Transferring Dual Stochastic Graph Convolutional Network for Facial Micro-expression Recognition

Hui Tang, Li Chai and Wanli Lu

Abstract—Micro-expression recognition has drawn increasing attention due to its wide application in lie detection, criminal detection and psychological consultation. To improve the recognition performance of the small micro-expression data, this paper presents a transferring dual stochastic Graph Convolutional Network (TDSGCN) model. We propose a stochastic graph construction method and dual graph convolutional network to extract more discriminative features from the micro-expression images. We use transfer learning to pre-train SGCNs from macro expression data. Optical flow algorithm is also integrated to extract their temporal features. We fuse both spatial and temporal features to improve the recognition performance. To the best of our knowledge, this is the first attempt to utilize the transferring learning and graph convolutional network in micro-expression recognition task. In addition, to handle the class imbalance problem of dataset, we focus on the design of focal loss function. Through extensive evaluation, our proposed method achieves state-of-the-art performance on SAMM and recently released MMEW benchmarks. Our code will be publicly available accompanying this paper.

Index Terms—Class, IEEEtran, L\LaTeX, paper, style, template, typesetting.

I. INTRODUCTION

Facial expression is an effective and universal way of expressing human emotions and intentions [7]. There are two kinds of facial expression, daily macro-expression and hidden micro-expression. Micro-expressions are genuine and involuntary emotions which people usually attempt to control and conceal under high-stake situations. As the development of artificial intelligence, micro-expression recognition attracts more and more attentions due to its importance in lie detection, criminal detection and psychological consultation [12], [4], [13].

Micro-expression recognition is a complicated and challenging task in the field of computer vision [3], [49], [16]. The available databases are usually small. The biggest dataset consists only 300 samples. Their intensity is low and duration is short. Instead of a single image, image sequences are necessary in the micro-expression recognition task.

One of the most efficient ways is to extract more discriminative features from image sequences. Traditional methods utilize handcrafted features to deal with the micro-expression recognition problem, including local binary patterns (LBP) [37], LBP on three orthogonal planes (LBP-TOP) [53], and histogram of oriented gradients (HOG) [8]. These methods are restricted by the limited training samples. Their performance need to be further improved. With the success of deep learning in many computer vision applications, deep features also obtain state-of-the-art predictive performance in micro-expression recognition. Recently, Ben et al. [3] gave a comprehensive and systematic survey of the major challenges and developments in micro-expression recognition area. They also presented a new dataset with more samples called micro- and macro expression warehouse (MMEW). State-of-the-art method for the micro-expression recognition is the transferring long-term convolutional neural network (TLCNN) [42], which extracted both spatial and temporal features by feeding the CNN features to Long Short Term Memory (LSTM). The recognition accuracy on MMEW is 69.4% and SAMM is 73.5%. This detection and recognition is improved but still limited by the lack of large datasets.

We observe that most of existing methods extract facial features on the basis of pixels. However, face pixels may not reveal the distinct feature of micro-expression images since its salient feature is low intensity and transit.

Inspired by the outstanding performance of Graph Convolutional Network (GCN) on both grid and non-grid data [50], [11], [26], [6], we propose a transferring stochastic dual GCN (TSDGCN) model to improve the performance of micro-expression recognition. Transfer learning is efficient to tackle the small data problem. We are the first to integrate the GCN and transfer learning to deal with the small data problem in micro-expression recognition. To enhance the discriminative learning, we present a novel graph construction method. We propose a dual graph convolutional network (DGCN), based on which two feature extractors are employed. We observe that previous GCN work on grid data often construct graph by their nearest neighbors. Nodes have edges among the nearest neighbor nodes. This results in redundant links and can be computationally expensive. We propose a stochastic graph construction method to extract geometric feature between pixels named $p + q$ graph construction. Nodes have edges among the $p$ nearest neighbors and $q$ neighbors stochastically selected. There is a simple example of $2+2$ graph construction shown in Figure 1. With the same number of neighbors, the adjacency matrix of our method may encode broader and more distinctive information, which benefits from the technique of stochastic sampling.

The class imbalance problem is a common problem in the existing micro-expression datasets. Class imbalance seriously decreases the recognition performance. Instead of cross-entropy, this paper attempts to design the focal loss function to deal with the problem of class imbalance.

To conclude, the main contributions of this paper are as follows:

This paper was supported by the National Natural Science Foundation of China under grants (61800339)
Previous work

Image

4NN graph construction

Adjacency matrix

Ours

Image

2+2 graph construction

Adjacency matrix

Fig. 1. (Top): Previous work often construct graph by their nearest neighbors (NN) to handle the image data. Nodes have edges among the nearest neighbors. Each node has 4 edges in the 4-NN graph. (Bottom): Our work proposes a stochastic \( p + q \) graph construction to improve the discriminative learning. Nodes have edges among the \( p \) nearest neighbors and \( q \) stochastic neighbors. Each node has 2 nearest neighbors and 2 stochastic neighbors in the \( 2+2 \) graph. The potential nodes for stochastic edges are linked in dotted line and represented in blue 1’s in the adjacency matrix.

- We propose a novel stochastic GCN model to improve the performance of micro-expression recognition. SGCN can extract both local and global features with less computational complexity.
- To the best of our knowledge, we are the first to propose a dual GCN architecture to extract more representative features.
- Our proposed method can outperform the state of the art results on four benchmarks.

II. RELATED WORK

Micro-expression recognition Traditional classification methods, like support vector machine (SVM) [43], extreme learning machine (ELM) [44] and K nearest neighbor [41] et al., rely on artificially designed features, which are simple to use but with poor performance. Deep learning methods can automatically extract an optimal feature representation [38], [39], [23]. Khor et al. [23] presented an Enriched Long-term Recurrent Convolutonal Network (ELRCN) to enrich the subtle movements. However, all existing micro-expression databases are small. Deep learning with small data may not achieve good performance. Transfer learning shows great potential to deal with the small data problem, which uses knowledge from a related domain (in which large datasets are available) [37], [55], [56]. Most of the existing transfer learning methods are CNN-based architectures. Just like CNN can capture the most significant information within pixels in images, a graph-based learning algorithm learns the relation between each node from the data described in the form of graph. To the best of our knowledge, this is the first work that integrates GCN and transfer learning to handle the micro-expression recognition problem.

Graph convolutional network Recently, it has been shown that GCN approaches are powerful for extracting discriminative geometric features in both grid and non-grid data application [47]. There are two mainstreams to define a GCN, the spectral methods [5], [11], [25] and the spatial methods [34], [1], [36], [15]. The spectral domain models the representation in the Fourier domain based on eigen-decomposition. The spatial method directly implements operators on the graph node and its neighbors. Up to now, there are few published GCN related work to deal with micro-expression problem. Lo et al. [33] used 3D ConvNets to extract actions units (AUs) features and applied GCN layers to discover the relations between AU nodes for micro-expression recognition. Xie et al. [49] exploited AUs relational information and proposed AU-assisted Graph Attention Convolutional Network for micro-expression recognition. We propose a novel graph construction method by stochastic sampling and we employ two GCN feature extractors to graph data.

Optical flow The idea of optical flow was first introduced by Horn et al. [18] to describe the movement of brightness patterns in an image. The basic concept is to find the distance of an identical object in different frames. By utilizing the pixel-wise difference between consecutive frames in a video can thus be obtained [52]. Owing to the fact that optical flow could capture temporal patterns between consecutive frames, one of the most employed architecture is to combine optical flow feature with CNN to further recognize spatial patterns [48], [14]. On one side, we integrate the optical flow feature and SDGCN to extract temporal features. On the other side, optical flow information is also utilized in the data pre-processing to select meaningful frames in the entire micro-expression sequences.

Focal loss Class imbalance is a common problem in . Li and Deng [27], [28] used local preservation loss to maintain the locality of each examples, making the local neighborhood in each class as compact as possible. Mollahosseini et al. [35] and Ji et al. [22] respectively used the weighted cross entropy loss function to improve the problem of class imbalance by weighting the loss function of each class. The imbalance problem in the field of expression is still difficult to solve. Lin et al. [30] proposed the definition of focal loss to handle the class imbalance problem of binary classification in the target detection scene of one stage. It down-weights the loss assigned to well-classified examples. This paper introduces the focal loss to deal with the problem of class imbalance in micro-expression recognition.

III. APPROACH

A. Overview

As shown in Figure 2, our proposed method consists of two parts. The upper part is the source network by macro-expression images and the bottom part is the target network by micro-expression images.

In source network, the input macro images are normalized in the pre-processing step. Macro images are fed into stochastic graph convolutional network. After each layer of stochastic graph convolution, an activation layer and pooling layer are appended. The classification is accomplished by computing the loss function at the fully connected layer. The SGCNs trained by source network are transferred to the target network.
In target network, there are two branches including spatial and temporal branch. The inputs of spatial and temporal branch are micro pixel images and optical flow images, respectively. In spatial branch, we use optical flow method to realize frame normalization to select meaningful frames in the entire micro-expression sequence. To improve the recognition performance, we adopt two transferred SGCNs (SGCN1 and SGCN2) to extract features, named dual SGCN (DSGCN). In temporal branch, we also utilize transferred SGCN1 and SGCN2 to extract features. Features from spatial and temporal branch are integrated to represent the discriminative features. The integrated features are fed into a fully connected layers.

B. Stochastic Graph Construction

We first give the detail of how our stochastic graph construction works. Given an image $I$, we convert it into an undirected and connected graph $G = (V, E, A)$. $V$ is a finite set of $|V| = N$ vertices, where each node is corresponding to a pixel. $E$ is a finite set of edges, which are decided by their neighbors. $x \in \mathbb{R}^N$ represents the signal defined on the graph where $x_i$ is the value of pixel at the $i$th node.

Our work presents a novel $p+q$ graph construction method. Each vertex has $p$ nearest neighbors and $q$ neighbors stochastically selected. The adjacency matrix $A$ associated to $G$ can be constructed. For node $i \in \{1 \cdots N\}$, we design a distance threshold $T$ to select its potential neighbors. We calculate the Euclidean distance $d_{ij}$ between node $i$ and node $j$, where $z_i$ is node $i$’s 2D coordinate. If $d_{ij}$ is not more than threshold $T$, node $j$ is a neighbor of node $i$. Otherwise, there is no edge between node $i$ and $j$. The weights between node $i$ and node $j$ are calculated in (1) and the number of node $i$’s neighbors is denoted by $|N_i|$.

$$a_{ij} = \begin{cases} \exp(-\frac{||z_i - z_j||^2}{\sigma^2}), & \text{if } d(z_i, z_j) \leq T \\ 0, & \text{otherwise.} \end{cases}
$$

where $z_i$ is the 2D coordinate of pixel $i$ and the $\sigma$ is the average distance between each vertex and the vertex that is the farthest from it.

Algorithm 1 gives the details of our adjacent matrix construction. For node $i$, we extract its potential neighbors $N_i$ by (1) with given threshold $T$. We choose $p$ nearest neighbors as fixed neighbors. We compute the weights of $p$ nearest neighbors by Gaussian kernel function. We stochastically choose $q$ neighbors among the left. The weights of remaining neighbors are zero.

**Algorithm 1** Adjacency matrix of stochastic $p + q$ graph

**Input:** An image with $N$ vertices and node $i$’s 2D coordinate denoted by $z_i$, $i = 1 \cdots N$. The number of nearest neighbors denoted by $p$ and stochastic neighbors denoted by $q$. The distance threshold denoted by $T$.

**Output:** Adjacency matrix $A$.

for node $i = 1 ; i \leq N ; i++$ do
Step 1: Calculating the Euclidean distance $d(z_i, z_j)$ between node $i$ and node $j$. With threshold $T$, we can choose node $i$’s potential neighbors $N_i = \{i_1, i_2, \cdots, i_{|N_i|}\}$. The weights between them are calculated by (1);
Step 2: Choosing $p$ nearest neighbors as fixed neighbors and stochastically choosing $q$ neighbors from the left, denoted these $p+q$ neighbors by $S$, where $S \subset N_i$;
Step 3: The weights of remaining neighbors are 0, where $j \notin S$ and $j \in N_i / S$.
end for
Figure 3 gives an example of $p = 8$ and $q = 2$ graph construction for the reference node with green color. We design the distance threshold $T = 2\sqrt{5}$. There are 24 potential neighbors for the middle reference node. We choose 8 nearest neighbors as fixed neighbors. We stochastically select 2 neighbors from the left blue vertices. The remaining ones are unlinked.

Each pixel is a vertex and the signal is the value of pixel

Each node has $p+q$ edges according to the distance between pixels

Fig. 3. A simple example of $8 + 2$ graph construction. (Left): Node construction. (Right): Edge construction: the reference node (green) has 8 nearest neighbors with orange color and 2 neighbors stochastically selected from the potential nodes with blue color.

C. Dual Network and Transfer Learning

We propose a dual network to integrate two different SGCNs (SGCN1 and SGCN2). SGCN1 uses the $p_1 + q_1$ graph construction and SGCN2 uses $p_2 + q_2$ graph construction. We use transfer learning to achieve feature extraction in our dual network. We use macro images to train the SGCN in source network. We use transfer learning to achieve feature extraction in our dual network.

As illustrated in Section III-B, we construct different adjacency matrices $A_1$ and $A_2$ corresponding to SGCN1 and SGCN2. The corresponding Laplacian matrices $L_1$ and $L_2$ are computed as

$$L_1 = I_N - D_1^{-\frac{1}{2}} A_1 D_1^{-\frac{1}{2}},$$

$$L_2 = I_N - D_2^{-\frac{1}{2}} A_2 D_2^{-\frac{1}{2}},$$

where $D_1$ and $D_2$ are the diagonal degree matrices with $D_1(i, i) = \sum_j A_1(i, j)$ and $D_2(i, i) = \sum_j A_2(i, j)$. $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix.

There are $L$ stochastic graph convolutional layers in our dual network. We utilize the truncated Chebyshev polynomials to achieve the graph convolutional filtering [11]. For layer $l$, the output feature is denoted as $X^l \in \mathbb{R}^{M \times d_l}$, where $M$ is the number of input samples and $d_l$ is the dimension of output features. The convolutional coefficients $w^l \in \mathbb{R}^{K^l \times d_l}$ are trained by macro images in source network, where $K^l$ denotes the truncated order of Chebyshev polynomials in $l$-th layer. The dual stochastic graph convolution for the $l$-th layer ($1 \leq l \leq L$) is implemented

$$X^l_1 = \sum_{k=0}^{K^l-1} w^l_1 T_k(L_1) X^{l-1}_1,$$

$$X^l_2 = \sum_{k=0}^{K^l-1} w^l_2 T_k(L_2) X^{l-1}_2,$$

where $w^l_1$ and $w^l_2$ are the trained weights from the source network by macro images. $L_1 = \frac{1}{\lambda_{max}}I_N - \lambda_{max}^{-1} I_N$ and $\lambda_{max}$ denotes the largest eigenvalues of $L_1$. $L_2 = \frac{2}{\lambda_{max}}I_N - I_N$ and $\lambda_{max}$ denotes the largest eigenvalues of $L_2$. The interval of the eigenvalues in $L$ is $[-1, 1]$.

We feed the outputs of $L$-th layer $X^l_1 \in \mathbb{R}^{M \times d_{l_1}}$ and $X^l_2 \in \mathbb{R}^{M \times d_{l_2}}$ into the trained fully connected layer. The outputs of fully connected layer are denoted as $X^{fc}_1 \in \mathbb{R}^{M \times d_{fc_1}}$ and $X^{fc}_2 \in \mathbb{R}^{M \times d_{fc_2}}$, where $d_{fc_1}$ and $d_{fc_2}$ are the dimensions of fully connected layer in SGCN1 and SGCN2, respectively.

After concatenation, the fused features of dual network are as follows

$$X^D = [X^{fc}_1 \; X^{fc}_2] \in \mathbb{R}^{M \times (d_{fc_1} + d_{fc_2})}. \quad (4)$$

D. Features Extraction of Spatial and Temporal Branches

In our proposed model, the spatial branch is designed to capture the spatial relationships between pixels across different frames. As we have mentioned in Section I, optical flow images can provide temporal information between different frames. The spatial and temporal features of micro-expressions at different expression-states are encoded using DSGCNs.

The feature extraction in the spatial branch is shown in Figure 4. Specifically, given a video, the spatial branch of $m$-th frame is represented as $G_m = (V_m, E_m, A_m)$, where $m = 1 \cdots M_s$ and $M_s$ is the number of image sequences selected from the video. As shown in Section III-B, there are $N_s$ nodes in each image and the signal defined on the image is $x \in \mathbb{R}^{N_s}$. The initial input features are represented by $X^{in}_m = (x_1, x_2, \cdots, x_{M_s}) \in \mathbb{R}^{M_s \times N_s}$, where $x_m$ is the signal of $m$-th image. We feed the initial features $X^{in}_s$ into the transferred DSGCN. By Eq. (2), (3) and (4), we obtain the output features of spatial branch $X^D_s \in \mathbb{R}^{M_s \times (d_{fc_1} + d_{fc_2})}$.

Fig. 4. Illustration of feature extraction in the spatial branch

The feature extraction in temporal branch is shown in Figure 5. We compute the optical flow features between sequences. We select $M_t$ samples from the optical flow sequences. The initial input features of temporal branch are represented by $X^{in}_t \in \mathbb{R}^{M_t \times N_t}$, where $M_t$ is the number of pixels of optical flow image. We also feed the initial features $X^{in}_t$ into the transferred DSGCN. By Eq. (2), (3) and (4), we obtain the output features of temporal branch $X^{D}_t \in \mathbb{R}^{M_t \times (d_{fc_1} + d_{fc_2})}$. 
We concatenate the features $X^s$, $X^t$, and $X^d$ obtained by spatial and temporal branches,\[X^{in} = [X^s, X^t, X^d] \in \mathbb{R}^{(M_s + M_t) \times (d_{fc1} + d_{fc3})}\tag{5}\]

We train a fully connected layer by the fused features $X^{in}$ in target network. The output feature of the target network is denoted by $X^{out} \in \mathbb{R}^{(M_s + M_t) \times d_{fc3}}$, where $d_{fc3}$ is the dimension of fully connected layer appended by the spatial and temporal branch.

### E. Focal Loss Function

Lin et al. [30] proposed the definition of focal loss to handle the class imbalance in scene detection. The focal loss is defined as:

$$FL(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t)$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases}$$

where $\alpha$ is a weighting factor to balance the importance for positive and negative example, $\gamma$ is the focusing parameters to balance the difference between easy and hard examples. $y \in \{0, 1\}$ is the ground-truth label and $p \in [0, 1]$ is the model’s estimated probability for the class with label $y = 1$.

### IV. Experiments

We set up a TensorFlow 1.14.0 environment on a Windows 10 computer with an i7-9700k processor and NVIDIA GeForce RTX2070 graphics card.

| Layer        | Parameters |
|--------------|------------|
| SGCN layer 1 | $K^1 = 9, d^1 = 32, \text{ReLU}, s_1 = 2$ |
| SGCN layer 2 | $K^2 = 9, d^2 = 32, \text{ReLU}, s_2 = 2$ |
| SGCN layer 3 | $K^3 = 6, d^3 = 64, \text{ReLU}, s_3 = 1$ |
| SGCN layer 4 | $K^4 = 6, d^4 = 64, \text{ReLU}, s_4 = 1$ |
| SGCN layer 5 | $K^5 = 4, d^5 = 128, \text{ReLU}, s_5 = 1$ |
| SGCN layer 6 | $K^6 = 4, d^6 = 128, \text{ReLU}, s_6 = 1$ |
| Fully connected | $d_{fc} = 512, \text{ReLU}, \text{Dropout}$ |

**TABLE I**

The network configuration of SGCN

### A. Datasets

To evaluate our proposed approach, we conduct experiments using two databases: SAMM (the most commonly used) and MMEW (a newly published).

**MMEW** As a newly published dataset, there are 300 samples in micro-and-macro expression warehouse (MMEW) [3], including happiness (36), anger (8), surprise (89), disgust (72), fear (16), sadness (13) and others (102), which is the biggest published micro-expression dataset. There are 36 participants including 9 females and 27 males, whose average age is 22.35. 900 macro-expression samples with the same category by the same group of participants are also provided.

**SAMM** The Spontaneous Actions and Micro-Movements (SAMM) contains 159 samples (image sequences containing spontaneous micro-expressions) recorded by a high-speed camera in a well controlled laboratory environment [9], [10], [51]. There are 32 participants with a mean age of 33.24 years, and an even male-female gender split. Originally intended for investigating micro-facial movements, the SAMM was induced based on the 7 basic emotions including happiness (24), surprise (13), anger (20), disgust (8), sadness (3), fear (7), others (84).

### B. Experimental Settings

As shown in [3], the performance results of published algorithms on micro-expression recognition are reported across different experimental settings. Ben et al. conduct a fair comparison on datasets SAMM and MMEW, which are the most suitable datasets for recognition evaluation. We design the same experimental settings as [3]. In the MMEW dataset, 226 samples from 5 classes (i.e., happiness, surprise, disgust, fear, sadness) were used; in the SAMM dataset, 72 samples from 5 classes (i.e., happiness, surprise, anger, disgust, fear) were used.

The network configuration of SGCN is designed in Table I. There are six stochastic convolutional layers. $K^l$ is the order of the ChebNet filter in the $l$-th layer, $d^l$ denotes the feature number of undirected graphs output from the $l$-th layer to the next layer, $s_l$ represents the number of times of graph coarsening in the $l$-th layer, $d_{fc}$ represents the size of fully connected layer.

The networks are trained using adaptive epochs or early stopping with a maximum set to 100 epochs. Basically, the training for each fold will stop when the loss score stops improving. We use Adaptive Moment Estimation (ADAM) as the optimizer, with a learning rate of $10^{-5}$ and decay of $10^{-6}$.

### C. Comparison to State-of-the-art Methods

To validate the effectiveness of our proposed method on the micro-expression recognition problem, we compare our proposed method with several recent state-of-the-art methods on MMEW and SAMM. All the published results in [3] are also kept for the convenience of comparison. Table II summarizes the comparison results.

It is obvious that all deep learning methods perform better than those utilizing handcrafted features. It can be seen that our
proposed method achieves the best recognition performance (72.7% on MMEW and 75.0% on SAMM), which outperforms state-of-the-art method TLCNN (the best recognition performance is 69.4% on MMEW and 73.5% on SAMM) [42]. The success of TLCNN [42] demonstrates that the knowledge of macro-expressions is useful for micro-expression recognition under the CNN architecture. This paper explores the transferring macro-expression knowledge to assist micro-expression recognition based on the GCN architecture. The training and testing sets of MMEW and SAMM were set as [3]. Table III lists the data source and accuracy of the pre-training, fine-tuning and testing results for each experiment. Compared to existing methods, our proposed method considers the stochastic geometric features of different pixels, which can provide more discriminative and robust information. These experimental results validate the superiority of our method.

Results on SAMM

We also design efficient parameters of focal loss on SAMM. The results of different weight factor $\alpha$ and focus factor $\gamma$ are presented in Table V. The best result 75.00% occurs with the parameters $\alpha = 1$ and $\gamma = 1$. The confusion matrix in Figure 7 indicates that our method can totally recognize “Happiness”, “Surprise” and ”Angry”. It is difficult to train ”Disgust” and ”Fear” since their small samples.

D. Ablation Study

For further analysis, we perform an extensive ablation study by removing certain portions of our proposed TSDGCN to see how that affects performance. This was carried out using the databases MMEW and SAMM. To verify the impact of DGCN, we train both single and dual GCN for the spatial branch and temporal branch under the same setting as Table III. We select 10 sets of typical graph architectures containing 4 single GCN and 6 DGCN. For the convenience, we utilize the cross entropy loss here. The extensive experimental results on both MMEW and SAMM are reported in Table VI. It is obvious that most DGCNs obtain higher accuracies of pre-training, fine-tuning and testing than single GCNs. Comparing the testing accuracies, the best graph architecture is 8+2 and 4+0 on both MMEW and SAMM, which is marked in bold. We also observe that all the spatial branches have better recognition performance than temporal ones.

To verify the impact of spatial branch and the temporal branch, we train the “Spatial branch” and “Temporal branch” under the same experiment setting and graph architecture. With the aid of Table VI, we discuss the graph architecture 8+2 and 4+0. The experimental results on both MMEW and SAMM are reported in Table VII. On MMEW, the recognition accuracy of “Spatial + CE” is 65.9% and the recognition accuracy of

| Methods                                      | Recognition rate (%) | MMEW | SAMM |
|----------------------------------------------|----------------------|------|------|
| FDM [50]                                     | 34.6                 | 34.1 |
| ResNet10 [17]                                 | 36.6                 | 39.3 |
| Handcrafted features + deep learning [19]     | 36.6                 | 47.1 |
| LBP+TOP [53]                                  | 38.9                 | 37.0 |
| Selective deep features [38]                  | 39.0                 | 42.9 |
| ELRCN [23]                                    | 41.5                 | 46.2 |
| DCP-TOP [2]                                   | 42.5                 | 36.8 |
| ESCSTF [24]                                   | 42.7                 | 46.9 |
| LHPW-TOP [2]                                  | 43.2                 | 41.7 |
| LBP-MOP [45]                                  | 43.9                 | 35.3 |
| LBP-SIP [46]                                  | 43.9                 | 37.4 |
| DiSTLBP-RIP [20]                              | 44.0                 | 46.2 |
| RHWP-TOP [2]                                  | 45.9                 | 38.1 |
| STLBP-IP [21]                                 | 46.6                 | 42.9 |
| ApexME [29]                                   | 48.8                 | 50.0 |
| Transfer Learning [40]                        | 52.4                 | 55.9 |
| Multi-task mid-level feature learning [16]    | 54.2                 | 55.0 |
| KGS [54]                                      | 56.9                 | 48.6 |
| Sparse MDMO [31]                              | 60.0                 | 52.9 |
| MDMO [32]                                     | 65.7                 | 50.0 |
| DTSCNN [39]                                   | 65.9                 | 69.2 |
| TLCNN [42]                                    | 69.4                 | 73.5 |
| Ours                                          | 72.7                 | 75.0 |

**TABLE II**

RECOGNITION RATES (%) OF MICRO-EXPRESSION USING THE STATE-OF-THE-ART METHODS ON MMEW AND SAMM. THE BEST RECORD OF EACH DATASET IS MARKED IN BOLD.

Fig. 6. Confusion matrix of recognizing 5 expressions on MMEW dataset

Fig. 7. Confusion matrix of recognizing 5 expressions on SAMM dataset

Results on MMEW

We design efficient parameters of focal loss on MMEW. The results are presented in Table IV. The best result is 72.73% occurs with the parameters $\alpha = 1.5$ and $\gamma = 0.2$. We also present confusion matrices in Figure 6, from which we observe that in MMEW, the “disgust” and “surprise” samples can be highly recognized on MMEW, instead the “fear” samples are difficult to train. Because the number of fear samples of MMEW is only 16, which is too small to train a good classifier.

Results on SAMM

We also design efficient parameters of focal loss on SAMM. The results are presented in Table V. The best result 75.00% occurs with the parameters $\alpha = 1$ and $\gamma = 1$. The confusion matrix in Figure 7 indicates that our method can totally recognize “Happiness”, “Surprise” and ”Angry”. It is difficult to train ”Disgust” and ”Fear” since their small samples.

D. Ablation Study

For further analysis, we perform an extensive ablation study by removing certain portions of our proposed TSDGCN to see how that affects performance. This was carried out using the databases MMEW and SAMM.

To verify the impact of DGCN, we train both single and dual GCN for the spatial branch and temporal branch under the same setting as Table III. We select 10 sets of typical graph architectures containing 4 single GCN and 6 DGCN. For the convenience, we utilize the cross entropy loss here. The extensive experimental results on both MMEW and SAMM are reported in Table VI. It is obvious that most DGCNs obtain higher accuracies of pre-training, fine-tuning and testing than single GCNs. Comparing the testing accuracies, the best graph architecture is 8+2 and 4+0 on both MMEW and SAMM, which is marked in bold. We also observe that all the spatial branches have better recognition performance than temporal ones.

To verify the impact of spatial branch and the temporal branch, we train the “Spatial branch” and “Temporal branch” under the same experiment setting and graph architecture. With the aid of Table VI, we discuss the graph architecture 8+2 and 4+0. The experimental results on both MMEW and SAMM are reported in Table VII. On MMEW, the recognition accuracy of “Spatial + CE” is 65.9% and the recognition accuracy of

---

**TABLE III**

DATA SOURCE AND ACCURACY OF THE PRE-TRAINING, FINE-TUNING AND TESTING RESULTS FOR EACH EXPERIMENT.
“Temporal + CE” is only 45.7%. The performance of “Spatial + Temporal branch + CE” varies with 8+2 and 4+0 graph architecture is the better than “Spatial + CE” and “Temporal branch + CE”, which shows that “Temporal branch” compensates the performance of “Spatial branch”. On SAMM, the recognition accuracy of “Spatial + CE” is 68.8% and the recognition accuracy of “Temporal + CE” is only 36.9%. The performance of “Spatial + Temporal branch + CE” is the same with “Spatial+ CE”, which means “Temporal branch” doesn’t improve the performance.

Table 4 shows mean and variance of optical flow sequences on MMEW and SAMM. It is obvious that optical flow sequences of MMEW vary among samples and optical flow sequences of SAMM change a few. It means temporal branch may not contain useful information, which is why temporal branch doesn’t improve the performance in SAMM.

To verify the impact of focal loss, we compare the recognition performance between cross entropy and focal loss under the same architecture settings. “Spatial+Temporal branch + FL” achieves the best performance on both MMEW and SAMM with the graph architecture 8+2 and 4+0, which shows the advantage of focal loss function.

### References

1. James Atwood and Don Towsley. Diffusion-convolutional neural network. *Adv. Neural Inform. Process. Syst.*, pages 2001–2009, 2016.
2. Xianye Ben, Xitong Jia, Rui Yan, Xin Zhang, and Weixiao Meng. Learning effective binary descriptors for micro-expression recognition transferred by macro-information. *Pattern Recognition Letters*, 107:50–58, 2018.
3. Xianye Ben, Yi Ren, Junping Zhang, Su jing Wang, Kidiyo Kpalma, Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Specral networks and locally connected networks on graphs. *Int. Conf. Learn. Represent.*, 2014.
4. Xinlei Chen, Li-Jia Li, Fei-Fei Li, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7239–7248, 2018.
5. Ciprian A. Corneanu, Marc Olivi, Jeffrey F. Cohn, and Sergio Escalera. Survey on RGB, 3D, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(8):1548–1568, 2016.
6. Xianye Ben, Xiuting Jia, Rui Yan, Xin Zhang, and Weixiao Meng. Learning effective binary descriptors for micro-expression recognition transferred by macro-information. *Pattern Recognition Letters*, 107:50–58, 2018.
7. Ciprian A. Corneanu, Marc Olivi, Jeffrey F. Cohn, and Sergio Escalera. Survey on RGB, 3D, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(8):1548–1568, 2016.
8. Xinlei Chen, Li-Jia Li, Fei-Fei Li, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7239–7248, 2018.
TABLE VI
RECOGNITION RATES COMPARISON BETWEEN DUAL AND SINGLE GRAPH CONVOLUTIONAL NETWORK

| Method | Graph Architecture | Dataset | Pre-training | Spatial branch | Pre-training | Spatial branch | Fine-tuning | Testing | Fine-tuning | Testing |
|--------|--------------------|---------|--------------|----------------|--------------|----------------|------------|---------|------------|---------|
|        |                    |         | Fine-tuning   | Testing        | Fine-tuning   | Testing        |            |         |            |         |
|        |                    |         | 84%          | 68.8%          | 99.8%         | 65.9%          | 36.8%      | 28.4%   | 98.9%      | 36.8%   |
|        | Spatial + CE       |         | 84%          | 68.8%          | 99.8%         | 65.9%          | 36.8%      | 28.4%   | 98.9%      | 36.8%   |
|        | Temporal + CE      |         | 84%          | 68.8%          | 99.8%         | 65.9%          | 36.8%      | 28.4%   | 98.9%      | 36.8%   |
|        | Temporal + Temporal + CE |         | 84%          | 68.8%          | 99.8%         | 65.9%          | 36.8%      | 28.4%   | 98.9%      | 36.8%   |
|        | Temporal + Temporal + FL |         | 84%          | 68.8%          | 99.8%         | 65.9%          | 36.8%      | 28.4%   | 98.9%      | 36.8%   |

TABLE VII
RECOGNITION RATES OF OUR PROPOSED MODEL WITH DIFFERENT GRAPH ARCHITECTURES AND LOSS FUNCTIONS. "SPATIAL + CE" MEANS WE ONLY FEED PIXEL IMAGES INTO SGCNS AND THE LOSS IS CROSS ENTROPY. "TEMPORAL" MEANS WE ONLY FEED OPTICAL FLOW IMAGES INTO SGCNS. "SPATIAL + TEMPORAL" MEANS WE FEED BOTH PIXEL AND OPTICAL FLOW IMAGES INTO SGCNS. "FL" MEANS WE FEED THE FOCAL LOSS.
ence on Pattern Recognition, pages 2258–2263, 2017. 2, 6
[39] Min Peng, Chongyang Wang, Tong Chen, Guangyuan Liu, and Xiaolan Fu. Dual temporal scale convolutional neural network for micro-expression recognition. Frontiers in Psychology, 8:1745–1756, 2017. 2, 6
[40] Min Peng, Zhan Wu, Zhizhao Zhang, and Tong Chen. From macro to micro expression recognition: deep learning on small datasets using transfer learning. IEEE International Conference on Automatic Face and Gesture Recognition, pages 657–661, 2018. 6
[41] Avinash Ravichandran Riawan Chaudhry, Gregory Hager, and Rene Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. IEEE Conf. Comput. Vis. Pattern Recog., pages 1932–1939, 2009. 2
[42] Sujing Wang, Bingjun Li, Yongjin Liu, Wenjing Yan, Xinyu Ou, Xiaohua Huang, Feng Xu, and Xiaolan Fu. Micro-expression recognition with small sample size by transferring long-term convolutional neural network. Neurocomputing, 312:251–262, 2018. 1, 6
[43] Sujing Wang, Wen-Jing Yan, Xiaoai Li, Guoying Zhao, Chunguang Zhou, Xiaolan Fu, Minghao Yang, and Jianhua Tao. Micro-expression recognition using color spaces. IEEE Trans. Image Process., 24(12):6034–6047, 2015. 2
[44] Sujing Wang, Wen-Jing Yan, Guoying Zhao, Xiaolan Fu, and Chunguang Zhou. Micro-expression recognition using robust principal component analysis and local spatiotemporal directional features. Eur. Conf. Comput. Vis., pages 325–338, 2014. 2
[45] Yuan Zong, Xiaohua Huang, Zhen Cui, and Guoying Zhao. Micro-expression recognition using robust principal component analysis and local spatiotemporal directional features. ACM Int. Conf. Multimedia, pages 79–84, 2017. 1, 6
[46] Yuan Zong, Wenming Zheng, Zhen Cui, and Guoying Zhao. Toward bridging micro-expressions from different domain. IEEE Transactions on Cybernetics, 50(12):5047–5060, 2019. 2