IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 14, 2019, accepted May 29, 2019, date of publication June 5, 2019, date of current version June 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921220

Recognizing Facial Expressions Using a Shallow Convolutional Neural Network

SI MIAO[1], HAOYU XU[2], ZHENQI HAN[1], AND YONGXIN ZHU[1,3], (Senior Member, IEEE)
[1]Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China
[2]Lenovo Research, Shanghai 201203, China
[3]School of Microelectronics, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding authors: Haoyu Xu (xuhy@sari.ac.cn) and Yongxin Zhu (zhuyongxin@sari.ac.cn)

This work was supported in part by the Science and Technology Commission of Shanghai Municipality under Grant 17511106400 and Grant 17511106402, in part by the Aerospace System Department under Grant 30508020301, and in part by the Natural Science Foundation of China under Grant 61201059.

ABSTRACT Generally, facial expressions could be classified into two categories: static facial expressions and micro-expressions. There are many promising applications of facial expression recognition, such as pain detection, lie detection, and babysitting. Traditional convolutional neural network (CNN)-based methods suffer from two critical problems when they are adopted to recognize micro-expressions. First, they are usually dependent on very deep architectures that overfit on small datasets. However, reliable expressions are relatively difficult to collect and relevant datasets are usually relatively small. Second, for micro-expressions, these methods usually neglect the temporal redundancy of micro-expressions which could be utilized to reduce the temporal complexity. In this paper, we propose a shallow CNN (SHCNN) architecture with only three layers to classify static expressions and micro-expressions simultaneously without big training datasets. To better explain the functionality of our SHCNN architecture, we improve the saliency maps by introducing a shrinkage factor after studying the vanishing gradient problem of existing saliency maps. Experiments are conducted on five open datasets: FER2013, FERPlus, CASME, CASME II, and SAMM. To the best of our knowledge, by comparing with other methods offering source code (or pseudo code), we believe that our method would be the best on FERPlus, CASME, and CASME II and competitive on FER2013 and SAMM.

INDEX TERMS Facial expression, CNN, saliency analysis.

I. INTRODUCTION

Inferring unspoken meaning from facial cues is a human instinct. Scientists conduct extensive researches on expression recognition in many fields such as acoustic, natural language processing, neuroscience and computer vision. Facial expression recognition is the most popular among them since vision is the most basic human sense. Since Picard [1] proposed affective computing, recognizing expressions by machine has been a frontier research topic. Micro-expressions only last for 1/25 to 1/5 second and their movements are subtle, therefore recognizing them is more challenging than recognizing static facial expressions.

In the literature, micro-expression recognition is implemented in a great variety of approaches [2]. These approaches are usually categorized into three groups, i.e., statistical [3], [4] approaches, deep learning approaches [5] and Apex frame based approaches [6]–[8]:

Statistical approaches attempt to combine traditional machine learning methods (e.g., SVM, random forest) and image processing methods (e.g., LBP, HOG, HOOF [9]). They first extract the handcraft features of expressions, then use statistical classifiers to classify these handcraft features. However, with the success of deep learning, their performances are no longer the best (Table 6, 7, 8).

Deep Learning based approaches use deep architectures to learn intrinsic features of expressions. Unfortunately, high-resolution expressions are very difficult to collect, therefore the relevant datasets are relatively small compared to large high-resolution datasets like ImageNet [10] and UCF-101 [11]. The deep architectures might overfit on these datasets that are relatively small. Some approaches employ spatiotemporal architectures, such as 3D convolution [12], [13] and CNN-LSTM architecture [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

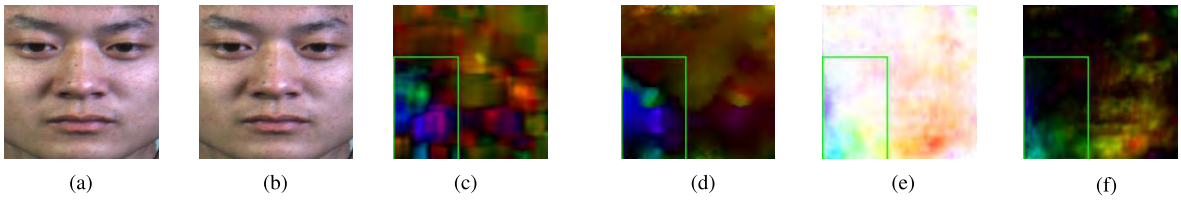


FIGURE 1. Comparison of three kinds of optical flow. (a) and (b) are two frames selected from CASME II. The differences between them are hard to recognize. (c) LK optical flow, (d) TV-L1 optical flow and (e)(f) flownet2 are used to visualize the differences. The left-down corner highlights the micro-expression despite that the movement is very subtle.

These spatiotemporal architectures are somehow inspired by video action recognition. However, they do not take full advantage of a characteristic of micro-expressions (but not a characteristic for video actions): the micro-expression clips are usually short and have much temporal redundancy, i.e., the frames do not vary too much from each other. For example, if a person is smiling in one frame, then the person is very likely to smile in the nearby frames. By taking advantage of temporal redundancy, we could simplify the neural networks to better fit on fewer samples. For example, the Temporal Interpolation Model (TIM) [14] is used to reduce the video frames, and even some researches only use Apex frames. The datasets in action recognition, e.g., UCF-101 [11] and HMDB51 [15]–[17], are usually large. Therefore, spatiotemporal architectures are successful in the field of action recognition. Due to the limited size of expression datasets, these architectures may not work well on expression recognition. In addition, spatiotemporal architectures use massive parameters to fit the temporal relationships in micro-expressions. However, temporal relationships in micro-expressions are relatively simple. Expressions contained in consecutive frames are highly similar. Therefore intuitively, we should design a shallower network with lower temporal complexity.

Apex frame based approaches notice the temporal redundancy. Li *et al.* [6] and Zhang *et al.* [7] only use frames near apex frames for training. However, these approaches face a problem. The training samples are few even using all frames. If we only use apex frames (or frames near them) and discard other frames, much useful information is given up.

To address the insufficient performance issue in statistical methods, overfitting issue in deep learning approaches and missing information issue in apex frame based approaches, we propose a shallow CNN named SHCNN. Firstly, instead of using temporal architectures, we use TV-L1 optical flow [18], [19] (Fig. 1), which is fast and accurate, to extract the temporal features. Since TV-L1 is enough to achieve good results, we give up FlowNet2 [20] in our experiments since it is slow and difficult to be integrated into our architecture. For each video with n frames, we calculate the optical flow between the first frame and the last n-1 frames and get n-1 optical flow images. Then we use SHCNN to classify each optical flow image and finally employ the voting strategy to decide the final category of the video. Moreover, since SHCNN does not include spatiotemporal architecture,

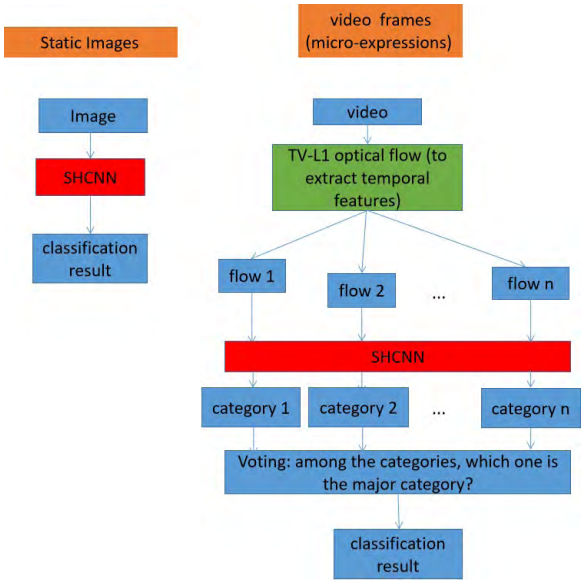


FIGURE 2. General pipeline of static expression/micro-expression classification. The two tasks(static expression recognition and micro-expression recognition) share SHCNN architecture but have different weights. The saliency map is used to visualize the functionality of the proposed SHCNN.

it could also be used for static expression analysis (Fig. 2). One might ask why SHCNN classifies optical flow images instead of classifying video frames directly. That is because micro-expressions are embodied in movements. Our main contributions are follows:

- We propose to use a shallow network (SHCNN) that alleviates the overfitting issue for datasets that are relatively small.
- We propose a simple but practical pipeline (Fig. 2) without deep spatiotemporal architectures. The pipeline could take full advantage of temporal redundancy in micro-expressions.
- We study the vanishing gradient problem of the original saliency map [21]. Moreover, we improve the saliency map by introducing a shrinkage factor to better visualize SHCNN.
- Experiments on five public datasets (FER2013, FERPlus, CASME, CASME II, SAMM) show that our method performs favorably against the state-of-the-art.

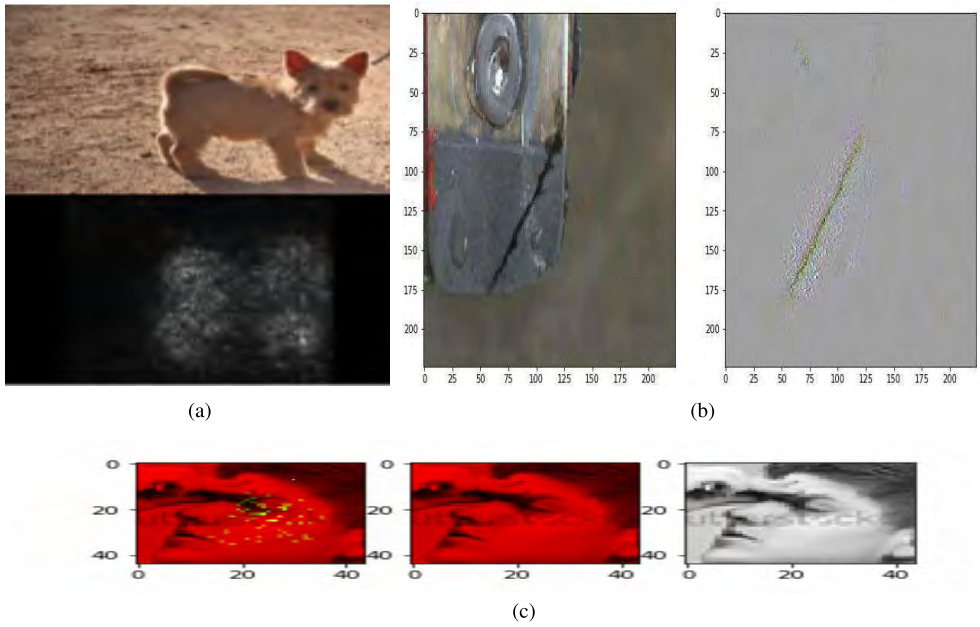


FIGURE 3. Saliency maps of objects and expressions. (a) Dog and its saliency map (taken from [21]). (b) Crack and its saliency map (our method). (c) Saliency map of facial expression (marked by green points, taken from FER2013 [32]).

II. RELATED WORK

A. STATIC EXPRESSION RECOGNITION

Before 2012 when Alexnet [22] achieved huge success on ImageNet [10], handcraft features and SVM were widely used to classify expressions. For example, [23]–[26] used Local Binary Pattern (LBP) to extract features. After 2012, with stronger GPUs and larger datasets, CNNs were widely used in expression recognition. Y.Tang listed the top in the kaggle FER2013 competition, he used a simple neural network to extract features of static images and SVM hinge loss for classification, indicating that neural network is able to do better without handcraft features [27]. In 2016, Barsoum *et al.* [28] proposed four training strategies on FERPlus, getting ∼85% accuracy and showing that noisy label problem was critical for FER tasks. Yang *et al.* [29] used original images and LBP features as two kinds of features and proposed a double channel CNN to classify them. A geometric model (DFSN) was proposed in [30] to preprocess the facial features and the preprocessing method helped to increase the recognition accuracy.

B. MICRO-EXPRESSION RECOGNITION

We briefly review the representative researches for the three approaches mentioned in the Introduction:

(a) *Statistical approaches*: Li *et al.* [3] used LBP and HOOF to extract features, TIM to reduce temporal complexity and SVM for classification.

(b) *Deep Learning approaches*: Khor *et al.* [5] proposed ELRCN architecture. ELRCN used CNN to extract spatial features and LSTM to learn the temporal relationship between these spatial features. They finally got 50.0% F1-score on CASME II database. Wang *et al.* [13] proposed a method based on two-stream 3D CNN pretrained on macro-expression datasets.

(c) *Apex frame based approaches*: Li *et al.* [6] used frames near the apex frames for micro-expression recognition. To highlight expressions, they employed the Eulerian method [31] to magnify the subtle changes. Their work is a milestone for apex frame based methods. The main idea of these approaches is reducing the temporal complexity and enhancing the video movements.

C. SALIENCY MAPS

To visualize CNN, Simonyan *et al.* [21] proposed the saliency map of CNNs. Its basic idea is to determine the effect of each pixel to the classification of whole image I (HxW). Mathematically, if $S_i(1 \leq i \leq n)$ are n score functions (the outputs after softmax), saliency map determines the effect of each pixel by calculating the gradients of S_i with respect to I: $g_i := \nabla_I S_i$, and if g_i is large at some pixel P, increasing the intensity of P increases the probability that the image is recognized as class i. If the image I is categorized as class j, we define the pixels of I that have positive correlations with the score function S_j as saliency pixels:

$$saliency \quad pixels := \{p \in I | \frac{\partial S_j}{\partial p} \geq 0\}. \tag{1}$$

Fig. 3(a) shows the saliency map of a dog [21] from the ImageNet [10]. Fig. 3(b) shows that the saliency area of the crack is very close to the actual place, therefore our saliency map could be used for localization. Fig. 3(c) is a demo of the saliency map on the face. Unlike the saliency area of objects,

the attention points in the saliency map of expressions are dispersed, but we notice that the saliency pixels mainly gather around the man's left eye, so we can infer that it is the left eye that affects his expression (neutral) most.

III. PROPOSED METHOD

A. THE FACIAL EXPRESSION DATASETS

The proposed algorithm is tested on five public datasets: FER2013 [32], FERPlus [28], CASME [33], [34], CASME II [35] and SAMM [36]. The details of datasets are listed below:

(1) *FER2013*. The FER2013 dataset consists of 35887 grayscale face images of size 48×48 collected from the internet and used for kaggle challenge. Each image is labeled with one of the seven kinds of expressions (angry, disgust, fear, happy, sad, surprise and neutral) depending on its item on the internet. Due to the unreliability of the Internet, there are many noisy labels in this dataset [28]. It contains three subsets: Training, PublicTest and PrivateTest, containing 28709, 3589, 3589 grayscale images respectively. We use PrivateTest for validation and PublicTest for test.

(2) *FERPlus*. To solve the noisy label problem in FER2013, Barsoum *et al.* [28] tagged the images again and used probability distribution instead of a unique tag to determine the category of each image. For example, if 4 taggers think an image is neutral while the other 6 taggers think it is disgust, then the image is tagged as {neutral: 4, disgust: 6, others: 0} while one-hot encoding treats it as {disgust: 1, others: 0}. Images and three sets are totally the same with FER2013, but tags are different. We discard 520 images in the PublicTest because they are labeled as ''unclear''.

(3) *CASME*. There are eight kinds of expressions in CASME: tense, happiness, repression, surprise, disgust, fear, contempt and sadness. There are 19 subjects, 189 videos in the dataset. The distribution is: tense (69 videos), happiness (9 videos), repression (38 videos), surprise (20 videos), disgust (44 videos), fear (2 videos), contempt (1 video), sadness (6 videos). We only use the first 5 classes because there are few samples in the last 3 classes.

(4) *CASME II*. Seven kinds of expressions are included in CASME II, they are happiness, others, disgust, repression, surprise, fear and sadness. There are 26 subjects, 255 videos and 16781 frames in the dataset. The distribution is: happiness (32 videos, 2319 frames), others (99 videos, 6336 frames), disgust (63 videos, 4153 frames), repression (27 videos, 2150 frames), surprise (25 videos, 1514 frames), fear (2 videos, 66 frames), sadness (7 videos, 243 frames). For the same reason, we only use the first 5 classes of CASME II. We use LOSO (Leave One Subject Out) protocol in our experiments on CASME and CASME II.

(5) *SAMM*. Compared with CASME and CASME II which only consist of Chinese subjects, SAMM is a novel dataset whose subjects are selected from a diverse range of age and ethnicity. We classify the SAMM into three categories: Positive (Happiness), Negative (Anger, Fear, Disgust, Contempt) and Surprise. These three categories contain 26, 92, 15 samples respectively, hence SAMM is an unbalanced database.

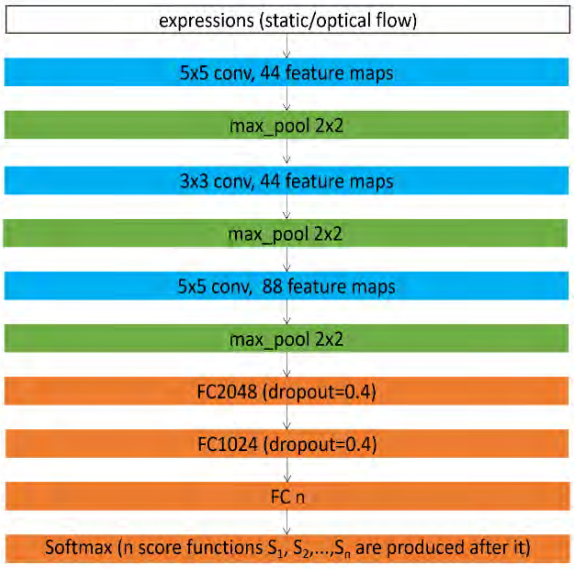


FIGURE 4. SHCNN architecture.

B. SHCNN ARCHITECTURE

The network architecture is shown in Fig. 4.

The n in Fig. 4 denotes the number of expression categories. For FER2013, $n = 7$, for FERPlus, $n = 8$, for CASME and CASME II, $n = 5$, for SAMM, $n = 3$. The output of softmax function is a tensor of shape $(batchsize, n)$. For each image I in a batch, there is an array $S = [S_1, S_2, \ldots, S_n]$ indicating the likelihood estimation of the image I, we call S the score functions of I and S_i the score function of the i-th category.

LeakyReLU [37] is used in the SHCNN to avoid ''Dead ReLU problem''. We set coefficient α of LeakyReLU to 0.02. We briefly explain the usage of LeakyReLU: using ReLU is able to converge theoretically, but when we are working on the FER2013 and FERPlus datasets, we find the loss defined in Eq. 2 is stuck at about 1.80 and fails to fall even after 80 epochs. However, with LeakyReLU ($\alpha = 0.02$), the loss falls to 0.0004 and the network converges to a considerable state. We also choose other coefficients of LeakyReLU, for example, 0.2, 0.1, but the performance changes very slightly. We could draw an empirical conclusion: as long as the network could converge, the coefficient α does not matter significantly. However, using ReLU instead of LeakyReLU might cause the failure of convergence.

C. PREPROCESSING

We first convert all video frames into grayscale images. For a video V and every frame F in V, if F is not the first frame of V, we calculate the TV-L1 optical flow between the first frame of V and F. The optical flow is firstly resized into 112×112 and then transformed into an HSI image.

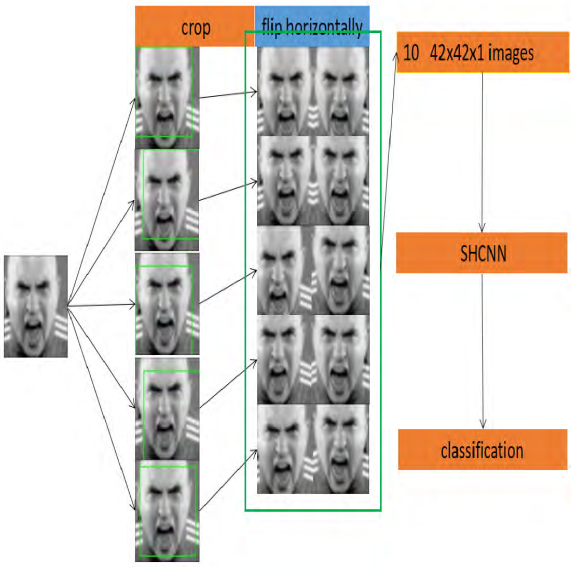


FIGURE 5. Classification pipeline for static expressions.

The saturation component is set to 255 constantly, the hue component and the intensity component represent argument and magnitude of the motion vectors in the optical flow respectively. Finally the HSI image is transformed into a RGB image (like images in Fig. 1) which is used to train SHCNN.

As for static expressions, we crop the top left, top right, bottom left, bottom right and middle 42×42 patches from each image, and flip each patch horizontally. Therefore, 10 patches are derived from one single original image. The preprocessing pipeline for static expressions is illustrated in Fig. 5.

D. TRAINING

The loss $\mathcal{L}$ of SHCNN is defined as

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n} p_{ij} log\, S_{ij} + \lambda \sum_{w} ||w||^2, \tag{2}$$

where N means the batch size, n means the number of classes and λ is a l_2-regularizer. p_{ij} denotes the probability that the i-th image in the batch is in the j-th class. For FERPlus we use the given probability tags and for other datasets we use one-hot tags. We set the initial learning rate of Adam [38] optimizer to 0.001 for both tasks. We set λ to 0.0003 and N to 100. We do a comparative study of the learning rate in Fig. 8. When the learning rate is 0.001, the SHCNN converges faster on FERPlus than 0.0001. As for micro-expressions, the difference of using learning rates of 0.001 and 0.0001 is very slight. The difference is random (after running our codes several times, sometimes 0.001 is slightly better and sometimes 0.0001 is slightly better), thus we could categorize the difference as a random error. We use Tensorflow and a GTX 1080Ti to implement our algorithms. For source code, please look at `https://github.com/miaosiSari/Affective`.

TABLE 1. Explanation of the four cases.

$S_j \to 1$	$j = k$	$\frac{\partial S_j}{\partial o_k} = S_j(1 - S_j) \to 0$
$S_j \to 1$	$j \neq k$	$\frac{\partial S_j}{\partial o_k} = -S_j S_k \to 0 (S_k \to 0)$
$S_j \to 0$	$j = k$	$\frac{\partial S_j}{\partial o_k} = S_j(1 - S_j) \to 0$
$S_j \to 0$	$j \neq k$	$\frac{\partial S_j}{\partial o_k} = -S_j S_k \to 0$

E. INFERENCE

For micro-expressions, we employ a voting method for classifying each video. We simply see which class is the major class in the video. For example, a video has 3 frames, two of them are judged as neutral and the third one is judged as sad, then we judge the video as neutral. If there are exactly two major classes, we simply use the class of the apex frame as the class of the video. Fortunately there is always an evident major class for each video in our experiment, so that dilemma never happens in our evaluation.

As for static images, we employ a boosting method for evaluation. For each test image P, let P_1, P_2, …, P_{10} be 10 derived images (Fig. 5) and $S_{ij}(1 \le i \le 10, 1 \le j \le n)$ be the score functions after the softmax layer of the i-th derived image of P. We determine the category of a test image P by: $class(P) = \underset{1\le j\le n}{\text{argmax}} \sum_{1\le i\le 10} S_{ij}$.

F. SALIENCY MAP

The method in [21] is important for CNN visualization. However we find it has vanishing gradient problem if the network is trained quite well (some class function is extremely close to 1). The **score function** S_j is computed by the softmax function:

$$S_j := \frac{exp(o_j)}{\sum_{i=1}^{n} exp(o_i)}, \tag{3}$$

where $o_1, o_2, \ldots, o_n$ are the outputs of the last Fully Connected layer. The original saliency map

$$original \quad saliency \quad map := \nabla_I S_j, \tag{4}$$

computes the gradients of the score function S_j with respect to the original image I. By chain rule:

$$\nabla_I S_j = \sum_{k=1}^{n} \frac{\partial S_j}{\partial o_k} \nabla_I o_k. \tag{5}$$

Take l_2-estimation on both sides of Eq. 5:

$$||\nabla_I S_j||_2 \le (\sum_{k=1}^{n} \left|\frac{\partial S_j}{\partial o_k}\right|) \max_{1\le k\le n} ||\nabla_I o_k||_2. \tag{6}$$

Through the definition of S_j, we get:

$$\frac{\partial S_j}{\partial o_k} = \begin{cases} S_j(1 - S_j) & j = k, \\ -S_j S_k & j \neq k. \end{cases} \tag{7}$$

The problem is if the network works very well, the network is likely to have very high belief on one class. Therefore there

TABLE 2. Three numerical examples are given to illustrate the vanishing gradient problem.

Examples	$[o_i, S_i, V_i]$	$S(S_{ij} = \frac{\partial S_i}{\partial o_j})$	Our score function $V(V_{ij} = \frac{\partial V_i}{\partial o_j})$
Example 1	$\begin{pmatrix} 1 & 1.23e{-}4 & 0.21 \\ 3 & 9.11e{-}4 & 0.26 \\ 10 & 9.99e{-}1 & 0.53 \end{pmatrix}$	$\begin{pmatrix} 1.23e{-}4 & -1.12e{-}7 & -1.23e{-}4 \\ -1.12e{-}7 & 9.10e{-}4 & -9.10e{-}4 \\ -1.23e{-}4 & -9.10e{-}4 & 1.03e{-}3 \end{pmatrix}$	$\begin{pmatrix} 0.017 & -0.0056 & -0.011 \\ -0.0056 & 0.019 & -0.014 \\ -0.011 & -0.014 & 0.025 \end{pmatrix}$
Example 2	$\begin{pmatrix} 1 & 1.01e{-}43 & 0.21 \\ 7 & 4.08e{-}41 & 0.22 \\ 100 & 1.00 & 0.57 \end{pmatrix}$	$\begin{pmatrix} 1.01e{-}43 & -4.13e{-}84 & -1.01e{-}43 \\ -4.13e{-}84 & 4.08e{-}41 & -4.08e{-}41 \\ -1.01e{-}43 & -4.08e{-}41 & 0 \end{pmatrix}$	$\begin{pmatrix} 0.0017 & -0.00047 & -0.0012 \\ -0.00047 & 0.0017 & -0.0013 \\ -0.0012 & -0.0013 & 0.0025 \end{pmatrix}$
Example 3	$\begin{pmatrix} 1 & 0.27 & 0.38 \\ 2 & 0.73 & 0.62 \end{pmatrix}$	$\begin{pmatrix} 0.20 & -0.20 \\ -0.20 & 0.20 \end{pmatrix}$	$\begin{pmatrix} 0.12 & -0.12 \\ -0.12 & 0.12 \end{pmatrix}$

TABLE 3. Results on FER2013.

Methods	Accuracy
AlexNet[22]	0.6110[43]
AlexNet (reproduced by us)	0.6854
HOG+CNN[44]	0.6186
Xception[45]	0.6620
Network from [43]	0.6640
VGG-8	0.6804
FaceLiveNet[46]	0.6860
Ours (without augmentation)	0.6498
Ours (without regularization)	0.6818
Ours (dropout=0.2)	0.6846
Ours	0.6910

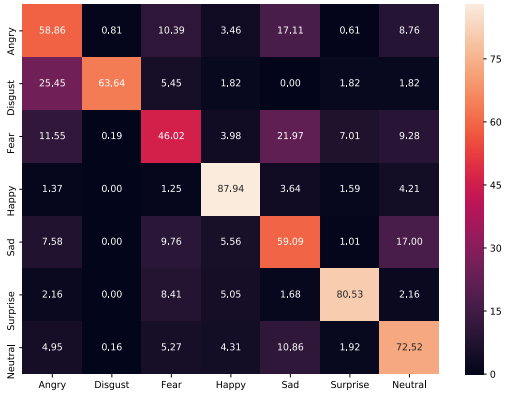


FIGURE 6. Confusion matrix of FER2013.

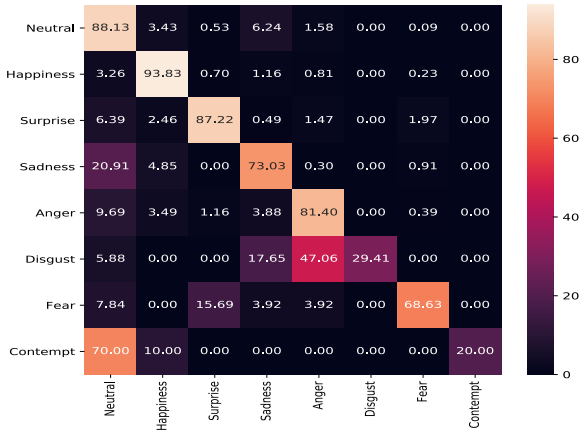


FIGURE 7. Confusion matrix of FERPlus.

likely exists a score function S_j that is extremely close to 1 and other score functions are close to 0 (for example, score functions are [9.99e-01, 1.90e-07, 7.94e-10, 8.75e-05, 5.15e-06, 4.31e-10, 1.96e-11, 2.76e-05] and the first score function 9.99e-01 is extremely close to 1). In that case $\frac{\partial S_j}{\partial o_k}$ is close to 0 no matter S_j is close to 1 or close to 0 (four cases are described in Table 1). The gradient $\nabla_I S_j$ would become very small (sometimes less than 10^{-8}) because $\frac{\partial S_j}{\partial o_k}$ vanishes for every k (Eq. 6). Therefore, the saliency map would become unclear (Fig. 13) and the computation becomes unreliable due to the precision of float32. To tackle this problem, we introduce a shrinkage factor:

$$M := \max_{1 \le j \le n} |o_j|, \tag{8}$$

and define:

$$V_j := \frac{exp(o_j/M)}{\sum_{i=1}^{n} exp(o_i/M)}. \tag{9}$$

Our improved saliency map is defined as:

$$improved \quad saliency \quad map := \nabla_I V_j. \tag{10}$$

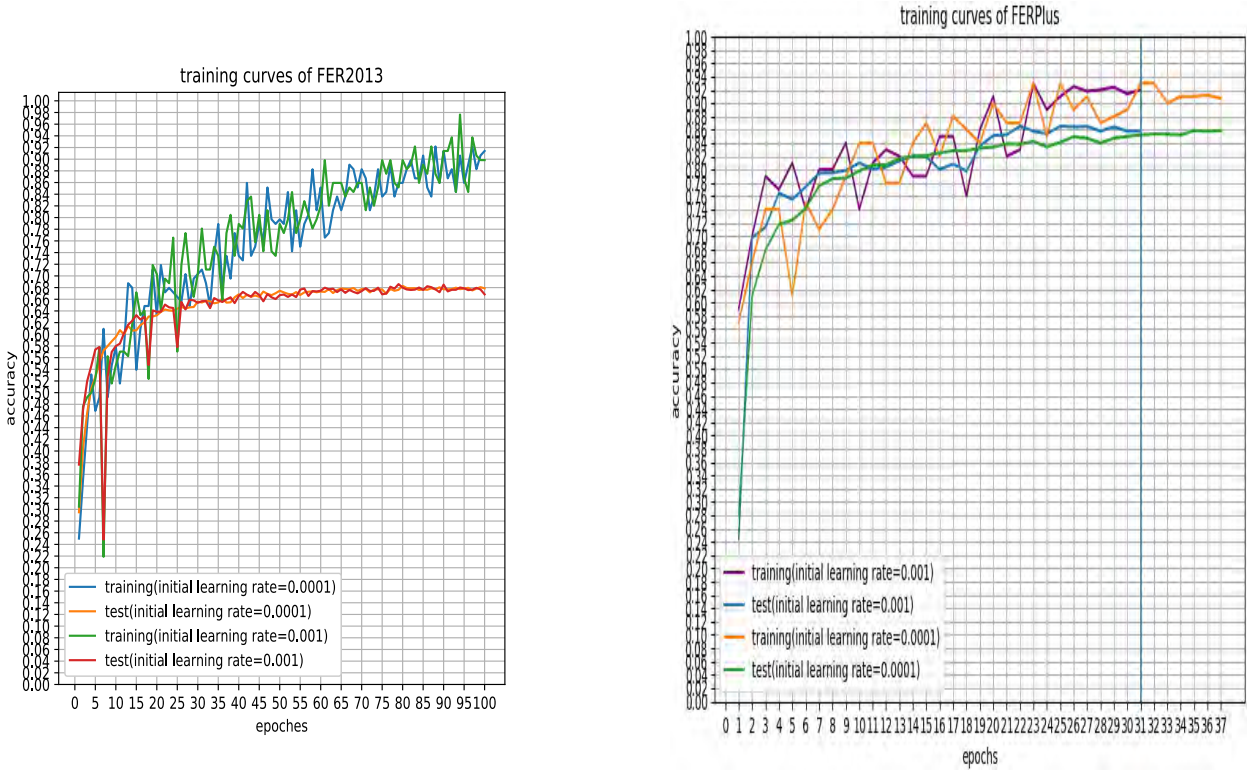
Similar to Eq. 1, the improved saliency pixels are defined as:

$$improved \quad saliency \quad pixels := \{p \in I | \frac{\partial V_j}{\partial p} \ge 0\}. \tag{11}$$

The partial derivative of V_j with respect to o_k is:

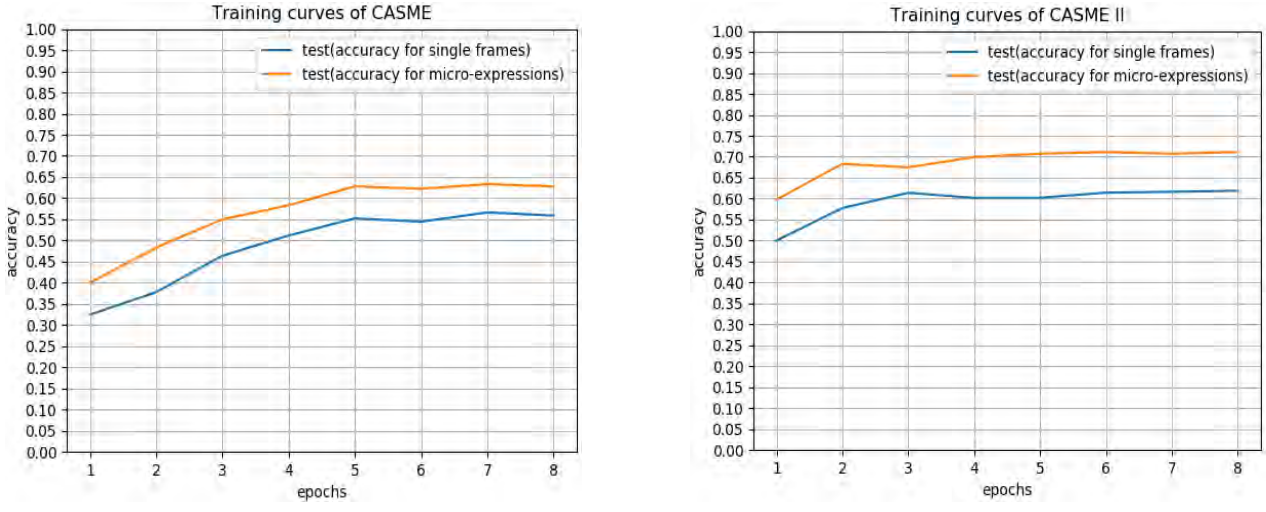
$$\frac{\partial V_j}{\partial o_k} = \begin{cases} \frac{1}{M} V_j(1 - V_j) & j = k, \\ -\frac{1}{M} V_j V_k & j \ne k. \end{cases} \tag{12}$$

Determining the relationship of $\frac{\partial V_j}{\partial o_k}$ and $\frac{\partial S_j}{\partial o_k}$ quantitatively could be very difficult, but we are able to explain the relationship qualitatively. $|\frac{\partial V_j}{\partial o_k}|$ is not required to be greater than $|\frac{\partial S_j}{\partial o_k}|$, but when some $S_k (1 \le k \le n)$ is close to 1 and other S_k are close to 0, using V_k for saliency map solves this problem,



(a) (b)

FIGURE 8. Training curves of FER2013 and FERPlus. SHCNN overfits on FER2013. However, compared with deeper models, SHCNN has higher accuracy (Table 3). (a) Training curves of FER2013. (b) Training curves of FERPlus.



(a) (b)

FIGURE 9. Training curves of all subjects from CASME and CASME II. (a) Training curves of CASME. (b) Training curves of CASME II.

that is because dividing these o_j by M makes these V_j get closer to each other and farther from 1, thus the product $|V_jV_k|$ becomes much bigger than $|S_jS_k|$. Although we have to divide $-V_jV_k$ by M to get $\frac{\partial V_j}{\partial o_k}$ (Eq. 12), $|\frac{\partial V_j}{\partial o_k}|$ is still larger than $|\frac{\partial S_j}{\partial o_k}|$. We give 3 numerical examples in Table 2. For the first example, the max score function, which is $9.98965779e{-}1$, is close to 1, therefore the gradients are very small, but they do not vanish. For the second example, the max score function is extremely close to 1, so the original saliency map is as small as $1e{-}43$ and they vanish. However, with our score functions $\{V_i\}$, the improved saliency map does not vanish for the first

TABLE 4. Training time and parameters of several networks (we use a 1080Ti and float32 data format). The parameters shown in this table are much fewer than the ones trained on ImageNet since the latter use 224 × 224 images as input.

Methods	training time	Parameters
VGG-16	0.14s/100 images	104020999
VGG-8	0.10s/100 images	100886599
Xception	0.10s/100 images	24734048
AlexNet	0.06s/100 images	10033863
SHCNN (ours)	0.05s/100 images	8710963

TABLE 5. Results on FERPlus.

Methods	Accuracy
DenseNet[47]	0.8151[48]
VGG-8	0.8200
VGG-13+MV[28]	0.8448
VGG-13+ML[28]	0.8432
VGG-13+PLD[28]	0.8535
VGG-13+CEL[28]	0.8496
Li et al.[49]	0.8430
Ours (without augmentation)	0.8016
Ours (without regularization)	0.8586
Ours (dropout=0.2)	0.8609
Ours	**0.8654**

two examples. For the third example, the max score function 0.73105858 is still far from 1, so our method does not help a lot in this case.

When we are visualizing the SHCNN on five datasets, the vanishing gradient problem also exists. Sometimes the SHCNN outputs very high belief on one class (the score function S on that class is 0.999998 which is very close to 1), thus the saliency map $\nabla_I S$ vanishes. To improve it, we use $\{V_j\}$ defined in Eq. 9 and calculate $\nabla_I V$ as the improved saliency map instead of $\nabla_I S$. The comparisons with the original saliency maps are listed in Fig. 13 and discussed in Section IV-G.

IV. EVALUATION AND RESULTS

This section discusses the evaluation results on the five benchmark datasets. We use accuracy as an evaluation metric for static expressions. As for micro-expressions, due to the limited samples and the unbalanced datasets, we use F1-Score besides accuracy for evaluation. At last, to prove the problem of the existing saliency map and the effectiveness of the improved saliency map, we offer five examples in Fig. 13 and discuss the correctness of the improved saliency map in subsection IV-G. During evaluation, we mainly compare our work with the researches which offer their source codes or pseudo codes and use similar evaluation protocols with us.

TABLE 6. Results on CASME. (LOSO, in the column'Task', 0 represents disgust, 1 represents surprise, 2 represents repression, 3 represents tense, 4 represents happiness, *:Original paper uses all 8 classes of CASME, so we reproduce it using the same 5 classes as we use. PNSO: The paper classifies the expressions into positive, negative, surprise and others. Our work gets 114 correct among the 180 videos.)

Methods	Task	Accuracy	F1
LBP-SIP[4]	0,1,2,3,4	0.3684	N/A
LBP+SVM	0,1,2,3,4	0.4536	0.4044
3D-FCNN[50]	0,1,2,3,4	0.5444	N/A
FDM[51]	0,1,2,3,4	0.5614*	0.5499*
MDMO (reproduced by us)	0,1,2,3,4	0.5629	0.5551
STCLQP[52]	0,1,2,3	0.6402	N/A
DiSTLBP-IIP[53]	0,1,2,3	0.6341	0.5736
Fuzzy Histogram[54]	PNSO	0.6701	0.5489
Ours (dropout=0.2)	0,1,2,3,4	0.6167	0.6092
Ours (without regularization)	0,1,2,3,4	0.6222	0.6038
Ours	0,1,2,3,4	**0.6333**	**0.6142**

TABLE 7. Results on CASME II. (LOSO, *: Inferred from confusion matrix, &: Original paper uses all 7 classes of CASME II, so we reproduce it using only the same 5 classes as we use. Our work gets 175 correct among the 246 videos.)

Methods	Accuracy	F1
LBP+SVM	0.4605	0.3097
FDM[51]	0.4607&	0.3609&
LBP-SIP[4]	0.4656	0.4480
STCLQP[52]	0.4730	N/A
MDMO (reproduced by us)	0.5169	0.4966
ELRCN[5]	0.5244	0.5000
3D-FCNN[50]	0.5911	N/A
Kim et al.[55] (with a spatiotemporal architecture)	0.6098	N/A
Apex frame based MagGA[6]	0.6330	0.5819*
Hierarchical network[56]	0.6445	0.6185
Hu et al.[57]	0.6620	N/A
Ours (dropout=0.2)	**0.7154**	0.6889
Ours (without regularization)	0.7114	0.6904
Ours	0.7114	**0.6981**

A. RESULTS ON FER2013

Results on FER2013 are listed in Table 3. The number of parameters and the inference time are shown in Table 4. We compare SHCNN with several standard networks such as VGG and some methods from newly published papers. As is seen from Fig. 8(a), it takes about 80 epochs for our SHCNN to reach the summit. The confusion matrix is listed in Fig. 6.

B. RESULTS ON FERPLUS

As for FERPlus, we mainly compare SHCNN with classical networks and the four strategies proposed by Barsoum *et al.* [28]. The DenseNet, which is the most complicated model in Table 5, does not perform considerably on FERPlus. That is possibly because too many parameters

TABLE 8. Results on SAMM. LOSO: Leave One Subject Out. LOVO: Leave One Video Out. CDE: Composite Database Evaluation.

Methods	Protocol	Accuracy	F1
LBP-TOP	LOSO	0.4361	0.3896
M. Peng et al.[58]	LOSO	0.7059	0.5400
ELRCN[5]	CDE	0.7519	0.6409
STSTNet[41]	LOSO	0.7744	0.6588
STRCN[42]	LOSO	0.7860	N/A
STRCN[42]	LOVO	0.8360	0.7920
Ours (dropout=0.2)	LOSO	0.8421	0.7931
Ours (without regularization)	LOSO	0.8496	0.7923
Ours	LOSO	**0.8647**	**0.8029**

TABLE 9. Confusion matrix of SAMM. SHCNN has a considerable performance on SAMM, which is an unbalanced dataset.

	Negative	Positive	Surprise
Negative	**87 (94.56%)**	2 (2.17%)	3 (3.26%)
Positive	9 (34.62%)	**17 (65.38%)**	0 (0.00%)
Surprise	3 (20.00%)	1 (6.67%)	**11 (73.33%)**

cause severe overfitting on FERPlus. Fig. 8(b) illustrates the training curve of FERPlus. When the initial training rate is 0.001, it reaches the summit (0.8654) in the 22nd epoch and the training process finishes in 31 epochs. However when the initial learning rate is 0.0001, the training process is longer (37 epochs) and the best accuracy is lower (0.8537). The confusion matrix is listed in Fig. 7.

C. RESULTS ON CASME

Table 6 shows that our method not only performs high accuracy but also has the highest F1-Score, which means it is able to deal with extremely unbalanced data. Although DiSTLBP-IIP performs higher accuracy, it only involves four classes while our method involves five classes. We have trained our SHCNN 19 times because there are 19 subjects in CASME and we use LOSO policy. We place training curves of CASME in Fig. 9(a). The confusion matrix is listed in Fig. 10.

D. RESULTS ON CASME II

The accuracy/F1-Score, training curves and confusion matrix could be found in Table 7, Fig. 9(b), Fig. 11 respectively. We can see from Table 7 that our method performs the best both in accuracy and F1-Score.

E. RESULTS ON SAMM

The accuracy/F1-Score and the confusion matrix are placed in Table. 8 and Table. 9 respectively. We successfully reproduce the codes of ELRCN and STSTNet shared on Github [39], [40]. ELRCN employs CNN-LSTM architecture. STSTNet [41] employs 3D convolution. STRCN [42] employs 3D convolution and face alignments. However, we find that using SHCNN only without spatiotemporal architectures is enough to get better results.

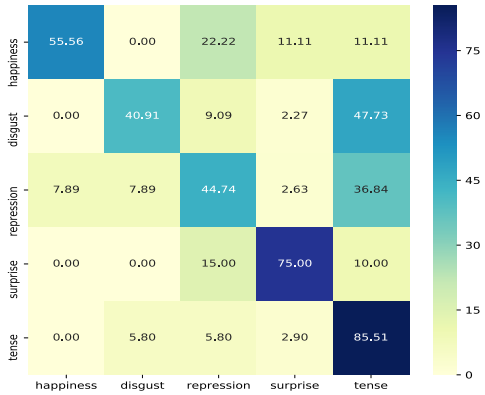


FIGURE 10. Confusion matrix of CASME.

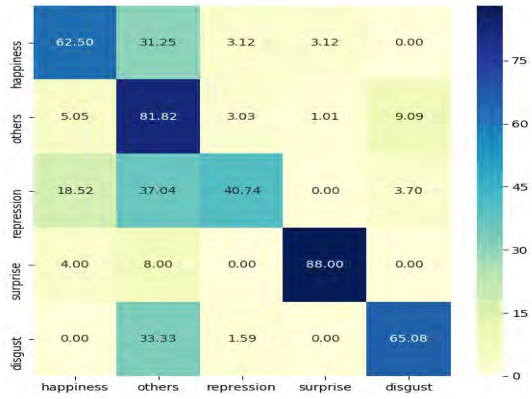


FIGURE 11. Confusion matrix of CASME II.

F. IMPORTANCE OF OPTICAL FLOW

Since SHCNN classifies optical flow images instead of classifying frames directly (Fig. 2), the readers might ask: is SHCNN able to classify the video frames directly instead of the optical flow? The answer is NO. We set up two control groups as follows:

- Group A: we use SHCNN to classify frames of micro-expressions directly without pretrained weights form static expressions.
- Group B: we classify frames directly instead of optical flow images using the pretrained weights from FERPlus.

The results on three micro-expression datasets (CASME, CASME II and SAMM) are presented in Table 10. Table 10 shows that classifying video frames directly would get unsatisfactory accuracies.

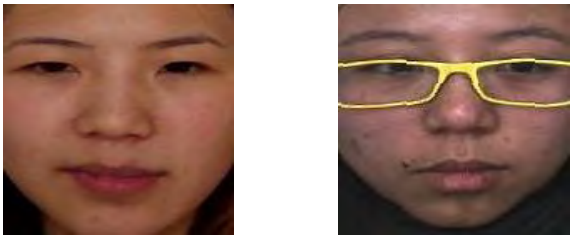
When we use the pretrained weights from FERPlus to classify frames from micro-expressions, we find most frames ($\sim$ 95%), even some apex frames (Fig. 12), are recognized as "neutral". The micro-expressions are embodied in movements, therefore highlighting the movements by optical flow is critically important.

G. DISCUSSIONS ON SALIENCY MAPS

Five examples of saliency maps are listed in Fig. 13. From the first, second, fourth, fifth examples we could see that the vanishing gradient problem exists in the original saliency map

TABLE 10. Classification accuracy (with/without optical flow). If we classify video frames directly without optical flow, the performance would be unsatisfactory.

	CASME	CASME II	SAMM
Group A: without optical flow (not pretrained)	49/180 (0.2722)	55/246 (0.2236)	80/133 (0.6015)
Group B: without optical flow (pretrained)	58/180 (0.3222)	60/246 (0.2439)	89/133 (0.6692)
with optical flow (original pipeline)	114/180 (0.6333)	175/246 (0.7114)	115/133 (0.8647)



(a) (b)

FIGURE 12. Apex frames of micro-expressions are sometimes recognized as neutral when they are viewed as static expressions. (a) Apex frame(1). (b) Apex frame(2).

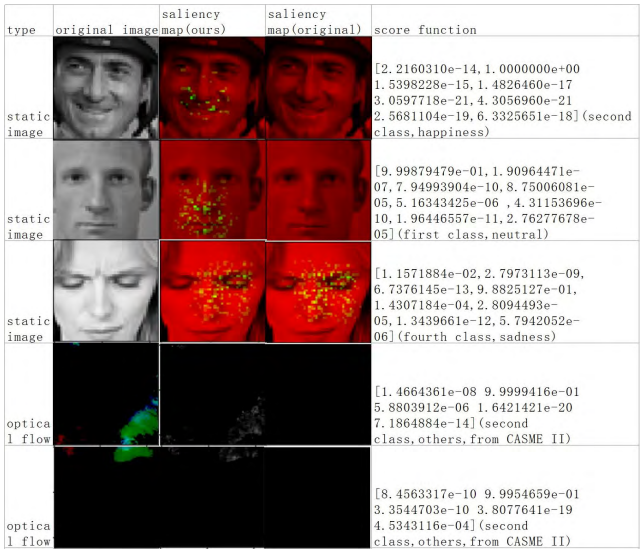


FIGURE 13. Saliency maps of static expressions and optical flow images. The original saliency map suffers the vanishing gradient problem and fails to visualize the SHCNN correctly if one of score functions is close to 1.

(Eq. 4) when one of score functions is close to 1. The saliency pixels also vanish in these four examples. With the improved saliency map (Eq. 10), the vanishing gradient problem is alleviated. Our method does not show much difference on the third example, whose score functions S_i are

$$\begin{bmatrix} 1.1571884e{-}02 \\ 2.7973113e{-}09 \\ 6.7376145e{-}13 \\ 9.8825127e{-}01 \\ 1.4307184e{-}04 \\ 2.8094493e{-}05 \\ 1.3439661e{-}12 \\ 5.7942052e{-}06 \end{bmatrix}$$

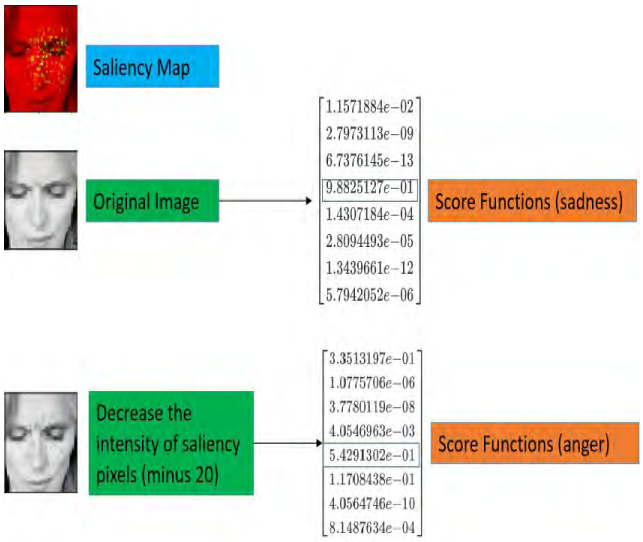


FIGURE 14. Explanation of the third saliency map.

and $max\, S_i = S_4 = 9.8825127e{-}01$. It is because S_4 is not very close to 1, so the vanishing gradient problem does not hold in this case ($\frac{\partial S_4}{\partial o_1} = -1.1571884e{-}02 * 9.8825127e{-}01 = -0.01144$). We conduct a simple adversarial attack on the third image of Fig. 13. We decrease the intensity of the improved saliency pixels (Eq. 11) by 20. Before decreasing, SHCNN judges the original image as'sadness' correctly. However, after decreasing the intensity of saliency pixels, SHCNN treats the image as'anger'. The adversarial example proves that the improved saliency pixels indeed affect the classification (Fig. 14).

V. CONCLUSION

Motivated by the limited training samples and the temporal redundancy, we propose the SHCNN without deep temporal architectures like LSTM and 3D convolution. Some researches, like Apex frames based CNNs, indeed avoid these architectures and achieve good results. However, they also give up some training images and useful information, hence intensify the "data hungry"problem and impair the performance. On the contrary, the SHCNN is able to take full advantages of the training samples. Moreover, SHCNN is simple and could be used for static expression recognition.

Extensive experiments on the five datasets (FER2013, FERPlus, CASME, CASME II and SAMM) show that the shallow architecture (SHCNN) is able to learn both static

expressions and micro-expressions. Moreover, we study the vanishing gradient problem of the original saliency map proposed by Simonyan et al. [21] (Eq. 4) and propose an improved saliency map (Eq. 10) to alleviate the problem.

ACKNOWLEDGEMENT

The authors would like to thank X. Miao for his kind suggestions on the presentation and organization of this paper.

REFERENCES

[1] R. W. Picard, *Affective computing*. Cambridge, MA, USA: MIT Press, 1995.
[2] Y. Oh, J. See, A. C. L. Ngo, R. C. Phan, and V. M. Baskaran, "A survey of automatic facial micro-expression analysis: Databases, methods and challenges," 2018, *arXiv:1806.05781*. [Online]. Available: https://arxiv.org/abs/1806.05781
[3] X. Li, X. Hong, A. Moilanen, X. Huang, T. Pfister, G. Zhao, and M. Pietikäinen, "Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods," *IEEE Trans. Affective Comput.*, vol. 9, no. 4, pp. 563–577, Oct. 2018.
[4] Y. Wang, J. See, R. C. Phan, and Y. Oh, "LBP with six intersection points: Reducing redundant information in LBP-TOP for micro-expression recognition," in *Proc. ACCV*, 2014, pp. 525–537.
[5] H.-Q. Khor, J. See, R. C. W. Phan, and W. Lin, "Enriched long-term recurrent convolutional network for facial Micro-Expression recognition," in *Proc. IEEE Conf. Autom. Face Gesture Recognit. (FG)*, May 2018, pp. 667–674.
[6] Y. Li, X. Huang, and G. Zhao, "Can micro-expression be recognized based on single apex frame?" in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3094–3098.
[7] Z. Zhang, T. Chen, H. Meng, G. Liu, and X. Fu, "SMEConvnet: A convolutional neural network for spotting spontaneous facial micro-expression from long videos," *IEEE Access*, vol. 6, pp. 71143–71151, 2018.
[8] S. Liong, J. See, K. Wong, and R. C. Phan, "Less is more: Micro-expression recognition from video using apex frame," *Signal Process., Image Commun.*, vol. 62, pp. 82–92, Mar. 2018.
[9] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.
[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.
[11] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*. [Online]. Available: https://arxiv.org/pdf/1212.0402.pdf
[12] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.
[13] S.-J. Wang, B.-J. Li, Y.-J. Liu, W.-J. Yan, X. Ou, X. Huang, F. Xu, and X. Fu,"Micro-expression recognition with small sample size by transferring long-term convolutional neural network," *Neurocomputing*, vol. 312, pp. 251–262, Oct. 2018.
[14] Z. Zhou, G. Zhao, and M. Pietikäinen, "Towards a practical lipreading system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 137–144.
[15] 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.
[16] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
[17] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2556–2563.
[18] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.
[19] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, no. 1, pp. 89–97, 2004.
[20] E. Ilg, N. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1647–1655.
[21] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034* [Online]. Available: https://arxiv.org/pdf/1312.6034
[22] A. Krizhevsky, Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.
[23] C. Shan, S. Gong, and P. McOwan, "Robust facial expression recognition using local binary patterns," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2005, pp. 370–373.
[24] G. Zhao and M. Pietikäinen, "Dynamic texture recognition using Local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, Jun. 2007.
[25] T. H. H. Zavaschi, A. L. Koerich, and L. E. S. Oliveira, "Facial expression recognition using ensemble of classifiers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 1489–1492.
[26] M. Richter, T. Gehrig, and H. K. Ekenel, "Facial expression classification on Web images," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 3517–3520.
[27] Y. Tang, "Deep learning using linear support vector machines," 2013, *arXiv:1306.0239*. [Online]. Available: https://arxiv.org/abs/1306.0239
[28] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *Proc. ACM Int. Conf. Multimodal Interact. (ICMI)*, 2016, pp. 279–283.
[29] B. Yang, J. Cao, R. Ni, and Y. Zhang, "Facial expression recognition using weighted mixture deep neural network based on double-channel facial images," *IEEE Access*, vol. 6, pp. 4630–4640, 2017.
[30] Y. Tang, X. Zhang, and H. Wang, "Geometric-convolutional feature fusion based on learning propagation for facial expression recognition," *IEEE Access*, vol. 6, pp. 42532–42540, 2018.
[31] H.-Y. Wu, M. Rubinstein, E. Shih, J. V. Guttag, F. Durand, and W. T. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM Trans. Graph.*, vol. 31, no. 4, p. 65, 2012.
[32] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, and B. Hamner, "Challenges in representation learning: A report on three machine learning contests," in *Proc. Int. Conf. Neural Inf. Process.*, 2013, pp. 117–124.
[33] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, and X. Fu, "CASME database: A dataset of spontaneous Micro-Expressions collected from neutralized faces," in *Proc. IEEE Conf. Autom. Face Gesture Recognit. (FG)*, Apr. 2013, pp. 1–7.
[34] W. Yan, S. J. Wang, Y. Liu, Q. Wu, and X. Fu, "For micro-expression recognition: Database and suggestions," *Neurocomputing*, vol. 136, pp. 82–87, Jun. 2014.
[35] W. J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, and X. Fu,"CASME II: An improved spontaneous micro-expression database and the baseline evaluation," *PLoS ONE*, vol. 9, no. 1, 2014, Art. no. e86041.
[36] A. K. Davison, C. Lansley, N. Costen, K. Tan, and M. H. Yap, "SAMM: A spontaneous Micro-Facial movement dataset," *IEEE Trans. Affective Comput.*, vol. 9, no. 1, pp. 116–129, Jun. 2018.
[37] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 30, 2013, pp. 1–3.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[39] H.-Q. Khor, J. See, R. C. W. Phan, and W. Lin. (2018). *Enriched Long-Term Recurrent Convolutional Network for Facial Micro-Expression Recognition*. [Online]. Available: https://github.com/IcedDoggie/Micro-Expression-with-Deep-Learning
[40] S. Liong, Y. S. Gan, J. See, and H. Khor. (2019). *A Shallow Triple Stream Three-Dimensional CNN (STSTNet) for Micro-Expression Recognition System*. [Online]. Available: https://github.com/christy1206/STSTNet
[41] S. T. Liong, Y. S. Gan, J. See, and H. Q. Khor, "A shallow triple stream three-dimensional CNN (STSTNet) for micro-expression recognition system," 2019, *arXiv:1902.03634*. [Online]. Available: http://arxiv.org/abs/1902.03634

[42] Z. Xia, X. Hong, X. Gao, X. Feng, and G. Zhao, "Spatiotemporal recurrent convolutional networks for recognizing spontaneous micro-expressions," 2019, *arXiv:1901.04656*. [Online]. Available: http://arxiv.org/abs/1901.04656
[43] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–10.
[44] G. Zeng, J. Zhou, X. Jia, W. Xie, and L. Shen, "Hand-crafted feature guided deep learning for facial expression recognition," in *Proc. IEEE Conf. Autom. Face Gesture Recognit. (FG)*, May 2018, pp. 423–430.
[45] M. V. O. Arriaga and P. G. Plager. *Implement of Exception*. Accessed: 2017. [Online]. Available: https://github.com/oarriaga/face_classification
[46] Z. Ming, J. Chazalon, M. M. Luqman, M. Visani, and J.-C. Burie, "FaceLiveNet: End-to-end networks combining face verification with interactive facial expression-based liveness detection," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3507–3512.
[47] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
[48] Z. Lian, Y. Li, J. Tao, J. Huang, and M. Niu, "Region based robust facial expression analysis," in *Proc. Asian Conf. Affect. Comput. Intell. Interact. (ACII Asia)*, May 2018, pp. 1–5.
[49] M. Li, H. Xu, X. Huang, Z. Song, X. Liu, and X. Li, "Facial expression recognition with identity and emotion joint learning," *IEEE Trans. Affective Comput.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8528894
[50] J. Li, Y. Wang, J. See, and W. Liu, "Micro-Expression recognition based on 3D flow convolutional neural network," *Pattern Anal. Appl.*, pp. 1–9, Nov. 2018. [Online]. Available: https://link.springer.com/article/10.1007/s10044-018-0757-5. doi: 10.1007/s10044-018-0757-5.
[51] F. Xu, J. Zhang, and J. Z. Wang, "Micro-expression identification and categorization using a facial dynamics map," *IEEE Trans. Affective Comput.*, vol. 8, no. 2, pp. 254–267, Apr./Jun. 2017.
[52] X. Huang, G. Zhao, X. Hong, W. Zheng, and M. Pietikäinen, "Spontaneous facial micro-expression analysis using spatiotemporal completed local quantized patterns," *Neurocomputing*, vol. 175, pp. 564–578, Jan. 2016.
[53] X. Huang, S. Wang, X. Liu, G. Zhao, X. Feng, and M. Pietikäinen, "Spontaneous facial micro-expression recognition using discriminative spatiotemporal local binary pattern with an improved integral projection," 2016, *arXiv:1608.02255*. [Online]. Available: https://arxiv.org/abs/1608.02255
[54] S. L. Happy and A. Routray, "Fuzzy histogram of optical flow orientations for micro-expression recognition," *IEEE Trans. Affective Comput.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/7971947
[55] D. H. Kim, W. J. Baddar, J. Jang, and Y. M. Ro, "Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition," *IEEE Trans. Affective Comput.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/7904596
[56] Y. Zong, X. Huang, W. Zheng, Z. Cui, and G. Zhao, "Learning from hierarchical spatiotemporal descriptors for micro-expression recognition," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3160–3172, Nov. 2018.
[57] C. Hu, D. Jiang, H. Zou, X. Zuo, and Y. Shu, "Multi-task micro-expression recognition combining deep and handcrafted features," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 946–951.
[58] M. Peng, Z. Wu, Z. Zhang, and T. Chen, "From macro to micro expression recognition: Deep learning on small datasets using transfer learning," in *Proc. IEEE Int. Conf. Autom. Face Gesture (FG)*, May 2018, pp. 657–661.

SI MIAO received the B.Sc. degree from the Computer Science Department, Huazhong University of Science and Technology (HUST), in 2018. He is currently pursuing the master's degree from the Shanghai Advanced Research Institute, Chinese Academy of Sciences.

HAOYU XU received the B.S. degree in electronic engineering and information system from the University of Science and Technology of China (USTC), the M.Phil. degree in information engineering from the Chinese University of Hong Kong (CUHK), and the Ph.D. degree in communication and information system from the University of Chinese Academy of Sciences.

He has diversified skills across academia, industry and business sectors. He has published around 20 research papers and authorized seven invention patents. He currently serves as the Director for AR/VR Solution Center in the Lenovo Research, Shanghai. He was an Associate Professor with the Shanghai Advanced Research Institute, Chinese Academy of Sciences. His current research interests include computer vision and natural language processing as well as their applications in real scenarios.

ZHENQI HAN received the master's degree from the Computer Science Department, Shanghai University. He is currently a Research Assistant with the Shanghai Advanced Research Institute, Chinese Academy of Sciences. His research interests include computer vision and 3D deep learning, and he has published more than 10 journals or patents.

YONGXIN ZHU received the Ph.D. degree in computer science from the National University of Singapore, Singapore, in 2001.

He was with the National University of Singapore as a Research Fellow, from 2002 to 2005, and with the School of Microelectronics, Shanghai Jiao Tong University as an Associate Professor, from 2006 to 2017. In 2017, he joined the Shanghai Advanced Research Institute, Chinese Academy of Sciences, as a Full Professor. He is also an Adjunct Professor with Shanghai Jiao Tong University and the University of Chinese Academy of Sciences. He has authored more than 130 English papers and 50 Chinese papers. His research interests include computer architectures, system-level IC design, and big data processing.

Dr. Zhu is a Senior Member of the China Computer Federation. He has served more than 30 conferences and journals as an editor, the program chair, the publicity chair, and a TPC member.