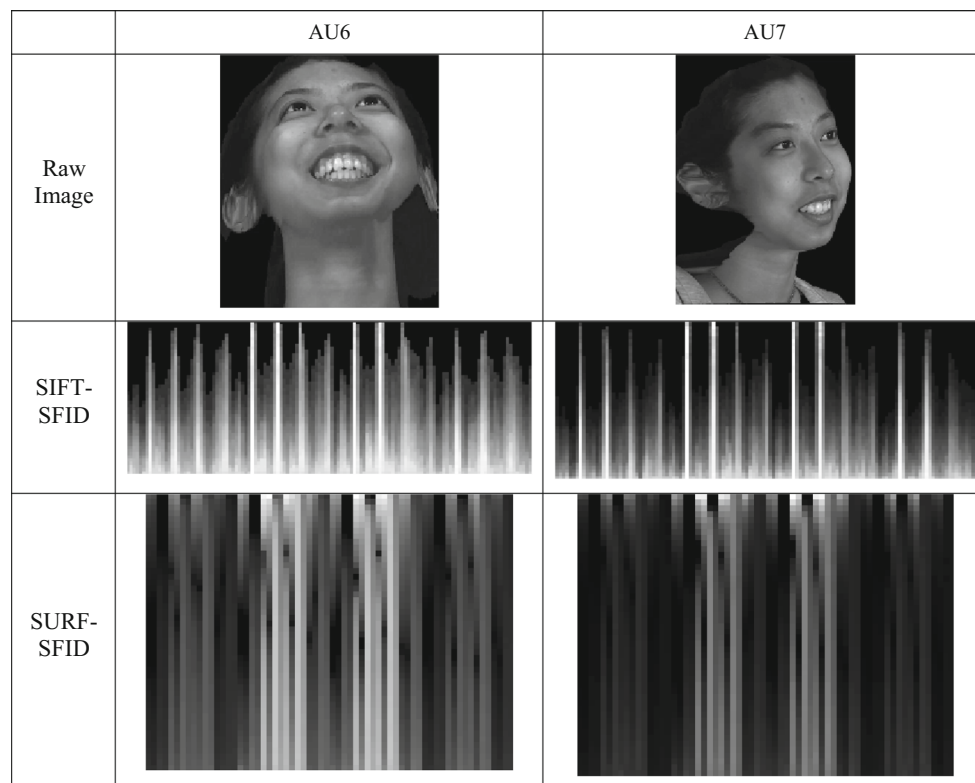


Fig. 19 Examples of SFIDs for AU6 and AU7 (SFID images are transposed)



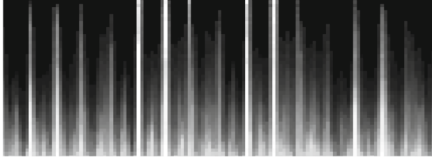
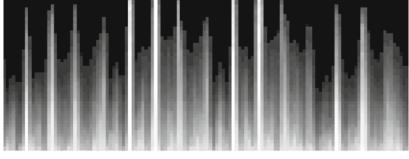
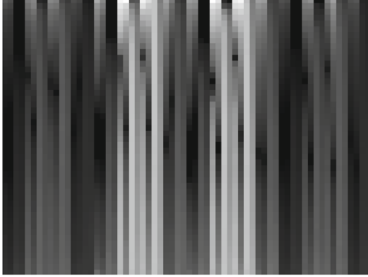
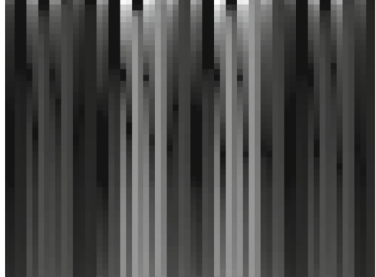
	AU10	AU12
Raw Image		
SIFT-SFID		
SURF-SFID		

Fig. 20 Examples of SFIDs for AU10 and AU12 (SFID images are transposed)



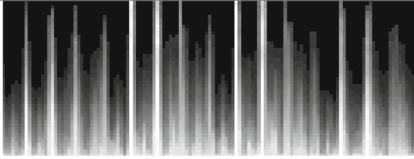
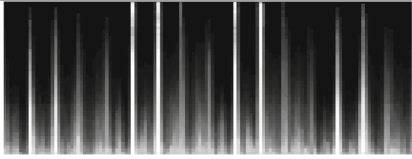
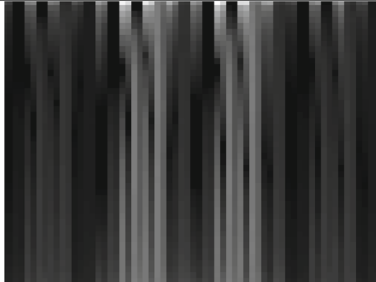
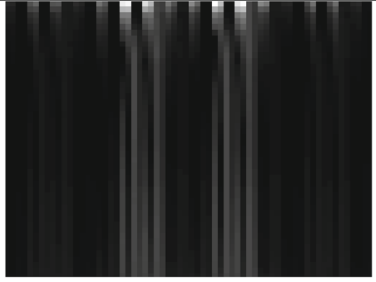
	AU14	AU15
Raw Image		
SIFT-SFID		
SURF-SFID		

Fig. 21 Examples of SFIDs for AU14 and AU15 (SFID images are transposed)

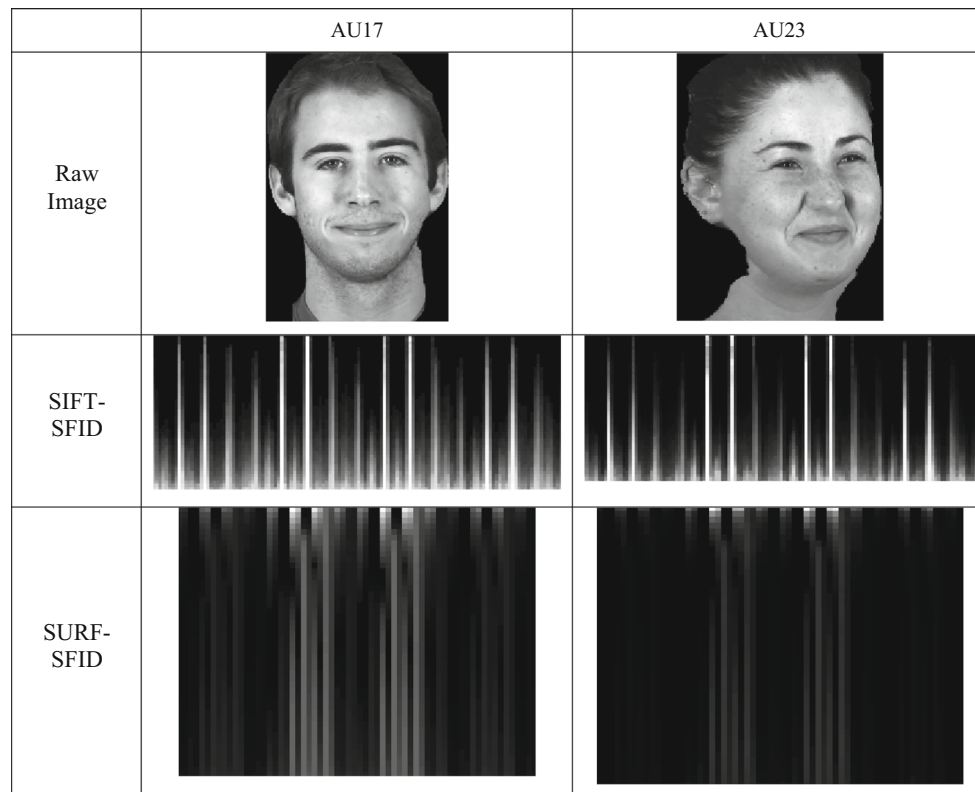


Fig. 22 Examples of SFIDs for AU17 and AU23 (SFID images are transposed)

References

- Sariyanidi, E., Gunes, H., Cavallaro, A.: Automatic analysis of facial affect: a survey of registration, representation, and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(6), 1113–1133 (2015)
- Kumar, P., Happy, L.S., Routray, A.: A real-time robust facial expression recognition system using HOG features. In: International Conference on Computing, Analytics and Security Trends (CAST) (2016)
- Eleftheriadis, S., Rudovic, O., Pantic, M.: Multi-conditional latent variable model for joint facial action unit detection. In: IEEE International Conference on Computer Vision (2015)
- Tian, Y.I., Kanade, T., Cohn, J.F.: Recognizing action units for facial expression analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(2), 97–115 (2001)
- Ambadar, Z., Schooler, J.W., Cohn, J.F.: Deciphering the enigmatic face: the importance of facial dynamics in interpreting subtle facial expressions. *Psychol. Sci.* **16**(5), 403–410 (2005)
- Tong, Y., Liao, W., Ji, Q.: Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(10), 1683–1699 (2007)
- Kauser, N., Sharma, J.: Automatic facial expression recognition: a survey based on feature extraction and classification techniques. In: International Conference on Business Industry & Government (ICTBIG) (2016)
- Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., Ma, Y.: PCANet: a simple deep learning baseline. *IEEE Trans. Image Process.* **24**(12), 5017–5032 (2015)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 1097–1105 (2012)
- Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008)
- Hany, U., Lutfi, A.: Speeded-up robust feature extraction and matching for fingerprint recognition. In: International Conference on Electrical Engineering and Information Communication Technology (ICEEICT) (2015)
- Siritanawan, P., Kotani, K.: Facial action units detection by robust temporal features. In: 7th International Conference on Soft Computing and Pattern Recognition (SoCPaR) (2015)
- Yang, J., Wu, S., Wang, S., Ji, Q.: Multiple facial action unit recognition enhanced by facial expressions. In: 23rd International Conference on Pattern Recognition (ICPR) (2016)
- Lopes, A.T., de Aguiar, E., De Souza, A.F., Oliveira-Santos, T.: Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recogn.* **61**, 610–628 (2017)
- Zhao, K., Chu, W.S., Zhang, H.: Deep region and multi-label learning for facial action unit detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
- Ghasemi, A., Denman, S., Sridharan, S., Fookes, C.: Discovery of facial motions using deep machine perception. In: IEEE Winter Conference on Applications of Computer Vision (2016)
- Lv, Y., Feng, Z., Xu, C.: Facial expression recognition via deep learning. In: International Conference on Smart Computing (SMARTCOMP) (2014)
- Zhu, Y., Shang, Y., Shao, Z., Guo, G.: Automated depression diagnosis based on deep networks to encode facial appearance and dynamics. *IEEE Trans. Affect. Comput.* (2017)

19. Jaiswal, S., Valstar, M.: Deep learning the dynamic appearance and shape of facial action units. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2016)
20. Mishra, B., Fernandes, S.L., Abhishek, K., Alva, A., Shetty, C., Ajila, C.V., Shetty, P.: Facial expression recognition using feature based techniques and model based techniques: a survey. In: *2nd International Conference on Electronics and Communication Systems (ICECS)* (2015)
21. Drira, H., Amor, B.B., Srivastava, A., Daoudi, M., Slama, R.: 3D face recognition under expressions, occlusions, and pose variations. *Trans. Pattern Anal. Mach. Intell.* **35**(9), 2270–2283 (2013)
22. Elaiwat, S., Bennamoun, M., Boussaid, F., El-Sallam, A.: 3-D face recognition using curvelet local features. *Signal Process. Lett.* **21**(2), 172–175 (2014)
23. Tie, Y., Guan, L.: A deformable 3-D facial expression model for dynamic human emotional state recognition. *IEEE Trans. Circuits Syst. Video Technol.* **23**(1), 142–157 (2013)
24. Zheng, W.: Multi-view facial expression recognition based on group sparse reduced-rank regression. *IEEE Trans. Affect. Comput.* **5**(1), 71–85 (2014)
25. Bishops, C.M.: Mixture models. In: *Pattern Recognition and Machine Learning*, pp. 424–460. Springer (2006)
26. Sotak, G., Boyer, K.: The Laplacian-of-Gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output. *Comput. Vis. Gr. Image Process.* 147–189 (1989)
27. Huertas, A., Medioni, G.: Detection of intensity changes with subpixel accuracy using Laplacian–Gaussian masks. *IEEE Trans. Pattern Anal. Mach. Intell.* 651–664 (1986)
28. Hu, M.K.: Visual pattern recognition by moment invariants. *IRE Trans. Inf. Theory* **8**(2), 179–187 (1962)
29. A European network of excellence in social signal processing [Online]. Available: <http://sspnet.eu/fera2017/>
30. Swets, A.J.: *Signal Detection Theory and ROC Analysis in Psychology and Diagnostics: Collected Papers*. Lawrence Erlbaum Associates, Mahwah (1996)

Masoud Z. Lifkooee received an MSc degree in electrical engineering from University of Guilan in 2013 where he worked on image processing and pattern recognition projects such as sign language recognition. Then, he moved to Louisiana State University to continue his research with Dr. Omer Soysal in 2015 where he worked on some machine learning and machine vision projects such as vehicle classification and facial expression recognition. He is currently working with Dr. Xin Li in Louisiana State University department of computer engineering on human body modeling for virtual reality environments and especially to be used in the Cave Automatic Virtual Environment.

Ömer M. Soysal earned his PhD in 2009 in Computer Science from the Louisiana State University and A&M College, USA. His BS is in Electronics Engineering from Hacettepe University, Turkey, in 1992. His research interests include computer vision, machine learning, data mining, medical informatics, geographic information systems, and spatiotemporal data analysis. He serves as an Assistant Professor in Southeastern Louisiana University, USA, and as an adjunct faculty in Louisiana State University, USA.

Kazim Sekeroglu received B.S. degree in Electronics and Computer Education from Gazi University, Ankara, Turkey, in 2007, M.S. degree in Electrical Engineering from Southern University and A&M College, Baton Rouge, LA, in 2011, and Ph.D. degree in Electrical Engineering from Louisiana State University, Baton Rouge, LA, in 2017. He is now an instructor in Computer Science and Industrial Technology at Southeastern Louisiana University, Hammond, LA. His research interests are computer vision, image processing, and data analysis.