FaceTopoNet: Facial Expression Recognition Using Face Topology Learning

Mojtaba Kolahdouzi, Alireza Sepas-Moghaddam, Member, IEEE, and Ali Etemad, Senior Member, IEEE

Abstract—Prior work has shown that the order in which different components of the face are learned using a sequential learner can play an important role in the performance of facial expression recognition systems. We propose FaceTopoNet, an end-to-end deep model for facial expression recognition, which is capable of learning an effective tree topology of the face. Our model then traverses the learned tree to generate a sequence, which is then used to form an embedding to feed a sequential learner. The devised model adopts one stream for learning structure and one stream for learning texture. The structure stream focuses on the positions of the facial landmarks, whereas the main focus of the texture stream is on the patches around the landmarks to learn textural information. We then fuse the outputs of the two streams by utilizing an effective attention-based fusion strategy. We perform extensive experiments on four large-scale in-the-wild facial expression datasets—namely AffectNet, FER2013, ExpW, and real-world affective face database—and one lab-controlled dataset (Cohn–Kanade) to evaluate our approach. FaceTopoNet achieves state-of-the-art performance on three of the five datasets and obtains competitive results on the other two datasets. We also perform rigorous ablation and sensitivity experiments to evaluate the impact of different components and parameters in our model. Finally, we perform robustness experiments and demonstrate that FaceTopoNet is more robust against occlusions in comparison to other leading methods in the area.

Impact Statement—Systems that perform facial expression recognition can be used in a variety of applications in real-world problems such as human-machine interaction, robotics, mental health management systems, and more. Our model achieves better results in comparison to other solutions in the field, allowing for development of more accurate systems. Moreover, our proposed method demonstrates more robustness against occlusions, which can be beneficial for in-the-wild scenarios. Lastly, our model can learn trees that are optimized, yet also generalize well across different datasets and perform considerably better than human-designed face trees.

Index Terms—Face graphs, facial expression recognition (FER), tree topology learning.

I. INTRODUCTION

FACIAL expressions play a key role in conveying a wide range of human emotions [1], [2]. Thus, considerable attention has been given to automatic facial expression recognition (FER) using machine learning and deep learning methods, where the goal is to identify different expressions from images or videos of the face [3], [4], [5]. Recent progress in deep neural networks and computational resources, along with the availability of vast amounts of data, have resulted in great progress in the area of FER [6]. Applications of FER include but are not limited to emotion classification [4], pain intensity detection [7], age estimation [8], estimation of the user’s experience during consumption of multimedia content [9], and detection and prediction of the severity of depression [10], among many others. A detailed recent review of FER is presented in [11].

While many early deep FER approaches, such as [12] and [13], focused on holistic representations of the face, many newer methods have shifted focus to part-based approaches to describe the face as distinct [14] or disentangled components [15]. The reason for this focus toward part-based methods is that certain regions of the face, such as mouth or periocular areas, have been shown to play different roles in reflecting different expressions [16]. Part-based FER techniques typically analyze face regions separately to extract local features, which are then merged to produce global representations [14]. As a result, given that graph-based methods generally lend themselves well to part-based problems, they have recently gained popularity for FER [17], [18], [19], [20], [21], [22]. In such approaches, each node in a graph can be associated with a facial region, where an edge can reflect the connections or relationships between the regions, for instance their joint movement or distance.

Despite the fact that a number of graph-based FER methods have shown promising results [18], [23], the underlying graph topology used to learn the structure of the face is not generally agreed upon. In other words, there is no single way of knowing how various face regions should be connected such that the model’s performance would be maximized. Graph-based FER methods assume the topology of underlying graphs as being either fully connected [18], [19], [21] or partially connected [17], [20], [22]. In the fully connected scenario, one must add $\binom{n}{2}$ edges to the underlying graph, which makes the training process cumbersome. In addition, some of the edges are unnecessary within the graph, which may cause overfitting. In the partially connected scenario, in light of the fact that the topology is predefined and engineered as opposed to being learned, it may not include the optimal or near-to-optimal edges between face regions. In any case, the underlying graph topology can significantly impact the performance of any subsequent learner that
adopts the information obtained from different facial regions toward part-based FER. As an example, Fig. 1 depicts a face image together with four random trees, which have been used to connect different regions of the face. Traversal of the trees yields four distinct sequence-based embeddings, using which we can conduct a seven-class FER task using four independent sequential learners [long short-term memory (LSTM) has been used in this case]. The associated recognition rates (RR) are provided beneath each image in the Fig. 1, where we observe a 3.7% variance, demonstrating the necessity of selecting the optimal tree topology.

In this article, we present an end-to-end pipeline, FaceTopoNet, that learns an effective tree of the face for robust FER in a deep neural architecture. First, we assume a weighted fully connected graph obtained from facial landmarks, which is then optimized by an evolutionary algorithm [24]. This graph is then used to form a minimum-cost spanning tree, the traversal of which forms a sequence of facial nodes. This sequence is then used for two subsequent streams, which we name texture and structure streams. In the structure stream, the obtained sequence is used to feed a sequential learner followed by soft attention to focus on the salient cells of the learner. In parallel, in the texture stream, we apply local patches around each of the extracted nodes and encode each region using a simple ConvNet. These encoded representations are used to feed a sequential learner similar to the structure stream, followed by soft attention. The final embeddings of the two streams are then fused for final expression recognition. To the best of our knowledge, our work is the first to optimize the relevant order of facial regions for sequential learning in the context of FER. We perform rigorous experiments on four in-the-wild large-scale FER datasets, AffectNet [25], FER2013 [26], real-world affective face database (RAF-DB) [27], and ExpW [28], and one lab-controlled dataset, i.e., Cohn–Kanade (CK+), to illustrate the efficacy of our devised method. Our method outperforms the state-of-the-art results on three of the datasets and achieves competitive results on the other two. We further perform additional experiments on ablations and even evaluate the robustness of our method to occlusions in different key regions of the face.

Our following contributions are threefold.
1) We propose a novel end-to-end deep architecture to identify and use the effective tree topologies for FER. These face trees are found by using the idea of weighted complete graphs and minimum-cost spanning trees. They are then used to determine the input order of the embeddings to be fed to two sequential learners located in two separate streams responsible for learning the structure and texture of the face.
2) Our experiments demonstrate the strong performance of our method by setting new state-of-the-art results on AffectNet, FER2013, and ExpW, and performs competitively on RAF-DB and CK+.
3) Our solution is robust to different occlusions added to input images. Further, we find that our method automatically adapts and finds different face trees for each dataset, and uses them to obtain improved performance. It is worth mentioning that in traditional landmark-based methods, the input order of the sequential learner is not learned as opposed to FaceTopoNet. In fact, FaceTopoNet provides a way to simultaneously learn the input orderings of a sequential learner and its free parameters.

This article is an extension of our previous work [29], which has been published in FG 2021 as a short article. In comparison, this article adds the followings.

i) We use three additional publicly available datasets, namely RAF-DB, ExpW, and CK+, to evaluate the proposed method.

ii) We analyze and discuss the effects of using different sequential learners and encoders with different pre trainings.

iii) We perform robustness experiments toward occlusions against other state-of-the-art methods by manually removing different key regions of the face.

iv) We study the effect of different patch sizes that are located around the facial landmarks on the overall recognition rates.

v) The effects of the number of extracted facial landmarks is studied.

vi) We perform a cross-dataset study utilizing face trees.

vii) Finally, we perform a direct comparison between the learned face trees and prior human-defined trees used for FER.

The rest of this article is organized as follows. Section II provides a summary of the related work in the field. The proposed method, including the detailed description of tree topology learning, is given in Section III. Section IV presents the results of the extensive experiments along with the detailed ablation analysis. Finally, Section V concludes this article.

II. RELATED WORK

FER systems can be divided into static and dynamic categories with respect to the type of input data [11]. In static-based methods, the facial features are encoded based on the spatial information available in a single image, whereas dynamic-based methods, in addition, exploit the temporal relationships between adjacent frames available in the input video. Our focus in this
article lies in the first category. Hence, we review relevant article for image-based FER in this section. In addition, given the use of graphs in our method, we provide a review of the available techniques that incorporate graph theory in FER systems.

A. Facial Expression Recognition With Deep Learning

In one of the early neural network-based methods for FER, a shallow convolutional neural network (CNN) was used for feature learning, followed by a linear support vector machine (SVM) for classification [30]. Another method, HoloNet, was introduced in [31], combining residual structures with concatenated rectified linear activation scheme [32]. In addition, an inception-residual structure was adopted in the final layers to enable multiscale feature extraction. In [33], a nine-layer CNN with a multiclass SVM classifier was used for FER.

In [15], the notion of disentanglement was used to make FER systems more robust against pose, illumination, and other variations. In this context, a radial metric learning strategy was proposed for disentangling identity-specific factors from facial input images. In [34], an attention method was proposed to selectively focus on facial muscle moving regions. To guide the attention learning process, prior domain knowledge was incorporated by considering the differences between neutral and expressive faces during training. In [35], another attention-based method was suggested. The method used a region-attention network to identify the importance of different facial regions for FER, thus assigning higher weights to facial regions with greater importance. This resulted in further robustness to occlusions and pose variations.

In [36], the so-called advanced softmax loss was proposed to reduce the negative effects caused by data imbalance, particularly when the classifier is biased toward the majority class. Kollias et al. [37] synthesized facial affects from a neutral image for data augmentation in FER systems. Their model accepts a neutral image and a basic facial expression as inputs and applies a 3-D Morphable model [38] to the neutral image, deforming it for adding the expression. A model called MA-Net was proposed in [39] for conducting in-the-wild FER. The proposed solution includes three components. In the first component, named feature preextractor, mid-level features of face images are extracted. The second component tries to fuse different features which are related to different receptive fields, and finally, in the third component, the network is guided to be more focused on the more important local features. A model named TransFER was proposed in [40] for FER. The approach is capable of learning complex local representations and is comprised of three subnetworks:

1) multiattention dropping, forcing the model to come up with more diverse local patches;
2) vision transformer to model the relations existing among different local patches; and
3) multihed self-attention dropping that drops a self-attention module randomly to make the model learn different local patches in a diverse way.

Since facial attributes, which are associated with identity, such as race or age, can hinder the efficacy of FER systems, an identity-free FER system is proposed in [41]. Cai et al. [41] had made use of a generative adversarial network to transform a given face image with a specific identity to its “average” identity version while preserving the same expression. A new way to view the facial expressions was proposed in [42]. Specifically, Ruan et al. [42] utilized a feature decomposition network to extract shared information among expressions and a feature reconstruction network to extract expression-specific information. In [43], for data augmentation purpose, Choi et al. [44] used StarGAN to synthesize identity-specific face images with basic emotions. Then, a deep network was used to extract latent embeddings from both synthesized and real images. These embeddings were used to train a Mahalanobis metric network for performing FER. Bisogni et al. [45] designed an FER system to assist healthcare systems. In this regard, they propose different convolutional neural network architectures to handle multiresolution facial images. Hossain et al. [46] proposed a three-component FER system. First component includes the detection of facial regions, the second component uses CNNs to extract the embeddings, and the third component adopts different techniques, such as transfer learning, to extract more distinct embeddings. A novel data augmentation technique is introduced in [47] for performing FER. This technique includes applying edge enhancement filters, such as bilateral filtering on the images. Although deep learning methods provide the state-of-the-art results in FER, some articles, such as [48], use a conventional machine learning approach (feature extraction, feature classification, and classification) to perform FER and achieved reasonable results.

In [49], a novel residual-based CNN architecture, bounded residual gradient network (BReG-Net), was proposed for FER. This CNN architecture includes 39 layers of residual blocks, where the standard skip-connections in the ResNet blocks were replaced by a differentiable function with a bounded gradient, thus avoiding the gradient vanishing and exploding problems. As an extension to [49], Hasan et al. [50] proposed BReG-NeXt, which includes more trainable parameters and offers more flexibility. A trainable parameter has also been added to the residual units. While this complex mapping adds few more parameters to each residual block, it extracts more representative features in comparison to its counterparts like ResNet-50.

B. Applications of Graphs in FER

Neural networks on graph topologies have recently gained momentum for visual analysis tasks [51], [52], including FER [17], [18], [19], [20]. In [19], Gabor features were first extracted from texture information around 49 facial landmarks and were then combined with the Euclidean distances calculated between landmark pairs to feed a bidirectional gated recurrent unit. In [18], local regions around five landmarks, including center of left and right eyes, nose, and left and right mouth corners, were extracted to be then learned using a GCN. Similarly in [21], 28 landmarks corresponding to eyebrows and mouth were selected to create three separate graphs for left/right eyebrows and mouth. These graphs were then processed using a graph temporal convolutional network. In [20], 34 landmarks,
corresponding to three facial components, including left eye and eyebrow, right eye and eyebrow, and mouth, were first detected, and the connections between these components were established with respect to their psychological semantic relationships. Histogram of orientation features and the XY-coordinates of the nodes were then combined to feed a spatial-temporal semantic graph network. A novel multitask learning was proposed in [53], which adopted GCNs to model the dependencies among the discrete and continuous models of emotions. To be more specific, a shared representation was learned for both discrete and continuous models of emotion, and the final classifier of the discrete model and the regressor of the continuous model were jointly learned through a GCN. Liu et al. [54] adopted a GCN into a well-known CNN-RNN-based model for performing FER. The role of the GCN was to help the model learn more salient features. In addition, an LSTM layer was utilized to learn long dependencies between the GCN-extracted features.

Unlike all these techniques, which employ fully connected graphs for FER, a few solutions, such as [20] and [17], assume a predefined subset of connections between the nodes followed by a variation of graph convolutional networks. Nonetheless, to the best of our knowledge, all the graph-based FER solutions utilize a predefined graph topology (whether fully connected or otherwise) to be then used for every scenario. The problem with these predefined graphs is that they do not necessarily include the optimal set of connections, and once designed, stay unchanged for different datasets. We aim to tackle this problem by proposing a novel pipeline in which a unique effective tree topology can be learned given the problem space.

### III. Proposed Method

#### A. Problem Setup and Solution Overview

Graphs can be used to model the topology and the relationship between different facial landmarks. Each vertex in these graphs can represent a facial landmark, whereas the existence of an edge between two vertices (facial landmarks) would consequently demonstrate that there is a relationship between the two. In [20], it was shown that these graphs provide a better representation when compared with holistic approaches. The constructed graphs can then be processed using deep networks, such as 3D-CNNs [55], GCNs [54], or RNNs [19] (which we refer to as sequential learners) [56]. As shown in Fig. 1, when using a sequential learner, the accuracy of a solution can be affected by the topology of the graph being processed. Nonetheless, to the best of our knowledge, no technique for learning the topology of graphs for FER has been previously proposed in the literature.

To address the aforementioned problem, we propose a new solution in this article to learn the topology of the graph in the form a face tree, and then, use this tree to further learn effective representations from the face.

In our designed solution, for learning the salient information of the face, two important streams are considered: structure and texture. The backbone of these two streams are two sequential learners. Accordingly, our proposed end-to-end method uses the spatial information of the extracted facial landmarks in the structure stream and the texture information of the patches around these facial landmarks in the texture stream. The final embeddings to be fed into the sequential learners are determined by the topology of the learned face tree. In other words, we traverse the face tree to generate a sequence, and this sequence determines the ordering of the embeddings to be fed into the sequential learners. Our proposed pipeline’s architecture is shown in Fig. 2. What follows is the detailed description of our tree topology learning procedure along with the two-stream solution.

#### B. Tree Topology Learning

First, we employ a landmark extractor $\Phi$ to detect facial landmarks from an input image $I$ such that

$$\lambda = \Phi(I)$$

where $\lambda = [\lambda_1 \cdots \lambda_i \cdots \lambda_n]$, $\lambda_i$ represent the coordinates of the $i$th facial landmark in Euclidean space, and $n$ is the number of extracted facial landmarks. The optimal value of $n$ will be studied in Section IV. In this work, we use a deep regression architecture [57] as $\Phi()$. Next, we suppose that the extracted facial landmarks are vertices of a graph. We, thus, have a graph with $n$ vertices where each vertex represents a facial landmark. We then add all the possible edges, obtaining a complete graph $K_n$.

Next, an optimization algorithm (described in the following section) assigns weights to the edges of $K_n$, making it a weighted complete graph. Following, we extract a minimum-cost spanning tree from $K_n$. This minimum-cost spanning tree can be expressed as follows:

$$\text{MST} = \{(V_{\text{MST}}, E_{\text{MST}})\}$$

where MST is the minimum-cost spanning tree, and $V_{\text{MST}}$ and $E_{\text{MST}}$ represent the vertices and edges of MST, respectively. Here, $V_{\text{MST}} = V_{K_n}$ and $E_{\text{MST}} \subseteq E_{K_n}$, where $V_{K_n}$ and $E_{K_n}$ denote the vertices and edges of $K_n$. Moreover, in this setup, $\sum_{e_{ij} \in E_{\text{MST}}} c_{ij}$ is minimum. For finding MST, we use Algorithm 1 based on [58].

By utilizing the MST, we aim to generate a sequence that determines the ordering of the embeddings for both structure and texture streams. The reason behind converting the complete graph $K_n$ to a tree is that trees are acyclic in nature and are, thus, better suited for sequence formation. For the purpose of sequence formation, we traverse MST using the preorder depth first algorithm described in [59]. In the context of our problem, we start the MST traversal from the most centered vertex, i.e., nose tip. Ultimately, a sequence of vertices is built. We depict the tree traversal in Fig. 3. In Fig. 3, for a clear visualization, we explain the process for a tree with only $n = 9$ nodes instead of the entire set of landmarks.

Suppose that the sequence generated after the tree traversal is denoted by $v = [v_1 \cdots v_n]$, where $v_1$ is the first vertex (landmark) of the tree, and $\gamma$ is one of the existing vertices between 1 and $n$. Accordingly, the number of elements in vector $v$ is provably $2n - 1$. By utilizing $v$, we build matrix $U = [u_1 \cdots u_n]$, where $u_1$ is a vector whose elements are zeros except for the first element. Similarly, $u_\gamma$ is a vector whose elements are zeros except for the $\gamma^{th}$ element. Matrix $U$ is then used to generate
Fig. 2. Pipeline of the proposed FaceTopoNet.

Fig. 3. Details of the tree traversal and the following embedding formation. The built minimum-cost spanning tree is traversed by using preorder depth first order, which produces the sequence 1-2-3-2-1-4-5-4-1-6-7-8-7-9-7-6-1. The resulting embedding of the sequence is fed to a sequential learner.

the input embeddings of the sequential learners, which will be described in Section III-C.

To learn an effective face tree structure, the weights of $K_n$ need to be updated. In other words, changing the weights of $K_n$ results in different face trees, thus different sequences. Given the greedy nature of Algorithm 1, it is difficult to express the output of our pipeline as a differentiable function of the weights of $K_n$. In dealing with such problems, various optimization algorithms can be used to learn effective trees. In this work, we utilize the cooperatively coevolving particle swarms for large scale optimization (CCPSO2) algorithm [24], which is an evolutionary method, to optimize the weights of $K_n$. CCPSO2 belongs to a family of metaheuristic cooperative algorithms, which makes use of swarm intelligence as well as their cooperation to estimate the solution of optimization problems of up to 2000 variables effectively [24]. Due to the fact that our search space in this work has $\binom{n}{2}$ dimensions, for a reasonably constrained value, such as $n = 50$, CCPSO2 can perform effectively for this problem. We use this optimization technique to learn the weights of $K_n$ while minimizing the loss function of the model $L_{total}$, which we will define later in Section III-F. It is worth mentioning that while CCPSO2 does not promise an optimal solution (it can get stuck in local optima like many other optimizers), what matters is the generalization ability of the converged face tree on unseen data, which will be extensively tested on five different FER datasets in Section IV.

C. Structure Stream

Here, we aim to learn a sequential learner by using the spatial information of the extracted facial landmarks. Specifically, the input embedding of this sequential learner is determined by the matrix $U$. As shown in Fig. 2, the inputs to the structure stream

\[
\text{Algorithm 1: Minimum Cost Spanning Tree.} \\
\textbf{Result:} \text{MST} \\
\textbf{Initialization:} \text{visited} = \{1\}, \text{unvisited} = \{0, 1, \ldots, n - 1, n + 1, \ldots, \binom{n}{2}\}, \text{MST} = \emptyset \\
\text{while } \text{unvisited} \neq \emptyset \text{ do} \\
\quad \text{find edge } e = (P, Q) \ni P \in \text{visited} \text{ and } Q \in \text{unvisited} \text{ and } e \text{ has the smallest weight} \\
\quad \text{MST} = \text{MST} \cup \{e\} \\
\quad \text{visited} = \text{visited} \cup \{Q\} \\
\quad \text{unvisited} = \text{unvisited} - \{Q\} \\
\text{end} \]
are the matrix $\lambda$ that contains the spatial information of the facial landmarks, together with the $U$ matrix. These two matrices are used to form the input embedding for the sequential learner, for example, an LSTM with peephole connections [60]. The input embedding $\zeta$ is generated by

$$\zeta = \lambda \times U. \quad (3)$$

An LSTM network consists of a number of cells, the output of which continuously evolves with regards to its memory content. Moreover, each cell has a common cell state that is necessary for keeping track of long-term dependencies within the LSTM network. Two gates control the follow of information in the common cell state: input and forget gates, which make the LSTM cell decide when to forget or update the cell’s state. The output of an LSTM cell is its hidden state, which is also controlled by the common cell state. We input $\zeta$ to the LSTM network and the aggregation of hidden states makes the final output of the LSTM network.

Next, we incorporate an attention mechanism [61] over the output of our sequential learner to focus on the salient parts of the embedding. To do so, each hidden state vector of the sequential learner cell ($h_i$) is multiplied by a learnable weight $\delta_i$. $\delta_i$ controls the amount of attention each $h_i$ receives and can be expressed as follows:

$$\delta_i = \frac{e^{o_i}}{\sum_{k=1}^{n} e^{o_k}}. \quad (4)$$

where $n$ illustrates the number of units in the sequential learner and $o_i$ can be calculated using the following:

$$o_i = \tanh(W_hh_i + b_h). \quad (5)$$

Here, $W_h$ and $b_h$ denote the set of trainable weights and biases, respectively. We calculate the final attentive output of the structure stream $H_{\text{structure}}$ using

$$H_{\text{structure}} = \sum_i \delta_i h_i. \quad (6)$$

D. Texture Stream

In this stream, the texture information of the input images is used to learn a sequential learner. Like the structure stream, the matrix $U$ determines the input embedding of the sequential learner. Inputs to this stream are the input image $I$, the matrix $\lambda$, and the matrix $U$. To use the texture information of the input image $I$, we first create $a \times a$ patches around the center of each facial landmark (the optimal value of parameter $a$ is determined empirically in Section IV-E). The patches are then cropped and fed to a pretrained encoder, such as ResNet-50 pretrained on the VGG-Face2 dataset [62].

Suppose that the encoder generates an embedding for the first patch. We call this embedding $\psi_1$. We stack the outputs of the encoder for all the patches in a single matrix denoted by $\Psi = [\psi_1 \psi_2 \cdots \psi_n]$. By multiplying $\Psi$ and $U$, we compute the input embedding of the second sequential learner (recall, the first sequential learner was used in the structure stream). Finally, we take the output of this sequential learner from its cells’ hidden states and employ an attention mechanism on them similar to the structure stream.

E. Fusion

To combine the outputs of the two attention mechanisms at the end of each of the streams, we employ the fusion strategy proposed in [63] and used in other prior work, such as [64]. This fusion strategy was initially created with the goal of preserving the original properties of its input embeddings, and has obtained state-of-the-art results in many areas [63]. We add two dense layers for the output of structure and texture streams. These dense layers act like an encoder to generate stream-specific embeddings. We then employ soft attention to learn the weight $\eta$ using the following:

$$\eta = \text{softmax}(\tanh(W_f[T^*, S^*] + b_f)) \quad (7)$$

where $W_f$ and $b_f$, respectively, denote the trainable weights and biases, and $T^*$ and $S^*$, respectively, denote the stream-specific embeddings from texture and structure streams. Finally, for learning the optimal combination of the two stream-specific embeddings, we utilize a dense layer as follows:

$$y = \tanh(W_y[(1 + \eta_T)H_{\text{texture}} + (1 + \eta_S)H_{\text{structure}}] + b_y) \quad (8)$$

where $y$ denotes the final embedding, and $W_y$ and $b_y$ represent learnable weights and biases, respectively. Finally, $y$ is input to a softmax layer for determining the expression class probability. Fig. 4 summarizes the fusion strategy used in this work.

F. Loss Function and Training

Considering each training epoch, a set of weights related to $K_n$ is generated by the CCPSO2 algorithm. These weights are used to form the MST, the traversal of which yields the matrix $U$. Next, we freeze the weights of $K_n$ and matrix $U$ and fully train the following texture and structure streams using ADAM optimizer [65]. When the training of the two streams is complete, we feed the final loss value to the CCPSO2 optimization algorithm as its loss function. Given the new loss value, the optimization algorithm creates a set of new weights for the complete graph. This process repeats until convergence.
We define the total loss ($L_{\text{total}}$) for both CCPSO2 and ADAM optimizers in our end-to-end method as follows:

$$L_{\text{total}} = \frac{1}{3} (L_1 + L_2 + L_3)$$

where $L_1$ denotes the fusion loss, and $L_2$ and $L_3$ represent the losses for the structure and texture streams, respectively. Generally, FER datasets are highly skewed toward a specific expression, usually happy. To demonstrate this, we depict the probability mass function (pmf) of the datasets used in this study (AffectNet, FER2013, RAF-DB, ExpW, and CK+) in Fig. 5. Fig. 5 demonstrates that the distributions are highly skewed toward happy class. This issue makes it hard for FER algorithms to perform well on minority classes [66]. To address this issue, we adopt focal loss [67] for all three loss terms in (9). Therefore, the loss terms are defined as follows:

$$L_i = -\alpha_i (1 - p_b)^{\gamma} \log(p_b) \quad \text{for } i = 1, 2, 3$$

where $\alpha_i$ is a balancing factor, and $p_b$ is defined as follows:

$$p_b = \begin{cases} p & y = 1 \\ 1 - p & \text{otherwise}, \end{cases}$$

where $y$ is the ground truth and $p$ is the probability calculated by the model. In (10), $\gamma$ denotes the focusing parameter. Empirically, we set $\gamma$ and $\alpha_i$ to 2 and 0.25, respectively. Our use of focal loss allows the model to be less focused on the majority class by downweighting the samples that are well classified, as shown in the equation.

### G. Implementation Details

We implement the proposed architecture using TensorFlow [68] and train it using two Nvidia RTX 2080 Ti GPUs. For ADAM optimizer, we set learning rate, first-momentum decay, and second-momentum-decay as 0.001, 0.9, and 0.99, respectively. For all the experiments, the face images are cropped and resized to $60 \times 60 \times 3$ pixels. In addition, for data augmentation purpose, each image is horizontally flipped with the probability of 0.25. Similar to [37], [50], and [69], the reported results are achieved on the validation set of the respective datasets. However, for CK+ dataset, we use leave-one-subject-out cross validation [70].

Considering CCPSO2 algorithm, the number of iterations is set to 40, as convergence occurs at or before this iteration, after which no changes are observed in the generated trees. In addition, for random grouping of the variables we use the set $\{1, 5, 7\}$, and the population size for each swarm is set to 10. As CCPSO2 algorithm is a stochastic method, we run it 30 times. However, the converged tree did not change, indicating the consistency of the approach.

### IV. Experiments

We use four well-known public in-the-wild FER datasets to evaluate our method. In this section, we first introduce these datasets, followed by the details of our experiments, and obtained results.

#### A. Datasets

**AffectNet [25]:** Introduced in 2017, AffectNet includes more than 1 million face images. All the images are collected from the Internet using three different search engines. Within the dataset, about 450 000 images, have been annotated manually while the remaining have been automatically annotated, i.e., using deep neural networks. In addition, this dataset provides the ground truth for both the categorical model of affect (happy, sad, surprise, disgust, fear, anger, contempt, and neutral) and the dimensional model of affect, i.e., arousal and valence values for each image. The validation set of AffectNet includes 500 images for each expression class.

**FER2013 [26]:** Introduced in 2013, FER2013 comprises 28 709 and 7178 images, respectively, for training and validation. All the images have the resolution of 48 $\times$ 48 pixels. Moreover, this dataset labels each image in one of the following six expression classes: happy, sad, surprise, disgust, fear, anger, and neutral.

**RAF-DB [27]:** The real-world affective face database (RAF-DB) provides 29 672 face images with their associated expression labels. Crowdsourcing has been used to assign each face image to one of the seven basic and 11 compound expression classes. In addition, the dataset has been split into train and validation sets to provide a standard benchmark to the community.

**ExpW [28]:** This dataset is collected from the Internet by using the Google search engine and offers 91 793 face images. All the images have been labeled manually as one of the seven basic expression classes.

**CK+ [70]:** Comprising 327 videos, the extended CK+ dataset is one of the well-known precursors in the FER datasets. Each video sequence demonstrates a change from neutral to the peak expression. Moreover, CK+ provides the expression labels as being one of the six basic expressions plus contempt expression. For evaluation purpose, we select three peak expression frames within each expression sequence to build train and validation sets.
TABLE I

| Dataset    | Encoder         | Pretrained on  | Sequential Learner |
|------------|-----------------|----------------|-------------------|
|            |                 | ResNet-50      | NAME_1 | Bi-LSTM | GRU | Bi-GRU |
| AffectNet  | ResNet-50       | ImageNet       | NAME_2   | 70.62   | 69.81 | 69.51 |
|            | ResNet-50       | VGG-Face2      | LSTM      | 69.37   | 68.90 | 69.23 |
|            | VGG-16          | ImageNet       | LSTM      | 68.21   | 68.28 | 68.72 |
|            | BREG-NeXt-50    | VGG-Face2      | LSTM      | 69.43   | 69.15 | 69.57 |
| FER2013    | ResNet-50       | VGG-Face2      | LSTM      | 72.66   | 72.51 | 72.57 |
|            | ResNet-50       | ImageNet       | LSTM      | 71.26   | 71.26 | 71.31 |
|            | VGG-16          | ImageNet       | LSTM      | 71.21   | 71.19 | 71.21 |
|            | BREG-NeXt-50    | VGG-Face2      | LSTM      | 72.48   | 72.37 | 72.46 |
| ExpW       | ResNet-50       | VGG-Face2      | LSTM      | 71.91   | 71.83 | 71.83 |
|            | ResNet-50       | ImageNet       | LSTM      | 70.74   | 70.66 | 70.75 |
|            | VGG-16          | ImageNet       | LSTM      | 70.49   | 70.49 | 70.54 |
| RAF-DB     | ResNet-50       | VGG-Face2      | LSTM      | 87.80   | 87.78 | 87.81 |
|            | ResNet-50       | ImageNet       | LSTM      | 86.93   | 86.95 | 87.02 |
|            | VGG-16          | ImageNet       | LSTM      | 86.82   | 86.77 | 86.94 |
| CK+        | ResNet-50       | VGG-Face2      | LSTM      | 99.19   | 98.9  | 99.19 |
|            | ResNet-50       | ImageNet       | LSTM      | 98.4    | 97.8  | 97.3  |
|            | VGG-16          | ImageNet       | LSTM      | 96.3    | 95.9  | 96.1  |

Bold values indicate the highest recognition rates for each dataset.

TABLE II

RESULTS OF FACETOPONET ON AFFECTNET DATASET. “NE.,” “HA.,” “SA.,” “SU.,” “FE.,” “DI.,” “AN.,” AND “CO.” DENOTE NEUTRAL, HAPPY, SAD, SURPRISE, FEAR, DISGUST, ANGER, AND CONTEMPT, RESPECTIVELY

| Authors   | Method            | RR | Precision | Recall |
|-----------|-------------------|----|-----------|--------|
|           |                   |    | Ne. | Ha. | Sa. | Su. | Fe. | Di. | An. | Co. | Ne. | Ha. | Sa. | Su. | Fe. | Di. | An. | Co. |
| Hewitt et al. [71] | CNN              | 58 | 45 | 76 | 59 | 55 | 60 | 55 | 57 | 59 | 49 | 72 | 56 | 60 | 64 | 63 | 43 | 56 |
| Hua et al. [72] | Ensemble         | 62.11 | 58 | 72 | 64 | 62 | 57 | 59 | 62 | . | 59 | 70 | 67 | 66 | 61 | 59 | 57 | 57 |
| Hasani et al. [50] | BREG-NeXt-50    | 66.74 | 58 | 72 | 64 | 62 | 57 | 59 | 62 | . | 59 | 70 | 67 | 66 | 61 | 59 | 57 | 57 |
| Hasani et al. [50] | BREG-NeXt-50    | 68.50 | 72 | 78 | 62 | 66 | 67 | 62 | 63 | 67 | 53 | 89 | 66 | 74 | 63 | 77 | 61 | 58 |
| Chen et al. [73] | Residual multitask learning | 61.98 | 72 | 78 | 62 | 66 | 67 | 62 | 63 | 67 | 53 | 89 | 66 | 74 | 63 | 77 | 61 | 58 |
| Zhang et al. [74] | Pseudoasiamese structure | 63.70 | 72 | 78 | 62 | 66 | 67 | 62 | 63 | 67 | 53 | 89 | 66 | 74 | 63 | 77 | 61 | 58 |
| Farzaneh et al. [75] | Deep attentive center loss | 65.20 | 72 | 78 | 62 | 66 | 67 | 62 | 63 | 67 | 53 | 89 | 66 | 74 | 63 | 77 | 61 | 58 |
| Ours       | FaceTopoNet      | 70.02 | 72 | 63 | 75 | 65 | 68 | 72 | 74 | 77 | 38 | 98 | 66 | 78 | 63 | 88 | 68 | 59 |

Bold values indicate the highest recognition rates for each column.

B. Encoders and Sequential Learners

First, we aim to identify the best encoder and sequential learner to be used in FaceTopoNet. Accordingly, we consider several candidates for each submodule. For the sequential learner, we test LSTM, bidirectional LSTM (Bi-LSTM) [78], gated recurrent unit (GRU) [56], and bidirectional GRU (Bi-GRU) [56], whereas for the encoder, we explore ResNet-50 pretrained on VGG-Face2, ResNet-50 pretrained on ImageNet [79], VGG-16 pretrained on ImageNet, and BREG-NeXt-50 pretrained on VGG-Face2 for encoder. These submodules have been selected for experimentation given their strong performance in prior work [50], [80]. We report the FER results for each dataset in Table I. Given the results, the best combination for four of the datasets is achieved with a ResNet-50 pretrained on VGG-Face2 and LSTM. The only exception is on the RAF-DB dataset in which GRU exhibits slightly (marginal) better performance than the LSTM. In addition, regarding CK+, GRU-ResNet-50 combination delivers the same accuracy as the LSTM-ResNet-50 combination. Hence, for the sake of consistency in our experiments, we use a ResNet-50 and LSTM in FaceTopoNet. We observe from the table that pretraining on VGG-Face2 obtains better results compared with ImageNet, which is predictable given that VGG-Face2 only includes face images with different poses and illuminations as opposed to ImageNet, which contains various image classes of different objects. This can help the encoder to extract embeddings that are discriminative and important in the context of human faces. Moreover, we observe that the added depth and residual connections of ResNet-50 results in better performance compared with VGG-16.

C. Performance

We report the RR values for FaceTopoNet along with other state-of-the-art benchmarks on all five datasets in Tables II–VI. Considering AffectNet (see Table II), FaceTopoNet outperforms the state-of-the-art FER methods BREG-NeXt-50 and BREGNeXt-32 by 1.52% and 3.28%, respectively. With regards to FER2013 dataset (see Table III), FaceTopoNet shows an improvement of 1.13% over the state of the art, BREG-NeXt-50. The results for ExpW are tabulated in Table V. In this table, we can observe that FaceTopoNet has the highest recognition rate (71.91%) compared with other methods. This is followed by [81], which also uses part-based methods, dividing the face into different sections, to perform FER. The results for RAF-DB are tabulated in Table IV. As it is evident from this table, our method achieves the second-best recognition rate, being marginally below [82] by only 0.27%. A reason for this could be the use of a new loss term by [82], which is equipped to handle class imbalances, which is a common issue...
in FER datasets, by monitoring the last layers of a CNN. Finally, regarding CK+ dataset, FaceTopoNet achieves the second best recognition rate. The current state of the art [42] offers the recognition rate of 99.54%, whereas FaceTopoNet achieves slightly lower recognition rate, 99.19%.

As some of the state-of-the-art methods, in addition, reported precision and recall values for both AffectNet and FER2013 datasets, we also consider these measures for each emotion class in Tables II and III. As it is evident from Table II, for most of the expression classes, FaceTopoNet outperforms other state-of-the-art methods in terms of precision and recall. More specifically, FaceTopoNet obtains the best precision and recall in disgust and contempt expressions, which are considered as more difficult classes due to their small numbers of training samples in the AffectNet dataset. Given FER2013 dataset, FaceTopoNet delivers the highest precision in most of the expression classes, i.e., neutral, sad, fear, and angry, while achieving the state-of-the-art recall in three expression classes.

For a deeper comparison between our proposed model and one of the best performing benchmarks in both AffectNet and FER2013 datasets, BreG-NeXt-50, F1 scores corresponding to the expression classes are shown in Fig. 6 using one-vs-all method [83]. Considering AffectNet, our proposed method obtains higher F1 scores in six out of eight expressions when compared with BreG-NeXt-50, whereas in FER2013, we outperform BreG-NeXt-50 in four of the seven classes.

We depict the obtained confusion matrices between the ground truth and the predicted expressions using our proposed method in Fig. 7. For AffectNet, three cases, including contempt–happy, sad–surprise, and neutral–happy (the first and second elements are, respectively, the ground truth and the predicted expression) have the largest confusions. Confusion between contempt and happy is prevalent and expected in FER systems given the high semantic similarity of the two classes, and as evidenced by high confusion rates obtained from the perspective of human observers [50]. Regarding FER2013, disgust–happy has the largest confusion. This is due to the fact that happy and disgust are, respectively, the majority and minority expression classes within the dataset causing a bias within the model (see the pmf for FER2013 in Fig. 5). Despite the fact that we adopt focal loss to address the issue of the imbalanced training set, FaceTopoNet still identifies some of the disgust cases (minority class) as happy (majority class). Apart from disgust, FaceTopoNet performs well on the FER2013 dataset overall.

For a deeper comparison between our proposed model and one of the best performing benchmarks in both AffectNet and FER2013 datasets, BreG-NeXt-50, F1 scores corresponding to the expression classes are shown in Fig. 6 using one-vs-all method [83]. Considering AffectNet, our proposed method obtains higher F1 scores in six out of eight expressions when compared with BreG-NeXt-50, whereas in FER2013, we outperform BreG-NeXt-50 in four of the seven classes.

We depict the obtained confusion matrices between the ground truth and the predicted expressions using our proposed method in Fig. 7. For AffectNet, three cases, including contempt–happy, sad–surprise, and neutral–happy (the first and second elements are, respectively, the ground truth and the predicted expression) have the largest confusions. Confusion between contempt and happy is prevalent and expected in FER systems given the high semantic similarity of the two classes, and as evidenced by high confusion rates obtained from the perspective of human observers [50]. Regarding FER2013, disgust–happy has the largest confusion. This is due to the fact that happy and disgust are, respectively, the majority and minority expression classes within the dataset causing a bias within the model (see the pmf for FER2013 in Fig. 5). Despite the fact that we adopt focal loss to address the issue of the imbalanced training set, FaceTopoNet still identifies some of the disgust cases (minority class) as happy (majority class). Apart from disgust, FaceTopoNet performs well on the FER2013 dataset overall. On the RAF-DB dataset, the largest confusions occur in the case of disgust–sad as well as disgust–neutral. The same issue can be seen in some of the prior works on RAF-DB [92]. As for the ExpW dataset, fear is confused with neutral, happy, and sad expressions. This can be justified as fear only constitutes 1.1% of the ExpW dataset. Finally, considering CK+ dataset,
happy and disgust expressions have the highest accuracy. In addition, the largest confusions occur in contempt–fear and sad–disgust cases. Among the expressions within the CK+, disgust expression shows the lowest error, whereas fear shows the highest error.

The key concept behind the good performance of our method is the “ordering” of the embeddings. In Fig. 1, we have demonstrated that the ordering of the embeddings can affect the performance of sequential learners. Thus, our model optimizes the ordering of the embeddings, such that it becomes easy for a sequential learner to model the dependencies that exist among these embeddings. In this way, the sequential learner delivers superior performance.

**D. Optimal Number of Extracted Facial Landmarks**

To choose the optimal value of the parameter \( n \) (number of the extracted facial landmarks), we increase it by 10 each time and report the performance. The resulting curve is shown in Fig. 8 for both AffectNet and FER2013 datasets. Considering AffectNet dataset, there is a direct relationship between \( n \) and the performance, up until 50 facial landmarks, from where the performance starts to drop. This behavior can be justified as the \( n \) increases, the overlap between the patches around these landmarks, and thus, the redundant information increases. Almost the same behavior can be seen for the FER2013 dataset. To be consistent in all the experiments, we set the value of \( n \) as 50.

**E. Optimal Patch Size**

As mentioned earlier in Section III-D, we apply \( a \times a \) patches over each of the extracted facial landmarks, and subsequently feed these patches to the encoder to extract texture information. In this experiment, we aim to find the optimal value for the parameter \( a \). To this end, we first consider \( 4 \times 4 \) patches for training FaceTopoNet on both AffectNet and FER2013 datasets. Next, we repeat the experiment by increasing \( a \) by 1. We continue this process until no performance increase is observed. Fig. 9 depicts the changes in RR with respect to the changes in \( a \) for both datasets. Fig. 9 shows that the highest RR for AffectNet and FER2013 occurs at \( a = 17 \) and \( a = 16 \), respectively. The small difference between the optimal patch sizes of the two datasets indicates low dependency of \( a \) on the dataset. For consistency, we set \( a = 17 \) throughout our experimental setup.

**F. Robustness Occlusion**

Here, we evaluate the robustness of FaceTopoNet and the state-of-the-art BreG-NeXt-50 against synthetic occlusions placed on the key areas of the face. Considering the availability of the state-of-the-art pretrained models on AffectNet and FER2013, we decide to perform this experiment on these two datasets. To this end, we extract the key components of face images, namely two eyes, mouth, and nose, using OpenCV [93] and dlib [94] libraries. We then add black patches on these key areas, and subsequently build three new validation sets for each of the AffectNet and FER2013 datasets to test our model against. We name these validation sets “eye-removed,” “mouth-removed,”
Fig. 10. Added occlusions on an image from AffectNet dataset. (a) Original image. (b) Eye-removed. (c) Mouth-removed. (d) Nose-removed.

Fig. 11. Changes of RR with regards to different types of occlusions.

Fig. 12. Face trees obtained by FaceTopoNet and Human-designed face tree borrowed from [20]. (a) Face Tree evolution during the training of FaceTopoNet. (b) Human-designed tree.

and “nose-removed.” Fig. 10 shows a sample image along with the associated occlusions. The performance of our model on both datasets is shown in Fig. 11, where FaceTopoNet delivers better RRs (0.6% average gain) compared with Breg-NeXt-50 in all the occlusion scenarios. This demonstrates the robustness of our method against the considered occlusions. In addition, this experiment reveals that mouth and eyes have approximately the same contribution in expression recognition, since “eye-removed” and “mouth-removed” scenarios show roughly the same RR drop. This finding is in line with previous studies such as [95].

G. Tree Topology Evolution

The evolution of the topology of the face trees in the training phase of FaceTopoNet can provide some additional insights into the learning process. Fig. 12 depicts the topological evolution of the minimum-cost spanning trees generated by FaceTopoNet in iterations 1 and 40 for AffectNet, FER2013, RAF-DB, and ExpW datasets. In the beginning of the training phase, as the weights of $K_n$ are random, the trees are also random. However, as the training progresses, the face trees become more resembling of the structure of a human face. Moreover, from the face trees created in iteration 40, it is evident that there exists variations between the face tree created for each dataset, particularly around nose and mouth regions. This is not unexpected given the importance of these regions in FER.

H. Cross-Dataset Study and Comparison to Human-Designed Face Tree

Here, we aim to explore whether or not the trees generated by FaceTopoNet can be used in transfer learning. To this end, we perform a cross-dataset study by using the final tree originally learned using AffectNet for expression recognition on the FER2013 dataset, and vice versa. Table VII tabulates the results where we observe an RR drop of 1.54% and 1.26% for AffectNet and FER2013 datasets, respectively. From this experiment, we conclude that while the learned trees are customized and optimized for individual datasets, given the similarities between the datasets, it is reasonably feasible to perform cross-dataset tree-based transfer learning.

Next we compare the trees generated by FaceTopoNet with human-designed trees, which have been used in other literature [20] [see Fig. 12(b)]. This tree is built based on the physiological and psychological properties of human faces. Thus, we use this tree to perform FER on AffectNet and FER2013 datasets instead of using the customized trees learned for each individual dataset. Table VII illustrates the results where we observe a drop of 4.11% and 5.4% for AffectNet and FER2013 datasets, respectively. We can, therefore, infer that FaceTopoNet-generated trees are more optimized for conducting FER for various datasets in comparison to human-designed trees.

| Dataset   | Cross-Dataset Study | RR     |
|-----------|---------------------|--------|
| AffectNet | FER2013             | 68.48  |
| FER2013   | AffectNet           | 71.40  |
| AffectNet | Human designed [20] | 65.91  |
| FER2013   | Human designed [20] | 87.26  |
TABLE VIII
ABBLATION STUDY

| Removed          | FaceTopoNet | FER2013 | AffectNet | FER2013 |
|------------------|-------------|---------|-----------|---------|
| Tree topology     | 66.51%      | 67.11%  | 5.0%      | 7.50%   |
| Structure stream | 64.90%      | 66.03%  | 7.30%     | 9.14%   |
| Texture stream   | 62.14%      | 63.17%  | 11.25%    | 13.06%  |
| Fusion strategy  | 66.87%      | 68.24%  | 4.50%     | 6.08%   |

Recognition rates (RR) and the resulting drop in RR are provided when a specific component is removed from FaceTopoNet.

TABLE IX
INFEERENCE TIME OF DIFFERENT METHODS

| Author          | Method            | Inference time (ms) |
|-----------------|-------------------|---------------------|
| Wang et al.     | Region attention  | 24                  |
| Kuo et al. [96] | Compact CNN       | 21                  |
| Ours            | FaceTopoNet       | 33                  |

I. Ablation Study

We conduct ablation experiments to evaluate the contribution of different components of our pipeline. This is done by systematic removal of the components of FaceTopoNet and calculating the resulting drop in performance. We present the results of this study in Table VIII, where

1) the tree topology learning step is removed and replaced by a random tree since the model requires a tree;
2) the texture stream is removed;
3) the structure stream is removed; and
4) the final fusion step is removed and replaced by a simple concatenation.

From the results outlined in Table VIII, we observe that the removal of the tree topology learning step results in substantial performance drop of 5.0% and 7.60%, respectively, for AffectNet and FER2013 datasets. Moreover, it is shown that while both texture and structure streams are critical toward the final performance, the texture stream plays a more critical role. This can be attributed to the higher richness of information in texture patches versus structure coordinates. Finally, the model also exhibits a drop in performance when the two-stream fusion strategy is replaced with a simple feature-level fusion, revealing the importance of conducting a proper fusion in the final step of FaceTopoNet.

J. Limitations

We perform an experiment to estimate the inference time of our method and compare it with that of prior work. To this end, we apply FaceTopoNet over 100 test images and average the corresponding inference time. Also, we find [35] and [96] which report their inference time in the context of FER. For a fair comparison, we rerun their method on our local machine with the same hardware used to record the inference time of FaceTopoNet. From Table IX, it is evident that FaceTopoNet’s runtime is higher, but only by a small margin. One of the ways by which FaceTopoNet can be improved is using faster algorithms for the calculation of the minimum-cost spanning tree.

V. Conclusion

In this article, we present a new end-to-end deep pipeline, termed FaceTopoNet, for learning the topology of face graphs for facial expression recognition systems. FaceTopoNet first extracts facial landmarks, which are then used to create a fully connected graph. The weights of the graph are then optimized to create a minimum-cost spanning tree, which is then traversed to create a sequence. This sequence is subsequently used to design the input embeddings of two sequential learners in two parallel streams within our model. The first stream relies on the coordinates of the landmarks to learn the structure of the face, whereas the second stream learns texture information using patches around each extracted landmark. The outputs of each stream are then passed through an attention mechanism to focus on more salient features, prior to being fused. We perform extensive experiments on four popular in-the-wild FER datasets, AffectNet, FER2013, RAF-DB, and ExpW. Results indicate that not only FaceTopoNet sets new state-of-the-art recognition rate in AffectNet, FER2013, and ExpW datasets, but also shows more robustness against different occlusions. We also perform detailed ablation and sensitivity studies to study our model in depth.

Future work may include using new optimization approaches, such as reinforcement learning for further optimization of the tree structure. Moreover, our approach may be used for the optimization of the structure of graph convolutional neural networks used for face representation learning.

REFERENCES

[1] N. Jaques, D. McDuff, Y. L. Kim, and R. Picard, “Understanding and predicting bonding in conversations using thin slices of facial expressions and body language,” in Proc. Int. Conf. Intell. Virtual Agents, 2016, pp. 64–74.
[2] R. W. Picard, Affective Computing. Cambridge, MA, USA: MIT Press, 2000.
[3] A. Sepas-Moghaddam, A. Etemad, F. Pereira, and P. L. Correia, “Capsfield: Light field-based face and expression recognition in the wild using capsule routing,” IEEE Trans. Image Process., vol. 30, pp. 2627–2642, 2021, doi: 10.1109/TIP.2021.3084476.
[4] R. Shavendra and A. Etemad, “Self-supervised contrastive learning of multi-view facial expressions,” in Proc. Int. Conf. Multimedia Interact., 2021, pp. 253–257.
[5] A. Sepas-Moghaddam, A. Etemad, F. Pereira, and P. L. Correia, “Facial emotion recognition using light field images with deep attention-based bidirectional lstm,” in Proc. ICASSP IEEE Int. Conf. Acoust., Speech Signal Process., 2020, pp. 3367–3371.
[6] A. Sepas-Moghaddam, A. Etemad, P. L. Correia, and F. Pereira, “A deep framework for facial emotion recognition using light field images,” in Proc. 8th Int. Conf. Affect. Comput. Interact., 2019, pp. 1–7.
[7] G. Bargheshy, X. Zhou, R. C. Deo, J. Soar, F. Whittaker, and H. Wang, “Enhanced deep learning algorithm development to detect pain intensity from facial expression images,” Expert Syst. Appl., vol. 149, 2020, Art. no. 113305.
[8] W. Pei, H. Dibeklioglu, T. Baltrušaitis, and D. M. Tax, “Attended end-to-end architecture for age estimation from facial expression videos,” IEEE Trans. Image Process., vol. 29, pp. 1972–1984, 2020.
[9] S. Porcu, A. Floris, J.-N. Voigt-Antons, L. Atzori, and S. Möller, “Estimation of the quality of experience during video streaming from facial expression and gaze direction,” IEEE Trans. Netw. Service Manag., vol. 17, no. 4, pp. 2702–2716, Dec. 2020.
[10] X. Li, W. Guo, and H. Yang, “Depression severity prediction from facial expression based on the drr_depressionnet network,” in Proc. IEEE Int. Conf. Biomet. Biomed., 2020, pp. 2757–2764.
[11] S. Li and W. Deng, “Deep facial expression recognition: A Survey,” IEEE Trans. Affect. Comput., vol. 13, no. 3, pp. 1195–1215, Jul.–Sep. 2022.

[12] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, “Joint fine-tuning in deep neural networks for facial expression recognition,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 2983–2991.

[13] D. Ghimire, S. Jeong, J. Lee, and S. H. Park, “Facial expression recognition based on local region specific features and support vector machines,” Multimedia Tools App., vol. 76, no. 6, pp. 7803–7821, 2017.

[14] S. Happy, A. Dantcheva, and F. Bremond, “Expression recognition with deep features extracted from holistic and part-based models,” Image Vis. Comput., vol. 105, 2020, Art. no. 104038.

[15] X. Liu, B. V. Kumar, P. Jia, and J. You, “Hard negative generation for identity-disentangled facial expression recognition,” Pattern Recognit., vol. 88, pp. 1–12, 2019.

[16] M. Yu, H. Zheng, Z. Peng, J. Dong, and H. Du, “Facial expression recognition based on a multi-task global-local network,” Pattern Recognit. Lett., vol. 131, pp. 166–171, 2020.

[17] X.-X. Xie, L. Lo, H.-H. Shuai, and W.-H. Cheng, “Au-assisted graph attention convolutional network for micro-expression recognition,” in Proc. ACM Int. Conf. Multimedia, 2020, pp. 2871–2880.

[18] Y. Xie, T. Chen, T. Ph, H. Wu, and L. Lin, “Adversarial graph representation adaptation for cross-domain facial expression recognition,” in Proc. ACM Int. Conf. Multimedia, 2020, pp. 1255–1264.

[19] L. Zhong, C. Bai, J. Li, T. Chen, S. Li, and Y. Liu, “A graph-structured representation with brm for static-based facial expression recognition,” in Proc. IEEE Int. Conf. Autom. Face Gesture Recognit., 2019, pp. 1–5.

[20] J. Zhou, X. Zhang, Y. Liu, and X. Lan, “Facial expression recognition using spatial-temporal semantic graph network,” in Proc. IEEE Int. Conf. Image Process., 2020, pp. 1961–1965.

[21] L. Lei, J. Li, T. Chen, and S. Li, “A novel graph-TCN with a graph structured representation for micro-expression recognition,” in Proc. ACM Int. Conf. Multimedia, 2020, pp. 2237–2245.

[22] X. Xu, Z. Ruan, and L. Yang, “Facial expression recognition based on graph neural network,” in Proc. Int. Conf. Image. Vis. Comput., 2020, pp. 211–214.

[23] Y. Liu, X. Zhang, Y. Lin, and H. Wang, “Facial expression recognition via deep action units graph network based on psychological mechanism,” IEEE Trans. Comput. Vis. Syst., vol. 12, no. 2, pp. 311–322, Jun. 2020.

[24] X. Li and X. Yao, “Cooperatively coevolving particle swarms for large scale optimization,” IEEE Trans. Evol. Comput., vol. 16, no. 2, pp. 210–224, Apr. 2012.

[25] A. Mollahosseini, B. Hasani, and M. H. Mahoor, “Affectnet: A database for facial expression, valence, and arousal computing in the wild,” IEEE Trans. Affect. Comput., vol. 10, no. 1, pp. 18–31, Jan.–Mar. 2019.

[26] I. J. Goodfellow et al., “Challenges in representation learning: A report on three machine learning contests,” in Proc. Int. Conf. Neural Inf. Process., 2013, pp. 7–17.

[27] S. Li, W. Deng, and J. Du, “Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild,” in Proc. IEEE Int. Conf. Vis. Pattern Recognit., 2017, pp. 2584–2593.

[28] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, “From facial expression recognition to interpersonal relation prediction,” Int. J. Comput. Vis., vol. 126, no. 5, pp. 550–569, 2018.

[29] M. Kolahdouzi, A. Sepas-Moghadam, and A. Etemad, “Face trees for statistical machine translation,” in Proc. Autom. Face Gesture Recognit., 2017, pp. 1023–1036.

[30] E. B. Öztürk, A. Etemad, and A. Zengin, “Gender classification using automatically selected geometric eye and eyebrow features,” J. Ambient Intell. Humanized Comput., vol. 13, no. 2, pp. 721–735, 2022.

[31] I. Ozcel, G. Yolec, C. Oz, S. Kazar, and F. Bunea, “Ifer: Facial expression recognition using automatically selected geometric eye and eyebrow features,” J. Electron. Imag., vol. 27, no. 2, 2018, Art. no. 023003.

[32] B. Hasani, P. S. Negi, and M. H. Mahoor, “Bounded residual gradient networks (breg-net) for facial affect computing,” in Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit., 2019, pp. 1–7.

[33] B. Hasani, P. S. Negi, and M. H. Mahoor, “BReg-NET: Bounded gradient affect computing using adaptive residual networks with bounded gradient,” IEEE Trans. Affect. Comput., vol. 13, no. 2, pp. 1023–1036, Apr.–Jun. 2022.

[34] D. Teney, L. Liu, and A. van Den Hengel, “Graph-structured representations for visual question answering,” in Proc. IEEE Int. Conf. Vis. Pattern Recognit., 2017, pp. 1–9.

[35] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” in Proc. ICLR, 2015, pp. 1–20.

[36] P. Antoniadis, P. P. Filntisis, and P. Maragos, “Exploiting emotional dependencies with graph convolutional networks for facial expression recognition,” in Proc. 16th IEEE Int. Conf. Automat. Face Gesture Recognit. (FG 2021), pp. 1–8, 2021.

[37] D. Lin, H. Zhang, and P. Zhou, “Video-based facial expression recognition using graph convolutional networks,” in Proc. 25th Int. Conf. Pattern Recognit., 2021, pp. 607–614.

[38] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, “Deeply learning deformable facial action parts model for dynamic expression analysis,” in Proc. Asian Conf. Comput. Vis., 2014, pp. 143–157.

[39] D. V. Sang et al., “Facial expression recognition using deep convolutional neural networks,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 2217–2225.

[40] D. Teney, L. Liu, and A. van Den Hengel, “Graph-structured representations for visual question answering,” in Proc. IEEE Int. Conf. Vis. Pattern Recognit., 2017, pp. 1–9.

[41] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” in Proc. ICLR, 2015, pp. 1–20.

[42] D. Lin, H. Zhang, and P. Zhou, “Face-to-face expression recognition using graph convolutional networks,” in Proc. 25th Int. Conf. Pattern Recognit., 2021, pp. 607–614.

[43] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, “Deeply learning deformable facial action parts model for dynamic expression analysis,” in Proc. Asian Conf. Comput. Vis., 2014, pp. 143–157.

[44] K. Cho et al., “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” in EMNLP, 2014.

[45] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou, “A deep regression architecture with two-stage re-initialization for high performance facial landmark detection,” in Proc. IEEE Int. Conf. Vis. Pattern Recognit., 2017, pp. 3317–3326.

[46] B. Jiang and L. Zhang, “Research on minimum spanning tree based on prim algorithm,” Comput. Eng. Des., vol. 30, no. 13, pp. 3244–3247, 2009.
[59] D. C. Kozen, “Depth-first and breadth-first search,” in The Design and Analysis of Algorithms. New York, NY, USA: Springer, 1992, pp. 19–24.

[60] A. Fers and J. Schmidhuber, “Recurrent nets that time and count,” in Proc. IEEE-ENNS-ENNS Int. Joint Conf. Neural Netw. IJCNN. Neural Comput.: New Challenges Perspectives New Millennium, 2000, pp. 189–194.

[61] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom, “Reasoning about entailment with neural attention,” CoRR, vol. abs/1509.06664, 2016.

[62] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “VGGFace2: A dataset for recognising faces across pose and age,” in Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit., 2018, pp. 67–74.

[63] Y. Gu, K. Yang, S. Fu, S. Chen, X. Li, and I. Marsic, “Hybrid attention based multimodal network for spoken language classification,” in Proc. 27th Int. Conf. Comput. Linguistics, 2018, pp. 2379–2390.

[64] G. Zhang and A. Etemad, “RFNet: Riemannian fusion network for eggbased brain-computer interfaces,” 2020, arXiv:2008.08633.

[65] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR, vol. abs/1412.6980, 2015.

[66] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[67] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2980–2988.

[68] M. Abadi et al., “Tensorflow: A system for large-scale machine learning,” in Proc. 12th (USENIX) Symp. Operating Syst. Design. Implementation, 2016, pp. 265–283.

[69] V. Veliuzhe, S. Pateux, and F. Jurie, “Temporal multimodal fusion for video emotion classification in the wild,” in Proc. 19th ACM Int. Conf. Multimodal Interact., 2017, pp. 569–576.

[70] 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 Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.-Workshops, 2010, pp. 94–101.

[71] C. Hewitt and H. Gunes, “CNN-based facial affect analysis on mobile devices,” 2018, arXiv:1807.08775.

[72] W. Hua, F. Dai, L. Huang, J. Xiong, and G. Gui, “Hero: Human emotions recognition for realizing intelligent Internet of Things,” IEEE Access, vol. 7, pp. 24321–24332, 2019.

[73] B. Chen, W. Guan, P. Li, N. Ibeda, K. Hirasawa, and H. Lu, “Residual multi-task learning for facial landmark localization and expression recognition,” Pattern Recognit., vol. 115, 2021, Art. no. 107893.

[74] W. Zhang, X. Ji, K. Chen, Y. Ding, and C. Fan, “Learning a facial expression embedding disentangled from identity,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 6759–6768.

[75] A. H. Farzaneh and X. Qi, “Facial expression recognition in the wild via deep attentive center loss,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., 2021, pp. 2402–2411.

[76] C. Shi, C. Tan, and L. Wang, “A facial expression recognition method based on a multibranch cross-connection convolutional neural network,” IEEE Access, vol. 9, pp. 39255–39274, 2021.

[77] A. P. Fard and M. H. Mahoor, “Ad-corre: Adaptive correlation-based loss for facial expression recognition in the wild,” IEEE Access, vol. 10, pp. 26756–26768, 2022.

[78] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process, 2013, pp. 6645–6649.

[79] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.

[80] S. Rajan, P. Chenniappan, S. Devaraj, and N. Madian, “Novel deep learning model for facial expression recognition based on maximum boosted CNN and LSTM,” IET Image Process., vol. 14, no. 7, pp. 1573–1581, 2020.

[81] Z. Liu, Y. Li, J.-H. Tao, J. Huang, and M.-Y. Liu, “Expression analysis based on face regions in real-world conditions,” Int. J. Automat. Comput., vol. 17, no. 1, pp. 96–107, 2020.

[82] H. Li, N. Wang, X. Ding, X. Yang, and X. Gao, “Adaptively learning facial expression representation via cf labels and distillation,” IEEE Trans. Image Process., vol. 30, pp. 2016–2028, 2021.

[83] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, “An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes,” Pattern Recognit., vol. 44, no. 8, pp. 1761–1776, 2011.

[84] Z. Wang, F. Zeng, S. Liu, and B. Zeng, “Oaenet: Oriented attention ensemble for accurate facial expression recognition,” Pattern Recognit., vol. 112, 2021, Art. no. 107694.

[85] K. Wang, X. Peng, J. Yang, S. Lu, and Y. Qiao, “Suppressing uncertainties for large-scale facial expression recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6897–6906.

[86] N. K. Benamara et al., “Real-time facial expression recognition using smoothed deep neural network ensemble,” Integ. Comput.-Aided Eng., vol. 28, no. 1, pp. 97–111, 2021.

[87] X. Peng, Y. Gu, and P. Zhang, “Au-guided unsupervised domain-adaptive facial expression recognition,” Appl. Sci., vol. 12, no. 9, 2022, Art. no. 4366.

[88] D. Ruan, Y. Yan, S. Chen, J.-H. Xue, and H. Wang, “Deep disturbance-disentangled learning for facial expression recognition,” in Proc. 28th ACM Int. Conf. Multimedia, 2020, pp. 2833–2841.

[89] H. Yang, U. Cifci, and L. Yin, “Facial expression recognition by de-expression residue learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2168–2177.

[90] J. Zeng, S. Shan, and X. Chen, “Facial expression recognition with inconsistently annotated datasets,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 222–237.

[91] X. Liu, B. Kumar, J. You, and B. Jia, “Adaptive deep metric learning for identity-aware facial expression recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2017, pp. 20–29.

[92] V. Mavani, S. Raman, and K. P. Miyapuram, “Facial expression recognition using visual saliency and deep learning,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2017, pp. 2783–2788.

[93] G. Bradski and A. Kaehler, “OpenCV,” Dr Dobb’s J. Softw. Tools, vol. 3, p. 120, 2000.

[94] D. E. King, “Dlib-ml: A machine learning toolkit,” J. Mach. Learn. Res., vol. 10, pp. 1755–1758, 2009.

[95] C. Koch, “The role of the eyes and mouth in facial emotions,” in Proc. 46th Annu. Meeting Psychonomic Soc. Abstracts Psychonomic Society, 2005, Art. no. 128.

[96] C.-M. Kuo, S.-H. Lai, and M. Sarkis, “A compact deep learning model for facial expression recognition based on maximum boosted CNN and LSTM,” IET Image Process., vol. 115, 2021, Art. no. 107893.