Micro-expression Recognition Based on Facial Graph Representation Learning and Facial Action Unit Fusion

Ling Lei[1], Tong Chen[1], Shigang Li[1, 2*] and Jianfeng Li[1*]
[1]Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, China
[2]Graduate School of Information sciences, Hiroshima City University, Hiroshima, Japan
leiling_swu@163.com, c_tong@swu.edu.cn, shigangli@hiroshima-cu.ac.jp, popqlee@swu.edu.cn

Abstract

Micro-expressions recognition is a challenge because it involves subtle variations in facial organs. In this paper, first, we propose a novel pipeline to learn a facial graph (nodes and edges) representation to capture these local subtle variations. We express the micro-expressions with multi-patches based on facial landmarks and then stack these patches into channels while using a depthwise convolution (DConv) to learn the features inside the patches, namely, node learning. Then, the encoder of the transformer (ETran) is utilized to learn the relationships between the nodes, namely, edge learning. Based on node and edge learning, a learned facial graph representation is obtained. Second, because the occurrence of an expression is closely bound to action units, we design an AU-GCN to learn the action unit's matrix by embedding and GCN. Finally, we propose a fusion model to introduce the action unit's matrix into the learned facial graph representation. The experiments are comparing with SOTA on various evaluation criteria, including common classifications on CASME II and SAMM datasets, and also conducted following Micro-expression Grand Challenge 2019 protocol.

1. Introduction

Facial expression is the most direct way of expressing human emotion, and it is also a very important way to understand human intention in human-centered computing. Facial expressions are divided into macro-expressions and micro-expressions (MEs). From a temporal perspective, macro-expressions last between 0.75 s and 2 s, while MEs last between 0.04 s and 0.2 s [1]. In spatial terms, facial muscle movements of MEs are more slight than those in macro-expressions. In addition, MEs are spontaneous, meaning that a person does not know when he or she is exhibiting MEs. Consequently, MEs can show people's real emotions; therefore, MEs are often examined on important occasions, such as the diagnosis of mental diseases and during interrogation of major crime cases. Recognizing the emotional categories is difficult because revealing the spatial-temporal aspect in MEs is not obvious. Important micro information is easy to ignore especially when people use the naked eye to observe. A Micro-Expression Training Tool (METT) [2] has been developed to aid recognition. Even for professionals with special training, the results are still not ideal. Therefore, computer technology is needed to assist with micro-expression recognition (MER).

In recent years, the development of ME has been very rapid. The literature includes three major categories as follows: LBP-based [22, 23, 25, 26, 27, 28, 29, 30, 31], optical-flow-based [32, 33, 34, 35, 36], and other novel methods (mainly deep learning) [1, 3, 4, 9, 14, 15, 16, 17, 20, 37, 38, 39, 40, 41, 42, 43, 44]. The expansion of knowledge in the deep learning community has made great strides in the development of computer vision, as well as the field of MER. The current technology, after a series of preprocessing operations, first uses the method based on optical flow [1, 3, 4, 5, 6, 7, 8, 9] to extract the features of MEs. Efficient neural networks, usually CNN [10, 11], LSTM [12], GCN [13], etc., and their variant structures or combinations are designed to complete the feature learning task. In addition, there are some novel methods for MER, such as transfer learning [14], capsule networks [15], and knowledge distillation [16].

The variation in ME is caused by the subtle movements of facial muscles. Meanwhile, the variation of muscle movements reflected in the ME frame is mainly geometry variation. In contrast, texture variation is more subtle, and is easily disturbed by factors such as race and light environment. The optical flow-based method mentioned above mainly focuses on extracting facial geometry features of MEs, while the LBP-based method mainly focuses on extracting facial texture features of MEs.

[*] Corresponding Author

Therefore, the feature extraction method based on LBP is not the best for recognition effect. However, because feature extraction using the optical flow method is the result of manual calculation, it is less self-adaptable when compared with the deep learning method. In a previous study [17], Lei *et al*. proposed a method from the perspective of deep learning to introduce a learning-based video motion magnification network (MagNet) [18] into the magnification step of MEs through transfer learning. The magnified shape representation, which is also called geometry features, is extracted from the intermediate layer as inputs for further feature extraction and learning. This method had the best accuracy rate at the time of report, which proves that magnified geometry features have a robust contribution to MER. However, in the subsequent process, this method roughly reduces the two-dimensional feature to a one-dimensional vector and ignores the loss of spatial information. Moreover, this method does not take into account the mechanism by which facial expressions are encoded by action units (AUs). For example, the expression of happiness is encoded by AU6 (cheek raise), AU12 (lip corner puller), and AU25 (lips part) [19]. Therefore, there is room for improvement.

In the facial action coding system (FACS) [19] [54], the AUs explain the occurrence of facial expressions as facial movement based on muscles. Each AU corresponds to the facial movement of a specific area, and different classes of expressions correspond to the combination of different AUs. Therefore, the information contained in the AUs can be helpful for facial expression recognition. As a kind of facial expression, ME is also applicable, especially in the recognition method based on geometric features, and adding the information of the AUs will improve the recognition performance. In 2020, the latest two papers on this topic [16, 20] both propose a network structure in which AU information is introduced through dual channels. One channel is generally a common feature extraction method for MEs, and the other channel contains AU information. In the channel with AU information, AUs are introduced by the network structures of knowledge distillation [21] or GCN [13]. In the end, the two channels merge in a specific way. It is common knowledge that the variation of MEs is very subtle in the whole face; therefore, it is very rough to drop the whole image into the network directly. Thus, Lei *et al.* [17] utilized a facial graph proposed by Zhong *et al*. [53] to focus on these subtle but important parts, while using magnified geometric features as nodes, which has been proven to be more effective.

In response to the above problems, in this paper, the main contributions to the MER field are as follows:

(1) We propose a novel pipeline to learn a facial graph (nodes and edges) representation based on magnified geometry features. Depthwise convolution is adopted for node learning, and the encoder of the transformer is adopted for edge learning.

(2) The AU information is learned through the GCN in the form of an adjacency matrix based on conditional probability. Moreover, we propose a reasonable two-channel fusion mechanism that efficiently combines the AU matrix with facial graph representation.

(3) Finally, we propose an end-to-end trainable MER network, which achieves the best recognition rate in two public datasets and their composite dataset.

2. Proposed method

In this paper, our proposed network structure is that one channel learns a facial graph representation, another channel learns an action unit matrix, and a novel mechanism is utilized to fuse the outputs of these two channels to recognize MEs. In our proposed method, as Figure 1 shows, there is one onset frame and one apex frame input into MagNet [18] to extract the magnified shape

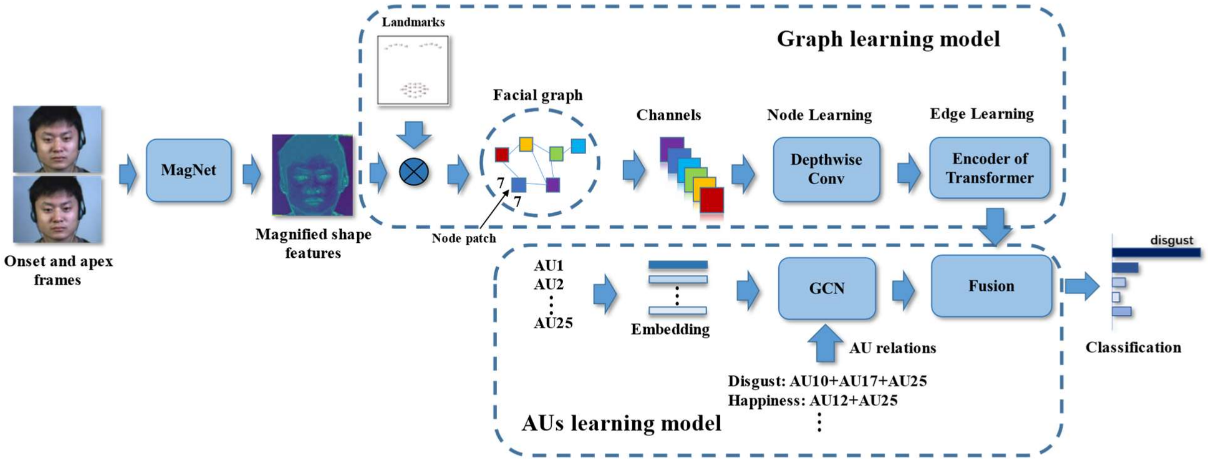

Figure 1: Proposed network structure.

features from the intermediate layer. Similar to [17], we extracted 30 node patches with a size of *7×7* based on the eyebrow and mouth landmarks, which can be seen as a facial graph. Then, the graph representation was learned by the proposed node learning and edge learning models. On the other hand, nine AUs belonging to the eyebrow and mouth areas were embedded and fed into the GCN with their relationships to achieve the AU feature matrix [20]. Finally, using a designed fusion strategy, the final ME classification was performed by combining AUs with the learned facial graph representation.

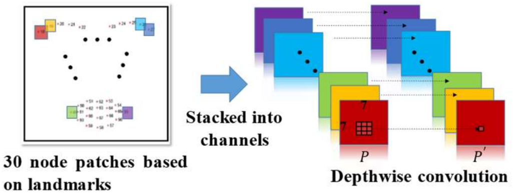

Figure 2: ***Node learning*****: DConv.** The facial graph is seen as the multi-channels patches and learns the nodes features by channel-wise convolution. $\boldsymbol{P}, \boldsymbol{P'} \in \boldsymbol{R}^{\boldsymbol{H}\times\boldsymbol{W}\times\boldsymbol{N}}$.

2.1. Node learning: Integrating features inside node patches by depthwise convolution

Chollet *et al*. [45] proposed the mechanism by which the mapping of cross-channel correlations and spatial correlations in the feature maps of convolutional neural networks can be entirely decoupled. Based on this, depthwise separable convolution (DWSConv) was designed, which consists of depthwise convolution and pointwise convolution. In short, DWSConv decomposes the complete traditional convolution process into two steps. The first step independently performs a spatial convolution over each channel while keeping the number of channels unchanged. The second step performs a *1×1* convolution to project the channel space from the input onto the output. Xception, an architecture, based on the DWSConv with residual connections, has proven to be practical on many datasets.

As Figure 1 shows, a magnified shape representation was extracted from MagNet, and patches based on landmarks as nodes were extracted from the shape representation. At this time, a multidimensional matrix with a size of *N* × *H* × *W* was obtained, where *N* is the number of landmarks and *H* and *W* are the height and width of the patch size, respectively. To address these patches, one straightforward method [17] is to compress the two-dimensional patches with a size of *H* × *W* directly into a one-dimensional vector with a size of *1*× *(H*×*W)* to represent the nodes of the graph structure. However, the rough process loses the vertical spatial information between pixels inside the patches. Therefore, to integrate the spatial information in patches using deep learning, in this paper, we innovatively regard the patches as channels. From this perspective, our proposed graph structure with *N* node patches can be seen as an image with *N* channels, and then depthwise convolution can be applied to integrate features from each channel. Finally, the proposed depthwise convolution (DConv) can preserve the internal spatial information of each node patch. We call this step node learning.

As shown in Figure 2, DConv can ensure that patches (*H* × *W*) from *N* channels are convolved separately and do not interfere with each other. The two-dimensional spatial features inside the patches are extracted by convolution to preserve spatial information. See section 3 for the implementation details. In the subsequent ablation analysis, we also proved the effectiveness of this module and performed parameter analysis experiments on the size of the convolution kernel.

2.2. Edge learning: Learning relationship between node patches by encoder of transformer

The vanilla transformer [46] consists of two modules, the encoder and decoder. The encoder consists of six layers, each mainly consisting of a multi-head self-attention mechanism and a fully connected feed-forward network. The decoder is also a stack of six layers, but each layer is mainly composed of a multi-head self-attention mechanism, a multi-head self-attention mechanism combined with the output of the encoder and decoder, and a fully connected feed-forward network. The function of the encoder is to learn the input features based on a multi-head self-attention mechanism to obtain an effective feature map. The decoder

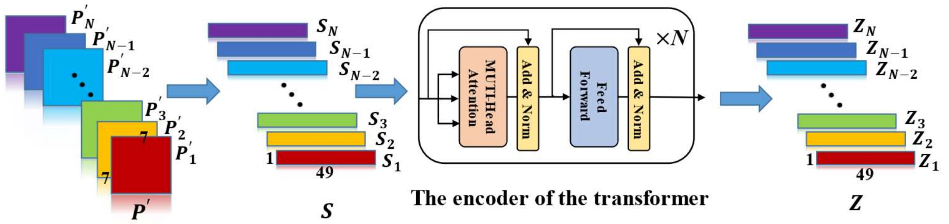

Figure 3: ***Edge learning*****: ETran.** The encoder of the transformer. The multi-channels node patches $\boldsymbol{P'}$ are transformed to sequential vectors $\boldsymbol{S}$. The encoder of the transformer is used to learn the edges features. $\boldsymbol{S}, \boldsymbol{Z} \in \boldsymbol{R}^{\boldsymbol{1}\times\boldsymbol{D}}, \boldsymbol{D} = \boldsymbol{H}\times\boldsymbol{W}$.

will combine with the feature map learned by the encoder to carry out feature learning based on a multi-head self-attention and predict the output of each position. Currently, the transformer and its improved versions are widely used in the NLP community.

In the feature learning of depthwise convolution described above, node learning is considered. The next step requires global feature learning between node patches, namely, edge learning. First, we utilized the attention mechanism ETran (Encoder of Transformer) for ME feature extraction due to the mechanism of the transformer, which can compute the relationships between components in a sequence. Each node patch as an independent component was fed into ETran, and then it is possible to learn the relationship, namely, edges, between nodes. Up to this step, the whole graph representation has been automatically learned from DConv and ETran.

As shown in Figure 3, to feed the node patches $P' \in R^{N\times H\times W}$ into ETran, patches $P_i' \in R^{H\times W},\ i \in (1,2,..,N)$ are compressed to vectors $S_i \in R^{1\times D},\ i \in (1,2,..,N),\ D = H \times W$. Because node learning has been performed before this step, the spatial feature loss inside patches resulting from compression can be avoided to some extent. Then, following the rules of transformers, each of the vectors S_i produces Q_i, K_i, and V_i, $i \in (1,2,..,N)$, which are queries and a set of key-value pairs, respectively. The rules of the multi–head self-attention [46] are as follows:

For each head:

$$O_i(1,\dots,N) = Softmax(\frac{Q_i K_1^T}{\sqrt{d_K}},\dots,\frac{Q_i K_N^T}{\sqrt{d_K}}) \tag{1}$$

$$Y_i = O_i(1)V_1 +,\dots,+ O_i(N)V_N \tag{2}$$

For 8 heads:

$$Z_i = Concat\left(Y_i^{head1},\dots,Y_i^{head8}\right)W^O \tag{3}$$

W^O is a weight matrix. We use eight heads and ignore the subsequent operations of residual connections and feed forward to simplify formula presentation. In the case of S_1, its Q_1 would be multiplied by the K_1, …, K_N, divided by the square root of d_k, and then calculated by Softmax to give $O_1(1,\dots,N)$. Each element in O_1 is multiplied by V_1..., V_N, and then added to get Y_1, which is the result after attention matching with the global information. That is, each output Y_i is fully integrated with every S_i (one S_i means one node patch). As Figure 3 shows, we obtained a new matrix. See section 3 for the detailed implementation. The effectiveness of ETran has also been proven in the subsequent ablation analysis, and the analysis experiment of parameters in ETran has also been carried out.

2.3. AUFusion: Importing AUs by GCN

GCN [13] performs convolution based on a graph that can be understood as a non-Euclidean structure (CNN cannot be used on this structure). A graph consists of nodes and edges (a relationship between nodes). When GCN is performed, nodes and edges are input as node matrix $X \in R^{n\times d}$ and adjacency matrix $A \in R^{n\times n}$, in which n is the number of nodes and d is the dimension of each node vector. The layer-wise propagation rules [13] are as follows:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{4}$$

$$\tilde{A} = A + I_N \tag{5}$$

$$\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij} \tag{6}$$

$H^{(l)}$ is the input data of the l-th layer, and $H^{(0)} = X$. I_N is the identity matrix. $\sigma(\cdot)$ represents the nonlinear activation function. $W^{(l)}$ is the trainable weight matrix. Based on these rules, node features can be updated according to the relationship between them.

ME can be represented by different combinations of AUs. When an ME appears, the AUs associated with the ME of this emotion type will be activated. In view of this, there is a dependency between each AU, and the cooccurrence in the training set can be used to describe the relationship between them [20]. As mentioned previously, our method only utilizes features from the eyebrows and mouth, and we selected nine AUs involved according to the FACS [19]. Thus, the cooccurrence relationship between these nine AUs can form an adjacent matrix $A_{AU} \in R^{9\times 9}$. On the other hand, we applied word embedding [56] to learn the node matrix of these nine AUs. Embedding is a learned lookup table that is popular in the NLP community. Embedding can map the word sequences from their idx space $\{0,1,2,\dots,n\}$ to high-dimensional space $X \in R^{n\times d}$, which can make the machine better understand language. Here, nine AUs can be expressed as $X_{AU} \in R^{9\times d}$, which is learned by back propagation in embedding. Finally, we used GCN to learn

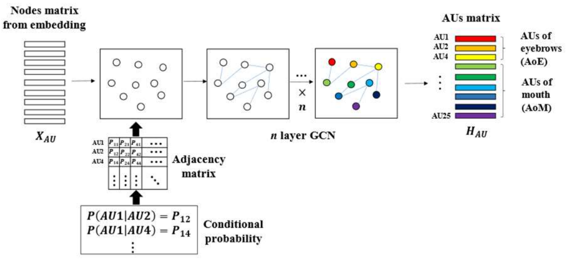

Figure 4: AUGCN

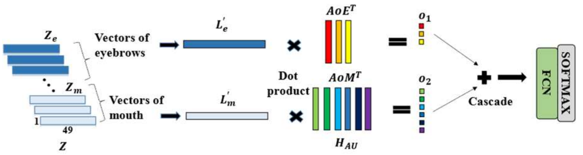

Figure 5: AUFusion

more precise features for the AU node matrix $H_{AU} \in R^{9\times d}$ based on the adjacent matrix and trainable weight matrix in Equation (4). At this point, the network has achieved the AU matrix and a learned facial graph representation (described in edge learning). To combine AUs with our facial graph representation, we designed a fusion network to combine $H_{AU} \in R^{9\times d}$ with learned facial features in edge learning. We named this pipeline AUFusion.

We designed a model, namely, AUGCN, which can be seen in Figure 4. The node matrix was generated by word embedding [56]. More specifically, 9 AUs are represented by 0, 1, 2, …, 8 and stored in an input vector. Then, the embedding class (torch.nn.Embedding) in Pytorch is utilized to map the input vector to the nodes matrix. The adjacency matrix was generated by conditional probability [20]. Two of them are fed into the GCN of two layers for feature learning. After this, as seen in Figures 4 and 5, the output H_{AU} was divided into two parts, since three of the AUs used only occur in the eyebrows (AoE), and the six remaining AUs only occur in the mouth (AoM). This is also an advantage of using facial graph representation that compares other representations. We can flexibly separate features (two ETran for two regions separately), eyebrow facial features fusing eyebrow AU features while mouth facial features fuse mouth AU features. As seen in Figure 5, the AUs of the eyebrows (AoE) and AUs of the mouth (AoM) take the dot product with each corresponding vector from ETran. Then, o_1 and o_2 are cascades for classification. See section 3 for the detailed implementation. Experiments on the selection of the method to generate a node matrix have also been carried out.

3. Implementation of the experiments

3.1. Datasets

In our experiment, the Chinese Academy of Sciences Micro-expression II (CASME II) [47], Spontaneous Activity and Micro-Movements (SAMM) [48] datasets, and Spontaneous micro-expression corpus (SMIC) [24] were used, which are currently the three most commonly used datasets.

In CASME II, the recording rate of the camera is 200 fps, the resolution is *640×480*, and the facial resolution is *280×340*. The participants are one ethnicity. The total number of samples is 255. CASME II has emotion labels, apex frame labels and AU labels. Since AU labels for five of the samples were not been provided, the total number of samples in our experiment was 250.

In SAMM, the recording rate of the camera is 200 fps, the resolution is *2040×1088*, and the facial resolution is *400×400*. The participants are 13 different ethnicities. The total number of samples with emotion labels, apex frame labels, and AU labels was 159.

In SMIC, the recording rate of the camera is 100 fps, the resolution is 640 ×480. The ethnicities of participants are diverse (Asian, Caucasians, and African). The total number of samples without apex frame labels, and AU labels was 164. Follow the previous work [3], the apex frames can be roughly spotted by the difference deviation.

In our experiments, evaluations based on four classifications and five classifications were carried out on CASME II and SAMM, respectively. At the same time, we also evaluated the composite database (CASME II + SAMM with 409 of the total samples) based on four classifications. The dataset partitioning method of the four classifications refers to [3], and the five classifications refer to [8, 17]. The AUs associated with the eyebrow and mouth areas are AU1, AU2, AU4, AU10, AU12, AU14, AU15, AU17, and AU25 [19]. Besides, we utilized the composite database evaluation (CDE) protocol from the Second Facial Micro-Expressions Grand Challenge (MEGC 2019) [55]. The CDE reorganizes CASME II, SAMM, and SMIC into 442 samples with 3 class and 68 subjects. Because the SMIC doesn't provide the annotations of the AUs, we only use the AUs information of the CASME II and SAMM to generate the adjacency matrix to apply our proposed methods to CDE.

3.2. Preprocessing

The onset and apex frames were extracted from the sequence of ME frames. The onset frame is the beginning of the ME, and the apex frame is the moment when the muscle movement of the ME is most intense. These frames were obtained through the annotation of the dataset. Then, the frames were aligned, cropped, gray processed and augmented. We obtained images I_1 (onset frame) and I_2 (apex frame) with a size of *256×256.*

3.3. Patches on the facial landmarks from magnified shape representation

First, following [17], I_1 and I_2 were put into MagNet [18] to obtain the magnified g (shape representation) from the intermediate layer. The specific operation of the network was to multiply the shape difference (between two shape representations which are from I_1 and I_2) with the α (magnified factor) and then add it to the I_2 to obtain the magnified g with the size of *128×128*. Based on this, the different α (1.2, 1.4, 1.6, …, 2.8, 3.0) can be used to augment the dataset again. The coordinates of the 68 facial landmarks of the magnified g were obtained based on DILB. Then, the *7×7* patches from 30 facial landmarks of the eyebrow and mouth regions were extracted. The size of the P was *30×7×7*. There were 30 landmarks, also called 30 channels. The height and width were *7×7.*

3.4. Global feature learning

The P was sent into the DConv module (one layer) to make each channel of P convolve independently and learn node features. The specific operation was to set the parameters of conv2D as in_channels = out_channels = groups = 30 in Pytorch, the deep learning framework, and use the same padding method to keep the size of the feature map unchanged. After this operation, P' with a size of *30×7×7* was obtained. We transformed P' into one-dimensional sequential vectors S with a size of *30×49*. In 30 facial landmarks, the first 10 were points of the eyebrows, and the remaining 20 were points of the mouth, with a ratio of 1:2. Therefore, we divided S into S_e with a size of *10×49* and S_m with a size of *20×49*. S_e and S_m used the ETran module (six layers of encoder) for edge learning based on multi-head self-attention to obtain Z_e and Z_m with sizes of *10×49* and *20×49,* respectively. Then, Z_e and Z_m were transformed into one-dimensional vectors L_e and L_m with lengths of 490 and 980, respectively. Next, two fully connected layers were used to obtain L'_e and L'_m, and the lengths were both transformed into 160 to match the transformation of the dimension when fusing with another channel containing AU information.

3.5. Learning the features of AUs

According to previous work [20], the GCN is used to learn the features of AUs. The GCN has two important parts as follows: the adjacency matrix and the node matrix. The adjacency matrix A_{AU} uses conditional probability to construct the AU information. The size of A_{AU} is *9×9*. There are nine nodes, which means that nine AUs related to the eyebrows and mouth were selected. The difference in our experiment is that when node matrix $X_{AU}$is constructed, we adopted the word embedding [56] approach. The size of X_{AU} is *9×40*, and 40 is the dimension of each node. Then, X_{AU} and A_{AU} were fed into the two-layer GCN. Node matrix X_{AU} learns features according to A_{AU} to obtain feature output H_{AU} , which has a size of *9×160*. The dimension of the output after the processing of the two-layer GCN was 160.

3.6. Fusion of the features from two channels

In matrix H_{AU}, which represents the nine AUs, the first three rows are related to AUs of the eyebrows (AoE), and the remaining six rows are related to AUs of the mouth (AoM). Therefore, H_{AU} can be divided into H_{AUe} and H_{AUm} with sizes of *3×160* and *6×160*, respectively. L'_e and L'_m are the dot products with ${H_{AUe}}^T$ and ${H_{AUm}}^T$, respectively. The results are o_1 and o_2 with lengths of 3 and 6, respectively. Finally, o_1 and o_2 are cascaded to one vector o. The o is sent to fully connected layers and Softmax for the classification.

4. Ablative analysis

In our ablative analysis, we analyzed the effectiveness of our designed models (DConv, ETran, AUGCN and AUfusion), the impact of the parameters in ETran and DConv, and the performance of different methods that generate the node matrix. All the experiments in the ablative analysis were conducted on CASME II with 4 classes. The reason for this design is that our goal is to find the optimized configuration of network parameters and structure on an ethnically homogenous dataset (CASME II) and use this configuration to generalize to another ethnically diverse dataset (SAMM) as well as the composite dataset (CASME II + SAMM). In the experiments, the leave-one-subject-out (LOSO) protocol was used to evaluate our proposed method. Accuracy and F1-score were used to compute and evaluate the results. In Equation (7), T represents the total number of correct predictions, and N represents the total number of test samples. In Equation (8), P represents the precision and R represents the recall.

$$acc = \frac{T}{N} \times 100\% \tag{7}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{8}$$

In the model analysis, we designed three experiments as comparison groups, which remove one model of our proposed models. The three comparison groups were ETran + AUGCN + AUFusion, DConv + AUGCN + AUFusion, and DConv + ETran. Our proposed method is DConv + ETran + AUGCN + AUFusion. From Table 1, we see that our proposed models all contribute to MER. Among them, the DConv model contributes the most. This result means that preserving the spatial information of the window patches is helpful for improving the effectiveness of feature learning. Besides, the Graph-tcn [17] also utilized the facial graph to learn the representation of ME. In order to prove our proposed facial graph learning channel (DConv+Etran) is effective, the Graph-tcn [17] is reproduced on CASME II with 4 classes and the accuracy is 73.60%. The accuracy of the DConv+Etran is 78.80%, which can show that our proposed facial graph learning channel surpass the Graph-tcn.

In the parameter analysis of ETran and DConv, we conducted some experiments on d_k , d_v and n_h of the ETran and k of the kernel size in DConv. For the parameter analysis of ETran, we referred to the method of [46], which sets d_k and d_v as equal and keeps their product with n_h as a constant value. For DConv, we changed the kernel size of the convolution from 3 to 5. As shown in Table 2, the optimal combination of parameters is d_k, d_v=16, n_h=8, and $k = 3$.

In the experiment to select the methods that generate the node matrix, we reproduced the one-hot method that is used in [20]. In this experiment, nodes of the AUs were generated by handcraft. In our proposed method, we utilized the word embedding [56] to generate the node matrix, which can be updated by learning in backpropagation. As shown in Table 3, the results of the experiment show that our method is effective.

Table 1: Experiments with the selection of different models.

Methods	Accuracy
Etran+AUGCN+AUFusion	73.20%
DConv+AUGCN+AUFusion	76.40%
DConv+ETran	78.80%
DConv+ETran+AUGCN+AUFsuion	**80.80%**
Graph-tcn [17]	73.60%

Table 2: Experiments with the main parameters of ETran and DConv.

d_k, d_v	n_h	k	Accuracy
16	**8**	**3**	**80.80%**
16	8	5	76.00%
8	16	3	77.20%
8	16	5	79.20%
32	4	3	76.40%
32	4	5	79.20%

Table 3: Experiments to select the method to generate node matrix.

Methods	Accuracy
One-hot	75.20%
Embedding	**80.80%**

Table 4: Experiment on CASME II with 4 classes.

Methods	Accuracy	F1-score
MDMO (2016) [32]	51.00%	41.80%
FDM (2017) [34]	41.70%	29.70%
Im-based CNN (2017) [52]	44.40%	42.80%
Bi-WOOF (2018) [35]	58.90%	61.00%
Hier.STLBP-IP (2018) [31]	63.80%	61.10%
STRCN-A (2020) [3]	56.00%	54.20%
STRCN-G (2020) [3]	80.30%	74.70%
Graph-tcn (2020) [17]	73.60%	\
ours	**80.80%**	**78.71%**

Table 5: Experiment on SAMM with 4 classes

Methods	Accuracy	F1-score
Im-based CNN (2017) [52]	43.60%	42.90%
Bi-WOOF (2018) [35]	59.80%	59.10%
STRCN-A (2020) [3]	54.50%	49.20%
STRCN-G (2020) [3]	78.60%	74.10%
Graph-tcn (2020) [17]	80.50%	76.57%
ours	**82.39%**	**77.35%**

Table 6: Experiment on CASME II+SAMM with 4 classes.

Methods	Accuracy	F1-score
Im-based CNN (2017) [52]	36.50%	\
Bi-WOOF (2018) [35]	45.30%	\
STRCN-A (2020) [3]	49.50%	\
STRCN-G (2020) [3]	62.90%	\
ours	**79.95%**	**74.26%**

Table 7: Experiment on CASME II with 5 classes.

Methods	Accuracy	F1-score
CNN+LSTM (2016) [37]	60.98%	\
Bi-WOOF+Phase (2017) [49]	62.55%	65.00%
MagGA (2018) [39]	63.30%	\
Hier. STLBP-IP (2018) [31]	63.97%	61.25%
Sparse MDMO (2018) [33]	66.95%	69.11%
HIGO+Mag (2018) [50]	67.21%	\
DiSTLBP-RIP (2019) [30]	64.78%	\
ME-Booster (2019) [51]	70.85%	\
SSSN (2019) [8]	71.19%	71.51%
DSSN (2019) [8]	70.78%	72.97%
TSCNN (2019) [57]	**80.97%**	**80.70%**
Graph-tcn (2020) [17]	73.98%	72.46%
ours	74.27%	70.47%

Table 8: Experiment on SAMM with 5 classes

Methods	Accuracy	F1-score
SSSN (2019) [8]	56.62%	45.13%
DSSN (2019) [8]	57.35%	46.44%
TSCNN (2019) [57]	71.76%	69.42%
Graph-tcn (2020) [17]	**75.00%**	69.85%
ours	74.26%	**70.45%**

Through ablative analysis, we obtained the best optimized configuration of network parameters and structure on the CASME II with 4 classes. Then, our proposed method is compared with the existing methods of CASME II and SAMM with 4 classes. As table 4, 5, and 6 show, MDMO [32], FDM [34], Bi-WOOF [35], and Hier. STLBP-IP [31] are the handicraft features based methods. Im-based CNN [52], Graph-tcn [17], STRCN-A [3], and STRCN-G [3] are the deep learning-based methods. Our proposed method has the best recognition effect on CASME II after optimized for specific network parameters and structure. At the same time, it also has the best results on SAMM and composite dataset (CASME II +SAMM).

5. Further experimental results

Table 9: Experiment on CDE with 3 Classes

Methods	Full		SMIC		CASME II		SAMM	
	UF1	UAR	UF1	UAR	UF1	UAR	UF1	UAR
LBP-TOP (2007) [23]	0.5882	0.5785	0.2000	0.5280	0.7026	0.7429	0.3954	0.4102
Bi-WOOF (2018) [35]	0.6296	0.6227	0.5727	0.5829	0.7805	0.8026	0.5211	0.5139
OFF-ApexNet (2019) [5]	0.7196	0.7096	0.6817	0.6695	0.8764	0.8681	0.5409	0.5392
CapsuleNet [15] (2019)	0.6520	0.6506	0.5820	0.5877	0.7068	0.7018	0.6209	0.5989
Dual-Inception (2019) [4]	0.7322	0.7278	0.6645	0.6726	0.8621	0.8560	0.5868	0.5663
STST-Net (2019) [1]	0.7353	0.7605	0.6801	0.7013	0.8382	0.8686	0.6588	0.6810
EMR (2019) [7]	0.7885	0.7824	**0.7461**	**0.7530**	0.8293	0.8209	**0.7754**	0.7152
ours	**0.7914**	**0.7933**	0.7192	0.7215	**0.8798**	**0.8710**	0.7751	**0.7890**

In order to further verify the generalization ability of our proposed method, we utilized the best configuration of network parameters and structure configuration in ablation analysis to directly carry out the 5-classification experiment without any adjustment. In the experiments, the leave-one-subject-out (LOSO) protocol was also used to evaluate our proposed method. Accuracy and F1-score were used to compute and evaluate the results. The comparison results with the existing 5-classification methods are shown in the table 7 and 8. Sparse MDMO [33], Bi-WOOF+Phase, HIGO+Mag [50], DiSTLBP-RIP [30], ME-Booster [51], and Hier. STLBP-IP [31] are the handicraft features based methods. MagGA [39], Im-based CNN [52], SSSN [8], DSSN [8], TSCNN [57], Graph-tcn [17], STRCN-A [3], and STRCN-G [3] are the deep learning-based methods. Among them, accuracy of our proposed method is only worse than TSCNN [57] on CASME II, but better than TSCNN [57] on SAMM. The accuracy of our proposed method on SAMM is very close to Graph-tcn [17], while better than Graph-tcn [17] on CASME II. As we can see, [17] [57] and our method have respective excellent performance on 5-classification. It is noticed that their accuracies in tables are all directly from their papers, which is in best settings, while our proposed method performs much the same with the existing optimal results [17] and [57] without adjusting network parameters and structure for specific tasks.

To prove our method further, we utilized the CDE of 2019 MEGC [55] to make a more robust experimental evaluation. The UF1 and UAR which used in MEGC 2019 [55] are utilized to evaluate the results. In Equation (9) and (10), the UF1 is determined by averaging F1-scores of the per-class c (of C classes). In Equation (11), n_C is the number of samples of the c-th class.

$$F1_C = \frac{2TP_C}{2TP_C + FP_C + FN_C} \tag{9}$$

$$UF1 = \frac{1}{C}\sum_{C} F1_C \tag{10}$$

$$UAR = \frac{1}{C}\sum_{C} \frac{TP_C}{n_C} \tag{11}$$

Though the annotation of the AUs is not provided in SMIC, we just use the AUs information form CAMSE II and SAMM, which maybe make our proposed method doesn’t perform at its best. As the table 9 shows, the UF1 and UAR of our proposed method obviously lag behind the current methods only in the SMIC part. But the overall performance goes beyond the current methods. It can be concluded that our proposed method is also valid on CDE even if there is no AU annotation provided on SMIC.

6. Conclusion

In this paper, we proposed a novel pipeline to learn a facial graph representation, which includes node learning and edge learning. Node learning can avoid the spatial loss inside each patch and extract features inside each patch. Edge learning can extract the relationship between patches based on multi-head self-attention. Besides, we introduce the AUs information into the facial graph representation by word embedding and GCN. The results of the experiment prove that our method is feasible and effective.

Acknowledgement

This work was supported in part by the Fundamental Research Funds for the Central Universities (XDJK2020C016).

References

[1] S. Liong, Y. S. Gan, J. See, H. Khor and Y. Huang. Shallow Triple Stream Three-dimensional CNN (STSTNet) for Micro-expression Recognition. 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-5.

[2] P. Ekman, Microexpression Training Tool (METT). San Francisco, CA, USA: University California, 2002.

[3] Z. Xia, X. Hong, X. Gao, X. Feng and G. Zhao. Spatiotemporal Recurrent Convolutional Networks for Recognizing Spontaneous Micro-Expressions. IEEE Transactions on Multimedia, vol. 22, no. 3, pp. 626-640, March 2020.

[4] L. Zhou, Q. Mao and L. Xue. Dual-Inception Network for Cross-Database Micro-Expression Recognition. 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-5.

[5] Gan, Y. S., Liong, S. T., Yau, W. C., Huang, Y. C., & Tan, L. K. Off-apexnet on micro-expression recognition system. Signal Processing: Image Communication, 74, 129-139, 2019.

[6] M. Peng, C. Wang, T. Bi, Y. Shi, X. Zhou and T. Chen. A Novel Apex-Time Network for Cross-Dataset Micro-Expression Recognition. 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), Cambridge, United Kingdom, 2019, pp. 1-6.

[7] Y. Liu, H. Du, L. Zheng and T. Gedeon. A Neural Micro-Expression Recognizer. 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-4.

[8] H. Khor, J. See, S. Liong, R. C. W. Phan and W. Lin. Dual-stream Shallow Networks for Facial Micro-expression Recognition. 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 2019, pp. 36-40

[9] H. Khor, J. See, R. C. W. Phan and W. Lin. Enriched Long-Term Recurrent Convolutional Network for Facial Micro-Expression Recognition. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, 2018, pp. 667-674.

[10] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner. Gradient-based learning applied to document recognition. in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. pp. 1097-1105. 2012.

[12] Hochreiter, Sepp and Schmidhuber, Jurgen. Long short-term memory. Neural Computation, 9(8), 1997.

[13] Kipf, T. N., & Welling, M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. 2016.

[14] M. Peng, Z. Wu, Z. Zhang and T. Chen. From Macro to Micro Expression Recognition: Deep Learning on Small Datasets Using Transfer Learning. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, 2018, pp. 657-661.

[15] N. V. Quang, J. Chun and T. Tokuyama. CapsuleNet for Micro-Expression Recognition. 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-7.

[16] B. Sun, S. Cao, D. Li, J. He and L. Yu. Dynamic Micro-Expression Recognition Using Knowledge Distillation. in IEEE Transactions on Affective Computing. 2020

[17] Lei, L., Li, J., Chen, T., & Li, S. A Novel Graph-TCN with a Graph Structured Representation for Micro-expression Recognition. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 2237-2245). 2020.

[18] Oh, T. H., Jaroensri, R., Kim, C., Elgharib, M., Durand, F. E., Freeman, W. T., & Matusik, W. Learning-based video motion magnification. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 633-648). 2018.

[19] B. Martinez, M. F. Valstar, B. Jiang and M. Pantic. Automatic Analysis of Facial Actions: A Survey. in IEEE Transactions on Affective Computing, vol. 10, no. 3, pp. 325-347, 1 July-Sept. 2019.

[20] L. Lo, H. -X. Xie, H. -H. Shuai and W. -H. Cheng. MER-GCN: Micro-Expression Recognition Based on Relation Modeling with Graph Convolutional Networks. 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Shenzhen, Guangdong, China, 2020, pp. 79-84.

[21] Hinton, G., Vinyals, O., & Dean, J. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. 2015.

[22] T. Pfister, Xiaobai Li, G. Zhao and M. Pietikäinen. Recognising spontaneous facial micro-expressions. 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 1449-1456.

[23] G. Zhao and M. Pietikainen. Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 915-928, June 2007.

[24] X. Li, T. Pfister, X. Huang, G. Zhao and M. Pietikäinen. A Spontaneous Micro-expression Database: Inducement, collection and baseline. 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Shanghai, 2013, pp. 1-6.

[25] S. Wang, W. Yan, X. Li, G. Zhao and X. Fu. Micro-expression Recognition Using Dynamic Textures on Tensor Independent Color Space. 2014 22nd International Conference on Pattern Recognition, Stockholm, 2014, pp. 4678-4683.

[26] S. Wang *et al*. Micro-Expression Recognition Using Color Spaces. in IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 6034-6047, Dec. 2015.

[27] Y. Guo, Y. Tian, X. Gao and X. Zhang. Micro-expression recognition based on local binary patterns from three orthogonal planes and nearest neighbor method. 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, 2014, pp. 3473-3479.

[28] Wang, Y., See, J., Phan, R. C. W., & Oh, Y. H. Lbp with six intersection points: Reducing redundant information in lbp-top for micro-expression recognition. In Asian conference on computer vision (pp. 525-537). Springer, Cham. 2014.

[29] X. Huang, S. Wang, G. Zhao and M. Piteikäinen. Facial Micro-Expression Recognition Using Spatiotemporal Local Binary Pattern with Integral Projection. 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, 2015, pp. 1-9.

[30] X. Huang, S. Wang, X. Liu, G. Zhao, X. Feng and M. Pietikäinen. Discriminative Spatiotemporal Local Binary Pattern with Revisited Integral Projection for Spontaneous Facial Micro-Expression Recognition. in IEEE Transactions on Affective Computing, vol. 10, no. 1, pp. 32-47, 1 Jan.-March 2019.

[31] Y. Zong, X. Huang, W. Zheng, Z. Cui and G. Zhao. Learning From Hierarchical Spatiotemporal Descriptors for Micro-Expression Recognition. in IEEE Transactions on Multimedia, vol. 20, no. 11, pp. 3160-3172, Nov. 2018.

[32] Y. Liu, J. Zhang, W. Yan, S. Wang, G. Zhao and X. Fu. A Main Directional Mean Optical Flow Feature for Spontaneous Micro-Expression Recognition. in IEEE Transactions on Affective Computing, vol. 7, no. 4, pp. 299-310, 1 Oct.-Dec. 2016.

[33] Y. Liu, B. Li and Y. Lai. Sparse MDMO: Learning a Discriminative Feature for Spontaneous Micro-Expression Recognition. in IEEE Transactions on Affective Computing.

[34] F. Xu, J. Zhang and J. Z. Wang. Microexpression Identification and Categorization Using a Facial Dynamics Map. in IEEE Transactions on Affective Computing, vol. 8, no. 2, pp. 254-267, 1 April-June 2017.

[35] Liong, S. T., See, J., Wong, K., & Phan, R. C. W. Less is more: Micro-expression recognition from video using apex frame. Signal Processing: Image Communication, 62, 82-92. 2018.

[36] S. L. Happy and A. Routray. Fuzzy Histogram of Optical Flow Orientations for Micro-Expression Recognition. in IEEE Transactions on Affective Computing, vol. 10, no. 3, pp. 394-406, 1 July-Sept. 2019.

[37] Kim, D. H., Baddar, W. J., & Ro, Y. M. Micro-expression recognition with expression-state constrained spatio-temporal feature representations. In Proceedings of the 24th ACM international conference on Multimedia (pp. 382-386). 2016.

[38] Devangini Patel, X. Hong and G. Zhao. Selective deep features for micro-expression recognition. 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, 2016, pp. 2258-2263.

[39] Y. Li, X. Huang and G. Zhao. Can Micro-Expression be Recognized Based on Single Apex Frame?. 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 3094-3098.

[40] Z. Xia, X. Feng, X. Hong and G. Zhao. Spontaneous Facial Micro-expression Recognition via Deep Convolutional Network. 2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA), Xi'an, 2018, pp. 1-6.

[41] S. P. Teja Reddy, S. Teja Karri, S. R. Dubey and S. Mukherjee. Spontaneous Facial Micro-Expression Recognition using 3D Spatiotemporal Convolutional Neural Networks. 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8.

[42] Chen, B., Zhang, Z., Liu, N., Tan, Y., Liu, X., & Chen, T. Spatiotemporal Convolutional Neural Network with Convolutional Block Attention Module for Micro-Expression Recognition. Information, 11(8), 380. 2020.

[43] Wang, C., Peng, M., Bi, T., & Chen, T. Micro-attention for micro-expression recognition. Neurocomputing, 410, 354-362. 2020.

[44] Davison, A. K., Merghani, W., & Yap, M. H. Objective classes for micro-facial expression recognition. Journal of Imaging, 4(10), 119. 2018.

[45] Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258). 2017.

[46] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008). 2017.

[47] Wen-Jing Yan, Xiaobai Li, Su-Jing Wang, Guoying Zhao, Yong-Jin Liu, YuHsin Chen, Xiaolan Fu. CASME II: An Improved Spontaneous MicroExpression Database and the Baseline Evaluation. PLoS ONE. 9, 1 (2014), e86041. 2014.

[48] Adrian K. Davison, Cliff Lansley, Nicholas Costen, Kevin Tan, and Moi Hoon Yap. SAMM: A Spontaneous Micro-Facial Movement Dataset. IEEE Transactions on Affective Computing. 9, 1 (2018), 116-129. 2018.

[49] Sze-Teng Liong and KokSheik Wong. Micro-expression recognition using apex frame with phase information. In Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. 534-537. 2017.

[50] Xiaobai Li, Xiaopeng Hong, Antti Moilanen, Xiaohua Huang, Tomas Pfister, Guoying Zhao, and Matti Pietikäinen. Towards Reading Hidden Emotions: A Comparative Study of Spontaneous Micro-Expression Spotting and Recognition Methods. IEEE Transactions on Affective Computing. 9, 4 (2018), 563-577. 2018.

[51] Wei Peng, Xiaopeng Hong, Yingyue Xu, and Guoying Zhao. A Boost in Revealing Subtle Facial Expressions: A Consolidated Eulerian Framework. In Proceedings of IEEE International Conference on Automatic Face & Gesture Recognition. 1-5. 2019.

[52] M. A. Takalkar and M. Xu. Image based facial micro-expression recognition using deep learning on small datasets. in International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp. 1–7.

[53] L. Zhong, C. Bai, J. Li, T. Chen, S. Li and Y. Liu. A Graph-Structured Representation with BRNN for Static-based Facial Expression Recognition. 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-5.

[54] R. Ekman. What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA, 1997.

[55] J. See, M. H. Yap, J. Li, X. Hong and S. Wang. MEGC 2019 – The Second Facial Micro-Expressions Grand Challenge. 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-5.

[56] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 3111–3119. 2013.

[57] B. Song et al. Recognizing Spontaneous Micro-Expression Using a Three-Stream Convolutional Neural Network. in IEEE Access, vol. 7, pp. 184537-184551, 2019.