Pattern Recognition Letters 189 (2025) 166–174

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Entire-detail motion dual-branch network for micro-expression recognition

Bingyang Ma [a], Lu Wang [a,*], Qingfen Wang [a], Haoran Wang [a], Ruolin Li [a], Lisheng Xu [b], Yongchun Li [c], Hongchao Wei [c]

[a] School of Computer Science and Engineering, Northeastern University, Shenyang, 110169, China
[b] College of Information Science and Engineering, Northeastern University, Shenyang, 110819, Liaoning, China
[c] Shenyang Contain Electronic Technology Co., Ltd., Shenyang, 110167, Liaoning, China

ARTICLE INFO

Edited by: Prof. S. Sarkar

Keywords:
Micro-expression recognition
Motion features
Attention
Swin transformer

ABSTRACT

Micro-expression recognition is becoming an increasingly attractive research topic due to its useful applications in a widespread area including psychology, criminology, and security. Different from macro-expressions, the facial muscle movements of micro-expressions have the characteristics of being short duration, and weak intensity, which makes micro-expression recognition extremely challenging. To deal with these problems, we propose a dual-branch classification network that integrates entire and detail motions for effective micro-expression recognition. In this network, one branch is responsible for capturing the overall motion, while the other branch focuses on capturing the detail motion. In addition, to improve the recognition accuracy, we also design a Swin-Transformer module with accumulated attention to focus more on the Region of Interest. By utilizing Grad-CAM to obtain the facial expression activation heatmaps, we find a good match between the activated regions and facial action units. Finally, we validate the effectiveness of the method on the SMIC, CASME II, SAMM, and MMEW datasets, achieving recognition performance that are more competitive than many other state-of-the-art methods. Code is available at https://github.com/likemby/EDMDBN.

1. Introduction

Facial expressions can generally be categorized into two types: macro-expressions and micro-expressions. Macro-expressions are the most common and widely recognizable facial expressions in daily life. When a macro-expression occurs, individuals can typically observe and readily identify the corresponding emotion. This is attributed to the longer duration of macro-expressions (0.5 s to 2 s), the broader involvement of facial motion, and more intensive muscle movements. In contrast, micro-expressions are rapid, spontaneous, and low-intensity facial expressions that often appear when individuals attempt to conceal their true emotions, particularly in high-pressure, nervous, and anxious situations. Micro-expressions last for a short period of time, typically ranging from 1/25 to 1/3 s [1]. Additionally, micro-expressions have minimal coverage in the facial movement area, primarily focusing on the mouth, nose, and eyes, with subtle muscle movements. These characteristics of micro-expressions make them challenging to conceal, and thus convey genuine human emotions [2], lending them greater credibility than macro-expressions. Micro--expressions find applications in various fields such as national security, judicial trials, clinical medicine and public service, among others [3,4,5,6].

The characteristics of micro-expressions such as subtle facial muscle movements and short duration make accurate recognition of micro-expressions highly challenging. Many algorithms have been developed thus far to recognize micro-expressions. From the perspective of model feature extraction, these algorithms are mostly based on spatial features, temporal features, and spatiotemporal features of micro-expressions. For instance, Liu et al. [7] and Li et al. [8] focused on micro-expression recognition based on single-frame facial images. However, due to the small magnitude of micro-expression movements, it often requires motion enhancement techniques to achieve satisfactory recognition results [7,8]. Zhi et al. [9] and Hong et al. [10] employed the RGB image sequences as input. While this maximizes feature retention, redundant spatial information may introduce more background noise, making it difficult for the model to accurately identify and distinguish key features of micro-expressions during the learning process, thus deteriorating the model performance. Consequently, additional studies [11,12] based on sequential input aim to eliminate background information through techniques like optical flow and pixel difference,

* Corresponding author.
E-mail address: wanglu@cse.neu.edu.cn (L. Wang).

https://doi.org/10.1016/j.patrec.2025.01.021
Received 7 May 2024; Received in revised form 1 October 2024; Accepted 18 January 2025
Available online 19 January 2025

leading to improved performance. This observation indirectly underscores the importance of temporal features, namely motion features, in micro-expression analysis. Some motion-based works [13–15], while maintaining model conciseness, have demonstrated improved recognition results.

A complete micro-expression motion comprises the onset of facial movement, reaching its peak, and returning to a neutral state, which can be described by the onset, apex, and offset frames. Some studies [13–15], directly extracted motion features from the onset and apex frames of micro-expressions, which is proved to be straightforward and effective. However, this kind of approaches inevitably overlook details of the motion information spanning from the apex to the offset frame. Other approaches [9,10] extracted motion information from adjacent frames of the entire sequence, which can yield richer motion information. Nevertheless, these methods do not fully leverage the annotated apex frames and may entail redundant action representations. Based on the considerations, we propose a dual-branch micro-expression classification network that integrates entire and detail motions to address the aforementioned challenges.

Additionally, micro-expressions predominantly occur in localized facial regions, specifically the mouth, nose, and eye areas. Hence, there are typically two types of input for micro-expression recognition models: the region of interest (ROI) and the entire face. In previous research, Van et al. [16] demonstrated that selecting the ROIs as the model input helps mitigate interference from irrelevant areas. Conversely, the recent work by Li et al. [15] aimed to automate ROIs selection and allocated greater attention to the area of interest. In this study, to increase focus on distinct facial regions, we introduce an accumulated attention Swin Transformer module. Serving as a network backbone, this module directs the model to prioritize the region of interest.

The main contributions of this paper are outlined as follows:

A micro-expression recognition network with entire and detail motion extraction is proposed, which can improve the performance by capturing both the overall and detailed movement of the expression.

A Swin-Transformer module with accumulated attention is developed, which can effectively enhance the features extracted by the network.

Experimental results on SMIC, CASME II, SAMM, and MMEW datasets demonstrate the effectiveness of our proposed model in micro-expression recognition.

The rest of this paper is organized as follows: Section 2 briefly reviews the related work. Section 3 introduces the technical details of our proposed method. Section 4 firstly demonstrates the experimental results on four common datasets to justify the effectiveness of our proposed method on the MER task, and then presents the ablation studies and visual analysis for further discussion. Finally, we make a conclusion in Section 5.

2. Related work

2.1. Motion extraction

Many previous works have focused on studying the motion features for micro-expression recognition [9,10,13,14]. At first, optical flow features were manually extracted to represent micro-expression motions, which were then fed into CNNs for further feature extraction. However, due to the high computational complexity of optical flow, many recent studies opted to automatically learn micro-expression motion features using deep networks. Fan et al. [18] developed a self-supervised learning framework, in which motion features are learned through reconstructing apex frames from the onset frames after which Micro-expression classification was performed using the learned motion features. However, such approaches typically require substantial amounts of data to learn good feature representations and two training phases, thus increasing the overall training cost. In the study conducted by Li et al. [15], a motion extractor was designed for supervised micro-expression classification. This extractor computes the difference between two frames before further learning motion features through CNNs. In this paper, manual optical flow features are discarded in favor of representing motion through frame differences, which provide a simpler and computationally more efficient alternative.

2.2. Backbone network

In the field of micro-expression recognition, CNN [19], GCN [20] and Vision Transformer (ViT) [21] have been employed as backbone networks by previous studies. Among them, ViT, as a visual model based on self-attention mechanism, excels in processing global information and various image features. ViT divided the images into fixed-size patches, akin to word tokens in a language model like BERT [22]. This processing approach renders ViT somewhat more interpretable compared to CNNs, as it enables self-attention calculations among input tokens, facilitating a more intuitive understanding of the model's attention distribution across different image parts. However, ViT requires a large amount of data for model training. Swin-Transformer [23] addresses the issue by leveraging the pyramid structure and locality in CNN. In this paper, given the sequential nature of the micro-expression recognition data, we also employ the video version of Swin-Transformer [24] (Swin-T) as the backbone network for feature learning.

2.3. Localization of ROI areas

In the early work [16], Van et al. cropped the ROI area in the face as network input, which can enhance the micro-expression recognition performance by eliminating the interference from redundant facial information. Despite the requirement for manual labor, precise localization of the ROIs is necessary as it aids in more accurate micro-expression recognition. The trend of current study is on automating attention design to enable the model to allocate varying attention levels to distinct areas. Li et al. [15] utilized CNNs as the backbone network and incorporated spatial attention modules to integrate multiple stage-wise attentions. In contrast, He et al. [17] employed ViT as the backbone and aggregated self-attention feature maps from various stages to enhance attention effects. In this paper, we propose a Token Attention module with Swin-Transformer as the backbone to achieve accumulated attention, denoted as Accumulated Attention (AA).

2.4. Data scarcity

Micro-expression datasets are typically small, which can lead to challenges such as overfitting and insufficient generalization ability during model training. To address these challenges, it is essential to ensure that the complexity of the model matches the complexity of the data, specifically through two strategies: reducing model complexity and increasing data complexity. Alzubaidi at al. [25] suggested that methods such as transfer learning, self-supervised learning, and generative adversarial networks (GANs) [26] can be applied to address this issue. Transfer learning and self-supervised learning reduce the difficulty of micro-expression recognition tasks by designing pre-training tasks and reusing parameters from pre-trained models. Additionally, Huang at al. [27] combined the LBP-TOP concept with integral projection for micro-expression recognition, where integral projection reduces global feature dimensions, simplifying the model's feature representation. Moreover, GANs can generate new samples to expand the training dataset and increase data complexity, while online data augmentation can effectively enhance sample diversity, improving the model's generalization ability. These methods collectively offer strong support in tackling the challenges posed by small-scale datasets. In this paper, we increase the diversity of training samples through online data augmentation methods.

3. Methodology

As shown in Fig. 1, the algorithm consists of five stages: data preprocessing, extraction of entire and detail motions, motion feature extraction, feature fusion, and feature classification. Firstly, the original micro-expression sequence undergoes preprocessing. Next, entire and detail motions are extracted from the preprocessed sequence, and further motion features are extracted using the Swin-Transformer module with accumulated attention (Swin-T with AA). Subsequently, the extracted entire and detail motion features are fused, and finally, the fused features are fed into the classifier for micro-expression recognition.

3.1. Data preprocessing

Data preprocessing involves spatial and temporal normalization to ensure that the faces are aligned and the height, width, and number of frames in the input micro-expression sequence are uniform.

Spatial normalization of faces includes face alignment, cropping, and image scale normalization. First, facial keypoints are detected in the first frame of the micro-expression sequence. Then, the coordinates of the keypoints including the left outer eye, right outer eye, and nose tip are selected and matched with the corresponding keypoints in a standard template face to calculate the affine transformation matrix between them. After that, the affine transformation matrix is applied to every frame in the sequence to achieve face alignment. Then, based on the coordinates of keypoints on the chin, eyebrows, and eyes, the cropping boundary is determined and used to crop the face. Finally, the size of the cropped images is set uniformly to $H \times W$.

Due to the numbers of frames for different micro-expression video samples vary, we perform temporal normalization by linearly interpolating the three keyframes (onset frame, apex frame, and offset frame) of the micro-expression sample to $T+1$ frames. It needs to be noted that the SMIC dataset lacks the apex frame annotation. Therefore, we designate the middle frame of the sequence as the apex frame.

3.2. Extraction of entire and detail motions

After the preprocessing, a micro-expression sequence consists of $T+1$ frames, denoted as $\{F_1, F_2, F_3, \ldots, F_{T+1}\}$, where F_i represents the i-th frame in the sequence, and each frame tensor has a shape of $H \times W \times 3$. In addition, we denote the starting frame of the sequence as F_{onset} and the apex frame as F_{apex}.

In our approach, entire motion is defined as the pixel difference between the apex frame and the onset frame, represented as $M_{entire} = F_{apex} - F_{onset}$, with a tensor shape of $H \times W \times 3$; whereas detail motion is defined as the pixel difference between adjacent frames throughout the sequence, denoted as $M_{detail} = \{F_2 - F_1, F_3 - F_2, F_4 - F_3, \ldots, F_{T+1} - F_T\}$, with a tensor shape of $T \times H \times W \times 3$.

3.3. Motion feature extraction

To extract features from the entire and detail motions, we have developed a Swin-Transformer module with an accumulated attention mechanism, as is illustrated in Fig. 2.

Specifically, the module is utilized for the two branches respectively to extract entire and detail motion features. Taking the detail motion branch as an example, the extracted detail motion is first divided into multiple spatio-temporal blocks with each block having a size of $8 \times 4 \times 4 \times 3$. Since the size of the detail motion is $T \times H \times W \times 3$, we obtain $T/8 \times H/4 \times W/4$ blocks and each block (or Token) comprises a 384-dimensional feature vector. Subsequently, a linear embedding layer is applied to project the features of each Token to a C dimensional space. Due to the characteristics of Swin-Transformer, the backbone network executes spatial downsampling twice in the Patch Merging layer of each stage, reducing the number of Tokens to 1/4 of the original and doubling the dimensionality.

In addition, a Token Attention module that calculates the accumulating attention is integrated into every stage. The implementation details of the Token Attention module are illustrated in Fig. 3. In particular, it receives the output from the current Video Swin-Transformer block and performs weighting operations. Firstly, the current attention weight A_{cur} is calculated.

$$(A_{cur} = Sig(FC([Max(TK); Avg(TK)]))) \tag{1}$$

In Eq. (1), TK denotes the input Token, with a size of $N \times D$, where N represents the number of Tokens and D represents the dimensionality of each Token. Additionally, $Max(TK)$ represents its maximum value along the feature dimension, $Avg(TK)$ denotes its average value along the feature dimension, the symbol ";" represents a concatenate operation in the feature dimension, Sig represents the Sigmoid activation function, and FC represents the Fully Connected layer.

Then, the attention matrix of the last stage is combined to obtain the final attention matrix TK_{atten} of this stage.

$$(A_{final} = A_{cur} \odot DownSampling(A_{prev})) \tag{2}$$

where A_{prev} denotes the attention matrix from the previous stage. Since Swin-T conducts a $2\times$ spatial downsampling at each stage, we also need to perform a $2\times$ spatial downsampling on A_{prev}.

Finally, the input Token undergoes element-wise multiplication with the attention weight TK_{final} of this stage, resulting in the output Token TK'.

For the entire motion branch, its processing logic is similar to the module shown in Fig. 2, except that the temporal dimension is removed, as the input is an image.

3.4. Feature fusion

The feature dimensions of entire and detail motions differ, being $H \times W \times 3$ and $T \times H \times W \times 3$ respectively. As each spatial block and each spatiotemporal block has an initial size of $4 \times 4 \times 3$ and $8 \times 4 \times 4 \times 3$ respectively, after three stages of processing by the backbone network, the number of spatial blocks produced by entire motion is ultimately $N_{entire} = H/16 \times W/16$, while the number of spatiotemporal blocks produced by detail motion is $N_{detail} = T/8 \times H/16 \times W/16$. Here, N_{detail}

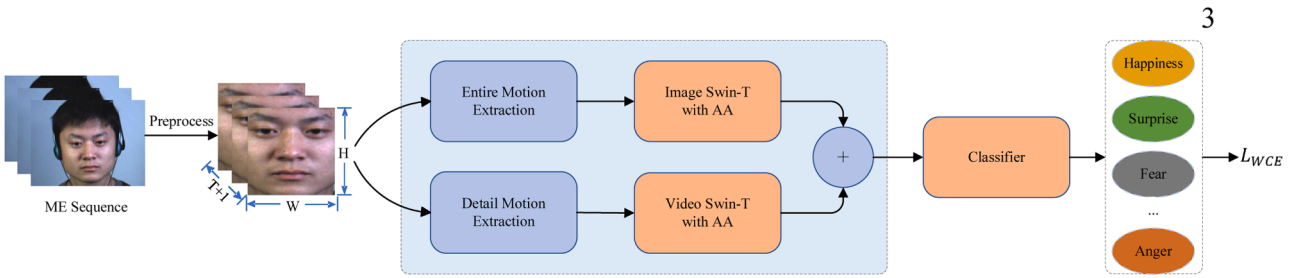

Fig. 1. Pipeline of the entire-detail motion dual-branch network (EDMDBN).

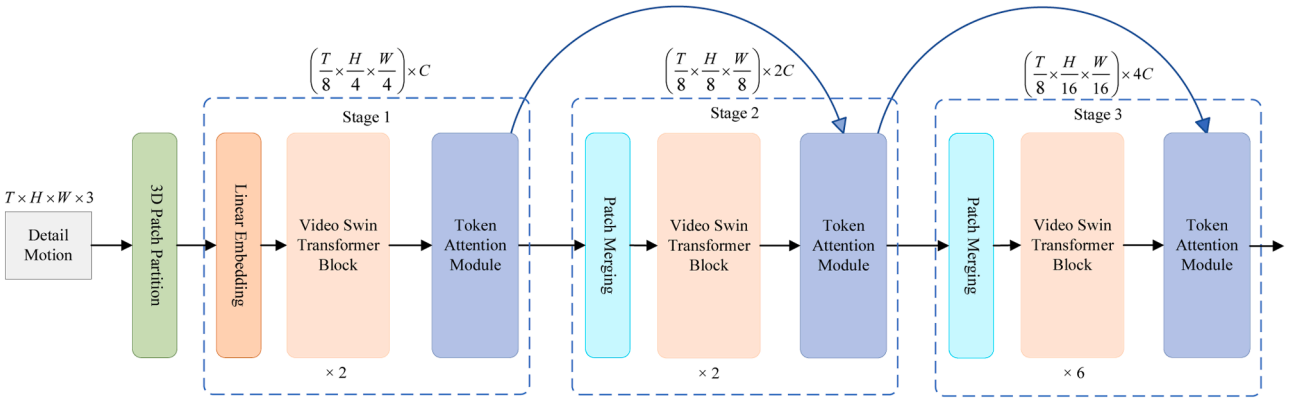

Fig. 2. The schematic diagram of the Swin-Transformer structure with accumulated attention (Taking the detail motion branch as an example, the formula above each stage in the figure consists of two parts, representing the number of Tokens at that stage and the number of channels).

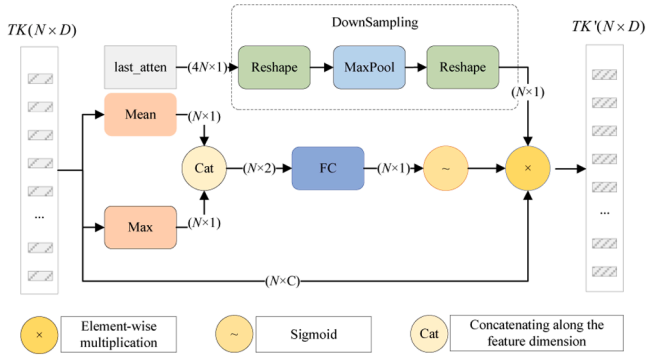

Fig. 3. Token Attention module.

is $T/8$ times N_{entire}, which may lead to the inability to directly perform feature fusion.

To address this issue, we perform average pooling on the spatio-temporal blocks in the time dimension for the same spatial location. This ensures that the number of spatiotemporal blocks is equal to the number of spatial blocks, as shown in Fig. 4.

3.5. Feature classification

The classifier consists of a fully connected layer, and the model is optimized through the cross-entropy loss function.

The input dimension of the fully connected layer is determined by $C_{in} = 4C \times N$, and the output dimension C_{out} is determined by the number of categories. For a classification problem with K categories, the formula for weighted cross-entropy loss is as follows:

$$L_{WCE}(\mathbf{y}, \mathbf{t}, \mathbf{w}) = -\Sigma_{i=1}^{K} w_i t_i \cdot \log\,(y_i) \tag{3}$$

Here, $\mathbf{y} = [y_1, y_2, \ldots, y_K]$ represents the model's predicted probability for each category; $\mathbf{t} = [t_1, t_2, \ldots, t_K]$ denotes the one hot encoded vector of actual labels; and $\mathbf{w} = [w_1, w_2, \ldots, w_K]$ represents the weight vector of

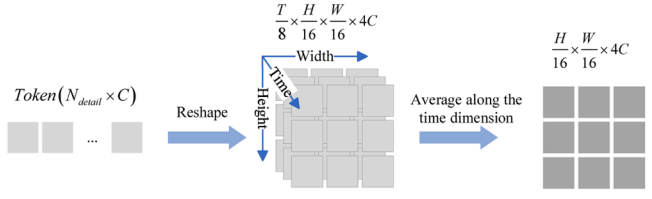

Fig. 4. The spatiotemporal blocks are averaged pooled along the time dimension.

categories.

4. Experiments

To validate the effectiveness of the proposed method, experiments were conducted on four common micro-expression datasets, including SMIC [28], CASME II [29], SAMM [30] and MMEW [31]. In this section, we will first introduce the datasets used in the experiment, the evaluation method, implementation details, etc. Then, we compare our method with other state-of-the-art methods and perform the ablation study. Finally, analysis is made to further disclose the method.

4.1. Datasets

The SMIC [28] dataset specifically refers to the HS data subset, consisting of 164 samples with 16 participants and 3 emotional labels. The CASME II [29] dataset comprises 249 samples and 5 emotional labels, involving 26 participants with an average age of 22.59, all from China. Similarly, the SAMM [30] dataset contains 159 samples and 8 emotional labels, involving 32 participants with a male-to-female ratio of 1 and an average age of 33.24 years, from 13 ethnic groups. Lastly, the MMEW [31] dataset includes 300 micro-expression sequences and 7 emotional labels, with 36 participants averaging 22.35 years, all from China.

Table 1 provides further details on the contents of each dataset. The "Three Emotions" row represents the reclassification based on basic emotions, such as categorizing Happiness as Positive emotions and Repression and Disgust as Negative emotions. The "Five Emotions" row lists the emotion labels used in the five-class classification for the

Table 1
The distribution of data in SMIC—HS, CASME II, SAMM, and MMEW datasets.

	SMIC—HS	CASME II	SAMM	MMEW
Subjects	16	25	28	30
FPS	100	200	200	90
Three Emotions	Positive (51), Negative (73), Surprise (42)	Positive (32), Negative (90), Surprise (28)	Positive (26), Negative (92), Surprise (15)	Positive (36), Negative (109), Surprise (89)
Five Emotions	–	Happiness (32), Repression (27), Surprise (28), Disgust (63), Others (99)	Happiness (26), Surprise (15), Anger (57), Contempt (12), Others (26)	Happiness (36), Surprise (89), Disgust (72), Fear (16), Others (66)

CASME II, SAMM, and MMEW datasets.

4.2. Evaluation methods

The evaluation uses leave-one-subject-out (LOSO) cross-validation to reduce dependence on individuals and accurately gauge generalization. In LOSO, each subject's samples serve as the test set while others form the training set, with metrics averaged across test sets. The evaluation metrics include accuracy (Acc) for correct classification, unweighted average recall (UAR), and unweighted F1 score (UF1). They are calculated as:

$$Acc = \frac{\Sigma_{i=1}^{K} TP_i}{M} \tag{4}$$

$$UAR = \frac{1}{K}\sum_{i=1}^{K}\frac{TP_i}{M_i} \tag{5}$$

$$UF1 = \frac{1}{K}\sum_{i=1}^{K}\frac{2T{\cdot}P_i}{2{\cdot}TP_i + FP_i + FN_i} \tag{6}$$

where K represents the number of micro-expression categories, M_i denotes the total number of samples belong to the i-th micro-expression category, M represents the total number of all samples. TP_i represents the number of true positive samples, FP_i represents the number of false positive samples, and FN_i represents the number of false negative samples for the i-th class.

4.3. Implementation details

Since CASME II and MMEW provide preprocessed facial data, facial alignment and cropping were only performed on the SMIC and SAMM datasets. The data preprocessing involves cropping facial keypoints using the Dlib algorithm [32] and the template face coordinates from Openface [33]. The input data for the model consists of RGB frame sequences, with T frames set to 8, and frame height H and width W both set to 224. The hyperparameter C is set to 128. The weight vector $\mathbf{w}$ is calculated according to the formula $w_i = M/(K{\cdot}M_i)$ on the SAMM dataset. In other datasets, $\mathbf{w}$ defaults to a vector of all ones.

The model is trained from scratch using AdamW [34] as the optimizer. The learning rate is linearly warmed up to 5e-5 for the first 15 epochs, followed by cosine decay for the next 45 epochs. The total number of epochs is 60, with a batch size of 4. The online data augmentation includes horizontal flipping, random cropping, and random rotation $(-4^{\circ}, +4^{\circ})$.

All experiments in this work were conducted on Ubuntu 22.04, with a GPU environment consisting of one GeForce GTX 1080 Ti. The deep learning framework used is PyTorch 1.12.

4.4. Model summary

Summary of our proposed EDMDBN model is shown in Table 4. The model has 50.8 million parameters, which is fewer than conventional Transformer-based image classifier like ViT-Base [21] and has lower computational complexity. In the GPU environment used in this study, the inference speed is 8.20 ms per video, approximately 975 frames per second (fps), which can meet the real-time processing requirements for most videos. Overall, our model achieves a good balance between parameter count and inference speed, making it suitable for applications that require real-time performance.

4.5. Comparison of with other methods

As shown in Table 2, experiments were conducted on the SMIC—HS, CASME II, and SAMM datasets, and our method achieves state-of-the-art results. On the SMIC—HS dataset, our method demonstrates improvements of 3.73 % and 2.4 % in ACC and UF1 indicators, respectively, over the second-best method AMAN [37]. On the CASME II dataset, our method shows improvements of 6.37 % and 2.91 % in Acc and UF1 indicators, respectively, compared to the second-best method MERSiamC3D [12]. On the SAMM dataset, our method exhibits improvements of 7.32 % and 5.28 % in Acc and UF1, respectively, compared to the second-best method AUGCN [36].

Table 2
Comparison on individual datasets.

Methods	SMIC—HS (3 classes)		CASMEII (5 classes)		SAMM (5 classes)	
	Acc (%)	UF1	Acc (%)	UF1	Acc (%)	UF1
MERSiamC3D (2021) [12]	73.56	0.7598	81.89	0.8300	68.75	0.6400
GEME (2021) [35]	64.63	0.6158	75.20	0.7354	55.88	0.4538
AUGCN (2021) [36]	N/A	N/A	74.27	0.7047	74.26	0.7045
SLSTT-LSTM (2022) [11]	75.00	0.7400	75.81	0.7530	72.39	0.6400
AMAN (2022) [37]	79.87	0.7708	75.40	0.7125	68.85	0.6682
TACL (2023) [38]	75.61	0.7584	76.30	0.7366	68.38	0.5436
FRL-DGT (2023) [39]	N/A	N/A	75.70	0.7480	N/A	N/A
EDMDBN (ours)	**83.60**	**0.7948**	**88.26**	**0.8591**	**81.58**	**0.7573**

*Data in bold indicates best results, underlines indicate next best results, and N/A indicates no reported results.

The composite dataset is the union of the three-classification datasets SMIC—HS, CASME II, and SAMM. As shown in Table 3, our method achieves the best performance on the composite dataset, as well as on each individual dataset, demonstrating the effectiveness of our method. Specifically, compared to the performance of TACL [38] on the composite dataset, our method shows improvements of 4.52 % and 8.42 % in terms of UF1 and UAR, respectively. This highlights the robustness and generalization capability of our method across different micro-expression recognition datasets.

Due to that the MMEW micro-expression dataset was released very recently, few studies have reported the results for both three and five classifications simultaneously. As shown in Table 5, our method achieves the Acc and UF1 of 92.70 % and 0.9216 respectively in the three-classification experiments. In the five-classification experiments, compared to Micro-ExpMultNet [42], our method exhibits slightly lower Acc and UF1 by 1.26 % and 2.9 %, respectively. Nevertheless, it should be noted that our method does not require computing optical flow sequences, making it more efficient than Micro-ExpMultNet [42].

4.6. Ablation & comparative experiments

Ablation experiments on the entire and detail branch. As shown in Table 6, we conducted three-classification experiments on the composite dataset and five-classification experiments on the CASME II and SAMM datasets for the ablation study. In the three-classification scenario, the fusion effect of the two branches, as indicated by the UF1 metric, is superior to that of the detail-motion branch, albeit slightly inferior to the entire motion branch. In the five-classification scenario, both the UF1 and Acc metrics demonstrate that the effect of using both entire and detail motion branches surpasses that of using single branches, with the advantages being more pronounced. From the perspective of the improved metrics, it can be seen that our model is more suitable for scenarios requiring fine-grained classification.

Ablation experiments on Token Attention module. Three-classification experiments were performed on the composite dataset, and five classification experiments were carried out on the CASME II and SAMM datasets, as presented in Table 8. According to the experimental results, the Swin-T model with accumulated attention demonstrates

Table 3
Comparison with other state-of-the-art methods on the three-classification composite dataset.

Methods	Composite		SMIC—HS		CASME II		SAMM	
	UF1	UAR	UF1	UAR	UF1	UAR	UF1	UAR
MERSiamC3D (2021) [12]	0.8068	0.7986	0.7356	0.7598	0.8818	0.8763	0.7475	0.7280
GEME (2021) [35]	0.7221	0.7303	0.6038	0.6087	0.8831	0.8790	0.5843	0.5455
AUGCN (2021) [36]	0.7914	0.7933	0.7192	0.7215	0.8798	0.8710	0.7751	0.7890
SLSTT-LSTM (2022) [11]	0.8160	0.7900	0.7400	0.7200	0.9010	0.8850	0.7150	0.6430
FeatRef (2022) [40]	0.7838	0.7832	0.7011	0.7083	0.8915	0.8873	0.7372	0.7155
TACL (2023) [38]	0.8369	0.8092	0.7739	0.7584	0.9370	0.9271	0.7919	0.7404
FRL-DGT (2023) [39]	0.8120	0.8110	0.7430	0.7490	0.9190	0.9030	0.7720	0.7580
EDMDBN (ours)	**0.8821**	**0.8933**	**0.7948**	**0.8085**	**0.9484**	**0.9619**	**0.8336**	**0.8661**

*Data in bold indicates best results, and underlines indicate next best results.

Table 4
Summary of the proposed EDMDBN model.

Parameter Count	Model Size	Computational Complexity	Inference Speed[1]
50.8 M	101.64 MB	13.2 G FLOPs	8.20 ms/video

[1] Inference Speed is calculated in the GPU environment of this experiment, averaging over 1000 preprocessed video (8 RGB frames, 224×224 pixels.).

Table 5
Three-classification and five-classification experiments on the MMEW dataset.

Methods	Three classifications		Five classifications	
	Acc (%)	UF1	Acc (%)	UF1
Micro-ExpMultNet (2022) [42]	N/A	N/A	**82.97**	**0.8086**
LD-FMEN (2023) [41]	88.23	0.8787	N/A	N/A
EDMDBN (ours)	**92.70**	**0.9216**	81.71	0.7796

Table 6
Ablation experiments on the entire and detail branch.

Methods	Composite (3 classes)		CASME II (5 classes)		SAMM (5 classes)	
	Acc (%)	UF1	Acc (%)	UF1	Acc (%)	UF1
Only entire branch	90.86	**0.8839**	82.98	0.8107	77.96	0.7381
Only detail branch	88.89	0.8705	86.26	0.8421	76.89	0.6837
Both	**91.03**	0.8821	**88.26**	**0.8591**	**81.58**	**0.7573**

superior performance compared to the baseline Swin-T model. The accumulated attention brought by the Token Attention Module enables the model to effectively capture relevant spatial information, enhancing its ability to extract discriminative features from the input, thereby improving classification accuracy.

Comparative experiments on feature fusion. The results presented in Table 9 demonstrate that feature addition yields better results compared to feature concatenation in our three-classification experiments on the composite dataset, and five-classification experiments on CASME II and SAMM datasets. Moreover, this effect is more pronounced in the five-classification process.

Comparative experiments on Feature vs. Pixel Subtraction. As shown in Table 7, we conducted three-classification experiments on the composite dataset and five-classification experiments on the CASME II and SAMM datasets. The experimental results indicated that directly using pixel subtraction for motion feature extraction yields better results than using feature subtraction. We think the reasons are as follows. On the one hand, the convolution operation in the feature extraction process may lead to subtle and local changes being blurred or overlooked, thereby reducing sensitivity to micro-expression motion. On the other hand, the data used in the current micro-expression experiments were collected in a controlled laboratory environment, with strict lighting and a uniform background, which makes it easier to detect and represent micro-expression actions at the pixel level.

Table 7
Comparative experiments on Feature vs. Pixel Subtraction.

Methods	Composite (3 classes)		CASME II (5 classes)		SAMM (5 classes)	
	Acc (%)	UF1	Acc (%)	UF1	Acc (%)	UF1
Feature subtraction[1]	87.84	0.8508	79.72	0.7469	0.7665	0.6999
Pixel Subtraction	**91.03**	**0.8821**	**88.26**	**0.8591**	**0.8158**	**0.7573**

[1] An additional learnable ResNet block is added before the dual-branch network to extract features frame by frame, while the rest remains unchanged.

Table 8
Ablation experiments on Token Attention module.

Methods	Composite (3 classes)		CASME II (5 classes)		SAMM (5 classes)	
	Acc (%)	UF1	Acc (%)	UF1	Acc (%)	UF1
Swin-T	90.74	0.8750	85.51	0.8231	0.8028	0.7279
Swin-T with AA	**91.03**	**0.8821**	**88.26**	**0.8591**	**0.8158**	**0.7573**

Table 9
Comparative experiments on feature fusion.

Methods	Composite (3 classes)		CASME II (5 classes)		SAMM (5 classes)	
	Acc (%)	UF1	Acc (%)	UF1	Acc (%)	UF1
Feature concatenation	89.50	0.8777	84.56	0.8116	77.90	0.7264
Feature addition	**91.03**	**0.8821**	**88.26**	**0.8591**	**81.58**	**0.7573**

4.7. Analysis and discussion

As depicted in Fig. 5, we present the confusion matrices for both composite dataset and individual datasets. Overall, it can be observed that our model demonstrates high consistency in performing category-based micro-expression recognition.

It can be seen from the confusion matrix shown in Fig. 5(a) that our model achieves recognition accuracies of 82 % or above for all the three categories on the composite dataset. Considering the confusion matrix illustrated in Fig. 5(b), our model attains a recognition accuracy of 93 % for Surprise on the CASME II dataset. However, Repression tends to be misclassified, indicating potentially low discriminative features.

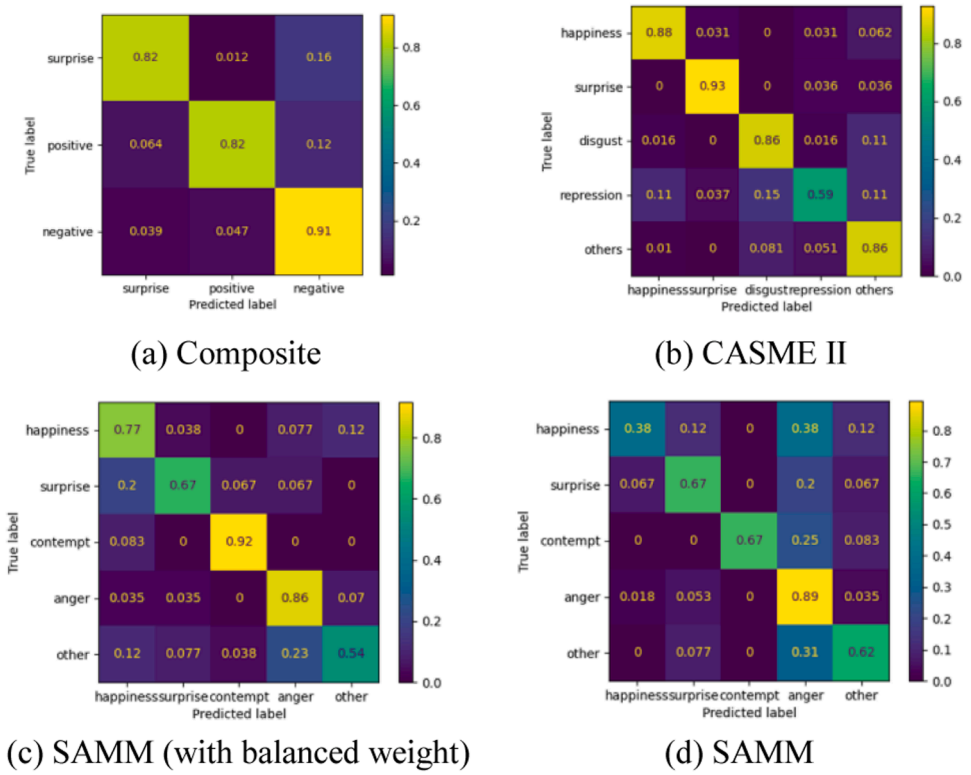

Fig. 5. Confusion matrices on composite and single datasets.

Regarding the confusion matrix shown in Fig. 5(c) and Fig. 5(d), the SAMM dataset exhibits significant class imbalance, with nearly a four-fold difference in sample counts between two classes. Consequently, we employed a weighting strategy to enhance the loss function, and the recognition accuracies of Happiness and Contempt witness notable increases by 39 % and 25 %, respectively.

To further explain our model, we use Grad-CAM to visualize the attention map. Grad-CAM is a gradient-based visual explanation technique that effectively highlights attention regions in deep neural networks, elucidating the model's predictive basis for specific categories. The heatmaps generated by Grad-CAM help us understand which facial regions are more attended to for specific categories of micro-expressions.

As illustrated in Fig. 6, on the CASME II dataset, different facial regions are activated during specific micro-expressions. For instance, when the micro-expression Happiness occurs, we observe activation in

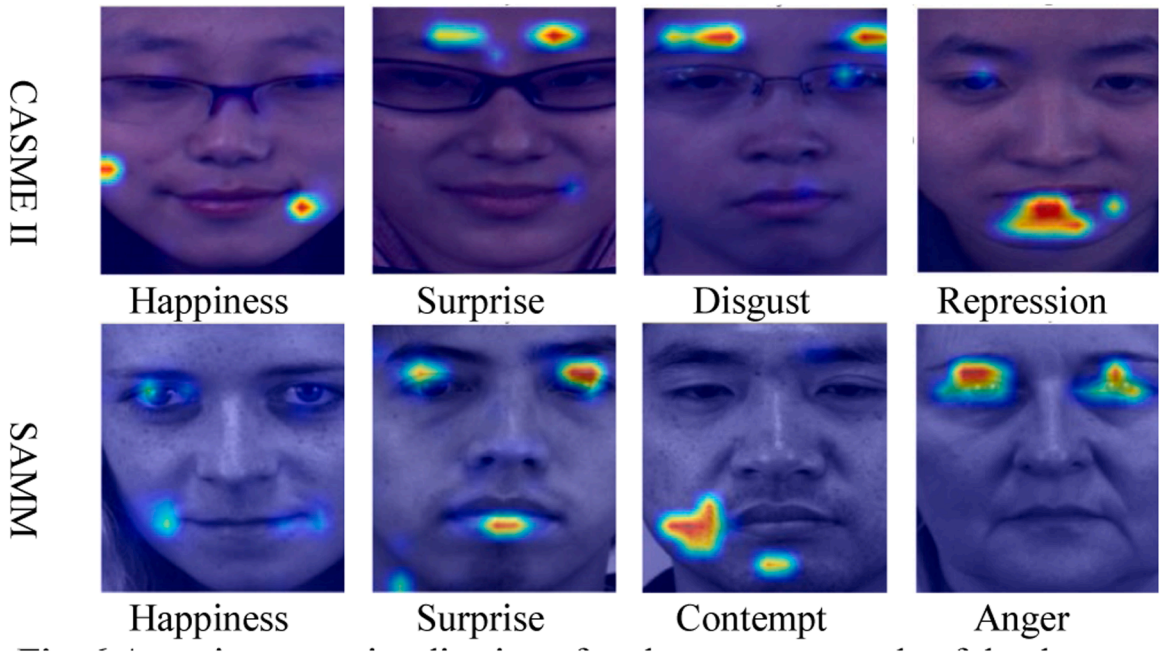

Fig. 6. Visualization of the attention maps of samples from different categories of the datasets.

the corners of the mouth (AU 12, mouth stretch) and cheeks (AU14, cheek raise). Conversely, during the occurrence of the micro-expression Surprise, the activation is prominent in the eyebrows (AU1 AU2, eyebrow raise) region. Similarly, for the micro-expression Disgust, the activation mainly occurs in the eyebrows (AU4, eyebrow lower) area, while for the micro-expression Repression, the chin (AU17, chin raise) region is activated. On the other hand, similar observations can be made for the SAMM dataset. For example, the micro-expression Happiness primarily activates the mouth corners (AU12, mouth stretch). Conversely, the micro-expression Surprise prompts activations in both the eyelids (AU5, eyelid raise) and mouth (AU 27, mouth open) regions. Furthermore, the micro-expression Contempt is characterized by activations in the mouth corners (AU12 L, left mouth stretch) and chin (AU17, chin raise). Lastly, the micro-expression Anger elicits activations in both the eyebrows (AU4, eyebrow lower) and eyelids (AU6, eyelid tighten) regions. The visualization of the attention maps further demonstrates the reliability of our proposed model.

5. Conclusions

In this study, based on the analysis of the characteristics of entire and detail motions in micro-expressions, we introduce a dual-branch network, named EDMDBN, for micro-expression recognition. Additionally, to better focus on the ROIs of micro-expressions, we propose incorporating the accumulated attention Swin-Transformer module to improve the model's capability in extracting micro-expression motion features. Experimental results on four micro-expression datasets, namely SMIC, CASME II, SAMM, and MMEW, showcase the superiority of our approach compared to state-of-the-art methods.

In our future work, we intend to introduce some contrastive learning methods to enhance the model's discrimination capability for those easily confused categories. Additionally, pixel differences are highly sensitive to factors such as noise and lighting changes, which may lead to unstable motion feature extraction. In practical applications, it is necessary to increase the model's robustness to noise and lighting changes through data augmentation techniques and adversarial training methods.

CRediT authorship contribution statement

Bingyang Ma: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Conceptualization. **Lu Wang:** Writing – review & editing, Supervision, Resources. **Qingfen Wang:** Methodology, Conceptualization. **Haoran Wang:** Writing – review & editing. **Ruolin Li:** Writing – review & editing. **Lisheng Xu:** Writing – review & editing. **Yongchun Li:** Funding acquisition. **Hongchao Wei:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Shenyang Science and Technology Plan Fund (No. 21–104–1–24), the National Natural Science Foundation of China (No. U21A20487 and No 62273082), the Natural Science Foundation of Liaoning Province (No. 2021-YGJC-14), the Basic Scientific Research Project (Key Project) of the Education Department of Liaoning Province (LJKZ00042021), and Fundamental Research Funds for the Central Universities (No. N2119008).

Data availability

Data will be made available on request.

References

[1] E. Sariyanidi, H. Gunes, A. Cavallaro, Learning bases of activity for facial expression recognition, IEEE Trans. Image Process. 26 (2017) 1965–1978, https://doi.org/10.1109/TIP.2017.2662237.
[2] P. Ekman, W.V. Friesen, Nonverbal leakage and clues to deception, Psychiatry 32 (1969) 88–106, https://doi.org/10.1080/00332747.1969.11023575.
[3] D. Matsumoto, H.S. Hwang, Evidence for training the ability to read microexpressions of emotion, Motiv. Emot. 35 (2011) 181–191, https://doi.org/10.1007/s11031-011-9212-2.
[4] P. Seidenstat, F.X. Splane, Protecting airline passengers in the age of terrorism, Bloomsbury Publishing USA, 2009.
[5] M. O'Sullivan, M.G. Frank, C.M. Hurley, J. Tiwana, Police lie detection accuracy: the effect of lie scenario, Law Hum Behav 33 (2009) 530–538, https://doi.org/10.1007/s10979-008-9166-4.
[6] U. Segerstråle, P. Molnár (Eds.), The evolution of emotions: the nonverbal basis of human social organization, Nonverbal Communication: Where nature meets culture, 1997, pp. 211–223.
[7] Y. Liu, H. Du, L. Zheng, T. Gedeon, A neural micro-expression recognizer, in: Proceedings of the 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), IEEE, 2019, pp. 1–4, https://doi.org/10.1109/FG.2019.8756583.
[8] Y. Li, X. Huang, G. Zhao, Joint local and global information learning with single apex frame detection for micro-expression recognition, IEEE Trans. Image Process. 30 (2021) 249–263, https://doi.org/10.1109/TIP.2020.3035042.
[9] R. Zhi, H. Xu, M. Wan, T. Li, Combining 3d convolutional neural networks with transfer learning by supervised pre-training for facial micro-expression recognition, IEICE Trans. Inf. Syst. E102.D (2019) 1054–1064, https://doi.org/10.1587/transinf.2018EDP7153.
[10] J. Hong, C. Lee, H. Jung, Late fusion-based video transformer for facial micro-expression recognition, Appl. Sci. 12 (2022) 1169, https://doi.org/10.3390/app12031169.
[11] L. Zhang, X. Hong, O. Arandjelović, G. Zhao, Short and long range relation based spatio-temporal transformer for micro-expression recognition, IEEE Trans. Affect. Comput. 13 (2022) 1973–1985, https://doi.org/10.1109/TAFFC.2022.3213509.
[12] S. Zhao, H. Tao, Y. Zhang, T. Xu, K. Zhang, Z. Hao, E. Chen, A two-stage 3D CNN based learning method for spontaneous micro express-ion recognition, Neurocomputing 448 (2021) 276–289, https://doi.org/10.1016/j.neucom.2021.03.058.
[13] Y.S. Gan, S.-T. Liong, W.-C. Yau, Y.-C. Huang, L.-K. Tan, OFF-ApexNet on micro-expression recognition system, Signal Process. Image Commun. 74 (2019) 129–139, https://doi.org/10.1016/j.image.2019.02.005.
[14] S.-T. Liong, Y.S. Gan, J. See, H.-Q. Khor, Y.-C. Huang, Shallow triple stream three-dimensional CNN (STSTNet) for micro-expression recognition, in: Proceedings of the 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), IEEE, 2019, pp. 1–5, https://doi.org/10.1109/FG.2019.8756567.
[15] H. Li, M. Sui, Z. Zhu, F. Zhao, MMNet: muscle motion-guided network for micro-expression recognition. IJCAI, Elsevier, 2022, pp. 1074–1080, https://doi.org/10.24963/ijcai.2022/150.
[16] N.V. Quang, J. Chun, T. Tokuyama, Capsulenet for micro-expression recognition, in: Proceedings of the 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), IEEE, 2019, pp. 1–7, https://doi.org/10.1109/FG.2019.8756544.
[17] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, TransFG: a transformer architecture for fine-grained recognition, in: Proc. AAAI Conf. Artif. Intell. 36, 2022, pp. 852–860, https://doi.org/10.1609/aaai.v36i1.19967.
[18] X. Fan, X. Chen, M. Jiang, A.R. Shahid, H. Yan, SelfME: self-supervised motion learning for micro-expression recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2023, pp. 13834–13843, https://doi.org/10.1109/CVPR52729.2023.01329.
[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90, https://doi.org/10.1145/3065386.
[20] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: ICLR, Curran Associates, 2017. http://arxiv.org/abs/1609.02907.
[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16×16 words: transformers for image recognition at scale, in: ICLR, Curran Associates, 2021. http://arxiv.org/abs/2010.11929.
[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL, Association for Computational Linguistics, 2019, pp. 4171–4186. http://arxiv.org/abs/1810.04805.
[23] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2021, pp. 9992–10002, https://doi.org/10.1109/ICCV48922.2021.00986.
[24] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, H. Hu, in: Video swin transformer, IEEE, 2022, pp. 3192–3201, https://doi.org/10.1109/CVPR52688.2022.00320.

[25] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A.S. Albahri, B.S.N. Al-dabbagh, M.A. Fadhel, M. Manoufali, J. Zhang, A.H. Al-Timemy, Y. Duan, A. Abdullah, L. Farhan, Y. Lu, A. Gupta, F. Albu, A. Abbosh, Y. Gu, A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications, J. Big. Data 10 (2023) 1–82, https://doi.org/10.1186/s40537-023-00727-2.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural. Inf. Process. Syst. 27 (2014) 2672–2680. https://dl.acm.org/doi/10.5555/2969033.2969125.

[27] X. Huang, S.-J. Wang, X. Liu, G. Zhao, X. Feng, M. Pietikainen, Discriminative spatiotemporal local binary pattern with revisited integral projection for spontaneous facial micro-expression recognition, IEEE Trans. Affect. Comput. 10 (2019) 32–47, https://doi.org/10.1109/TAFFC.2017.2713359.

[28] X. Li, T. Pfister, X. Huang, G. Zhao, M. Pietikainen, A spontaneous micro-expression database: inducement, collection and baseline, in: Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), IEEE, 2013, pp. 1–6, https://doi.org/10.1109/FG.2013.6553717.

[29] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, X. Fu, CASME II: an improved spontaneous micro-expression database and the baseline evaluation, PLoS ONE 9 (2014) e86041, https://doi.org/10.1371/journal.pone.0086041.

[30] A.K. Davison, C. Lansley, N. Costen, K. Tan, M.H. Yap, SAMM: a spontaneous micro-facial movement dataset, IEEE Trans. Affect. Comput. 9 (2018) 116–129, https://doi.org/10.1109/TAFFC.2016.2573832.

[31] X. Ben, Y. Ren, J. Zhang, S.-J. Wang, K. Kpalma, W. Meng, Y.-J. Liu, Video-based facial micro-expression analysis: a survey of datasets, features and algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2021) 5826–5846, https://doi.org/10.1109/TPAMI.2021.3067464.

[32] D.E. King, Dlib-ml: a machine learning toolkit, The J. Mach. Learn. Res. 10 (2009) 1755–1758.

[33] T. Baltrusaitis, A. Zadeh, Y.C. Lim, L.-P. Morency, in: OpenFace 2.0: facial behavior analysis toolkit, IEEE, 2018, pp. 59–66, https://doi.org/10.1109/FG.2018.00019.

[34] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in: ICLR, Curran Associates, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

[35] X. Nie, M.A. Takalkar, M. Duan, H. Zhang, M. Xu, GEME: dual-stream multi-task GEnder-based micro-expression recognition, Neurocomputing 427 (2021) 13–28, https://doi.org/10.1016/j.neucom.2020.10.082.

[36] L. Lei, T. Chen, S. Li, J. Li, Micro-expression recognition based on facial graph representation learning and facial action unit fusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) IEEE/CVF, 2021, pp. 1571–1580, https://doi.org/10.1109/CVPRW53098.2021.00173.

[37] M. Wei, W. Zheng, Y. Zong, X. Jiang, C. Lu, J. Liu, in: A novel micro-expression recognition approach using attention-based magnification-adaptive Networks, Proceedings of the ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 2420–2424, https://doi.org/10.1109/ICASSP43922.2022.9747232.

[38] T. Wang, L. Shang, Temporal augmented contrastive learning for micro-expression recognition, Pattern Recognit. Lett. 167 (2023) 122–131, https://doi.org/10.1016/j.patrec.2023.02.003.

[39] Z. Zhai, J. Zhao, C. Long, W. Xu, S. He, H. Zhao, in: Feature representation learning with adaptive displacement generation and transformer fusion for micro-expression recognition, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2023, pp. 22086–22095, https://doi.org/10.1109/CVPR52729.2023.02115.

[40] L. Zhou, Q. Mao, X. Huang, F. Zhang, Z. Zhang, Feature refinement: an expression-specific feature learning and fusion method for micro-expression recognition, Pattern Recognit. 122 (2022) 108275, https://doi.org/10.1016/j.patcog.2021.108275.

[41] R. Ni, B. Yang, X. Zhou, S. Song, X. Liu, Diverse local facial behaviors learning from enhanced expression flow for microexpression recognition, Knowl. Based Syst. 275 (2023) 110729, https://doi.org/10.1016/j.knosys.2023.110729.

[42] X. Zhao, Y. Lv, Z. Huang, in: Multimodal fusion-based swin transformer for facial recognition micro-expression recognition, Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA), IEEE, 2022, pp. 780–785, https://doi.org/10.1109/ICMA54519.2022.9856162.