Pattern Analysis and Applications (2019) 22:1331–1339
https://doi.org/10.1007/s10044-018-0757-5

ORIGINAL ARTICLE

Micro-expression recognition based on 3D flow convolutional neural network

Jing Li[1] · **Yandan Wang**[1] · **John See**[2] · **Wenbin Liu**[1]

Received: 5 June 2017 / Accepted: 29 October 2018 / Published online: 8 November 2018

Abstract
Micro-expression recognition (MER) is a growing field of research which is currently in its early stage of development. Unlike conventional macro-expressions, micro-expressions occur at a very short duration and are elicited in a spontaneous manner from emotional stimuli. While existing methods for solving MER are largely non-deep-learning-based methods, deep convolutional neural network (CNN) has shown to work very well on such as face recognition, facial expression recognition, and action recognition. In this article, we propose applying the 3D flow-based CNNs model for video-based micro-expression recognition, which extracts deeply learned features that are able to characterize fine motion flow arising from minute facial movements. Results from comprehensive experiments on three benchmark datasets—SMIC, CASME/CASME II, showed a marked improvement over state-of-the-art methods, hence proving the effectiveness of our fairly easy CNN model as the deep learning benchmark for facial MER.

Keywords Facial micro-expressions · Micro-expression recognition · 3D CNN · Optical flow · CASME · SMIC

1 Introduction

In pattern recognition research, deep learning, from the early efforts of Hinton [9], has penetrated various domains such as machine vision, speech recognition and natural language processing due to its superior self-learning ability. Very recently, these techniques have found their ways to the task of recognizing facial expressions, achieving a good measure of success [2, 13, 15]. In our work, we introduce deep learning to facial *micro-expression* recognition (MER) with the efficient CNN as the benchmark due to its simplicity and effectiveness in most applications.

Micro-expressions are different from normal facial expressions (or macro-expressions in this case), in terms of its brief duration and spontaneous occurrence, a form of involuntary response toward an emotional stimuli. It was first discovered by Haggard and Isaacs [7] as a kind of 'micro-momentary' expression, or a nonverbal communication during psychotherapy, reported again by [3] 3 years later. Based on Ekman's account, they examined a taped video, a talk between a patient and a psychologist. Despite the patient portraying a happiness expression throughout the talk, they found that the patient showed a quick subtle and abnormal expression in her face (this was done by examining the video frame by frame with naked eyes). The patient also confessed her tendency for suicide under the psychologist's cross-examination.

Micro-expressions also occur in high-stakes situations, where people have something to lose or gain. The importance of micro-expression study is useful in many areas, e.g., clinical psychology, criminal interrogation, public security. In clinical psychology, it can help the psychologist diagnose

This work is funded in part by the Zhejiang Provincial Natural Science Foundation of China (Grants Nos. LQ17F020002, R1110261) and the National Science Foundation of China (Grants Nos. 61572367, 61272018)

✉ Yandan Wang
yandan.wang@wzu.edu.cn

Jing Li
ywhzjjer@163.com

John See
johnsee@mmu.edu.my

Wenbin Liu
wbliu6910@gmail.com

[1] Faculty of Computer Science and Technology, Wenzhou University, Wenzhou 325035, Zhejiang, China

[2] Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia

whether the patient is concealing the truth; in criminal interrogation, it can help the policemen decide whether the suspects are lying; for public security, it can help identify potential dangerous persons at public areas such as airport and railway station. Therefore, to develop an automatic micro-expression detection and recognition system is much desired.

CNN is one of the most popular deep learning methods and widely used especially on image classification [8, 16, 22, 23, 30]. Furthermore in [2], the authors presented CNN for facial expression recognition with a fairly small dataset without overfit issues, and in [14], the authors presented CNN for human action recognition with gray images, optical flow field and gradient as input data. Based on those superior works of the state of the art, we will give a maiden attempt of introducing a 3D flow-based spatiotemporal CNN into our work and some popular CNNs improvement techniques will also be introduced to boost the results. We believe that more and more deep learning work will be done in micro-expression recognition. In non-deep learning-based work, LBP-TOP was applied as the benchmark, while we wish our simple CNN model will be the benchmark for the deep learning work in micro-expression recognition.

The rest of the article is organized as follows: First, we will give a brief review of micro-expression recognition and convolutional neural network in related work. Next, we will detail our methods. Experiments will be reported in next section. Finally, we will conclude and discuss our work.

2 Related work

Since micro-expression recognition is still in its early stage, there are not many published works every year. Therefore, we present a concise review of related works on micro-expression recognition, and convolutional neural networks related to our model.

2.1 Facial micro-expression recognition

Facial micro-expression recognition has been a challenging research topic since 2009. Although it has been a decade, there are still not many related works. In the early days, the lack of spontaneous facial micro-expression data is a stumbling block. However, with the joint effort of psychologists and computer scientists, three datasets have been created recently, all elicited in a spontaneous way: SMIC [19] by University of Oulu, and CASME [29]/CASME II [28] by Chinese Academy of Sciences.

The most comprehensive and largest dataset, CASME II, established the baseline method using Local Binary Pattern-Three Orthogonal Planes (LBP-TOP) for spatiotemporal feature extraction and support vector machine (SVM) for classification. Later works followed the use of LBP-TOP but using different base features based on optical strain [20] and monogenic signals [21]. These works enhance the ability of extracting LBP-TOP features by processing more meaningful data. In the work of [25, 26], the authors reconstructed two spatiotemporal methods, LBP-SIP and LBP-MOP, both derived from the concept of LBP-TOP. However, LBP-SIP is not robust enough when the data are processed by other methods such as noise filtering or frame interpolation. LBP-MOP is frame interpolation independent and thus only works well when a suitable smoothing filter is applied to reduce the influence of noise. Wang et al. [24] introduced video motion signal magnification to preprocess video signals and used LBP-TOP for feature extraction that get the results significantly enhanced. However, the method is not robust as the noise could also be amplified when the subtle facial expression get magnified. Moreover, the Leave-One-Video-Out (LOVO) cross-validation method lead by the baseline has been discarded and replaced by the Leave-One-Subject-Out (LOSO) cross-validation method, due to the subject dependency policy of the LOVO caused the unfair and high recognition results. Goswami et al. [6] proposed another variant called Local Ordinal Contrast Pattern-Three Orthogonal Plane (LOCP-TOP) which was found to be slightly better than the rest [11].

Optical flow can play a significant role in computing how expression changes in face across time. Liong et al. [20] improved the recognition performance by assigning weights computed from the optical strain (a derivative of the optical flow field) to each local frame block. A more recent work by [27] also used principal optical flow directions to construct facial dynamic maps. In short, these previous works manually construct and design the descriptors and representations based on researchers' analysis, which may not learn the minute spatiotemporal changes in micro-expressions in the most natural manner.

2.2 Brief convolutional neural network frameworks

After a decade since Hinton's introductory idea of learning deeper neural networks [9], the deep convolutional neural network (CNN), one of the more popular architectures, has gained significant interest in fields such as image classification, object detection and speech recognition, after its first successful application to handwritten digits recognition [18]. Since there are tones of CNN works, we only mention works that is related to our model only. Our model starts with AlexNet 7 layers, then improved by techniques in other works, and derived our model with 12 layers based on our intensive experiments.

Krizhevky et al. [16] proposed AlexNet by using CNN and produced the best result in the ImageNet competition (ILSVRC) in 2012. The AlexNet has 7 layers, which is

deeper than the LeNet-5 [18]. In their paper, the authors introduced ReLU and dropout techniques. ReLU is used to accelerate convergence and solve the problem of vanishing gradients, while dropout is used to reduce the overfitting problem. Since then, the advancement of GPUs has rapidly propelled the wide usage of deep CNN. In 2014, the top two entries to the ILSVRC went to GoogLeNet [23] and VggNet [22], respectively. Both models exploited the effectiveness of increasing depth in a CNN architecture with 'deeper' models. The 19-layer VggNet used smaller convolution kernels than its predecessors [16, 30] to learn more detailed features. GoogLeNet presented a wide cascaded topology called inception that uses 1×1 convolution kernels to resolve the concentration of neurons in a single region. Recently, the far-reaching advantages of CNN have seen its penetration into facial expression recognition [2, 13, 15]. Byeon et al. [2] proposed a five-layer CNN, extending convolution operations to 3D convolutions for video sequences. Subsequent works by [15] and [13] proposed models that jointly learn from two types of data, i.e., shape/geometry and appearance; the former was used for expression recognition, while the latter was used to identify the occurrence of facial action units (AUs).

In this article, we proposed an application of 3D flow-based CNN (3D-FCNN) model, which exploits the advantages of state-of-the art work as the attempt at learning deeper models for micro-expression recognition. Motivated by the VggNet [22], we used a small 3D convolution kernel of $3 \times 3 \times 3$ to better represent the minute spatial changes at each local region. In contrast to two recent 3D CNN models proposed by [2] and [14], we introduce optical flow (dynamic information) together with standard grayscale frames (appearance information) as input data in a 12-layer deep network (5 pairs of convolution and pooling layers, a fully connected layer and an output softmax layer). Our final model is derived based on our extensive experiments. To avoid overfitting our model in a deep network, we utilize batch normalization (BN) and dropout techniques in our model. Furthermore, we applied rectified linear unit (ReLU) activation functions [5] to tackle the problem of vanishing gradients and for faster training, while zero-padding (ZP) is introduced to avoid information loss caused by the convolution operation.

3 Methods

In this section, we will present our proposed 3D flow-based CNN architecture in greater detail followed by network improvement techniques we applied that we utilize in our work.

3.1 Proposed 3D flow-based CNN architecture

The architecture of our 3D flow-based CNN is illustrated in Fig. 1 with the number of feature cubes and feature maps. We design a 12-layer architecture, which consists of inputs from three 'data streams,' i.e., grayscale frame sequence or

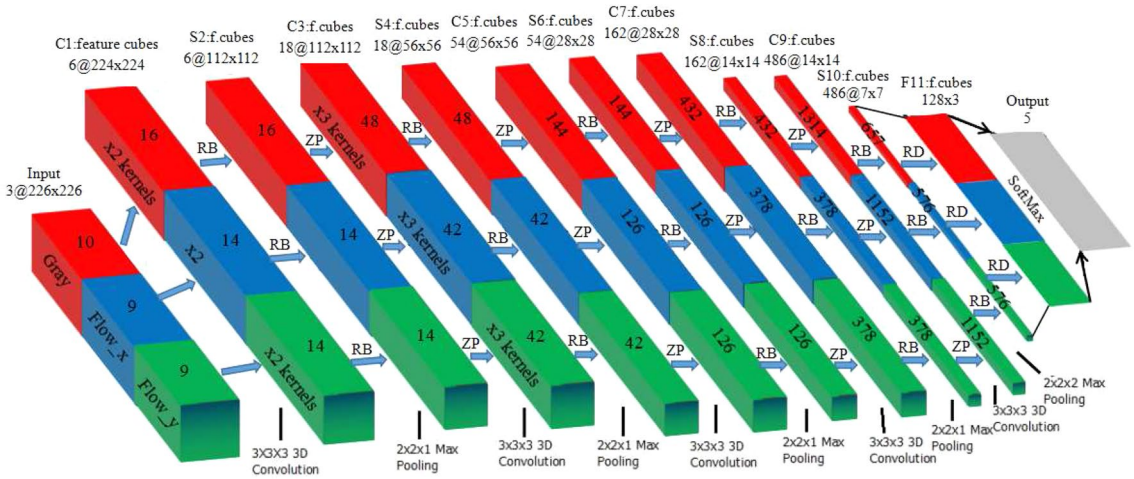

Fig. 1 Architecture of the proposed 3D flow-based CNN. It consists of 3 data stream sub-networks that each sub-network has input data with one tube channel only: grayscale frame sequences, 'gray' (red cube), the vertical and horizontal optical flow, 'flow_x' and 'flow_y' (blue and green cubes). The number of feature cubes is computed and is shown on the top notes with k@m × n, where k is the number of feature cubes, $m \times n$ is the size of feature map. C is the convolution layer, and S is the pooling layer. The number of convolutional kernels used is shown in the left face of the cubes, where 2 kernels are used in layer C1, while 3 kernels are used in other layers. The number of feature maps is shown on the top of its corresponding cube (this example is based on TIM10; see Sect. 4.2). RB refers to ReLu + Batch normalization, $ZP_{1\times1\times1}$ for 3D zero-padding with size $1 \times 1 \times 1$ (except C9 which uses $ZP_{1\times1\times2}$), and RD is ReLU + Dropout (color figure online)

'gray' (G) (shown as red cube), vertical component of optical flow or 'x_flow' (XF) sequence (shown as blue cubic) and the horizontal component of optical flow or 'y_flow' (YF) sequence (shown as green cube). There are five 3D convolutional layers labeled as C*i*, five 3D max-pooling layers labeled as S*i*, one fully connected (FC) layer labeled as F*i*, and the final output softmax layer.

In the 3D convolutional layers, a 3D convolution kernel of size m_x x m_y x m_t is introduced to extract spatiotemporal feature information. For instance, a $3 \times 3 \times 3$ convolutional kernel sets the temporal dimension as 3, which means it handles adjoining three frames with spatial window sized 3×3. Figure 2 shows an example how the 3D convolution works in a data cube with the video of 6 frames. The labeled 3D convolution kernel with $3 \times 3 \times 3$ is performing the convolution operations on every three adjoining images (either blue or red, sliding from first three images to last three images) with a spatial window of size 3×3 sliding from left to right and top to bottom. Let W be the convolution kernel matrix, I_i as the ith image and f_i the ith convolution output. Since convolving every three I_i yields a single output f, the 3D output size is $[h - m_x + 1, w - m_y + 1, l - m_t + 1]$, where h, w and l are the height, width and length of the I sequence, respectively. Based on the example in Fig. 2, six I video frames yield four f output maps after the 3D convolution operation:

$$f_i = I_i * W_1 + I_{i+1} * W_2 + I_{i+2} * W_3 \quad (1)$$

In order to extract features with a finer coverage, we utilize more than one 3D convolution kernels. Formally, the pixel at (x, y, z) on the jth feature map of the ith layer is given as:

$$v_{ij}^{xyz} = relu\left\{ b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} \right\} \quad (2)$$

where w is the weight of the 3D convolution kernel, v on the RHS of equation is the value of previous layer and v on the LHS of the equation is the value after convolution and ReLU. All the computations in Eq. 2 are in three dimensions, while the ReLU function adds the nonlinearity with better acceleration toward convergence.

3D max-pooling layer is applied to 3D spatiotemporal data in the same manner as 3D convolution by considering all three axes, X, Y and T. The pooling process summarizes the outputs of neighboring groups of neurons in the same feature map to reduce its dimensionality for the next layer. However, because of the short length of T dimension (in relative to the network depth), we only apply 3D spatiotemporal max-pooling in the last subsampling layer, while standard 2D spatial max-pooling is applied to all other subsampling layers. By experiments, we found that max-pooling

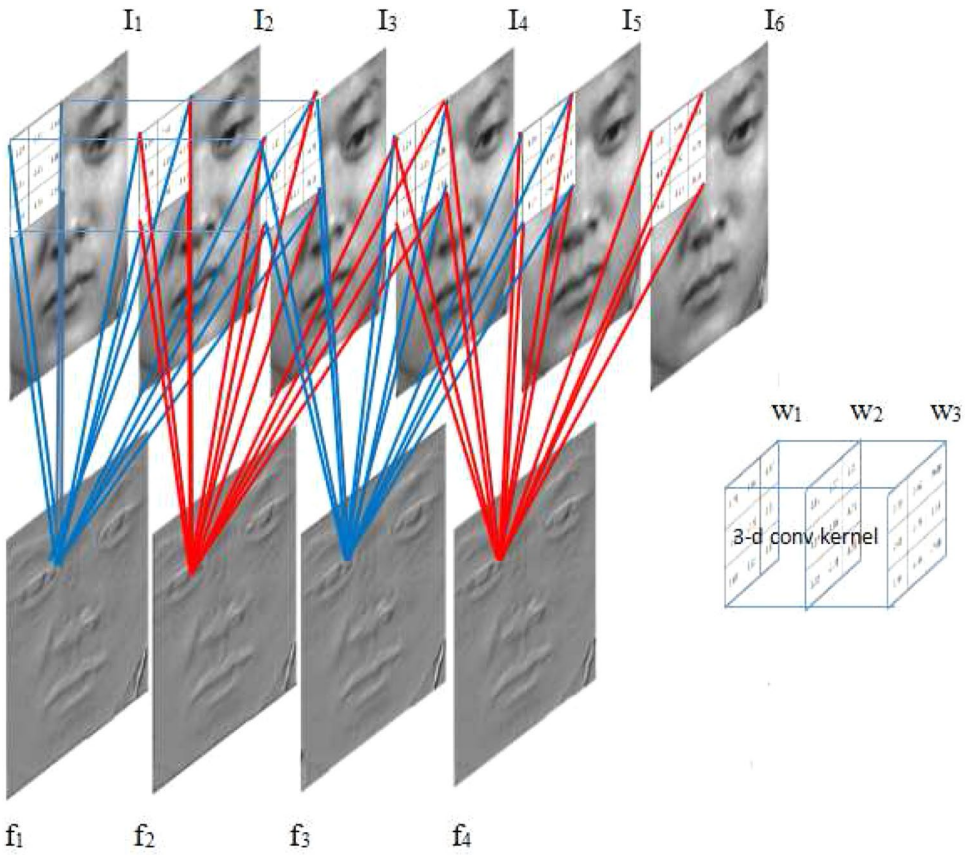

Fig. 2 The illustration of 3D convolutional operation

performs better than average-pooling on the task of micro-expression recognition, as similarly concluded in other problem domains [1, 16]. Hence, we opt for max-pooling in our architecture.

The ReLU and BN techniques are applied to each convolutional layer before going to the following pooling layer, while zero-padding is applied to each pooling layer (except S10) before going to its subsequent convolutional layer. The size of all convolutional kernels is set to $3 \times 3 \times 3$. The pooling size is set to $2 \times 2 \times 1$ except the pooling in layer S10 ($2 \times 2 \times 2$). The output from S10 for all three data cube streams is flattened and concatenated, reducing the final dimensionality to 128×3.

The output layer takes the size corresponding to the number of classes, which is derived by the softmax classifier,

$$h_\theta(x^{(i)}) = \frac{1}{\sum_{j=0}^{k-1} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_0^T x^{(i)}} \\ e^{\theta_1^T x^{(i)}} \\ \vdots \\ e^{\theta_{k-1}^T x^{(i)}} \end{bmatrix} \tag{3}$$

where θ_j are the weights and $k = \{5, 3\}$ represents the number of classes on the CASME/CASME II and SMIC datasets, respectively. The predicted class for input $x^{(i)}$ is the one with the largest probability $h_\theta(x^{(i)})$.

3.2 Network improvements

We apply zero-padding to the inputs of all pooling layers (except S10 which is connected to a FC). Zeros are padded at the edge of input volumes to control the desired spatial size of the output volumes. This prevents the convolutional operation from reducing the dimensionality of feature maps as what VggNet does, since we use pooling to reduce the dimensionality. The size of $1 \times 1 \times 1$ zero-padding is applied, except layer S8 uses size of $1 \times 1 \times 2$. This way, we can increase the depth of our architecture to learn more specific subtle spatial changes that occur in micro-expressions.

Batch normalization (BN) [12] is popularly used for CNN training as it helps accelerate the rate of convergence even with small learning rates. We introduce BN to the convolutional layers of our 3D CNN, after the ReLU activations. This is a characteristic not found yet in other existing 3D CNN models [14]. In BN, to normalize input (in a layer) $\mathbf{x} = \{x^{(1)}, \ldots, x^{(d)}\}$, its kth dimension can be normalized as Eq. 4:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}]}} \tag{4}$$

where E and Var represent the mean and variance of the batch. If we normalize inputs to a nonlinear activation function (e.g., sigmoid), then the output would only be bounded to the linear part. Thus, we can scale and shift the normalized input,

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)} \tag{5}$$

where $\gamma^{(k)}$ and $\beta^{(k)}$ are the scaling and shifting parameters, respectively, that need to be learnt. Following [23], we use mean and variance of the batch for normalization during training, while the mean and variance of the dataset are used for testing.

Dropout fine-tuning is a regularization technique that helps reduce the overfitting problem in neural networks by preventing complex coadaptations on training data [10] caused by fully connected layers. Given a dropout rate p, which is set to 0.5 in our NNs, 50% units will be retained, while another 50% will be omitted and re-insert into the networks with initialized weights. Simply, 'dropout' just randomly ignores some neurons. However in testing stage, the output of each neuron will be weighted by the factor $1 - p$ (the retain rate) to keep it the same effect as dropout made in training stage.

4 Experiments

In this section, we briefly describe the datasets used and present the experimental results with analysis and discussion.

4.1 Datasets

SMIC [19] consists of micro- and macro-expressions, and we select only the micro-expression samples. There are a total of 164 videos from 16 subjects, capturing spontaneous micro-expressions from 3 classes (positive, negative and surprise). The samples were recorded with a 100 *fps* high-speed camera.

CASME [29] was the first dataset built by Chinese Academy of Sciences (CAS). It consists of 180 videos from 19 subjects, and it was recorded at 60 fps. A total of 35 participants (13 females, 22 males) were recruited with an average age of 22.03 years ($\sigma = 1.60$) in their study. It consists of 8 identified micro-expressions, i.e., amusement, sadness, disgust, surprise, contempt, fear, repression and tense. The top five classes (tense, disgust, happiness, surprise and repression) that had the most samples were selected for experiments (the rest have too few samples for proper evaluation)

CASME II [28] was also built by CAS and is publicly available for research. It is composed of 247 videos from 26 subjects with 9 classes (happiness, surprise, disgust, fear, sadness, anger, repression, tense and negative). The top five classes (tense, disgust, happiness, surprise and repression) that had the most samples were selected for experiments, while the other four were discarded due to lack of samples.

Fig. 3 The illustration of a video frames in happy micro-expression

The dataset was recorded using a 200-fps high-speed camera. Figure 3 presents frames of a happiness micro-expression video clip. Unlike the macro-expression that the facial movement is obvious and observable, the muscle movements of micro-expression are subtle. During the entire video, we cannot see obvious changes by our naked eyes, which brings big challenges analyzing the discriminant features manually, hence enhancing even a minor recognition accuracy is a difficult task.

4.2 Parameter settings

In order to keep evaluation comparisons fair, our implementations follow the same data preprocess and evaluation protocol used in previous works: (1) Frames are resized into 226×226. (2) The length of sequences (number of frames) are interpolated to 10 and 15 using temporal interpolation model (TIM) [32]. (3) Leave-One-Subject-Out (LOSO) cross-validation [17] is used in our evaluation to prevent subject bias during learning.

In our proposed 3D-FCNN approach, input using only grayscale sequences is denoted as 'G,' while the same input with 5×5 block partitions is denoted as 'G-5x5.' In a similar fashion to block partitioning used in LBP-TOP [31], each frame is partitioned into 5×5 blocks. In our architecture, all blocks are concatenated in the input layer before passing through the network. With additional optical flow information, we also tested with all three data streams—grayscale frame sequences, and the optical flow sequences in the x- and y-directions (input sequence is less by one), denoted by 'G,XF,YF.' Similarly, the 'G-5x5,XF,YF' has the same inputs as 'G,XF,YF' except that each grayscale frame is partitioned into 5×5 blocks. All these different input configurations used the same 3D-FCNN architecture shown in Fig. 1, except for the input layer.

The model is composed of five 3D convolutional layers with kernel size $3 \times 3 \times 3$. There are four max-pooling layers of size $2 \times 2 \times 1$ while the final max-pooling layer is of size $2 \times 2 \times 2$. Henceforth, the zero-padding size is set to $1 \times 1 \times 1$, except the last padding of $1 \times 1 \times 2$.

The final extracted features are flattened and concatenated for classification by the last softmax layer based on LOSO cross-validation. Learning rate is set to 0.01, decay is set to 5×10^{-4}, and weights are randomly initialized as proposed in [4]. As comparison with traditional descriptors, we implement the LBP-TOP, LOCP-TOP and LBP-SIP methods on the three datasets under same data preprocess and evaluation protocols, which frames are also interpolated to length of 10 and 15, respectively, and partitioned into 5×5 blocks for feature extraction before by using SVM with LOSO cross-validation.

4.3 Experiment results

Table 1 shows the results of the proposed 3D-FCNN method on the SMIC, CASME and CASME II datasets, after interpolating to 10 and 15 frames using temporal interpolation model (TIM) [32]. TIM was used in the baseline of SMIC [19] and adopted as well by many later works [17, 20].

Table 1 The experiment results of the proposed 3D-FCNN method with various input data

Methods	Accuracy (%)	
SMIC	TIM10	TIM15
3D-FCNN (G)	36.59	42.07
3D-FCNN (G-5 × 5)	38.41	39.02
3D-FCNN (G, XF, YF)	47.56	26.83
3D-FCNN (G-5 × 5,XF,YF)	**55.49**	**55.49**
CASME		
3D-FCNN (G)	31.67	33.89
3D-FCNN (G-5 × 5)	36.11	27.22
3D-FCNN (G, XF, YF)	48.89	42.22
3D-FCNN (G-5x5, XF, YF)	**54.44**	**43.89**
CASME II		
3D-FCNN (G)	38.06	37.65
3D-FCNN (G-5 × 5)	40.10	41.30
3D-FCNN (G, XF, YF)	**59.11**	**52.23**
3D-FCNN (G-5 × 5, XF, YF)	55.06	50.61

Overall, we find that interpolating to 10 frames (TIM10) is superior to that of 15 frames (TIM15), while flow-based approach performs much better than using grayscale frames only, which demonstrates the importance of learning dynamic information in a CNN. Meanwhile, by further block partitioning the grayscale frames with the flow information, we observe a generally better performance except on the CASME II. We hypothesize that this might be caused by too little changes at local block level in the case of high-speed capture (200 fps). Nevertheless, the results show the effectiveness of our proposed 3D-FCNN model for MER compared to that using standard image appearance information.

We take one video from sub01 of CASME II dataset and visualize the feature maps of each convolutional layer in Fig. 4, presenting 3 sample feature maps from each data cube channel. As the depth increases (C1 to C5), we can clearly see from the 'gray' channel that the feature map starts losing the irrelative visual information. Gradually, the network begins to learn specific subtle spatial changes, focusing on vital regions of interests, i.e., eyes and mouth regions (see C5). Likewise, the 'x-flow' and 'y-flow' channels also depict more details in the deeper layers, preserving finer facial details.

The best flow-based results on the SMIC, CASME and CASME II datasets (from Table 1) are chosen for further comparisons with other state-of-the-art methods in Table 2. For fair benchmarking, we re-implemented these methods applying the same preprocess: Length of the video sequence is interpolated to 10 frames and spatially resized to 226x226 pixels. From the table, our proposed 3D-FCNN model outperforms the other state-of-the-art methods: LBP-TOP, LOCP-TOP and LBP-SIP on all evaluated datasets (SMIC, CASME, CASME II), which are 6.1%, 5.55% and 2.43% better compared to LBP-TOP on three datasets, respectively, 9.76%, 4.44%, 2.43% better compared to LOCP-TOP, respectively, 12.81%, 8.33%, 5.26% better than that of LBP-SIP, respectively. Overall, we can see that the improvements on CASME II are not obvious but still promising, while the improvements on other two datasets are significant and pleasant. LBP-SIP performed slightly worse than

Table 2 The comparison between the proposed method and recent state-of-the-art methods

Method/accuracy (%)	SMIC	CASME	CASME II
LBP-TOP* [19, 28]	49.39	48.89	56.68
LOCP-TOP* [6]	45.73	50.00	56.68
LBP-SIP* [26]	42.68	46.11	53.85
i2D [21]	43.29	N/A	45.53
OSW [20]	53.05	N/A	41.70
FDM [27]	54.88	42.02	41.96
Proposed method*	55.49	54.44	59.11

*These methods are based on our implementation with TIM10

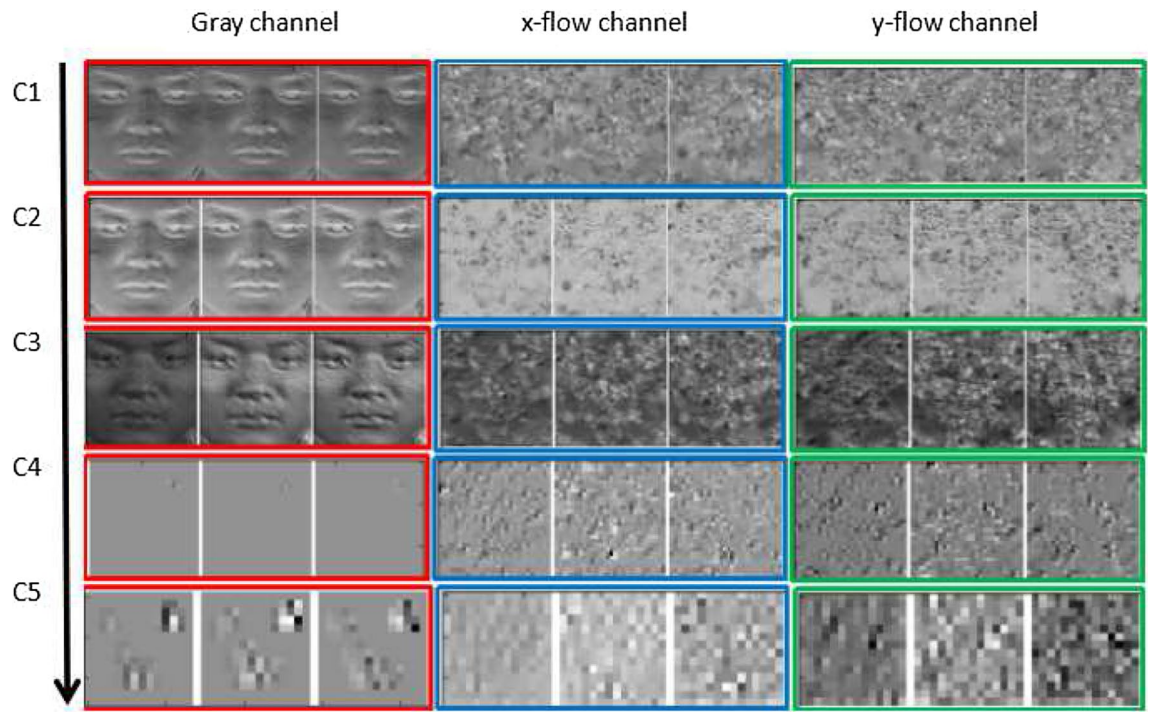

Fig. 4 The feature map examples produced at each convolutional layer of sub-network stream with corresponding input cube channel respectively

LOCP-TOP and LBP-TOP possibly due to the effects of frame interpolation, while the performance of LBP-TOP and LOCP-TOP is somewhat at a similar level. Micro-expression recognition is a fairly difficult task that the subtle changes in face cannot be observed and analyzed by our naked eyes, thus enhancing the recognition accuracy is an extremely difficult challenge. However, from Table 1 our proposed method performs best, and the results are still promising.

In comparison with other non-TIM-based methods (i2D, OSW and FDM), our method still performs exceedingly well, particularly in the CASME and CASME II datasets that are around 12% and 17% better, respectively. Despite the need for the interpolation operation (warranted in our case to reduce the complexity of the input layer and keep the data size consistency), our 3D-FCNN method still shows its ability to self-learn its own features without needing to specific features design or heuristic rules.

5 Conclusions and discussions

In this article, we demonstrated the effectiveness of our proposed 3D flow-based CNN (3D-FCNN) compared to the state-of-the-art traditional approaches for facial micro-expression recognition (MER). Our experiments cover an extensive evaluation on three existing benchmark MER datasets: SMIC, CASME, and CASME II. Although CNN naturally requires a longer training time, we show that it can learn minute spatiotemporal changes in a deep neural network, with the aid of essential dynamic information from optical flow. With the increased leveraging of GPUs, the complexity of training deep CNN is no longer a key issue. Our proposed 12-layer model yields promising results, but there are further problems to explore in future. Among many, we intend to investigate how deeply learned action units (AUs) can be jointly propagated in our architecture as a separate new data stream. Also, a more robust keyframe selection method may be able to choose better frames than the current temporal interpolation scheme.

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation for the donation of a Quadro K5200 GPU used in this work.

References

1. Boureau YL, Ponce J, LeCun Y (2010) A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th international conference on machine learning, pp 111–118
2. Byeon YH, Kwak KC (2014) Facial expression recognition using 3d convolutional neural network. Int J Adv Comput Sci Appl 5(12):107–112
3. Ekman P, Friesen WV (1969) Nonverbal leakage and clues to deception. Psychiatry 32(1):88–106
4. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. Aistats 9:249–256
5. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Aistats, vol 15, p 275
6. Goswami B, Chan C.H, Kittler J, Christmas B (2010) Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication. In: 2010 fourth IEEE international conference on biometrics: theory applications and systems (BTAS). IEEE, pp 1–6
7. Haggard E.A, Isaacs K.S (1966) Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy. In: Methods of research in psychotherapy. Springer, pp 154–165
8. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385
9. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
10. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580
11. Huang X, Wang SJ, Zhao G, Piteikainen M (2015) Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection. In: Proceedings of the IEEE international conference on computer vision workshops, pp. 1–9
12. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167
13. Jaiswal S, Valstar M (2016) Deep learning the dynamic appearance and shape of facial action units. In: IEEE winter conference on applications of computer vision (WACV), pp 1–8
14. Ji S, Xu W, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
15. Jung H, Lee S, Yim J, Park S, Kim J (2015) Joint fine-tuning in deep neural networks for facial expression recognition. In: IEEE international conference on computer vision (ICCV)
16. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
17. Le Ngo A.C, Phan R.C.W, See J (2014) Spontaneous subtle expression recognition: imbalanced databases and solutions. In: Asian conference on computer vision. Springer, pp 33–48
18. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
19. Li X, Pfister T, Huang X, Zhao G, Pietikainen M A (2013) spontaneous micro-expression database: Inducement, collection and baseline. In: 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG). IEEE, pp 1–6
20. Liong S.T, See J, Phan R.C.W, Le Ngo A.C, Oh Y.H, Wong K (2014) Subtle expression recognition using optical strain weighted features. In: Asian conference on computer vision workshops, pp 644–657
21. Oh YH, Le Ngo AC, Phari RCW, See J, Ling HC (2016) Intrinsic two-dimensional local structures for micro-expression recognition. In: IEEE international conference on acoustics, speech, and signal processing (ICASSP), pp 1851–1855
22. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
23. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
24. Wang Y, See J, Oh YH, Phan RCW, Rahulamathavan Y, Ling HC, Tan SW, Li X (2016) Effective recognition of facial

micro-expressions with video motion magnification. Multimed Tools Appl. https://doi.org/10.1007/s11042-016-4079-6
25. Wang Y, See J, Phan RCW, Oh YH (2015) Efficient spatio-temporal local binary patterns for spontaneous facial micro-expression recognition. PLoS ONE 10(5):e0124–674
26. Wang Y, See J, Phan RCW, Oh YH (2015) LBP with six intersection points: reducing redundant information in lbp-top for micro-expression recognition. In: Asian conference on computer vision, pp 525–537
27. Xu F, Zhang J, Wang J (2016) Microexpression identification and categorization using a facial dynamics map. IEEE Trans Affect Comput 8(2):254–267
28. Yan WJ, Li X, Wang SJ, Zhao G, Liu YJ, Chen YH, Fu X (2014) Casme II: an improved spontaneous micro-expression database and the baseline evaluation. PLoS ONE 9(1):e86041
29. Yan WJ, Wu Q, Liu YJ, Wang SJ, Fu X (2013) Casme database: a dataset of spontaneous micro-expressions collected from neutralized faces. In: 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG). IEEE, pp 1–7
30. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Computer vision–ECCV 2014. Springer, pp 818–833
31. Zhao G, Pietikainen M (2007) Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans Pattern Anal Mach Intell 29(6):915–928
32. Zhou Z, Zhao G, Guo Y, Pietikainen M (2012) An image-based visual speech animation system. IEEE Trans Circ Syst Video Technol 22(10):1420–1432