Typical Facial Expression Network Using a Facial Feature Decoupler and Spatial-Temporal Learning

Jianing Teng, Dong Zhang*, Wei Zou, Ming Li, *Senior Member, IEEE,* and Dah-Jye Lee, *Senior Member, IEEE*

Abstract—Facial expression recognition (FER) accuracy is often affected by an individual's unique facial characteristics. Recognition performance can be improved if the influence from these physical characteristics is minimized. Using video instead of single image for FER provides better results but requires extracting temporal features and the spatial structure of facial expressions in an integrated manner. We propose a new network called Typical Facial Expression Network (TFEN) to address both challenges. TFEN uses two deep two-dimensional (2D) convolutional neural networks (CNNs) to extract facial and expression features from input video. A facial feature decoupler decouples facial features from expression features to minimize the influence from inter-subject face variations. These networks combine with a 3D CNN and form a spatial-temporal learning network to jointly explore the spatial-temporal features in a video. A facial recognition network works as an adversarial network to refine the facial feature decoupler and the network performance by minimizing the residual influence of facial features after decoupling. The whole network is trained with an adversarial algorithm to improve FER performance. TFEN was evaluated on four popular dynamic FER datasets. Experimental results show TFEN achieves or outperforms the recognition accuracy of state-of-the-art approaches.

Index Terms—Facial expression recognition, Facial feature decoupling, Spatial-temporal features, Adversarial training algorithm, 3D CNN.

1 Introduction

Facial expression recognition (FER) automatically classifies images and videos of human faces into certain typical emotions such as anger, disgust, fear, happiness, sadness, and surprise. FER is an important component in many human computer interaction applications [1], [2], [3] because it reveals the user's emotional state and intention. Existing FER approaches fall into two categories: frame-based and video-based. Frame-based FER methods capture spatial features from a single image or frame for expression recognition. Video-based FER algorithms characterize the spatial-temporal structure from contiguous frames in a video and use the spatial-temporal features for expression recognition.

Recent research has achieved promising results for FER using still images [4], [5], [6]. Even though the emotional state or intention visible in a human expression usually lasts for only a split second, the facial expressions often contain characteristic patterns with specific and important spatial-temporal structures that cannot be captured with a single image. Video-based FER explores these expression changes themselves and is able to capture the emotional state and intention using consecutive frames.

Traditional video-based FER methods use two separate networks to capture the spatial and temporal features and combine them with post-hoc integration methods for expression recognition [7], [8]. Although many approaches for video-based FER have been proposed in recent years and obtained promising results, recognizing expressions using integrated spatial-temporal features in an optimized manner remains an open challenge.

Another challenge in video-based FER is how to alleviate the negative influence from each individual's unique facial (biological) features. Facial expression features can be regarded as being formed by encoding typical facial expression (TFE) features (or refined expression features) with unique facial features (or biological features). The same TFE can differ from person to person because of the influence of individual facial features. Likewise, the appearance of two different TFEs, affected by the unique facial features of different individuals, may look similar. For example, each row in Fig. 1 shows a single TFE expressed by three people; the expressions may look different because of variations in faces or facial features. Also in Fig. 1, Person A's anger expression may look similar to Person B's disgust expression because of interference from their unique facial features. Facial feature interference makes facial expression recognition more challenging.

Our hypothesis is that minimizing the influence of an individual's unique facial features can improve FER accu-

- *Jianing Teng, Dong Zhang, and Wei Zou are with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China, 510006, and Shunde Research Institute, Shunde, China, 528300. E-mail: tengjn@qq.com, zhangd@mail.sysu.edu.cn, zouw23@mail2.sysu.edu.cn.*
Ming Li is with the Data Science Research Center of Duke Kunshan University, Kunshan, China, 215316, and the School of Computer Science, Wuhan University, Wuhan, China, 430072. E-mail: ming.li369@dukekunshan.edu.cn.
Dah-Jye Lee is with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, Utah, USA, 84602. E-mail: djlee@byu.edu.
The authors contributed equally to this paper.
**Corresponding author: Dong Zhang, Email: zhangd@mail.sysu.edu.cn*

*Manuscript received *; revised *.*

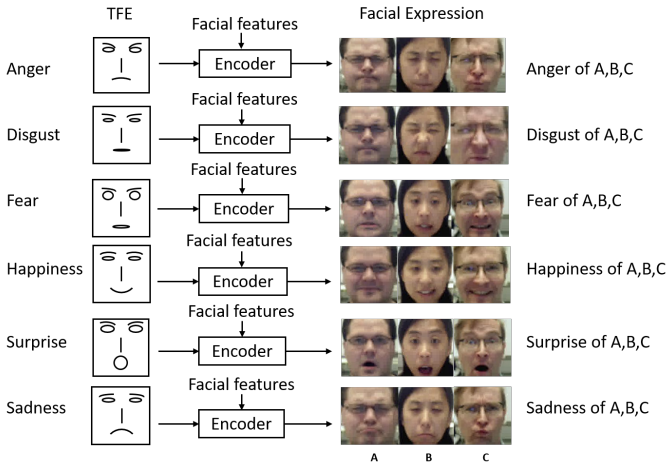

Fig. 1. Facial expressions are formed by encoding typical facial expression (TFE) features (or refined expression features) with unique facial features (or biological features). A TFE may appear differently in an individual's facial expression because of the influence from unique facial features. The appearance of two different TFEs, affected by the unique facial features of different people, may look similar. For example, the anger expressions of persons A, B, and C look different from each other. Person A's anger and person B's disgust expressions look similar. Persons A, B, and C were randomly selected from the Oulu-CASIA dataset.

racy. To alleviate the influence from facial variations, we pretrain a standard facial recognition network as a facial feature extractor to extract the unique features that are normally used for facial recognition. We design a facial feature decoupler to decouple facial features from expression features. The filtered expression features or so-called TFE features are then sent to a three-dimensional (3D) convolutional neural network (CNN) module to extract the spatial-temporal structure of the expression for recognition. A face classifier serving as an adversarial network is used to measure the residual influence of the facial characteristics in the TFE features after decoupling. The facial identification error of the adversarial network is then fed back to the expression extractor, decoupler, and 3D CNN to update the network parameters. The entire network is trained with an adversarial algorithm designed specifically to improve the efficiency of facial feature decoupling.

We performed experiments on three widely used lab-setting datasets, including CK+, MMI, Oulu-CASIA, and a large-scale "in the wild" dataset called DFEW. Experimental results show our proposed method achieved or outperformed the recognition accuracy of the state-of-the-art methods. Experiments also show that the accuracy of facial expression recognition is improved by decoupling facial characteristics from expression features.

2 Related Work

To explore a user's emotional state and intention in a more efficient way, many past works treated the FER problem as a dynamic, natural event and proposed to capture features from the spatial and temporal structures of the video for facial expression recognition [9], [10], [11]. How to extract temporal features and the spatial structure of a facial expression in an integrated manner has become the key challenge for video-based FER.

Among traditional approaches for video-based FER, many effective pattern descriptors have been used to explore spatial and temporal information in facial expression videos. Klaser et al., inspired by the success of HoG-based descriptors for image-based FER [12], characterized the spatial-temporal structure of video based on the histograms of oriented 3D spatial-temporal gradients. Zhao et al. extended the concept of texture to the temporal domain and proposed local binary patterns from three orthogonal planes (LBP-TOP) to describe the spatial-temporal structure of the video for FER [13]. Scovanner et al. proposed to describe the nature of video data as a bag of spatial-temporal words using 3D SIFT descriptor [14]. These methods extended traditional hand-crafted features from a single frame to consecutive frames and attempted to use the low-level features to explicitly describe the true spatial-temporal nature of the video data. However, methods based on hand-crafted descriptors lack the ability to describe high-level semantic features and fail to extract powerful temporal features hidden in the video.

In the past few years, deep neural networks have shown superior performance for various vision and pattern recognition tasks [15], [16], [17]. Some researchers have employed deep neural networks to explore temporal relationships among frames of expression videos and achieved promising results compared with methods based on hand-crafted descriptors. Jung et al. [7] captured temporal appearance and geometry features separately with a deep network based on two different models. Their network, DTAGN, obtained superior recognition accuracy on the CK+ and Oulu-CASIA datasets and outperformed all other methods using hand-crafted features. Zhang et al. [8] utilized a hierarchical bidirectional recurrent neural network (PHRNN) and a multi-signal convolutional neural network (MSCNN) to extract the partial-whole, geometry-appearance, and dynamic-still information and obtained better accuracy than DTAGN. An important idea shared by Jung and Zhang was that a recurrent neural network (RNN) and a CNN are able to extract discriminative temporal and spatial hidden information from facial expression videos in different abstract levels. These works have demonstrated the potential of integrating spatial and temporal information from a unified optimization framework to improve video-based FER performance.

The 3D CNN has become one of the successful models in the field of action recognition as it explores the motion information implied in multiple consecutive frames [18]. In order to jointly localize the facial action parts and learn the part-based representation of faces, Liu et al. adapted the 3D CNN with deformable action parts constraints for dynamic expression analysis [18]. Their proposed model, 3DCNN-DAP, obtained promising results on the CK+ and MMI datasets. However, a 3D CNN usually requires a large number of parameters, which makes it easy to overfit, especially on small datasets.

Although the aforementioned research has achieved promising performance, inter-subject face variations still pose a big challenge to both frame and video-based FER. To tackle the problem caused by inter-subject face variations, Li et al. [19] used two different CNNs with their corresponding

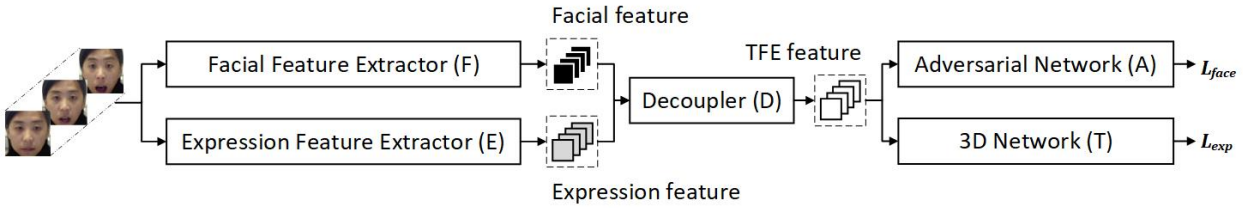

Fig. 2. Typical Facial Expression Network (TFEN) structure. The Facial Feature Extractor and the Expression Feature Extractor extract facial features and expression features from the input videos, respectively. The Decoupler decouples facial features from the expression features to construct typical facial expression (TFE) features. The 3D Network extracts spatial-temporal information of TFE features integrally. The Adversarial Network performs facial recognition and measures the residual influence of facial features.

data to learn the emotion and facial features separately. The features learned by these two networks were combined and fed into the subsequent fully connected layers. The whole model was trained with facial expression training data and used for frame-based FER. As the model alleviated the influence from facial features to the expression description, this approach obtained superior performance in terms of recognition accuracy compared to other state-of-the-art frame-based FER methods.

In our previous work, we attempted to minimize the influence of inter-subject face variations for video-based FER by employing a decoupler to decouple facial feature information from expression features [20]. The underlying idea of our previous work was innovative, but our preliminary network design failed to fulfill its full potential. In this work, we perfect that idea by designing a mechanism to measure the residual influence from facial features after decoupling and feed this measurement back to the networks to update parameters.

In summary, a review of extant literature shows that FER research still faces two challenges: (1) how to take advantage of deep learning methods to extract spatial-temporal features effectively and efficiently from expression video; and (2) how to alleviate the negative influence caused by inter-subject variability.

3 Approach

Herein, we propose the Typical Facial Expression Network (TFEN) for video-based FER. TFEN employs a facial feature decoupler to alleviate the influence of inter-subject facial variations and a 3D CNN based network to capture the spatial-temporal structure of facial expression in video. We feed the decoupled expression features or TFE features instead of raw images into the 3D Network to simplify the 3D CNN module and reduce the risk of overfitting on small datasets.

Differing from our previous work [20], this improved version employs a face classifier as the Adversarial Network to measure the performance of facial feature decoupling. We design a unique adversarial algorithm to improve the performance of the whole network. In our adversarial algorithm, we train the Expression Feature Extractor, Decoupler, and 3D Network to achieve the best FER performance and simultaneously train the Adversarial Network to help the entire network in minimizing the influence from facial characteristics. Specifically, the Adversarial Network's facial recognition result, as a measurement of the influence from facial characteristics, is fed back to the Expression Feature Extractor, Decoupler, and 3D Network to update their parameters to improve their performance. These improvements greatly enhance the performance of the proposed network.

3.1 Overview

Our TFEN is composed of five modules: Facial Feature Extractor, Expression Feature Extractor, Decoupler, 3D Network and Adversarial Network. The TFEN architecture is shown in Fig. 2. The Facial Feature Extractor is used to extract pure facial features that are normally used for facial recognition, while the Expression Feature Extractor characterizes and extracts expression features from frames of the input video. The Decoupler filters out facial features and generates the typical facial expression (TFE) features to minimize the influence from each individual's unique facial characteristics.

The 3D Network extracts the spatial-temporal information from the constructed TFE features. The Adversarial Network is essentially a face classifier. It is regarded as an adversary by the Decoupler and 3D Network in order to enhance the efficiency of facial feature decoupling. For the convenience of description, we use (F) to denote the Facial Feature Extractor, (E) for the Expression Feature Extractor, (D) for the Decoupler, (A) for the Adversarial Network, and (T) for the 3D Network. The parameter configuration of our TFEN is shown in Table 1. The total number of parameters in our proposed TFEN is 32.386 M.

3.2 Facial Feature Extractor

To construct the Facial Feature Extractor for the proposed TFEN, we first pretrain a facial recognition network with four blocks of Se-ResNet-18 [21] to recognize faces, as shown in Fig. 3. After pretraining, the first two blocks are used as the Facial Feature Extractor (F) as shown in Fig. 3. Parameters for (F) are constant throughout the training of the entire TFEN. The other two blocks form a face classifier and function as the Adversarial Network (A) in TFEN. Unlike network (F), the parameters for (A) are updated during training.

The pretraining of the facial recognition network in Fig. 3 uses training images (frames) of all subjects with face labels. Frames corresponding to the same subject are randomly split into training and test sets with a 4:1 ratio. The size of the output facial feature matrix is $28\times28\times128$, in which

TABLE 1
Details of the TFEN parameter configuration.

Output size	Facial Feature Extractor & Expression Feature Extractor	Decoupler	Adversarial Network	3D Network
112×112	$conv$, 7×7, 64, stride2 $max-pooling$, 3×3, stride2	\	\	\
56×56	$\begin{bmatrix} conv, 3\times3, 64 \\ conv, 3\times3, 64 \\ fc, [4, 64] \end{bmatrix} \times 2$	\	\	\
28×28	$\begin{bmatrix} conv, 3\times3, 128 \\ conv, 3\times3, 128 \\ fc, [8, 128] \end{bmatrix} \times 2$	$\begin{bmatrix} conv, 1\times1, 64 \\ conv, 1\times1, 64 \\ conv, 1\times1, 128 \end{bmatrix} \times 3$	\	\
14×14	\	\	$\begin{bmatrix} conv, 3\times3, 256 \\ conv, 3\times3, 256 \\ fc, [16, 256] \end{bmatrix} \times 2$	$\begin{bmatrix} conv, 3\times3\times3, 256 \\ conv, 3\times3\times3, 256 \end{bmatrix} \times 2$
7×7	\	\	$\begin{bmatrix} conv, 3\times3, 512 \\ conv, 3\times3, 512 \\ fc, [32, 512] \end{bmatrix} \times 2$	$\begin{bmatrix} conv, 3\times3\times3, 512 \\ conv, 3\times3\times3, 512 \end{bmatrix} \times 2$
1×1	Global average pooling dropout (with probability 0.2) 512-dimension fc			

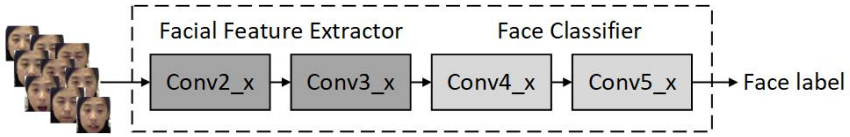

Fig. 3. The structure of the pretrained facial recognition network. Four blocks of Se-ResNet-18 [21] serve as the backbone of our facial recognition network. Among the four blocks, Conv2_x and Conv3_x blocks comprise the Facial Feature Extractor while Conv4_x and Conv5_x blocks make up the Face Classifier.

28 is the width and height of the feature maps and 128 is the number of feature maps. These facial features are decoupled from the expression features in the Decoupler.

3.3 Expression Feature Extractor

The Expression Feature Extractor (E) consists of the first two blocks of Se-ResNet-18 (Conv2_x and Conv3_x) [21] that are pretrained with ImageNet data. Its parameters are updated during training using the loss functions from the Adversarial Network (A) and the 3D Network (T) with our adversarial algorithm. Details of this process are discussed in Section 3.7. The size of the expression features output from the Expression Feature Extractor (E) is 28×28×128. This size is the same as the output facial feature matrix since the modules share the same network structure as explained in Section 3.2 and Fig. 3.

3.4 Decoupler

To alleviate the influence of facial features, we design a decoupler module to filter those influences and construct refined expression features or typical facial expression (TFE) features. As shown in Fig. 4, the Decoupler is composed of three groups of three convolutional layers and a fusion operator.

The three convolutional layers of each group use 128, 64, and 128 kernels of size 1×1 with the stride of 1 pixel, respectively. The two inputs to the Decoupler are the facial features and expression features from the Facial Feature Extractor (F) and the Expression Feature Extractor (E). We denote f_{FACE}, f_{EXP} and f_{TFE} as the three groups of convolutional layers in the Decoupler. The Decoupler performs

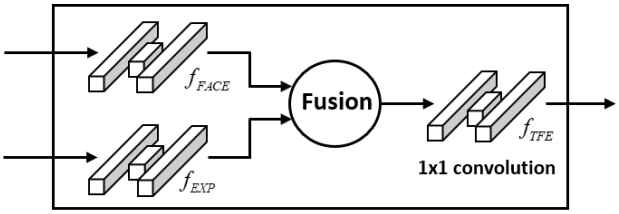

Fig. 4. Structure of the Decoupler. The Decoupler is composed of three groups of convolutional layers f_{FACE}, f_{EXP}, and f_{TFE}. Each group contains three 1 × 1 convolution layers. f_{FACE} and f_{EXP} refine features from the Facial Feature Extractor (F), and Expression Feature Extractor (E), respectively. Fusion performs an element-wise addition. f_{TFE} further refines the fused features and outputs the TFE features.

an element-wise addition of corresponding channels, and uses the third convolution group f_{TFE} and our adversarial algorithm to filter out the influence of facial features from expression features. The output of the Decoupler is typical facial expression features (or refined expression features) denoted as TFE. The TFE value at position (x, y) in the j^{-th} feature channel and i^{-th} frame can be formulated as Eq. (1).

$$TFE_{ij}(x,y) = f_{TFE}(f_{EXP}(E_{ij}(x,y)) + f_{FACE}(F_{ij}(x,y))) \quad (1)$$

The output of the Decoupler contains 128 feature maps (of size 28×28) for each input video frame.

3.5 Adversarial Network

We use the pre-trained face classifier (Conv4_x, and Conv5_x) in Fig. 3 as the Adversarial Network (A) with

the pretrained parameters as its initial parameters. It is employed here to measure the efficiency of facial feature decoupling. This Adversarial Network receives the decoupled features or TFE features from the Decoupler (D) and performs facial recognition.

We use the Adversarial Network to evaluate the residual influence of facial features in TFE features. Specifically, the lower the Adversarial Network's facial recognition accuracy, the less the residual influence from the facial features after decoupling. However, the fact that the Adversarial Network outputs almost equal recognition score for each face class indicates that very little residual influence of facial features is presented in TFE features. In order to help the Decoupler to alleviate the influence of facial features, the facial recognition loss of the Adversarial Network is back-propagated to update parameters with an adversarial training algorithm when we train the Expression Feature Extractor, the Decoupler, and the 3D Network. We provide a detailed description of the adversarial training algorithm in Section 3.7.

3.6 3D Network

The 3D Network (T) in Fig. 2 is a 3D convolutional network-based module. The 3D Network captures the spatial and temporal information from the TFE features. The 3D convolution can be regarded as an extension of 2D convolution. Eq. (2) shows the operation of this 3D convolution:

$$V_{ij}^{xyz} = \sigma(\sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} \cdot V_{(i-1)m}^{(x+p)(y+q)(z+r)} + b_{ij}), \tag{2}$$

where V_{ij}^{xyz} is the value at position (x, y, z) on the j^{-th} feature map in the i^{-th} layer; w_{ij}^{pqr} represents the $(p, q, r)^{-th}$ filter unit value connected to the m^{-th} feature map. $\sigma(\cdot)$ denotes a nonlinear function $\sigma(x) = max(0, x)$.

We choose a 3D-ResNet18 [22] model pre-trained on the Kinetics dataset [23] as the backbone of the 3D Network. The model is based on the 3D-ResNet18 in [24]. Two blocks of 3D-ResNet18 (Conv4_x and Conv5_x) are applied to the 3D Network. A 3D average pooling layer is used after the 3D Network with a kernel size of 2×7×7 and outputs a 512-length expression vector. The 3D Network (T) is connected to the output of the Decoupler (D). This arrangement allows the 3D Network to extract spatial and temporal information more efficiently by using the high-level TFE features output from the deep 2D CNNs (the Facial Feature Extractor and the Expression Feature Extractor) and the Decoupler. It also simplifies the 3D CNN model to avoid overfitting for small datasets.

3.7 Adversarial Training Algorithm

The adversarial training strategy of our TFEN network is designed as an adversarial process. The networks $[E, D, T]$ and the network (A) play a two-player minimax game with the value function $L(A, [E, D, T])$:

$$\min_{[E,D,T]} \max_{A} L(A, [E, D, T]) = \alpha L_{exp}(T(D(z)), y^{exp}) - L_{face}(A(D(z)), y^{face}), \tag{3}$$

where z represents the sample minibatch of M examples from the training dataset, $D(z)$ represents the features output from the Decoupler (D), and y^{exp} and y^{face} represent the corresponding expression and face labels of the minibatch examples. α is a hyper-parameter for adjusting the weights between the facial expression recognition error L_{exp} and the face identification error L_{face}.

The adversarial training algorithm is composed of two alternating iterative steps. In the first step of training, we train the Adversarial Network (A), and freeze the parameters of the networks (E), (D), and (T). As the facial expression recognition error L_{exp} is fixed, we want to improve the recognition accuracy of the Adversarial Network and decrease the face identification error L_{face}, which corresponds to maximizing the loss function $L(A, [E, D, T])$. In the second step, we train the networks (E), (D), and (T), and keep the parameters of the network (A) unchanged. We want to improve the accuracy of facial expression recognition (or decrease the facial expression recognition error L_{exp}) and make the Decoupler produce features that are difficult for the Adversarial Network to recognize. This corresponds to minimizing the loss function $L(A, [E, D, T])$. As the fact that the Adversarial Network outputs almost equal recognition scores for each face class indicates that very little residual influence of the facial features is present in TFE features, back-propagating the loss for parameter updating in the second step helps the Decoupler alleviate the influence of facial features from TFE features.

In the proposed adversarial training algorithm, we use cross-entropy with softmax as the loss function for facial expression recognition, denoted as $L_{exp}(g^{exp}, y^{exp})$, where g^{exp} represents the expression class score output from the 3D Network (T), and y^{exp} represents its corresponding label. This can be formulated as Eq. (4):

$$L_{exp}(g^{exp}, y^{exp}) = -\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{N_{exp}} y_{i,k}^{exp} \log(g_{i,k}^{exp}), \tag{4}$$

where M is the minibatch size, N_{exp} is the number of expression classes, $y_{i,k}^{exp}$ represents the expression label of the i^{-th} sample in the minibatch corresponding to the k^{-th} expression class, and g_i^{exp} represents the k^{-th} expression class scores of the i^{-th} sample in the minibatch.

We choose a mean square error (MSE)-based loss function for face identification and name it $L_{face}(g^{face}, y^{face})$, where g^{face} represents the face recognition score output from the Adversarial Network (A), and y^{face} represents its corresponding face label. Please note we use different labels for face identification in the two steps of the adversarial training algorithm.

In the first step of training, we set the face labels as one-hot vectors (1×N vectors). The face identification loss we used can be formulated as Eq. (5):

$$\begin{aligned} L_{face}(g^{face}, y^{face\_1}) &= \frac{1}{M} \sum_{i=1}^{M} (\sum_{k=1}^{N} \left\| y_{k,i}^{face\_1} - G_{k,i}^{face} \right\|_2) \\ &= \frac{1}{M} \sum_{i=1}^{M} (\sum_{k=1}^{N} \left\| y_{k,i}^{face\_1} - \frac{1}{S} \sum_{j}^{S} g_{j,k,i}^{face} \right\|_2). \end{aligned} \tag{5}$$

In Eq. (5), M denotes the minibatch size, N denotes the number of face labels in each expression dataset, and S denotes the number of frames sampled from the input video. $y_{k,i}^{face\_1}$ represents the i^{-th} minibatch example's face label corresponding to the k^{-th} person in the dataset, $G_{k,i}^{face}$ represents the average pooling result of S frames' face recognition scores, and $g_{j.k.i}^{face}$ represents the i^{-th} minibatch example's face recognition score corresponding to the j^{-th} frame and the k^{-th} person. A decrease in mean square error indicates that the outputs of the Adversarial Network (A) tend to recognize the correct face labels.

In the second step of training, we want the Adversarial Network to output an equal recognition score for each face class, not just make a misclassification. We use $(\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N})$ as the label for each training sample in face recognition, where N is the number of face classes. The face identification loss is formulated as Eq. (6):

$$\begin{aligned} L_{face}(g^{face}, y^{face\_2}) &= -\frac{1}{M}\sum_{i=1}^{M}(\sum_{k=1}^{N}\left\|\frac{1}{N} - G_{k,i}^{face}\right\|_2) \\ &= -\frac{1}{M}\sum_{i=1}^{M}(\sum_{k=1}^{N}\left\|\frac{1}{N} - \frac{1}{S}\sum_{j}^{S} g_{j,k,i}^{face}\right\|_2) \end{aligned} \quad (6)$$

Please note the minus sign at the right-hand side of Eq. (6). As the mean square error decreases, the loss of $L_{face}(g^{face}, y^{face\_2})$ increases, and the loss function $L(A, [E, D, T])$ tends to be minimized, which means the Adversarial Network (A) tends to output equal recognition scores for each face class. This result indicates fewer residual facial features in the output from the Decoupler, as it is more difficult for the Adversarial Network to classify the input features.

The adversarial training process is given in Algorithm 1.

4 Experiments

We evaluated the classification accuracy of our proposed method on three well-known lab-setting datasets: the Extended Cohn-Kanade (CK+) dataset [25], the MMI Facial Expression dataset [26], the Oulu-CASIA dataset [27], and a large-scale "in the wild" FER dataset called DFEW [34]. We compared our approach with other state-of-the-art video-based FER algorithms. We also investigated the effect of the adversarial algorithm and facial features on our TFEN.

4.1 Dataset

4.1.1 CK+

The CK+ dataset contains 327 labeled image sequences with seven expressions: anger, contempt, disgust, fear, happiness, sadness and surprise. All expression sequences in CK+ begin with a neutral expression and end with the peak state of the emotion. The image sequences in the CK+ were divided into 10 subsets using their IDs in ascending order to avoid subjects appearing in both the training and testing sets. Nine subsets were used for training our networks, and the remaining subset was used for validation.

4.1.2 MMI

The MMI dataset is composed of 208 image sequences labeled with six expressions (surprise, happiness, sadness, anger, fear, and disgust). Unlike the CK+ dataset, each of the sequences in the MMI dataset reflects the whole dynamic facial expression including the neutral, apex, and offset phases. Compared with other facial expression datasets, MMI is more challenging as some subjects wear accessories, and the number of images in the sequences is very limited. To perform 10-fold cross validation, we divided all sequences into 10 subsets in ascending order of their IDs.

4.1.3 Oulu-CASIA

The Oulu-CASIA dataset consists of six expressions from 80 people between 23 to 58 years old. Near Infrared (NIR) and visible light (VIS) cameras running at 25 frames per second were used to capture videos for this dataset. All expressions were captured under three different illumination conditions: normal, weak, and dark. There are 480 videos (80 subjects with 6 expressions each) for each illumination condition and each imaging system, for a total of 2880 video sequences in the dataset. We selected 480 videos captured by the VIS system under normal indoor illumination for experiments. The 80 subjects were divided into 10 groups in ascending order of their IDs for 10-fold cross validation experiments.

4.1.4 DFEW

The DFEW dataset contains 16372 video clips collected in unconstrained real-world scenarios, including extreme illuminations, severe occlusions, and capricious pose changes. All video clips are labeled with seven basic expressions: anger, disgust, fear, happiness, neutral, sadness, and surprise. The DFEW dataset is targeted at facial expression recognition in practical applications instead of controlled environments.

Following the same method used by Jiang et al. [34], we selected 11697 video clips annotated with single label in the DFEW dataset in our experiments. We directly utilized the preprocessed frames that were publicly released by the authors. It is challenging to perform experiments on such a large dataset. We performed stratified sampling on all single-labeled samples according to the proportion of each expression in the DFEW dataset. We split the dataset into 12 subsets and performed a subject-independent 10-fold cross validation on each subset. We finally obtained the recognition accuracy on the whole DFEW dataset by averaging the recognition accuracies of the 12 subsets. Considering that face labels are not provided by the DFEW dataset, we assigned unique face labels to all image sequences in each subset in the pretraining stage of the face recognition network.

4.2 Preprocessing and Data Augmentation

In order to avoid the influence from the video background, we detected and cropped the face in the input video for experiments using the SDM algorithm [3], which extracts 49 facial landmarks to determine the face region. The cropped faces were then rescaled to a fixed size of 244×244. CK+ includes both greyscale and color videos. We converted all color videos in CK+ to greyscale for data consistency.

Algorithm 1: TFEN Training

Input: Image sequences from the training set, learning rate (η), training epoch $(Epoch)$, Facial Feature Extractor (F), Adversarial Network (A) , 3D Network (T) , Expression Feature Extractor (E) and Decoupler (D) , the number of training steps (S) applied to the Adversarial Network.

Pre-train a facial recognition network with labels to obtain Facial Feature Extractor (F) and Adversarial Network (A);

Fix the parameters of Facial Feature Extractor (F);

for *i=1:Epoch* **do**

  Sample minibatch of M examples $\left\{z^{(1)}, ..., z^{(M)}\right\}$ from training set;

  if $i\%S == 0$ **then**

    Update θ_A (the parameters of network (A)) to maximize $L(A,[E,D,T]) = -\alpha\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{N_{exp}} y_{i,k}^{exp}\log(g_{i,k}^{exp}) - \frac{1}{M}\sum_{i=1}^{M}(\sum_{k=1}^{N}\left\|y_{k,i}^{face-1} - \frac{1}{S}\sum_{j}^{S} g_{j,k,i}^{face}\right\|_2)$;

    $\theta_A \leftarrow \theta_A - \eta L(A,[E,D,T])$;

  else

    Update θ_E, θ_D and θ_T (the parameters of network (E), (D), and (T), respectively) to minimize $L(A,[E,D,T]) = -\alpha\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{N_{exp}} y_{i,k}^{exp}\log(g_{i,k}^{exp}) + \frac{1}{M}\sum_{i=1}^{M}(\sum_{k=1}^{N}\left\|\frac{1}{N} - \frac{1}{S}\sum_{j}^{S} g_{j,k,i}^{face}\right\|_2)$;

    $\theta_E \leftarrow \theta_E - \eta L(A,[E,D,T])$;

    $\theta_D \leftarrow \theta_D - \eta L(A,[E,D,T])$;

    $\theta_T \leftarrow \theta_T - \eta L(A,[E,D,T])$;

  end

end

During the training process, we applied two data augmentation strategies: random cropping and random horizontal flipping. For random cropping, we randomly extracted a 224×224 region from input images (center crop in the testing phase). For random horizontal flipping, the cropped images were flipped horizontally with a probability of 0.5.

4.3 Experiment Setting

In order to improve computational efficiency, we limited the input to the network to seven frames rather than the whole video. In the training phase, all image sequences in the dataset were divided into seven segments with equal duration. We generated the input for training by randomly choosing one frame from each of these seven segments. We used the middle frame of each segment for testing.

In the training phase, we set the batch size to 32, and the momentum to 0.9. The learning rate for experiments on the CK+, MMI, and Oulu-CASIA datasets was initialized to 10e-3 and set to decrease to 1/10 every 100 epochs. For the experiment on the DFEW dataset, the learning rate was decreased to 1/2 every 100 epochs. The whole training procedure was set to stop at 300 epochs for the CK+, MMI, and Oulu-CASIA datasets and at 200 epochs for the DFEW dataset. The number of training steps applied to the Adversarial Network was set to 3. We also set the weight decay to 0.008 to avoid overfitting. The hyper-parameter α in the loss function was set to 5.

4.4 Results on CK+

We compared our method to the state-of-the-art methods with 10-fold cross validation using CK+. The accuracy is shown in Table 2. In the CK+ dataset, the average standard deviation of accuracy across the 10-fold validation was 1.609%. For the CK+ dataset, among all traditional methods for facial expression recognition, STM-ExpLet [32] achieved the best accuracy of 94.19%. Zhang et al. [8] developed a deep evolutional spatial-temporal network composed of PHRNN and MSCNN to extract temporal features and spatial features separately, and offered the best overall performance prior to the method we present here.

TABLE 2
Overall accuracy on the CK+ dataset. (All methods reported in this table used image sequences as input.)

Method	Descriptor	Accuracy
Liu et al. [18]	3DCNN	85.9%
Klaser et al. [12]	HOG 3D	91.44%
Liu et al. [18]	3DCNN-DAP	92.4%
Walecki et al. [30]	VSL-CRF	93.9%
Liu et al. [32]	STM-ExpLet	94.19%
Jung et al. [7]	CNN-DNN	97.25%
Kuo et al. [29]	CNN	98.41%
Zhang et al. [8]	PHRNN-MSCNN	98.50%
TFEN	**TFEN**	**98.78%**

Table 2 shows that our method obtained the accuracy of 98.78% on CK+, which is 4.59% higher than STM-ExpLet [32] and a 0.28% improvement compared to the work of Zhang et al. [8]. Further comparison discovered that the architecture employed in the work of Zhang et al. [8] fed all frames to PHRNN, which led to the increase of computational cost. Additionally, the algorithm in the work of Zhang et al. [8] fed only the last frame (the known frame with the expression at the peak state) to MSCNN. Its performance would have suffered if the input video did not end at the peak state of expression or did not know which frame represents the peak state. In contrast, our TFEN divided the input video into seven segments and randomly sampled one frame of each segment for model training. In the stage of model testing, TFEN used the middle frame of each segment. The results in terms of accuracy demonstrate the better adaptability of TFEN for dynamic FER tasks.

TABLE 3
Confusion matrix of TEFN results on the CK+ dataset (%). The labels in the leftmost column and on the top represent the ground truth and prediction results, respectively.

	An	Co	Di	Fe	Ha	Sa	Su
An	**95.56**	0	2.22	0	0	2.22	0
Co	0	**100**	0	0	0	0	0
Di	0	0	**100**	0	0	0	0
Fe	0	0	0	**96**	4	0	0
Ha	0	0	0	0	**100**	0	0
Sa	3.57	0	0	0	0	**93.43**	0
Su	0	0	0	0	0	0	**100**

Table 3 shows the confusion matrix of our experiments on the CK+ dataset. In Table 3, An, Co, Di, Fe, Ha, Sa, and Su represent anger, contempt, disgust, fear, happiness, sadness, and surprise, respectively. Our algorithm achieved 100% accuracy for Co, Di, Ha, and Su. The results proved the comprehensive performance of our TEFN on CK+.

4.5 Results on MMI

Similar to the experiments performed on the CK+ dataset, we compared our recognition accuracy with other state-of-the-art methods on the MMI dataset. The results are shown in Table 4. On the MMI dataset, the average standard deviation of accuracy across the 10-fold validation was 2.954%.

TABLE 4
Overall accuracy on the MMI dataset. (All methods reported in this table used image sequences as input.)

Method	Descriptor	Accuracy
Klaser et al. [12]	HOG 3D	60.89%
Jung et al. [7]	CNN-DNN	70.24%
Liu et al. [32]	STM-ExpLet	75.12%
Zhang et al. [8]	PHRNN-MSCNN	81.18%
TFEN	**TFEN**	**81.73%**

Table 4 shows that TFEN achieved the accuracy of 81.73% on the MMI dataset, which is 20.84% higher than the traditional HOG 3D method [12]. Compared with other deep learning-based methods, the recognition accuracy of TFEN also improved from 0.55% to 11.49%.

TABLE 5
Confusion matrix of TEFN results on the MMI dataset (%). The labels in the leftmost column and on the top represent the ground truth and prediction results, respectively.

	An	Di	Fe	Ha	Su	Sa
An	**93.94**	0	0	3.03	3.03	0
Di	3.13	**71.88**	0	15.63	6.25	3.13
Fe	7.14	7.14	**42.86**	7.14	3.57	32.14
Ha	0	0	0	**97.62**	0	2.38
Su	21.88	0	2.5	0	**78.13**	0
Sa	0	0	4.88	0	2.44	**92.68**

The confusion matrix of our experiments on the MMI dataset is shown in Table 5. TFEN performed well in recognizing anger, happiness, and surprise. However, the accuracy for fear was only 42.86%, which was dramatically lower than other emotions. Over 30% of the samples labeled with fear were misclassified as sadness. Manual examination of the MMI dataset showed that samples of these two emotions shared some similar appearances, including dropping of the eyebrows, wrinkling of the nose, and lowering of the upper lip.

4.6 Results on Oulu-CASIA

We also compared our method with the state-of-the-art methods on the Oulu-CASIA dataset. On the Oulu-CASIA dataset, the average standard deviation of accuracy across the 10-fold validation was 1.495%. Table 6 shows the work of Guo et al. [33] achieved a recognition accuracy of 75.52%, the best performance among the traditional methods. Our method achieved a recognition accuracy of 91.67%, much higher than the traditional methods and around 5% higher than most other deep learning-based methods. Although the accuracy of TFEN was the same as the work of Kuo et al. [29] on the Oulu-CASIA dataset, our method achieved 0.74% and 0.55% higher accuracy than the results reported by Kuo et al. [29] on the CK+ and MMI datasets, respectively, as shown in Tables 2 and 4. Experimental results show that our approach is an effective solution for video-based facial expression recognition.

TABLE 6
Overall accuracy on the Oulu-CASIA dataset. (All methods reported in this table used image sequences as input.)

Method	Descriptor	Accuracy
Klaser et al. [12]	HOG 3D	70.63%
Zhao et al. [13]	AdaLBP	73.54%
Jung et al. [7]	DNN	74.17%
Jung et al. [7]	CNN	74.38%
Guo et al. [33]	Atlases	75.52%
Jung et al. [7]	CNN-DNN	81.46%
Yu et al. [31]	DCPN	86.23%
Zhang et al. [8]	PHRNN-MSCNN	86.25%
Kuo et al. [29]	CNN	91.67%
TFEN	**TFEN**	**91.67%**

We further evaluated the performance of our method with cross validation on the Oulu-CASIA dataset. Table 7 shows the confusion matrix of our experiments on Oulu-CASIA. Our algorithm performed well in recognizing anger, happiness, fear, and surprise. We observed that the accuracy for disgust was much lower than other emotions. Most misclassifications labeled as 'disgust' and 'sadness' were recognized as 'anger'. Manual examination confirmed that a number of facial expressions in Oulu-CASIA labeled as 'disgust' and 'sadness' look extremely similar to the samples labeled as 'anger'.

TABLE 7
Confusion matrix of TEFN results on the Oulu-CASIA dataset (%). The labels in the leftmost column and on the top represent the ground truth and prediction results, respectively.

	An	Di	Fe	Ha	Su	Sa
An	**92.5**	2.5	1.25	1.25	0	2.5
Di	8.75	**85**	1.25	1.25	0	3.75
Fe	2.5	1.25	**87.5**	5	2.5	1.25
Ha	0	1.25	1.25	**97.5**	0	0
Su	0	0	2.5	0	**97.5**	0
Sa	10	0	0	0	0	**90**

4.7 Results on DFEW

The preceding small-scale datasets were collected in controlled environments. In order to evaluate the performance

of our TFEN on dataset in uncontrolled environments, we also performed experiments on a large-scale real world dataset DFEW [34]. We compared the performance of TFEN on the DFEW dataset with other methods, with results given in Table 8. As we performed the experiments on 12 subsets of the DFEW dataset, we calculated the average standard deviation of accuracy across 10-fold validation in each subset and obtained 1.568% by averaging the averaged standard deviation of each subset. Experimental results (Table 8) on the DFEW dataset show that our TFEN obtained better or comparable performance in terms of weighted average recall compared with the state-of-the-art methods reported in the work of Jiang et al. [34].

TABLE 8
Overall accuracy on the DFEW dataset. Unweighted average recall (UAR) is the accuracy per class divided by the number of classes without considerations of numbers of samples per class. Weighted average recall (WAR) is the recognition accuracy. (All methods reported in this table used image sequences as input.)

Method	Descriptor	UAR	WAR
Jiang et al. [34]	ResNet18+LSTM	42.86%	53.08%
Jiang et al. [34]	ResNet18+LSTM+EC-STFL	43.60%	54.72%
Jiang et al. [34]	3D ResNet18	44.73%	54.98%
Jiang et al. [34]	C3D+EC-STFL	45.10%	55.50%
Jiang et al. [34]	R3D18+EC-STFL	45.05%	56.19%
Jiang et al. [34]	I3D-RGB18+EC-STFL	45.05%	56.19%
Jiang et al. [34]	P3D+EC-STFL	45.22%	56.48%
Jiang et al. [34]	3D ResNet18+EC-STFL	45.35%	56.51%
TFEN	**TFEN**	**45.57**%	**56.60**%

Table 9 shows the confusion matrix of our experimental results on the DFEW dataset. TFEN performed well when recognizing happiness, neutral, and sadness expressions. However, the recognition accuracy for disgust was only 2.68%, which is far lower than other expressions. One possible reason is that the number of disgust samples in the DFEW dataset is only 1 percent of the total samples. TFEN misclassified disgust into other expressions because of the limited number of samples. A similar situation also occurred in the state-of-the-art method 3D Resnet18 + EC-STFL [34].

TABLE 9
Confusion matrix of TEFN results on the DFEW dataset (%). The labels in the leftmost column and on the top represent the ground truth and prediction results, respectively.

	An	Di	Fe	Ha	Ne	Sa	Su
An	**47.72**	0.88	3.59	7.46	21.51	10.73	8.11
Di	11.41	**2.68**	6.04	9.40	40.27	20.13	10.07
Fe	15.73	0.55	**23.15**	4.65	18.16	16.39	21.37
Ha	4.74	0.41	1.06	**75.21**	10.43	7.12	1.02
Ne	7.72	0.79	2.44	4.87	**66.15**	11.21	6.82
Sa	5.55	0.58	2.85	9.36	20.67	**56.24**	4.76
Su	7.90	0.75	5.11	2.59	25.80	10.01	**47.86**

4.8 Ablation Study

To validate the effectiveness of our proposed TFEN, we conducted an ablation study on the CK+, MMI, and Oulu-CASIA datasets.

We designed a TFEN-without-Decoupler model to explore the impact of the Decoupler in our proposed TFEN. The architecture is illustrated in Fig. 5. We removed TFEN's Decoupler and fused the facial and expression features by concatenation. We used a 1×1 convolutional layer and a batch normalization layer to match the number of input features required by the 3D Network and Adversarial Network. The setting of the remaining parts in TFEN-without-Decoupler was the same as TFEN.

To demonstrate the effect of the adversarial learning, we constructed a TFEN-without-adversarial model, as shown in Fig. 6. We removed the Adversarial Network and only fed the TFE features to the 3D Network to perform facial expression recognition. The optimization target was also changed from a two-player minimax function $L(A, [E, D, T])$ to only minimize the expression loss L_{exp}. The rest of the network settings of TFEN-without-adversarial model remained the same as TFEN.

We removed the Facial Feature Extractor and the Decoupler from TFEN and constructed a comparable network without the decoupling scheme to demonstrate the impact of decoupling the facial features from the expression features for expression recognition. The configuration of this network is illustrated in Fig. 7.

Finally, we replaced the 3D Network in TFEN with a 2D-CNN module to explore the effectiveness of capturing temporal relationships among input image sequences. The structure of the TFEN-without-3D-Network model is shown in Fig. 8. In TFEN-without-3D-Network, a 2D CNN module is formed by the Conv4_x, and Conv5_x blocks of Se-ResNet-18 [21]. The 2D CNN module recognizes the expression of each frame of the input sequence. The output of the 2D CNN module is obtained by averaging the recognition results of each frame.

In the ablation study, we preformed the same preprocessing, data augmentation, and hyper-parameter setting operations as before. The results are shown in Table 10.

TABLE 10
Accuracy comparison of different models in the ablation study

Model	CK+	MMI	Oulu-CASIA
TFEN	**98.78**%	**81.73**%	**91.67**%
TFEN-without-Decoupler	96.94%	77.88%	89.79%
TFEN-without-adversarial	92.46%	75.00%	88.46%
TFEN-without-facial-decoupling	97.61%	79.33%	89.58%
TFEN-without-3D-Network	89.60%	74.04%	82.29%

In Table 10, the accuracies of TFEN-without-Decoupler were 1.84%, 3.85%, and 1.88% lower than TFEN on the CK+, MMI, and Oulu-CASIA datasets, respectively. Omitting facial features decoupling degraded the performance of our TFEN method. This experiment demonstrates the importance of the Decoupler in the method.

In Table 10, the TFEN-without-adversarial obtained the accuracies of 92.46%, 75.00%, and 88.46% on CK+, MMI, and Oulu-CASIA, respectively. As the network has no feedback from the Adversarial Network, the efficiency of the Decoupler was much decreased as well as the overall performance of the network. This result shows the value of our adversarial training.

The experimental results of TFEN-without-Decoupler and TFEN-without-adversarial also show that in our proposed TFEN, the Decoupler needs to cooperate with the Adversarial Network. Without supervision from the Adversarial Network, the Decoupler has no feedback concerning

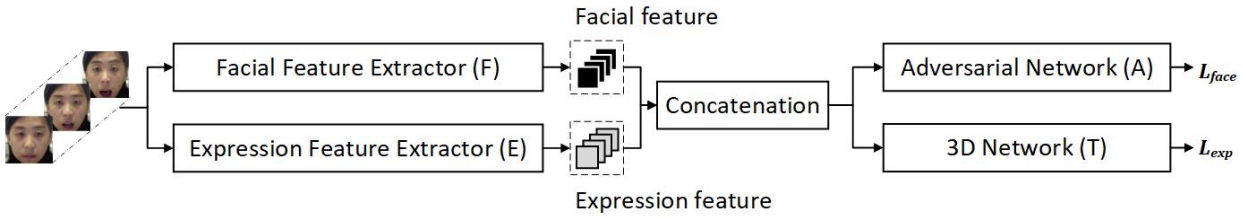

Fig. 5. Architecture of TFEN-without-Decoupler. Facial features and expression features are concatenated but no Decoupler follows the concatenation. A 1×1 convolutional layer and a batch normalization layer were used to match the number of input features required by the 3D Network and the Adversarial Network.

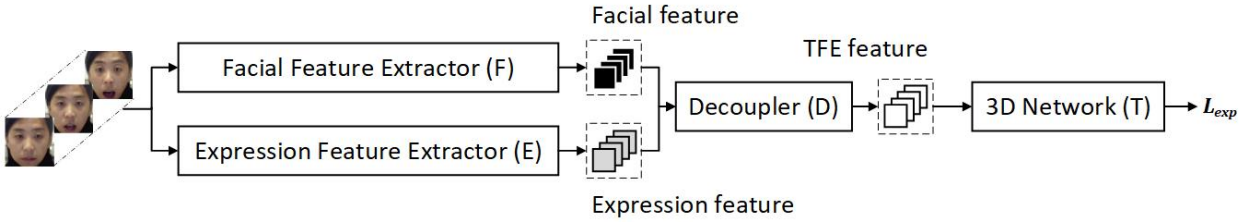

Fig. 6. Architecture of TFEN-without-adversarial. Compared with the TFEN structure, we removed the Adversarial Network and trained the network with only the expression recognition loss.

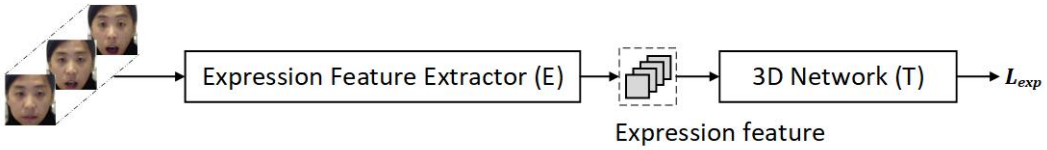

Fig. 7. Architecture of TFEN-without-facial-decoupling. Compared with the TFEN structure, we removed both the Adversarial Network and the Decoupler. Only a 3D Network is used to recognize the facial expression.

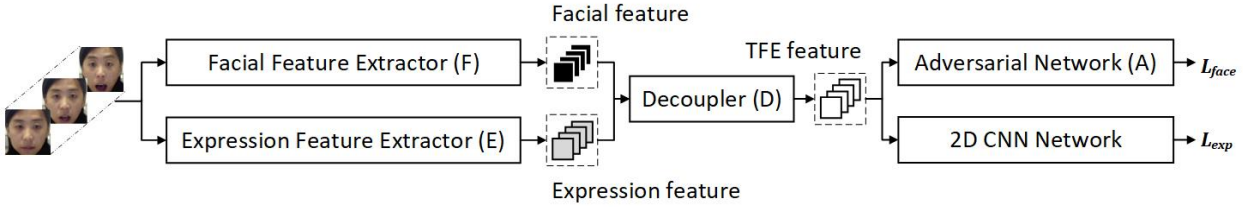

Fig. 8. Architecture of TFEN-without-3D-Network. Compared with the TFEN structure, we replaced the 3D Network with a 2D CNN module to capture the spatial relationships of input image sequence.

the residual influence from the facial features after decoupling. Without the Decoupler, the Adversarial Network would not employ the decoupled expression features (TFE features) to play the minimax game with the Facial Feature Extractor, the Decoupler, and the 3D Network, and the adversarial training algorithm would fail to alleviate the negative influence of facial features on FER. These results lead to the conclusion that omitting either the Decoupler or the Adversarial Network degrades TFEN performance.

Table 10 also shows that TFEN obtained accuracies that were 1.17%, 2.40%, and 2.09% higher than TFEN-without-facial-decoupling on CK+, MMI and Oulu-CASIA, respectively. In effect, TFEN-without-facial-decoupling behaves as a common video processing structure without special treatment for facial expression recognition. Decoupling the facial biometric features helps TFEN achieve better facial expression recognition accuracy.

Lastly, in Table 10, without the help of the 3D Network, which captures the temporal relationships among input image sequences, the accuracies of TFEN-without-3D-Network were 9.18%, 7.69%, and 9.38% lower than full TFEN on the three datasets, respectively. These results indicate that capturing spatial-temporal features from input image sequences improves the recognition accuracy of dynamic FER methods.

4.9 Visualization Results

To help understand what TFEN learns from the training data, we visualized the learned facial features, expression features, and TFE features. For each kind of feature, we calculated the mean value of all the feature maps output

from the Facial Feature Extractor (F), the Expression Feature Extractor (E), and the Decoupler (D). The results are shown in Fig. 9.

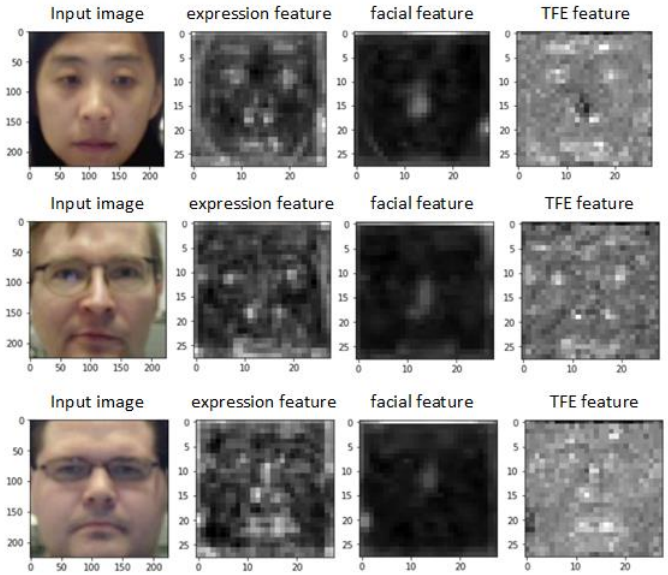

Fig. 9. Feature maps output from TFEN for visualization. The visualization results of expression features, facial features, and TFE features were obtained by evaluating the mean values of corresponding feature maps. We present the input images, facial expression features, facial features, and TFE features mentioned in Fig. 2.

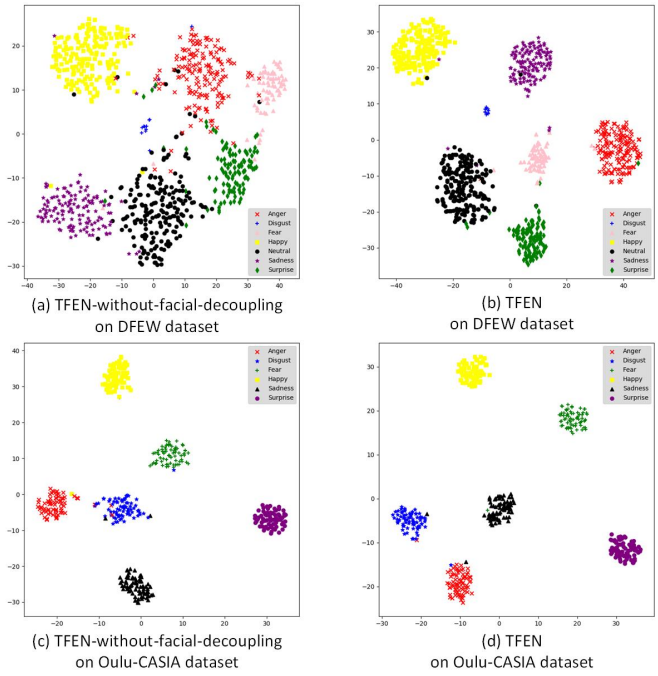

Fig. 10. t-SNE visualizations of the feature distributions for TFEN-without-facial-decoupling and TFEN on the DFEW and Oulu-CASIA datasets. (a) and (c) are the t-SNE visualization results of TFEN-without-facial-decoupling on two datasets. (b) and (d) are the t-SNE visualization results of TFEN. These are best viewed in color.

We selected three samples with same expression labels but belonging to different subjects to show the visualization results. The four columns from left to right are: input images, facial expression features, facial features, and TFE features, respectively. Visualization shows that the expression features pay more attention to the eyes, nose, mouth, and facial contour. The facial features focus more on the area around the nose. TFEN decouples the influence from the facial features from expression features by changing the weights of each feature.

In order to verify the contribution of decoupling facial features from expression features, we adopted t-SNE [35] to visualize the distributions of learned expression features. t-SNE [35] is a feature dimension reduction method used to visualize the features learned by deep learning models on a 2D plane. We visualized the distribution of features output from the 3D Network of both TFEN and TFEN-without-facial-decoupling with t-SNE. We chose the folds with the highest recognition accuracy on the DFEW and the Oulu-CASIA datasets for t-SNE visualization experiments. The results are shown in Fig. 10.

Fig. 10 shows that, on the DFEW dataset, the features learned by TFEN have smaller intra-class distances and larger inter-class distances compared to the features learned by TFEN-without-facial-decoupling. Fig. 10 also shows that, on the Oulu-CASIA dataset, the inter-class distances between the anger, disgust, and sadness expression classes of TFEN are larger than those of TFEN-without-facial-decoupling. The visualization results show that with facial feature decoupling, our proposed TFEN obtains better feature representation for FER.

We also evaluated the inter-class and intra-class distances of the t-SNE visualization results shown in Fig. 10. The results are shown in Tables 11 and 12. We calculated the mean value of the t-SNE features of all samples in each expression class and used it as the center for each class. The inter-class distances were evaluated by the Euclidean distance between two compared class centers. The intra-class distances were evaluated by the Euclidean distances between the t-SNE features of each sample and their corresponding class center.

Tables 11 and 12 show that in almost all cases the t-SNE visualization results of TFEN obtained a larger inter-class distance and a smaller intra-class distance compared with those of TFEN-without-facial-decoupling on the DFEW and the Oulu-CASIA datasets. The only exception is the variance of intra-class distance of Fe (Fear) obtained by TFEN in Table 12. This case is also revealed in Fig. 10 (d) where a t-SNE feature of Fe (Fear) fell into the cluster of Sa (Sadness).

5 Conclusion

In this study, we propose a Typical Facial Expression Network (TFEN) for video-based FER. We construct the TFE features by decoupling the facial features from expression features to solve the inter-subject variation problem. We explore the facial features encoded in facial expression and alleviate their influence on facial expression recognition. We apply an adversarial algorithm to train the network to improve the efficiency of facial feature decoupling. We build a deep network including 2D and 3D convolutions to integrate spatial and temporal information in an integrated

TABLE 11
The inter-class and intra-class distances of the t-SNE features for TFEN-without-facial-decoupling and TFEN on the DFEW dataset. In each table cell, the data to the left of '/' are the results obtained with TFEN-without-facial-decoupling, and the data to the right of '/' are the results given by TFEN. The inter-class distance is the Euclidean distance between two class centers. The intra-class distance was evaluated by the Euclidean distances between the t-SNE features and their corresponding class center.

	Inter-class distance							Intrar-class distance	
	An	Di	Fe	Ha	Ne	Sa	Su	Mean	Variance
An	-	20.08/45.84	20.18/28.19	36.62/71.99	31.60/54.75	49.59/37.64	24.46/39.25	7.96/5.69	24.44/10.17
Di	20.08/45.84	-	37.72/20.87	22.87/28.07	19.29/22.30	29.84/19.15	30.36/37.21	4.69/0.78	45.36/0.03
Fe	20.18/28.19	37.72/20.87	-	56.70/48.93	41.22/26.85	64.68/25.11	20.56/22.05	6.25/4.02	63.26/14.93
Ha	36.62/71.99	22.87/28.07	56.70/48.93	-	39.18/40.32	33.09/36.57	53.04/63.34	7.34/5.96	13.58/5.70
Ne	31.60/54.75	19.29/22.30	41.22/26.85	39.18/40.32	-	26.19/40.43	24.36/26.99	9.05/6.82	38.49/16.42
Sa	49.59/37.64	29.84/19.15	64.68/25.11	33.09/36.57	26.19/40.43	-	50.26/46.92	7.61/5.66	34.42/22.09
Su	24.46/39.25	30.36/37.21	20.56/22.05	53.04/63.34	24.36/26.99	50.26/46.92	-	7.19/5.02	38.09/23.31

TABLE 12
The inter-class and intra-class distances of the t-SNE features for TFEN-without-facial-decoupling and TFEN on the Oulu-CASIA dataset. In each table cell, the data to the left of '/' are the results obtained TFEN-without-facial-decoupling, and the data to the right of '/' are the results given by TFEN. The inter-class distance is the Euclidean distance between two class centers. The intra-class distance was evaluated by the Euclidean distances between the t-SNE features and their corresponding class center.

	Inter-class distance						Intrar-class distance	
	An	Di	Fe	Ha	Sa	Su	Mean	Variance
An	-	15.92/18.08	31.16/47.89	36.68/49.00	29.03/19.76	53.31/53.15	3.40/2.72	7.77/2.80
Di	15.92/18.08	-	19.19/43.52	35.28/35.71	20.26/26.26	37.40/59.89	3.14/2.66	4.61/3.03
Fe	31.16/47.89	19.19/43.52	-	24.84/25.32	35.63/30.60	31.33/33.17	3.08/2.94	4.86/14.43
Ha	36.68/49.00	35.28/35.71	24.84/25.32	-	55.41/40.27	55.47/57.33	3.31/2.50	13.41/1.24
Sa	29.03/19.76	20.26/26.26	35.63/30.60	55.41/40.27	-	36.71/34.48	3.23/3.01	6.97/8.17
Su	53.31/53.15	37.40/59.89	31.33/33.17	55.47/57.33	36.71/34.48	-	2.66/2.54	1.10/0.91

and optimized structure. Experimental results show that the proposed method achieves performance that exceeds or matches current state-of-the-art approaches on four widely used facial expression datasets. Our approach is an effective solution for video-based facial expression recognition.

ACKNOWLEDGMENTS

This work was supported by Guangzhou Municipal People's Livelihood Science and Technology Plan (201903010040), National Natural Science Foundation of China (61773413), Science and Technology Program of Guangzhou, China (202007030011), and the Science and Technology Planning Project of Guangdong Province of China (2019B070702004).

REFERENCES

[1] A. Vinciarelli, M. Pantic and H. Bourlard, "Social signal processing:Survey of an emerging domain,"*Image Vis. Comput.* vol. 27, no. 12, Nov. 2009, pp. 1743–1759.

[2] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K. M. Prkachin, "Automatically detecting pain using facial actions,"*In Proc. IEEE Int. Conf. Affective Comput. Intell. Interact. Workshops (ACII)* Sep. 2009, pp. 1–8.

[3] K. Dobs, J. Schultz, I. B¨ ulthoff, and J. L. Gardner, "Task-dependent enhancement of facial expression and identity representations in human cortex,"*NeuroImage* vol. 172, 2018, pp. 689–702.

[4] C. Shan, S. Gong, and P. W. McOwan, "Conditional mutual information based boosting for facial expression recognition," *In Proc. Brit. Mach. Vis. Conf. (BMVC)* Sep. 2005.

[5] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction," *In Proc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW),* Jun. 2003, p. 53.

[6] M. Pantic and L. J. M. Rothkrantz, "Facial action recognition for facial expression analysis from static face images," *IEEE Trans. Syst. Man, Cybern. B, Cybern.* vol. 34, no. 3, Jun. 2004, pp. 1449–1461.

[7] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," *In 2015 IEEE International Conference on Computer Vision (ICCV), IEEE* 2015, pp. 2983–2991.

[8] K. Zhang, Y. Huang, Y. Du, and L. Wang, "Facial expression recognition based on deep evolutional spatial-temporal networks," *IEEE Transactions on Image Processing* vol. 26, no. 9, 2017, pp. 4193–4203.

[9] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 31, no. 1, Jan. 2009, pp.39–58.

[10] P. Yang, Q. Liu, X. Cui, and D. N. Metaxas, "Facial expression recognition using encoded dynamic features," *In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* Jun. 2008, pp. 1–8.

[11] S. W. Chew, P. Lucey, S. Lucey, J. Saragih, J. F. Cohn, and S. Sridharan, "Person-independent facial expression detection using constrained local models," *In Proc. IEEE Int. Conf. Automat. Face Gesture Recognit. (FG)* Mar. 2011, pp. 915–920.

[12] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," *In Proc. Brit. Mach. Vis. Conf. (BMVC)* Sep. 2008, pp. 275.

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

[14] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," *In Proc. 15th ACM Int. Conf. Multimedia* Sep. 2007, pp. 357–360.

[15] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *Advances in Neural Information Processing Systems* vol. 27, 2014, pp. 1988–1996.

[16] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *Computer Science* 2015.

[17] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," *In IEEE Conference on Computer Vision and Pattern Recognition* 2014, pp. 1891–1898.

[18] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, "Deeply learning deformable facial action parts model for dynamic Expression Analysis". *In: Asian Conference on Computer Vision (ACCV)* 2015, pp. 143–157.

[19] M. Li, H. Xu, X. Huang, Z. Song, X. Liu, and X. Li, "Facial expression recognition with identity and emotion joint learning." *IEEE Transactions on Affective Computing* (2018): 1-1.

[20] J.N. Teng, D. Zhang, M. Li, and Y. D. Huang. Facial Expression Recognition with Identity and Spatial-temporal Integrated Learning. *In proceedings: 2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW 2019)* Cambridge, United Kingdom, 2019, pp. 100 – 104.
[21] J. Hu, L. Shen, G. Sun, "Squeeze-and-excitation networks," *In Proceedings of the IEEE conference on computer vision and pattern recognition* 2018, pp. 7132-7141.
[22] D. Tran, J. Ray, Z. Shou, S. Chang, M. Paluri, "Convnet architecture search for spatiotemporal feature learning." *CoRR* abs/1708.05038 (2017)
[23] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev and M. Suleyman, "The kinetics human action video dataset."*arXiv preprint* arXiv:1705.06950. May 2017.
[24] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?"*In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* 2018: pp. 6546-6555.
[25] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression." *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, IEEE* 2010. pp. 94–101.
[26] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietik¨ainen, "Facial expression recognition from near-infrared videos." *Image and Vision Computing* 29(9), 2011, pp. 607–619.
[27] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," *In 2005 IEEE International Conference on Multimedia and Expo (ICME), IEEE* 2005, pp. 5-11.
[28] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion." *Journal of personality and social psychology* vol. 17, no. 2, 1971.pp. 124–129.
[29] C. M. Kuo, S. H. Lai, and M. Sarkis, "A Compact Deep Learning Model for Robust Facial Expression Recognition," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* IEEE Computer Society, 2018.
[30] R. Walecki, O. Rudovic, V. Pavlovic, and M. Pantic, "Variable-state latent conditional random fields for facial expression recognition and action unit detection," *In Proc. IEEE Int. Conf. Workshops Automat. Face Gesture Recognit. (FG)* May 2015, pp. 1–8.
[31] Z. Yu, Q. Liu, and G. Liu, "Deeper cascaded peak-piloted network for weak expression recognition," *The Visual Computer* 2017, pp. 1–9.
[32] M. Liu, S. Shan, R. Wang, and X. Chen, "Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition," *In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* Jun. 2014, pp. 1749–1756.
[33] Y. Guo, G. Zhao, and M. Pietikäinen, "Dynamic facial expression recognition using longitudinal facial expression atlases," *In Proc. Comput. Vis. (ECCV)* 2012, pp. 631–644.
[34] X. Jiang, Y. Zong, W. Zheng, C. Tang, W. Xia, C. Lu and J. Liu, "DFEW: A Large-Scale Database for Recognizing Dynamic Facial Expressions in the Wild," *Proceedings of the 28th ACM International Conference on Multimedia* 2020, pp. 2881-2889.
[35] L.v.d. Matten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research* 2008, pp. 2579-2605.

Jianing Teng was born in Jilin, China, in 1995. He received his bachelor's degree from Sun Yat-sen University, China, in 2017. He is currently a postgraduate student in the school of Electronics and Information Technology, Sun Yat-sen University. His research interests include deep learning and image processing.

Dong Zhang received his B.S.E.E. and M. S. degrees from Nanjing University, China, in 1999 and 2003, respectively, and Ph.D. degree from Sun Yat-sen University, China, in 2009. He is currently an associate professor in the school of Electronics and Information Technology, Sun Yat-sen University. His research interests include image processing, pattern recognition, and information hiding.

Wei Zou received his bachelor's degree from Xidian University, China, in 2019. He is currently a postgraduate student in the school of Electronics and Information Technology, Sun Yat-sen University. His research interests include deep learning and facial expression recognition.

Ming Li received his Ph.D. in Electrical Engineering from University of Southern California in 2013. He is currently an Associate Professor of Electrical and Computer Engineering and Principal Research Scientist in the Data Science Research Center at Duke Kunshan University. He is also an Adjunct Professor in School of Computer Science at Wuhan University. His research interests are in the areas of audio, speech and language processing as well as multimodal behavior signal analysis and interpretation.

Dah-Jye Lee received his B.S. degree from National Taiwan University of Science and Technology in 1984, and M.S. and Ph.D. degrees in Electrical Engineering from Texas Tech University in 1987 and 1990, respectively. He also received his MBA degree from Shenandoah University, Winchester, Virginia, in 1999. He worked in the machine vision industry for 11 years prior to joining BYU in 2001. He is currently a Professor in the Department of Electrical and Computer Engineering at Brigham Young University. His research work focuses on object recognition, hardware implementation of real-time vision algorithms, and machine vision applications.