IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 May 2022, accepted 16 June 2022, date of publication 21 June 2022, date of current version 27 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185132

Cross-Database Micro-Expression Recognition Based on a Dual-Stream Convolutional Neural Network

BAOLIN SONG[1], YUAN ZONG[2], (Member, IEEE), KE LI[1], JIE ZHU[1], JINGANG SHI[3], AND LI ZHAO[1]

[1]Key Laboratory of Underwater Acoustic Signal Processing of Ministry of Education, School of Information Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Child Development and Learning Science of Ministry of Education, School of Biological Science and Medical Engineering, Southeast University, Nanjing 210096, China
[3]Center for Machine Vision and Signal Analysis, University of Oulu, 90014 Oulu, Finland

Corresponding authors: Yuan Zong (xhzongyuan@seu.edu.cn) and Li Zhao (101008849@seu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1305200; in part by the National Natural Science Foundation of China under Grant 61921004, Grant 61902064, Grant 61572009, Grant 61906094, Grant 61673108, Grant 61571106, and Grant 61703201; in part by the Jiangsu Provincial Key Research and Development Program under Grant BE2016616; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20170765; and in part by the Fundamental Research Funds for the Central Universities under Grant 2242018K3DN01 and Grant 2242019K40047.

ABSTRACT Cross-database micro-expression recognition (CDMER) is a difficult task, where the target (testing) and source (training) samples come from different micro-expression (ME) databases, resulting in the inconsistency of the feature distributions between each other, and hence affecting the performance of many existing MER methods. To address this problem, we propose a dual-stream convolutional neural network (DSCNN) for dealing with CDMER tasks. In the DSCNN, two stream branches are designed to study temporal and facial region cues in ME samples with the goal of recognizing MEs. In addition, in the training process, the domain discrepancy loss is used to enforce the target and source samples to have similar feature distributions in some layers of the DSCNN. Extensive CDMER experiments are conducted to evaluate the DSCNN. The results show that our proposed DSCNN model achieves a higher recognition accuracy when compared with some representative CDMER methods.

INDEX TERMS Micro-expression recognition, CDMER, convolutional neural networks, domain adaptation.

I. INTRODUCTION

Compared with macro-expressions, micro-expressions (MEs) are one type of particular dynamic facial expression that have the characteristics of short duration and low intensity. The duration of ME is usually only 1/25 s to 1/3 s, with the facial muscle actions emerging in only small regions of the face. Although micro-expression recognition (MER) is an exceedingly arduous task, it has attracted many researchers. Many effective methods based on machine learning and deep learning have been proposed in recent years. For example, the conventional methods usually extract handcrafted features, e.g., LBP-TOP [1] and its variant (STLBP [2], DSLBP [3], LBP-SIP [4], and Hierarchical STLBP-IP [5]), MDMO [6], FHOFO [7], Bi-WOOF [8], and LTOGP [9], and then construct various types of classifiers, e.g., SVM [8], RF [10], k-NN [11], SRC [12], relaxed K-SVD [13], and GSL [5], especially for MER tasks [14]. In contrast, some deep learning methods have also been devoted to MER tasks, e.g., long short-term memory (LSTM) [15], pre-trained CNNs (e.g., OFF-ApexNet [16], MagGA/SA [17] and 3D-CNNs [18]), CapsuleNet [19] and STRCN [20]. These networks can usually improve the representation ability of MEs and learn the spatio-temporal features as well as the classifier in an end-to-end way [21].

These above methods are evaluated in an ideal scenario in which the testing samples and training samples are sourced from the same databases. In this case, it can be thought

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

that such training and testing samples abide by the same or similar feature distributions. However, in many applications, the testing samples and training samples may come from different databases (e.g., the target database, and the source database) that recorded by different camera, different subjects, stimulus materials under different environments. Some theoretical and empirical results [21]–[26] have shown that different training and testing databases have the large feature distribution difference and increase the test error in proportion. It thus brings us a new topic in micro-expression analysis, i.e., cross-database micro-expression recognition (CDMER), in which the training and testing samples come from two different micro-expression databases collected by different cameras or under different environments [27]. The CDMER can be viewed as a domain adaptation problem (DA). For two different databases, traditional classifiers learned in a source domain do not necessarily transfer well to target domains. We may learn proper feature representations that are discriminative and domain invariant by optimizing the DA methods. Recently, there are many classical domain adaptation methods for cross-database recognition can be applied to cross-database micro-expression recognition, e.g., Zong *et al.* [28] proposed a domain adaptation method based on target sample regenerator (TSRG) to deal with CDMER problem. Hassan *et al.* [29] proposed an importance-weighted SVM (IW-SVM) to eliminate the feature distribution mismatch between different samples and improve the classification accuracy under different databases. In the work of [30], Long *et al.* proposed the application of transfer kernel learning (TKL) to learn a domain invariant kernel for eliminating the feature distribution difference between the samples that come from different databases. Gong *et al.* [31], [32] proposed a method called the geodesic flow kernel (GFK) to bridge two different databases and narrow their gaps with a well-designed geodesic flow kernel on a Grassmann manifold. Chu *et al.* [33], [34] proposed a selective transfer machine (STM) to model the relationship between the training samples and their AU information, which aims to ensure that the testing samples have the similar feature distribution as the training ones by studying a group of weight values in the STM. Fernando *et al.* [35] proposed another method called subspace alignment (SA) for seeking a mapping function that can align the subspace in which the source samples lie with respect to the target samples. Pan *et al.*, [36] proposed a transfer component analysis (TCA) method based on a reproducing kernel Hilbert space to eliminate the distribution difference of samples from different domains by seeking some transfer components across domains. Li *et al.* [37] proposed a target-adapted least-squares regression (TALSR) method based on the enabled learned regression coefficient matrix, which can learn a regression coefficient matrix from the source samples and their label information to suit the target ME database.

Benefiting from the above methods, we propose a dual-stream CNN (DSCNN) to address the CDMER task by studying a group of weight values from the labeled source samples and the unlabeled target samples. We calculate the MMD value [38] between the output distribution of two domains on some fully connected layers of DSCNN as the domain discrepancy loss in the training process, which can eliminate the feature distribution difference between samples from two domains. Two stream branches of the DSCNN can jointly learn spatio-temporal features through different input clues in ME samples, which aim at improving the representation ability of MEs and optimizing for cross-database micro-expression classification. In addition, we visualize the feature maps of intermediate activations that are output by various convolution and pooling layers in the DSCNN. Extensive cross-database experiments are conducted under the designed protocol in [39], and the experimental results are compared with some representative methods in dealing with CDMER tasks. This result proves that the DSCNN has advantages over these representative methods.

The rest of the paper is organized as follows. In Section II, we describe the dual-stream convolutional neural network (DSCNN) model for CDMER in detail. Extensive experiments and analyses are given in Section III. Finally, the conclusion is drawn in Section IV.

II. PROPOSED METHOD

The DSCNN consists of two stream branches, which can jointly learn spatio-temporal features from two separate input clues in ME video samples. Each branch in the DSCNN is a convolutional neural network that uses 2D convolution kernels, pooling cells, and fully connected cells, which have the same structure. The structure of the same branches can allow the DSCNN achieve parameter fitting in a brief time by reducing the redundant parameters and realizing parameter sharing. Specifically, each stream branch in the DSCNN consists of 9 network processing layers: 1 fully connected layer, 3 pooling layers, and 5 convolutional layers, as shown in Table. 1.

For 5 convolutional layers in each branch, the number of convolutional kernels (N) is set equal to 64, 64, 64, 128, and 128. The N value of the last two convolutional layers is much larger than that of the first three convolutional layers. Many studies [40], [41] show that the N value gradually increases from small to large and can learn more abstract features that come from some important facial regions related to expression, such as the mouth or eye region. For the convolutional kernel on the first convolution layer, we use a kernel size of 5 × 5 with a stride size of S=1, and the zero padding is set equal to "valid". Meanwhile, the kernel size on the other four convolutional layers is set equal to 3×3, the stride size is set equal to 1, and the zero padding is set equal to 1.

For 3 pooling layers in each branch, the number of kernels (N) is set equal to 64, 64, and 128. For the max pooling layer, we use a window size of 5 × 5 with a stride size of 2, and the zero padding is set equal to 2. For 2 average pooling layers, we use a window size of 3 × 3 with a stride size of 2, and the zero padding is set equal to 0 and 1. Three pooling

TABLE 1. The structure of the DSCNN used for cross-database micro-expression recognition.

Model	DSCNN					
Stream	Spatial stream ConvNet			Temporal stream ConvNet		
Conv1	Filter size : 5 × 5	Stride : 1	N : 64	Filter size : 5 × 5	Stride : 1	N : 64
Maxpool1	Filter size : 5 × 5	Stride : 2	N : 64	Filter size : 5 × 5	Stride : 2	N : 64
Conv2	Filter size : 3 × 3	Stride : 1	N : 64	Filter size : 3 × 3	Stride : 1	N : 64
Conv3	Filter size : 3 × 3	Stride : 1	N : 64	Filter size : 3 × 3	Stride : 1	N : 64
Avepool1	Filter size : 3 × 3	Stride : 2	N : 64	Filter size : 3 × 3	Stride : 2	N : 64
Conv4	Filter size : 3 × 3	Stride : 1	N : 128	Filter size : 3 × 3	Stride : 1	N : 128
Conv5	Filter size : 3 × 3	Stride : 1	N : 128	Filter size : 3 × 3	Stride : 1	N : 128
Avepool2	Filter size : 3 × 3	Stride : 2	N : 128	Filter size : 3 × 3	Stride : 2	N : 128
FC1	1024			1024		
FC2	2048					
Output	3					

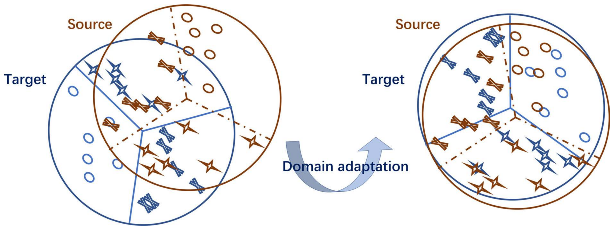


FIGURE 1. An illustration of how to solve the CDMER problem from the perspectives of DA and feature distribution. Eliminate the feature distribution difference between two domains through DA.

layers aim at downsampling the dimensions of features that are studied from spatio-temporal cues in ME video samples.

For the final connected layer in each branch, their output dimensions are all set equal to 1024, which aims to reduce the number of parameters in the DSCNN. At the end of two recognition stream branches, the output is merged into a 2048-dimensional feature vector. In the last fully connected layer of the DSCNN, the output dimension is set equal to the number of sample categories in the ME databases.

All hidden layers of the DSCNN are equipped with the PReLU function in [42], which is defined as follows:

$$PReLU(y_i) = \begin{cases} y_i, & y_i > 0 \\ a_i y_i, & y_i \leqslant 0 \end{cases} \tag{1}$$

where i is the channel number, and a_i is a parameter obtained in the training process. Compared with other activation functions, such as sigmoid, tanh, and ReLU, etc., the PReLU activation function can improve the classification ability of the CNN model at no cost of overfitting and computational complexity.

Micro-expressions are transient in an ME video, and the facial muscle actions emerge in only small regions of the face during a surprisingly short time. In the training process of the DSCNN, to reduce data redundancy and improve computational ability, we only use three important frames (i.e., the onset, apex, and offset frames) in each ME video. We use two stream ConvNets in the DSCNN to learn excellent feature representation from spatial and temporal cues in this three frames from ME videos. In each ME video, this three frames are resize to 48 × 48 after face alignment and face cropping. The apex frames of ME videos can be selected by the automatic apex frame spotting strategy in [43], which has the largest facial action amplitude and carries more expression information because facial muscle micro-movement of this frame is more obvious than that of other frames. The spatial stream ConvNet in the DSCNN operates on the gray image of the resized apex frame, learning some useful clues associated with particular facial action from the single frame. The input to the temporal stream ConvNet is the optical flow displacement field between three resized frames, which calculated by the method in [44]. Such input can explicitly describe the motion between video frames, and does not need to estimate series of subtle facial movements throughout the whole ME video implicitly. The temporal stream ConvNet ensures that the DSCNN can further learn higher-level features from temporal cues in ME videos for MER tasks.

To ensure that the DSCNN has sufficient training samples, we expand the number of samples by taking the gray image of the resized apex frame and the optical flow displacement field obtained from each ME video and applying a horizontal flip and clockwise/counterclockwise rotation in 5 or 10 degree increments a total of 10 times. When these sample is ready, we begin to train the DSCNN according to our purposes.

In the CDMER task, the testing samples and training samples may come from different databases, which can bring a large domain discrepancy and result in most MER methods being unsatisfactory [21]–[26]. Hence, directly training the DSCNN by using only the source samples often leads to overfitting of the distribution of the source samples, causing a significant reduction in the recognition performance in the target domain. For two biased datasets (left), traditional classifiers learned in a source domain do not necessarily transfer well to target domains, as outlined in Fig. 1. To address the feature distribution difference between this two different databases, we may learn proper feature representations in another feature space (right) that are discriminative and domain invariant by optimizing the ideas of domain adaptation (DA).

Benefiting from the DA, in the training process of the DSCNN, to ensure that source and target samples will have the similar feature distributions that are output by various convolution and pooling layers in the DSCNN, we should choose a proper metric to measure the feature distribution difference. There are many metrics to measure the difference of feature distribution between two different databases in some subspace, e.g., MMD [38], Wasserstein Distance [45], KLD [46], and A-distance [22]. In this paper, to measure the feature distribution difference between two domains on some layer of the DSCNN, we use the maximum mean discrepancy (MMD) from the work of [38] as the metrics. In addition, the MMD value is computed with respect to a kernel mapping operator, ϕ (.). In our DSCNN model, we define the output of deep features in the fully connected layer as ϕ (.), which operates on source data points, $x_s \in X_S$, and the target data points, $x_t \in X_T$. Then an empirical approximation to this

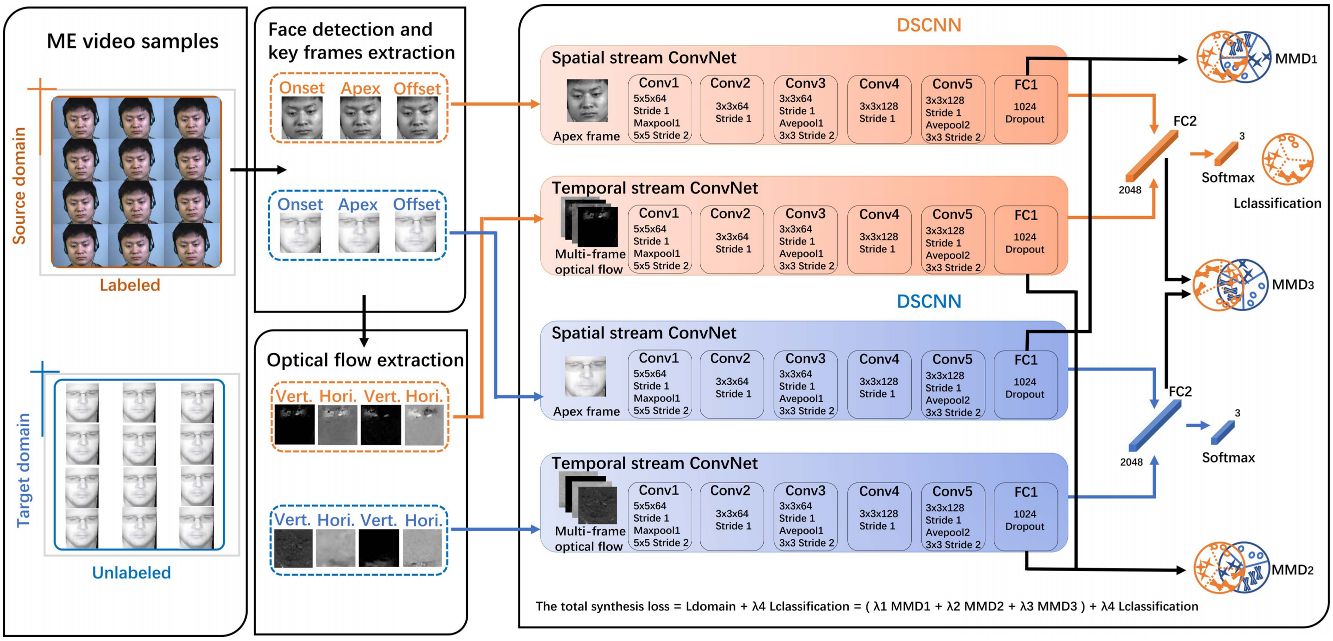


FIGURE 2. The framework of the DSCNN and the training process.

distance of the feature distribution between the source and target data on the connected layer can be defined as:

$$MMD(X_S, X_T) = \left\| \frac{1}{X_S} \sum_{x_s \in X_S} \phi\left(x_s\right) - \frac{1}{X_T} \sum_{x_t \in X_T} \phi\left(x_t\right) \right\|, \quad (2)$$

The smaller the MMD value, the more similar the distribution of the features obtained by the source sample and the target sample in each layer of the DSCNN.

The DSCNN is trained jointly on all labeled source data and unlabeled target data, as shown in in Fig. 2. Three MMD values (i.e., MMD_1, MMD_2, and MMD_3) are calculated based on the output of source data and target data on three selected connection layers in the DSCNN. To ensure that source and target samples will have similar feature distributions in some layer of the DSCNN, the domain discrepancy loss is defined as:

$$L_{domain} = \sum_{i=1,2,3} \lambda_i MMD_i(X_S, X_T), \quad (3)$$

where $MMD_1(X_S, X_T)$ denotes the feature distribution distance in the fully connected layer FC_1 of spatial stream ConvNet. $MMD_2(X_S, X_T)$ denotes the feature distribution distance in the fully connected layer FC_1 of temporal stream ConvNet. $MMD_3(X_S, X_T)$ denotes the feature distribution distance in the fully connected layer FC_2. The hyperparameter λ_i determines how strongly we would like to confuse two domains, and during the training process, the values of these parameters are determined by the best recognition results in the CDMER tasks.

In contrast, only labeled source samples are used to compute the classification loss of DSCNN, which can be defined as:

$$L_{classification} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{Y} \tau\left(y_n, j\right) \times \log P_{n,j}, \quad (4)$$

where N denotes the training sample size, Y denotes the category number of ME, y_n denotes the label of the n-th training sample and $P_{n,j}$ denotes the prediction value that the n-th training sample is predicted to be the j-th category.

To ensure that feature representations have good adaptation performance in CDMER tasks, the joint loss function used in the DSCNN can be defined as:

$$L_{total} = L_{domain} + \lambda_4 L_{classification}, \quad (5)$$

where $L_{classification}$ denotes the classification loss on labeled source data, and L_{domain} denotes the joint loss between the source data, X_S, and the target data, X_T on three selected connection layers in the DSCNN. We consider that such representations can offer strong semantic separation and have domain invariance in CDMER tasks.

The DSCNN uses the value of the joint loss function as a feedback signal to adjust the value of the weights by small amount in a direction that lowers the loss value for examples in CDMER tasks. This adjustment is the job of the "optimizer", which implements what is called the "back-propagation" algorithm (BP) [47]. We use the stochastic gradient descent algorithm with nesterov momentum as the training optimizer. The iterative process during the training of the DSCNN is shown as follows:

$$\begin{aligned} \upsilon_t &= \gamma \upsilon_{t-1} + \alpha \nabla_\theta J\left(\theta - \gamma \upsilon_{t-1}\right), \\ \theta &\leftarrow \theta - \upsilon_t. \end{aligned} \quad (6)$$

where α denotes the learning rate. The correction factor is set equal to 0.9, and the attenuation of the weight parameters is set equal to 10^{-5}. We use the strategy to minimize the value of the joint loss function in the DSCNN and gradually update the weight parameters to learn transferable feature representations between samples from two domains. After the recognition accuracy of the DSCNN in CDMER tasks tends to be stable, the optimization iteration process stops. Once the optimal weight parameters in the DSCNN are learned, we can use the DSCNN to address the CDMER tasks.

III. EXPERIMENTS

A. EXPERIMENTAL SETTING

In this section, we conduct experiments by using many domain adaptation (DA) methods for respectively investigating CDMER problem. In these experiments, we compare the proposed DSCNN with some representative methods including importance-weighted support vector machine (IW-SVM) [29], transfer kernel learning (TKL) [30], geodesic flow kernel (GFK) [31], selective transfer machine (STM) [33], subspace alignment (SA) [35], transfer component analysis (TCA) [36], target sample regenerator (TSRG) [28], DR in the Label Space (DRLS) [48], and region selective transfer regression (RSTR) [39]. For these DA methods, we employ the temporal interpolation model (TIM) [49] to normalize the frame number of all the micro-expression video clips to 16 and resize each frame image to 112×112. We compute uniform LBP-TOP [40] with fixed parameters using four types of spatial grids (1×1, 2×2, 3×3, and 4×4) in [39] to serve as the micro-expression features. For uniform LBP-TOP, neighboring radius R and number of the neighboring points P for LBP operator on three orthogonal planes are fixed at 3 and 8, respectively.

In our experiments, we choose uLSIF [50] to learn the importance weights for IW-SVM, which has shown its excellent performance in CDMER [28], [48]. For TKL, we determine the optimal value of ζ by searching from the parameter space [0.1: 0.1: 5]. The subspace of GFK, TCA, and SA are construct by principal component analysis (PCA) [51], and we search the optimal dimension k (the number of eigenvectors for composing the projection matrix) by trying all possible dimensions. The penalized coefficient in STM is set as $C = 1$, and the searching space of its second trade-off parameter λ is set as [0.001: 0.001: 0.009, 0.01: 0.01: 0.09, 0.1: 0.1: 1, 2: 1: 100, 1000, 10000]. The optimal values of two trade-off parameters λ and μ in TSRG and DRLS are determined by searching from [0.001, 0.01, 0.1, 1, 10, 100, 1000] (for λ) and [0.001: 0.001: 0.009, 0.01: 0.01: 0.09, 0.1: 0.1: 1, 2: 1: 10] (for μ). We search their optimal values from the preset parameter spaces, i.e., $\lambda \in$ [0.1, 1, 10, 100, 1000, 10000], $\mu \in$ [0.1: 0.1: 5], and $\tau \in$ [0.01: 0.01: 0.1].

In these experiments, we evaluate our DSCNN model using the same settings as in the work of [39]. Two publicly available ME databases (i.e., CASME II [52] and SMIC [53]) are used to build the CDMER tasks, which are often used in CDMER tasks. The two databases are shown as follows:

TABLE 2. The sample statistics of relabeled CASME II and SMIC (HS, VIS, NIR) used for CDMER experiments.

Micro-Expression Database	Micro-Expression Category		
	Positive	Negative	Surprise
Relabeled CASME II	32	91	25
SMIC (HS)	51	70	43
SMIC (VIS)	23	28	20
SMIC (NIR)	23	28	20

1) The CASME II database was created by Yan *et al.* from the Institute of Psychology, Chinese Academy of Science, which includes 247 ME samples with high resolution from 26 subjects. All samples were recorded at 200 fps, and categorized into 5 ME classes: happiness (32), surprise (25), disgust (64), repression (27), and others (99).
2) The SMIC database was created by Li *et al.* from the University of Oulu and has three subsets, i.e., SMIC (HS), SMIC (VIS) and SMIC (NIR). SMIC (HS) includes 164 ME samples from 16 subjects (i.e., 10 men and 6 women, 8 Caucasians and 8 Asians), and both SMIC (VIS) and SMIC (NIR) include 71 ME samples from 8 participants (i.e., 6 men and 2 women, 5 Caucasians and 3 Asians). These samples are recorded by a high-speed camera, a visual camera, and a near-infrared camera, respectively. All the ME samples are categorized into three categories: positive, negative, and surprise. Each subset can be used as an independent dataset, because they are recorded under different conditions.

To conduct cross-database experiments on the two databases, we need to make CASME II and SMIC have the same ME labeling. We select the samples of happiness, surprise, disgust, and repression from CASME II and then relabel them with the same ME labels in SMIC. The samples of happiness are relabeled as positive, and the samples of disgust and repression are relabeled as negative. The labels of surprise samples remain unchanged. The sample statistics of SMIC and relabeled CASME II can be found in Table 2.

In this paper, we conduct two types of CDMER experiments based on relabeled CASME II and subsets of SMIC. The TYPE-I type of experiments are conducted between either two subsets of SMIC (i.e., HS, VIS, NIR), such as HS → VIS, VIS → HS, HS → NIS, NIR → HS, VIS → NIR, and NIR → VIS. Meanwhile, the TYPE-II type of experiments are conducted between CASME II and one subset of SMIC (HS, VIS, NIR), such as CAS → HS, HS → CAS, CAS → VIS, VIS → CAS, CAS → NIR, and NIR → CAS. CAS, HS, VIS, and NIR are short for relabeled CASME II, SMIC (HS), SMIC (VIS), and SMIC (NIR). In the experiment

TABLE 3. The detailed information about two types of CDMER tasks.

TYPE	CDMER Task	Source Database (S)	Target Database (T)
TYPE-I	Expt.1: HS → VIS	SMIC (HS)	SMIC (VIS)
	Expt.2: VIS → HS	SMIC (VIS)	SMIC (HS)
	Expt.3: HS → NIR	SMIC (HS)	SMIC (NIR)
	Expt.4: NIR → HS	SMIC (NIR)	SMIC (HS)
	Expt.5: VIS → NIR	SMIC (VIS)	SMIC (NIR)
	Expt.6: NIR → VIS	SMIC (NIR)	SMIC (VIS)
TYPE-II	Expt.7: CAS → HS	Relabeled CASME II	SMIC (HS)
	Expt.8: HS → CAS	SMIC (HS)	Relabeled CASME II
	Expt.9: CAS → VIS	Relabeled CASME II	SMIC (VIS)
	Expt.10: VIS → CAS	SMIC (VIS)	Relabeled CASME II
	Expt.11: CAS → NIR	Relabeled CASME II	SMIC (NIR)
	Expt.12: NIR → CAS	SMIC (NIR)	Relabeled CASME II

of S → T, S and T denote the source and target ME databases, respectively. The statistics of these CDMER experiments are summarized in Table 3.

B. RESULTS AND ANALYSIS FOR CDMER

The mean F_1-score and accuracy are chosen as the evaluation metrics in the experiments. SVM is chosen as a baseline method to compare with other DA methods. The results of the TYPE-I and TYPE-II experiments are shown in Table 4 and Table 5, respectively. Compared with the SVM without domain adaptation, it is clear that these DA methods achieve significant improvement in the recognition ability in all the experiments. The results in Table 4 and Table 5 indicate that DA methods are effective ways to narrow the feature distribution gap between the samples from different ME databases when dealing with the CDMER problem. In addition, we also observe that the DSCNN achieves more promising results among all the representative DA methods selected for comparison. The DSCNN achieves an average mean F_1-score/accuracy of 0.7795/78.09% in the TYPE-I experiments and 0.6956/70.77% in TYPE-II experiments, which are significantly higher than those of the most DA methods for comparison. The performance of the DSCNN should be attributed to the design of two streams in the DSCNN and the idea of DA based on the domain discrepancy loss.

From Table 4 and Table 5, we observe that there are significant differences between the average results of each method in TYPE-I and TYPE-II experiments. TSRG, DRFS-T, and RSTR achieve the average mean F_1-score/accuracy of 0.6991/70.05%, 0.7128/71.23%, and 0.7381/73.98% in TYPE-I experiments, which are much higher than their achieved results (0.5348/56.22%, 0.5498/57.65%, and 0.5587/57.74%) in TYPE-II experiments. The result shows that TYPE-II experiments are significantly more difficult than TYPE-I experiments.

When SMIC (NIR) is used as the target database, i.e., Expt.3, Expt.5, and Expt.11, we can observe that the average performance of all the DA methods can reach 0.6888/69.08% and 0.7213/73.40% in Expt.3 and Expt.5, whose the source databases SMIC (HS) and SMIC (VIS) are relatively class-balanced. The result shows that the remaining one drops to 0.5443/55.43% in Expt.11, where the source databases of Expt.11 are relabeled CASME II, and very class-imbalanced.

Similarly, the average performance of all DA methods is also affected by the class-imbalanced target database. In Expt.6, Expt.4, and Expt.12, whose source database is fixed, i.e., SMIC (NIR), we observe that the average mean F_1-score / accuracy decreases from the level of 0.7552/76.46% (Expt.6: class-balanced) to 0.5710/58.06% (Expt.4: class-imbalanced) and 0.4530/47.30% (Expt.12: class-imbalanced), respectively.

From the results of TYPE-I and TYPE-II experiments, we notice that three subsets (i.e., HS, VIS, and NIR) of SMIC in TYPE-I experiments have the same subjects, stimulus materials, recording environments and different cameras, which results in the relatively small feature distribution difference. Meanwhile, compared with three subsets (i.e., HS, VIS, and NIR) of SMIC, relabeled CASME II used in TYPE-II experiments has substantially different subjects, stimulus materials, recording environments, and different cameras, which results in the relatively a large feature distribution difference. Therefore, the performance of all DA methods is affected by the class-imbalanced or heterogeneous problem between the source and target database when dealing with the CDMER tasks.

To test the structure of the DSCNN and its ability to learn salient characteristics from the ME samples, we compare the results between the DSCNN and OSCNN-I (or OSCNN-II), which only retains a single stream. We notice that the DSCNN achieves better performance than the single-stream networks in the TYPE-I and TYPE-II experiments. The result shows that the dual-stream structure in DSCNN can better utilize various forms of effective spatio-temporal characteristics for CDMER tasks, achieving better performance than some single-stream networks, such as OSCNN-I, and OSCNN-II.

C. DSCNN VISUALIZATION

In this section, to understand how pooling and convnet layers of the two stream ConvNets in the DSCNN transform their input, we visualize intermediate activations, which consists in displaying the feature maps that are output by various convolution and pooling layers in the DSCNN. This gives a view into how an input is decomposed into the different filters learned by the DSCNN.

We randomly choose a CDMER task from either TYPE-I or TYPE-II as an example of intermediate activation visualization, such as Expt.8: HS → CAS. When the training of the DSCNN is completed, we randomly choose an ME video from the target domain (i.e., CASME II) as the input, and visualize intermediate activations on various convolution and pooling layers in the DSCNN, as shown in Fig. 3 and Fig. 4.

Firstly, we observe that from Fig. 3 and Fig. 4, the feature maps extracted by a layer get increasingly abstract with the

TABLE 4. Experimental results (Mean F_1-score/accuracy) in the TYPE-I experiments, Where a series of CDMER tasks between three subsets of SMIC (i.e., HS, VIS, NIR). For short, HS = SMIC (HS), VIS = SMIC (VIS), and NIR = SMIC (NIR). The best results in each experiment are highlighted in bold.

Method	Expt.1: HS → VIR	Expt.2: VIS → HS	Expt.3: HS → NIR	Expt.4: NIR → HS	Expt.5: VIS → NIR	Expt.6: NIR → VIS	Average
SVM (Baseline)	0.8002/80.28	0.5421/54.27	0.5455/53.52	0.4847/54.88	0.6186/63.38	0.6078/63.38	0.6003/61.62
IW-SVM [29]	0.8868/88.73	0.5852/58.54	0.7469/74.65	0.5427/54.27	0.6620/69.01	0.7228/73.24	0.6911/68.07
TCA [36]	0.8269/83.10	0.5477/54.88	0.5828/59.15	0.5443/57.32	0.5810/61.97	0.6598/67.61	0.6238/64.01
GFK [31]	0.8448/84.51	0.5957/59.15	0.6977/70.42	**0.6197/62.80**	0.7619/76.06	0.8142/81.69	0.7223/72.44
SA [35]	0.8037/80.28	0.5955/59.15	0.7465/76.65	0.5644/56.10	0.7004/71.83	0.7394/74.65	0.6917/69.44
STM [33]	0.8253/83.10	0.5059/51.22	0.6628/66.20	0.5351/56.10	0.6427/67.61	0.6922/70.42	0.6440/65.78
TKL [30]	0.7742/77.46	0.5738/57.32	0.7051/70.42	0.6116/62.20	0.7558/76.06	0.7579/76.06	0.6964/69.92
TSRG [28]	**0.8869/88.73**	0.5652/56.71	0.6484/64.79	0.5770/57.93	0.7056/70.42	0.8116/81.69	0.6991/70.05
DRFS-T [48]	0.8643/85.92	0.5767/57.32	0.7179/71.83	0.6163/61.59	0.7286/73.24	0.7732/77.46	0.7128/71.23
DRLS [48]	0.8604/85.92	0.6120/60.98	0.6599/66.20	0.5599/55.49	0.6620/69.01	0.5771/61.97	0.6552/66.60
RSTR [39]	0.8721/87.32	0.6401/64.02	0.7466/74.65	0.5765/57.32	0.7506/76.06	0.8428/84.51	0.7381/73.98
OSCNN-I (ours)[a]	0.8154/81.46	0.6281/62.92	0.7021/70.36	0.5610/56.27	0.8146/81.58	0.8284/82.91	0.7249/72.58
OSCNN-II (ours)[b]	0.8332/83.19	0.6503/65.15	0.7239/72.58	0.5829/58.35	0.8413/84.02	0.8594/86.02	0.7485/74.89
DSCNN (ours)	0.8617/86.24	**0.6814/68.27**	**0.7567/75.73**	0.6184/62.17	**0.8726/87.32**	**0.8863/88.79**	**0.7795/78.09**
Average	0.8397/84.02	0.5900/59.28	0.6888/69.08	0.5710/58.06	0.7213/73.40	0.7552/76.46	-

[a] OSCNN-I - the network that only retains a single stream, trained on apex frames of ME samples.

[b] OSCNN-II - the network that only retains a single stream, trained on optical flow displacement fields between three important frames in ME samples.

TABLE 5. Experimental results (Mean F_1-score/accuracy) in The TYPE-II experiments, Where a series of CDMER tasks between The CASME II and one subsets of SMIC (i.e., HS, VIS, NIR). For short, CAS = relabeled CASME II, HS = SMIC (HS), VIS = SMIC (VIS), and NIR = SMIC (NIR). The best results in each experiment are highlighted in bold.

Method	Expt.7: CAS → HS	Expt.8: HS → CAS	Expt.9: CAS → VIS	Expt.10: VIS → CAS	Expt.11: CAS → NIR	Expt.12: NIR → CAS	Average
SVM (Baseline)	0.3697/45.12	0.3245/48.46	0.4701/50.70	0.5367/53.08	0.5295/52.11	0.2368/23.85	0.4112/45.55
IW-SVM [29]	0.3541/41.46	0.5829/62.31	0.5778/59.15	0.5537/54.62	0.5117/50.70	0.3456/36.15	0.4876/50.73
TCA [36]	0.4637/40.34	0.4870/53.08	0.6834/69.01	0.5789/59.23	0.4992/50.70	0.3937/42.31	0.5177/53.45
GFK [31]	0.4126/46.95	0.4776/50.77	0.6361/66.20	0.6056/61.50	0.5180/53.52	0.4469/46.92	0.5161/54.31
SA [35]	0.4302/47.56	0.5447/62.31	0.5939/59.15	0.5243/51.54	0.4738/47.89	0.3592/36.92	0.4877/50.90
STM [33]	0.3640/43.90	0.6115/63.85	0.4051/52.11	0.2715/30.00	0.3523/42.25	0.3850/41.54	0.3982/45.61
TKL [30]	0.3829/44.51	0.4661/54.62	0.6042/60.56	0.5378/53.08	0.5392/54.93	0.4248/43.85	0.4925/51.93
TSRG [28]	0.5042/51.38	0.5171/60.77	0.5935/59.15	0.6208/63.08	0.5624/56.34	0.4105/46.15	0.5348/56.22
DRFS-T [48]	0.4524/46.95	0.5460/60.00	0.6217/63.38	0.6762/68.46	0.5369/56.34	0.4653/50.77	0.5498/57.65
DRLS [48]	0.4924/53.05	0.5267/59.23	0.5757/57.75	0.5942/60.00	0.4885/49.83	0.3838/42.37	0.5102/53.71
RSTR [39]	0.5297/54.27	0.5622/60.77	0.5882/59.15	**0.7021/70.77**	0.5009/50.70	0.4693/50.77	0.5587/57.74
TALSR [37]	0.4934/49.39	0.5366/57.69	0.6362/63.38	0.5966/60.00	0.5907/59.15	0.4622/47.69	0.5526/56.21
OSCNN-I (ours)[a]	0.5841/60.11	0.6595/70.24	0.6942/68.93	0.6597/65.13	0.6704/67.52	0.6526/64.78	0.6534/66.12
OSCNN-II (ours)[b]	0.6229/64.52	0.6653/72.37	0.7026/71.57	0.6784/68.96	0.6891/69.13	0.6696/66.54	0.6713/68.85
DSCNN (ours)	**0.6524/67.13**	**0.6876/74.35**	**0.7234/73.48**	0.6984/70.44	**0.7012/70.30**	**0.6904/68.92**	**0.6956/70.77**
Average	0.4304/50.44	0.5464/60.72	0.6071/62.24	0.5890/59.33	0.5443/55.43	0.4530/47.30	-

[a] OSCNN-I - the network that only retains a single stream, trained on apex frames of ME samples.

[b] OSCNN-II - the network that only retains a single stream, trained on optical flow displacement fields between three important frames in ME samples.

depth of the layers in the DSCNN. The intermediate activations of layers higher-up carry less and less information about the specific input being seen, and more and more information about the class of the target: positive, negative, or surprise.

Secondly, in Fig. 3, we can observe that some intermediate activations in layers of the spatial stream ConvNet, which clearly show that some appearance and outline information of a whole face. It shows that the spatial stream ConvNet in the DSCNN operates on the gray image of the resized apex frame, learning some useful spatial clues associated with particular facial texture information from the single frame. Facial expressions are strongly associated with these particular facial texture information that is the most intuitive.

Thirdly, in Fig. 4, we can observe that some intermediate activations in layers of the temporal stream ConvNet, which clearly show that the muscle movements in the subject's eyebrows from the occurrence to the disappearance of an disgust micro-expression, although the amplitude of the facial muscle motion between adjacent frames is very small. It shows that the spatial stream ConvNet in the DSCNN operates on the optical flow displacement field between three resized frames, learning some useful temporal clues associated with the facial muscle actions during a short time.

Based on the above observations, two stream ConvNets in the DSCNN effectively act as an information distiller, with raw data going in, and getting repeatedly transformed so that irrelevant information gets filtered out while useful information from spatial and temporal cues in three frames of each ME video get magnified and refined.

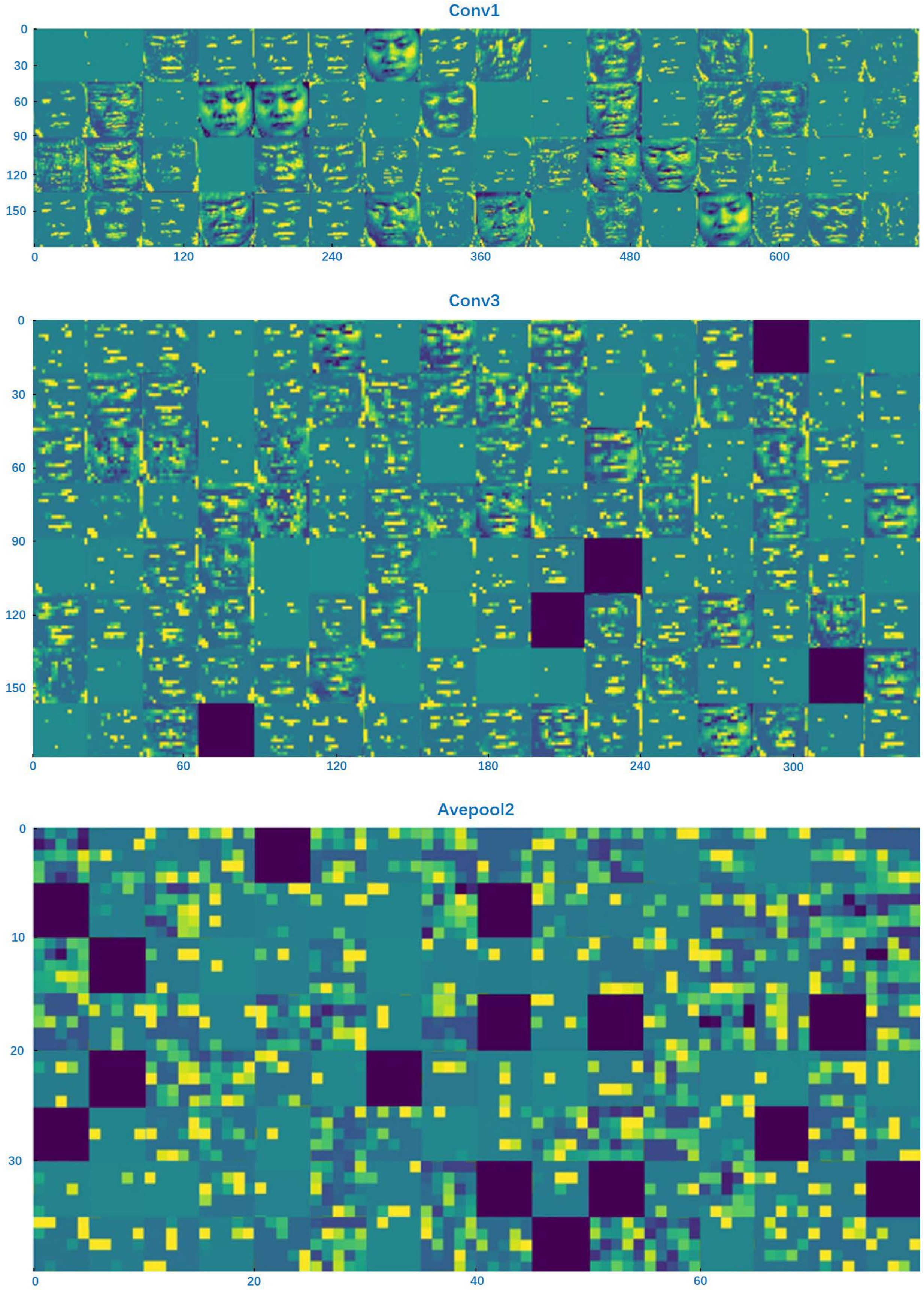
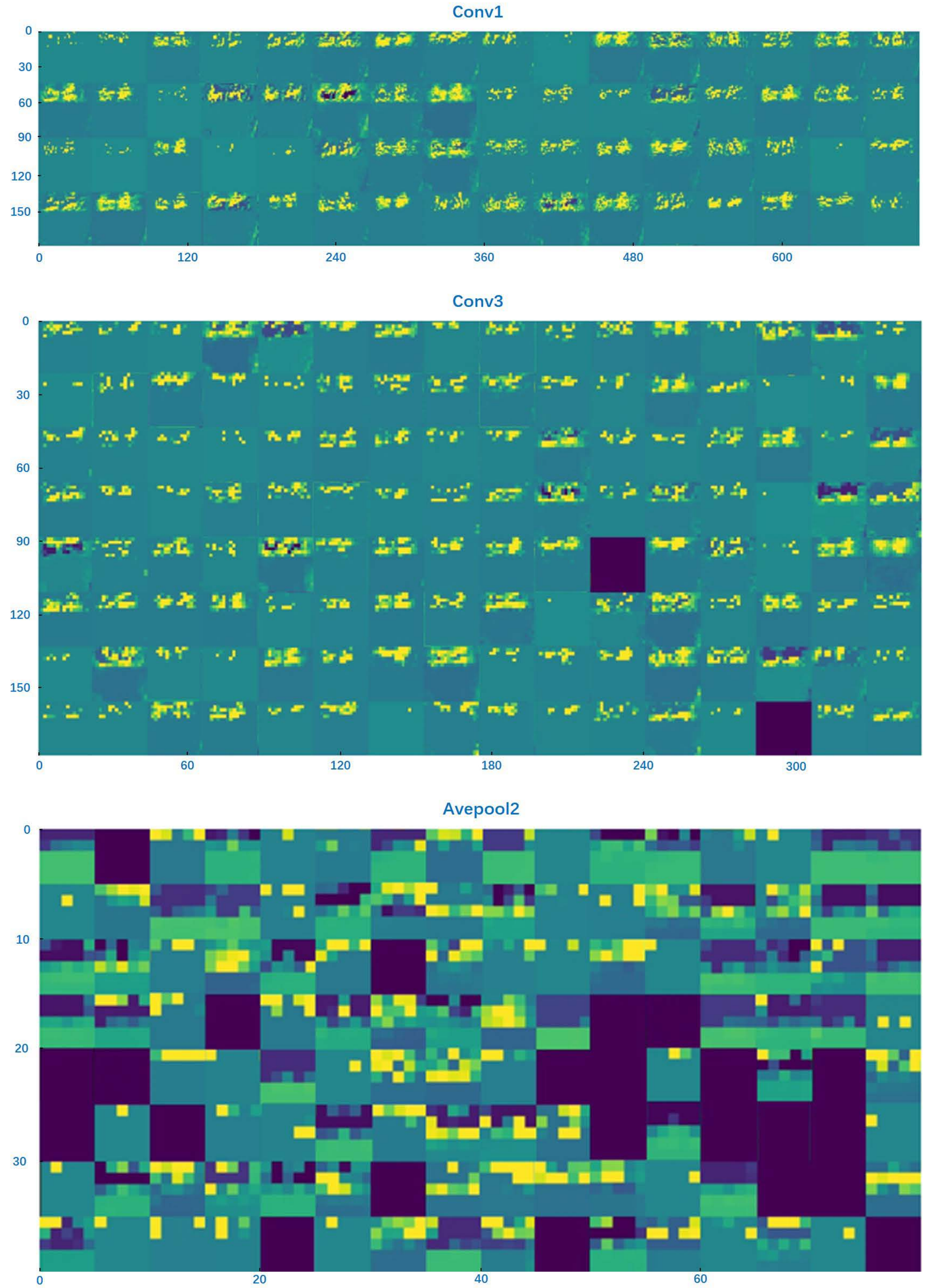


FIGURE 3. **The visualization of intermediate activations on some convolution and pooling layers in the spatial stream ConvNet of the DSCNN.**



Conv1

Conv3

Avepool2

FIGURE 4. The visualization of intermediate activations on some convolution and pooling layers in the temporal stream ConvNet of the DSCNN.

IV. CONCLUSION

In this paper, we propose a dual-stream convolutional neural network called DSCNN to address CDMER tasks. Our method is novel in that we take a domain discrepancy loss and a classification loss to minimize the feature distribution difference between the source and target domains. Two streams in the DSCNN can jointly learn spatio-temporal features of ME samples to optimize for cross-database micro-expression classification through different input clues in ME samples. To evaluate the performance of DSCNN, we conduct TYPE-I and TYPE-II experiments on relabeled CASME II and three subsets of SMIC (i.e., HS, VIS, and NIR). Compared with some representative DA methods, our proposed DSCNN has an overall superior performance. We observe that the performance of DA methods is affected by the class-imbalanced or heterogeneous problem between the source and target database when handing the CDMER tasks. In the future, we could focus on designing a better spatio-temporal feature extraction method for CDMER tasks, and studying faster optical flow calculation methods. In addition, we plan to design a simpler network structure with multiple recognition tubes to cope with CDMER tasks and verify the effectiveness of the proposed model on more ME databases.

REFERENCES

[1] T. Pfister, X. Li, G. Zhao, and M. Pietikainen, "Recognising spontaneous facial micro-expressions," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Washington, DC, USA, Nov. 2011, pp. 1449–1456, doi: 10.1109/ICCV.2011.6126401.

[2] X. Huang, S.-J. Wang, G. Zhao, and M. Piteikainen, "Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 1–9.

[3] X. Huang, S.-J. Wang, X. Liu, G. Zhao, X. Feng, and M. Pietikainen, "Discriminative spatiotemporal local binary pattern with revisited integral projection for spontaneous facial micro-expression recognition," *IEEE Trans. Affect. Comput.*, vol. 10, no. 1, pp. 32–47, Jan./Mar. 2019.

[4] Y. Wang, J. See, R. C.-W. Phan, and Y.-H. Oh, "LBP with six intersection points: Reducing redundant information in LBP-top for micro-expression recognition," in *Computer Vision—ACCV*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds. Cham, Switzerland: Springer, 2015, pp. 525–537.

[5] Y. Zong, X. Huang, W. Zheng, Z. Cui, and G. Zhao, "Learning from hierarchical spatiotemporal descriptors for micro-expression recognition," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3160–3172, Nov. 2018.

[6] Y.-J. Liu, B.-J. Li, and Y.-K. Lai, "Sparse MDMO: Learning a discriminative feature for micro-expression recognition," *IEEE Trans. Affect. Comput.*, vol. 12, no. 1, pp. 254–261, Mar. 2021.

[7] S. L. Happy and A. Routray, "Fuzzy histogram of optical flow orientations for micro-expression recognition," *IEEE Trans. Affect. Comput.*, vol. 10, no. 3, pp. 394–406, Jul./Sep. 2019.

[8] S. Liong, J. See, K. Wong, and R. C. Phan, "Less is more: Micro-expression recognition from video using apex frame," *Signal Process., Image Commun.*, vol. 62, pp. 82–92, Mar. 2018.

[9] M. Niu, J. Tao, Y. Li, J. Huang, and Z. Lian, "Discriminative video representation with temporal order for micro-expression recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2112–2116.

[10] X. Li, X. Hong, A. Moilanen, X. Huang, T. Pfister, G. Zhao, and M. Pietikäinen, "Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods," *IEEE Trans. Affective Comput.*, vol. 9, no. 4, pp. 563–577, Oct. 2018.

[11] 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," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 3473–3479.

[12] H. Zheng, "Micro-expression recognition based on 2D Gabor filter and sparse representation," *J. Phys., Conf. Ser.*, vol. 787, Jan. 2017, Art. no. 012013, doi: 10.1088/1742-6596/787/1/012013.

[13] H. Zheng, X. Geng, and Z. Yang, "A relaxed k-SVD algorithm for spontaneous micro-expression recognition," in *PRICAI 2016: Trends in Artificial Intelligence*, R. Booth and M.-L. Zhang, Eds. Cham, Switzerland: Springer, 2016, pp. 692–699.

[14] Y.-H. Oh, J. See, A. C. Le Ngo, R. C.-W. Phan, and V. M. Baskaran, "A survey of automatic facial micro-expression analysis: Databases, methods, and challenges," *Frontiers Psychol.*, vol. 9, p. 1128, Jul. 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2018.01128

[15] S.-J. Wang, B.-J. Li, Y.-J. Liu, W.-J. Yan, X. Ou, X. Huang, F. Xu, and X. Fu, "Micro-expression recognition with small sample size by transferring long-term convolutional neural network," *Neurocomputing*, vol. 312, pp. 251–262, Oct. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218307045

[16] Y. S. Gan, S.-T. Liong, W.-C. Yau, Y.-C. Huang, and L.-K. Tan, "OFF-ApexNet on micro-expression recognition system," *Signal Process., Image Commun.*, vol. 74, pp. 129–139, May 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596518310038

[17] Y. Li, X. Huang, and G. Zhao, "Joint local and global information learning with single apex frame detection for micro-expression recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 249–263, 2021.

[18] R. Zhi, H. Xu, M. Wan, and T. Li, "Combining 3D convolutional neural networks with transfer learning by supervised pre-training for facial micro-expression recognition," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 5, pp. 1054–1064, 2019.

[19] N. Liu, X. Liu, Z. Zhang, X. Xu, and T. Chen, "Offset or onset frame: A multi-stream convolutional neural network with CapsuleNet module for micro-expression recognition," in *Proc. 5th Int. Conf. Intell. Informat. Biomed. Sci. (ICIIBMS)*, Nov. 2020, pp. 236–240.

[20] Z. Xia, X. Hong, X. Gao, X. Feng, and G. Zhao, "Spatiotemporal recurrent convolutional networks for recognizing spontaneous micro-expressions," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 626–640, Mar. 2020.

[21] Z. Xia, W. Peng, H.-Q. Khor, X. Feng, and G. Zhao, "Revealing the invisible with model and data shrinking for composite-database micro-expression recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 8590–8605, 2020.

[22] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. NIPS*, 2007, pp. 137–144.

[23] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Proc. NIPS*, 2007.

[24] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 213–226.

[25] A. Torralba and A. Efros, "Unbiased look at dataset bias," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1521–1528.

[26] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, May 2010.

[27] Y. Zhang, Y. Liu, and H. Wang, "Cross-database micro-expression recognition exploiting intradomain structure," *J. Healthcare Eng.*, vol. 2021, pp. 1–9, May 2021, doi: 10.1155/2021/5511509.

[28] Y. Zong, X. Huang, W. Zheng, Z. Cui, and G. Zhao, "Learning a target sample re-generator for cross-database micro-expression recognition," in *Proc. MM*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 872–880, doi: 10.1145/3123266.3123367.

[29] A. Hassan, R. Damper, and M. Niranjan, "On acoustic emotion recognition: Compensating for covariate shift," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 7, pp. 1458–1468, Jul. 2013.

[30] M. Long, J. Wang, J. Sun, and P. S. Yu, "Domain invariant transfer kernel learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1519–1532, Jun. 2014.

[31] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.

[32] T. Liu and Y. Gu, "Unsupervised temporal-adaptation with multiple geodesic flow kernels for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 10111–10114.

[33] W.-S. Chu, F. De la Torre, and J. F. Cohn, "Selective transfer machine for personalized facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 529–545, Mar. 2017.

[34] W.-S. Chu, F. De la Torre, and J. F. Cohn, "Selective transfer machine for personalized facial action unit detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3515–3522.

[35] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.

[36] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2010.

[37] L. Li, X. Zhou, Y. Zong, W. Zheng, X. Chen, J. Shi, and P. Song, "Unsupervised cross-database micro-expression recognition using target-adapted least-squares regression," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 7, pp. 1417–1421, 2019.

[38] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, Jul. 2006, doi: 10.1093/bioinformatics/btl242.

[39] T. Zhang, Y. Zong, W. Zheng, C. L. P. Chen, X. Hong, C. Tang, Z. Cui, and G. Zhao, "Cross-database micro-expression recognition: A benchmark," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 544–559, Feb. 2022.

[40] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, Jun. 2007.

[41] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proc. ACM Int. Conf. Multimodal Interact.*, Nov. 2015, pp. 435–442.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[43] S.-T. Liong, J. See, K. Wong, A. C. Le Ngo, Y.-H. Oh, and R. Phan, "Automatic apex frame spotting in micro-expression database," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 665–669.

[44] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[45] V. M. Panaretos and Y. Zemel, "Statistical aspects of Wasserstein distances," *Annu. Rev. Statist. Appl.*, vol. 6, no. 22, pp. 405–431, Mar. 2019.

[46] P. Sankaran, S. Sunoj, and N. Nair, "Kullback–Leibler divergence: A quantile approach," *Statist. Probab. Lett.*, vol. 111, pp. 72–79, Apr. 2016.

[47] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[48] Y. Zong, W. Zheng, X. Huang, J. Shi, Z. Cui, and G. Zhao, "Domain regeneration for cross-database micro-expression recognition," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2484–2498, May 2018.

[49] Z. Zhou, G. Zhao, and M. Pietikainen, "Towards a practical lipreading system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 137–144.

[50] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, Jul. 2009.

[51] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[52] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, and X. Fu, "CASME II: An improved spontaneous micro-expression database and the baseline evaluation," *PLoS ONE*, vol. 9, no. 1, Jan. 2014, Art. no. e86041.

[53] X. Li, T. Pfister, X. Huang, G. Zhao, and M. Pietikäinen, "A spontaneous micro-expression database: Inducement, collection and baseline," in *Proc. 10th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit. (FG)*, Apr. 2013, pp. 1–6.

BAOLIN SONG received the B.S. degree from Qingdao University (QDU), in 2010, and the M.S. degree from Shandong Normal University (SNU), in 2014. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Southeast University, China. His research interests include computer vision, pattern recognition, and micro-expression recognition.

YUAN ZONG (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Nanjing Normal University, Nanjing, China, in 2011 and 2014, respectively, and the Ph.D. degree in medical engineering from Southeast University, Nanjing, in 2019. From 2016 to 2017, he was working as a Visiting Student with the Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland. He is currently a Lecturer with the Key Laboratory of Child Development and Learning Science of Ministry of Education, School of Biological Science and Medical Engineering, Southeast University. His research interests include affective computing, pattern recognition, and computer vision.

KE LI received the B.S. degree in information engineering from Southeast University, Nanjing, China, in 2019, where he is currently pursuing the M.E. degree with the Engineering Centre of Signal and Information Processing, School of Information Science and Engineering. His research interests include image processing, affective computing, and pattern recognition.

JIE ZHU received the B.S. degree in communication engineering from Hohai University, Nanjing, China, in 2018. She is currently pursuing the Ph.D. degree with the Key Laboratory of Signal and Information Processing, School of Information Science and Engineering, Southeast University, Nanjing. Her research interests include speech recognition and computer vision.

JINGANG SHI received the B.S. and Ph.D. degrees from the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. Since 2017, he has been a Postdoctoral Researcher with the Center for Machine Vision Research and Signal Analysis, University of Oulu, Finland. His current research interests include image restoration and face analysis.

LI ZHAO received the B.E. degree from the Nanjing University of Aeronautics and Astronautics, China, in 1982, the M.S. degree from Suzhou University, China, in 1988, and the Ph.D. degree from the Kyoto Institute of Technology, Japan, in 1998. He is currently a Professor with Southeast University, China. His research interests include speech signal processing and pattern recognition.